NSW Dynamic Life Cycle and Stimulus Checks Code Companion

Vegard M. Nygård, Bent E. Sørensen, and Fan Wang

2022-01-24

Contents

Preface **5**

1 Introduction **7**
- 1.1 Household Problem and Distributions . . . 7
- 1.2 Values of Checks Conditional on 2019 Information . . . 9
- 1.3 The Stimulus Check Planning Problem . . . 10

2 Parameters **13**
- 2.1 Model Parameters . . . 13
- 2.2 Model Controls . . . 41

3 Solving the Dynamic Life Cycle Problem **45**
- 3.1 Life Cycle Dynamic Programming with Marital Status, Children and Savings . . . 45

4 Alternative Value Function Solution Testing **61**
- 4.1 Small Test Exact Solution Looped Minimizer . . . 61
- 4.2 Small Test Grid Search Solution . . . 64
- 4.3 Small Test Exact Solution Vectorized Bisection . . . 78
- 4.4 Small Test Exact Solution Spousal Shocks . . . 92

5 Solution with Unemployment **107**
- 5.1 Life Cycle Dynamic Programming under Unemployment Shock . . . 107
- 5.2 Life Cycle Dynamic Programming under Great Recession Unemployment Shock . . . 125

6 Solution with First and Second Rounds CARES Stimulus **141**
- 6.1 Life Cycle Dynamic Programming under with CARES Act Stimulus Checks . . . 141

7 Household Life Cycle Distribution **161**
- 7.1 Distribution Grid Search . . . 161
- 7.2 Distribution Exact Savings Choices . . . 173
- 7.3 Distribution Exact Savings Choices Vectorized . . . 184
- 7.4 Distribution with One Period Policy Shift . . . 192

8 Value of Each Check **197**
- 8.1 2020 V and C without Unemployment . . . 197
- 8.2 2020 V and C with Unemployment . . . 209

9 2019/2020 and 2007/2008 V and C Full State Space **223**
- 9.1 2020 Full States EV and EC of One Check . . . 223
- 9.2 2019 (Biden/Trump Checks) Full States EV and EC of Two Checks . . . 242
- 9.3 2008 Value and Optimal Savings and Consumption Given Stimulus . . . 264
- 9.4 2007 (Bush 2008 Stimulus) Full States EV and EC of Two Checks . . . 284

10 2019 Expectations Given Income, Age, Kids and Marital Status **305**
- 10.1 2019 Age, Income, Kids, Marry EV and EC of One Check . . . 305
- 10.2 2019/2020 Age, Income, Kids, Marry EV and EC All Checks (Biden Checks) . . . 312
- 10.3 2007 Age, Income, Kids, Marry EV and EC All Checks (Bush Checks) . . . 316

11 Taxes and Stimulus **321**
11.1 Compute for Equilibrium Tax 321
11.2 Existing Stimulus as a Function of Income and Family Status 325
11.3 Existing Stimulus as a Function of Income and Family Status 331
11.4 Taxable Income and Tax Liabilities in 2008 336
11.5 Bush 2008 Stimulus as a Function of Income and Family Status 340

12 Calibration **347**
12.1 Model Calibration 347
12.2 Model Lockdown Calibration 349
12.3 UI Benefit Unemployment Lost Wage Recovery Parameter b Calibration 354

13 Summary Statistics **357**
13.1 2019 Full States MPC and Distributional Statistics by Marital, Kids, and Income Groups. 357

14 Selected Comparative Statistics **399**
14.1 Value and Consumption Low vs Higher Interest Rates Results Comparison 399
14.2 Value and Consumption Low vs Higher Unemployment Insurance Comparison 409
14.3 Value and Consumption Low vs Higher Discount Factor Comparison 422

A Index and Code Links **433**
A.1 ## Selected Comparative Statistics links links 433

Preface

This is a work-in-progress Matlab package consisting of functions that solve the dynamic life cycle model in Nygård, Sørensen and Wang (2021). The paper is titled Optimal allocations to heterogeneous agents with an application to the COVID-19 stimulus checks. Nygård, Sørensen and Wang (2021) supersedes two prior papers, Nygård, Sørensen and Wang (2020) as well as Wang (2020). The code companion presents solutions to the dynamic life-cycle problem, and methods for evaluating the marginal gains from allocating additional stimulus checks. Tested with Matlab 2020a (The MathWorks Inc, 2019).

All functions are parts of a matlab toolbox that can be installed:

> Download and install the Matlab toolbox: PrjOptiSNW.mltbx

The Code Companion can also be accessed via the bookdown site and PDF linked below:

> **bookdown pdf**, **MathWorks File Exchange**

This bookdown file is a collection of mlx based vignettes for functions that are available from PrjOptiSNW. Each Vignette file contains various examples for invoking each function.

The package relies on MEconTools, which needs to be installed first. The package does not include allocation functions, only simulation code to generate the value of each stimulus check increments for households. Allocation functions rely the R optimal allocation package PrjOptiAlloc.

The files below largely document contents in the Matlab-based PrjOptiSNW folder of the repository. The AllocateR folder contains additional documentation for various functions and files that solve allocation problems based on functions in the R PrjOptiAlloc package and solutions csv files generated by the dynamic programming files from the Matlab-based PrjOptiSNW folder. Some additional results for the paper are included in the additional_results folder.

The site is built using Bookdown (Xie, 2020).

Please contact FanWangEcon for issues or problems.

Chapter 1

Introduction

1.1 Household Problem and Distributions

In Nygaard, Sorensen and Wang (2020), we study the optimal allocation of COVID-19 stimulus checks as well as the 2008 Bush era stimulus checks. Congress spent $250 billion sending checks to individuals in March 2020 to provide economic stimulus. In the summer of 2008, the Bush administration sent stimulus chcks (in the form of tax rebates) to 150 million American households.

Could the same amount of stimulus have been achieved for less money? Using a life-cycle consumption-saving model with heterogeneous consumers, we calculate the consumption responses to cash transfers for, e.g., couples and singles with different levels of income and number of children. We calculate the aggregate consumption response for all feasible allocations of a stimulus checks program billion and, using a new algorithm that allows for the ranking of an arbitrarily large number of allocations, we find the optimal allocation under alternative constraints. The optimal policy allocates more toward low-income and younger consumers and can achieve the same stimulus effect at almost half the cost.

This Matlab based programming guide, package, and associated vignettes, provide examples and instructions on how the dynamic programming problem in Nygaard, Sorensen and Wang (2020) is solved. The R optimal allocation package PrjOptiAlloc takes inputs from the dynamic programming problems and solves for optimal allocations given varying planner objectives and constraints.

1.1.1 Flat Script and Code Package

There are two broad versions of the code. A number of files are included in the zflat folder, including the operation gateway file main. Files in the zflat folder provides a linear, easier to understand illustration/demonstration of the overall code structure. It is useful to review the overall algorithm design. However, it should not be called to implement the programs. Programs in the folder were written to help test out algorithm ideas.

The rest of the files inside PrjOptiSNW form a matlab package that can be downloaded and installed. Each component of the overall code program is programmed up separately with its own testing vignette and default parameter structure. Various solution algorithms are provided at each step, with the final checks problem relying on efficient and precise solution methods.

1.1.2 Dynamic Programming Solution Structure COVIDless World

First we solve for the optimal consumption/savings problem in the COVID-less world:

- **83**: 2020 or 2008 age groups, age 18 to 100 age groups
- **65**: grid of savings state-space grid, and exact continuous optimal savings choices using the **FF_VFI_AZ_BISEC_VEC** function from **MEconTools**.
- **6650 shocks**: 1330 productivity shocks for household head and sposue and 5 kids transition count shocks

- **2** permanent education states
- **2** permanent marital states

The state-space has: 2*2*6650*65*83= 143,507,000 elements. The choice-space is is continuous. Two important things two note:

1. The large number of shocks are needed to obtain accurate group-specific marginal propensity effects for small income bins that define the choice-set of the allocation problem.
2. While a choice-grid-based solution algorithm might sufficiently approximate the value function, but its policy function zig-zags. For the stimulus checks problem, where stimulus checks come in small increments, the zig-zags lead to fluctuating (negative and positive) marginal propensities to consume as resource availability increases for very small amounts of check increments. To deal with this challenge, we rely on the **FF_VFI_AZ_BISEC_VEC** function from **MEconTools** to provides efficient exact savings choices.

Solving this dynamic life-cycle programming problem requires approximately 10 to 20 minutes on a home-pc depending on computer speed. There is no processor requirements. Memory requirement is approximately 20GB. There are two core associated functions vignettes that solves the dynamic programming problem to obtain value/policy and distributions induceds by exogenous processes and the policy function:

- Core dynamic programming code: snwx_vfi_bisec_vec
- Core distribution code: snwx_ds_bisec_vec

Small testing vignettes of alternative solution algorithms for policy/value:

- Small test using matlab minimizer (very slow but identical results as core program): snwx_vfi_test
- Small test using grid-search-based solution algorithm (insufficiently precise for stimulus checks): snwx_vfi_test_grid_search
- Small test of core dynamic programming code: snwx_vfi_test_bisec_vec
- Small test of core dynamic programming code with spousal shock: snwx_vfi_test_bisec_vec_spousalshock

Testing vignettes for alternative solution algorithm for distribution:

- Grid serach distributional code (insufficiently precise) :snwx_ds_grid_search
- Core solution distribution code (vectorized for policy/value, looped for dist): snwx_ds_bisec_vec_loop
- Core solution distribution code (vectorized fully): snwx_ds_bisec_vec

1.1.3 Dynamic Programming Solution Structure during COVID Year

During the COVID year, we use the value function from the COVID-less world as the continuation value, and solve for consumption-savings policy/value functions during the COVID year. We solve once for households facing realized COVID surprise unemployment shocks, one more time for households who do not experience COVID unemployment shocks.

We solve for the marginal consumption differences and value given 244 increments of checks ($100) each check. This is done again by using the **FF_VFI_AZ_BISEC_VEC** function from **MEconTools**. While checks could be viewed as an additional state variable, we evaluate the marginal effects of check by solving for the equivalent household-specpfic variation in savings state that has the same effect as a stimulus check transfer. The process takes into account the nonlinear tax-schedule that households face as well as return on savings.

Overall:

- 286 million: Solve 143 million state-space points twice under COVID unemployment and COVID employment world
- 70 billion: Solve at the 143 million state-space elements 244 + 1 times for all possible check levels (244 checks + no check value/consumption) to arrive at 70 billion marginal propensity to consume

for households with heterogeneities in education level, marital status, children below 18 count (0 to 4), age, savings levels, household head and spouse shocks.

Associated functions vignettes: the core dynamic programming code: snwx_vfi_bisec_vec, has a third input which is the existing future value function. When this is provided, the dynamical programming problems solves for one period given already computed future value, and so the dynamic programming solution solves forward. When it is not provided, solves for value/policy backwards.

- snwx_vfi_unemp_bisec_vec provides the vignette given unemployment shock.
- snwx_a4chk_wrk_bisec_vec computes the marginal impacts of a particular stimulus check increment for those without unemployment shock in COVID year.
- snwx_a4chk_unemp_bisec_vec computes the marginal impacts of a particular stimulus check increment for those with unemployment shock in COVID year.
- snwx_evuvw20_jaeemk considers probabilities for getting hit with the COVID shock and considers the expected value conditional on age, savings level, shocks, educational status, kids count and marital status in 2020.

1.1.4 Dynamic Programming Solution Structure Bush Stimulus Checks

The Bush era stimulus checks problem is similar to the Covid problem, but there are some key differences.

1. The Bush stimulus were tax rebate, and had a more complicated schedule that is based on tax-liability.
2. The Bush stimulus were sent out prior to the unemployment shock, and hence in expectation of forthcoming shocks. In our setting, households can receive unemployment shock in 2009, and they optimize their savings/consumption decision in 2008 given this expectation. Computationally, this means the stimulus check effects do not need to be solved separately for unemployment and employed individuals as under the COVID stimulus. Instead, we solve the effects of stimulus checks on households in 2008, prior to shock realization.

More generally, stimulus checks can be given based on realized shocks or ex-ante state-space information prior to shocks. Given the information available to the IRS, which comes from the prior tax year, it seems that stimulus checks have been sent out during the Bush and Trump/Biden era based not on realized shocks, but on ex-ante information. Additionally, stimlus can be received during the period of crisis (COVID) or prior to it (Great Recession).

Our 2020 and 2008 programs rely on the same set of underling dynamic programming and distributional functions, however there are also some functions that are specific to each program year that are shown on the project webpage under headings with differing dates.

1.2 Values of Checks Conditional on 2019 Information

Eligibility for the stimulus checks was tied to each household's income and family size in the year prior to COVID-19. Consistent with this, we focus on the optimal allocation of stimulus checks given household characteristics in 2019.

1.2.1 Expected Outcomes given 2019 Information

In the actual stimulus checks allocation policy setting, 2020 realized individual level COVID shocks and other productivity shocks were not used by the IRS to determine allocation eligibility. Instead, the IRS used information available from 2019. Given the persistence of productivity shocks, as well as the correlation between surprised COVID shock and household income and age from 2019, 2019 information are good predicators of household status in 2020.

We compute the expected outcomes from 2019 perspective conditional on household attributes that are observed to the IRS in 2020 given information they gathered in 2019. The planner does not observe the full state-space so we intergrate from 2019 perspective given 2019 to 2020 COVID transition probabilities (conditional on state-space) and kids and shock transitions.

- 82 age groups: Age 18 to 99
- 5 kids groups: Children 0 to 4
- 2 marital groups: Marital Status 0 or 1
- 509 Income Groups: 476 bins below max actual phaseout: solved at $500 intervals between $0 and $238,000, and 33 bins after max actual phaseout: solved at $5000 interval after $238,000, where the 33rd final bin is between $401,130 and Maximum.

Together, these are: 5*2*83*509 = 422,470 groups/bins in 2019. We have (244+1) marginal average consumption gain, and value gain for each of the groups, so from 2019 planner perspective, we have 103.5 million expected MPCs (and expected value changes): 422470*(244+1)=103,505,150.

Each income group is composed of individuals of with different 2019 productivity shocks and savings levels. Given the transition probabilities, policy functions, and covid-less distributions across household types, we can compute the joint distribution of education type, shocks, and savings levels condition on income bins, martial status, kids count and age. This joint distribution is only well approximated when sufficient number of shocks were used when solving for value/policy in the covid-less world and covid-year world.

- snwx_evuvw19_jaeemk provides the expected outcomes conditional on 2019 age, savings levels, shocks, educational status, marital status and kids under 18 count, given the transition probabilities that incorporate the surprise COVID shock. Households do not optimize in 2019 given COVID shock probabilities, the shock is a surprise. But expected value in 2019 given 2019 state-space for consumption and value depends on COVID shock and the non-covid households dynamic consumption/savings choices in 2019.
- snwx_evuvw19_jmky provides the expected value in 2019 given not the full state-space, but the state-space that is potentially known to the IRS in 2019: age, marital status, kids count and 2019 household income. Household income is a function of savings, shock and educational status.
- snwx_evuvw19_jmky_allchecks solves for the marginal incremental effects of a vector of checks for all household types, and stores the results to file. This operation is fully parallelizable.

Given the solution, snwx_evuvw19_jaeemk_mky shows key overall distributional statistics, and distributional statistics by kids, marital and income bins.

1.2.2 Dynamic Programming Timing

Overall time requirements for solving the checks problem on a standard ($1000) dollar desktop with 4 cores:

- 30 minutes to 1 hour: solving the dynamic programming problem 3 times for covid-less world, for covid-world with unemployement shock and without.
- 3 hours: derive the distributions induced by policy functions and shock processes.
- 650 seconds: the time it takes to compute the marginal effects of one check. This step is fully scalable. With a cloud computer with larger memory and cores, the problem over all checks could be solved fully parallelly.

Due to the large state-space, especially the large shock-grid, the memory requirement for storing the various multi-dimensional matrixes is high. Solving the problem requires 20GB of memory.

On a workstation with 12 cores with 190 GB of memory (12 workers same time), the computer is able to fully solve the problem from start to finish for 244 check increments in less than 24 hours. On a computer with 4 to 6 cores and 36 GB of memory, the computer is able to fully solve the problem from start to finish for 244 check increments in about 48 hours.

1.3 The Stimulus Check Planning Problem

The planner chooses the amount of stimulus checks for each group, where groups are defined by marital status, number of children, income, and age in 2019.

1.3.1 2019 Information Planning Problem

Given the expected outcomes we computed conditional on 2019 information, we can solve the planning problem. We have a number of different planning problems that we solve given different individual level constraints and what the planner can condition allocations on.

For FEASIBLE allocation, there are **970=5*2*97** types/cells of households:

- 5 children groups
- 2 spousal groups
- 97 income bins: the allocation planner sees approximately $2500 income bins between $0 and $238,800, and 1 bin after $238,800. There are 97 bins

for OPTIMAL G4 (4 age groups 18 to 64) allocation, there are **3880=5*2*97*4** types/cells of households:

- 5 children groups
- 2 spousal groups
- 4 age groups
- 97 income bins

for OPTIMAL G47 (47 age groups) allocation, there are **45590=5*2*97*47** types/cells of households:

- 5 children groups
- 2 spousal groups
- 47 age groups
- 97 income bins

Optimal G4 has a + 1 version where we allocate for a fifth age group of individuals older than 64 years of age. Optimal G47 has a + 35 version where optimal allocation for all age groups are determined.

1.3.2 Allocation Functions

Functions in the AllocateR/alloc_discrete_fun_R folder of the project repository page is responsible for feeding the dynamic programming results into the allocation functions. The functions in this folder call the ffp_snw_process_inputs function to solve the allocation problems and compute REV, and call the ffp_snw_graph_feasible function to generate allocation graphs. These two functions are a part of the PrjOptiAlloc package.

Chapter 2

Parameters

2.1 Model Parameters

This is the example vignette for function: **snw_mp_param** from the **PrjOptiSNW Package.** This function sets and gets different parameters.

2.1.1 Parameters Used for Test Simulation

Rather than solving for all ages between 18 to 100, this solves for age groups, and has limited shocks and asset levels. Used for testing.

```
mp_params = snw_mp_param('default_small', true, 100, 6);
```

```
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_preftechpricegov Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                i     idx     value
                                --    ---    -------

    Bequests                     1     1           0
    a0                           2     2       0.258
    a1                           3     3       0.768
    a2                           4     4      1.5286
    a2_bushchkyr_2008            5     5      1.5286
    a2_covidyr                   6     6         NaN
    a2_covidyr_manna_heaven      7     7      1.5286
    a2_covidyr_tax_fully_pay     8     8      12.718
    a2_greatrecession_2009       9     9      1.5286
    bequests_option             10    10           1
    beta                        11    11     0.86389
    cons_allocation_rule        12    12           2
    g_cons                      13    13     0.17576
    g_n                         14    14     0.05101
    gamma                       15    15           2
    invbtlock                   16    16           1
    it_yrs_per_period           17    17           5
    jret                        18    18          13
    r                           19    19     0.21665
    theta                       20    20     0.56523
    throw_in_ocean              21    21           1

----------------------------------------
```

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_intlen Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

	i	idx	value
n_agrid	1	1	25
n_educgrid	2	2	2
n_eta_H_grid	3	3	5
n_eta_S_grid	4	4	1
n_etagrid	5	5	5
n_jgrid	6	6	18
n_kidsgrid	7	7	3
n_marriedgrid	8	8	2

----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

	i	idx	ndim	numel	rowN	colN	sum	mean	st
inc_grid	1	7	2	201	201	1	578.5	2.8781	1.
pi_unemp	2	10	2	240	48	5	47.034	0.19598	0.09
pi_unemp_2009_edu_age	3	11	2	96	48	2	6.6005	0.068755	0.04
pi_unemp_2020_april	4	12	2	240	48	5	47.034	0.19598	0.09
pi_unemp_2020_juneadj	5	13	2	240	48	5	16.17	0.067373	0.03

xxx TABLE:inc_grid xxxxxxxxxxxxxxxxxx

	c1
r1	0
r2	0.026667
r3	0.053333
r4	0.08
r5	0.10667
r6	0.13333
r7	0.16
r8	0.18667
r9	0.21333
r10	0.24
r11	0.26667
r12	0.29333
r13	0.32
r14	0.34667
r15	0.37333
r16	0.4
r17	0.42667
r18	0.45333
r19	0.48
r20	0.50667
r21	0.53333
r22	0.56
r23	0.58667
r24	0.61333
r25	0.64
r26	0.66667

r27	0.69333
r28	0.72
r29	0.74667
r30	0.77333
r31	0.8
r32	0.82667
r33	0.85333
r34	0.88
r35	0.90667
r36	0.93333
r37	0.96
r38	0.98667
r39	1.0133
r40	1.04
r41	1.0667
r42	1.0933
r43	1.12
r44	1.1467
r45	1.1733
r46	1.2
r47	1.2267
r48	1.2533
r49	1.28
r50	1.3067
r152	4.06
r153	4.12
r154	4.18
r155	4.24
r156	4.3
r157	4.36
r158	4.42
r159	4.48
r160	4.54
r161	4.6
r162	4.66
r163	4.72
r164	4.78
r165	4.84
r166	4.9
r167	4.96
r168	5.02
r169	5.08
r170	5.14
r171	5.2
r172	5.26
r173	5.32
r174	5.38
r175	5.44
r176	5.5
r177	5.56
r178	5.62
r179	5.68
r180	5.74
r181	5.8
r182	5.86
r183	5.92
r184	5.98
r185	6.04

r186	6.1
r187	6.16
r188	6.22
r189	6.28
r190	6.34
r191	6.4
r192	6.46
r193	6.52
r194	6.58
r195	6.64
r196	6.7
r197	6.76
r198	6.82
r199	6.88
r200	6.94
r201	7

xxx TABLE:pi_unemp xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0.36194	0.22237	0.17262	0.14265	0.083133
r2	0.36194	0.22237	0.17262	0.14265	0.083133
r3	0.36194	0.22237	0.17262	0.14265	0.083133
r4	0.36194	0.22237	0.17262	0.14265	0.083133
r5	0.36194	0.22237	0.17262	0.14265	0.083133
r6	0.36194	0.22237	0.17262	0.14265	0.083133
r7	0.36194	0.22237	0.17262	0.14265	0.083133
r8	0.36194	0.22237	0.17262	0.14265	0.083133
r9	0.36194	0.22237	0.17262	0.14265	0.083133
r10	0.36194	0.22237	0.17262	0.14265	0.083133
r11	0.36194	0.22237	0.17262	0.14265	0.083133
r12	0.36194	0.22237	0.17262	0.14265	0.083133
r13	0.36194	0.22237	0.17262	0.14265	0.083133
r14	0.3534	0.21383	0.16408	0.13411	0.074592
r15	0.3534	0.21383	0.16408	0.13411	0.074592
r16	0.3534	0.21383	0.16408	0.13411	0.074592
r17	0.3534	0.21383	0.16408	0.13411	0.074592
r18	0.3534	0.21383	0.16408	0.13411	0.074592
r19	0.3534	0.21383	0.16408	0.13411	0.074592
r20	0.3534	0.21383	0.16408	0.13411	0.074592
r21	0.3534	0.21383	0.16408	0.13411	0.074592
r22	0.3534	0.21383	0.16408	0.13411	0.074592
r23	0.3534	0.21383	0.16408	0.13411	0.074592
r24	0.34917	0.2096	0.15984	0.12988	0.070361
r25	0.34917	0.2096	0.15984	0.12988	0.070361
r26	0.34917	0.2096	0.15984	0.12988	0.070361
r27	0.34917	0.2096	0.15984	0.12988	0.070361
r28	0.34917	0.2096	0.15984	0.12988	0.070361
r29	0.34917	0.2096	0.15984	0.12988	0.070361
r30	0.34917	0.2096	0.15984	0.12988	0.070361
r31	0.34917	0.2096	0.15984	0.12988	0.070361
r32	0.34917	0.2096	0.15984	0.12988	0.070361
r33	0.34917	0.2096	0.15984	0.12988	0.070361
r34	0.35656	0.21699	0.16723	0.13727	0.077749
r35	0.35656	0.21699	0.16723	0.13727	0.077749
r36	0.35656	0.21699	0.16723	0.13727	0.077749
r37	0.35656	0.21699	0.16723	0.13727	0.077749

r38	0.35656	0.21699	0.16723	0.13727	0.077749
r39	0.35656	0.21699	0.16723	0.13727	0.077749
r40	0.35656	0.21699	0.16723	0.13727	0.077749
r41	0.35656	0.21699	0.16723	0.13727	0.077749
r42	0.35656	0.21699	0.16723	0.13727	0.077749
r43	0.35656	0.21699	0.16723	0.13727	0.077749
r44	0.40989	0.27032	0.22056	0.1906	0.13108
r45	0.40989	0.27032	0.22056	0.1906	0.13108
r46	0.40989	0.27032	0.22056	0.1906	0.13108
r47	0.40989	0.27032	0.22056	0.1906	0.13108
r48	0.40989	0.27032	0.22056	0.1906	0.13108

xxx TABLE:pi_unemp_2009_edu_age xxxxxxxxxxxxxxxxxx

	c1	c2
r1	0.17919	0.12517
r2	0.17919	0.12517
r3	0.17919	0.12517
r4	0.17919	0.12517
r5	0.17919	0.12517
r6	0.17919	0.12517
r7	0.17919	0.12517
r8	0.086103	0.032088
r9	0.086103	0.032088
r10	0.086103	0.032088
r11	0.086103	0.032088
r12	0.086103	0.032088
r13	0.086103	0.032088
r14	0.086103	0.032088
r15	0.086103	0.032088
r16	0.086103	0.032088
r17	0.086103	0.032088
r18	0.086103	0.032088
r19	0.086103	0.032088
r20	0.086103	0.032088
r21	0.086103	0.032088
r22	0.086103	0.032088
r23	0.086103	0.032088
r24	0.086103	0.032088
r25	0.086103	0.032088
r26	0.086103	0.032088
r27	0.086103	0.032088
r28	0.086103	0.032088
r29	0.086103	0.032088
r30	0.086103	0.032088
r31	0.086103	0.032088
r32	0.086103	0.032088
r33	0.086103	0.032088
r34	0.086103	0.032088
r35	0.086103	0.032088
r36	0.086103	0.032088
r37	0.086103	0.032088
r38	0.06902	0.015005
r39	0.06902	0.015005
r40	0.06902	0.015005
r41	0.06902	0.015005
r42	0.06902	0.015005

r43	0.06902	0.015005
r44	0.06902	0.015005
r45	0.06902	0.015005
r46	0.06902	0.015005
r47	0.06902	0.015005
r48	0.06902	0.015005

xxx TABLE:pi_unemp_2020_april xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0.36194	0.22237	0.17262	0.14265	0.083133
r2	0.36194	0.22237	0.17262	0.14265	0.083133
r3	0.36194	0.22237	0.17262	0.14265	0.083133
r4	0.36194	0.22237	0.17262	0.14265	0.083133
r5	0.36194	0.22237	0.17262	0.14265	0.083133
r6	0.36194	0.22237	0.17262	0.14265	0.083133
r7	0.36194	0.22237	0.17262	0.14265	0.083133
r8	0.36194	0.22237	0.17262	0.14265	0.083133
r9	0.36194	0.22237	0.17262	0.14265	0.083133
r10	0.36194	0.22237	0.17262	0.14265	0.083133
r11	0.36194	0.22237	0.17262	0.14265	0.083133
r12	0.36194	0.22237	0.17262	0.14265	0.083133
r13	0.36194	0.22237	0.17262	0.14265	0.083133
r14	0.3534	0.21383	0.16408	0.13411	0.074592
r15	0.3534	0.21383	0.16408	0.13411	0.074592
r16	0.3534	0.21383	0.16408	0.13411	0.074592
r17	0.3534	0.21383	0.16408	0.13411	0.074592
r18	0.3534	0.21383	0.16408	0.13411	0.074592
r19	0.3534	0.21383	0.16408	0.13411	0.074592
r20	0.3534	0.21383	0.16408	0.13411	0.074592
r21	0.3534	0.21383	0.16408	0.13411	0.074592
r22	0.3534	0.21383	0.16408	0.13411	0.074592
r23	0.3534	0.21383	0.16408	0.13411	0.074592
r24	0.34917	0.2096	0.15984	0.12988	0.070361
r25	0.34917	0.2096	0.15984	0.12988	0.070361
r26	0.34917	0.2096	0.15984	0.12988	0.070361
r27	0.34917	0.2096	0.15984	0.12988	0.070361
r28	0.34917	0.2096	0.15984	0.12988	0.070361
r29	0.34917	0.2096	0.15984	0.12988	0.070361
r30	0.34917	0.2096	0.15984	0.12988	0.070361
r31	0.34917	0.2096	0.15984	0.12988	0.070361
r32	0.34917	0.2096	0.15984	0.12988	0.070361
r33	0.34917	0.2096	0.15984	0.12988	0.070361
r34	0.35656	0.21699	0.16723	0.13727	0.077749
r35	0.35656	0.21699	0.16723	0.13727	0.077749
r36	0.35656	0.21699	0.16723	0.13727	0.077749
r37	0.35656	0.21699	0.16723	0.13727	0.077749
r38	0.35656	0.21699	0.16723	0.13727	0.077749
r39	0.35656	0.21699	0.16723	0.13727	0.077749
r40	0.35656	0.21699	0.16723	0.13727	0.077749
r41	0.35656	0.21699	0.16723	0.13727	0.077749
r42	0.35656	0.21699	0.16723	0.13727	0.077749
r43	0.35656	0.21699	0.16723	0.13727	0.077749
r44	0.40989	0.27032	0.22056	0.1906	0.13108
r45	0.40989	0.27032	0.22056	0.1906	0.13108
r46	0.40989	0.27032	0.22056	0.1906	0.13108
r47	0.40989	0.27032	0.22056	0.1906	0.13108

```
    r48    0.40989    0.27032    0.22056     0.1906     0.13108

xxx TABLE:pi_unemp_2020_juneadj xxxxxxxxxxxxxxxxxx
             c1          c2          c3          c4          c5
          -------    --------    --------    --------    --------

    r1     0.11257    0.062283    0.046026    0.036173    0.035471
    r2     0.11257    0.062283    0.046026    0.036173    0.035471
    r3     0.11257    0.062283    0.046026    0.036173    0.035471
    r4     0.11257    0.062283    0.046026    0.036173    0.035471
    r5     0.11257    0.062283    0.046026    0.036173    0.035471
    r6     0.11257    0.062283    0.046026    0.036173    0.035471
    r7     0.11257    0.062283    0.046026    0.036173    0.035471
    r8     0.11257    0.062283    0.046026    0.036173    0.035471
    r9     0.11257    0.062283    0.046026    0.036173    0.035471
    r10    0.11257    0.062283    0.046026    0.036173    0.035471
    r11    0.11257    0.062283    0.046026    0.036173    0.035471
    r12    0.11257    0.062283    0.046026    0.036173    0.035471
    r13    0.11257    0.062283    0.046026    0.036173    0.035471
    r14    0.11994    0.069654    0.053397    0.043545    0.042842
    r15    0.11994    0.069654    0.053397    0.043545    0.042842
    r16    0.11994    0.069654    0.053397    0.043545    0.042842
    r17    0.11994    0.069654    0.053397    0.043545    0.042842
    r18    0.11994    0.069654    0.053397    0.043545    0.042842
    r19    0.11994    0.069654    0.053397    0.043545    0.042842
    r20    0.11994    0.069654    0.053397    0.043545    0.042842
    r21    0.11994    0.069654    0.053397    0.043545    0.042842
    r22    0.11994    0.069654    0.053397    0.043545    0.042842
    r23    0.11994    0.069654    0.053397    0.043545    0.042842
    r24    0.11038    0.060097     0.04384    0.033988    0.033285
    r25    0.11038    0.060097     0.04384    0.033988    0.033285
    r26    0.11038    0.060097     0.04384    0.033988    0.033285
    r27    0.11038    0.060097     0.04384    0.033988    0.033285
    r28    0.11038    0.060097     0.04384    0.033988    0.033285
    r29    0.11038    0.060097     0.04384    0.033988    0.033285
    r30    0.11038    0.060097     0.04384    0.033988    0.033285
    r31    0.11038    0.060097     0.04384    0.033988    0.033285
    r32    0.11038    0.060097     0.04384    0.033988    0.033285
    r33    0.11038    0.060097     0.04384    0.033988    0.033285
    r34    0.12326    0.072969    0.056712     0.04686    0.046157
    r35    0.12326    0.072969    0.056712     0.04686    0.046157
    r36    0.12326    0.072969    0.056712     0.04686    0.046157
    r37    0.12326    0.072969    0.056712     0.04686    0.046157
    r38    0.12326    0.072969    0.056712     0.04686    0.046157
    r39    0.12326    0.072969    0.056712     0.04686    0.046157
    r40    0.12326    0.072969    0.056712     0.04686    0.046157
    r41    0.12326    0.072969    0.056712     0.04686    0.046157
    r42    0.12326    0.072969    0.056712     0.04686    0.046157
    r43    0.12326    0.072969    0.056712     0.04686    0.046157
    r44    0.16597     0.11568    0.099422     0.08957    0.088867
    r45    0.16597     0.11568    0.099422     0.08957    0.088867
    r46    0.16597     0.11568    0.099422     0.08957    0.088867
    r47    0.16597     0.11568    0.099422     0.08957    0.088867
    r48    0.16597     0.11568    0.099422     0.08957    0.088867

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy Scalars
```

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                i     idx      value
                                --    ---    ---------

    TR                          1     1     0.0015999
    b                           2     2             1
    fl_stimulus_adult_first     3     3          1200
    fl_stimulus_adult_second    4     4           600
    fl_stimulus_child_first     5     5           500
    fl_stimulus_child_second    6     6           600
    n_incgrid                   7     8           201
    n_welfchecksgrid            8     9            45
    scaleconvertor              9    14         62502
    xi                         10    16          0.75

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy String
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                          i     idx               string
                         ---    ----    -----------------------------

    st_biden_or_trump    "1"    "15"    "st_biden_or_trump_undefined"

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_statesgrid ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                  i    idx    ndim    numel    rowN    colN        sum           mean         std
                  -    ---    ----    -----    ----    ----    -----------    -----------    ------

    agrid         1     1      2       25       25      1          878.91         35.156    41.372
    eta_H_grid    2     2      2        5        5      1     -2.2204e-16    -4.4409e-17    1.4543
    eta_S_grid    3     3      2        5        5      1               0              0         0

xxx TABLE:agrid xxxxxxxxxxxxxxxxxx
              c1
           ---------

    r1             0
    r2     0.0097656
    r3      0.078125
    r4       0.26367
    r5         0.625
    r6        1.2207
    r7        2.1094
    r8        3.3496
    r9             5
    r10       7.1191
    r11       9.7656
    r12       12.998
    r13       16.875
    r14       21.455
    r15       26.797
    r16       32.959
    r17           40
    r18       47.979
    r19       56.953
```

```
    r20       66.982
    r21       78.125
    r22       90.439
    r23       103.98
    r24       118.82
    r25          135

xxx TABLE:eta_H_grid xxxxxxxxxxxxxxxxxx
             c1
          --------

    r1     -1.8395
    r2    -0.91976
    r3           0
    r4     0.91976
    r5      1.8395

xxx TABLE:eta_S_grid xxxxxxxxxxxxxxxxxx
          c1
          --

    r1    0
    r2    0
    r3    0
    r4    0
    r5    0

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_exotrans ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                            i    idx    ndim    numel    rowN    colN     sum       mean        std
                            -    ---    ----    -----    ----    ----    ------    -------    ------

    cl_mt_pi_jem_kidseta    1     2      2         1       1       1         0          0
    pi_H_eta                2     3      2        25       5       5         5        0.2     0.3851
    pi_eta                  3     5      2        25       5       5         5        0.2     0.3851
    pi_kids                 4     6      5       648       3     216       216    0.33333     0.3561
    psi                     5     7      2        18      18       1    14.251    0.79171     0.3125

xxx TABLE:cl_mt_pi_jem_kidseta xxxxxxxxxxxxxxxxxx
          c1
          --

    r1    0

xxx TABLE:pi_H_eta xxxxxxxxxxxxxxxxxx
             c1            c2            c3           c4            c5
          ----------    ----------    ----------    ---------    ----------

    r1         0.925      0.075001    4.8068e-10            0             0
    r2     0.0026569       0.96788      0.029459    2.602e-11             0
    r3    1.1558e-12     0.0096913       0.98062    0.0096913    1.1559e-12
    r4       1.28e-29     2.602e-11      0.029459      0.96788     0.0026569
    r5    2.8504e-54    1.8802e-27    4.8068e-10     0.075001         0.925

xxx TABLE:pi_eta xxxxxxxxxxxxxxxxxx
             c1            c2            c3           c4            c5
```

```
       ----------    ----------    ----------    ---------    ----------

   r1        0.925      0.075001    4.8068e-10            0             0
   r2    0.0026569       0.96788      0.029459    2.602e-11             0
   r3   1.1558e-12     0.0096913       0.98062    0.0096913    1.1559e-12
   r4     1.28e-29     2.602e-11      0.029459      0.96788     0.0026569
   r5   2.8504e-54    1.8802e-27    4.8068e-10     0.075001         0.925

xxx TABLE:pi_kids xxxxxxxxxxxxxxxxxx
          c1           c2          c3        c214    c215    c216

       ---------    --------    ---------    ----    ----    ----

   r1      0.88584     0.11137    0.0027905     1       0       0
   r2     0.051343     0.66234      0.28632     1       0       0
   r3    0.0015025    0.063309      0.93519     1       0       0

xxx TABLE:psi xxxxxxxxxxxxxxxxxx
          c1

        -------

   r1     0.99935
   r2     0.99623
   r3     0.99635
   r4     0.99537
   r5     0.99299
   r6     0.98956
   r7     0.98547
   r8     0.98022
   r9     0.96914
   r10    0.95071
   r11    0.92082
   r12    0.87772
   r13    0.81394
   r14    0.70638
   r15    0.54032
   r16    0.34767
   r17    0.18848
   r18          0

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_exotrans Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                           i    idx    value

                           -    ---    -----

   bl_store_shock_trans    1     1       0
   pi_S_eta                2     4       1

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_typelife ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
              i    idx    ndim    numel    rowN    colN     sum       mean       std      coefvari

              -    ---    ----    -----    ----    ----    ------    ------    -------    --------

   SS         1     1      2       36       18      2       2.916     0.081    0.11695     1.4439
   epsilon    2     2      2       36       18      2      39.526    1.0979    0.85451    0.77828
```

```
xxx TABLE:SS xxxxxxxxxxxxxxxxxx
          c1      c2
         ----    -----

    r1       0       0
    r2       0       0
    r3       0       0
    r4       0       0
    r5       0       0
    r6       0       0
    r7       0       0
    r8       0       0
    r9       0       0
    r10      0       0
    r11      0       0
    r12      0       0
    r13   0.22   0.266
    r14   0.22   0.266
    r15   0.22   0.266
    r16   0.22   0.266
    r17   0.22   0.266
    r18   0.22   0.266

xxx TABLE:epsilon xxxxxxxxxxxxxxxxxx
           c1        c2
         ------    ------

    r1         1         1
    r2    1.0778    1.1836
    r3    1.2546    1.6124
    r4     1.397    1.9418
    r5    1.5022    2.1452
    r6    1.5712    2.2394
    r7    1.6064    2.2588
    r8    1.6097    2.2341
    r9    1.5815     2.182
    r10   1.5204    2.1034
    r11   1.4243    1.9846
    r12   1.2917    1.8041
    r13        0         0
    r14        0         0
    r15        0         0
    r16        0         0
    r17        0         0
    r18        0         0

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_stat ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                          i    idx    ndim    numel    rowN    colN     sum      mean       std
                          -    ---    ----    -----    ----    ----    ------   -------   --------

    Pop                   1     1      2       18       18       1     9.8945   0.54969    0.31889
    stat_distr_educ       2     3      2        2        1       2          1       0.5     0.2786
    stat_distr_eta        3     4      2        5        1       5          1       0.2    0.24003
    stat_distr_kids       4     5      3       12        2       6          4   0.33333    0.33166
```

```
    stat_distr_married    5     6      2        4        2      2            2        0.5    0.073381

xxx TABLE:Pop xxxxxxxxxxxxxxxxxx
              c1
           --------

    r1           1
    r2     0.95085
    r3     0.90129
    r4     0.85442
    r5     0.80919
    r6     0.76452
    r7     0.71982
    r8     0.67493
    r9     0.62947
    r10    0.58044
    r11    0.52505
    r12    0.46001
    r13    0.38416
    r14    0.29751
    r15    0.19995
    r16     0.1028
    r17   0.034004
    r18   0.006098

xxx TABLE:stat_distr_educ xxxxxxxxxxxxxxxxxx
           c1       c2
          -----    -----

    r1    0.697    0.303

xxx TABLE:stat_distr_eta xxxxxxxxxxxxxxxxxx
             c1          c2         c3         c4          c5
          ---------    -------    -------    -------    ---------

    r1    0.0069316    0.19567    0.59479    0.19567    0.0069316

xxx TABLE:stat_distr_kids xxxxxxxxxxxxxxxxxx
            c1         c2          c3         c4           c5           c6
          -------    -------    --------    -------    ----------    --------

    r1    0.75801    0.44877      0.1564    0.32041       0.08559     0.23083
    r2    0.97627     0.7604    0.023626     0.2173    0.00010011    0.022305

xxx TABLE:stat_distr_married xxxxxxxxxxxxxxxxxx
            c1        c2
          ------    ------

    r1    0.5635    0.4365
    r2    0.4364    0.5636

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_stat String
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                          i     idx                           string
                         ---    ---    --------------------------------------------------------
```

```
    st_old_age_depend    "1"    "2"    "Old-age dependency ratio (ratio of 65+/(18-64))=0.1155"
```

2.1.2 Documentation Run Parameters Docdense

Parameters used for documentation vig. "docdense" uses less shocks than the version of the model used to implement the allocation problems in the Nygaard, Sorensen and Wang (2020).

```
mp_params = snw_mp_param('default_docdense', true, 100, 6);
```

```
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_preftechpricegov Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                 i     idx      value
                                 --    ---     -------

    Bequests                      1     1            0
    a0                            2     2        0.258
    a1                            3     3        0.768
    a2                            4     4       1.5286
    a2_bushchkyr_2008             5     5       1.5286
    a2_covidyr                    6     6          NaN
    a2_covidyr_manna_heaven       7     7       1.5286
    a2_covidyr_tax_fully_pay      8     8       12.718
    a2_greatrecession_2009        9     9       1.5286
    bequests_option              10    10            1
    beta                         11    11      0.97116
    cons_allocation_rule         12    12            2
    g_cons                       13    13      0.17576
    g_n                          14    14         0.01
    gamma                        15    15            2
    invbtlock                    16    16            1
    it_yrs_per_period            17    17            1
    jret                         18    18           48
    r                            19    19         0.04
    theta                        20    20      0.56523
    throw_in_ocean               21    21            1

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_intlen Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                      i     idx     value
                      -     ---     -----

    n_agrid           1      1        65
    n_educgrid        2      2         2
    n_eta_H_grid      3      3        81
    n_eta_S_grid      4      4         5
    n_etagrid         5      5       405
    n_jgrid           6      6        83
    n_kidsgrid        7      7         5
    n_marriedgrid     8      8         2

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                              i    idx    ndim    numel    rowN    colN     sum      mean      st
```

```
                          -    ---    ----    -----    ----    ----    ------    --------    ----

    inc_grid              1     7      2       201      201      1      578.5      2.8781      1.
    pi_unemp              2    10      2       415       83      5     47.034     0.11333     0.1
    pi_unemp_2009_edu_age 3    11      2       166       83      2     6.6005    0.039762       0
    pi_unemp_2020_april   4    12      2       415       83      5     47.034     0.11333     0.1
    pi_unemp_2020_juneadj 5    13      2       415       83      5      16.17    0.038963    0.04

xxx TABLE:inc_grid xxxxxxxxxxxxxxxxxx
              c1
           --------

    r1            0
    r2     0.026667
    r3     0.053333
    r4         0.08
    r5      0.10667
    r6      0.13333
    r7         0.16
    r8      0.18667
    r9      0.21333
    r10        0.24
    r11     0.26667
    r12     0.29333
    r13        0.32
    r14     0.34667
    r15     0.37333
    r16         0.4
    r17     0.42667
    r18     0.45333
    r19        0.48
    r20     0.50667
    r21     0.53333
    r22        0.56
    r23     0.58667
    r24     0.61333
    r25        0.64
    r26     0.66667
    r27     0.69333
    r28        0.72
    r29     0.74667
    r30     0.77333
    r31         0.8
    r32     0.82667
    r33     0.85333
    r34        0.88
    r35     0.90667
    r36     0.93333
    r37        0.96
    r38     0.98667
    r39      1.0133
    r40        1.04
    r41      1.0667
    r42      1.0933
    r43        1.12
    r44      1.1467
    r45      1.1733
    r46         1.2
```

```
    r47       1.2267
    r48       1.2533
    r49         1.28
    r50       1.3067
    r152        4.06
    r153        4.12
    r154        4.18
    r155        4.24
    r156         4.3
    r157        4.36
    r158        4.42
    r159        4.48
    r160        4.54
    r161         4.6
    r162        4.66
    r163        4.72
    r164        4.78
    r165        4.84
    r166         4.9
    r167        4.96
    r168        5.02
    r169        5.08
    r170        5.14
    r171         5.2
    r172        5.26
    r173        5.32
    r174        5.38
    r175        5.44
    r176         5.5
    r177        5.56
    r178        5.62
    r179        5.68
    r180        5.74
    r181         5.8
    r182        5.86
    r183        5.92
    r184        5.98
    r185        6.04
    r186         6.1
    r187        6.16
    r188        6.22
    r189        6.28
    r190        6.34
    r191         6.4
    r192        6.46
    r193        6.52
    r194        6.58
    r195        6.64
    r196         6.7
    r197        6.76
    r198        6.82
    r199        6.88
    r200        6.94
    r201           7

xxx TABLE:pi_unemp xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4          c5
           -------    -------    -------    -------    --------
```

r1	0.36194	0.22237	0.17262	0.14265	0.083133
r2	0.36194	0.22237	0.17262	0.14265	0.083133
r3	0.36194	0.22237	0.17262	0.14265	0.083133
r4	0.36194	0.22237	0.17262	0.14265	0.083133
r5	0.36194	0.22237	0.17262	0.14265	0.083133
r6	0.36194	0.22237	0.17262	0.14265	0.083133
r7	0.36194	0.22237	0.17262	0.14265	0.083133
r8	0.36194	0.22237	0.17262	0.14265	0.083133
r9	0.36194	0.22237	0.17262	0.14265	0.083133
r10	0.36194	0.22237	0.17262	0.14265	0.083133
r11	0.36194	0.22237	0.17262	0.14265	0.083133
r12	0.36194	0.22237	0.17262	0.14265	0.083133
r13	0.36194	0.22237	0.17262	0.14265	0.083133
r14	0.3534	0.21383	0.16408	0.13411	0.074592
r15	0.3534	0.21383	0.16408	0.13411	0.074592
r16	0.3534	0.21383	0.16408	0.13411	0.074592
r17	0.3534	0.21383	0.16408	0.13411	0.074592
r18	0.3534	0.21383	0.16408	0.13411	0.074592
r19	0.3534	0.21383	0.16408	0.13411	0.074592
r20	0.3534	0.21383	0.16408	0.13411	0.074592
r21	0.3534	0.21383	0.16408	0.13411	0.074592
r22	0.3534	0.21383	0.16408	0.13411	0.074592
r23	0.3534	0.21383	0.16408	0.13411	0.074592
r24	0.34917	0.2096	0.15984	0.12988	0.070361
r25	0.34917	0.2096	0.15984	0.12988	0.070361
r26	0.34917	0.2096	0.15984	0.12988	0.070361
r27	0.34917	0.2096	0.15984	0.12988	0.070361
r28	0.34917	0.2096	0.15984	0.12988	0.070361
r29	0.34917	0.2096	0.15984	0.12988	0.070361
r30	0.34917	0.2096	0.15984	0.12988	0.070361
r31	0.34917	0.2096	0.15984	0.12988	0.070361
r32	0.34917	0.2096	0.15984	0.12988	0.070361
r33	0.34917	0.2096	0.15984	0.12988	0.070361
r34	0.35656	0.21699	0.16723	0.13727	0.077749
r35	0.35656	0.21699	0.16723	0.13727	0.077749
r36	0.35656	0.21699	0.16723	0.13727	0.077749
r37	0.35656	0.21699	0.16723	0.13727	0.077749
r38	0.35656	0.21699	0.16723	0.13727	0.077749
r39	0.35656	0.21699	0.16723	0.13727	0.077749
r40	0.35656	0.21699	0.16723	0.13727	0.077749
r41	0.35656	0.21699	0.16723	0.13727	0.077749
r42	0.35656	0.21699	0.16723	0.13727	0.077749
r43	0.35656	0.21699	0.16723	0.13727	0.077749
r44	0.40989	0.27032	0.22056	0.1906	0.13108
r45	0.40989	0.27032	0.22056	0.1906	0.13108
r46	0.40989	0.27032	0.22056	0.1906	0.13108
r47	0.40989	0.27032	0.22056	0.1906	0.13108
r48	0.40989	0.27032	0.22056	0.1906	0.13108
r49	0	0	0	0	0
r50	0	0	0	0	0
r51	0	0	0	0	0
r52	0	0	0	0	0
r53	0	0	0	0	0
r54	0	0	0	0	0
r55	0	0	0	0	0
r56	0	0	0	0	0
r57	0	0	0	0	0

r58	0	0	0	0	0
r59	0	0	0	0	0
r60	0	0	0	0	0
r61	0	0	0	0	0
r62	0	0	0	0	0
r63	0	0	0	0	0
r64	0	0	0	0	0
r65	0	0	0	0	0
r66	0	0	0	0	0
r67	0	0	0	0	0
r68	0	0	0	0	0
r69	0	0	0	0	0
r70	0	0	0	0	0
r71	0	0	0	0	0
r72	0	0	0	0	0
r73	0	0	0	0	0
r74	0	0	0	0	0
r75	0	0	0	0	0
r76	0	0	0	0	0
r77	0	0	0	0	0
r78	0	0	0	0	0
r79	0	0	0	0	0
r80	0	0	0	0	0
r81	0	0	0	0	0
r82	0	0	0	0	0
r83	0	0	0	0	0

xxx TABLE:pi_unemp_2009_edu_age xxxxxxxxxxxxxxxxxx

	c1	c2
	--------	--------
r1	0.17919	0.12517
r2	0.17919	0.12517
r3	0.17919	0.12517
r4	0.17919	0.12517
r5	0.17919	0.12517
r6	0.17919	0.12517
r7	0.17919	0.12517
r8	0.086103	0.032088
r9	0.086103	0.032088
r10	0.086103	0.032088
r11	0.086103	0.032088
r12	0.086103	0.032088
r13	0.086103	0.032088
r14	0.086103	0.032088
r15	0.086103	0.032088
r16	0.086103	0.032088
r17	0.086103	0.032088
r18	0.086103	0.032088
r19	0.086103	0.032088
r20	0.086103	0.032088
r21	0.086103	0.032088
r22	0.086103	0.032088
r23	0.086103	0.032088
r24	0.086103	0.032088
r25	0.086103	0.032088
r26	0.086103	0.032088
r27	0.086103	0.032088

r28	0.086103	0.032088
r29	0.086103	0.032088
r30	0.086103	0.032088
r31	0.086103	0.032088
r32	0.086103	0.032088
r33	0.086103	0.032088
r34	0.086103	0.032088
r35	0.086103	0.032088
r36	0.086103	0.032088
r37	0.086103	0.032088
r38	0.06902	0.015005
r39	0.06902	0.015005
r40	0.06902	0.015005
r41	0.06902	0.015005
r42	0.06902	0.015005
r43	0.06902	0.015005
r44	0.06902	0.015005
r45	0.06902	0.015005
r46	0.06902	0.015005
r47	0.06902	0.015005
r48	0.06902	0.015005
r49	0	0
r50	0	0
r51	0	0
r52	0	0
r53	0	0
r54	0	0
r55	0	0
r56	0	0
r57	0	0
r58	0	0
r59	0	0
r60	0	0
r61	0	0
r62	0	0
r63	0	0
r64	0	0
r65	0	0
r66	0	0
r67	0	0
r68	0	0
r69	0	0
r70	0	0
r71	0	0
r72	0	0
r73	0	0
r74	0	0
r75	0	0
r76	0	0
r77	0	0
r78	0	0
r79	0	0
r80	0	0
r81	0	0
r82	0	0
r83	0	0

xxx TABLE:pi_unemp_2020_april xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0.36194	0.22237	0.17262	0.14265	0.083133
r2	0.36194	0.22237	0.17262	0.14265	0.083133
r3	0.36194	0.22237	0.17262	0.14265	0.083133
r4	0.36194	0.22237	0.17262	0.14265	0.083133
r5	0.36194	0.22237	0.17262	0.14265	0.083133
r6	0.36194	0.22237	0.17262	0.14265	0.083133
r7	0.36194	0.22237	0.17262	0.14265	0.083133
r8	0.36194	0.22237	0.17262	0.14265	0.083133
r9	0.36194	0.22237	0.17262	0.14265	0.083133
r10	0.36194	0.22237	0.17262	0.14265	0.083133
r11	0.36194	0.22237	0.17262	0.14265	0.083133
r12	0.36194	0.22237	0.17262	0.14265	0.083133
r13	0.36194	0.22237	0.17262	0.14265	0.083133
r14	0.3534	0.21383	0.16408	0.13411	0.074592
r15	0.3534	0.21383	0.16408	0.13411	0.074592
r16	0.3534	0.21383	0.16408	0.13411	0.074592
r17	0.3534	0.21383	0.16408	0.13411	0.074592
r18	0.3534	0.21383	0.16408	0.13411	0.074592
r19	0.3534	0.21383	0.16408	0.13411	0.074592
r20	0.3534	0.21383	0.16408	0.13411	0.074592
r21	0.3534	0.21383	0.16408	0.13411	0.074592
r22	0.3534	0.21383	0.16408	0.13411	0.074592
r23	0.3534	0.21383	0.16408	0.13411	0.074592
r24	0.34917	0.2096	0.15984	0.12988	0.070361
r25	0.34917	0.2096	0.15984	0.12988	0.070361
r26	0.34917	0.2096	0.15984	0.12988	0.070361
r27	0.34917	0.2096	0.15984	0.12988	0.070361
r28	0.34917	0.2096	0.15984	0.12988	0.070361
r29	0.34917	0.2096	0.15984	0.12988	0.070361
r30	0.34917	0.2096	0.15984	0.12988	0.070361
r31	0.34917	0.2096	0.15984	0.12988	0.070361
r32	0.34917	0.2096	0.15984	0.12988	0.070361
r33	0.34917	0.2096	0.15984	0.12988	0.070361
r34	0.35656	0.21699	0.16723	0.13727	0.077749
r35	0.35656	0.21699	0.16723	0.13727	0.077749
r36	0.35656	0.21699	0.16723	0.13727	0.077749
r37	0.35656	0.21699	0.16723	0.13727	0.077749
r38	0.35656	0.21699	0.16723	0.13727	0.077749
r39	0.35656	0.21699	0.16723	0.13727	0.077749
r40	0.35656	0.21699	0.16723	0.13727	0.077749
r41	0.35656	0.21699	0.16723	0.13727	0.077749
r42	0.35656	0.21699	0.16723	0.13727	0.077749
r43	0.35656	0.21699	0.16723	0.13727	0.077749
r44	0.40989	0.27032	0.22056	0.1906	0.13108
r45	0.40989	0.27032	0.22056	0.1906	0.13108
r46	0.40989	0.27032	0.22056	0.1906	0.13108
r47	0.40989	0.27032	0.22056	0.1906	0.13108
r48	0.40989	0.27032	0.22056	0.1906	0.13108
r49	0	0	0	0	0
r50	0	0	0	0	0
r51	0	0	0	0	0
r52	0	0	0	0	0
r53	0	0	0	0	0
r54	0	0	0	0	0
r55	0	0	0	0	0

r56	0	0	0	0	0
r57	0	0	0	0	0
r58	0	0	0	0	0
r59	0	0	0	0	0
r60	0	0	0	0	0
r61	0	0	0	0	0
r62	0	0	0	0	0
r63	0	0	0	0	0
r64	0	0	0	0	0
r65	0	0	0	0	0
r66	0	0	0	0	0
r67	0	0	0	0	0
r68	0	0	0	0	0
r69	0	0	0	0	0
r70	0	0	0	0	0
r71	0	0	0	0	0
r72	0	0	0	0	0
r73	0	0	0	0	0
r74	0	0	0	0	0
r75	0	0	0	0	0
r76	0	0	0	0	0
r77	0	0	0	0	0
r78	0	0	0	0	0
r79	0	0	0	0	0
r80	0	0	0	0	0
r81	0	0	0	0	0
r82	0	0	0	0	0
r83	0	0	0	0	0

xxx TABLE:pi_unemp_2020_juneadj xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0.11257	0.062283	0.046026	0.036173	0.035471
r2	0.11257	0.062283	0.046026	0.036173	0.035471
r3	0.11257	0.062283	0.046026	0.036173	0.035471
r4	0.11257	0.062283	0.046026	0.036173	0.035471
r5	0.11257	0.062283	0.046026	0.036173	0.035471
r6	0.11257	0.062283	0.046026	0.036173	0.035471
r7	0.11257	0.062283	0.046026	0.036173	0.035471
r8	0.11257	0.062283	0.046026	0.036173	0.035471
r9	0.11257	0.062283	0.046026	0.036173	0.035471
r10	0.11257	0.062283	0.046026	0.036173	0.035471
r11	0.11257	0.062283	0.046026	0.036173	0.035471
r12	0.11257	0.062283	0.046026	0.036173	0.035471
r13	0.11257	0.062283	0.046026	0.036173	0.035471
r14	0.11994	0.069654	0.053397	0.043545	0.042842
r15	0.11994	0.069654	0.053397	0.043545	0.042842
r16	0.11994	0.069654	0.053397	0.043545	0.042842
r17	0.11994	0.069654	0.053397	0.043545	0.042842
r18	0.11994	0.069654	0.053397	0.043545	0.042842
r19	0.11994	0.069654	0.053397	0.043545	0.042842
r20	0.11994	0.069654	0.053397	0.043545	0.042842
r21	0.11994	0.069654	0.053397	0.043545	0.042842
r22	0.11994	0.069654	0.053397	0.043545	0.042842
r23	0.11994	0.069654	0.053397	0.043545	0.042842
r24	0.11038	0.060097	0.04384	0.033988	0.033285
r25	0.11038	0.060097	0.04384	0.033988	0.033285

r26	0.11038	0.060097	0.04384	0.033988	0.033285
r27	0.11038	0.060097	0.04384	0.033988	0.033285
r28	0.11038	0.060097	0.04384	0.033988	0.033285
r29	0.11038	0.060097	0.04384	0.033988	0.033285
r30	0.11038	0.060097	0.04384	0.033988	0.033285
r31	0.11038	0.060097	0.04384	0.033988	0.033285
r32	0.11038	0.060097	0.04384	0.033988	0.033285
r33	0.11038	0.060097	0.04384	0.033988	0.033285
r34	0.12326	0.072969	0.056712	0.04686	0.046157
r35	0.12326	0.072969	0.056712	0.04686	0.046157
r36	0.12326	0.072969	0.056712	0.04686	0.046157
r37	0.12326	0.072969	0.056712	0.04686	0.046157
r38	0.12326	0.072969	0.056712	0.04686	0.046157
r39	0.12326	0.072969	0.056712	0.04686	0.046157
r40	0.12326	0.072969	0.056712	0.04686	0.046157
r41	0.12326	0.072969	0.056712	0.04686	0.046157
r42	0.12326	0.072969	0.056712	0.04686	0.046157
r43	0.12326	0.072969	0.056712	0.04686	0.046157
r44	0.16597	0.11568	0.099422	0.08957	0.088867
r45	0.16597	0.11568	0.099422	0.08957	0.088867
r46	0.16597	0.11568	0.099422	0.08957	0.088867
r47	0.16597	0.11568	0.099422	0.08957	0.088867
r48	0.16597	0.11568	0.099422	0.08957	0.088867
r49	0	0	0	0	0
r50	0	0	0	0	0
r51	0	0	0	0	0
r52	0	0	0	0	0
r53	0	0	0	0	0
r54	0	0	0	0	0
r55	0	0	0	0	0
r56	0	0	0	0	0
r57	0	0	0	0	0
r58	0	0	0	0	0
r59	0	0	0	0	0
r60	0	0	0	0	0
r61	0	0	0	0	0
r62	0	0	0	0	0
r63	0	0	0	0	0
r64	0	0	0	0	0
r65	0	0	0	0	0
r66	0	0	0	0	0
r67	0	0	0	0	0
r68	0	0	0	0	0
r69	0	0	0	0	0
r70	0	0	0	0	0
r71	0	0	0	0	0
r72	0	0	0	0	0
r73	0	0	0	0	0
r74	0	0	0	0	0
r75	0	0	0	0	0
r76	0	0	0	0	0
r77	0	0	0	0	0
r78	0	0	0	0	0
r79	0	0	0	0	0
r80	0	0	0	0	0
r81	0	0	0	0	0
r82	0	0	0	0	0
r83	0	0	0	0	0

```
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                i     idx      value
                                --    ---    ---------

    TR                          1     1     0.0015999
    b                           2     2             1
    fl_stimulus_adult_first     3     3          1200
    fl_stimulus_adult_second    4     4           600
    fl_stimulus_child_first     5     5           500
    fl_stimulus_child_second    6     6           600
    n_incgrid                   7     8           201
    n_welfchecksgrid            8     9            45
    scaleconvertor              9    14         62502
    xi                         10    16          0.75

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy String
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                          i      idx              string
                         ---    ----    -----------------------------

    st_biden_or_trump    "1"    "15"    "st_biden_or_trump_undefined"

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_statesgrid ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                  i    idx    ndim    numel    rowN    colN        sum            mean          std
                  -    ---    ----    -----    ----    ----    -----------    -----------    ------

    agrid         1     1      2        65      65      1             2228         34.277    39.432
    eta_H_grid    2     2      2       405     405      1        9.015e-14     2.2259e-16    1.5783
    eta_S_grid    3     3      2       405     405      1      -1.7764e-14    -4.3861e-17    2.2103

xxx TABLE:agrid xxxxxxxxxxxxxxxxxx
              c1
          ----------

    r1             0
    r2    0.00051498
    r3     0.0041199
    r4      0.013905
    r5      0.032959
    r6      0.064373
    r7       0.11124
    r8       0.17664
    r9       0.26367
    r10      0.37542
    r11      0.51498
    r12      0.68544
    r13      0.88989
    r14       1.1314
    r15       1.4131
```

```
    r16         1.7381
    r17         2.1094
    r18         2.5301
    r19         3.0034
    r20         3.5323
    r21         4.1199
    r22         4.7693
    r23         5.4836
    r24         6.2658
    r25         7.1191
    r26         8.0466
    r27         9.0514
    r28         10.136
    r29         11.305
    r30          12.56
    r31         13.905
    r32         15.342
    r33         16.875
    r34         18.507
    r35         20.241
    r36          22.08
    r37         24.027
    r38         26.085
    r39         28.258
    r40         30.548
    r41         32.959
    r42         35.493
    r43         38.154
    r44         40.945
    r45         43.868
    r46         46.928
    r47         50.126
    r48         53.467
    r49         56.953
    r50         60.587
    r51         64.373
    r52         68.313
    r53         72.411
    r54         76.669
    r55         81.091
    r56          85.68
    r57         90.439
    r58         95.371
    r59         100.48
    r60         105.77
    r61         111.24
    r62         116.89
    r63         122.74
    r64         128.77
    r65            135

xxx TABLE:eta_H_grid xxxxxxxxxxxxxxxxxx
                c1
             ----------

    r1          -2.6968
    r2          -2.6294
    r3           -2.562
```

```
r4          -2.4945
r5          -2.4271
r6          -2.3597
r7          -2.2923
r8          -2.2249
r9          -2.1574
r10           -2.09
r11         -2.0226
r12         -1.9552
r13         -1.8878
r14         -1.8203
r15         -1.7529
r16         -1.6855
r17         -1.6181
r18         -1.5507
r19         -1.4832
r20         -1.4158
r21         -1.3484
r22          -1.281
r23         -1.2136
r24         -1.1461
r25         -1.0787
r26         -1.0113
r27        -0.94388
r28        -0.87646
r29        -0.80904
r30        -0.74162
r31         -0.6742
r32        -0.60678
r33        -0.53936
r34        -0.47194
r35        -0.40452
r36         -0.3371
r37        -0.26968
r38        -0.20226
r39        -0.13484
r40        -0.06742
r41     2.2204e-16
r42         0.06742
r43         0.13484
r44         0.20226
r45         0.26968
r46          0.3371
r47         0.40452
r48         0.47194
r49         0.53936
r50         0.60678
r356       -0.60678
r357       -0.53936
r358       -0.47194
r359       -0.40452
r360        -0.3371
r361       -0.26968
r362       -0.20226
r363       -0.13484
r364       -0.06742
r365    2.2204e-16
r366        0.06742
```

```
    r367       0.13484
    r368       0.20226
    r369       0.26968
    r370        0.3371
    r371       0.40452
    r372       0.47194
    r373       0.53936
    r374       0.60678
    r375        0.6742
    r376       0.74162
    r377       0.80904
    r378       0.87646
    r379       0.94388
    r380        1.0113
    r381        1.0787
    r382        1.1461
    r383        1.2136
    r384         1.281
    r385        1.3484
    r386        1.4158
    r387        1.4832
    r388        1.5507
    r389        1.6181
    r390        1.6855
    r391        1.7529
    r392        1.8203
    r393        1.8878
    r394        1.9552
    r395        2.0226
    r396          2.09
    r397        2.1574
    r398        2.2249
    r399        2.2923
    r400        2.3597
    r401        2.4271
    r402        2.4945
    r403         2.562
    r404        2.6294
    r405        2.6968

xxx TABLE:eta_S_grid xxxxxxxxxxxxxxxxxx
             c1
            ------

    r1      -3.122
    r2      -3.122
    r3      -3.122
    r4      -3.122
    r5      -3.122
    r6      -3.122
    r7      -3.122
    r8      -3.122
    r9      -3.122
    r10     -3.122
    r11     -3.122
    r12     -3.122
    r13     -3.122
    r14     -3.122
```

```
r15     -3.122
r16     -3.122
r17     -3.122
r18     -3.122
r19     -3.122
r20     -3.122
r21     -3.122
r22     -3.122
r23     -3.122
r24     -3.122
r25     -3.122
r26     -3.122
r27     -3.122
r28     -3.122
r29     -3.122
r30     -3.122
r31     -3.122
r32     -3.122
r33     -3.122
r34     -3.122
r35     -3.122
r36     -3.122
r37     -3.122
r38     -3.122
r39     -3.122
r40     -3.122
r41     -3.122
r42     -3.122
r43     -3.122
r44     -3.122
r45     -3.122
r46     -3.122
r47     -3.122
r48     -3.122
r49     -3.122
r50     -3.122
r356     3.122
r357     3.122
r358     3.122
r359     3.122
r360     3.122
r361     3.122
r362     3.122
r363     3.122
r364     3.122
r365     3.122
r366     3.122
r367     3.122
r368     3.122
r369     3.122
r370     3.122
r371     3.122
r372     3.122
r373     3.122
r374     3.122
r375     3.122
r376     3.122
r377     3.122
```

```
    r378     3.122
    r379     3.122
    r380     3.122
    r381     3.122
    r382     3.122
    r383     3.122
    r384     3.122
    r385     3.122
    r386     3.122
    r387     3.122
    r388     3.122
    r389     3.122
    r390     3.122
    r391     3.122
    r392     3.122
    r393     3.122
    r394     3.122
    r395     3.122
    r396     3.122
    r397     3.122
    r398     3.122
    r399     3.122
    r400     3.122
    r401     3.122
    r402     3.122
    r403     3.122
    r404     3.122
    r405     3.122

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_exotrans ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean
cl_mt_pi_jem_kidseta	1	2	2	1	1	1	0	0
pi_H_eta	2	3	2	6561	81	81	81	0.012346
pi_S_eta	3	4	2	25	5	5	5	0.2
pi_eta	4	5	2	1.6403e+05	405	405	405	0.0024691
pi_kids	5	6	5	8300	5	1660	1660	0.2
psi	6	7	2	83	83	1	78.16	0.94169

xxx TABLE:cl_mt_pi_jem_kidseta xxxxxxxxxxxxxxxxxx

	c1
r1	0

xxx TABLE:pi_H_eta xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c79	c80	c81
r1	0.44008	0.19741	0.16603	0	0	0
r2	0.26004	0.18401	0.1972	0	0	0
r3	0.12804	0.13527	0.18471	0	0	0
r4	0.051745	0.078413	0.13644	0	0	0
r5	0.016976	0.035843	0.079479	0	0	0

r6	0.0044863	0.012918	0.036507	0	0	0
r7	0.00094957	0.0036704	0.013221	0	0	0
r8	0.00016032	0.00082204	0.0037748	0	0	0
r9	2.1522e-05	0.0001451	0.00084955	0	0	0
r10	2.2921e-06	2.0182e-05	0.00015069	0	0	0
r11	1.933e-07	2.2115e-06	2.1061e-05	0	0	0
r12	1.2891e-08	1.9089e-07	2.3192e-06	0	0	0
r13	6.7901e-10	1.2976e-08	2.0116e-07	0	0	0
r14	2.8225e-11	6.9453e-10	1.3741e-08	0	0	0
r15	9.2521e-13	2.9264e-11	7.3906e-10	0	0	0
r16	2.3901e-14	9.7051e-13	3.1293e-11	0	0	0
r17	4.8636e-16	2.5328e-14	1.0429e-12	0	0	0
r18	7.7924e-18	5.2007e-16	2.735e-14	0	0	0
r19	9.8265e-20	8.4004e-18	5.6434e-16	0	0	0
r20	9.7502e-22	1.0672e-19	9.1603e-18	0	0	0
r21	7.6101e-24	1.0662e-21	1.1695e-19	0	0	0
r22	4.6713e-26	8.3759e-24	1.1741e-21	0	0	0
r23	2.2546e-28	5.1729e-26	9.269e-24	0	0	0
r24	8.5548e-31	2.5114e-28	5.7527e-26	0	0	0
r25	2.5514e-33	9.583e-31	2.8066e-28	0	0	0
r26	5.9805e-36	2.8738e-33	1.0762e-30	0	0	0
r27	1.1016e-38	6.7725e-36	3.2434e-33	0	0	0
r28	1.5943e-41	1.2541e-38	7.6811e-36	0	0	0
r29	1.8129e-44	1.8245e-41	1.4293e-38	0	0	0
r30	1.6194e-47	2.0853e-44	2.0897e-41	0	0	0
r31	1.1364e-50	1.8723e-47	2.4002e-44	0	0	0
r32	6.2635e-54	1.3205e-50	2.1657e-47	0	0	0
r33	2.7115e-57	7.3149e-54	1.535e-50	0	0	0
r34	9.2192e-61	3.1826e-57	8.5451e-54	0	0	0
r35	2.4617e-64	1.0875e-60	3.7362e-57	0	0	0
r36	5.1617e-68	2.9183e-64	1.283e-60	0	0	0
r37	8.4992e-72	6.1497e-68	3.4599e-64	0	0	0
r38	1.0989e-75	1.0176e-71	7.327e-68	0	0	0
r39	1.1156e-79	1.3223e-75	1.2185e-71	0	0	0
r40	8.8927e-84	1.3491e-79	1.5911e-75	0	0	0
r41	5.5655e-88	1.0807e-83	1.6313e-79	0	0	0
r42	2.7347e-92	6.7971e-88	1.3133e-83	0	0	0
r43	1.055e-96	3.3564e-92	8.3007e-88	0	0	0
r44	3.1951e-101	1.3012e-96	4.1192e-92	0	0	0
r45	7.5967e-106	3.9605e-101	1.6049e-96	0	0	0
r46	1.418e-110	9.4631e-106	4.9088e-101	0	0	0
r47	2.0777e-115	1.7751e-110	1.1787e-105	0	0	0
r48	2.3898e-120	2.6138e-115	2.2219e-110	0	0	0
r49	2.1579e-125	3.0215e-120	3.2881e-115	0	0	0
r50	1.5294e-130	2.7417e-125	3.8196e-120	0	0	0
r51	8.5093e-136	1.9529e-130	3.4831e-125	0	0	0
r52	3.7162e-141	1.0919e-135	2.4933e-130	0	0	0
r53	1.2739e-146	4.7921e-141	1.401e-135	0	0	0
r54	3.4277e-152	1.6509e-146	6.179e-141	0	0	0
r55	7.2393e-158	4.4641e-152	2.1392e-146	0	0	0
r56	1.2001e-163	9.4748e-158	5.8132e-152	0	0	0
r57	1.5615e-169	1.5784e-163	1.2399e-157	0	0	0
r58	1.5947e-175	2.064e-169	2.0759e-163	0	0	0
r59	1.2782e-181	2.1183e-175	2.7279e-169	0	0	0
r60	8.0416e-188	1.7064e-181	2.8135e-175	0	0	0
r61	3.9708e-194	1.0788e-187	2.2776e-181	0	0	0
r62	1.5389e-200	5.3534e-194	1.4472e-187	0	0	0
r63	4.6807e-207	2.085e-200	7.2168e-194	5.5511e-16	0	0

```
    r64    1.1174e-213    6.3733e-207    2.8246e-200    2.7311e-14    5.5511e-16             0
    r65    2.0936e-220     1.529e-213     8.677e-207    1.0428e-12    2.5424e-14    4.4409e-16
    r66    3.0785e-227    2.8789e-220     2.092e-213    3.1293e-11    9.7056e-13     2.387e-14
    r67    3.5527e-234    4.2543e-227    3.9585e-220    7.3906e-10    2.9264e-11    9.2526e-13
    r68    3.2178e-241     4.934e-234    5.8786e-227    1.3741e-08    6.9453e-10    2.8225e-11
    r69    2.2873e-248     4.491e-241    6.8517e-234    2.0116e-07    1.2976e-08    6.7901e-10
    r70     1.276e-255    3.2082e-248    6.2674e-241    2.3192e-06    1.9089e-07    1.2891e-08
    r71    5.5866e-263    1.7986e-255    4.4993e-248    2.1061e-05    2.2115e-06     1.933e-07
    r72    1.9196e-270    7.9137e-263     2.535e-255    0.00015069    2.0182e-05    2.2921e-06
    r73    5.1762e-278    2.7326e-270    1.1209e-262    0.00084955     0.0001451    2.1522e-05
    r74    1.0954e-285    7.4052e-278    3.8897e-270     0.0037748    0.00082204    0.00016032
    r75    1.8193e-293    1.5749e-285    1.0593e-277      0.013221     0.0036704    0.00094957
    r76    2.3712e-301    2.6286e-293     2.264e-285      0.036507      0.012918     0.0044863
    r77    2.4254e-309     3.443e-301    3.7975e-293      0.079479      0.035843      0.016976
    r78    1.9469e-317    3.5392e-309    4.9987e-301       0.13644      0.078413      0.051745
    r79              0    2.8551e-317    5.1639e-309       0.18471       0.13527       0.12804
    r80              0              0    4.1864e-317        0.1972       0.18401       0.26004
    r81              0              0              0       0.16603       0.19741       0.44008

xxx TABLE:pi_S_eta xxxxxxxxxxxxxxxxxx
             c1         c2        c3         c4         c5    
          --------    ------    -------    ------    --------

    r1    0.012224    0.2144    0.54675    0.2144    0.012224
    r2    0.012224    0.2144    0.54675    0.2144    0.012224
    r3    0.012224    0.2144    0.54675    0.2144    0.012224
    r4    0.012224    0.2144    0.54675    0.2144    0.012224
    r5    0.012224    0.2144    0.54675    0.2144    0.012224

xxx TABLE:pi_eta xxxxxxxxxxxxxxxxxx
                c1             c2             c3            c403          c404          c405   
            -----------    -----------    -----------    ----------    ----------    ----------

    r1   ...
```

2.1.3 Parameters Used for Paper Simulations

Full version of parameters used in Nygaard, Sorensen and Wang (2020). This is not printed to save space.

```
% mp_params = snw_mp_param('default_moredense_a65zh266zs5_e2m2', true, 100, 6);
```

2.2 Model Controls

This is the example vignette for function: **snw_mp_control** from the **PrjOptiSNW Package.** This function sets and gets different control parameters.

2.2.1 Test SNW_MP_CONTROLS Defaults

Call the function with defaults.

```
mp_controls = snw_mp_control('default_base', true);

pos = 43 ; key = options
  fmincon options:

   Options used by current Algorithm ('interior-point'):
   (Other available algorithms: 'active-set', 'sqp', 'sqp-legacy', 'trust-region-reflective')
```

```
   Set properties:
                     Display: 'off'

   Default properties:
                   Algorithm: 'interior-point'
              CheckGradients: 0
         ConstraintTolerance: 1.0000e-06
    FiniteDifferenceStepSize: 'sqrt(eps)'
        FiniteDifferenceType: 'forward'
        HessianApproximation: 'bfgs'
                  HessianFcn: []
          HessianMultiplyFcn: []
                 HonorBounds: 1
      MaxFunctionEvaluations: 3000
               MaxIterations: 1000
              ObjectiveLimit: -1.0000e+20
         OptimalityTolerance: 1.0000e-06
                   OutputFcn: []
                     PlotFcn: []
                ScaleProblem: 0
   SpecifyConstraintGradient: 0
    SpecifyObjectiveGradient: 0
               StepTolerance: 1.0000e-10
         SubproblemAlgorithm: 'factorization'
                    TypicalX: 'ones(numberOfVariables,1)'
                 UseParallel: 0

   Show options not used by current Algorithm ('interior-point')

pos = 44 ; key = options2
  fsolve options:

   Options used by current Algorithm ('trust-region-dogleg'):
   (Other available algorithms: 'levenberg-marquardt', 'trust-region')

   Set properties:
                    Display: 'off'

   Default properties:
                  Algorithm: 'trust-region-dogleg'
             CheckGradients: 0
   FiniteDifferenceStepSize: 'sqrt(eps)'
       FiniteDifferenceType: 'forward'
          FunctionTolerance: 1.0000e-06
     MaxFunctionEvaluations: '100*numberOfVariables'
              MaxIterations: 400
        OptimalityTolerance: 1.0000e-06
                  OutputFcn: []
                    PlotFcn: []
   SpecifyObjectiveGradient: 0
              StepTolerance: 1.0000e-06
                   TypicalX: 'ones(numberOfVariables,1)'
                UseParallel: 0

   Show options not used by current Algorithm ('trust-region-dogleg')

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
CONTAINER NAME: mp_controls Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                              i    idx    value
                                              --   ---    -----

    A_aux                                      1     1      NaN
    Aeq                                        2     2      NaN
    B_aux                                      3     3      NaN
    Beq                                        4     4      NaN
    bl_compute_drv_stats                       5     5        1
    bl_print_a4chk                             6     6        1
    bl_print_a4chk_verbose                     7     7        0
    bl_print_calibrate_2009                    8     8        1
    bl_print_calibrate_2009_verbose            9     9        0
    bl_print_ds                               10    10        1
    bl_print_ds_aggregation                   11    11        1
    bl_print_ds_aggregation_verbose           12    12        0
    bl_print_ds_verbose                       13    13        0
    bl_print_evuvw19_jaeemk                   14    14        1
    bl_print_evuvw19_jaeemk_verbose           15    15        0
    bl_print_evuvw19_jmky                     16    16        1
    bl_print_evuvw19_jmky_allchecks           17    17        1
    bl_print_evuvw19_jmky_allchecks_verbose   18    18        0
    bl_print_evuvw19_jmky_mass                19    19        1
    bl_print_evuvw19_jmky_mass_verbose        20    20        0
    bl_print_evuvw19_jmky_verbose             21    21        0
    bl_print_evuvw20_jaeemk                   22    22        1
    bl_print_evuvw20_jaeemk_verbose           23    23        0
    bl_print_find_tax_rate                    24    24        1
    bl_print_find_tax_rate_verbose            25    25        0
    bl_print_precompute                       26    26        1
    bl_print_precompute_verbose               27    27        0
    bl_print_v08_jaeemk                       28    28        1
    bl_print_v08_jaeemk_verbose               29    29        0
    bl_print_v08p08_jaeemk                    30    30        1
    bl_print_v08p08_jaeemk_verbose            31    31        0
    bl_print_v_planner                        32    32        1
    bl_print_v_planner_verbose                33    33        0
    bl_print_vfi                              34    34        1
    bl_print_vfi_verbose                      35    35        0
    bl_print_vu_vw                            36    36        1
    bl_print_vu_vw_verbose                    37    37        0
    bl_timer                                  38    38        1
    err                                       39    39        1
    fl_max_trchk_perc_increase                40    40      1.5
    nonlcon                                   41    42      NaN
    tol                                       42    45    0.005

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_controls String
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                       i      idx         string
                      ---    ----    --------------

    mp_params_name    "1"    "41"    "default_base"
```

Chapter 3

Solving the Dynamic Life Cycle Problem

3.1 Life Cycle Dynamic Programming with Marital Status, Children and Savings

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for policy function with vectorized bisection. Value function during COVIDless year.

3.1.1 Test SNW_VFI_MAIN_BISECT_VEC Defaults

Call the function with defaults.

```
mp_param = snw_mp_param('default_docdense');
[V_VFI,ap_VFI,cons_VFI] = snw_vfi_main_bisec_vec(mp_param);
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:83 of 82, time-this-age:9.6616
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:82 of 82, time-this-age:6.0665
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:81 of 82, time-this-age:6.0938
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:80 of 82, time-this-age:6.1322
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:79 of 82, time-this-age:5.731
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:78 of 82, time-this-age:5.9587
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:77 of 82, time-this-age:6.0286
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:76 of 82, time-this-age:6.0378
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:75 of 82, time-this-age:5.7227
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:74 of 82, time-this-age:6.0347
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:73 of 82, time-this-age:6.0197
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:72 of 82, time-this-age:6.0294
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:71 of 82, time-this-age:5.7663
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:70 of 82, time-this-age:5.7878
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:69 of 82, time-this-age:5.9087
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:68 of 82, time-this-age:5.9625
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:67 of 82, time-this-age:5.9427
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:66 of 82, time-this-age:5.526
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:65 of 82, time-this-age:5.9574
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:64 of 82, time-this-age:5.9754
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:63 of 82, time-this-age:5.9528
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:62 of 82, time-this-age:5.7483
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:61 of 82, time-this-age:6.0225
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:60 of 82, time-this-age:6.091
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:59 of 82, time-this-age:6.0448
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:58 of 82, time-this-age:6.0445
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:57 of 82, time-this-age:5.5331
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:56 of 82, time-this-age:6.0133
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:55 of 82, time-this-age:5.9281
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:54 of 82, time-this-age:5.9348
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:53 of 82, time-this-age:5.576
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:52 of 82, time-this-age:5.9723
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:51 of 82, time-this-age:6.1292
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:50 of 82, time-this-age:5.983
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:49 of 82, time-this-age:5.8518
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:48 of 82, time-this-age:5.6741
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:47 of 82, time-this-age:6.0199
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:46 of 82, time-this-age:6.0128
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:45 of 82, time-this-age:6.2299
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:44 of 82, time-this-age:5.8858
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:43 of 82, time-this-age:6.0074
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:42 of 82, time-this-age:6.1082
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:41 of 82, time-this-age:6.1896
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:40 of 82, time-this-age:5.9484
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:39 of 82, time-this-age:5.7184
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:38 of 82, time-this-age:6.0237
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:37 of 82, time-this-age:6.0886
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:36 of 82, time-this-age:5.9999
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:35 of 82, time-this-age:5.8859
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:34 of 82, time-this-age:6.089
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:33 of 82, time-this-age:6.0487
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:32 of 82, time-this-age:5.972
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:31 of 82, time-this-age:6.0053
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:30 of 82, time-this-age:5.671
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:29 of 82, time-this-age:5.8975
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:28 of 82, time-this-age:6.0471
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:27 of 82, time-this-age:6.0284
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:26 of 82, time-this-age:6.03
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:25 of 82, time-this-age:6.0227
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:24 of 82, time-this-age:5.9344
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:23 of 82, time-this-age:6.0962
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:22 of 82, time-this-age:6.1112
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:21 of 82, time-this-age:5.6128
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:20 of 82, time-this-age:6.0994
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:19 of 82, time-this-age:5.9906
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:18 of 82, time-this-age:5.9374
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:17 of 82, time-this-age:5.9326
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:16 of 82, time-this-age:5.6032
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:15 of 82, time-this-age:6.0086
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:14 of 82, time-this-age:6.0024
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:13 of 82, time-this-age:6.0767
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:12 of 82, time-this-age:5.8031
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:11 of 82, time-this-age:6.012
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:10 of 82, time-this-age:6.0142
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:9 of 82, time-this-age:5.9887
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:8 of 82, time-this-age:6.0433
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:7 of 82, time-this-age:5.5764
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:6 of 82, time-this-age:5.9101
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:5 of 82, time-this-age:5.9695
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:4 of 82, time-this-age:5.9063
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:3 of 82, time-this-age:5.7645
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:2 of 82, time-this-age:5.8967
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:1 of 82, time-this-age:6.0414
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_base;time=497.
```

3.1.2 Define Parameters

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

3.1.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

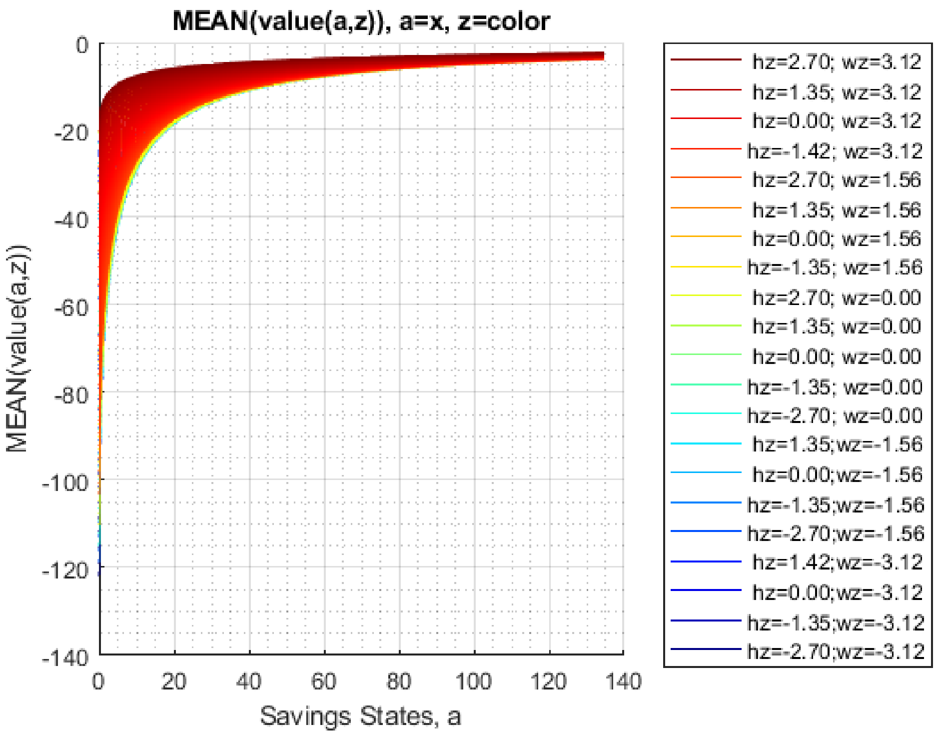
Tabulate value and policies along savings and shocks:

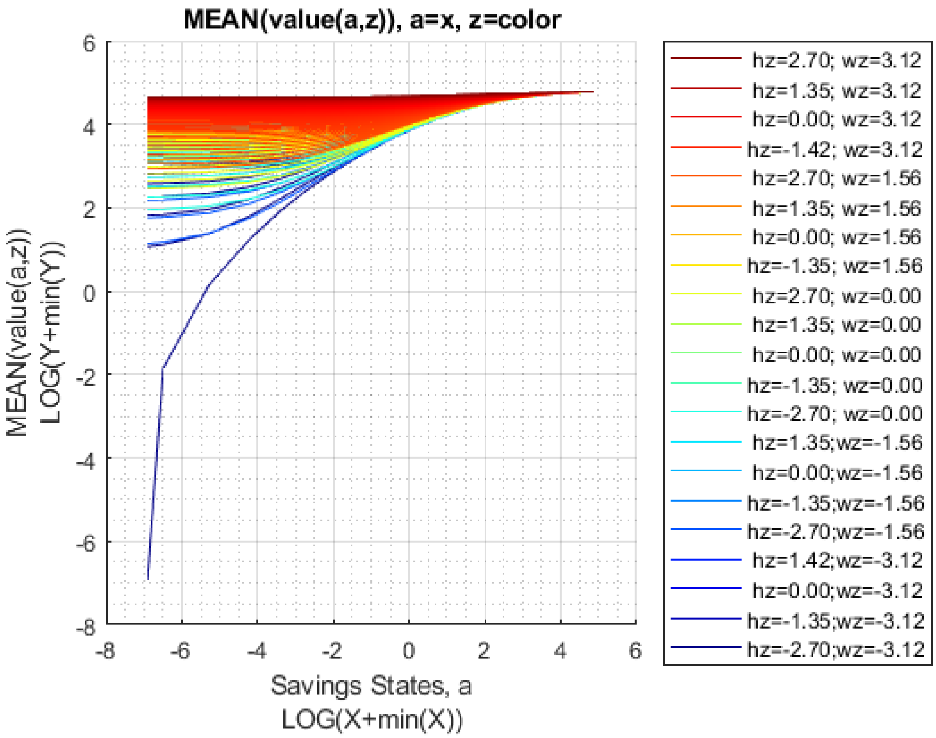
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm
```

```
xxx  MEAN(VAL(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0      -121.95       -119.04       -115.79       -112.34       -108.82       -1
```

```
xxx  MEAN(AP(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group      savings       mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5     mea
    -----    ----------     ----------     ----------     ----------     ----------     ----------     ---

      1              0              0              0              0              0              0
```

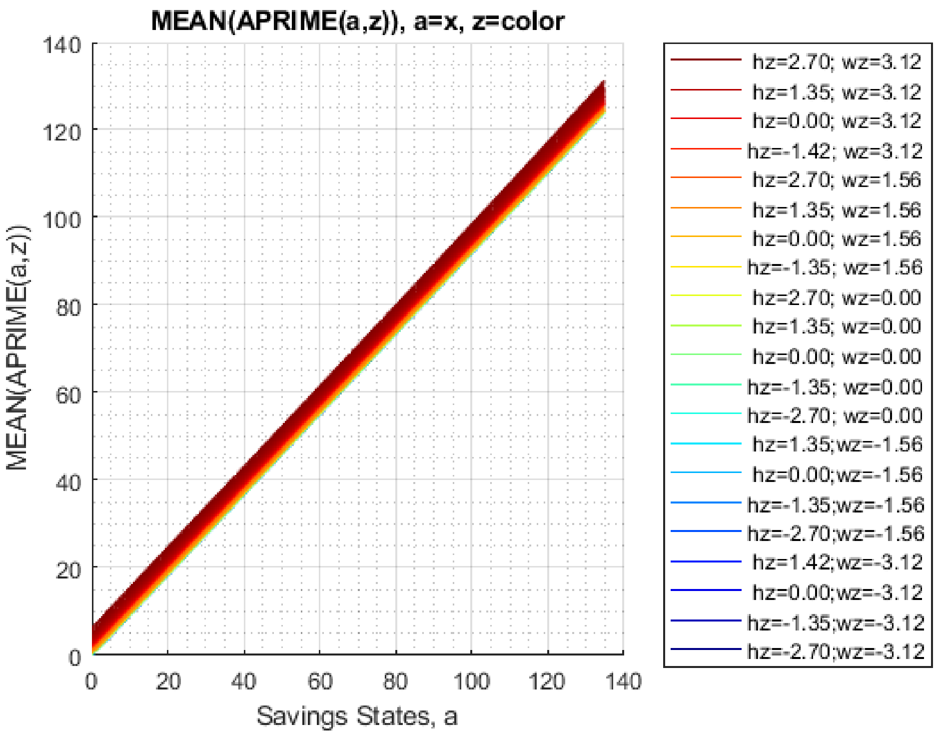
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

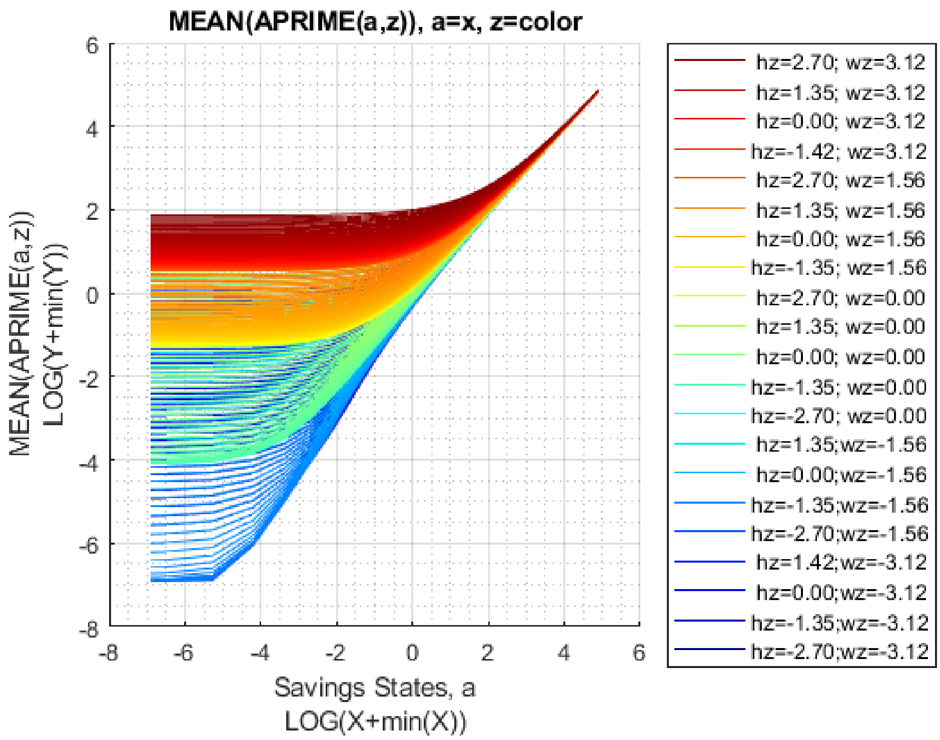




Graph Mean Savings Choices:

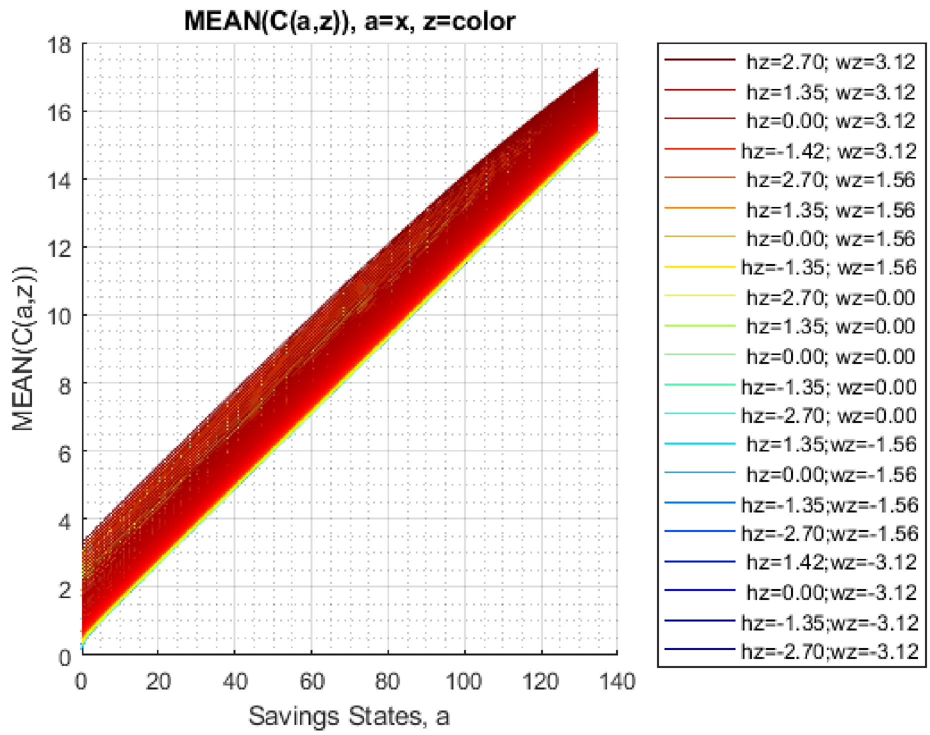
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(a,z))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

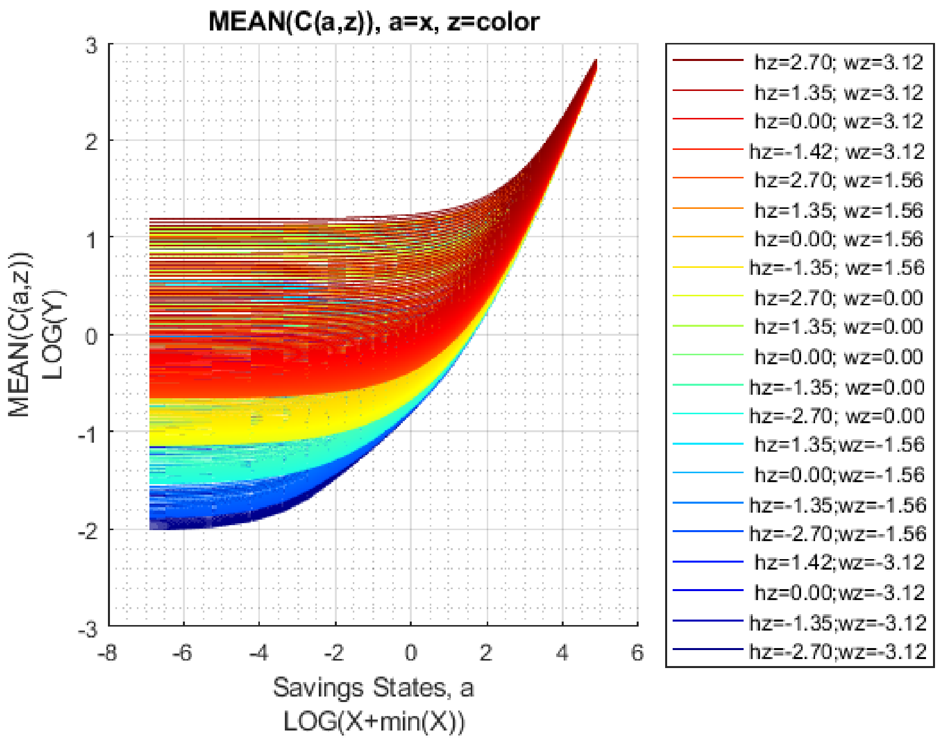




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





3.1.4 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(KM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per
```

```
xxx  MEAN(VAL(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        -38.153          -36.95       -35.802        -34.709        -33.718
      2        2       0        -45.736          -44.24       -42.778        -41.355        -40.041
```

```
      3       3       0        -49.467        -47.957        -46.467        -45.003        -43.644
      4       4       0        -53.322        -51.806        -50.292        -48.787        -47.384
      5       5       0        -56.129        -54.692        -53.245        -51.795        -50.441
      6       1       1        -26.282        -25.268        -24.319        -23.425        -22.612
      7       2       1        -30.883        -29.746         -28.66        -27.617         -26.65
      8       3       1        -33.096        -31.952        -30.853         -29.79        -28.803
      9       4       1        -35.694        -34.564        -33.469        -32.399        -31.401
     10       5       1        -37.748        -36.679        -35.641        -34.621        -33.672
```

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(KM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

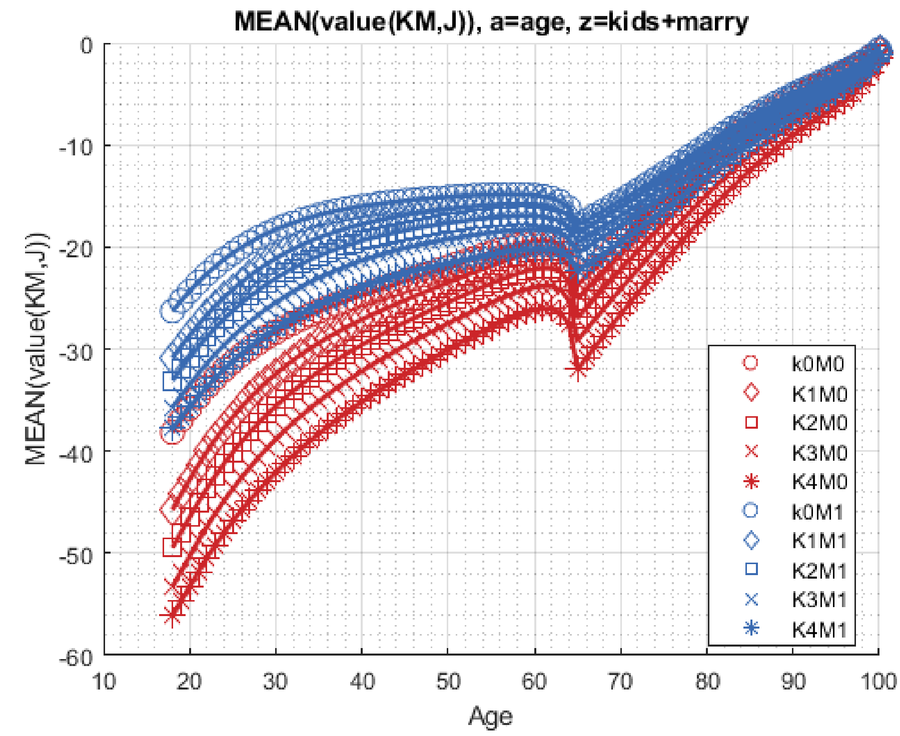
```
xxx  MEAN(AP(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1       1       0         34.497         34.459         34.419         34.455         34.493
      2       2       0         34.303         34.259         34.213         34.242         34.272
      3       3       0         34.149         34.105         34.058         34.086         34.115
      4       4       0         34.057         34.013         33.967         33.996         34.024
      5       5       0         33.974         33.933         33.889         33.919          33.95
      6       1       1         35.212          35.25         35.288         35.417         35.549
      7       2       1         34.954         34.979         35.003         35.113         35.226
      8       3       1         34.711         34.728         34.743         34.842         34.943
      9       4       1          34.51         34.519         34.527         34.617         34.708
     10       5       1         34.225         34.221         34.216          34.29         34.367
```

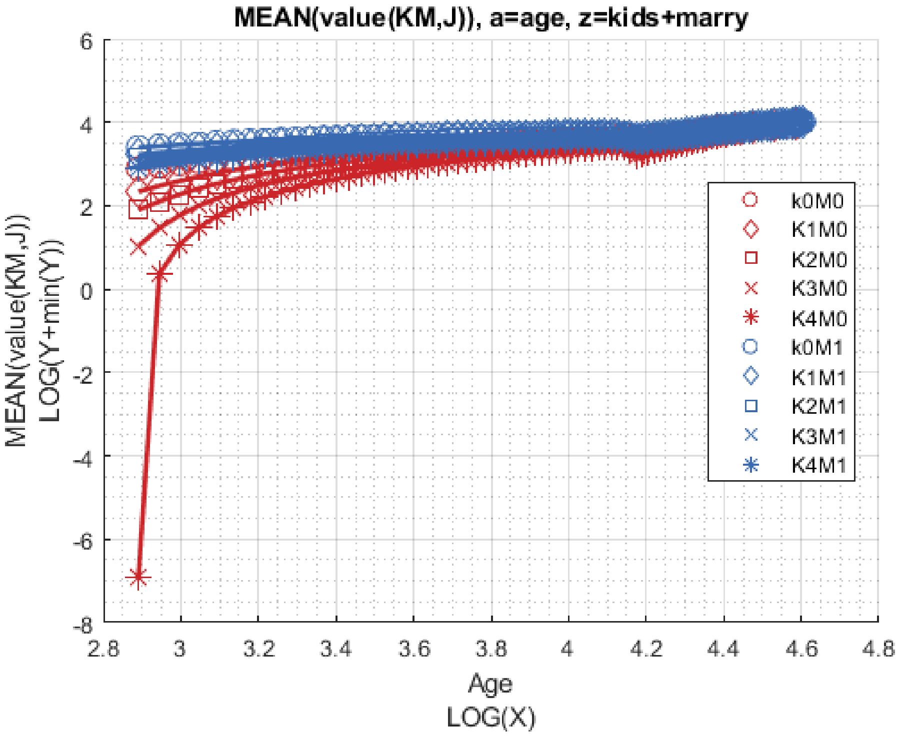
```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(KM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

```
xxx  MEAN(C(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1       1       0         2.0602         2.0988         2.1385         2.1805         2.2207
      2       2       0         2.2548         2.2985         2.3445         2.3938         2.4417
      3       3       0         2.4085         2.4527         2.4992         2.5498         2.5989
      4       4       0         2.5011         2.5444         2.5902         2.6404         2.6895
      5       5       0         2.5841          2.625         2.6684         2.7166         2.7635
      6       1       1         2.6152         2.6754         2.7367         2.8014         2.8634
      7       2       1          2.678         2.7363         2.7967         2.8619         2.9253
      8       3       1         2.7864         2.8428         2.9018          2.966         3.0285
      9       4       1         2.8495         2.9021         2.9575         3.0183         3.0775
     10       5       1         2.9142         2.9626         3.0136         3.0698          3.124
```

Graph Mean Values:

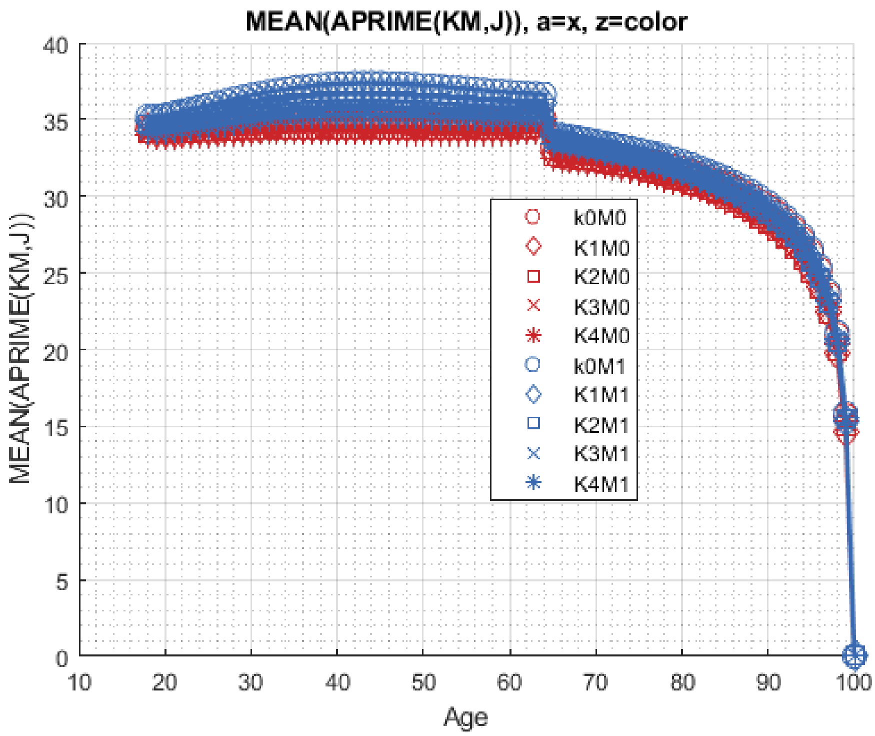
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(KM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

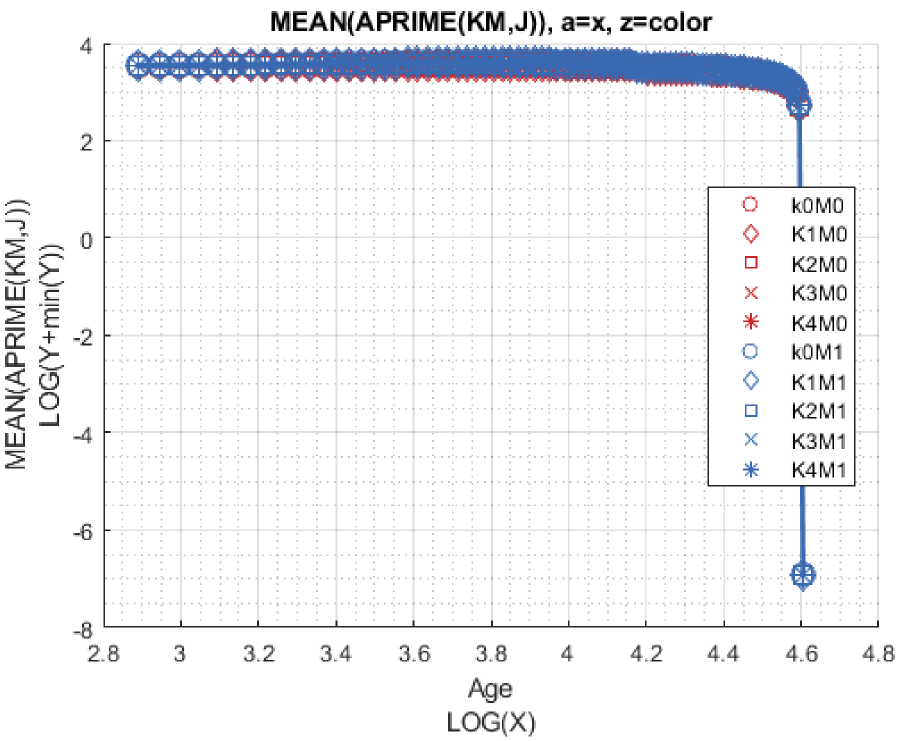




Graph Mean Savings Choices:

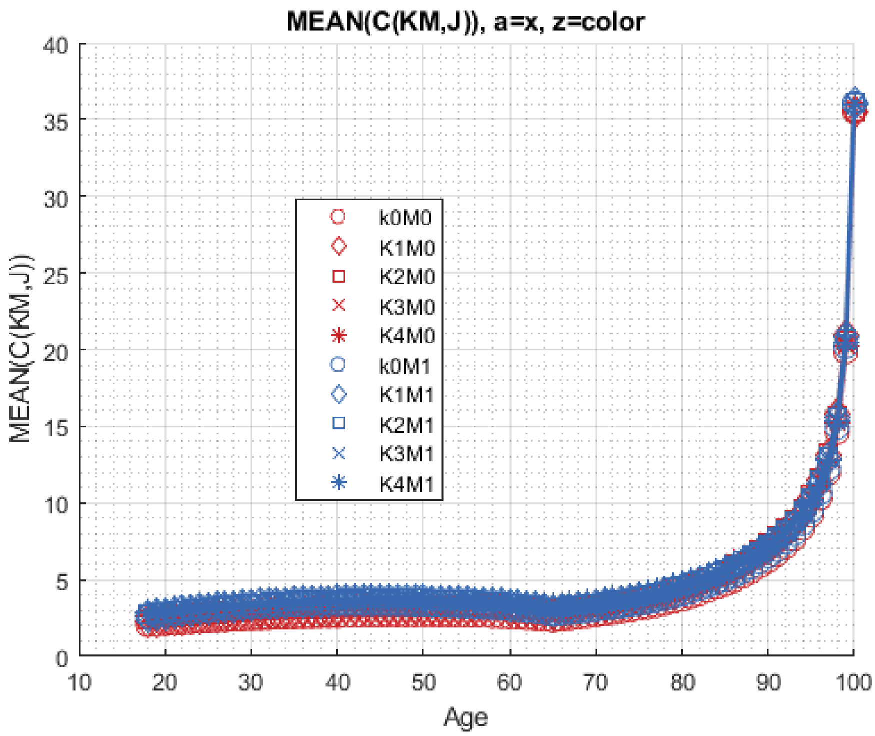
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(KM,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

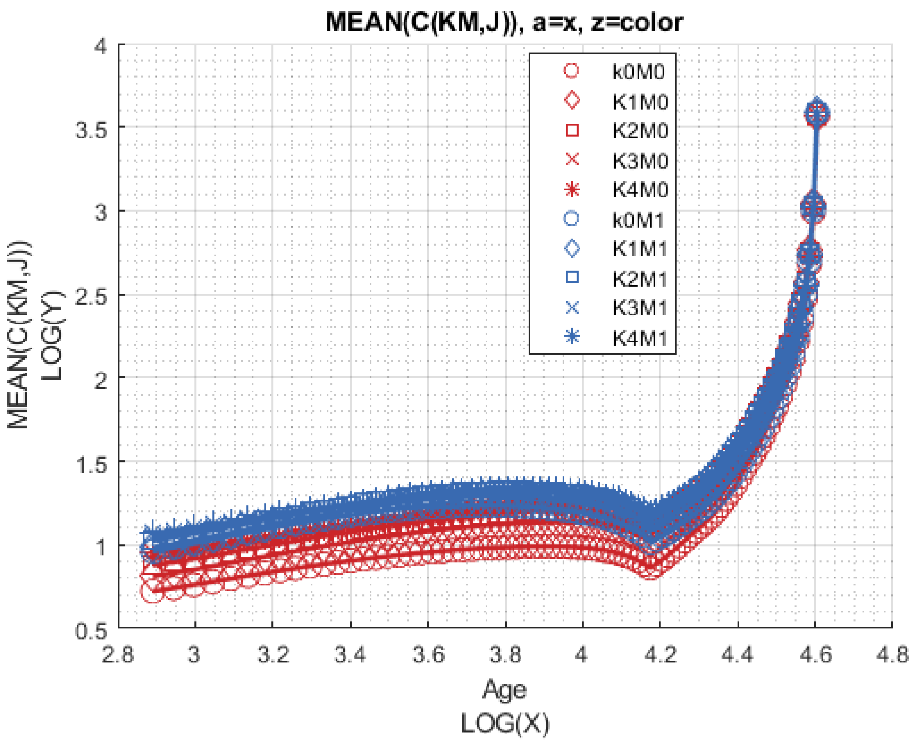




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





3.1.5 Analyze Education and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

```
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

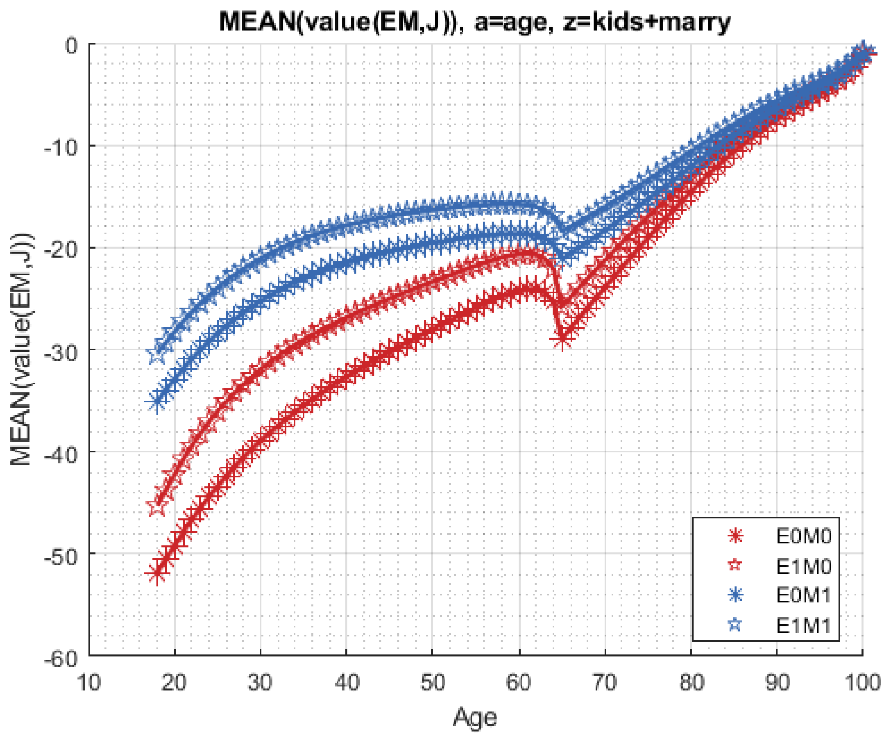
MEAN(VAL(EKM,J)), MEAN(AP(EKM,J)), MEAN(C(EKM,J))

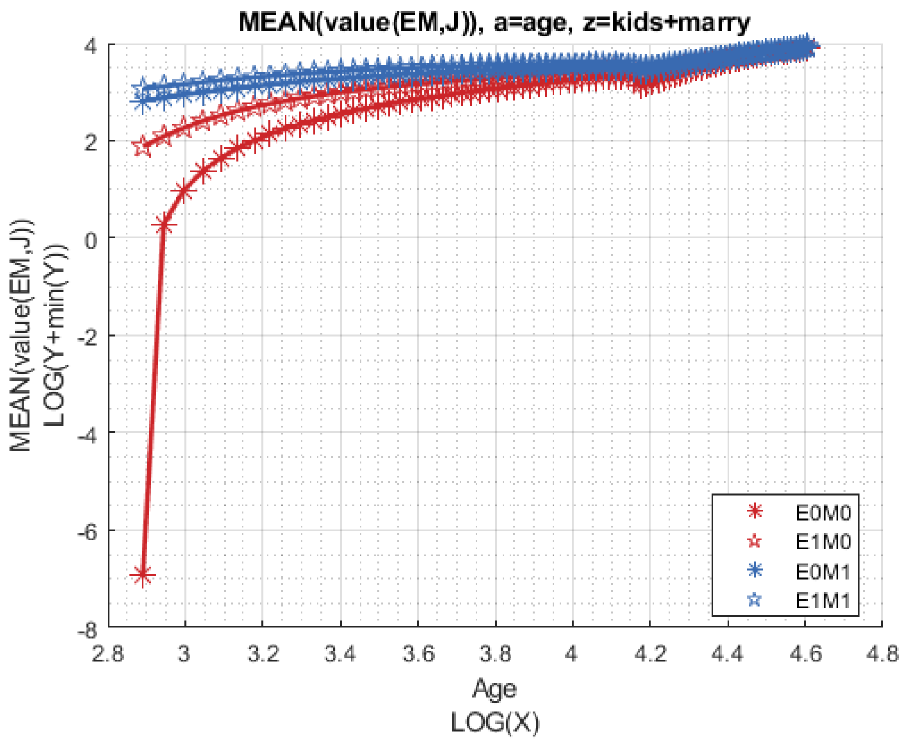
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(EKM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(VAL(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        -51.853        -50.513        -49.192        -47.891        -46.665
      2       1       0         -45.27        -43.745        -42.242        -40.768        -39.426
      3       0       1        -35.075        -33.966        -32.905        -31.884        -30.931
      4       1       1        -30.407        -29.317        -28.272        -27.257        -26.324

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(EKM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(AP(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        34.297         34.264          34.23          34.24         34.251
      2       1       0        34.094         34.043         33.989         34.039         34.091
      3       0       1        34.773         34.793         34.812         34.883         34.955
      4       1       1        34.672         34.686         34.698         34.828         34.962

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(EKM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(C(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        2.2603         2.2935         2.3281         2.3646         2.4003
      2       1       0        2.4632         2.5143         2.5682         2.6279         2.6854
      3       0       1         2.623         2.6679          2.714         2.7624         2.8096
      4       1       1        2.9143         2.9798         3.0485         3.1246         3.1978
```

Graph Mean Values:

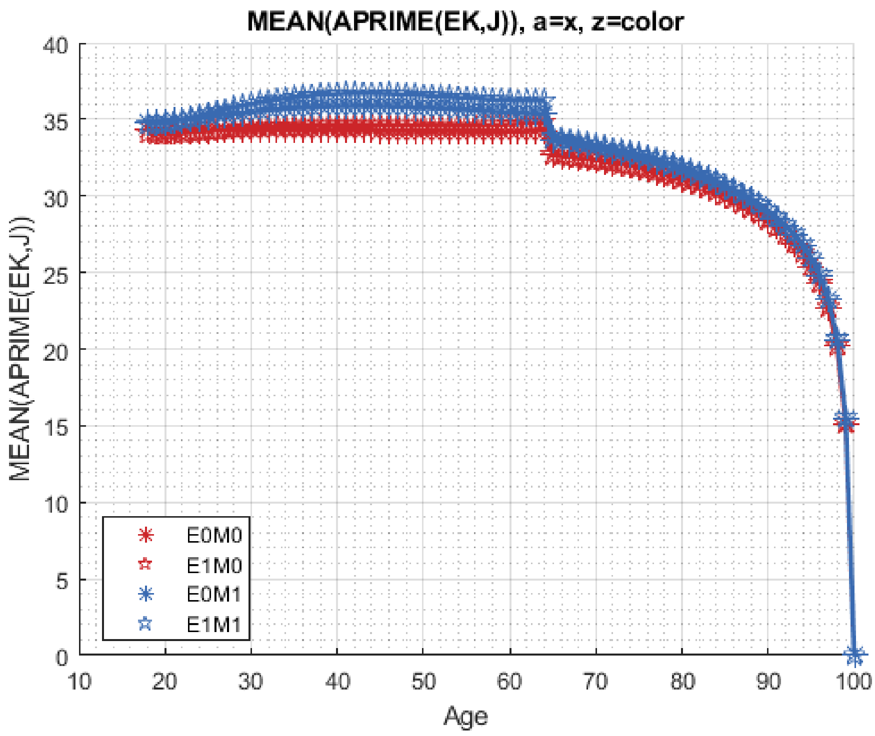
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(EM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

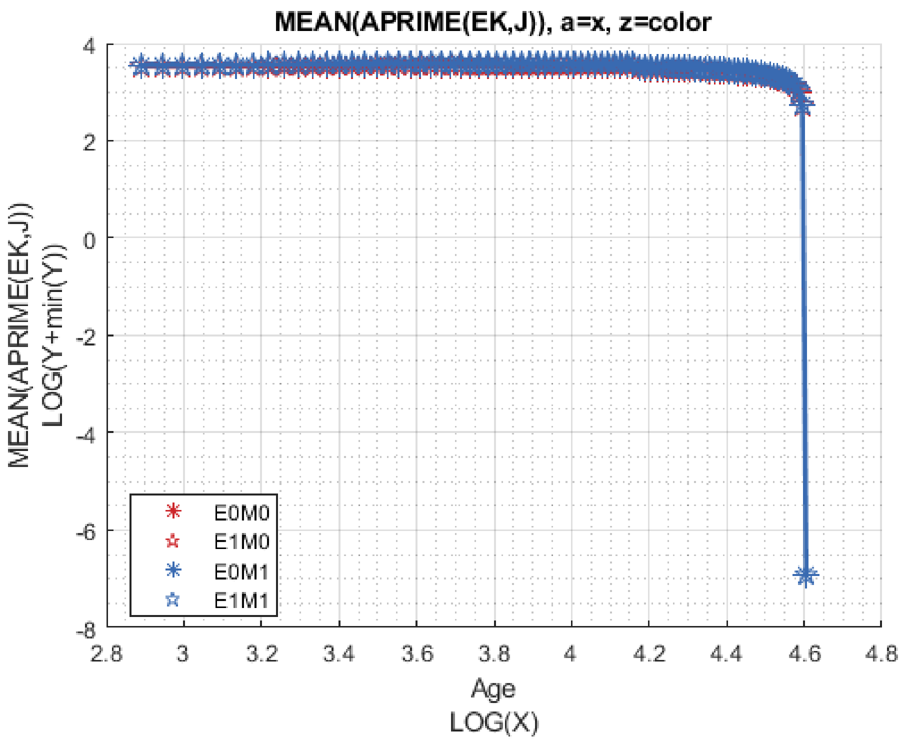




Graph Mean Savings Choices:

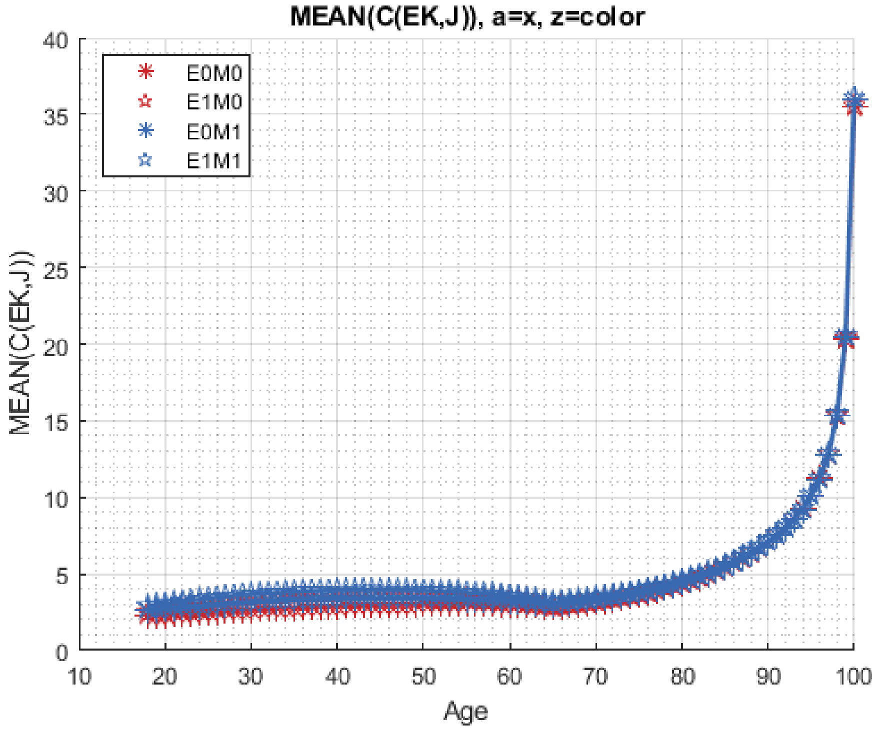
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(EK,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

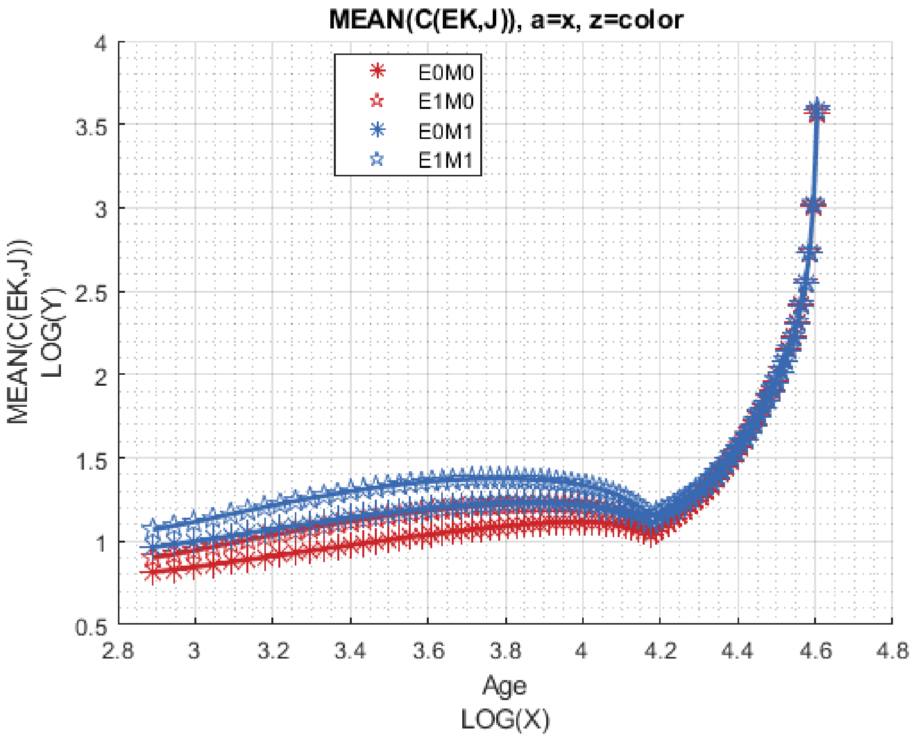




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(EK,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```


MEAN(C(EK,J)), a=x, z=color
E0M0
E1M0
E0M1
E1M1
MEAN(C(EK,J))
Age
0
5
10
15
20
25
30
35
40
10
20
30
40
50
60
70
80
90
100


MEAN(C(EK,J)), a=x, z=color
E0M0
E1M0
E0M1
E1M1
MEAN(C(EK,J))
LOG(Y)
Age
LOG(X)
0.5
1
1.5
2
2.5
3
3.5
4
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
4.8

Chapter 4

Alternative Value Function Solution Testing

4.1 Small Test Exact Solution Looped Minimizer

This is the example vignette for function: **snw_vfi_main** from the **PrjOptiSNW Package.** This function solves for policy function fully iteratively using matlab minimizer. Small Solution Analysis.This produces the same result as **snw_vfi_main_bisec_vec,** except slower. The purpose of this function is to confirm that the results from **snw_vfi_main_bisec_vec** is correct.

4.1.1 Test SNW_VFI_MAIN Defaults Small

Call the function with defaults parameters.

```
mp_param = snw_mp_param('default_small');
[V_VFI,ap_VFI,cons_VFI,mp_valpol_more] = snw_vfi_main(mp_param);
```

```
SNW_VFI_MAIN: Finished Age Group:18 of 18
SNW_VFI_MAIN: Finished Age Group:17 of 18
SNW_VFI_MAIN: Finished Age Group:16 of 18
SNW_VFI_MAIN: Finished Age Group:15 of 18
SNW_VFI_MAIN: Finished Age Group:14 of 18
SNW_VFI_MAIN: Finished Age Group:13 of 18
SNW_VFI_MAIN: Finished Age Group:12 of 18
SNW_VFI_MAIN: Finished Age Group:11 of 18
SNW_VFI_MAIN: Finished Age Group:10 of 18
SNW_VFI_MAIN: Finished Age Group:9 of 18
SNW_VFI_MAIN: Finished Age Group:8 of 18
SNW_VFI_MAIN: Finished Age Group:7 of 18
SNW_VFI_MAIN: Finished Age Group:6 of 18
SNW_VFI_MAIN: Finished Age Group:5 of 18
SNW_VFI_MAIN: Finished Age Group:4 of 18
SNW_VFI_MAIN: Finished Age Group:3 of 18
SNW_VFI_MAIN: Finished Age Group:2 of 18
SNW_VFI_MAIN: Finished Age Group:1 of 18
Elapsed time is 515.239525 seconds.
Completed SNW_VFI_MAIN;SNW_MP_PARAM=default_small;SNW_MP_CONTROL=default_base
```

4.1.2 Small Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = [19, 22:5:97, 100];
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

4.1.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm
```

```
xxx  MEAN(VAL(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5
    -----    ---------    ----------    ----------    ----------    ----------    ----------

      1              0       -17.721       -9.4697       -4.6571       -1.7924      -0.23581
      2      0.0097656       -17.284       -9.3219       -4.5706       -1.7215      -0.16909
      3       0.078125       -15.196       -8.4993       -4.1004       -1.3514       0.17326
      4        0.26367       -11.907       -7.0394       -3.3075      -0.78441       0.67194
      5          0.625       -8.4194       -5.2786       -2.3615      -0.19487        1.1461
      6         1.2207        -5.393       -3.5129       -1.3918       0.35026        1.5392
      7         2.1094       -3.0352       -1.9483      -0.50577       0.84352        1.8533
      8         3.3496       -1.2918      -0.66899       0.26902        1.2874        2.1081
      9              5     -0.030416       0.32906       0.92609        1.6707        2.3215
     10         7.1191       0.87699        1.0879        1.4656        1.9934        2.5052
     11         9.7656        1.5329        1.6594        1.8992         2.267        2.6661
     12         12.998        2.0119        2.0896        2.2435        2.4983        2.8056
     13         16.875         2.366        2.4149        2.5152        2.6918        2.9253
     14         21.455        2.6312        2.6629        2.7294        2.8524        3.0281
     15         26.797        2.8329        2.8539        2.8987        2.9852        3.1169
     16         32.959        2.9883        3.0025        3.0334        3.0948        3.1935
     17             40        3.1097        3.1195        3.1411        3.1853        3.2595
     18         47.979        3.2056        3.2125        3.2279        3.2601        3.3161
```

19	56.953	3.2822	3.2872	3.2984	3.322	3.3647
20	66.982	3.3441	3.3478	3.356	3.3736	3.4063
21	78.125	3.3947	3.3974	3.4035	3.4168	3.442
22	90.439	3.4363	3.4383	3.4429	3.4531	3.4727
23	103.98	3.4708	3.4724	3.4759	3.4837	3.4991
24	118.82	3.4997	3.5009	3.5036	3.5097	3.5218
25	135	3.5241	3.525	3.5271	3.5319	3.5415

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(A,Z))", ap_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per
```

xxx MEAN(AP(A,Z)) xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5
1	0	3.2168e-05	0.0034996	0.049878	0.24382	0.89299
2	0.0097656	0.00055444	0.0053208	0.053281	0.24787	0.89865
3	0.078125	0.021863	0.029684	0.083029	0.28062	0.93971
4	0.26367	0.13322	0.14773	0.20012	0.3888	1.0591
5	0.625	0.39134	0.41043	0.45332	0.64573	1.3087
6	1.2207	0.84131	0.86393	0.91226	1.0928	1.745
7	2.1094	1.5303	1.5542	1.6156	1.7559	2.3963
8	3.3496	2.4876	2.5118	2.573	2.6876	3.3398
9	5	3.7642	3.7887	3.8498	3.9922	4.592
10	7.1191	5.4275	5.4525	5.5145	5.6929	6.1933
11	9.7656	7.4794	7.5043	7.5679	7.7532	8.1877
12	12.998	9.9124	9.9329	9.9956	10.186	10.627
13	16.875	12.928	12.95	13.005	13.196	13.715
14	21.455	16.529	16.548	16.604	16.783	17.374
15	26.797	20.601	20.618	20.668	20.837	21.462
16	32.959	25.307	25.325	25.37	25.525	26.151
17	40	30.667	30.69	30.742	30.886	31.487
18	47.979	36.761	36.782	36.841	36.999	37.562
19	56.953	43.773	43.795	43.847	44.012	44.56
20	66.982	51.605	51.628	51.688	51.85	52.403
21	78.125	59.955	59.978	60.038	60.211	60.768
22	90.439	69.267	69.29	69.352	69.528	70.097
23	103.98	79.753	79.774	79.834	80.008	80.586
24	118.82	91.116	91.14	91.201	91.367	91.942
25	135	103.47	103.49	103.55	103.72	104.29

```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(A,Z))", cons_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per
```

xxx MEAN(C(A,Z)) xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5
1	0	0.30273	0.43104	0.68779	1.2165	2.3367
2	0.0097656	0.31374	0.44069	0.69581	1.2239	2.3424
3	0.078125	0.37308	0.49662	0.74605	1.2709	2.3811
4	0.26367	0.48039	0.59638	0.846	1.3793	2.478
5	0.625	0.64735	0.75736	1.0152	1.5439	2.6496
6	1.2207	0.89648	1.0013	1.2519	1.7913	2.9071
7	2.1094	1.2479	1.3498	1.5854	2.1634	3.2903
8	3.3496	1.7393	1.8394	2.0734	2.6754	3.7896
9	5	2.3872	2.4859	2.7182	3.2909	4.4564
10	7.1191	3.1917	3.289	3.5191	4.0542	5.3181

```
     11          9.7656        4.2188        4.3155         4.543          5.07        6.3986
     12          12.998        5.5439        5.6447        5.8722        6.3933        7.7142
     13          16.875        7.0334         7.133        7.3676        7.8866        9.1285
     14          21.455         8.754        8.8551        9.0887        9.6188        10.789
     15          26.797        10.886        10.989        11.228        11.768        12.903
     16          32.959        13.336        13.438        13.682        14.235        15.368
     17              40        16.151        16.249        16.485        17.049        18.207
     18          47.979        19.321         19.42        19.649          20.2        21.394
     19          56.953        22.728        22.827        23.062        23.605        24.816
     20          66.982        26.539        26.636        26.864         27.41        28.615
     21          78.125        31.124        31.221         31.45        31.985        33.186
     22          90.439        36.108        36.205        36.431        36.963        38.152
     23          103.98        41.345        41.444        41.673        42.206        43.386
     24          118.82        47.202        47.298        47.525        48.066        49.248
     25             135        53.632        53.731        53.962        54.496        55.685
```

4.2 Small Test Grid Search Solution

This is the example vignette for function: **snw_vfi_main_grid_search** from the **PrjOptiSNW Package.** This function solves for policy function using grid search. Small Solution Analysis. Small Solution Analysis, husband 5 shocks, wife 1 shocks.

4.2.1 Test SNW_VFI_MAIN_GRID_SEARCH Defaults Small

Call the function with defaults parameters.

```
mp_param = snw_mp_param('default_small');
[V_VFI,ap_VFI,cons_VFI,mp_valpol_more] = snw_vfi_main_grid_search(mp_param);
```

```
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:18 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:17 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:16 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:15 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:14 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:13 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:12 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:11 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:10 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:9 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:8 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:7 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:6 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:5 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:4 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:3 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:2 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:1 of 18
Elapsed time is 6.771761 seconds.
Completed SNW_VFI_MAIN_GRID_SEARCH;SNW_MP_PARAM=default_small;SNW_MP_CONTROL=default_base
```

4.2.2 Small Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = [19, 22:5:97, 100];
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
```

```
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'Hshock', eta_H_grid});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

4.2.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm
```

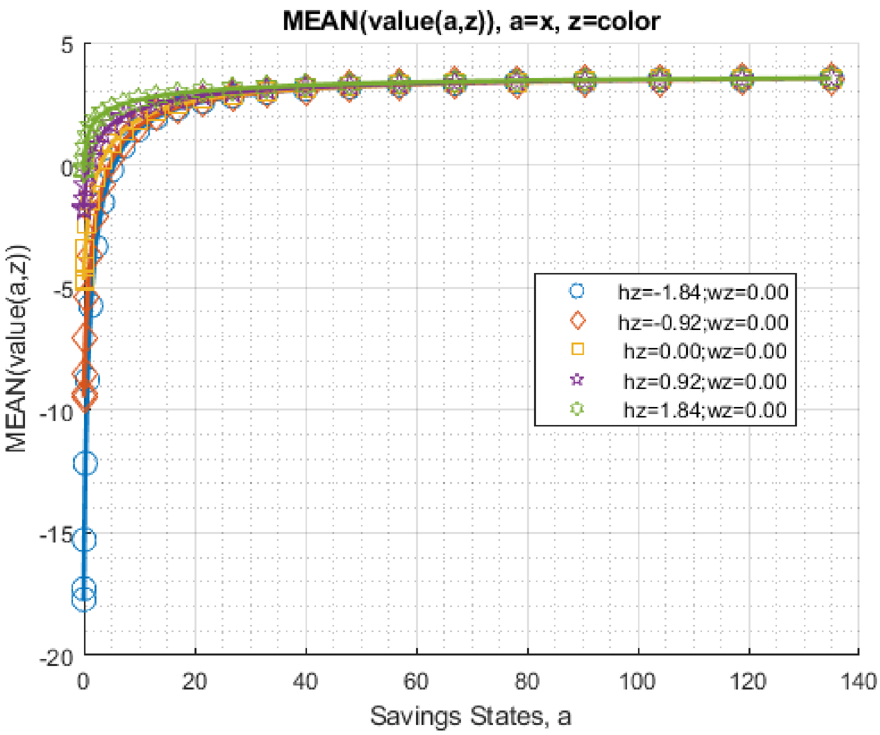
```
xxx  MEAN(VAL(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings     mean_Hshock__1_8395    mean_Hshock__0_91976    mean_Hshock_0    mean_Hshoc
    -----    ---------    -------------------    --------------------    -------------    ----------

      1             0           -17.723                 -9.4806             -4.7079             -1.8
      2     0.0097656           -17.287                 -9.3335             -4.6223             -1.7
      3      0.078125           -15.289                 -8.5102             -4.1607             -1.3
      4       0.26367           -12.169                 -7.0906             -3.3823             -0.82
      5         0.625           -8.7667                 -5.3975             -2.4071             -0.24
      6        1.2207           -5.7445                 -3.6744             -1.4216              0.29
      7        2.1094           -3.3257                  -2.112            -0.53821              0.80
      8        3.3496           -1.5195                -0.81731             0.22515               1.2
      9             5          -0.20516                 0.20391             0.87016               1.6
     10        7.1191           0.74607                 0.98431              1.4069               1.9
     11        9.7656            1.4347                  1.5779              1.8454               2.2
     12        12.998            1.9367                  2.0246              2.1961               2.4
     13        16.875            2.3099                   2.364               2.476               2.6
     14        21.455            2.5887                  2.6239              2.6967               2.8
     15        26.797            2.7998                  2.8231              2.8727               2.9
     16        32.959             2.963                  2.9785              3.0123               3.0
     17            40            3.0902                  3.1009              3.1242               3.1
     18        47.979            3.1902                  3.1978              3.2144               3.2
     19        56.953              3.27                  3.2754              3.2875               3.3
     20        66.982            3.3345                  3.3384              3.3471               3.3
     21        78.125            3.3871                  3.3899              3.3964               3.4
     22        90.439            3.4301                  3.4323              3.4372               3.4
```

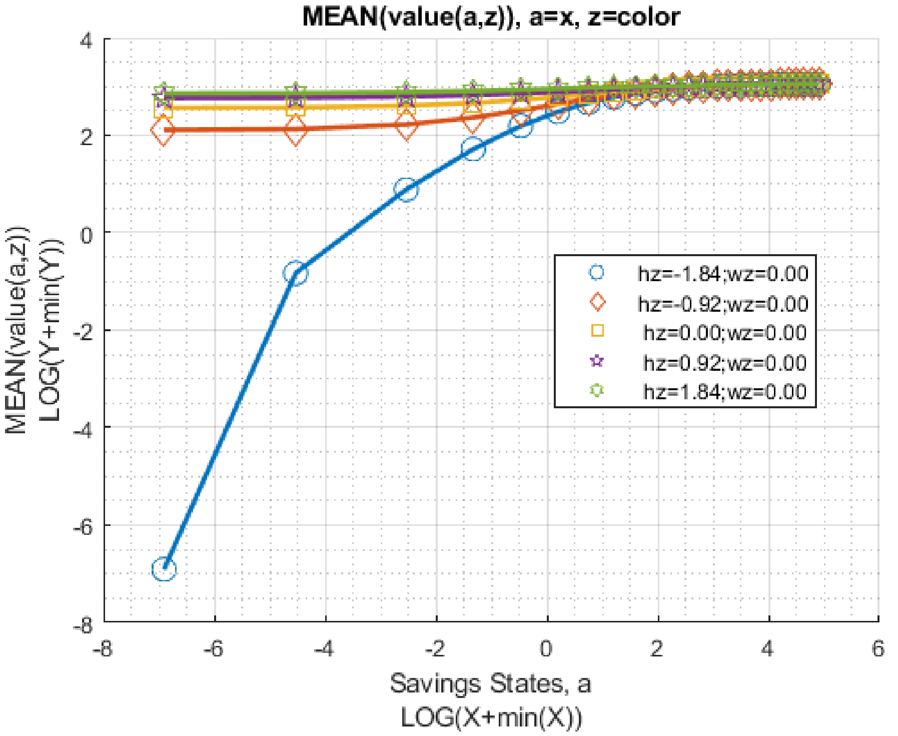
```
     23        103.98          3.4658                 3.4674               3.4712             3.4
     24        118.82          3.4957                 3.4969               3.4998             3.5
     25           135          3.5208                 3.5218                3.524             3.5

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(A,Z))", ap_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(AP(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings     mean_Hshock__1_8395    mean_Hshock__0_91976    mean_Hshock_0    mean_Hshoc
    -----    ---------    -------------------    --------------------    -------------    ----------

      1             0                  1               1.1435              1.5972           2.59
      2     0.0097656             1.0463                1.213              1.6574           2.61
      3      0.078125             1.8009               2.0093              2.1991            2.8
      4       0.26367             2.9491               3.0648              3.2454           3.62
      5         0.625             4.0602               4.1806              4.2546           4.54
      6        1.2207             5.1481               5.2454              5.2731           5.40
      7        2.1094             6.1389                6.213                6.25           6.25
      8        3.3496             7.0556               7.1019              7.1713            7.1
      9             5             7.9537               7.9815              8.0556           8.07
     10        7.1191             8.8611               8.8889              8.9398           9.00
     11        9.7656             9.7824               9.7963              9.8519           9.92
     12        12.998             10.606                10.63              10.648           10.7
     13        16.875             11.481               11.491              11.537           11.5
     14        21.455             12.407               12.407              12.431           12.4
     15        26.797             13.287               13.301              13.306           13.3
     16        32.959              14.13                14.13              14.167           14.1
     17            40             14.981               14.981              14.991           15.0
     18        47.979              15.88                15.88              15.884           15.9
     19        56.953              16.75               16.773              16.782           16.7
     20        66.982             17.681               17.685              17.699           17.7
     21        78.125             18.495                 18.5              18.509           18.5
     22        90.439             19.338               19.338              19.352            19.
     23        103.98              20.25               20.264              20.269           20.2
     24        118.82             21.097               21.097               21.13           21.1
     25           135             21.963               21.968              21.977           21.9

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(A,Z))", cons_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(C(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings     mean_Hshock__1_8395    mean_Hshock__0_91976    mean_Hshock_0    mean_Hshoc
    -----    ---------    -------------------    --------------------    -------------    ----------

      1             0            0.30277                0.43205             0.70498           1.24
      2     0.0097656            0.31384                0.44176             0.71311           1.25
      3      0.078125            0.38061                0.49936               0.768           1.31
      4       0.26367            0.50208                 0.6111             0.86902            1.4
      5         0.625            0.67677                0.76238              1.0363           1.58
      6        1.2207            0.89732                0.96685              1.2492           1.84
      7        2.1094             1.2189                 1.2789               1.543           2.24
      8        3.3496             1.6892                 1.7561              1.9651           2.69
      9             5             2.3251                 2.4024              2.5736           3.25
     10        7.1191             3.1269                 3.1903              3.3745           3.94
     11        9.7656             4.0839                 4.1689              4.3128            4.8
     12        12.998             5.4106                  5.457              5.6873           6.12
     13        16.875             6.9612                 7.0462              7.1563           7.63
     14        21.455             8.5924                 8.7131              8.8962           9.33
```

15	26.797	10.6	10.647	10.911	11.3
16	32.959	13.149	13.269	13.33	13.8
17	40	16.034	16.154	16.378	16.7
18	47.979	18.971	19.092	19.343	19.7
19	56.953	22.573	22.485	22.69	23.2
20	66.982	26.089	26.168	26.322	26.7
21	78.125	30.843	30.917	31.101	31.
22	90.439	35.929	36.049	36.177	36.6
23	103.98	40.986	40.918	41.144	41.7
24	118.82	47.072	47.192	47.018	47.5
25	135	53.493	53.538	53.682	54.0

Graph Mean Values:

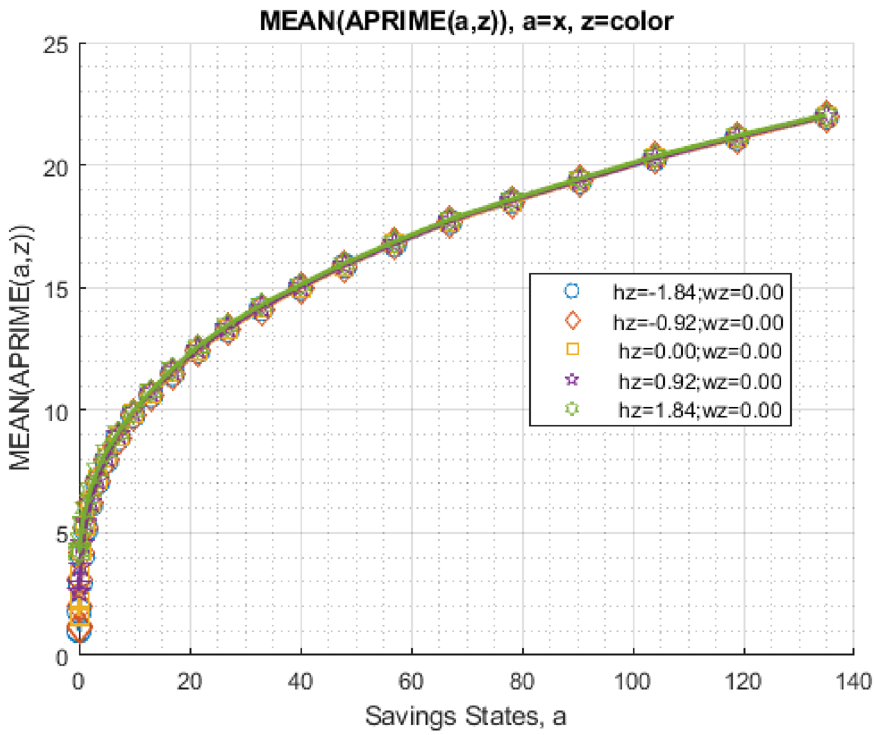
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

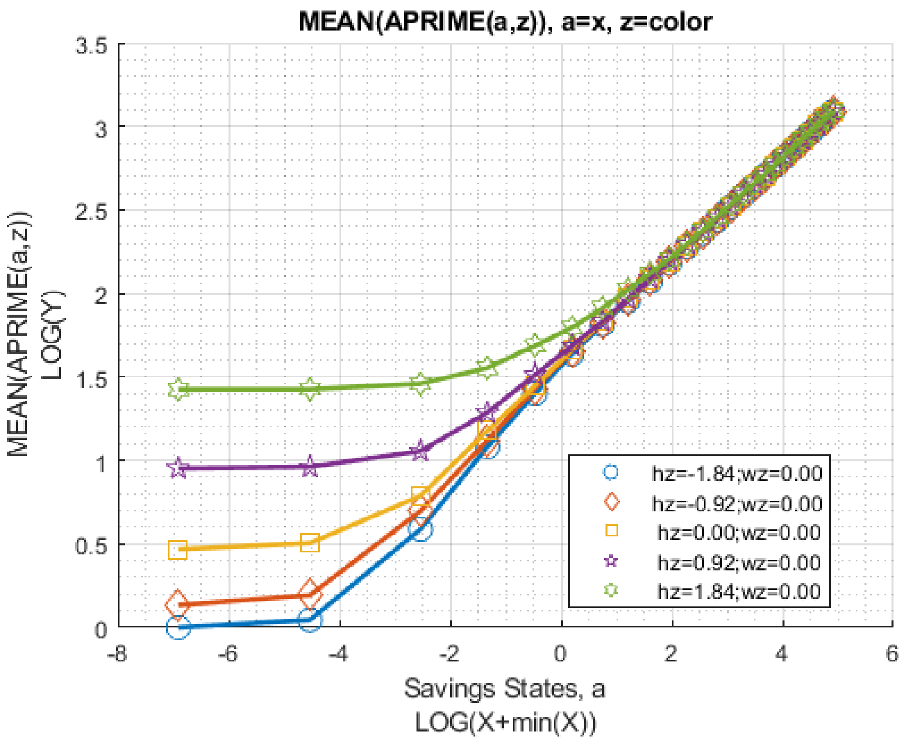




Graph Mean Savings Choices:

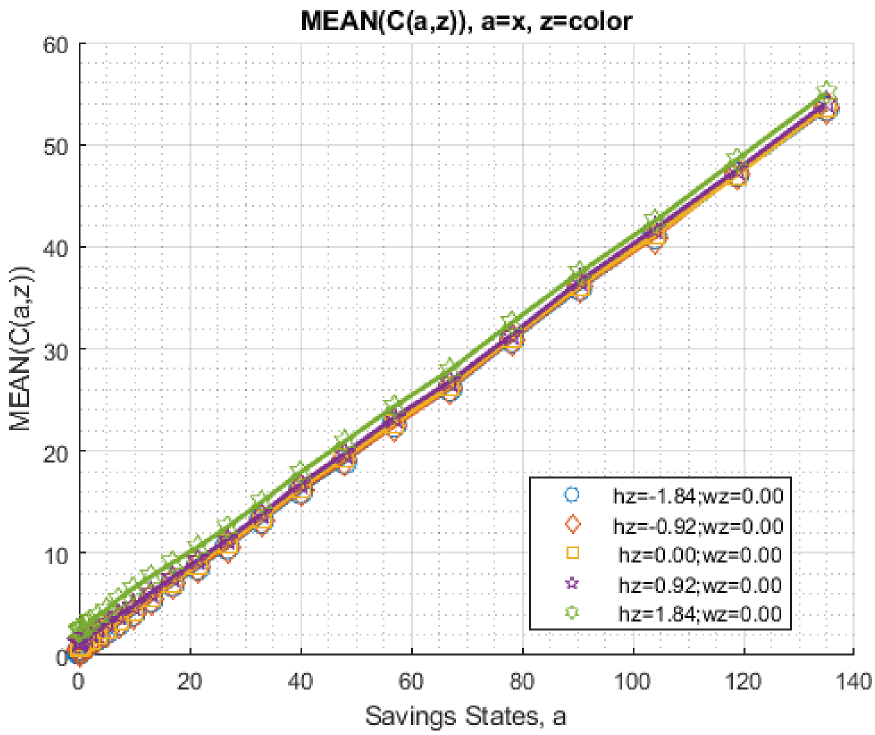
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(a,z))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

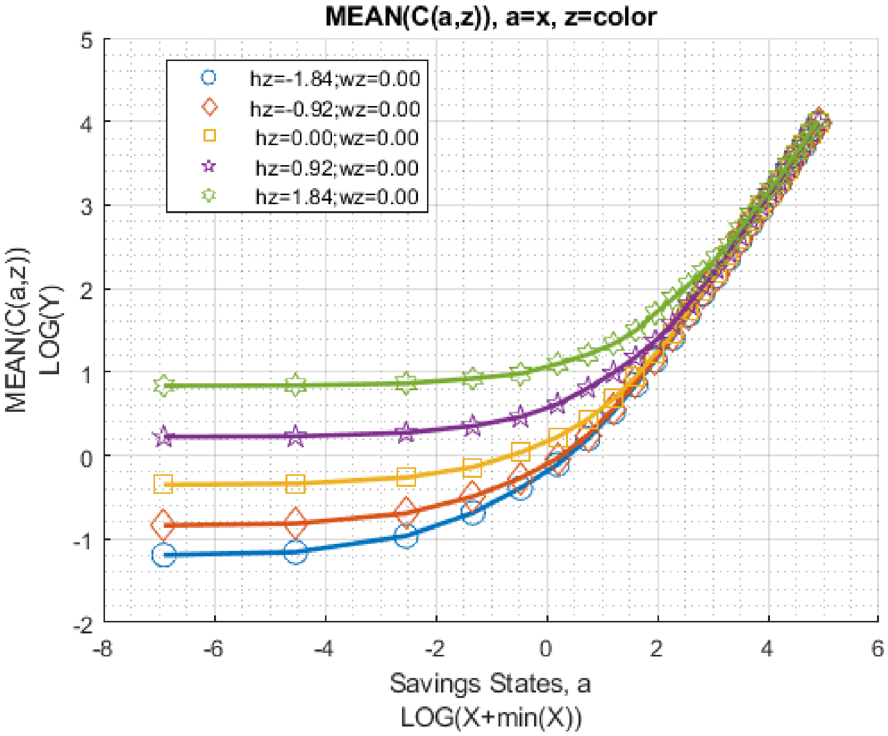




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





4.2.4 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["k0M0", "K1M0", "K2M0", "k0M1", "K1M1", "K2M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = { 'o', 'd' ,'s', 'o', 'd', 's'};
mp_support_graph('cl_colors') = {'red', 'red', 'red', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(KM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per
```

```
xxx  MEAN(VAL(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0          1.402         1.6857         1.8728         1.9257         1.894
      2        2       0       -0.12483        0.36646         0.7436         0.9457        1.0402
      3        3       0       -0.89708       -0.41863      -0.032067        0.18508       0.29597
      4        1       1          1.967         2.1822         2.3218         2.3638        2.3393
      5        2       1        0.96762         1.2863         1.5349         1.6741         1.739
      6        3       1        0.51874         0.8123         1.0493         1.1855        1.2514
```

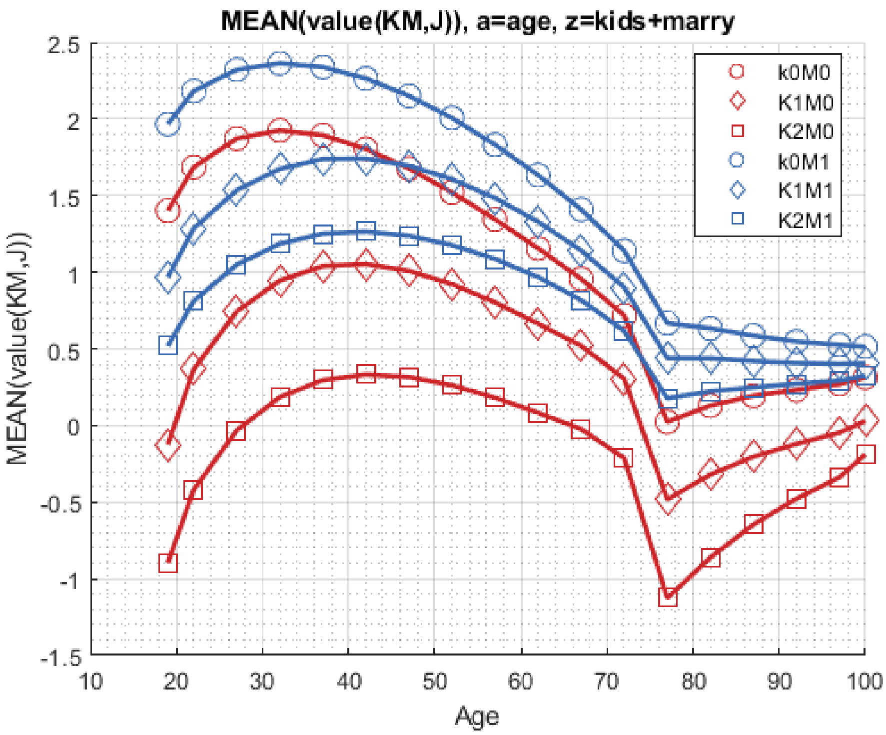
```
% Aprime Choice
```

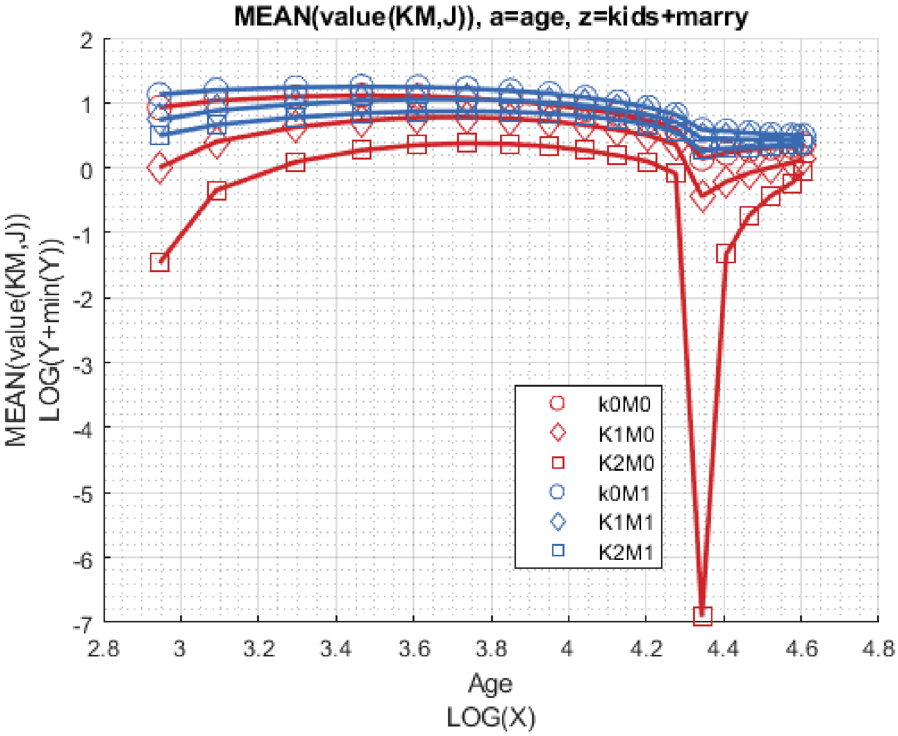
```
tb_az_ap = ff_summ_nd_array("MEAN(AP(KM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(AP(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0        12.948         12.924         13.052         13.152         13.224
      2        2       0        12.924          12.88         13.008         13.096          13.16
      3        3       0         12.86         12.848         12.972         13.084         13.108
      4        1       1         12.86         12.856         12.972         13.076          13.14
      5        2       1        12.876          12.82         12.956         13.028           13.1
      6        3       1        12.804         12.784         12.912         12.984          13.06
```

```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(KM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(C(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0        6.6347         6.7441         6.9773         7.1425         7.2307
      2        2       0        6.6476         6.7581         6.9904         7.1656         7.2723
      3        3       0        6.6679         6.7696         7.0001         7.1694         7.8468
      4        1       1         6.885         7.0096         7.2673         7.4584         7.5792
      5        2       1         6.856          6.987         7.2319         7.4245         7.5481
      6        3       1        6.8672         6.9855         7.2175         7.4148         7.5346
```

Graph Mean Values:

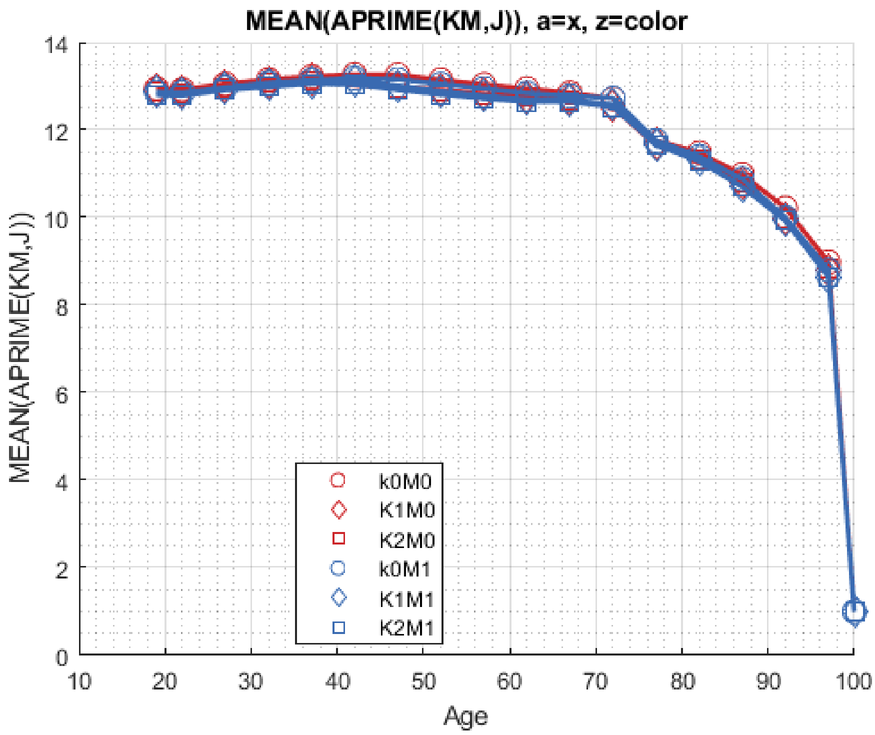
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(KM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

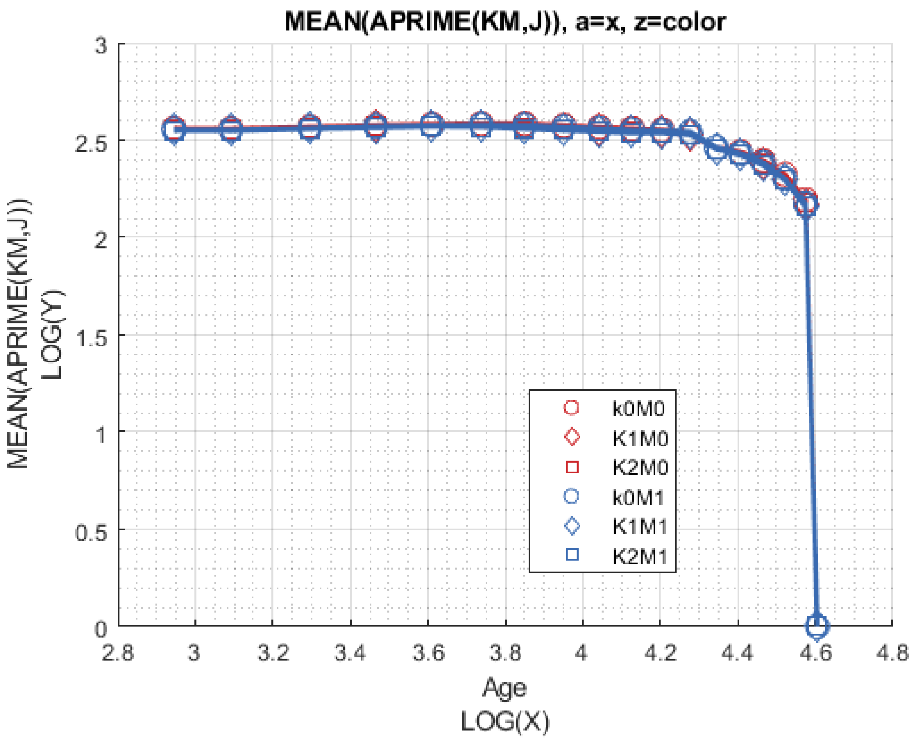




Graph Mean Savings Choices:

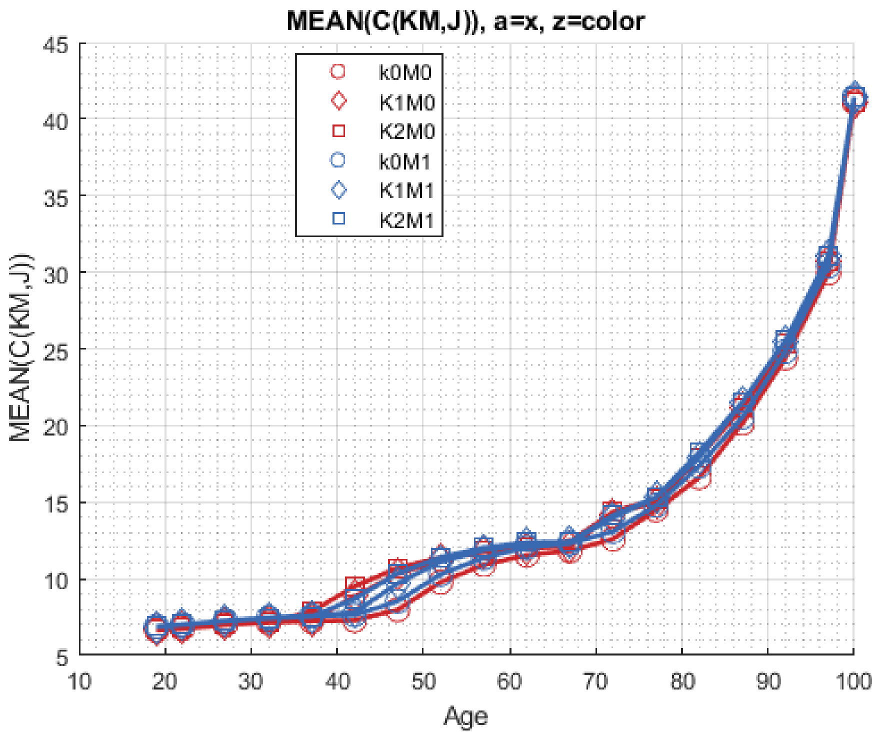
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(KM,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

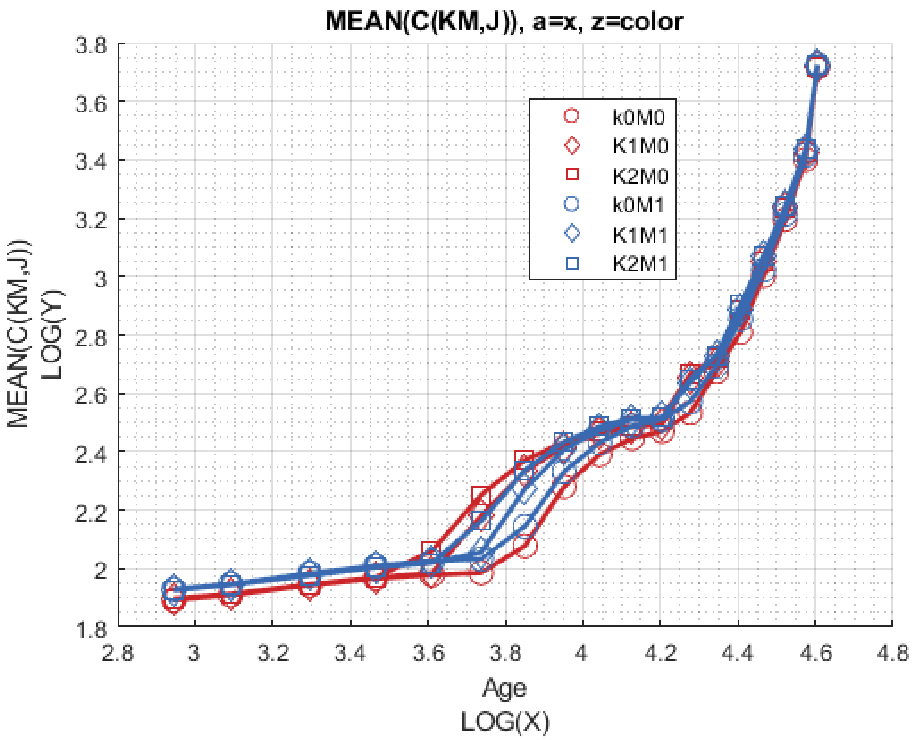




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





4.2.5 Analyze Education and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

MEAN(VAL(EKM,J)), MEAN(AP(EKM,J)), MEAN(C(EKM,J))

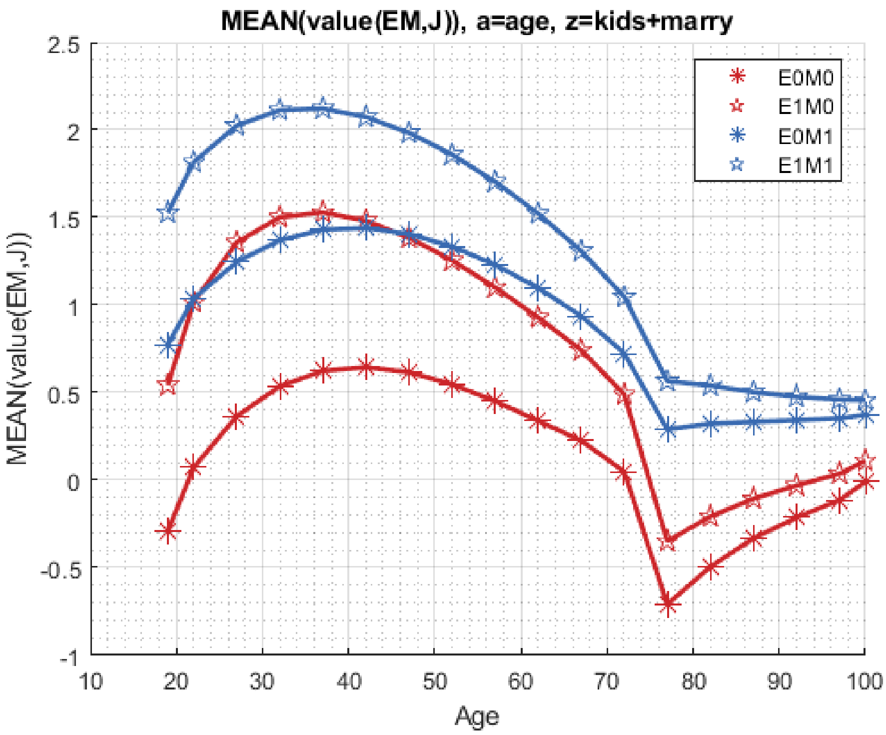
Tabulate value and policies:

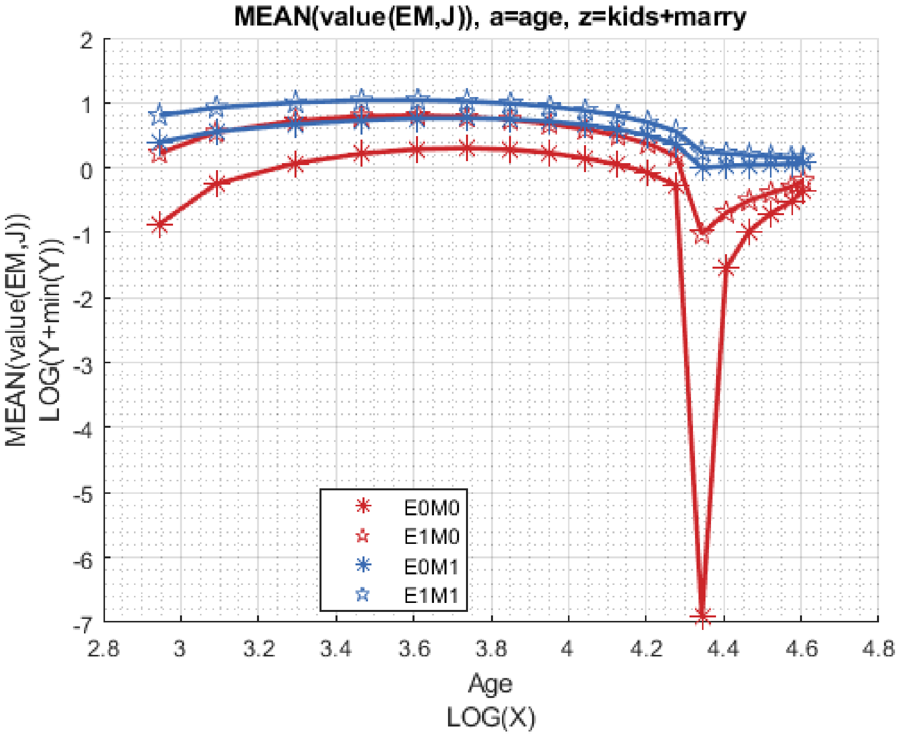
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(EKM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(VAL(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0       -0.28978       0.072789        0.36537        0.53629         0.6243
      2       1       0        0.54315         1.0162         1.3575         1.5014         1.5291
      3       0       1        0.77529          1.038         1.2458         1.3693         1.4312
      4       1       1         1.5269         1.8159         2.0249         2.1129         2.1219

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(EKM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p
```

```
xxx  MEAN(AP(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        12.989         12.979         13.035         13.093         13.131
      2       1       0        12.832         12.789         12.987         13.128         13.197
      3       0       1        12.933         12.923         12.976         13.021         13.072
      4       1       1         12.76         12.717         12.917         13.037         13.128

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(EKM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(C(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        6.6262           6.69         6.8285         6.9343         7.2515
      2       1       0        6.6738         6.8246         7.1501         7.3841         7.6483
      3       0       1        6.8114         6.8929         7.0479         7.1732           7.26
      4       1       1        6.9273         7.0952         7.4299          7.692          7.848
```

Graph Mean Values:

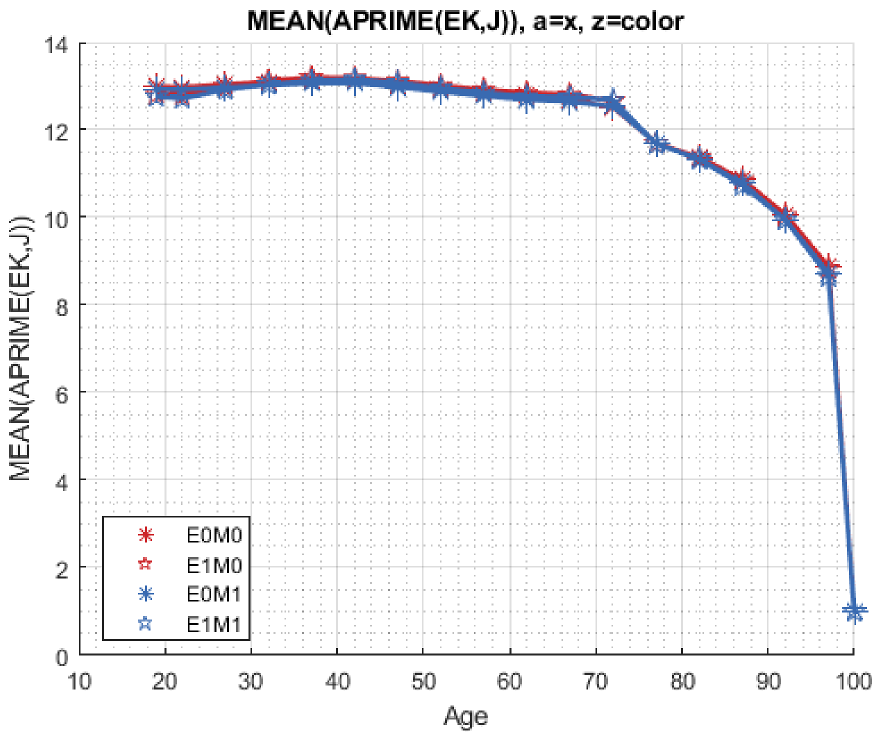
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(EM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

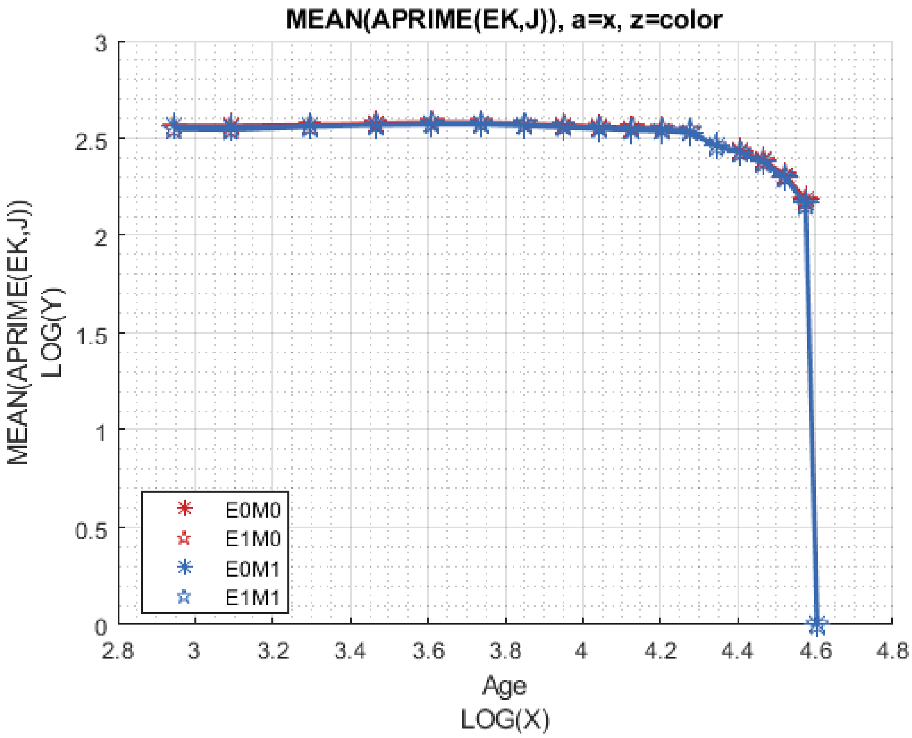




Graph Mean Savings Choices:

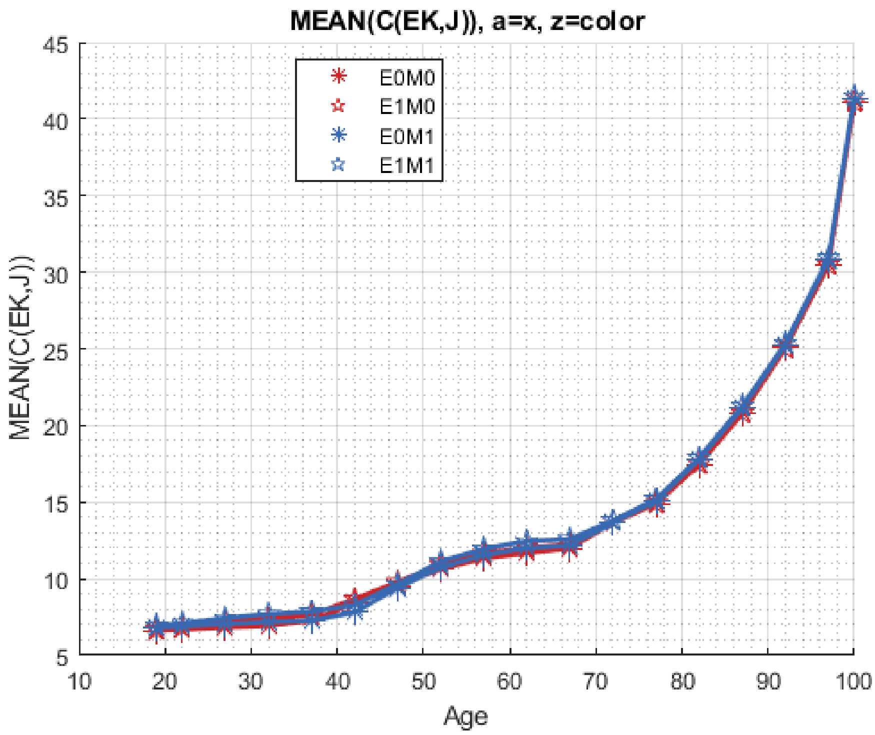
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(EK,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

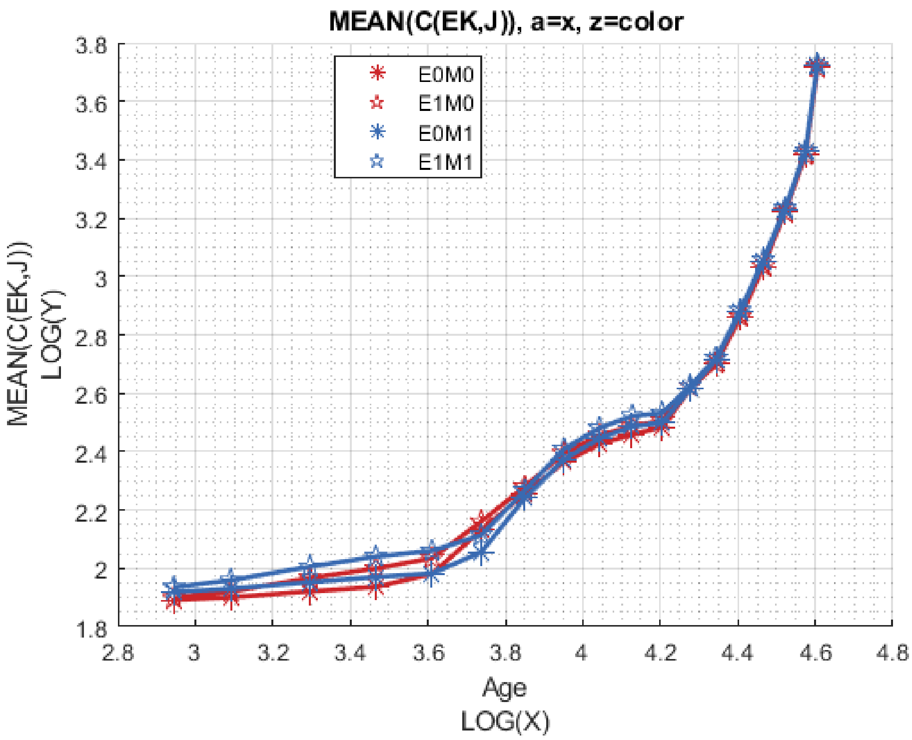




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(EK,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





4.3 Small Test Exact Solution Vectorized Bisection

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for policy function with vectorized bisection. Small Solution Analysis. Small Solution Analysis, husband 5 shocks, wife 1 shocks.

4.3.1 Test SNW_VFI_MAIN Defaults Small

Call the function with defaults parameters.

```
mp_param = snw_mp_param('default_small');
[V_VFI,ap_VFI,cons_VFI,mp_valpol_more] = snw_vfi_main_bisec_vec(mp_param);

SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:18 of 17, time-this-age:0.06059
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:17 of 17, time-this-age:0.052828
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:16 of 17, time-this-age:0.032745
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:15 of 17, time-this-age:0.029085
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:14 of 17, time-this-age:0.035583
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:13 of 17, time-this-age:0.034991
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:12 of 17, time-this-age:0.033648
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:11 of 17, time-this-age:0.032963
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:10 of 17, time-this-age:0.033174
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:9 of 17, time-this-age:0.036843
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:8 of 17, time-this-age:0.04052
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:7 of 17, time-this-age:0.028633
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:6 of 17, time-this-age:0.035108
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:5 of 17, time-this-age:0.033838
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:4 of 17, time-this-age:0.033585
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:3 of 17, time-this-age:0.03214
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:2 of 17, time-this-age:0.028888
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:1 of 17, time-this-age:0.031611
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_small;SNW_MP_CONTROL=default_base;time=0.72345
```

4.3.2 Small Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = [19, 22:5:97, 100];
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'Hshock', eta_H_grid});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

4.3.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm
```

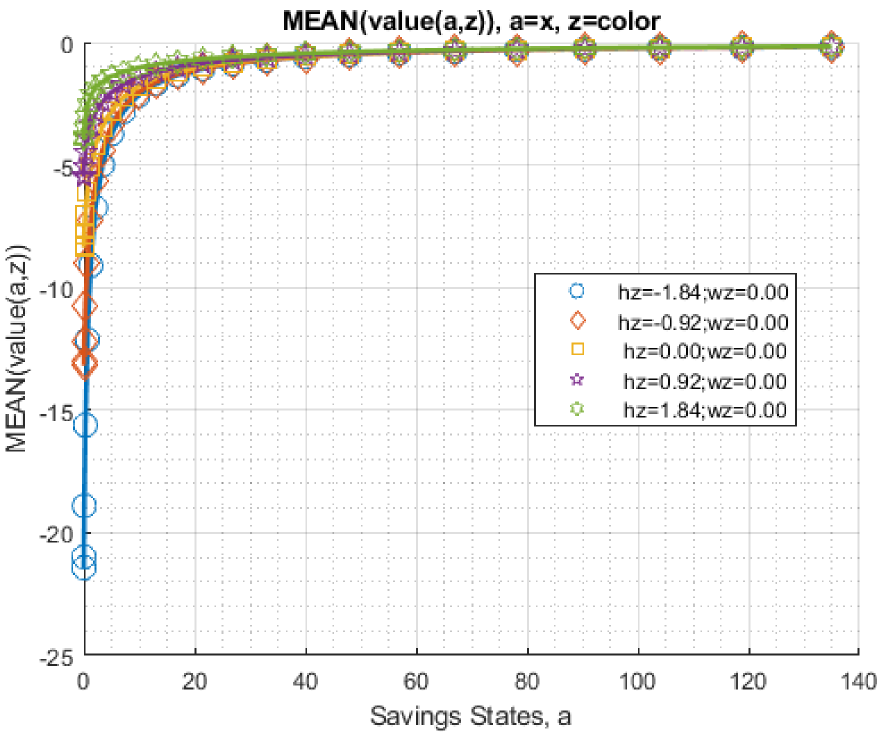
```
xxx  MEAN(VAL(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_Hshock__1_8395    mean_Hshock__0_91976    mean_Hshock_0    mean_Hshoc
    _____    _________    ___________________    ____________________    _____________    __________

      1              0          -21.426                 -13.175               -8.362            -5.4
      2      0.0097656          -20.989                 -13.027              -8.2755            -5.4
      3       0.078125          -18.901                 -12.204              -7.8053            -5.0
      4        0.26367          -15.612                 -10.744              -7.0124            -4.4
      5          0.625          -12.124                 -8.9835              -6.0664            -3.8
      6         1.2207          -9.0979                 -7.2177              -5.0967            -3.3
      7         2.1094          -6.7401                 -5.6532              -4.2107            -2.8
      8         3.3496          -4.9967                 -4.3739              -3.4359            -2.4
      9              5          -3.7353                 -3.3758              -2.7788            -2.0
     10         7.1191          -2.8279                  -2.617              -2.2393            -1.7
     11         9.7656           -2.172                 -2.0455              -1.8057            -1.4
     12         12.998           -1.693                 -1.6153              -1.4614            -1.2
     13         16.875          -1.3389                 -1.2899              -1.1896            -1.0
     14         21.455          -1.0737                  -1.042             -0.97552           -0.85
```

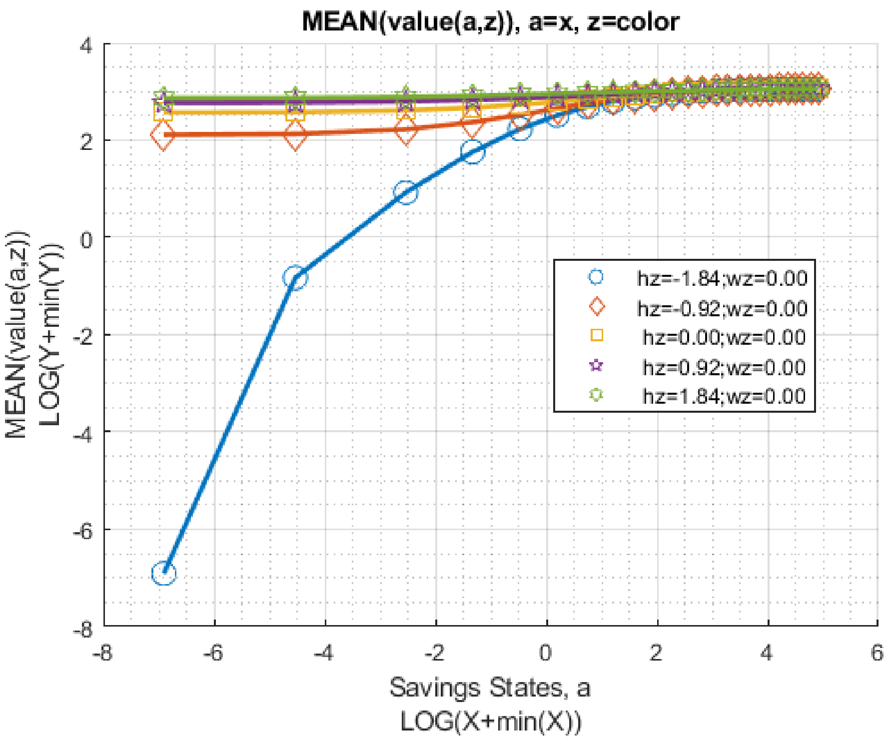
```
     15        26.797          -0.872             -0.85104           -0.80614           -0.71
     16        32.959        -0.71656             -0.70236           -0.67148           -0.61
     17            40        -0.59521             -0.58538           -0.56375            -0.5
     18        47.979        -0.49932             -0.49238           -0.47697           -0.44
     19        56.953        -0.42266             -0.41768           -0.40651           -0.38
     20        66.982        -0.36074              -0.3571           -0.34889           -0.33
     21        78.125        -0.31022             -0.30751           -0.30139           -0.28
     22        90.439        -0.26861             -0.26658           -0.26196           -0.25
     23        103.98        -0.23407             -0.23252           -0.22899           -0.22
     24        118.82        -0.20516             -0.20397           -0.20125           -0.19
     25           135         -0.1808             -0.17987           -0.17775           -0.17

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(A,Z))", ap_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(AP(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings     mean_Hshock__1_8395    mean_Hshock__0_91976    mean_Hshock_0    mean_Hshoc
    -----    ---------    -------------------    --------------------    -------------    ----------

      1             0        3.2159e-05             0.0034995            0.049878            0.24
      2     0.0097656        0.00055365             0.0052722            0.053281            0.24
      3      0.078125          0.021863              0.029676            0.083029             0.2
      4       0.26367           0.13323               0.14751             0.20012            0.38
      5         0.625           0.39134               0.41034             0.45315            0.64
      6        1.2207           0.84131               0.86393             0.91226             1.0
      7        2.1094            1.5303                1.5542              1.6156             1.7
      8        3.3496            2.4876                2.5118               2.573             2.6
      9             5            3.7642                3.7887              3.8498             3.9
     10        7.1191            5.4275                5.4525              5.5145             5.6
     11        9.7656            7.4794                7.5043              7.5679             7.7
     12        12.998            9.9124                9.9329              9.9956             10.
     13        16.875            12.928                 12.95              13.005             13.
     14        21.455            16.529                16.548              16.604             16.
     15        26.797            20.601                20.618              20.668             20.
     16        32.959            25.307                25.325               25.37             25.
     17            40            30.667                30.689              30.742             30.
     18        47.979            36.761                36.782              36.841             36.
     19        56.953            43.773                43.795              43.847             44.
     20        66.982            51.605                51.628              51.688              51
     21        78.125            59.954                59.977              60.037             60.
     22        90.439            69.265                69.288               69.35             69.
     23        103.98             79.75                79.771              79.831             80.
     24        118.82            91.112                91.136              91.198             91.
     25           135            103.47                103.49              103.54             103

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(A,Z))", cons_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(C(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings     mean_Hshock__1_8395    mean_Hshock__0_91976    mean_Hshock_0    mean_Hshoc
    -----    ---------    -------------------    --------------------    -------------    ----------

      1             0          0.30273               0.43104             0.68779             1.21
      2     0.0097656          0.31374               0.44074             0.69581             1.22
      3      0.078125          0.37308               0.49663             0.74605              1.2
      4       0.26367          0.48039               0.59659               0.846             1.37
      5         0.625          0.64735               0.75745              1.0153             1.54
      6        1.2207          0.89649                1.0013              1.2519             1.79
```

7	2.1094	1.2479	1.3498	1.5854	2.16
8	3.3496	1.7393	1.8394	2.0734	2.67
9	5	2.3872	2.4859	2.7182	3.29
10	7.1191	3.1917	3.289	3.5191	4.05
11	9.7656	4.2188	4.3155	4.543	5.
12	12.998	5.5439	5.6447	5.8722	6.39
13	16.875	7.0334	7.133	7.3676	7.88
14	21.455	8.754	8.8551	9.0887	9.61
15	26.797	10.886	10.989	11.228	11.7
16	32.959	13.336	13.439	13.682	14.2
17	40	16.151	16.249	16.485	17.0
18	47.979	19.321	19.42	19.65	20
19	56.953	22.728	22.827	23.062	23.6
20	66.982	26.539	26.637	26.865	27.
21	78.125	31.125	31.222	31.451	31.9
22	90.439	36.11	36.207	36.433	36.9
23	103.98	41.348	41.447	41.676	42.
24	118.82	47.206	47.302	47.528	48.
25	135	53.636	53.735	53.966	54.5

Graph Mean Values:

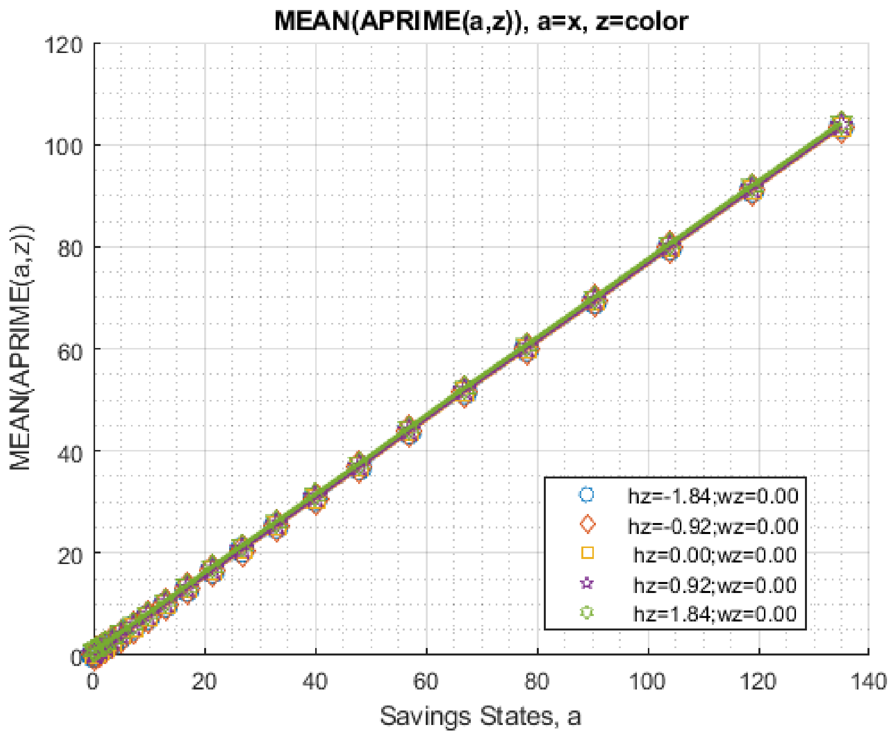
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

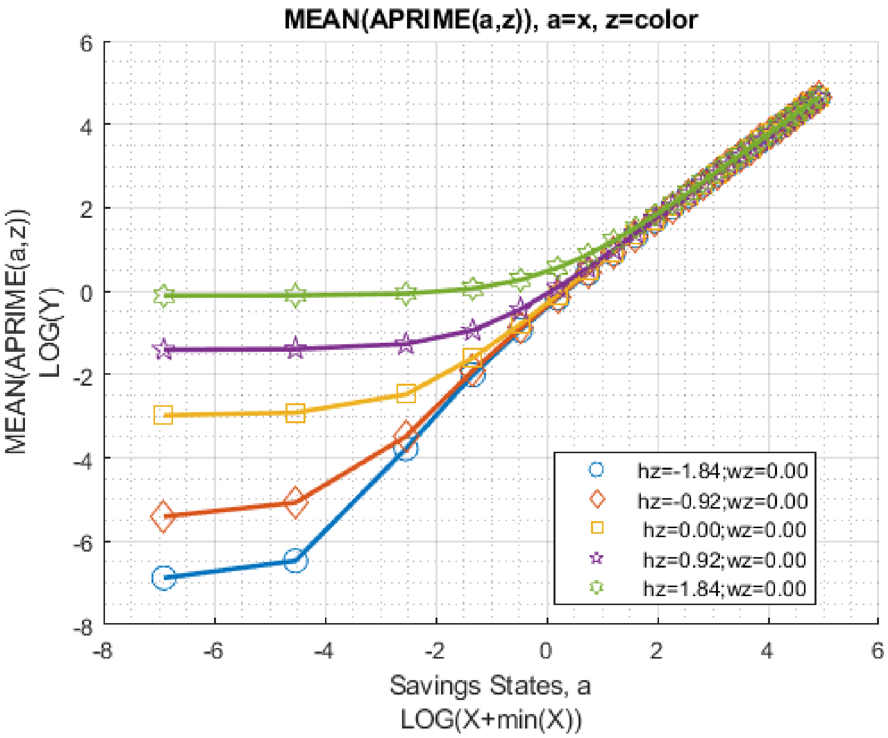




Graph Mean Savings Choices:

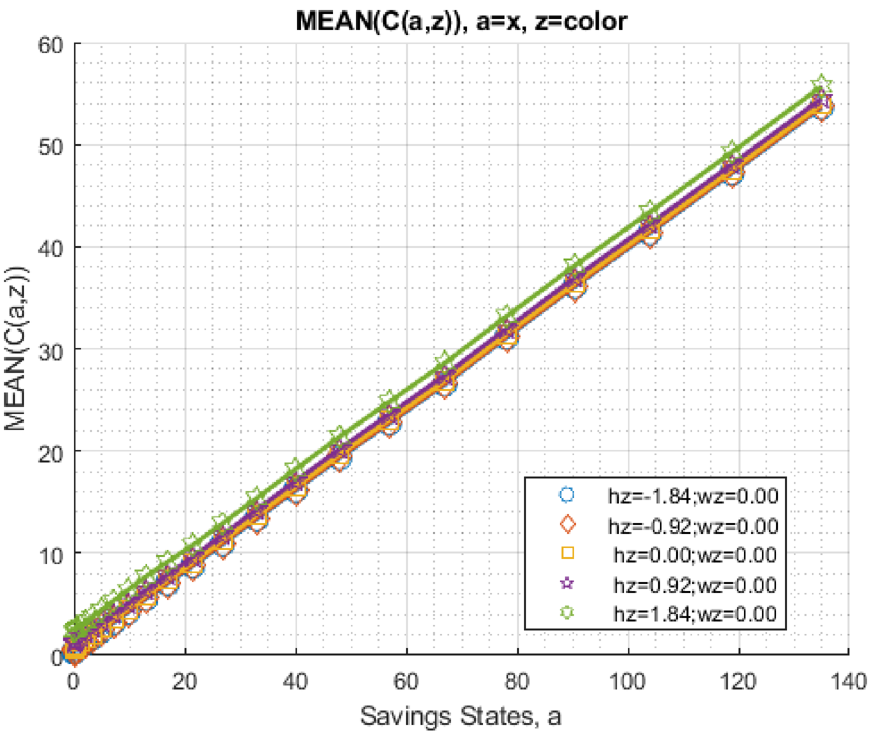
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(a,z))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

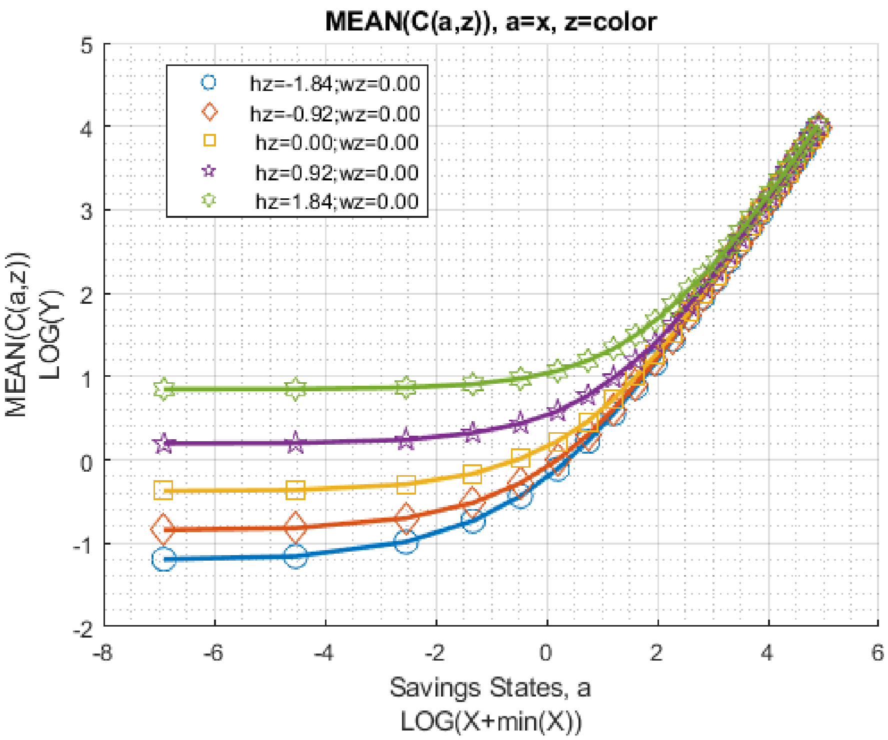




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





4.3.4 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["k0M0", "K1M0", "K2M0", "k0M1", "K1M1", "K2M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = { 'o', 'd' ,'s', 'o', 'd', 's'};
mp_support_graph('cl_colors') = {'red', 'red', 'red', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(KM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per
```

```
xxx  MEAN(VAL(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0        -4.7384        -4.2839        -3.9125        -3.6403        -3.4202
      2        2       0        -6.2307        -5.5732         -5.014        -4.5943        -4.2483
      3        3       0        -6.9818        -6.3368        -5.7685        -5.3334        -4.9708
      4        1       1        -4.1822        -3.7934        -3.4691        -3.2086         -2.984
      5        2       1         -5.157         -4.667        -4.2348        -3.8784        -3.5654
      6        3       1        -5.5929        -5.1267        -4.7056         -4.352        -4.0378
```

```
% Aprime Choice
```

```
tb_az_ap = ff_summ_nd_array("MEAN(AP(KM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

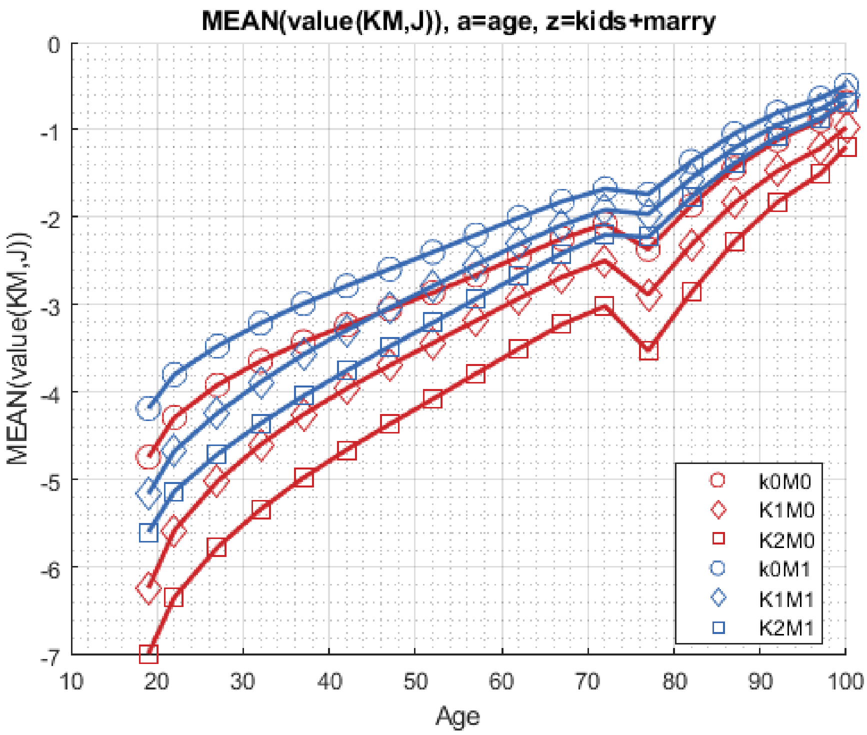
```
xxx  MEAN(AP(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        34.931         34.726         34.665         34.554         34.362
      2        2       0        34.603         34.334         34.198         33.995         33.692
      3        3       0        34.187         33.968         33.877         33.705         33.427
      4        1       1        34.821         34.617         34.566         34.458         34.268
      5        2       1         34.67          34.45         34.364         34.205         33.951
      6        3       1        34.303         34.118         34.065         33.937         33.705
```

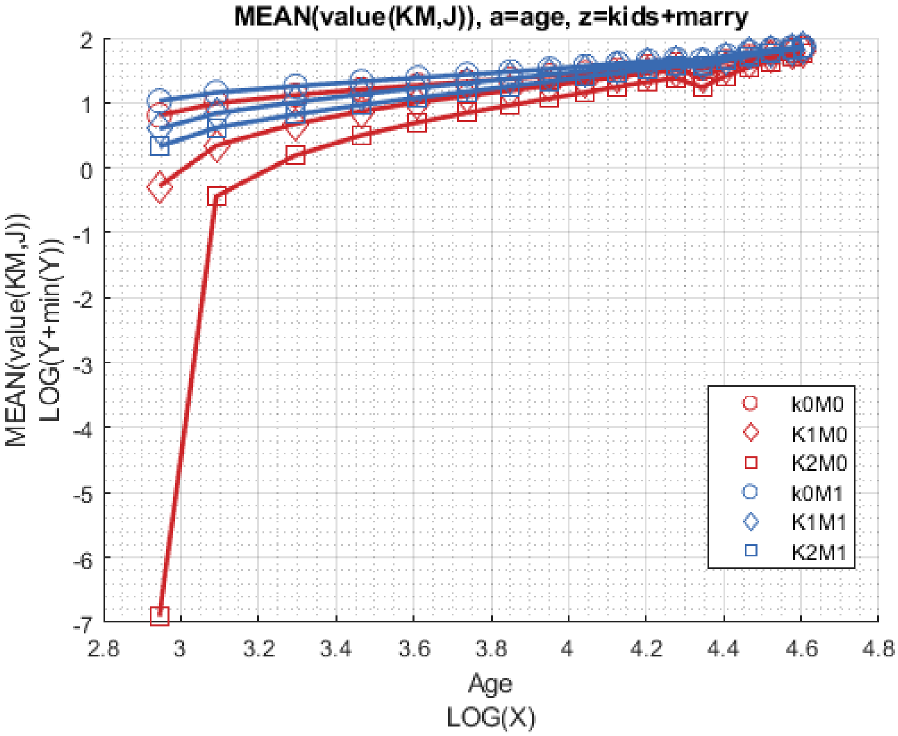
```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(KM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

```
xxx  MEAN(C(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        6.8531         7.1729         7.4988         7.8167         8.1435
      2        2       0         7.182         7.5653         7.9659         8.3756          8.813
      3        3       0        7.5973          7.931         8.2872         8.6657         9.0783
      4        1       1        7.1848         7.5242         7.8662         8.2047          8.552
      5        2       1        7.3021         7.6535         8.0269          8.412         8.8205
      6        3       1        7.6455         7.9599          8.297         8.6497         9.0324
```

Graph Mean Values:

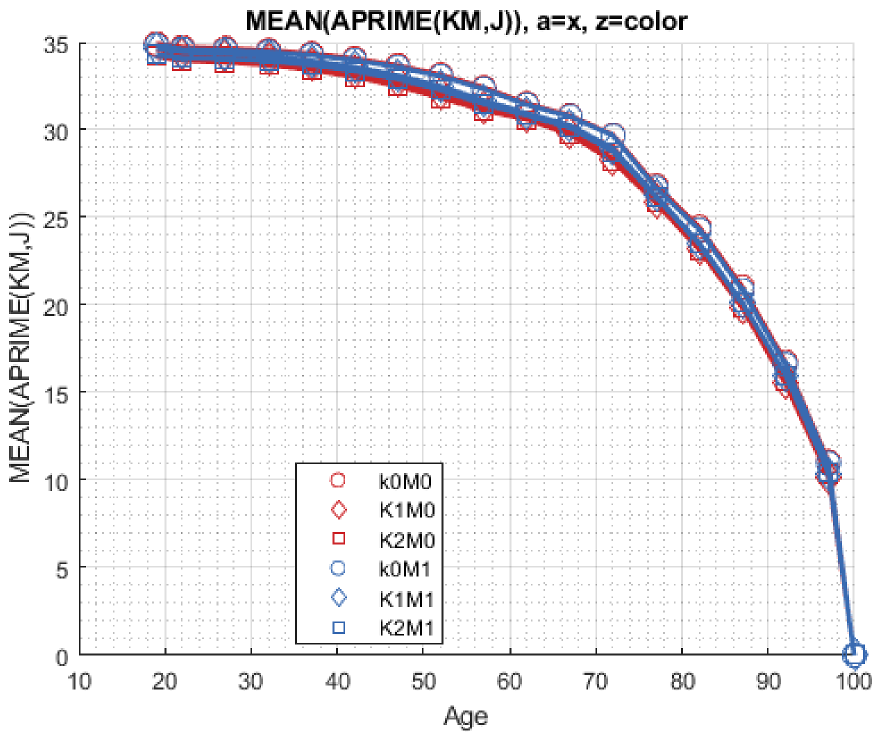
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(KM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

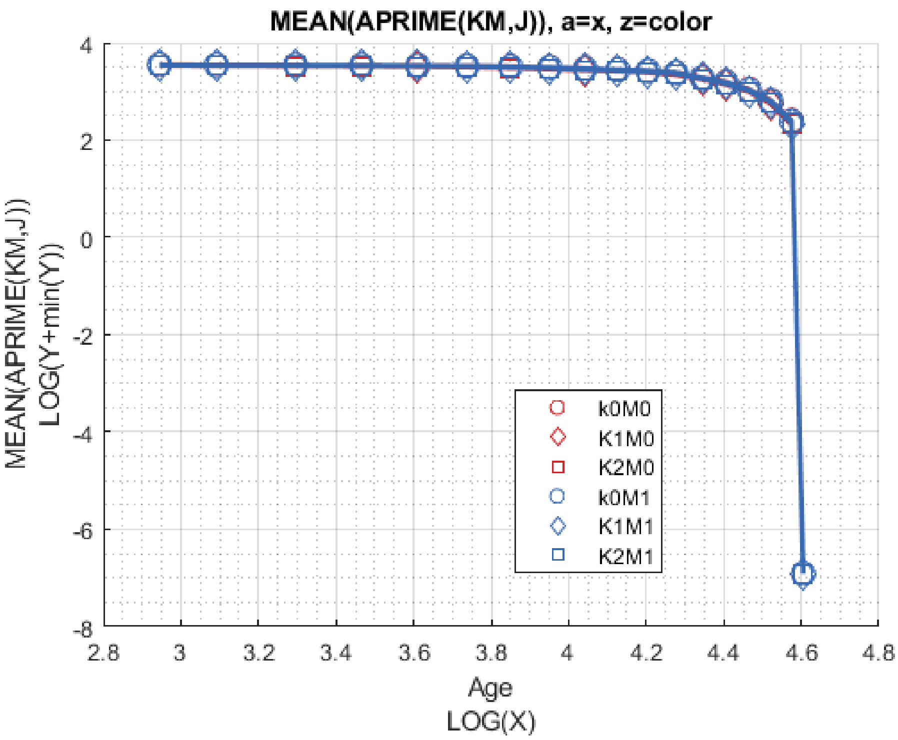




Graph Mean Savings Choices:

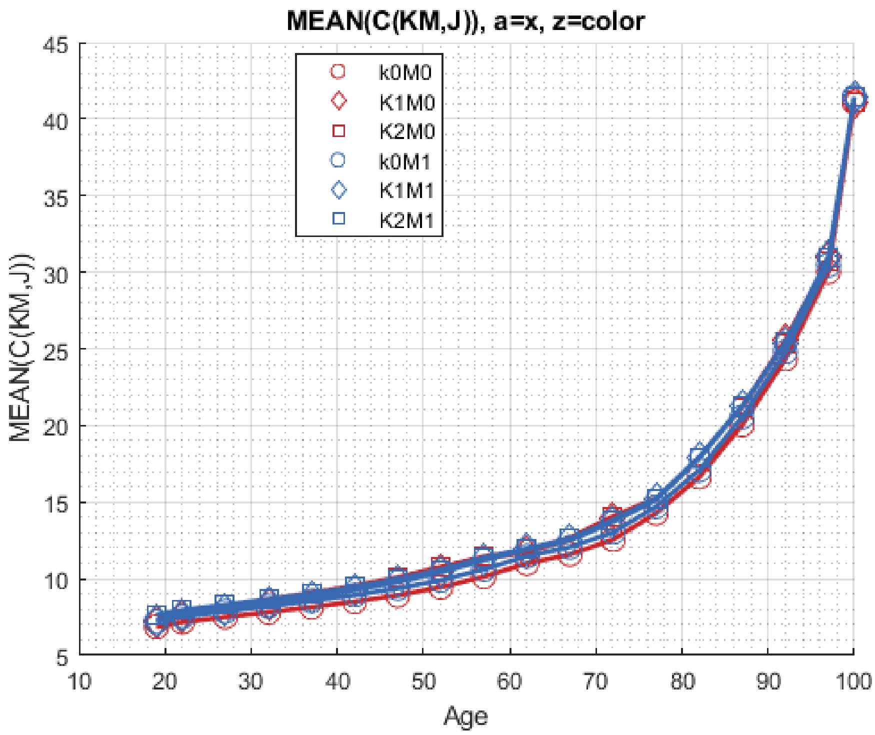
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(KM,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

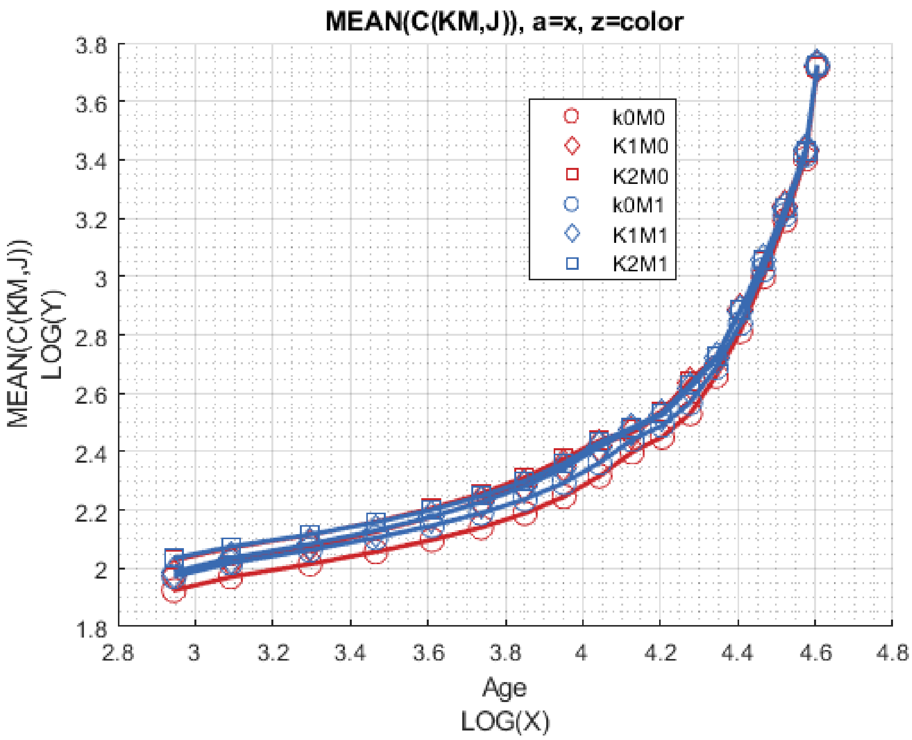




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





4.3.5 Analyze Education and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

MEAN(VAL(EKM,J)), MEAN(AP(EKM,J)), MEAN(C(EKM,J))

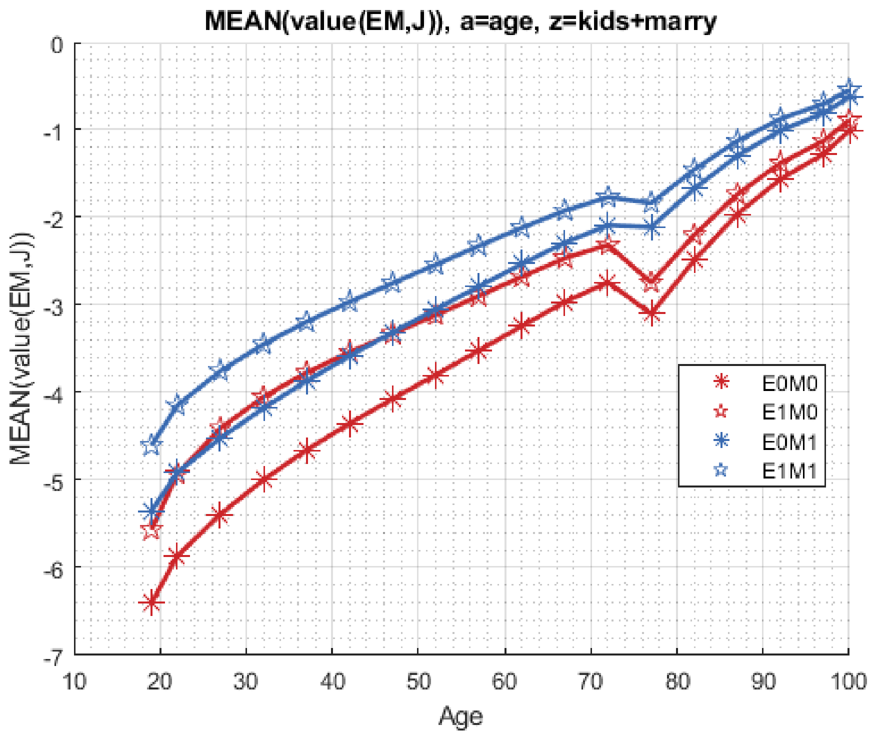
Tabulate value and policies:

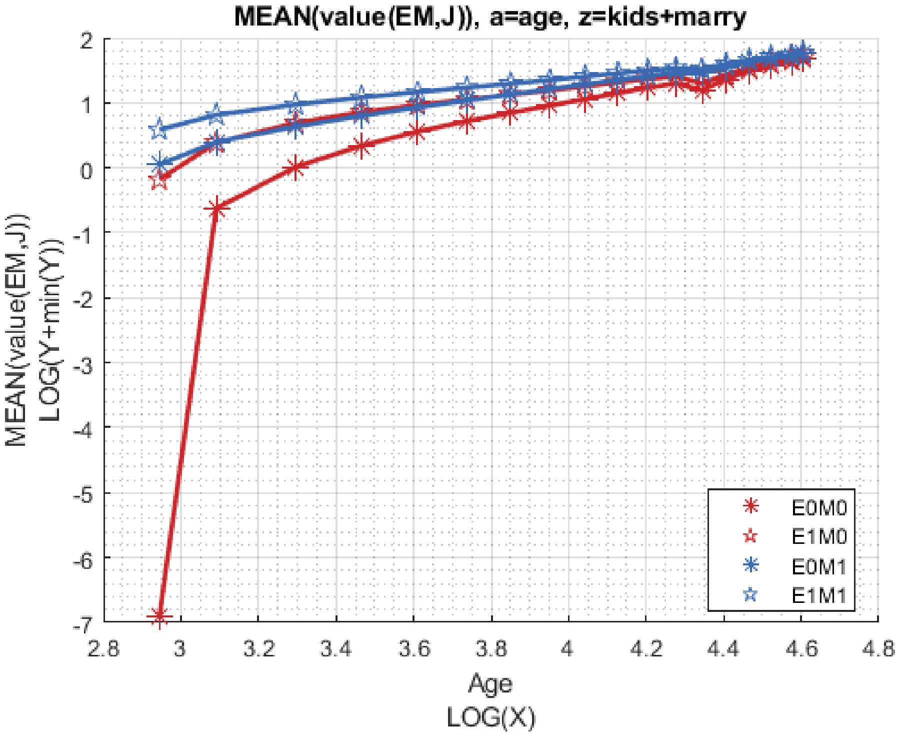
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(EKM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(VAL(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     edu     marry     mean_age_19     mean_age_22     mean_age_27     mean_age_32     mean_age_37
    -----     ---     -----     -----------     -----------     -----------     -----------     -----------

      1        0        0         -6.4015         -5.8666         -5.3879         -4.9966         -4.6557
      2        1        0         -5.5658         -4.9294         -4.4088         -4.0487         -3.7705
      3        0        1           -5.35          -4.913         -4.5196         -4.1777          -3.867
      4        1        1         -4.6046         -4.1451         -3.7534         -3.4483         -3.1912

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(EKM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p
```

```
xxx  MEAN(AP(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     edu     marry     mean_age_19     mean_age_22     mean_age_27     mean_age_32     mean_age_37
    -----     ---     -----     -----------     -----------     -----------     -----------     -----------

      1        0        0         34.682          34.444          34.272          34.048          33.753
      2        1        0         34.465          34.241          34.222          34.121          33.901
      3        0        1         34.725          34.514          34.372          34.177          33.914
      4        1        1          34.47          34.277          34.291          34.223          34.035

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(EKM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(C(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     edu     marry     mean_age_19     mean_age_22     mean_age_27     mean_age_32     mean_age_37
    -----     ---     -----     -----------     -----------     -----------     -----------     -----------

      1        0        0         7.1022          7.4087          7.7357          8.0845          8.4713
      2        1        0         7.3195          7.7041          8.0988          8.4875          8.8852
      3        0        1         7.2307          7.5253          7.8393          8.1757          8.5471
      4        1        1         7.5242          7.8997          8.2875          8.6685          9.0562
```

Graph Mean Values:

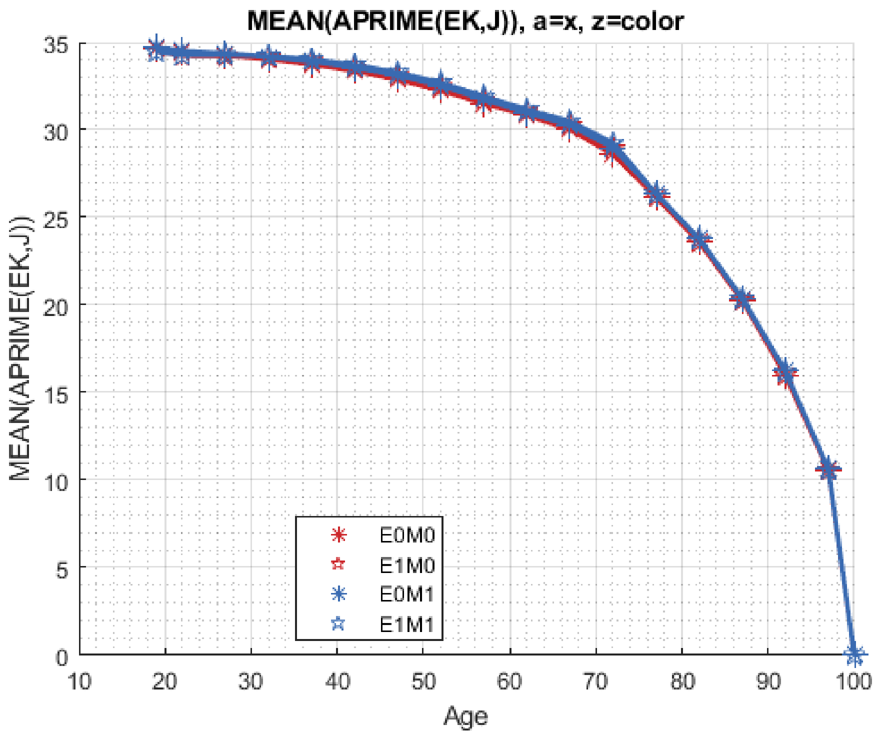
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(EM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

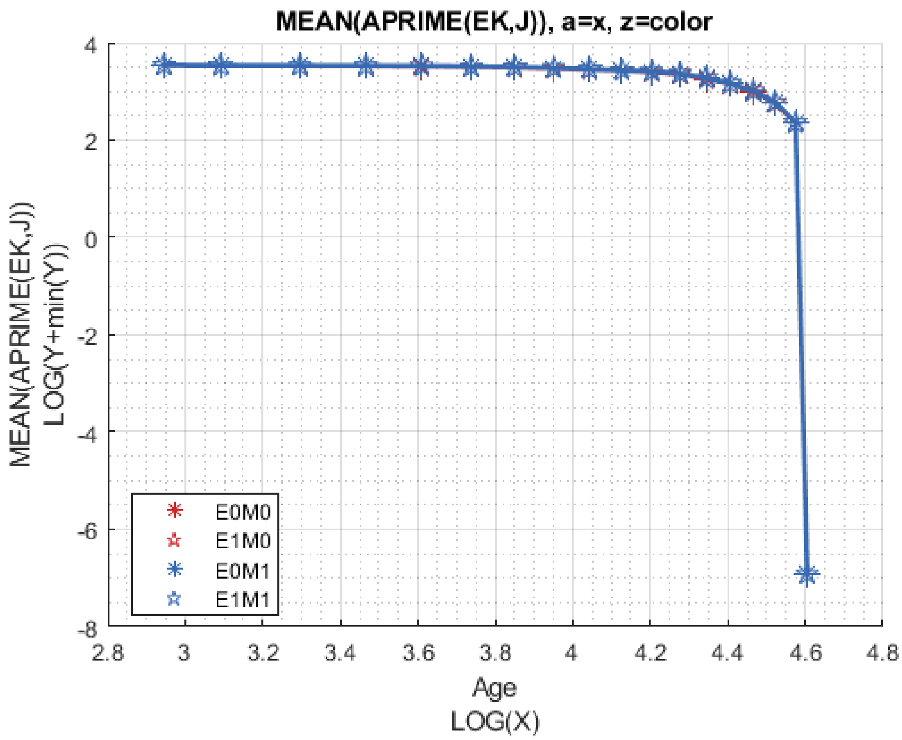




Graph Mean Savings Choices:

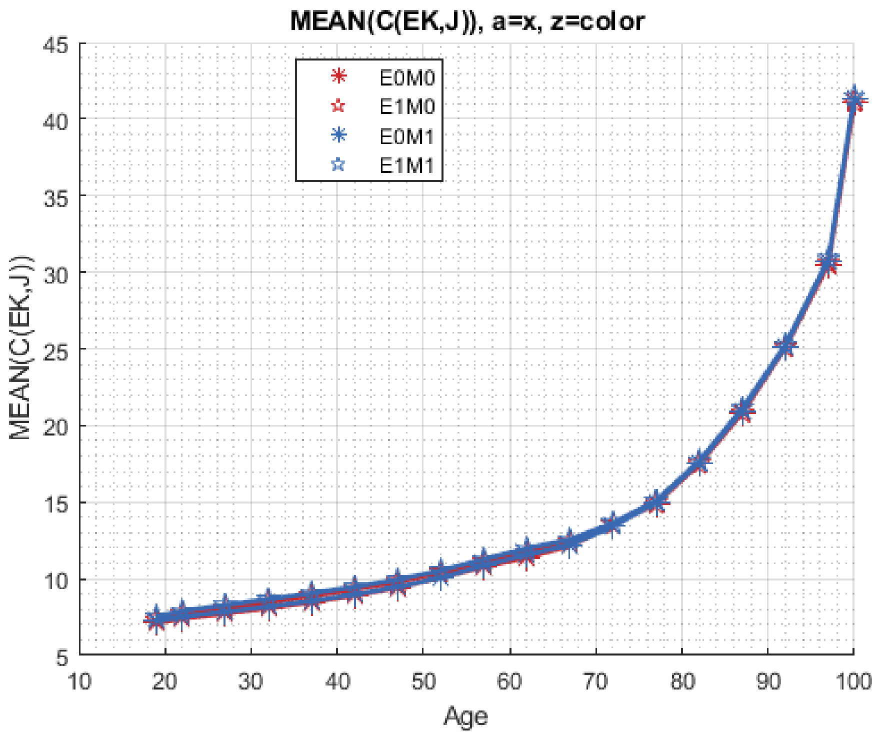
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(EK,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

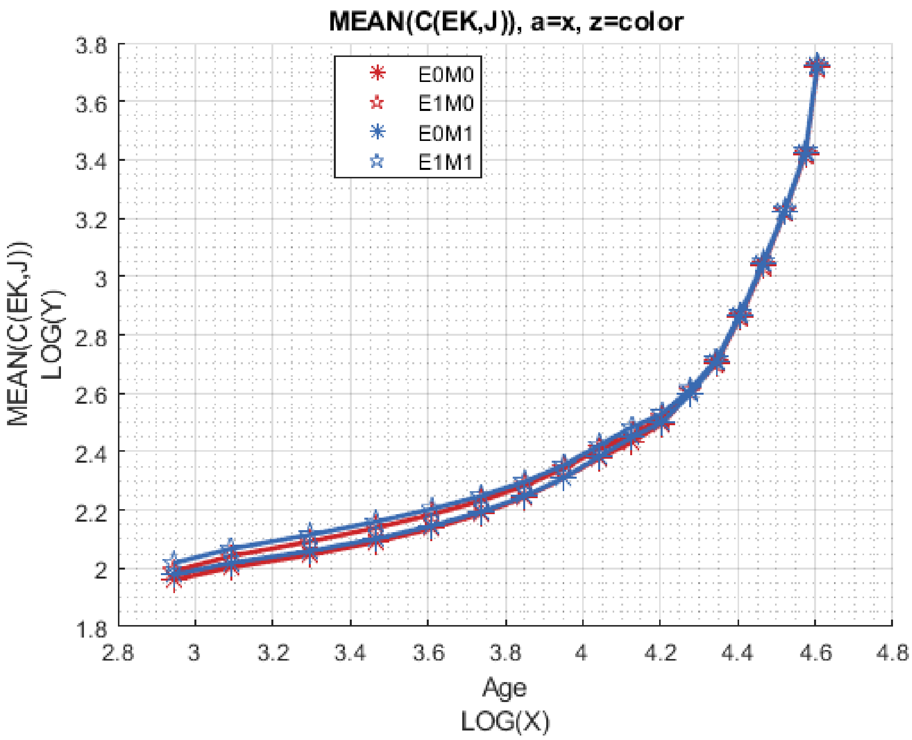




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(EK,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





4.4 Small Test Exact Solution Spousal Shocks

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for policy function with vectorized bisection. Small Solution Analysis, husband 5 shocks, wife 3 shocks.

4.4.1 Test SNW_VFI_MAIN Defaults Small

Call the function with default parameters.

```
mp_param = snw_mp_param('default_small53');
[V_VFI,ap_VFI,cons_VFI,mp_valpol_more] = snw_vfi_main_bisec_vec(mp_param);

SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:18 of 17, time-this-age:0.10593
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:17 of 17, time-this-age:0.086471
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:16 of 17, time-this-age:0.079532
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:15 of 17, time-this-age:0.095977
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:14 of 17, time-this-age:0.081549
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:13 of 17, time-this-age:0.08703
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:12 of 17, time-this-age:0.089059
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:11 of 17, time-this-age:0.094402
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:10 of 17, time-this-age:0.10011
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:9 of 17, time-this-age:0.093424
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:8 of 17, time-this-age:0.095447
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:7 of 17, time-this-age:0.10832
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:6 of 17, time-this-age:0.089896
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:5 of 17, time-this-age:0.09249
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:4 of 17, time-this-age:0.094699
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:3 of 17, time-this-age:0.087703
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:2 of 17, time-this-age:0.0939
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:1 of 17, time-this-age:0.087152
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_small53;SNW_MP_CONTROL=default_base;time=1.715
```

4.4.2 Small Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = [19, 22:5:97, 100];
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

4.4.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 9; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm
```

xxx MEAN(VAL(A,Z)) xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5	mean
1	0	-24.423	-14.383	-9.0838	-6.1681	-4.6368	-
2	0.0097656	-23.655	-14.141	-8.9428	-6.0477	-4.521	-2
3	0.078125	-20.274	-12.864	-8.2087	-5.441	-3.9447	-1
4	0.26367	-16.052	-10.951	-7.1333	-4.627	-3.2033	-1
5	0.625	-12.169	-8.9647	-6.0525	-3.918	-2.6128	-1
6	1.2207	-8.9956	-7.1143	-5.0287	-3.3224	-2.17	-8
7	2.1094	-6.6026	-5.5315	-4.126	-2.8111	-1.8354	-6
8	3.3496	-4.8705	-4.2629	-3.3542	-2.3677	-1.5738	-4
9	5	-3.6341	-3.2853	-2.7087	-1.9911	-1.3595	-3
10	7.1191	-2.7516	-2.5471	-2.1826	-1.6748	-1.1772	-2
11	9.7656	-2.1163	-1.9932	-1.7614	-1.4073	-1.0193	-2
12	12.998	-1.653	-1.5768	-1.4275	-1.1818	-0.88344	-1

```
     13        16.875      -1.3103        -1.2619        -1.1642       -0.99332       -0.76691        -1
     14        21.455      -1.0532        -1.0216       -0.95651       -0.83698       -0.66655        -1
     15        26.797     -0.85714       -0.83614       -0.79193        -0.7076       -0.57964       -0.
     16        32.959      -0.7057       -0.69138       -0.66083       -0.60072       -0.50456       -0.
     17            40     -0.58719       -0.57722       -0.55573       -0.51237       -0.43988       -0.
     18        47.979     -0.49334       -0.48626       -0.47088       -0.43924       -0.38431       -0.
     19        56.953     -0.41818       -0.41306       -0.40187       -0.37849       -0.33661       -0.
     20        66.982     -0.35736       -0.35359       -0.34532       -0.32783       -0.29568       -0.
     21        78.125     -0.30764       -0.30483       -0.29864       -0.28541       -0.26055       -0.
     22        90.439     -0.26663       -0.26451       -0.25982       -0.24971       -0.23034       -0.
     23        103.98     -0.23253       -0.23091       -0.22732       -0.21951       -0.20431       -0.
     24        118.82     -0.20396       -0.20271       -0.19994       -0.19386       -0.18184        -0
     25           135     -0.17985       -0.17888       -0.17671       -0.17193       -0.16238       -0.
```

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(A,Z))", ap_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per
```

```
xxx  MEAN(AP(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5     mean
    _____    _________    __________     __________     __________     __________     __________     ____

      1             0    3.2159e-05     0.0024668       0.032046        0.17185        0.75713       0.00
      2     0.0097656    0.00055365     0.0039513       0.034103        0.17524        0.76251       0.00
      3      0.078125      0.015815      0.020023       0.052271        0.20181         0.8015        0.0
      4       0.26367      0.094638      0.094808        0.12856        0.29193        0.91314         0.
      5         0.625       0.31851       0.32442        0.35753        0.52484         1.1543         0.
      6        1.2207       0.75143       0.75176        0.79303        0.95962         1.5772          0
      7        2.1094        1.4241        1.4284         1.4709         1.6269         2.2293          1
      8        3.3496        2.3733        2.3796          2.421         2.5737         3.1535          2
      9             5        3.6394        3.6466         3.6884         3.8506         4.3901          3
     10        7.1191        5.2875        5.2955         5.3372         5.5015         5.9876          5
     11        9.7656        7.3153        7.3234         7.3642         7.5288         8.0042          7
     12        12.998        9.7556        9.7616         9.8006          9.967          10.49          9
     13        16.875        12.766        12.773         12.807          12.97         13.562          1
     14        21.455        16.338        16.342         16.377         16.524         17.139          1
     15        26.797        20.401        20.403         20.431          20.57         21.185          2
     16        32.959        25.088        25.095         25.124         25.248         25.842          2
     17            40        30.463        30.471          30.51         30.633         31.193
     18        47.979        36.558        36.567         36.609         36.754         37.274          3
     19        56.953         43.57        43.576         43.612         43.757         44.279          4
     20        66.982        51.378        51.387         51.429         51.567         52.101          5
     21        78.125        59.666        59.675         59.721         59.878          60.41
     22        90.439         69.02        69.027         69.069         69.228         69.775
     23        103.98        79.509        79.516         79.557         79.706          80.27          7
     24        118.82         90.88        90.887         90.929         91.074         91.625          9
     25           135        103.23        103.23         103.27         103.42         103.97          1
```

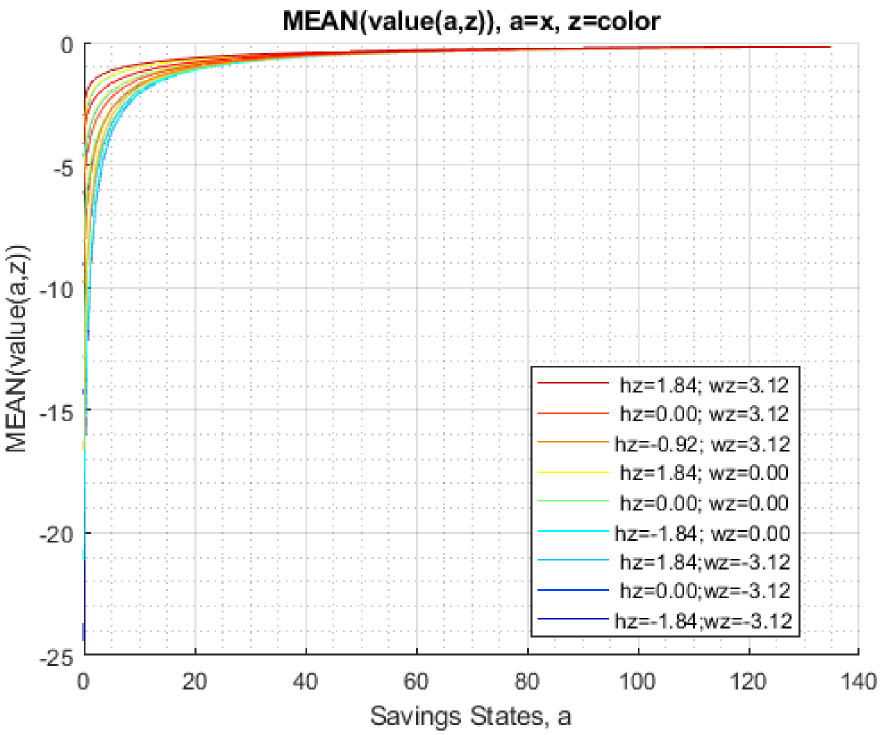
```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(A,Z))", cons_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per
```

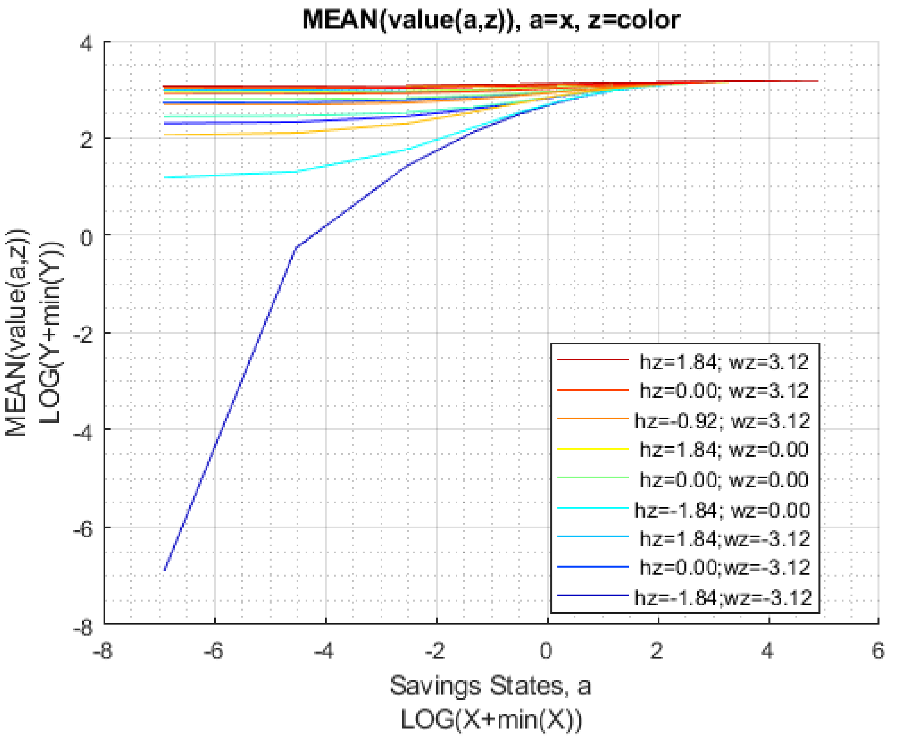
```
xxx  MEAN(C(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5     mean
    _____    _________    __________     __________     __________     __________     __________     ____

      1             0      0.16881        0.29125        0.55695         1.1299         2.3014        0.2
      2     0.0097656      0.17988        0.30127        0.56634         1.1379         2.3074        0.3
      3      0.078125      0.24566        0.36575        0.62833         1.1912         2.3483        0.3
      4       0.26367      0.38648        0.50942        0.76949          1.318         2.4533         0.
```

5	0.625	0.58919	0.70456	0.96359	1.5071	2.6337	0.6
6	1.2207	0.85722	0.97603	1.2247	1.7675	2.9054	0.
7	2.1094	1.2268	1.3395	1.5846	2.1362	3.2884	1.
8	3.3496	1.7279	1.837	2.0808	2.6338	3.8076	1.
9	5	2.3875	2.4944	2.7359	3.2775	4.4904	2.
10	7.1191	3.2081	3.3133	3.5534	4.0911	5.3563	3.
11	9.7656	4.2601	4.3643	4.6042	5.1402	6.4149	4.
12	12.998	5.5783	5.6843	5.9251	6.4581	7.684	5.
13	16.875	7.0733	7.1787	7.4234	7.9594	9.1154	7.
14	21.455	8.823	8.9302	9.1737	9.7243	10.857	8.
15	26.797	10.964	11.073	11.324	11.882	13.013	10
16	32.959	13.433	13.538	13.787	14.36	15.511	13
17	40	16.233	16.337	16.576	17.149	18.335	16
18	47.979	19.402	19.505	19.741	20.291	21.516	19
19	56.953	22.81	22.914	23.156	23.707	24.93	22
20	66.982	26.645	26.747	26.982	27.54	28.75	26
21	78.125	31.292	31.394	31.626	32.165	33.377	31
22	90.439	36.234	36.338	36.573	37.11	38.307	36
23	103.98	41.468	41.572	41.808	42.355	43.535	41
24	118.82	47.317	47.421	47.656	48.207	49.4	47
25	135	53.752	53.857	54.095	54.642	55.842	53

Graph Mean Values:

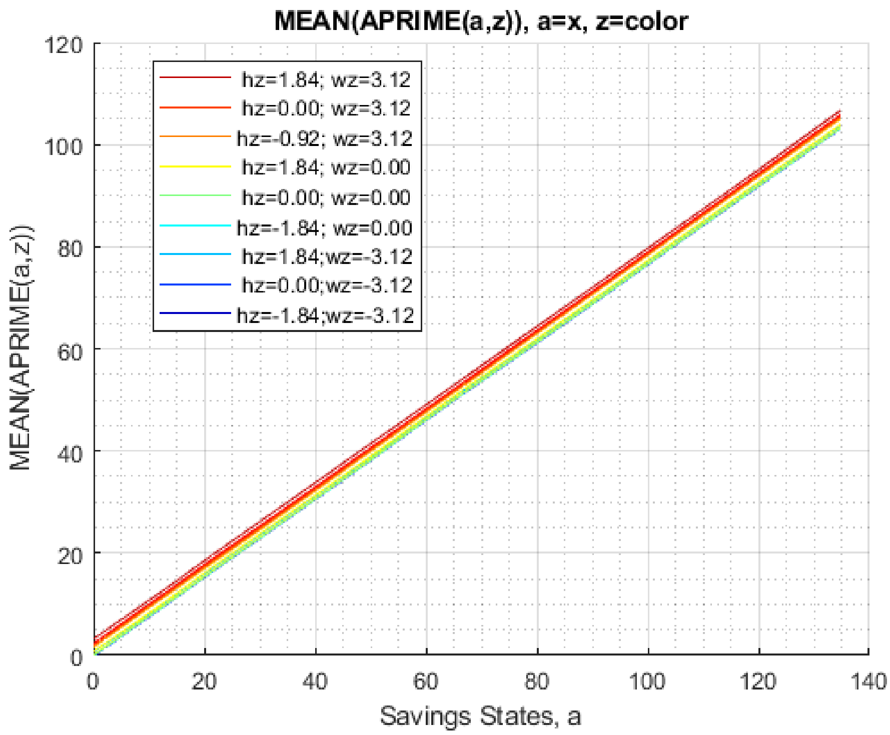
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

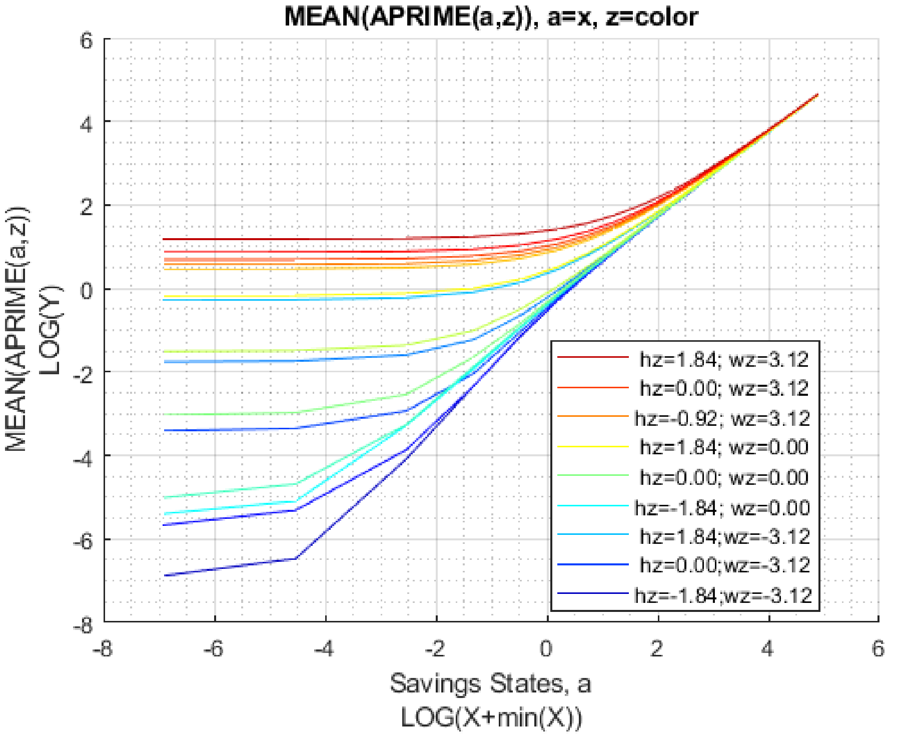




Graph Mean Savings Choices:

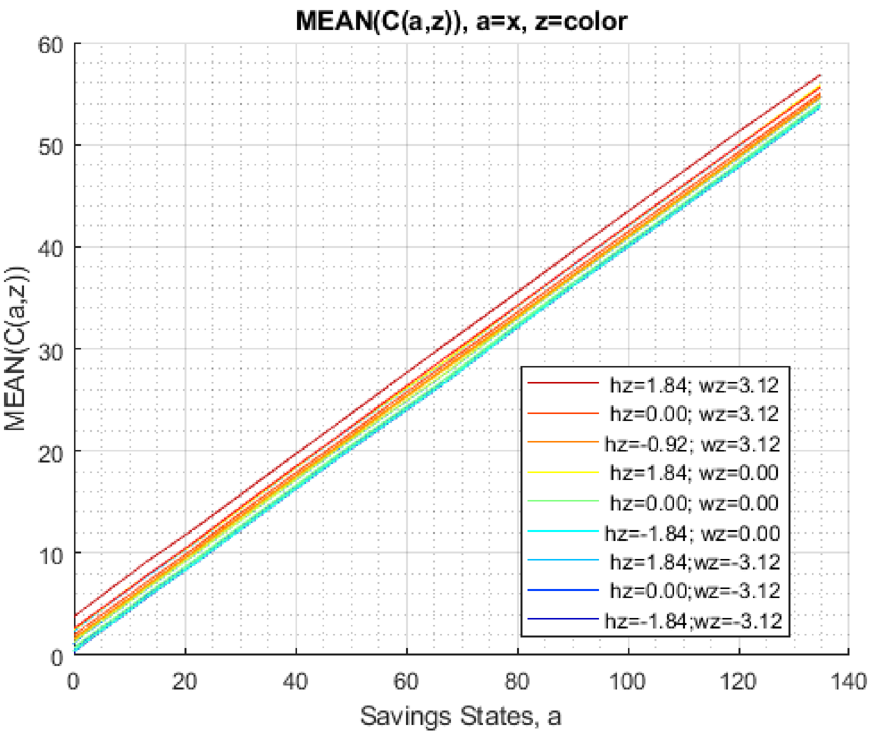
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(a,z))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

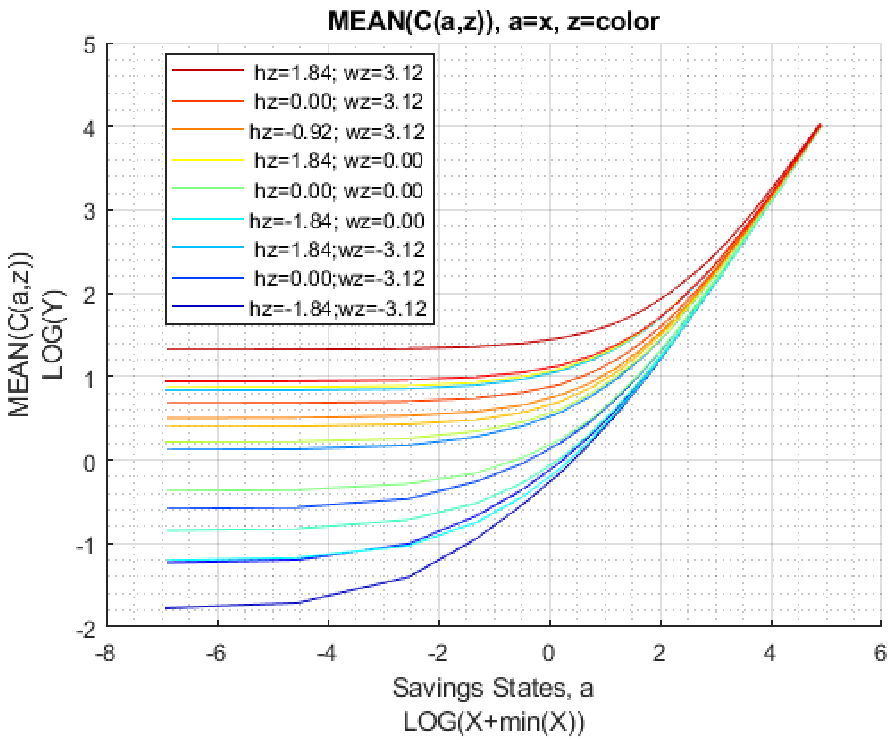




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





4.4.4 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["k0M0", "K1M0", "K2M0", "k0M1", "K1M1", "K2M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = { 'o', 'd' ,'s', 'o', 'd', 's'};
mp_support_graph('cl_colors') = {'red', 'red', 'red', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(KM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(VAL(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0        -4.7384        -4.2839        -3.9125        -3.6403        -3.4202
      2        2       0        -6.2307        -5.5732         -5.014        -4.5943        -4.2483
      3        3       0        -6.9818        -6.3368        -5.7685        -5.3334        -4.9708
      4        1       1        -3.4759        -3.1424        -2.8538        -2.6272        -2.4359
      5        2       1         -4.325        -3.9032        -3.5196        -3.2101        -2.9428
      6        3       1        -4.7282        -4.3225        -3.9432        -3.6317        -3.3597

% Aprime Choice
```

```
tb_az_ap = ff_summ_nd_array("MEAN(AP(KM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

xxx MEAN(AP(KM,J)) xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	kids	marry	mean_age_19	mean_age_22	mean_age_27	mean_age_32	mean_age_3
1	1	0	34.931	34.726	34.665	34.554	34.362
2	2	0	34.603	34.334	34.198	33.995	33.692
3	3	0	34.187	33.968	33.877	33.705	33.427
4	1	1	35.713	35.611	35.7	35.727	35.66
5	2	1	35.368	35.246	35.284	35.242	35.101
6	3	1	34.903	34.81	34.86	34.834	34.7

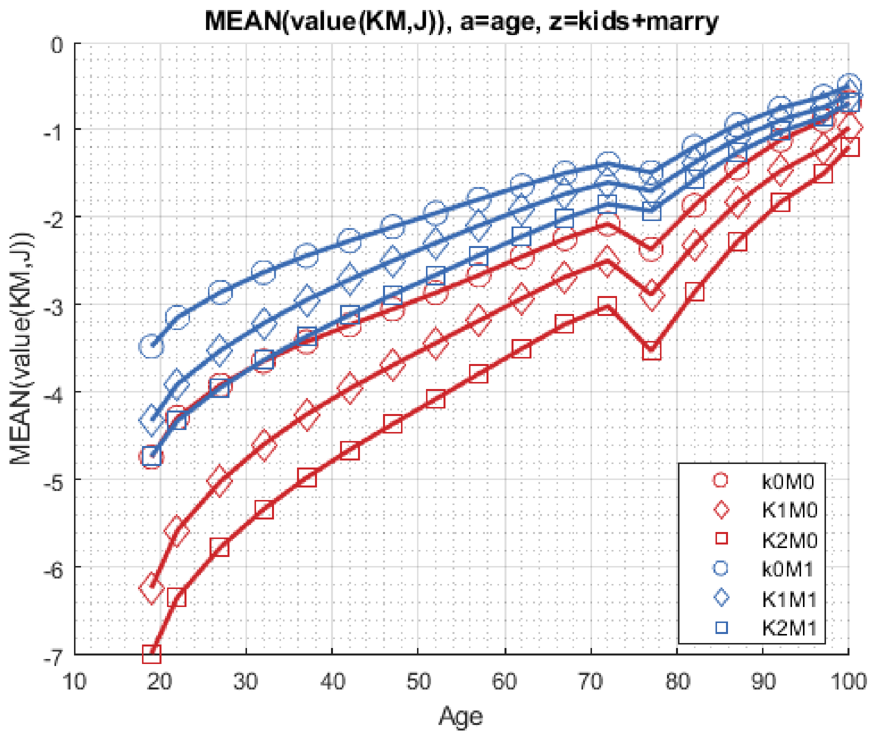
```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(KM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

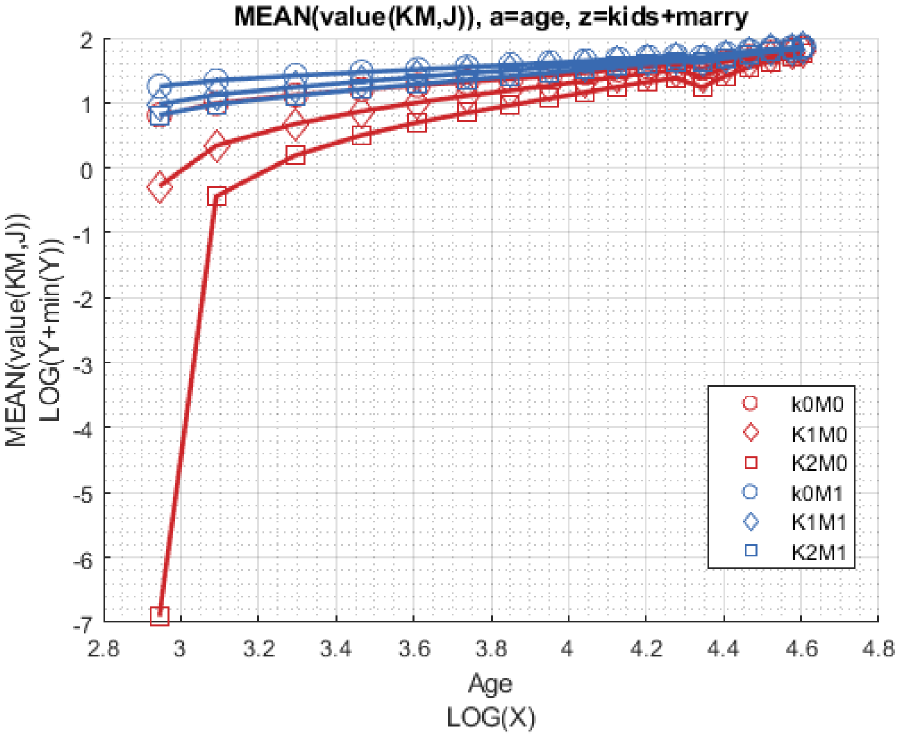
xxx MEAN(C(KM,J)) xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	kids	marry	mean_age_19	mean_age_22	mean_age_27	mean_age_32	mean_age_3
1	1	0	6.8531	7.1729	7.4988	7.8167	8.1435
2	2	0	7.182	7.5653	7.9659	8.3756	8.813
3	3	0	7.5973	7.931	8.2872	8.6657	9.0783
4	1	1	7.7992	8.182	8.5624	8.9321	9.311
5	2	1	7.879	8.2553	8.6555	9.0645	9.4908
6	3	1	8.1608	8.4911	8.8566	9.2297	9.6299

Graph Mean Values:

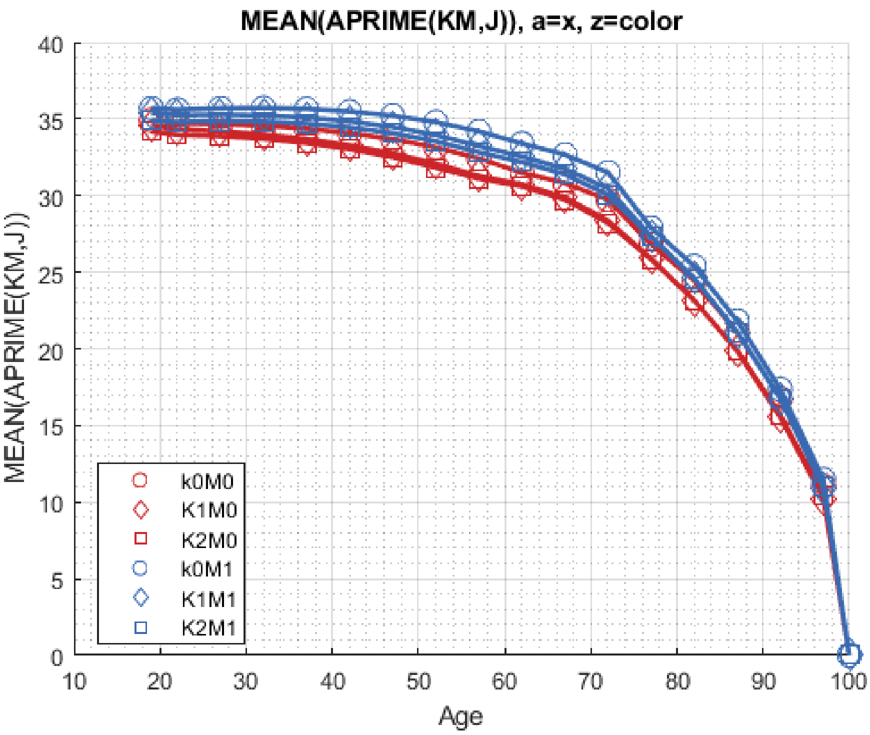
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(KM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

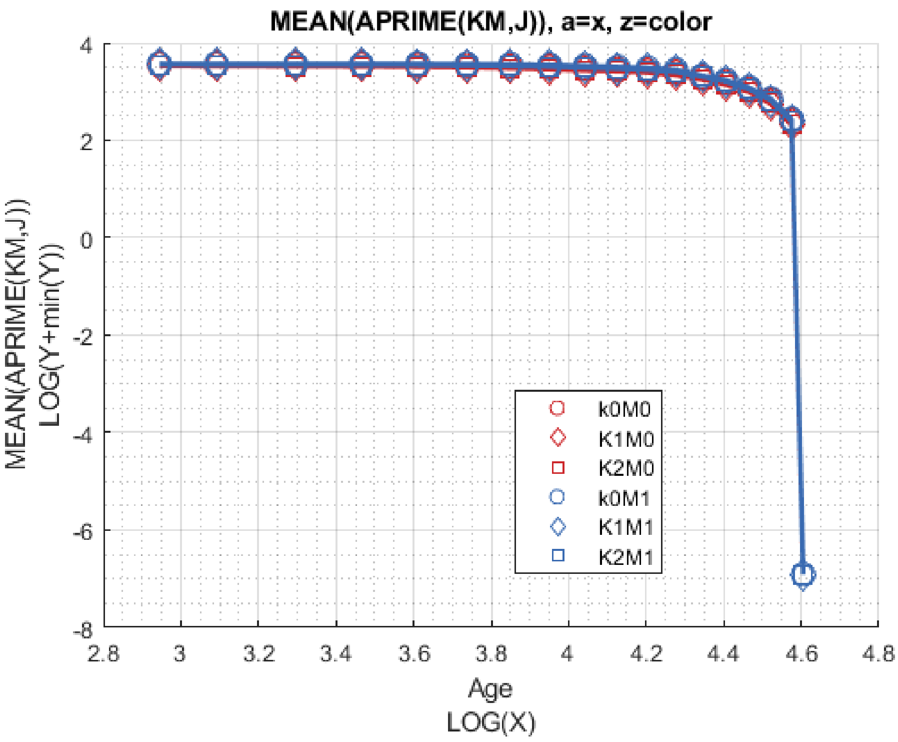




Graph Mean Savings Choices:

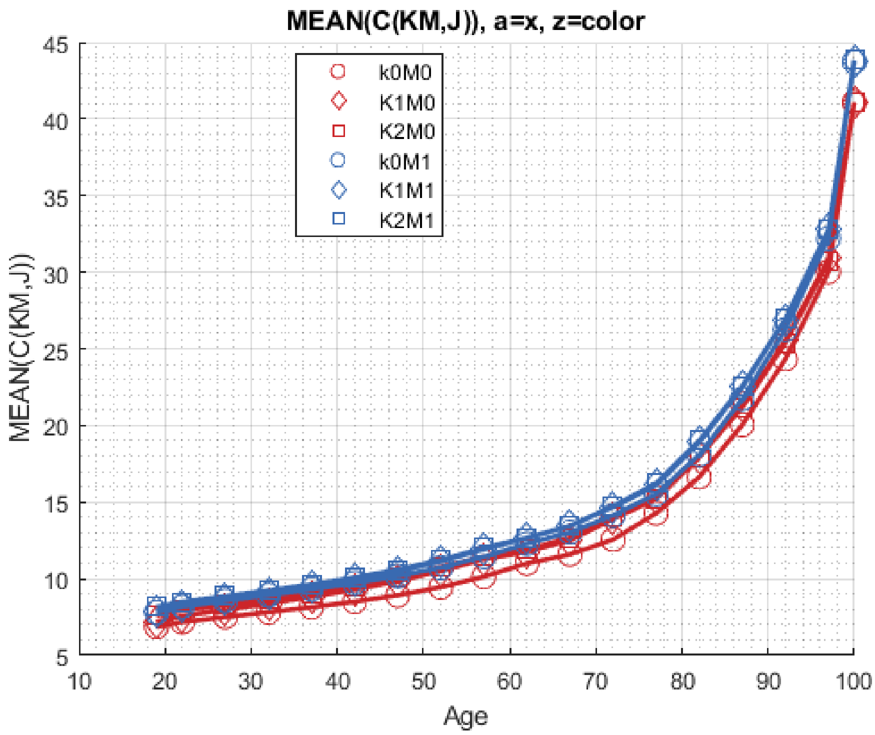
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(KM,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

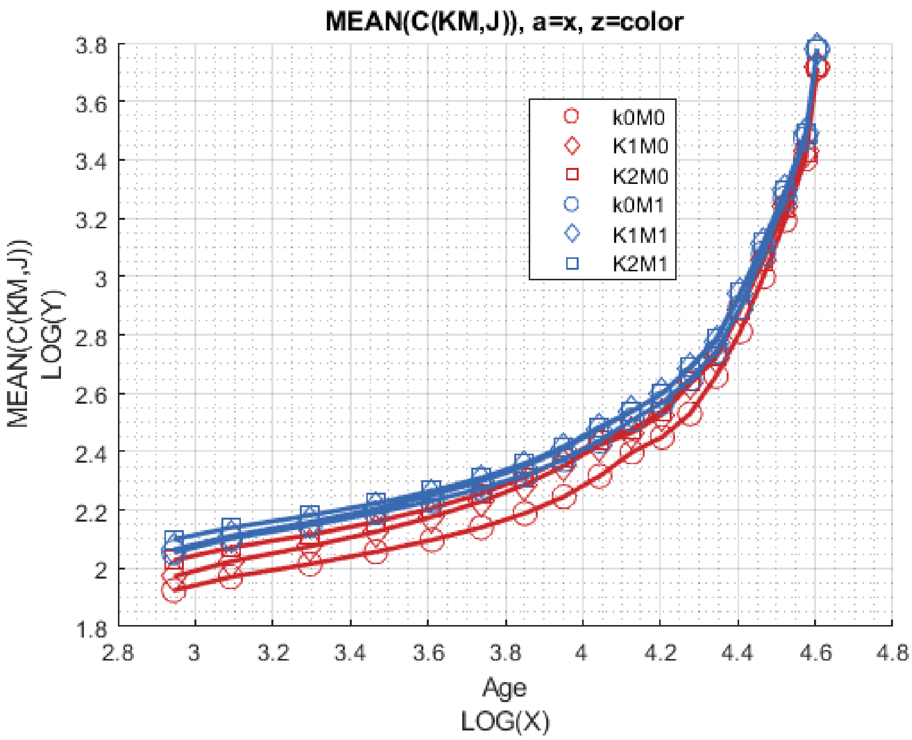




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





4.4.5 Analyze Education and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

MEAN(VAL(EKM,J)), MEAN(AP(EKM,J)), MEAN(C(EKM,J))

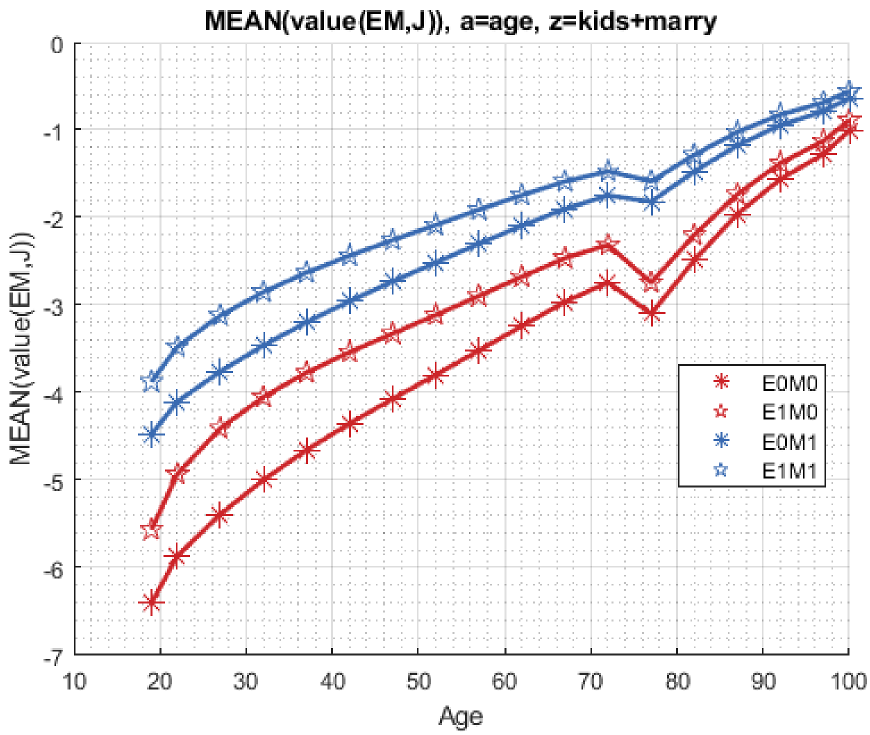
Tabulate value and policies:

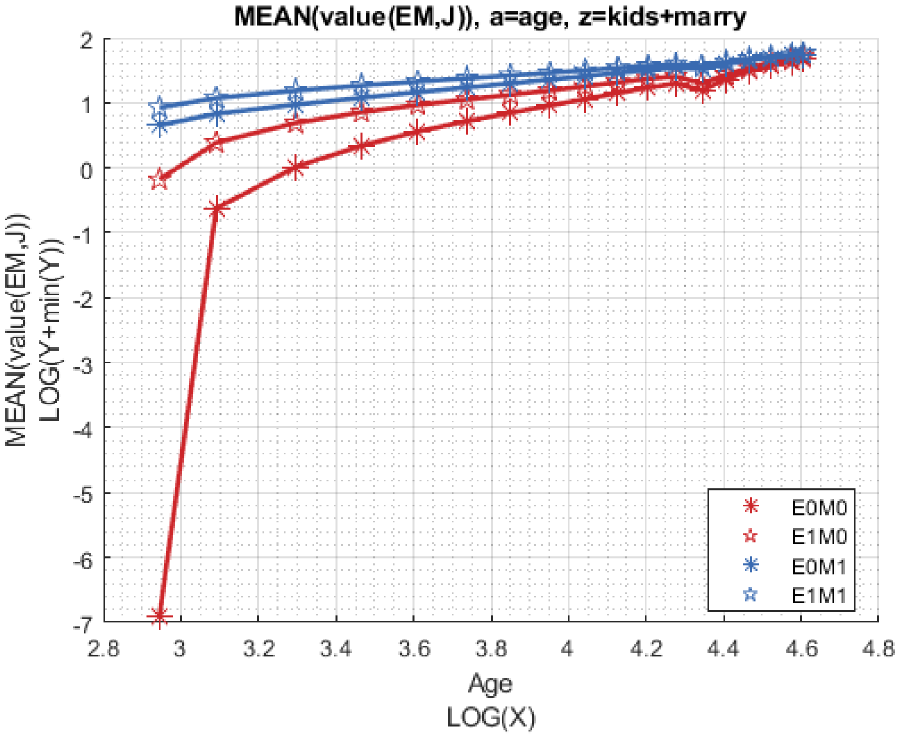
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(EKM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(VAL(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        -6.4015        -5.8666        -5.3879        -4.9966        -4.6557
      2       1       0        -5.5658        -4.9294        -4.4088        -4.0487        -3.7705
      3       0       1        -4.4764        -4.1029        -3.7581        -3.4622        -3.1968
      4       1       1        -3.8764        -3.4759        -3.1196        -2.8504        -2.6288

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(EKM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p
```

```
xxx  MEAN(AP(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        34.682         34.444         34.272         34.048         33.753
      2       1       0        34.465         34.241         34.222         34.121         33.901
      3       0       1        35.363         35.234         35.193         35.099         34.934
      4       1       1        35.293          35.21          35.37         35.437         35.374

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(EKM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(C(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        7.1022         7.4087         7.7357         8.0845         8.4713
      2       1       0        7.3195         7.7041         8.0988         8.4875         8.8852
      3       0       1        7.7587         8.0743         8.4083         8.7618         9.1489
      4       1       1         8.134         8.5446         8.9747         9.3891         9.8055
```

Graph Mean Values:

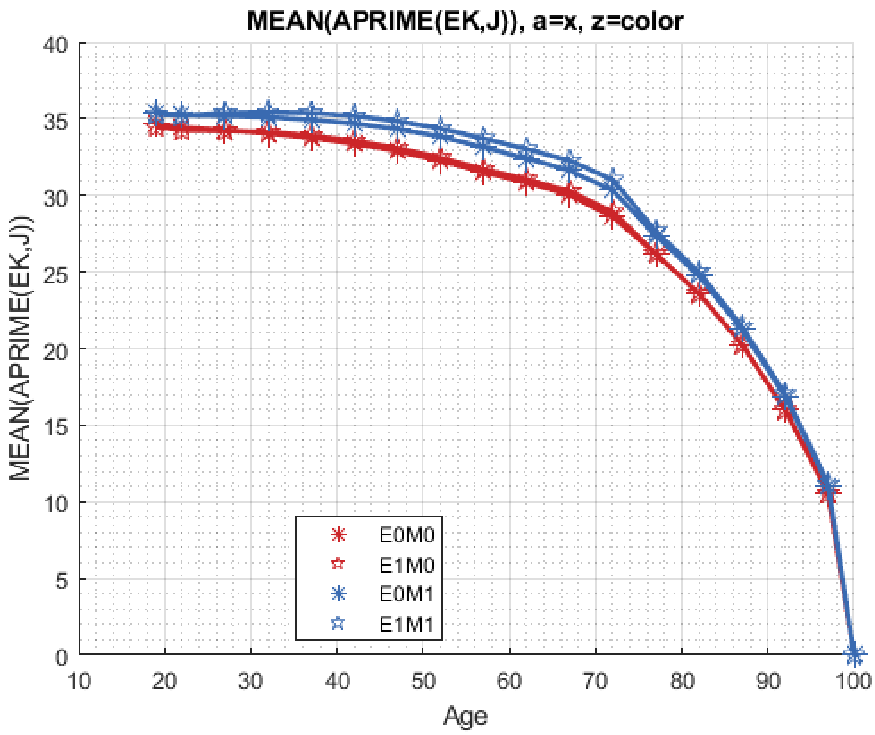
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(EM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

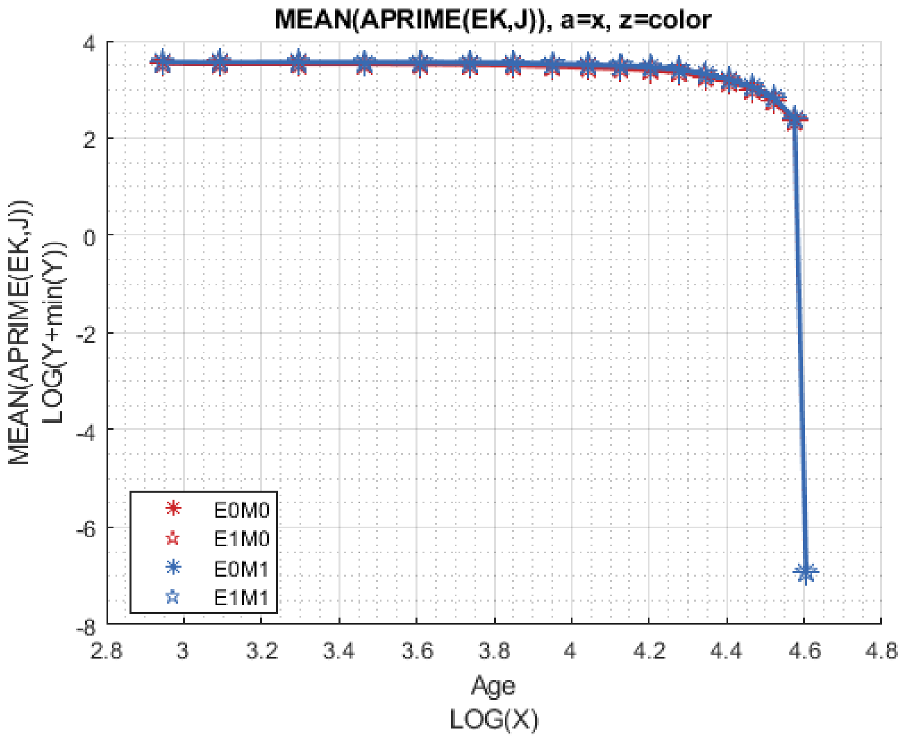




Graph Mean Savings Choices:

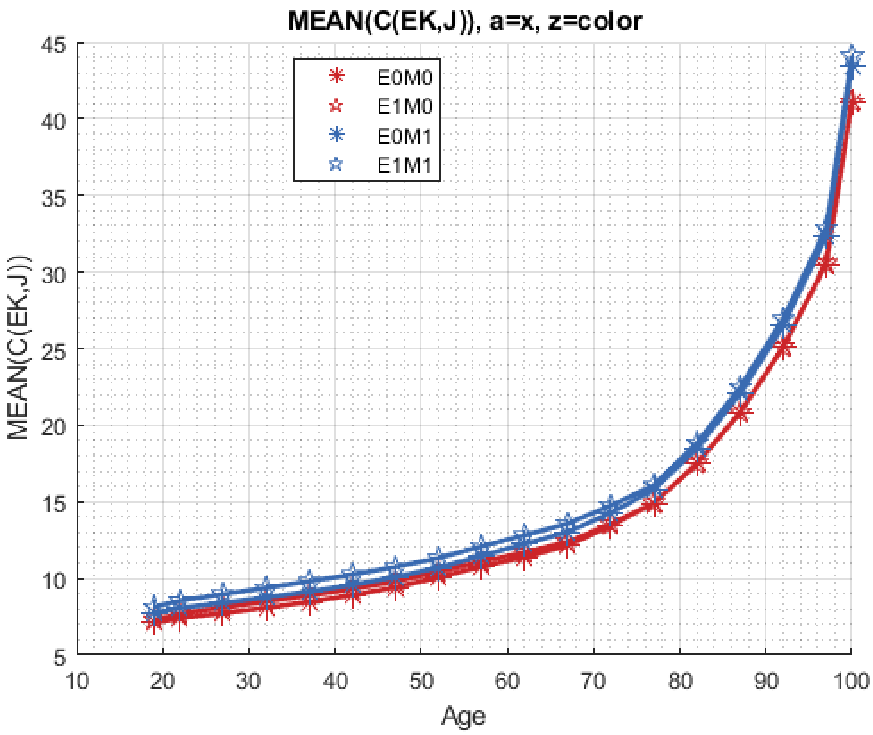
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(EK,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

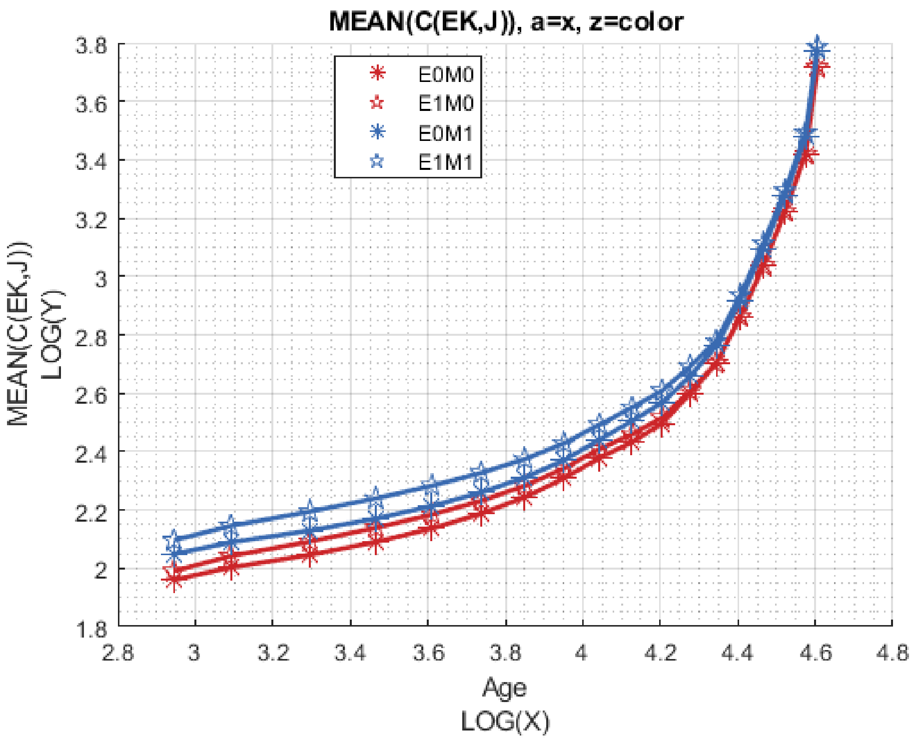




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(EK,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```




MEAN(C(EK,J)), a=x, z=color
E0M0
E1M0
E0M1
E1M1
MEAN(C(EK,J))
LOG(Y)
Age
LOG(X)
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
4.8

Chapter 5

Solution with Unemployment

5.1 Life Cycle Dynamic Programming under Unemployment Shock

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for policy function using Exact Vectorized Solution. Value in 2020 with surprise COVID unemployment Shock, with non-covid year Value as the continuation function. The file focuses on the change in value function, asset choice, and consumption choice given a one period unemployment shock (that does not reappear in the future again).

5.1.1 Test SNW_VFI_UNEMP

Solve the Regular Value and Also the Unemployment Value.

First, solve for value without unemployment issue (use the vectorized code that was previously tested):

```
mp_params = snw_mp_param('default_docdense');
mp_controls = snw_mp_control('default_test');
[V_VFI_ss,ap_VFI_ss,cons_VFI_ss,mp_valpol_more_ss] = ...
    snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:83 of 82, time-this-age:7.9376
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:82 of 82, time-this-age:6.2356
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:81 of 82, time-this-age:6.2155
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:80 of 82, time-this-age:6.3254
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:79 of 82, time-this-age:6.3074
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:78 of 82, time-this-age:6.2427
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:77 of 82, time-this-age:6.1961
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:76 of 82, time-this-age:6.4401
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:75 of 82, time-this-age:6.0576
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:74 of 82, time-this-age:6.3188
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:73 of 82, time-this-age:6.1781
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:72 of 82, time-this-age:6.3078
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:71 of 82, time-this-age:6.4338
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:70 of 82, time-this-age:6.33
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:69 of 82, time-this-age:6.4742
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:68 of 82, time-this-age:6.2434
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:67 of 82, time-this-age:6.196
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:66 of 82, time-this-age:6.3067
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:65 of 82, time-this-age:6.381
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:64 of 82, time-this-age:6.3403
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:63 of 82, time-this-age:6.4496
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:62 of 82, time-this-age:6.1753
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:61 of 82, time-this-age:6.3953
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:60 of 82, time-this-age:6.2145
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:59 of 82, time-this-age:6.3754
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:58 of 82, time-this-age:6.1826
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:57 of 82, time-this-age:6.4472
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:56 of 82, time-this-age:6.3577
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:55 of 82, time-this-age:6.3632
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:54 of 82, time-this-age:6.3623
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:53 of 82, time-this-age:6.477
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:52 of 82, time-this-age:6.4007
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:51 of 82, time-this-age:6.1299
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:50 of 82, time-this-age:6.0888
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:49 of 82, time-this-age:6.0058
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:48 of 82, time-this-age:6.4668
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:47 of 82, time-this-age:6.7856
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:46 of 82, time-this-age:6.6894
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:45 of 82, time-this-age:6.5303
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:44 of 82, time-this-age:6.5652
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:43 of 82, time-this-age:6.4601
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:42 of 82, time-this-age:6.5478
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:41 of 82, time-this-age:6.5693
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:40 of 82, time-this-age:6.4817
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:39 of 82, time-this-age:6.1035
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:38 of 82, time-this-age:6.3312
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:37 of 82, time-this-age:6.652
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:36 of 82, time-this-age:6.5706
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:35 of 82, time-this-age:6.3328
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:34 of 82, time-this-age:6.3866
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:33 of 82, time-this-age:6.3876
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:32 of 82, time-this-age:6.5786
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:31 of 82, time-this-age:6.4579
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:30 of 82, time-this-age:6.4423
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:29 of 82, time-this-age:6.5074
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:28 of 82, time-this-age:6.6582
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:27 of 82, time-this-age:6.6605
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:26 of 82, time-this-age:6.7467
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:25 of 82, time-this-age:6.567
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:24 of 82, time-this-age:6.6851
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:23 of 82, time-this-age:6.7011
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:22 of 82, time-this-age:6.3939
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:21 of 82, time-this-age:6.5634
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:20 of 82, time-this-age:6.4832
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:19 of 82, time-this-age:6.4651
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:18 of 82, time-this-age:6.5353
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:17 of 82, time-this-age:6.4967
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:16 of 82, time-this-age:6.387
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:15 of 82, time-this-age:6.345
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:14 of 82, time-this-age:6.577
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:13 of 82, time-this-age:6.7646
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:12 of 82, time-this-age:6.8183
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:11 of 82, time-this-age:6.4142
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:10 of 82, time-this-age:6.342
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:9 of 82, time-this-age:6.4692
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:8 of 82, time-this-age:6.5127
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:7 of 82, time-this-age:6.5417
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:6 of 82, time-this-age:6.5962
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:5 of 82, time-this-age:6.4304
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:4 of 82, time-this-age:6.3748
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:3 of 82, time-this-age:6.2745
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:2 of 82, time-this-age:6.5175
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:1 of 82, time-this-age:6.3803
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=535.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN          sum         mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -8.6673e+08    -19.834    28.17
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.4164e+09     32.412      36.
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05      2.131e+08     4.8764    8.326

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    --------

    r1     -376.05    -375.66    -373.17     -367.4    -358.05        -6.68      -6.5297      -6.379
    r2      -363.8    -363.41    -360.93    -355.25    -346.25      -6.4892      -6.3437      -6.197
    r3     -351.75    -351.36     -348.9    -343.44     -334.9      -6.2948      -6.1538      -6.011
    r4     -339.81    -339.45    -337.16    -332.06    -324.04       -6.095      -5.9584        -5.8
    r5     -328.99    -328.65    -326.51    -321.72    -314.17      -5.9054      -5.7725      -5.637
    r79    -14.033     -14.02    -13.926    -13.689    -13.287     -0.22848     -0.21775     -0.2076
    r80    -12.564     -12.55    -12.457     -12.22    -11.818     -0.17427     -0.16611     -0.1584
    r81    -10.778    -10.764    -10.671    -10.434    -10.032     -0.11927     -0.11368     -0.1084
    r82    -8.4226    -8.4089    -8.3155    -8.0786    -7.6766     -0.06597     -0.06284    -0.05992
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.020968    -0.019972    -0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2        c3            c4            c5         c526496    c526497    c526498    c52
           --    --    ----------    ---------    ----------    -------    -------    -------    ---

    r1     0     0      0.0005656    0.0075134      0.022901     114.76     120.42     126.29     132
    r2     0     0     0.00051498    0.0065334      0.021549     114.87     120.54     126.42     132
    r3     0     0     0.00051498    0.0049294      0.019875     114.98     120.67     126.57     132
    r4     0     0     0.00051498    0.0047937      0.019672     115.74     121.44     127.36     133
    r5     0     0     0.00048517    0.0046683      0.019484     116.51     122.22     128.16     134
    r79    0     0              0            0    0.00051498     81.091      85.68     90.325     94.
    r80    0     0              0            0             0     76.669      80.55     84.292     88.
    r81    0     0              0            0             0     68.313      71.52     74.459     77.
    r82    0     0              0            0             0     50.126     53.467     56.953     58.
    r83    0     0              0            0             0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5        c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.036717    0.037251    0.040426     0.04363    0.048012     9.6396     9.8066     9.9533
    r2     0.036717    0.037251    0.040477     0.04461    0.049364     9.8014     9.9571     10.088
    r3     0.036717    0.037251    0.040477    0.046214    0.051039     9.9664     10.108      10.22
    r4     0.038144    0.038678    0.041903    0.047776    0.052666     10.118     10.244     10.339
    r5     0.039534    0.040068    0.043323     0.04929    0.054241     10.258     10.369     10.446
    r79     0.19737     0.19791     0.20163     0.21175     0.23093     35.811     37.046     38.418
    r80     0.19737     0.19791     0.20163     0.21175     0.23145     40.207      42.15     44.426
    r81     0.19737     0.19791     0.20163     0.21175     0.23145     48.541     51.158     54.236
    r82     0.19737     0.19791     0.20163     0.21175     0.23145      66.71     69.193     71.724
```

```
r83     0.19737     0.19791     0.20163     0.21175     0.23145    116.82     122.65     128.66
```

Second, solve for the unemployment value, use the exact-bisec result code, call the snw_vfi_main_bisec_vec.m function with a third input of existing value. xi is the share of income lost during covid year given surprise covid shock, b is the share of income loss that is covered by unemployment insurance. xi=0.5 and b=0 means will lose 50 percent of income given COVID shocks, and the loss will not be covered at all by unemployment insurance.

```
mp_params('xi') = 0.5;
mp_params('b') = 0;
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
[V_VFI_unemp,ap_VFI_unemp,cons_VFI_unemp,mp_valpol_more_unemp] = ...
    snw_vfi_main_bisec_vec(mp_params, mp_controls, V_VFI_ss);
```

```
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 1 of 82, time-this-age:6.2923
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 2 of 82, time-this-age:6.5203
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 3 of 82, time-this-age:6.0245
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 4 of 82, time-this-age:6.5906
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 5 of 82, time-this-age:6.4748
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 6 of 82, time-this-age:6.515
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 7 of 82, time-this-age:6.1144
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 8 of 82, time-this-age:6.2132
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 9 of 82, time-this-age:6.5055
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 10 of 82, time-this-age:6.4562
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 11 of 82, time-this-age:6.0604
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 12 of 82, time-this-age:6.641
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 13 of 82, time-this-age:6.5864
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 14 of 82, time-this-age:6.3514
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 15 of 82, time-this-age:6.3017
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 16 of 82, time-this-age:6.2902
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 17 of 82, time-this-age:6.5771
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 18 of 82, time-this-age:6.163
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 19 of 82, time-this-age:6.2954
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 20 of 82, time-this-age:6.162
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 21 of 82, time-this-age:6.0779
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 22 of 82, time-this-age:6.1283
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 23 of 82, time-this-age:6.3978
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 24 of 82, time-this-age:6.4584
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 25 of 82, time-this-age:5.9292
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 26 of 82, time-this-age:6.1225
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 27 of 82, time-this-age:6.4045
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 28 of 82, time-this-age:6.4834
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 29 of 82, time-this-age:6.2843
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 30 of 82, time-this-age:6.2094
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 31 of 82, time-this-age:6.4612
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 32 of 82, time-this-age:6.1629
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 33 of 82, time-this-age:6.4799
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 34 of 82, time-this-age:6.4272
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 35 of 82, time-this-age:6.5508
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 36 of 82, time-this-age:6.1381
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 37 of 82, time-this-age:6.0588
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 38 of 82, time-this-age:6.1133
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 39 of 82, time-this-age:5.9731
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 40 of 82, time-this-age:6.2753
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 41 of 82, time-this-age:6.4228
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 42 of 82, time-this-age:6.0784
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 43 of 82, time-this-age:5.8926
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 44 of 82, time-this-age:6.1351
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 45 of 82, time-this-age:5.9147
```

```
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 46 of 82, time-this-age:6.4062
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 47 of 82, time-this-age:6.7344
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 48 of 82, time-this-age:6.278
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 49 of 82, time-this-age:6.2085
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 50 of 82, time-this-age:6.0966
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 51 of 82, time-this-age:6.3449
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 52 of 82, time-this-age:6.2717
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 53 of 82, time-this-age:6.4826
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 54 of 82, time-this-age:6.3431
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 55 of 82, time-this-age:5.9453
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 56 of 82, time-this-age:6.2613
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 57 of 82, time-this-age:6.2364
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 58 of 82, time-this-age:6.0578
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 59 of 82, time-this-age:6.1279
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 60 of 82, time-this-age:6.1238
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 61 of 82, time-this-age:6.2806
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 62 of 82, time-this-age:5.9344
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 63 of 82, time-this-age:6.0212
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 64 of 82, time-this-age:6.3424
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 65 of 82, time-this-age:5.863
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 66 of 82, time-this-age:6.1349
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 67 of 82, time-this-age:5.9904
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 68 of 82, time-this-age:5.9675
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 69 of 82, time-this-age:5.7964
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 70 of 82, time-this-age:6.0091
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 71 of 82, time-this-age:6.0331
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 72 of 82, time-this-age:5.8808
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 73 of 82, time-this-age:6.0943
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 74 of 82, time-this-age:6.0279
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 75 of 82, time-this-age:6.1533
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 76 of 82, time-this-age:6.0662
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 77 of 82, time-this-age:5.7827
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 78 of 82, time-this-age:6.1473
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 79 of 82, time-this-age:6.071
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 80 of 82, time-this-age:6.1187
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 81 of 82, time-this-age:5.9865
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 82 of 82, time-this-age:6.1363
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 83 of 82, time-this-age:7.7853
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN         sum         mean       std
                -    ---    ----    --------    ----    ---------   -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05   -8.9196e+08    -20.411    29.20
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05    1.3793e+09     31.564    36.67
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05    2.1007e+08      4.807    8.326

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
           c1         c2         c3         c4         c5        c526496      c526497      c526498
         -------    -------    -------    -------    -------    ---------    ---------    --------

    r1   -402.51    -401.01    -392.48    -379.06     -366.5      -6.8096      -6.6548      -6.500
    r2   -390.26    -388.76    -380.23    -366.81     -354.4       -6.618      -6.4683      -6.318
    r3   -378.21    -376.71    -368.18    -354.76    -342.64      -6.4227       -6.278      -6.132
    r4   -365.26    -363.87    -355.91    -343.19    -331.62      -6.2297      -6.0896      -5.948
```

```
    r5     -353.53    -352.23    -344.77    -332.67    -321.59       -6.0467       -5.9107       -5.773
    r79    -14.033     -14.02    -13.926    -13.689    -13.287       -0.2305      -0.21962      -0.2093
    r80    -12.564     -12.55    -12.457     -12.22    -11.818      -0.17582      -0.16751      -0.1596
    r81    -10.778    -10.764    -10.671    -10.434    -10.032      -0.12032      -0.11462      -0.1092
    r82    -8.4226    -8.4089    -8.3155    -8.0786    -7.6766     -0.066524     -0.063355     -0.06039
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206     -0.021146     -0.020134     -0.01918

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1     c2     c3     c4         c5         c526496    c526497    c526498    c526499    c526500
           --     --     --     --     ----------     -------    -------    -------    -------    -------

    r1     0      0      0      0       0.0092181     110.07     115.71     121.57     127.63     133.94
    r2     0      0      0      0        0.008238     110.04     115.69     121.55     127.63     133.96
    r3     0      0      0      0       0.0066341        110     115.66     121.54     127.64     133.99
    r4     0      0      0      0       0.0058019     110.29     115.97     121.86     127.98     134.34
    r5     0      0      0      0        0.004998     110.59     116.28     122.19     128.33      134.7
    r79    0      0      0      0      0.00051498     81.091     85.231     89.301     93.347     97.387
    r80    0      0      0      0               0     75.865     79.539     83.281     87.018     90.672
    r81    0      0      0      0               0      67.78      70.52     73.459     76.816     81.091
    r82    0      0      0      0               0     50.126     53.467     56.104     57.737     60.587
    r83    0      0      0      0               0          0          0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5         c526496    c526497    c526498
           --------    --------    --------    --------    --------     -------    -------    -------

    r1     0.018623    0.019158    0.022901    0.033062     0.04363     9.4621     9.6396     9.8066
    r2     0.018623    0.019158    0.022901    0.033062     0.04461     9.6318     9.8014     9.9571
    r3     0.018623    0.019158    0.022901    0.033062    0.046214     9.8074     9.9664     10.108
    r4     0.019354    0.019888    0.023632    0.033792    0.047776      9.971     10.118     10.244
    r5     0.020066    0.020601    0.024344    0.034504     0.04929     10.123     10.258     10.369
    r79     0.19737     0.19791     0.20163     0.21175     0.23093     34.787     36.471     38.418
    r80     0.19737     0.19791     0.20163     0.21175     0.23145     40.001      42.15     44.426
    r81     0.19737     0.19791     0.20163     0.21175     0.23145     48.074     51.158     54.236
    r82     0.19737     0.19791     0.20163     0.21175     0.23145     65.719     68.202     71.583
    r83     0.19737     0.19791     0.20163     0.21175     0.23145     115.84     121.66     127.68
```

Difference Between Value and Choices In Unemployment and Future Periods

```
V_VFI_unemp_drop = V_VFI_ss - V_VFI_unemp;
ap_VFI_unemp_drop = ap_VFI_ss - ap_VFI_unemp;
cons_VFI_unemp_drop = cons_VFI_ss - cons_VFI_unemp;
```

5.1.2 Define Parameter Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
```

```
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

5.1.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 15; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

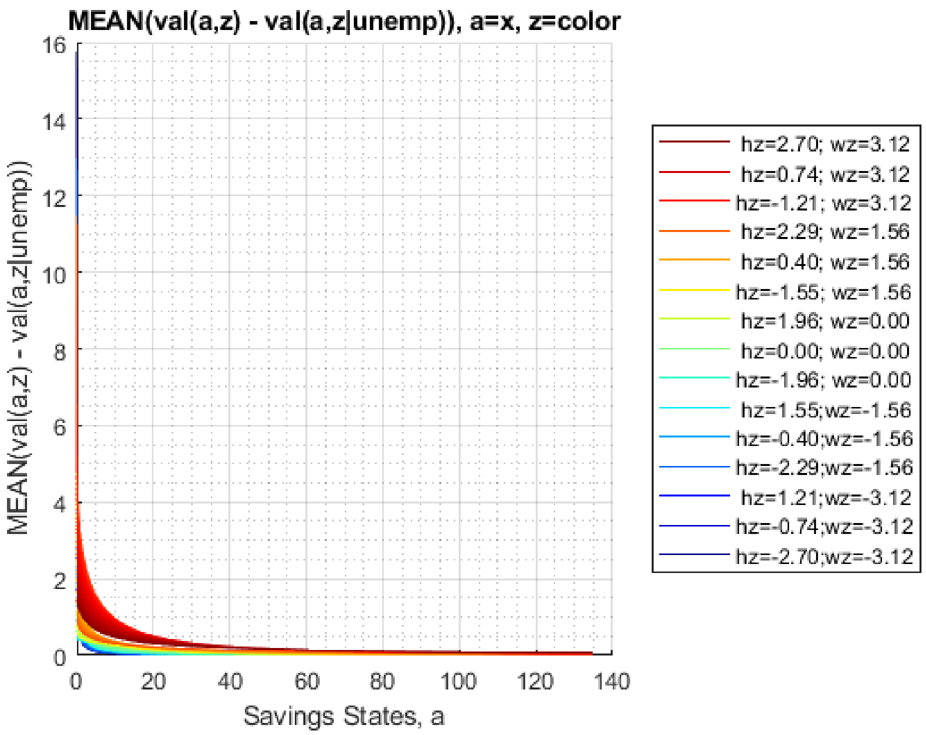
MEAN(VAL(A,Z) - VAL(A,Z|unemp)), MEAN(AP(A,Z) - AP(A,Z|unemp)), MEAN(C(A,Z) - C(A,Z|unemp))

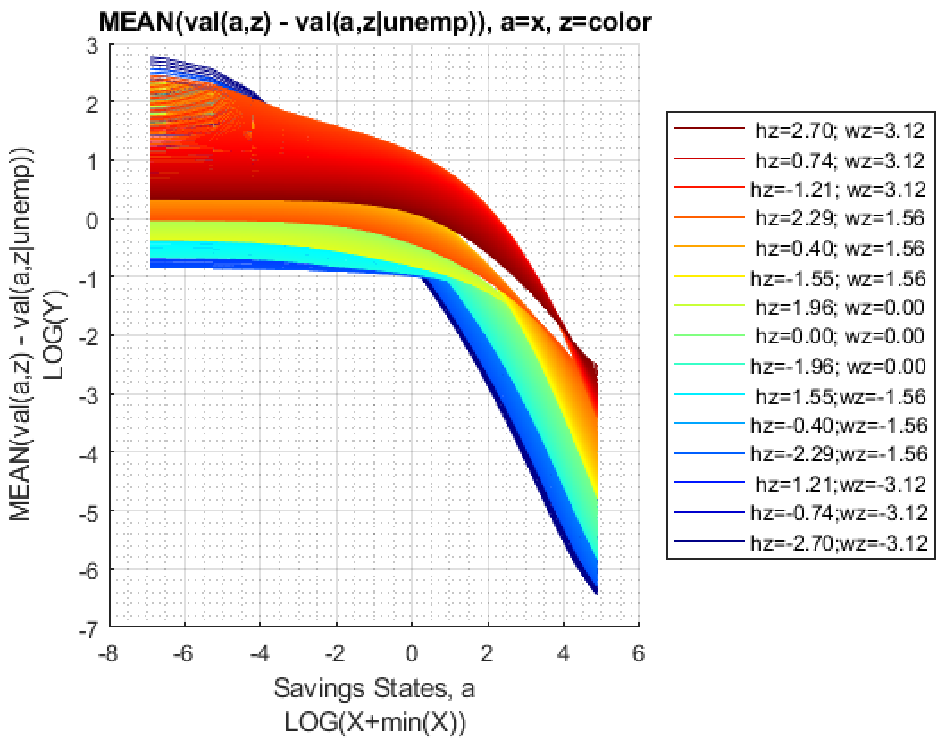
Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(v(A,Z) - v(A,Z|unemp))", V_VFI_unemp_drop, true, ["mean"], 4, 1, cl

xxx  MEAN(v(A,Z) - v(A,Z|unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0       15.757         14.81        13.917        13.077        12.285
```

1	0	0.019312	0.020444	0.021649	0.02293	0.024294	0

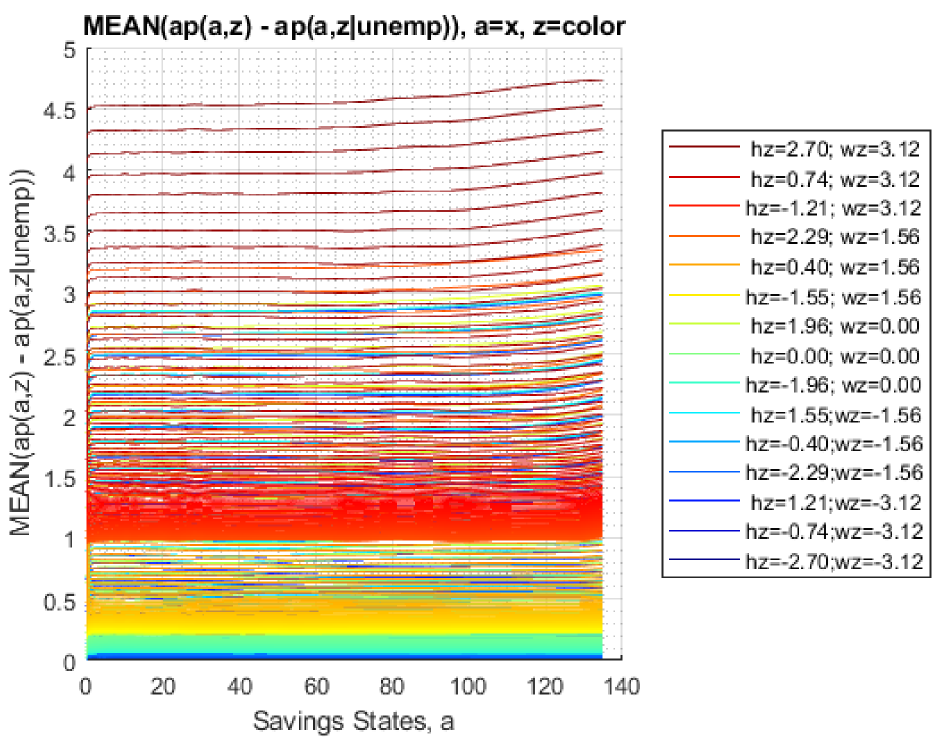
```
mp_support_graph('cl_st_graph_title') = {'MEAN(val(a,z) - val(a,z|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(val(a,z) - val(a,z|unemp))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

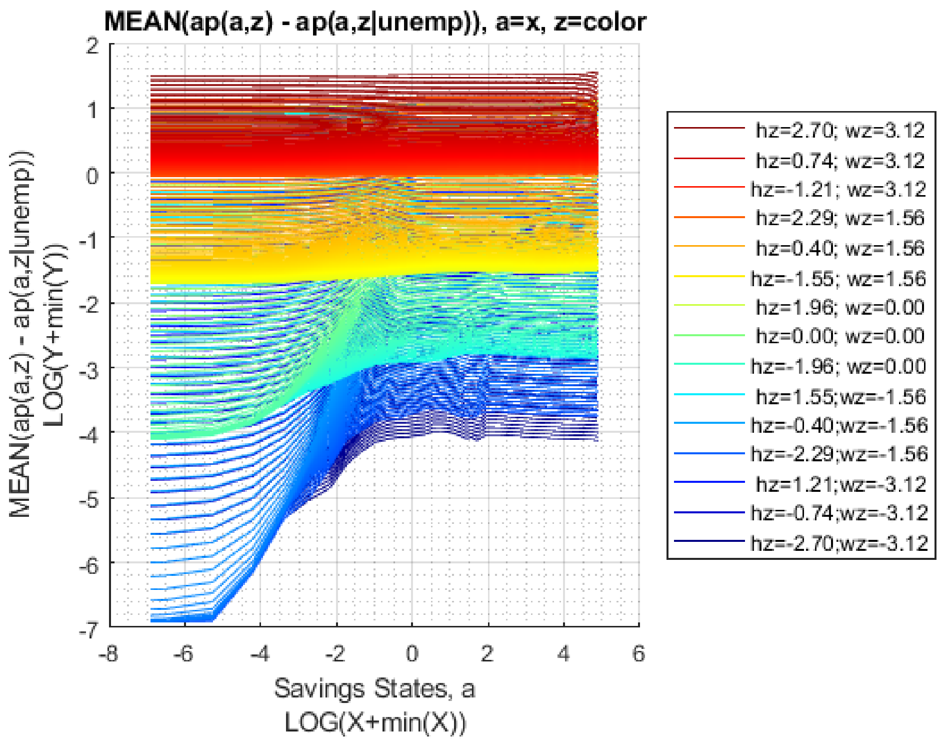




Graph Mean Savings Choices Change:

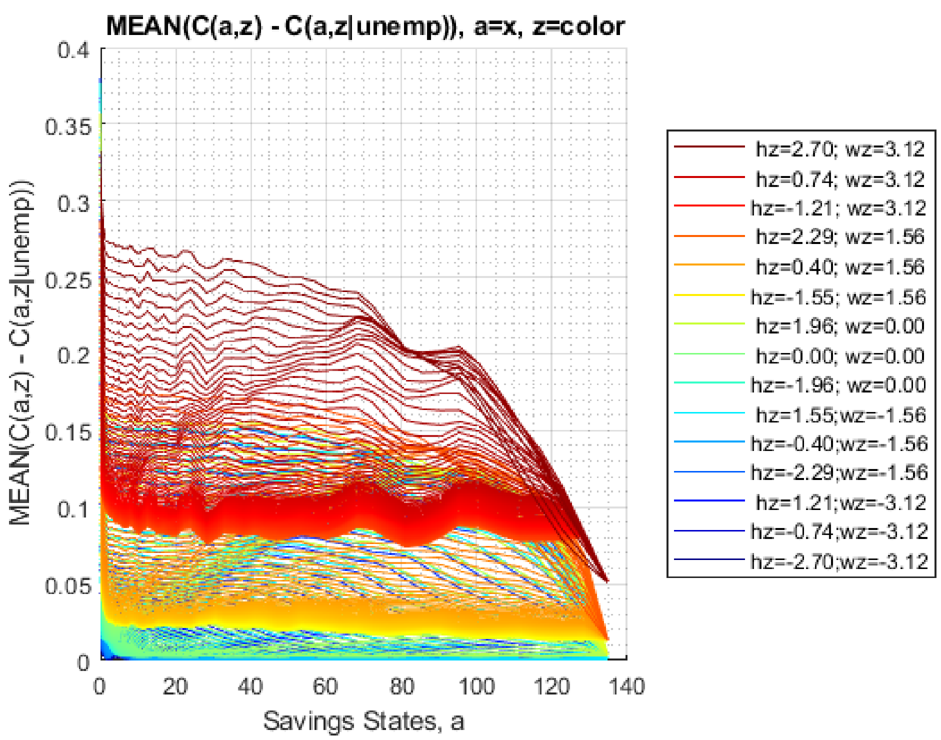
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(a,z) - ap(a,z|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(a,z) - ap(a,z|unemp))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

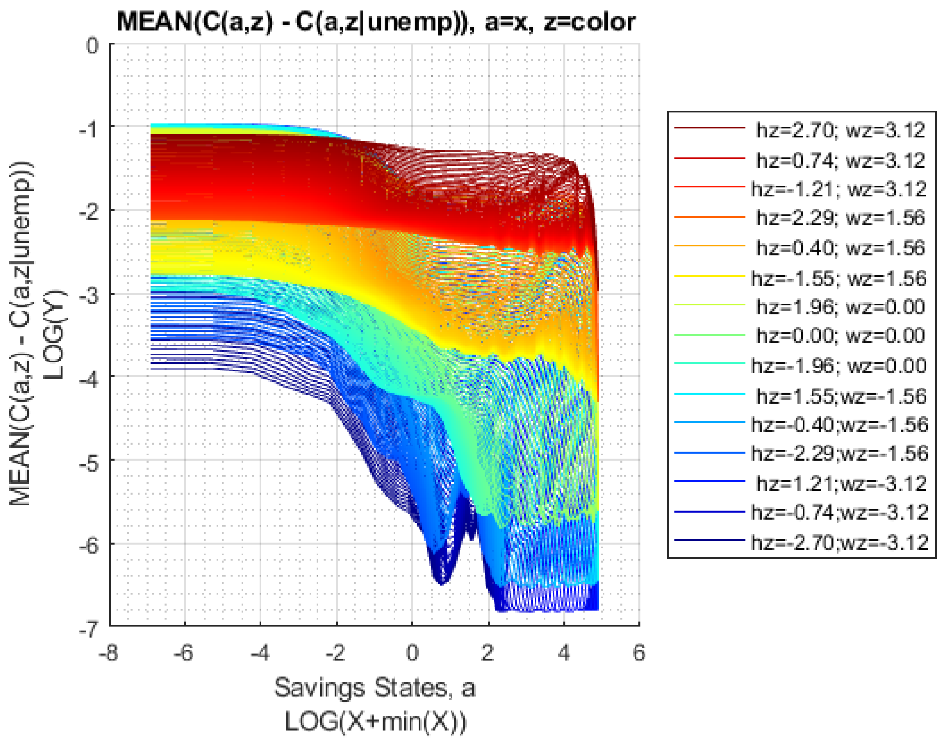




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z) - C(a,z|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z) - C(a,z|unemp))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





5.1.4 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(V(KM,J) - V(KM,J | unemp)), MEAN(ap(KM,J) - ap(KM,J | unemp)), MEAN(c(KM,J) - c(KM,J | unemp))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(V(KM,J) - V(KM,J | unemp))", V_VFI_unemp_drop, true, ["mean"], 3, 1

xxx  MEAN(V(KM,J) - V(KM,J | unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     kids     marry     mean_age_18     mean_age_19     mean_age_20     mean_age_21     mean_age_2
    _____     ____     _____     ___________     ___________     ___________     ___________     __________

      1         1        0         0.61725         0.59976         0.58199         0.56601         0.55229
```

```
     2       2       0          0.8282        0.80576        0.78226        0.75802        0.73679
     3       3       0         0.96839        0.94588        0.92133        0.89232        0.86693
     4       4       0          1.0956         1.0721         1.0458         1.0127        0.98375
     5       5       0          1.2019         1.1787         1.1519         1.1158         1.0843
     6       1       1         0.76869        0.75018        0.73196        0.71662        0.70371
     7       2       1         0.93097        0.90782        0.88415        0.86307        0.84465
     8       3       1          1.0192        0.99485        0.96959        0.94589        0.92513
     9       4       1          1.1177         1.0921         1.0652          1.039          1.016
    10       5       1          1.1589         1.1352         1.1089         1.0814         1.0577
```

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(ap(KM,J) - ap(KM,J | unemp))", ap_VFI_unemp_drop, true, ["mean"],
```

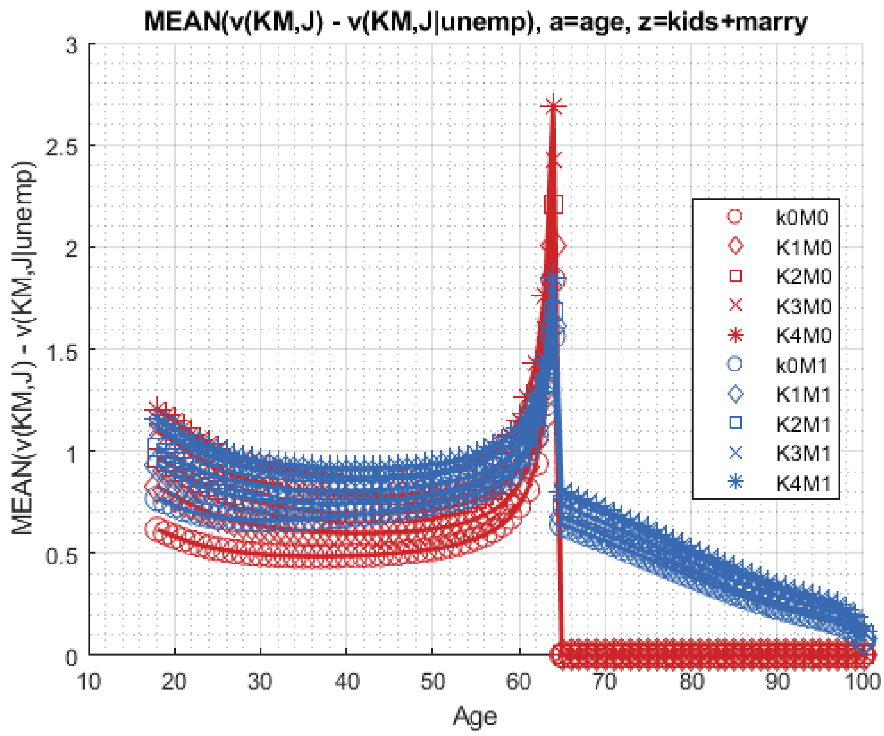
```
xxx  MEAN(ap(KM,J) - ap(KM,J | unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1       1       0         0.54442        0.54173        0.53857        0.57709         0.6155
      2       2       0         0.53845        0.53471        0.53034        0.56817        0.60591
      3       3       0         0.53193        0.52758         0.5228        0.56021        0.59766
      4       4       0         0.52782        0.52325        0.51825        0.55544        0.59269
      5       5       0         0.52378        0.51921         0.5141        0.55117        0.58841
      6       1       1          1.1324         1.1758           1.22         1.3121         1.4051
      7       2       1          1.0397         1.0754         1.1117         1.1944         1.2779
      8       3       1         0.97116         1.0022         1.0333         1.1099         1.1873
      9       4       1         0.89614        0.92283        0.94936         1.0215          1.094
     10       5       1         0.78037        0.79821        0.81602        0.87841        0.94111
```

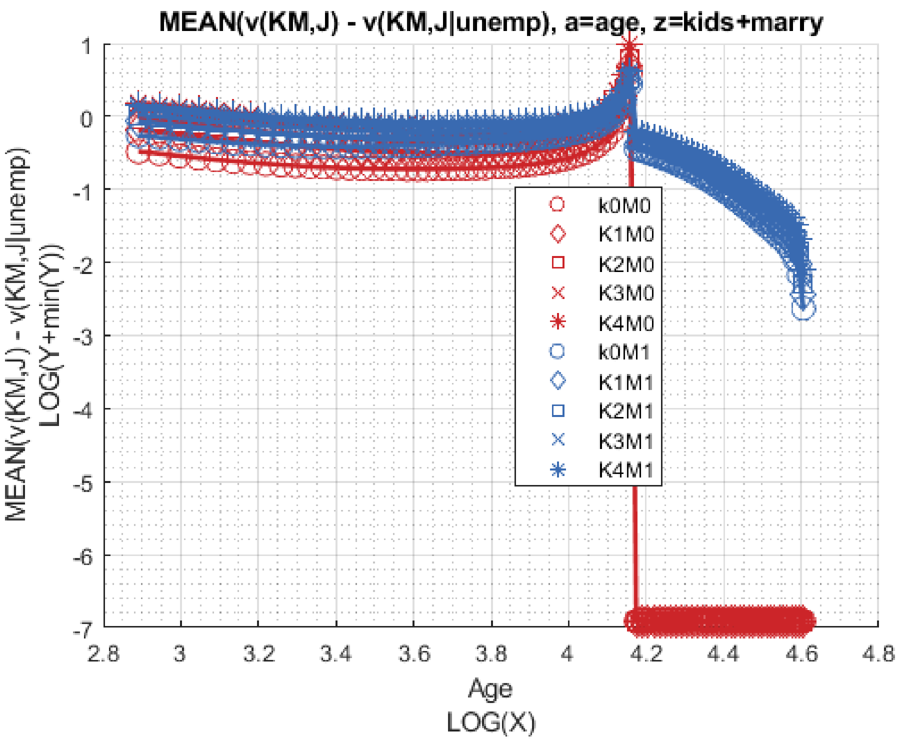
```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(c(KM,J) - c(KM,J | unemp))", cons_VFI_unemp_drop, true, ["mean"], 3
```

```
xxx  MEAN(c(KM,J) - c(KM,J | unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1       1       0        0.049956       0.052643       0.055802       0.056135        0.05627
      2       2       0         0.05592       0.059662       0.064032       0.065053       0.065864
      3       3       0        0.062441       0.066796       0.071572       0.073013       0.074107
      4       4       0        0.066548       0.071126       0.076127       0.077784        0.07908
      5       5       0        0.070592       0.075167       0.080271       0.082051       0.083363
      6       1       1        0.091533        0.09707        0.10273        0.10671        0.11016
      7       2       1        0.087319       0.093037       0.098883        0.10344        0.10744
      8       3       1        0.089145        0.09428        0.10003        0.10457         0.1086
      9       4       1        0.095251       0.099402        0.10424        0.10703         0.1095
     10       5       1          0.1016        0.10608        0.11098        0.11351        0.11573
```

Graph Mean Values Change:

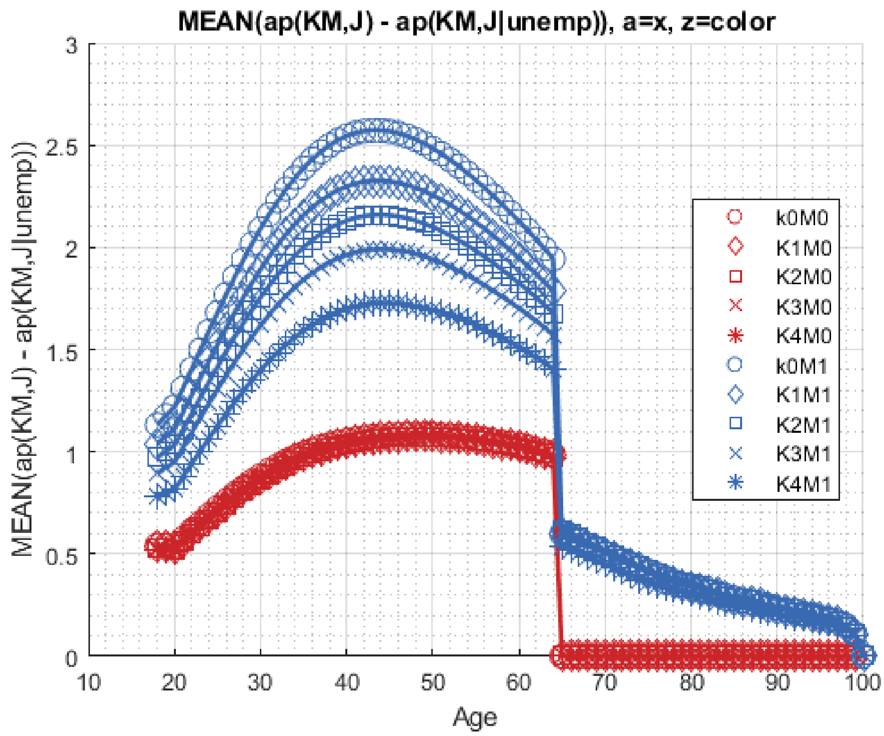
```
mp_support_graph('cl_st_graph_title') = {'MEAN(v(KM,J) - v(KM,J|unemp), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(v(KM,J) - v(KM,J|unemp)'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

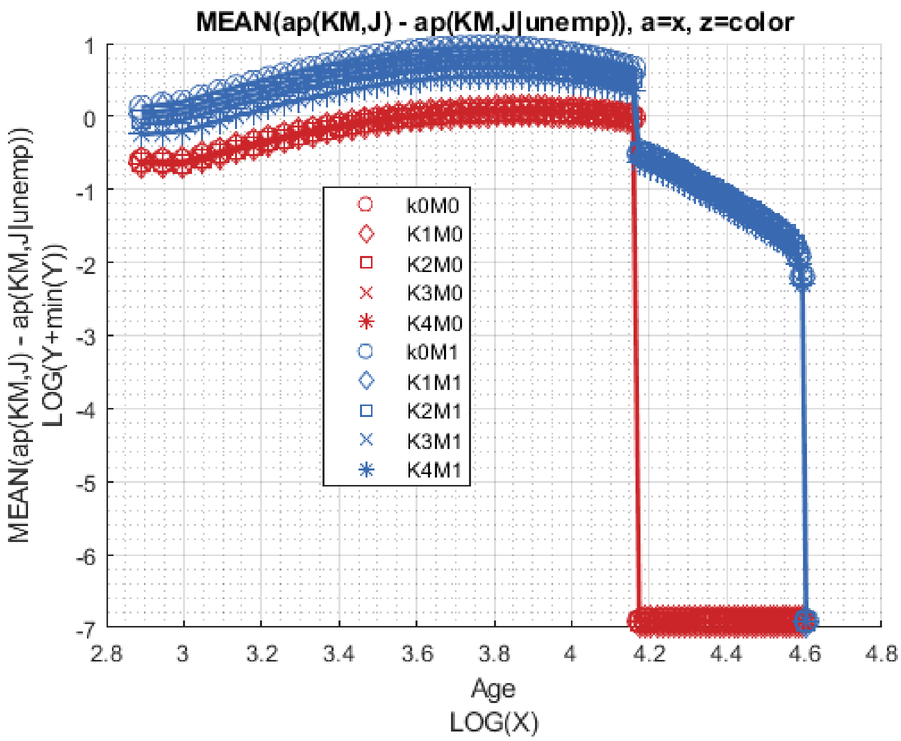




Graph Mean Savings Choices Change:

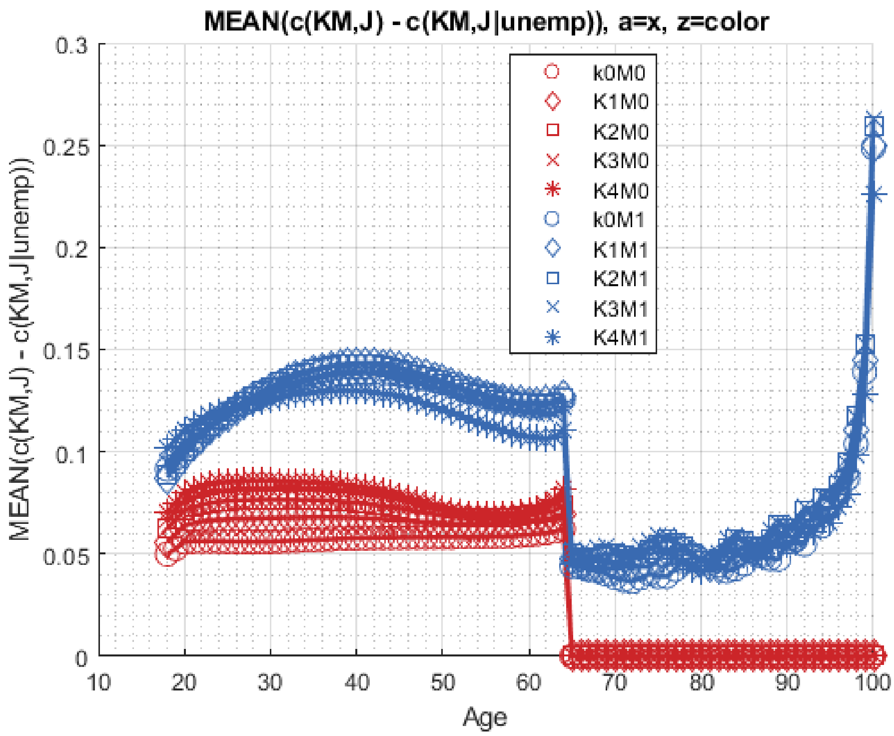
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(KM,J) - ap(KM,J|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(KM,J) - ap(KM,J|unemp))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

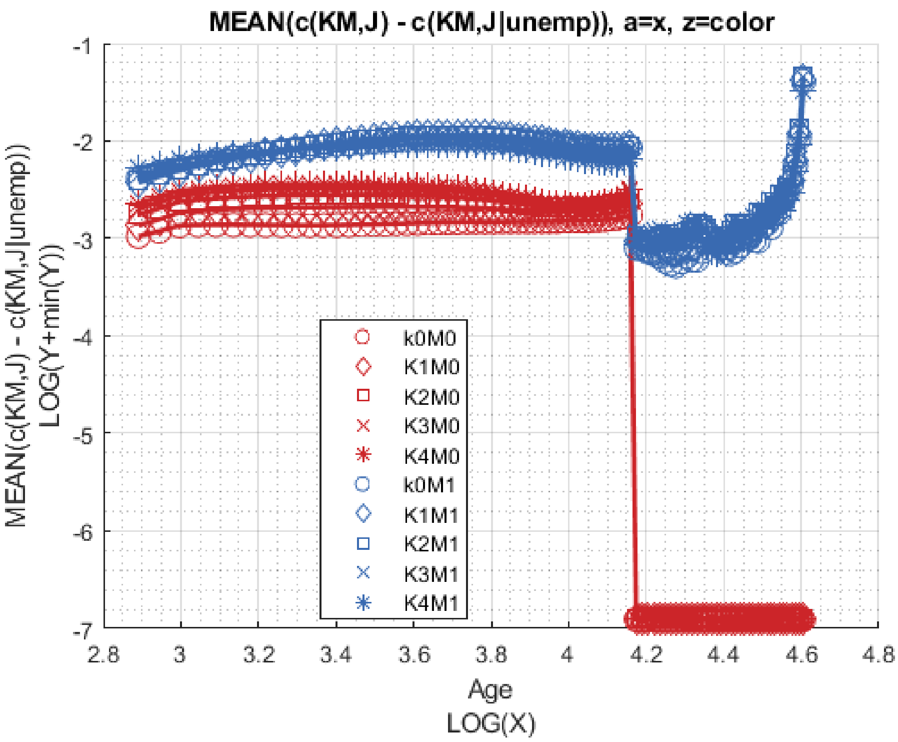




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(c(KM,J) - c(KM,J|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(c(KM,J) - c(KM,J|unemp))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





5.1.5 Analyze Education and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

```
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

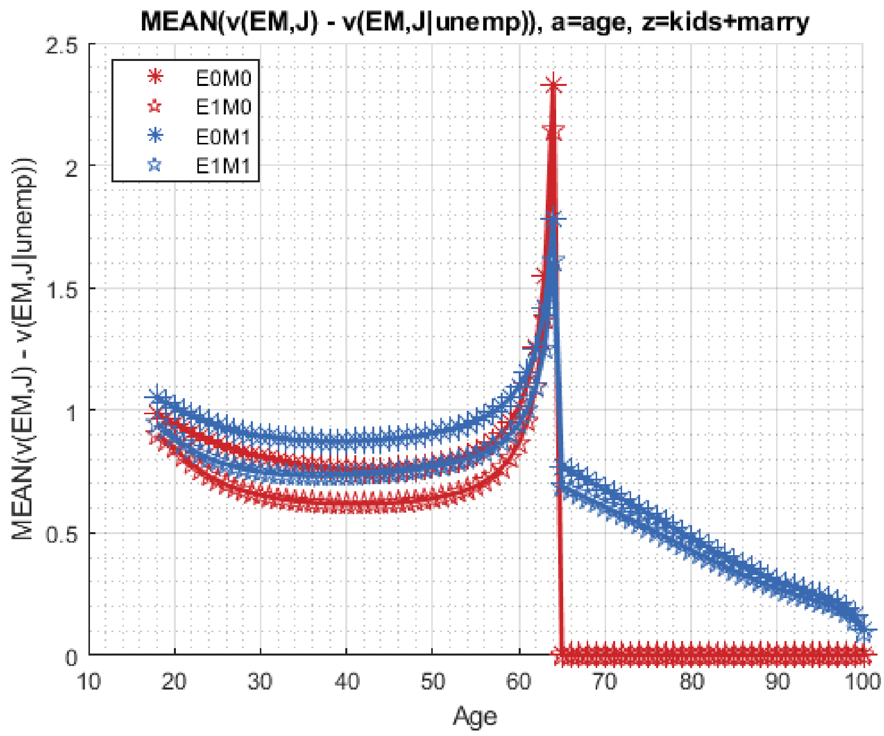
MEAN(v(EKM,J) - v(EKM,J|unemp)), MEAN(ap(EM,J) - ap(EM,J|unemp)), MEAN(c(EM,J) - c(EM,J|unemp))

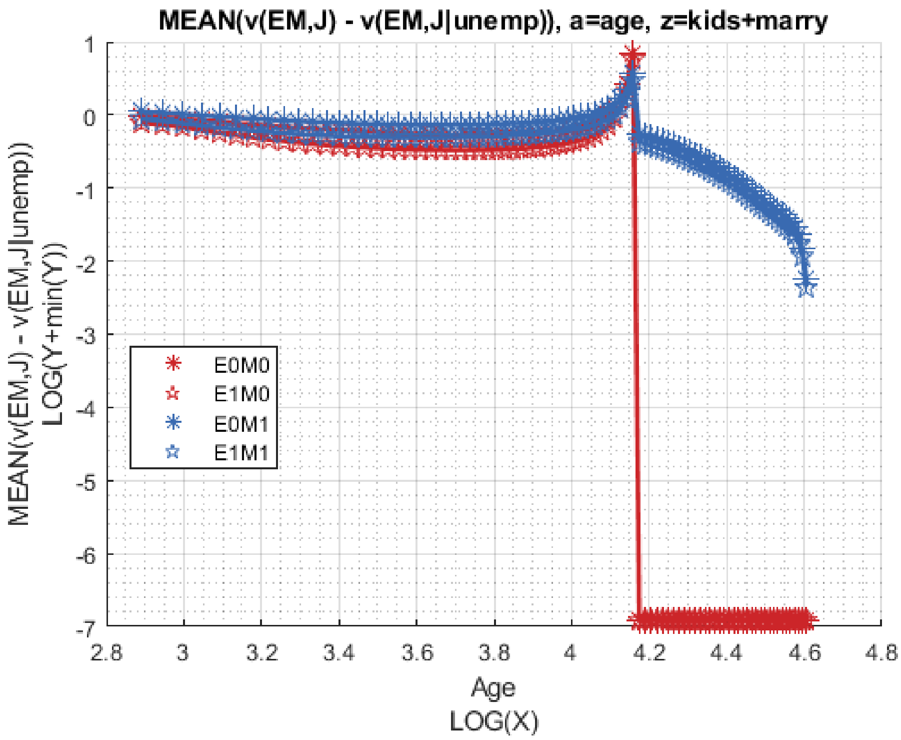
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(v(EM,J) - v(EM,J|unemp))", V_VFI_unemp_drop, true, ["mean"], 3, 1,

xxx  MEAN(v(EM,J) - v(EM,J|unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        0.98417        0.96522        0.94504        0.92586        0.90827
      2       1       0        0.90037        0.87569        0.84826         0.8121        0.78137
      3       0       1         1.0511         1.0316         1.0114        0.99333        0.97709
      4       1       1         0.9471        0.92052        0.89256        0.86506        0.84177

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(ap(EM,J) - ap(EM,J|unemp))", ap_VFI_unemp_drop, true, ["mean"], 3,

xxx  MEAN(ap(EM,J) - ap(EM,J|unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        0.54413        0.54211        0.53973        0.56238        0.58448
      2       1       0        0.52244        0.51648        0.50989        0.56245        0.61559
      3       0       1        0.93049        0.95921         0.9882         1.0448         1.1013
      4       1       1        0.99745         1.0306         1.0639         1.1617         1.2608

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(c(EM,J) - c(EM,J|unemp))", cons_VFI_unemp_drop, true, ["mean"], 3,

xxx  MEAN(c(EM,J) - c(EM,J|unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0       0.050247       0.052266       0.054642       0.055442        0.05623
      2       1       0       0.071936       0.077892        0.08448       0.086172       0.087244
      3       0       1       0.079086       0.082612       0.086438       0.089119       0.091706
      4       1       1        0.10685        0.11333        0.12031        0.12499        0.12887
```

Graph Mean Values Change:

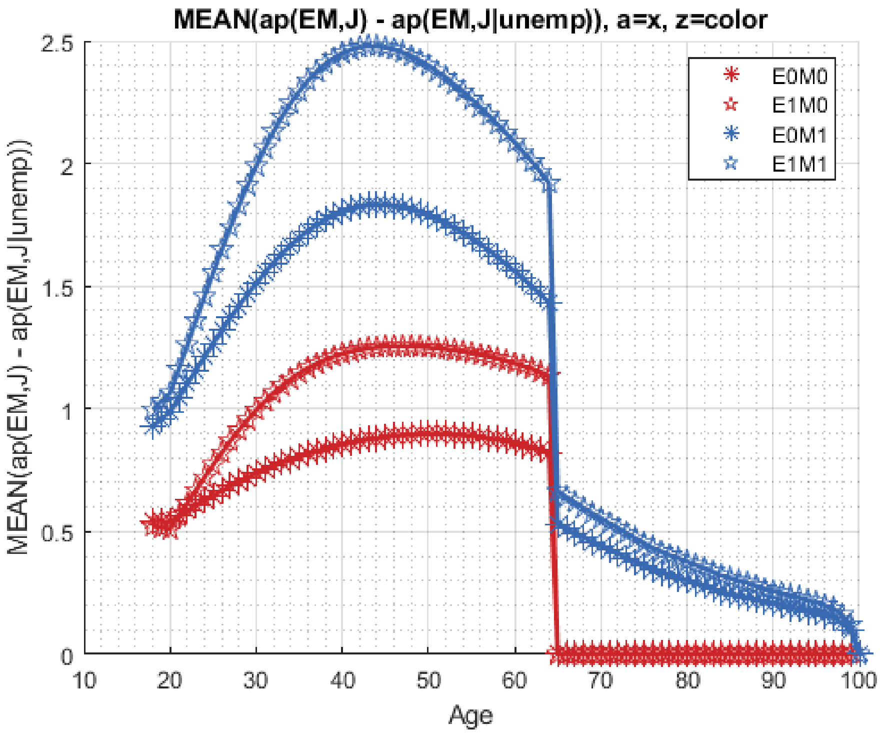
```
mp_support_graph('cl_st_graph_title') = {'MEAN(v(EM,J) - v(EM,J|unemp)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(v(EM,J) - v(EM,J|unemp))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

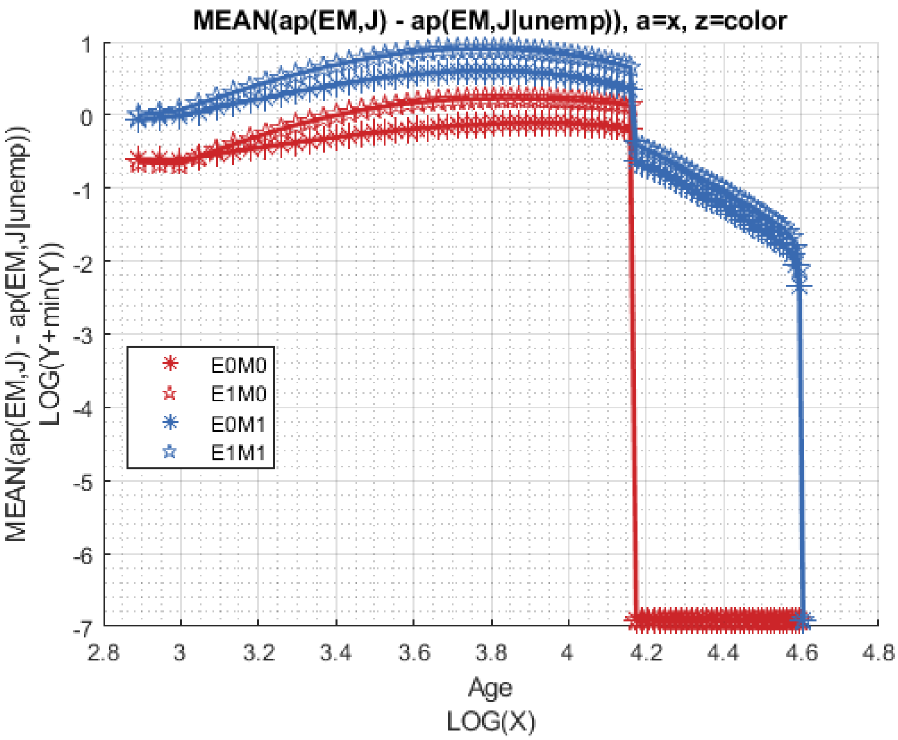




Graph Mean Savings Choices Change:

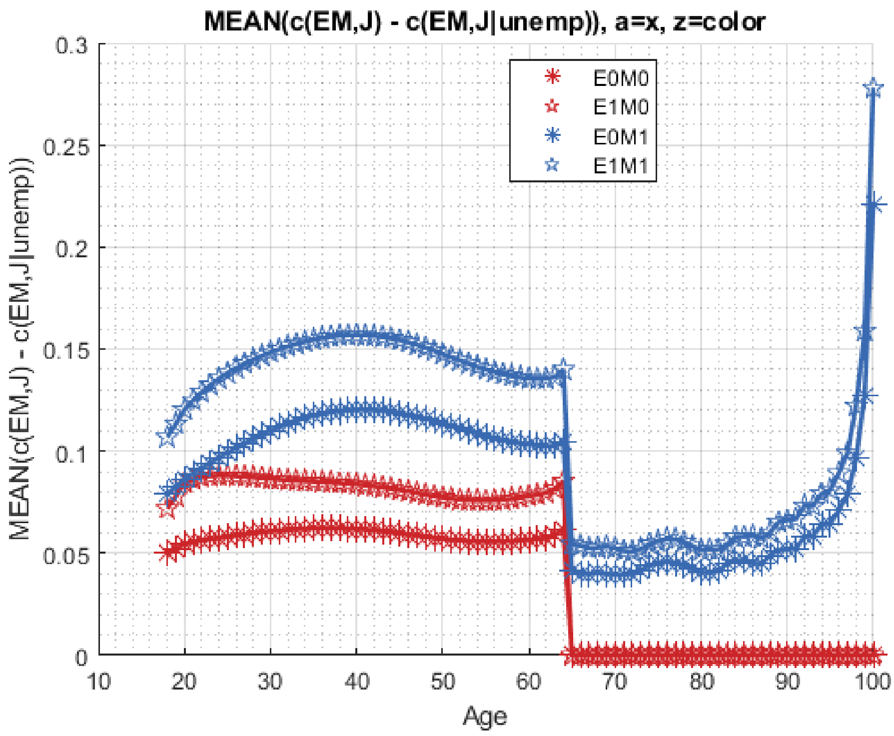
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(EM,J) - ap(EM,J|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(EM,J) - ap(EM,J|unemp))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

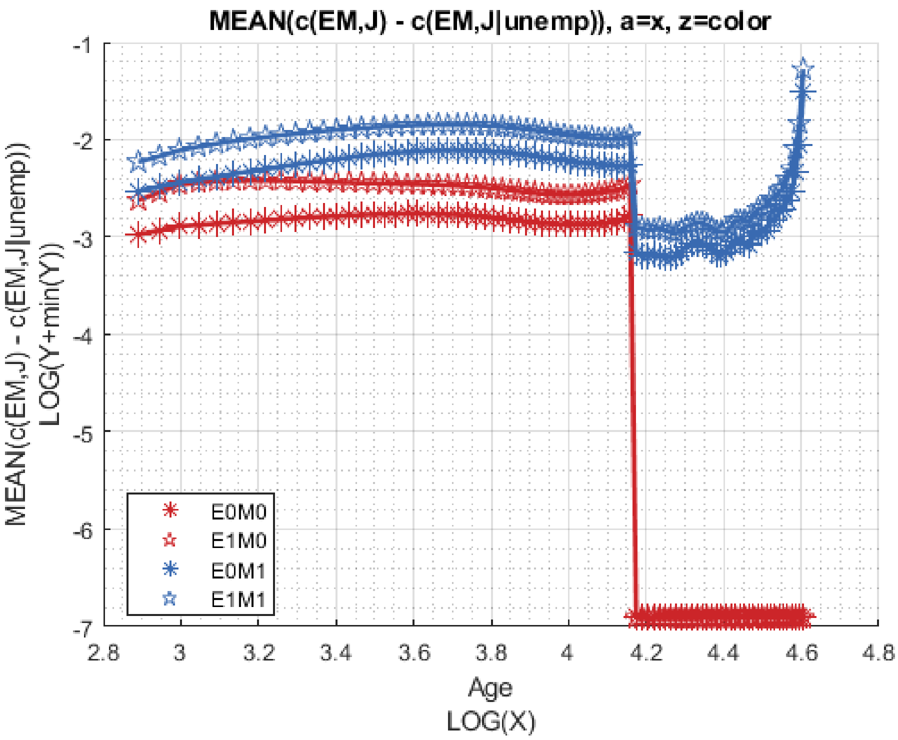




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(c(EM,J) - c(EM,J|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(c(EM,J) - c(EM,J|unemp))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





5.2 Life Cycle Dynamic Programming under Great Recession Unemployment Shock

This is the example vignette for function: **snw_v08p08_jaeemk** from the **PrjOptiSNW Package**. Solving the dynamic programming problem conditional on having an one period unemployment shock that is expected with known unemployment probability. Unemployment probability is a function of the realized state-space next year, specifically, it is determined by age and education. Bush 2008 checks were received by households in expectation of forth-coming unemployment shocks, ex-ante the realization of shocks. During COVID, the shocks were received ex-post the realization of shocks. In both cases, stimulus checks were determined based on ex-ante information.

Due to expected shock, households consume less and save more in 2008 than under steady-state, as shown below. Value/welfare overall is lower in 2008 than under steady-state.

5.2.1 Test SNW_V08P08_JAEEMK

First, solve for value without unemployment issue (use the vectorized code that was previously tested). This is the steady state results, but also the results in 2009 without unemployment.

```
mp_more_inputs = containers.Map('KeyType','char', 'ValueType','any');
mp_more_inputs('fl_ss_non_college') = 0.225;
mp_more_inputs('fl_ss_college') = 0.271;
mp_more_inputs('fl_scaleconvertor') = 54831;
% st_param_group = 'default_small';
% st_param_group = 'default_dense';
st_param_group = 'default_docdense';
mp_params = snw_mp_param(st_param_group, false, 'tauchen', false, 8, 8, mp_more_inputs);
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
[V_VFI_ss, ap_VFI_ss, cons_VFI_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=526.
```

```
V_emp_2009 = V_VFI_ss;
```

Second, solve for the unemployment value, use the exact-bisec result code, call the snw_vfi_main_bisec_vec.m function with a third input of existing value. xi is the share of income lost during covid year given surprise covid shock, b is the share of income loss that is covered by unemployment insurance. If xi=0.5 and b=0 means will lose 50 percent of income given 2009 great recession shocks, and the loss will not be covered at all by unemployment insurance.

```
mp_params('xi') = 0.532;
mp_params('b') = 0.37992;
mp_params('a2_covidyr') = mp_params('a2_greatrecession_2009');
[V_unemp_2009] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_VFI_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
```

Third, solve for 2008 policy and value funtion given employed and unemployed value function in 2009,

```
[V_2008, ap_2008, cons_2008, ev_empshk_2009] = ...
    snw_v08p08_jaeemk(mp_params, mp_controls, V_emp_2009, V_unemp_2009);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
Completed SNW_V08P08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=534.4681
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                 i    idx    ndim     numel      rowN      colN          sum          mean       std
                 -    ---    ----    --------    ----    ---------    -----------    -------    ----

    V_2008       1     1      6      4.37e+07     83     5.265e+05    -8.6418e+08    -19.775     28.
    ap_2008      2     2      6      4.37e+07     83     5.265e+05     1.4164e+09     32.413    36.7
    cons_2008    3     3      6      4.37e+07     83     5.265e+05     2.1314e+08     4.8774    8.32
```

xxx TABLE:V_2008 xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	-377.6	-377.2	-374.55	-368.41	-358.7	-6.6852	-6.5348	-6.384
r2	-365.34	-364.94	-362.31	-356.26	-346.87	-6.4943	-6.3486	-6.202
r3	-353.22	-352.83	-350.3	-344.47	-335.51	-6.3002	-6.1591	-6.016
r4	-341.2	-340.84	-338.49	-333.04	-324.62	-6.1007	-5.964	-5.825
r5	-330.3	-329.97	-327.77	-322.66	-314.72	-5.9113	-5.7784	-5.643
r79	-13.739	-13.726	-13.636	-13.409	-13.022	-0.22845	-0.21772	-0.2076
r80	-12.3	-12.287	-12.198	-11.97	-11.583	-0.17425	-0.16609	-0.158
r81	-10.552	-10.538	-10.449	-10.221	-9.8344	-0.11926	-0.11367	-0.1084
r82	-8.2458	-8.2327	-8.1431	-7.9156	-7.5286	-0.065967	-0.062837	-0.05992
r83	-4.9602	-4.9471	-4.8576	-4.6301	-4.2431	-0.020966	-0.019971	-0.01903

xxx TABLE:ap_2008 xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497
r1	0.00051498	0.00051498	0.0032921	0.0091501	0.023607	114.76	120.42
r2	0.00051498	0.00051498	0.0030624	0.0084089	0.022296	114.88	120.54
r3	0	0	0.0016162	0.0069216	0.020659	114.99	120.67
r4	0	0	0.0016317	0.0069002	0.020494	115.74	121.44
r5	0	0	0.0016507	0.0068855	0.020347	116.52	122.23
r79	0	0	0	0	0.00051498	81.091	85.68
r80	0	0	0	0	0	76.669	80.556
r81	0	0	0	0	0	68.313	71.526
r82	0	0	0	0	0	50.126	53.467
r83	0	0	0	0	0	0	0

xxx TABLE:cons_2008 xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	0.036202	0.036736	0.0377	0.041994	0.047306	9.6346	9.8024	9.9503
r2	0.036202	0.036736	0.037929	0.042735	0.048617	9.7969	9.9535	10.086
r3	0.036717	0.037251	0.039375	0.044222	0.050255	9.962	10.105	10.219
r4	0.038144	0.038678	0.040786	0.045669	0.051843	10.114	10.242	10.338
r5	0.039534	0.040068	0.042157	0.047073	0.053379	10.254	10.367	10.447
r79	0.2016	0.20214	0.20586	0.21598	0.23516	35.82	37.055	38.423
r80	0.2016	0.20214	0.20586	0.21598	0.23568	40.216	42.153	44.428
r81	0.2016	0.20214	0.20586	0.21598	0.23568	48.55	51.16	54.237
r82	0.2016	0.20214	0.20586	0.21598	0.23568	66.719	69.201	71.733
r83	0.2016	0.20214	0.20586	0.21598	0.23568	116.83	122.65	128.67

Difference Between Value and Choices In steady state and in 2008, given expected unemployment (one-period) shock due to the great recession, **snw__v08p08__jaeemk**.

```
V_VFI_unemp_drop = V_VFI_ss - V_2008;
ap_VFI_unemp_drop = ap_VFI_ss - ap_2008;
cons_VFI_unemp_drop = cons_VFI_ss - cons_2008;
```

5.2.2 Define Parameter Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
```

```
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

5.2.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 15; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

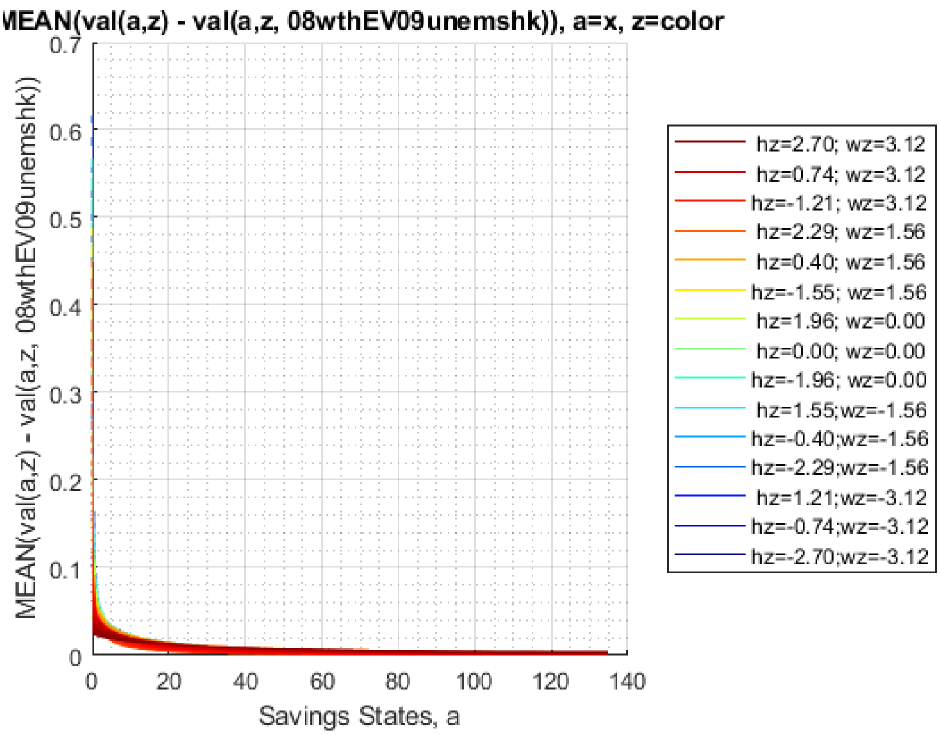
MEAN(VAL(A,Z) - VAL(A,Z, 08wthEV09unemshk)), MEAN(AP(A,Z) - AP(A,Z, 08wthEV09unemshk)), MEAN(C(A,Z) - C(A,Z, 08wthEV09unemshk))

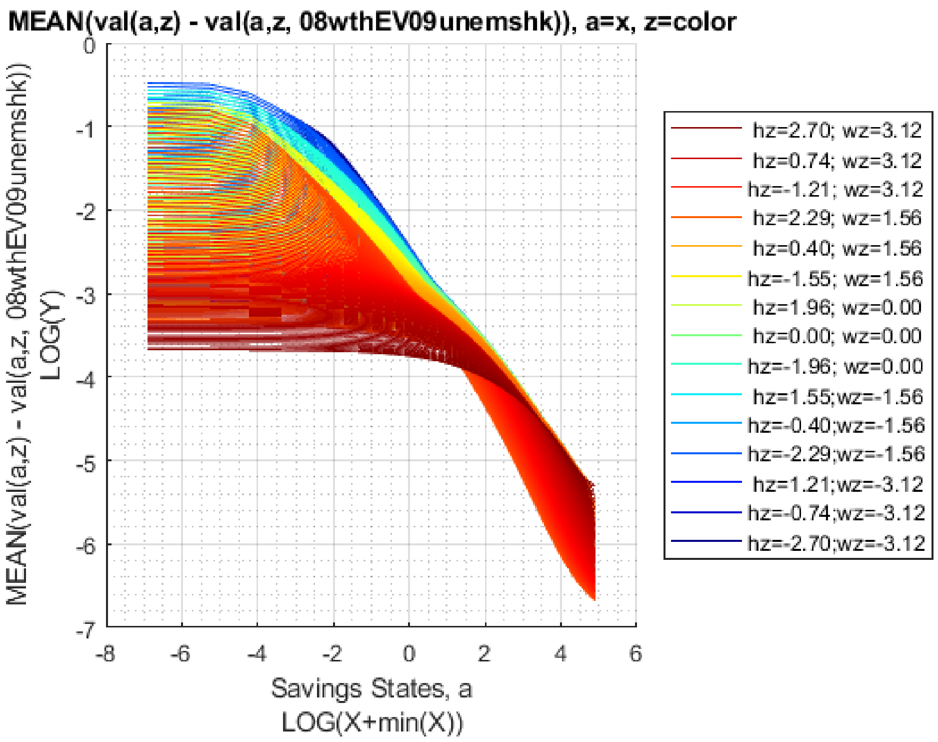
Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(v(A,Z) - v(A,Z, 08wthEV09unemshk))", V_VFI_unemp_drop, true, ["mean

xxx  MEAN(v(A,Z) - v(A,Z, 08wthEV09unemshk))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0       0.61601       0.59253       0.56551       0.53755        0.5101
```

```
5          0.032959     -0.00025565     -0.00039576     -0.00048282     -0.00053831     -0.00059111
```

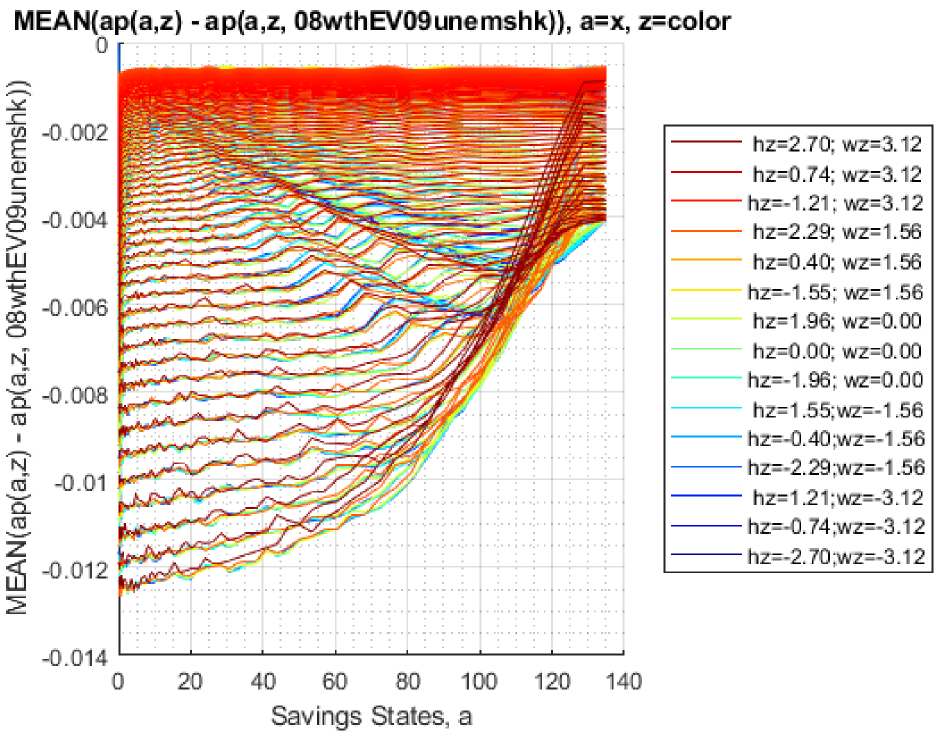
```
mp_support_graph('cl_st_graph_title') = {'MEAN(val(a,z) - val(a,z, 08wthEV09unemshk)), a=x, z=color'
mp_support_graph('cl_st_ytitle') = {'MEAN(val(a,z) - val(a,z, 08wthEV09unemshk))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

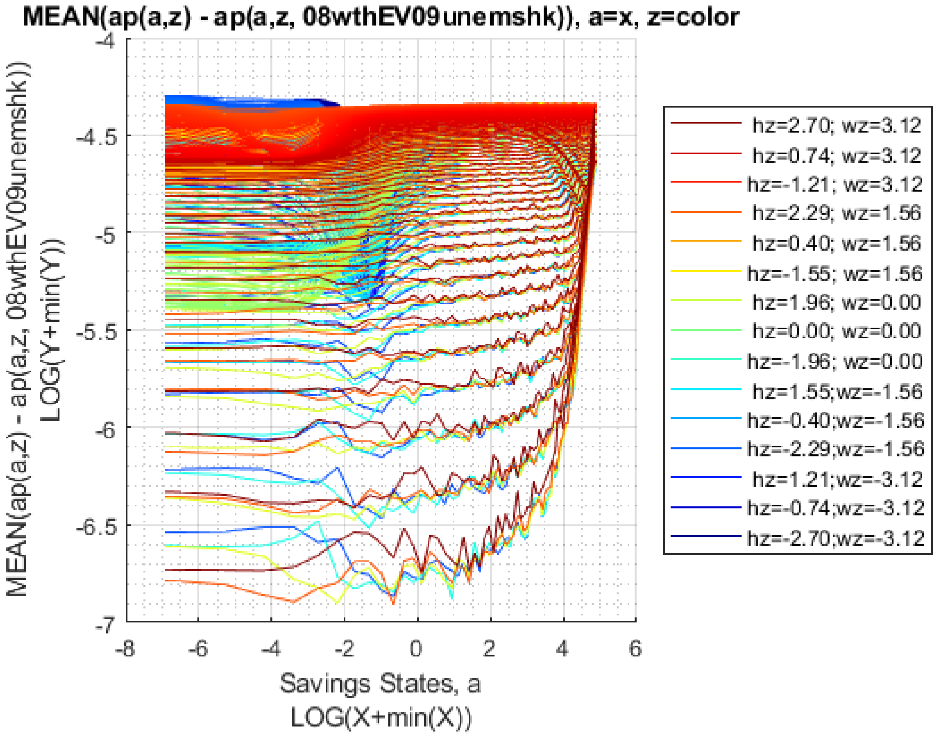




Graph Mean Savings Choices Change:

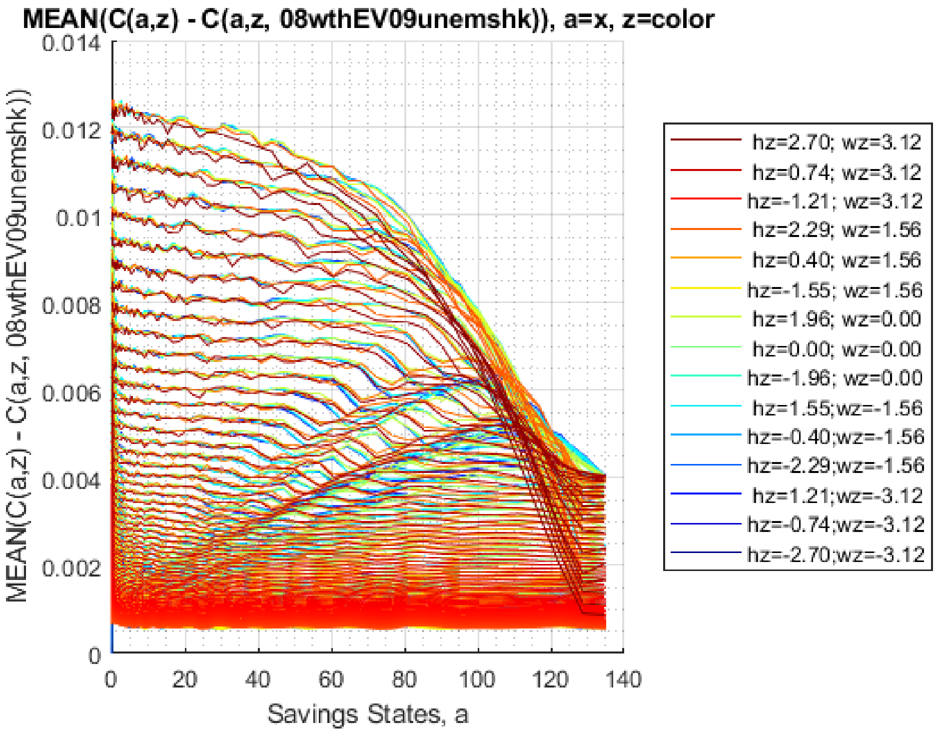
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(a,z) - ap(a,z, 08wthEV09unemshk)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(a,z) - ap(a,z, 08wthEV09unemshk))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

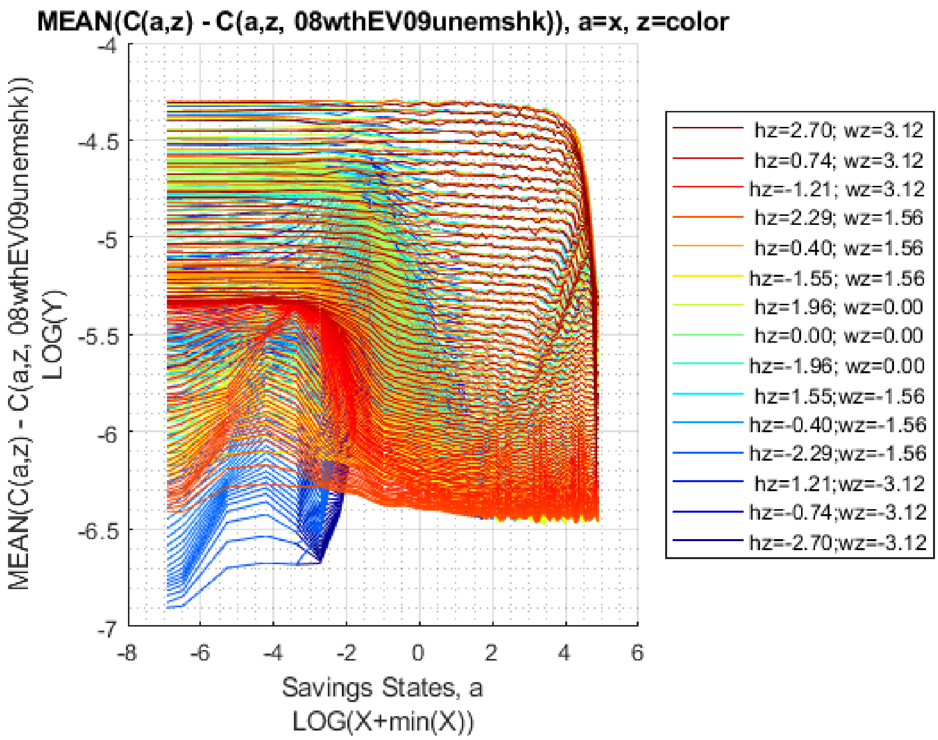




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z) - C(a,z, 08wthEV09unemshk)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z) - C(a,z, 08wthEV09unemshk))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





5.2.4 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(V(KM,J) - V(KM,J | unemp)), MEAN(ap(KM,J) - ap(KM,J | unemp)), MEAN(c(KM,J) - c(KM,J | unemp))

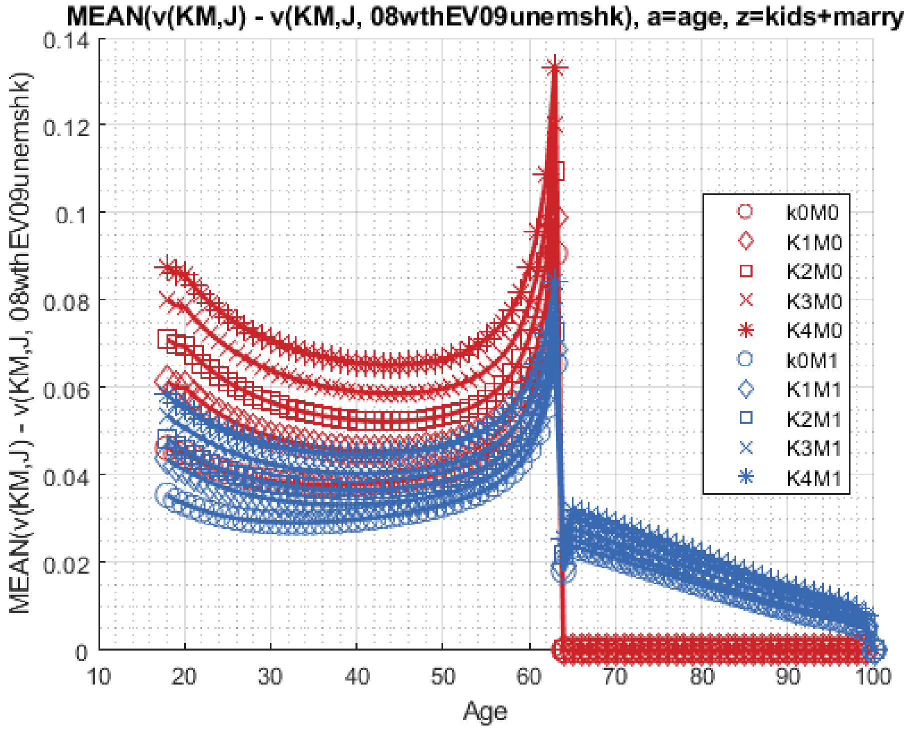
Tabulate value and policies:

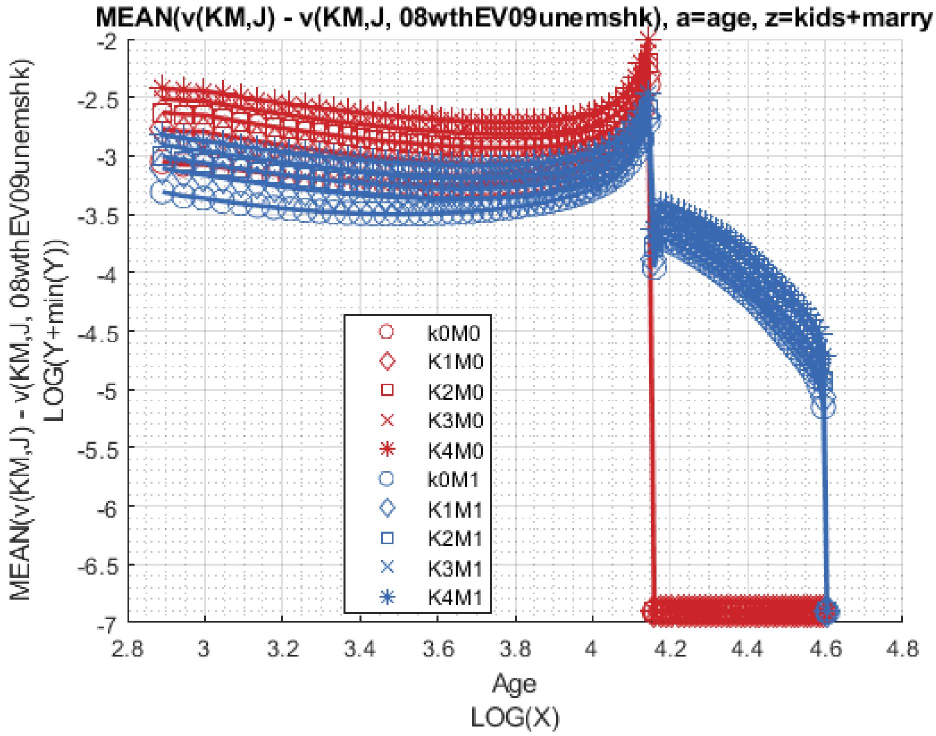
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(V(KM,J) - V(KM,J | unemp))", V_VFI_unemp_drop, true, ["mean"], 3, 1

xxx  MEAN(V(KM,J) - V(KM,J | unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0       0.046286       0.045154       0.045206       0.044045       0.043041
```

```
     2       2       0       0.061509       0.060105       0.060105       0.058309        0.05672
     3       3       0       0.071135       0.069801        0.06956        0.06751       0.065702
     4       4       0       0.080137         0.0788       0.078496       0.076191       0.074156
     5       5       0       0.087487       0.086234       0.085827       0.083393       0.081251
     6       1       1       0.035539       0.034243       0.033548       0.032741       0.032045
     7       2       1       0.043811       0.042205       0.041298       0.040208       0.039233
     8       3       1       0.048496       0.046783       0.045815       0.044615       0.043554
     9       4       1       0.053774       0.051966        0.05096       0.049624       0.048442
    10       5       1       0.058404       0.056626       0.055602        0.05418       0.052931

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(ap(KM,J) - ap(KM,J | unemp))", ap_VFI_unemp_drop, true, ["mean"],

xxx  MEAN(ap(KM,J) - ap(KM,J | unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1       1        0      -0.0020954     -0.0022922     -0.0023124     -0.0023781      -0.002452
      2       2        0      -0.0025384     -0.0028561     -0.0028769      -0.002923     -0.0029984
      3       3        0      -0.0029997     -0.0034133      -0.003403     -0.0033945     -0.0033908
      4       4        0      -0.0033272     -0.0037849     -0.0037592     -0.0037304      -0.003723
      5       5        0      -0.0036714     -0.0040673     -0.0040589     -0.0040155     -0.0039849
      6       1        1      -0.0033041      -0.003509     -0.0035692     -0.0036765     -0.0037787
      7       2        1      -0.0035178     -0.0037456      -0.003797     -0.0038875     -0.0039924
      8       3        1      -0.0038496     -0.0040777       -0.00411     -0.0041767     -0.0042559
      9       4        1       -0.004079     -0.0043493     -0.0043317     -0.0043657     -0.0044326
     10       5        1      -0.0043828     -0.0047223     -0.0045492     -0.0045848     -0.0046013

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(c(KM,J) - c(KM,J | unemp))", cons_VFI_unemp_drop, true, ["mean"], 3

xxx  MEAN(c(KM,J) - c(KM,J | unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1       1        0       0.0020954      0.0022922      0.0023124      0.0023781       0.002452
      2       2        0       0.0025384      0.0028561      0.0028769       0.002923      0.0029984
      3       3        0       0.0029997      0.0034133       0.003403      0.0033945      0.0033908
      4       4        0       0.0033272      0.0037849      0.0037592      0.0037304       0.003723
      5       5        0       0.0036714      0.0040673      0.0040589      0.0040155      0.0039849
      6       1        1       0.0033041       0.003509      0.0035692      0.0036765      0.0037787
      7       2        1       0.0035178      0.0037456       0.003797      0.0038875      0.0039924
      8       3        1       0.0038496      0.0040777        0.00411      0.0041767      0.0042559
      9       4        1        0.004079      0.0043493      0.0043317      0.0043657      0.0044326
     10       5        1       0.0043828      0.0047223      0.0045492      0.0045848      0.0046013
```

Graph Mean Values Change:

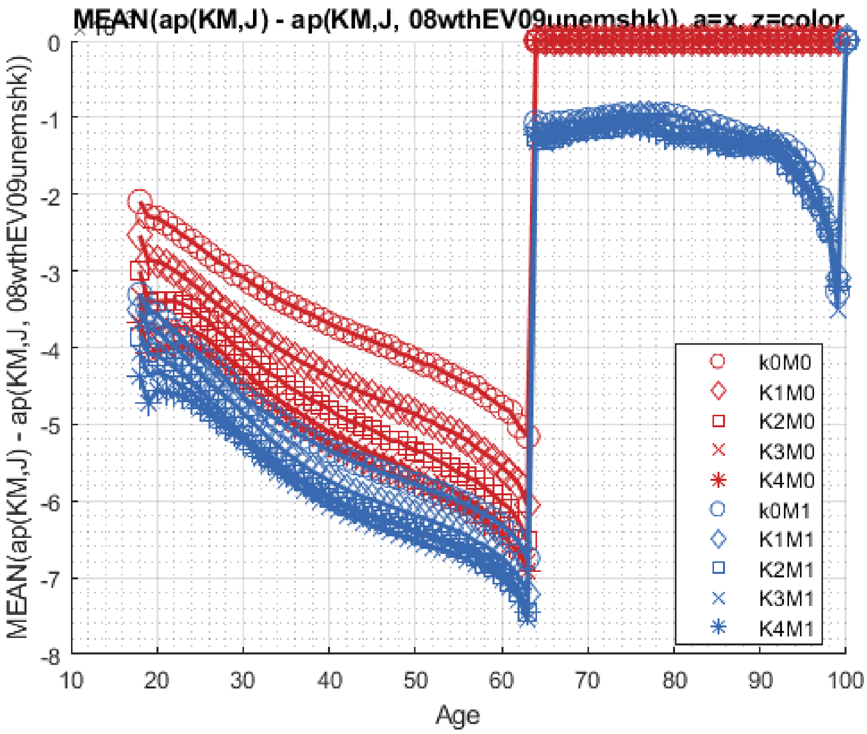
```
mp_support_graph('cl_st_graph_title') = {'MEAN(v(KM,J) - v(KM,J, 08wthEV09unemshk), a=age, z=kids+ma
mp_support_graph('cl_st_ytitle') = {'MEAN(v(KM,J) - v(KM,J, 08wthEV09unemshk)'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

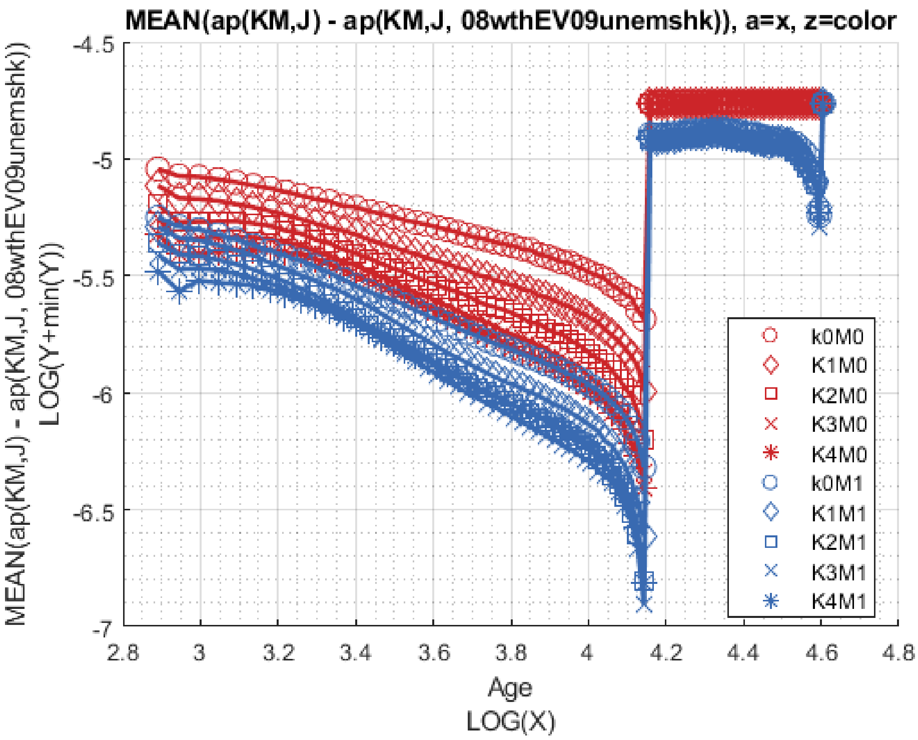




Graph Mean Savings Choices Change:

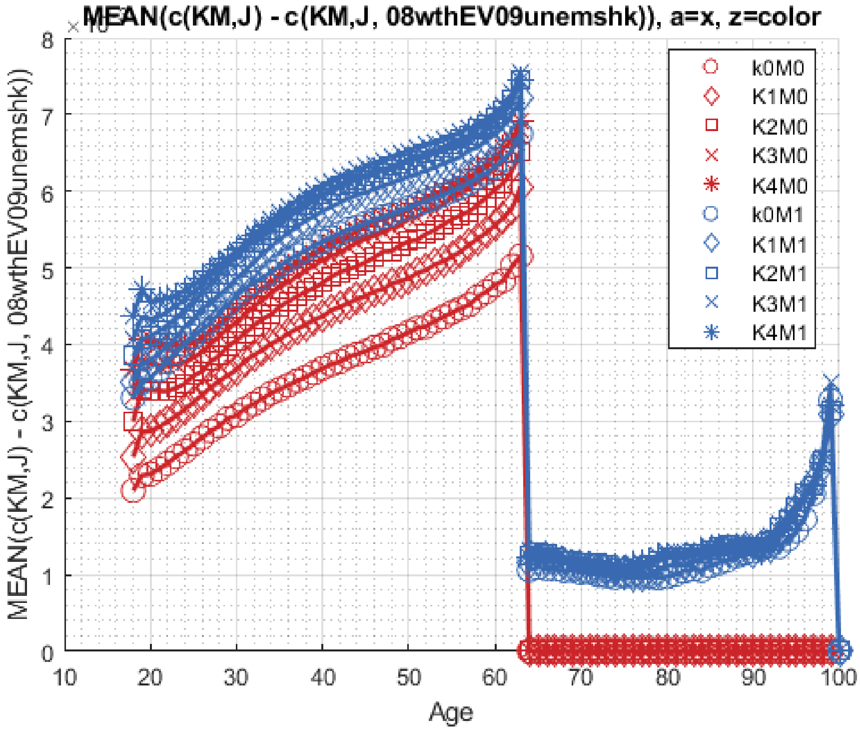
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(KM,J) - ap(KM,J, 08wthEV09unemshk)), a=x, z=color'
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(KM,J) - ap(KM,J, 08wthEV09unemshk))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

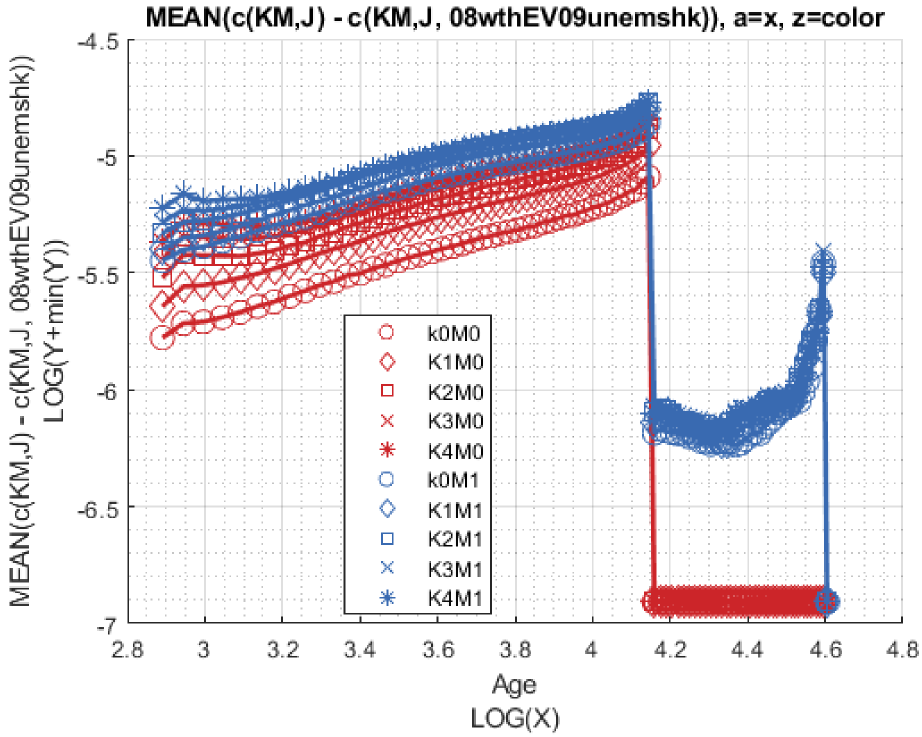




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(c(KM,J) - c(KM,J, 08wthEV09unemshk)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(c(KM,J) - c(KM,J, 08wthEV09unemshk))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





5.2.5 Analyze Education and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

```
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

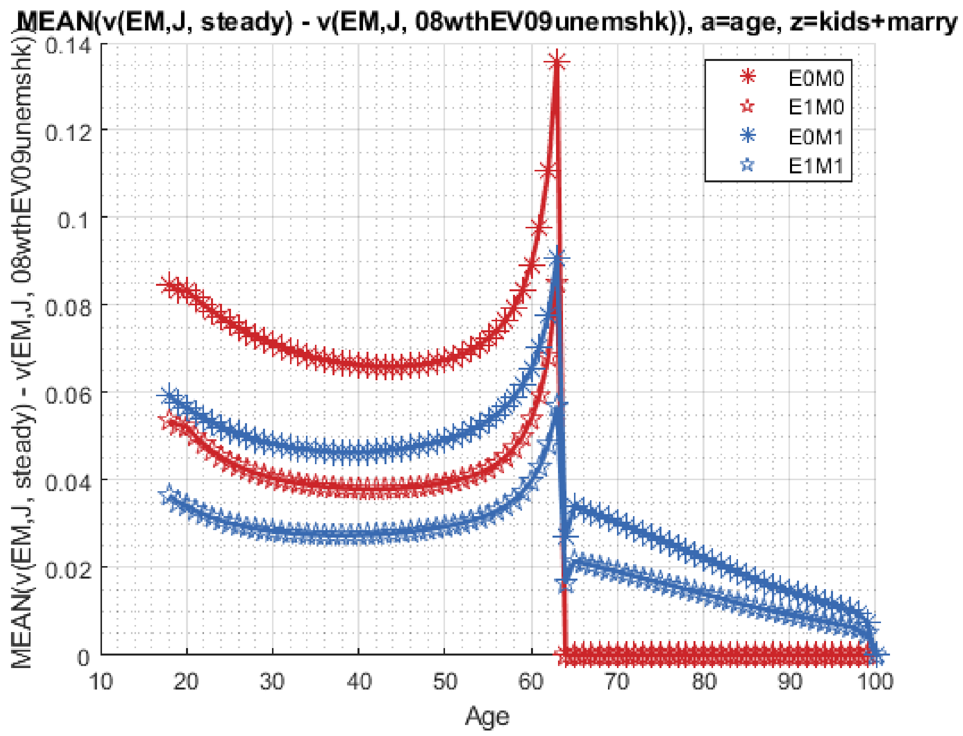
MEAN(v(EKM,J) - v(EKM,J, 08wthEV09unemshk)), MEAN(ap(EM,J, steady) - ap(EM,J, 08wthEV09unemshk)), MEAN(c(EM,J, steady) - c(EM,J, 08wthEV09unemshk))

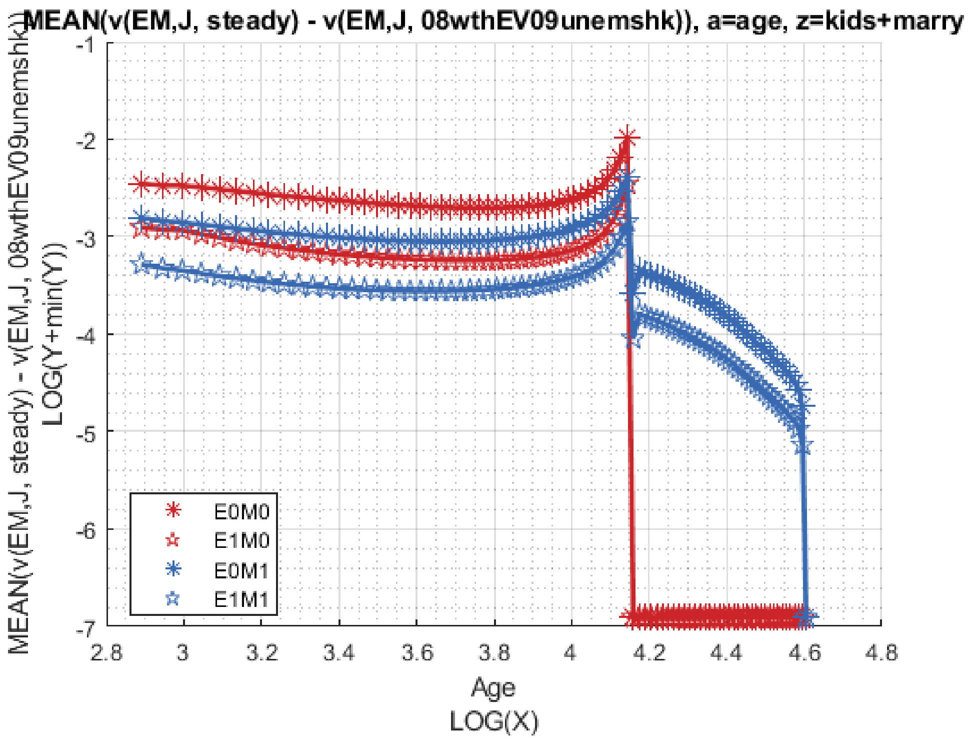
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(v(EM,J, steady) - v(EM,J, 08wthEV09unemshk))", V_VFI_unemp_drop, tr

xxx  MEAN(v(EM,J, steady) - v(EM,J, 08wthEV09unemshk))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0        0.084881       0.083361       0.083388        0.08165       0.080055
      2       1       0         0.05374       0.052677       0.052289       0.050129       0.048293
      3       0       1        0.059454       0.057684       0.056623       0.055357        0.05421
      4       1       1        0.036556       0.035045       0.034267        0.03319       0.032272

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(ap(EM,J, steady) - ap(EM,J, 08wthEV09unemshk))", ap_VFI_unemp_drop

xxx  MEAN(ap(EM,J, steady) - ap(EM,J, 08wthEV09unemshk))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0      -0.0028826      -0.003052     -0.0031902     -0.0032817     -0.0033667
      2       1       0      -0.0029702     -0.0035135      -0.003374     -0.0032948      -0.003253
      3       0       1       -0.003801     -0.0039575     -0.0040645     -0.0041885     -0.0043137
      4       1       1      -0.0038523      -0.004204     -0.0040784      -0.004088     -0.0041106

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(c(EM,J, steady) - c(EM,J, 08wthEV09unemshk))", cons_VFI_unemp_drop,

xxx  MEAN(c(EM,J, steady) - c(EM,J, 08wthEV09unemshk))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0       0.0028826       0.003052      0.0031902      0.0032817      0.0033667
      2       1       0       0.0029702      0.0035135       0.003374      0.0032948       0.003253
      3       0       1        0.003801      0.0039575      0.0040645      0.0041885      0.0043137
      4       1       1       0.0038523       0.004204      0.0040784       0.004088      0.0041106
```

Graph Mean Values Change:

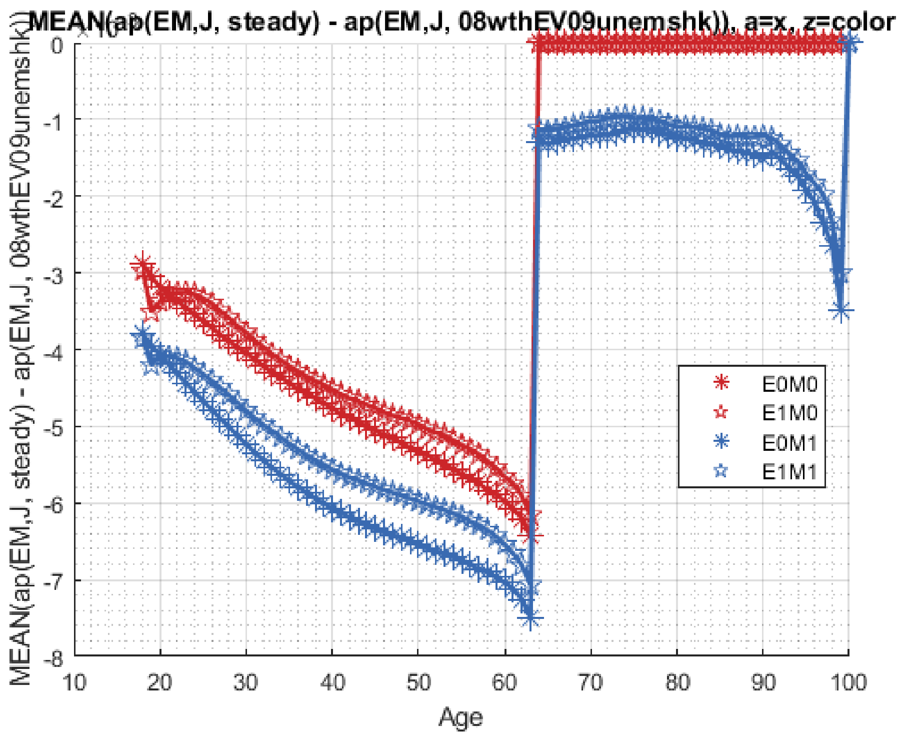
```
mp_support_graph('cl_st_graph_title') = {'MEAN(v(EM,J, steady) - v(EM,J, 08wthEV09unemshk)), a=age,
mp_support_graph('cl_st_ytitle') = {'MEAN(v(EM,J, steady) - v(EM,J, 08wthEV09unemshk))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

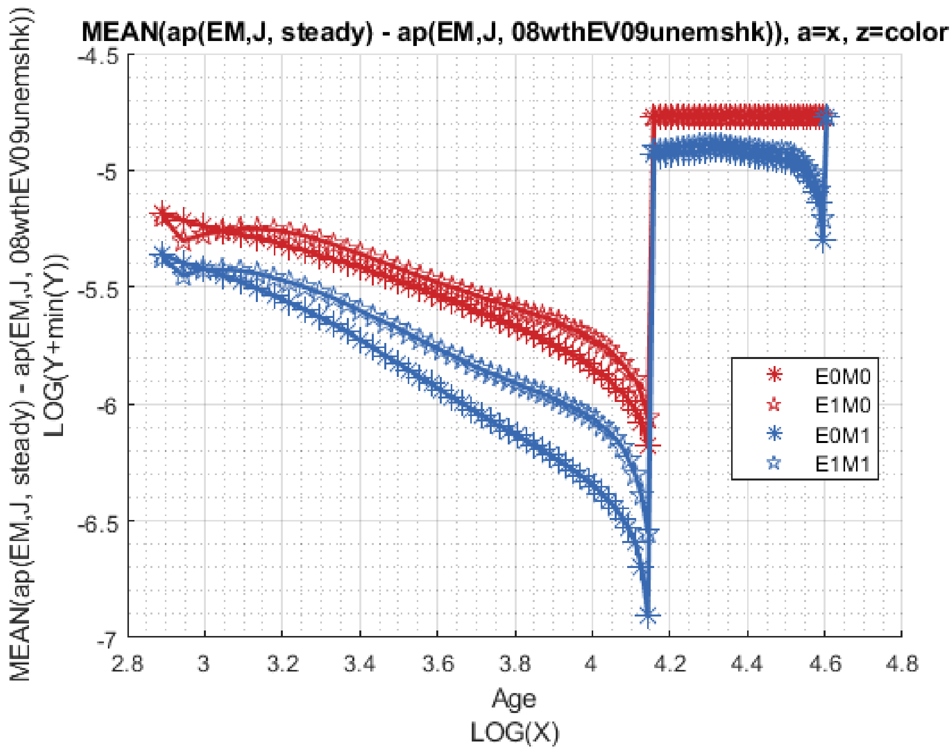




Graph Mean Savings Choices Change:

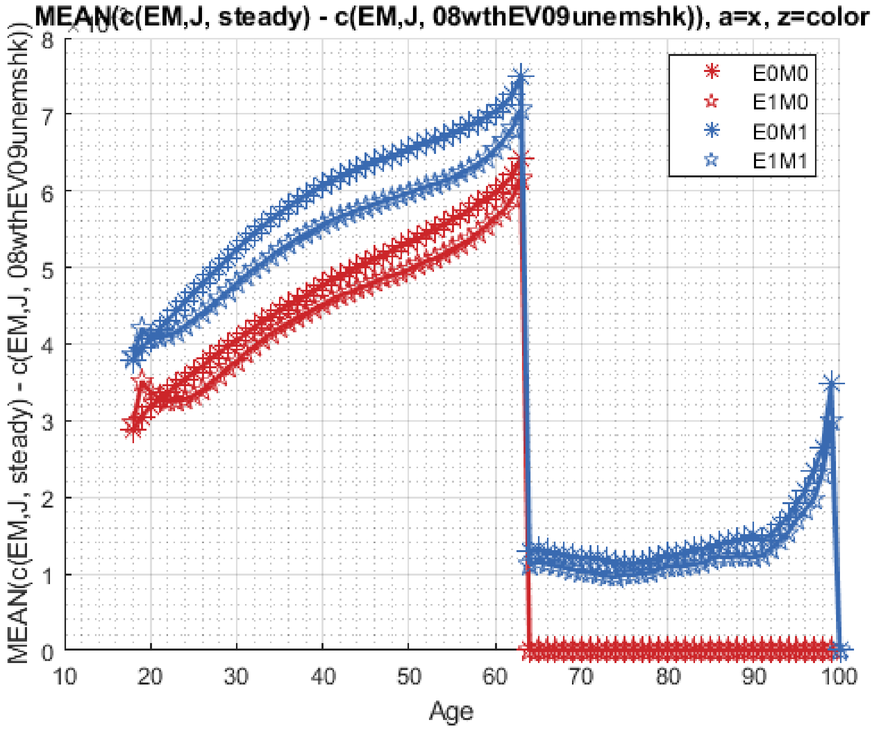
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(EM,J, steady) - ap(EM,J, 08wthEV09unemshk)), a=x,
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(EM,J, steady) - ap(EM,J, 08wthEV09unemshk))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

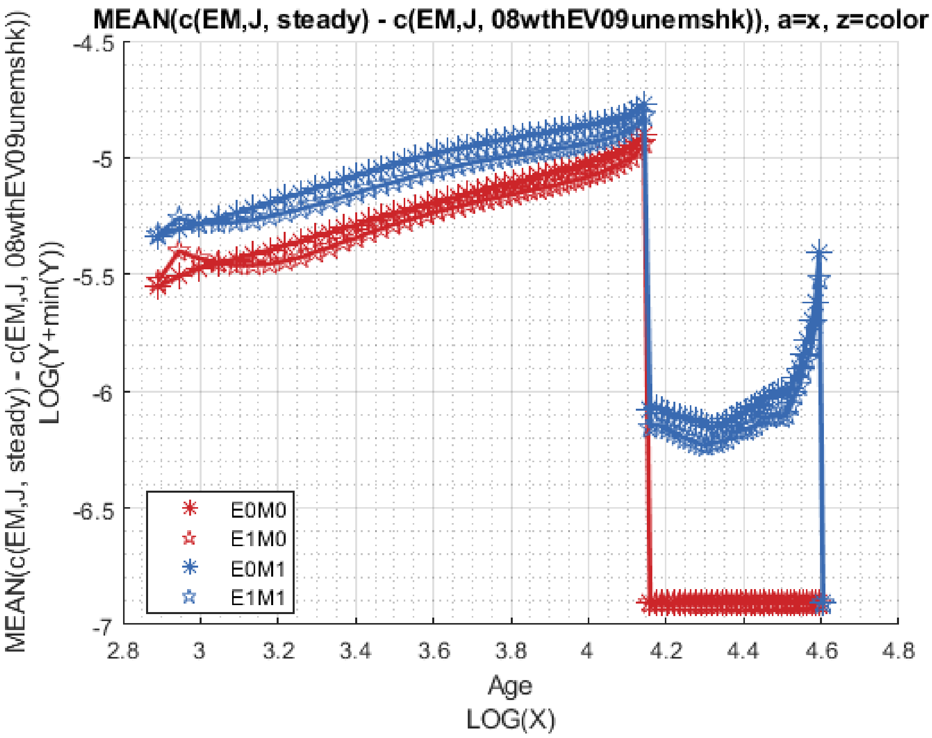




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(c(EM,J, steady) - c(EM,J, 08wthEV09unemshk)), a=x, z=
mp_support_graph('cl_st_ytitle') = {'MEAN(c(EM,J, steady) - c(EM,J, 08wthEV09unemshk))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```


MEAN(c(EM,J, steady) - c(EM,J, 08wthEV09unemshk)), a=x, z=color
MEAN(c(EM,J, steady) - c(EM,J, 08wthEV09unemshk))
Age
E0M0
E1M0
E0M1
E1M1


MEAN(c(EM,J, steady) - c(EM,J, 08wthEV09unemshk)), a=x, z=color
MEAN(c(EM,J, steady) - c(EM,J, 08wthEV09unemshk))
LOG(Y+min(Y))
Age
LOG(X)
E0M0
E1M0
E0M1
E1M1

Chapter 6

Solution with First and Second Rounds CARES Stimulus

6.1 Life Cycle Dynamic Programming under with CARES Act Stimulus Checks

This is the example vignette for function: **snw_vfi_main_bisec_vec_stimulus** from the **PrjOptiSNW Package.** This function solves for policy function using Exact Vectorized Solution. Value in 2020 with surprise COVID unemployment Shock, with non-covid year Value as the continuation function, and provides households with stimulus checks specified in the 1st and 2nd round under actual Trump admin policies. The file focuses on the change in value function, asset choice, and consumption choice given a one period unemployment shock (that does not reappear in the future again). Solving this provides the distribution needed for the Biden checks, American Rescue Plan, problem.

6.1.1 Test SNW_VFI_MAIN_BISEC_VEC_STIMULUS

Solve the Regular Value and Also the Unemployment Value.

First, solve for value without unemployment issue (use the vectorized code that was previously tested):

```
mp_params = snw_mp_param('default_docdense');
mp_controls = snw_mp_control('default_test');
[V_VFI_ss,ap_VFI_ss,cons_VFI_ss,mp_valpol_more_ss] = ...
    snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:83 of 82, time-this-age:8.1976
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:82 of 82, time-this-age:6.3715
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:81 of 82, time-this-age:6.1286
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:80 of 82, time-this-age:6.0961
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:79 of 82, time-this-age:6.1788
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:78 of 82, time-this-age:6.2505
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:77 of 82, time-this-age:6.1271
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:76 of 82, time-this-age:6.1446
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:75 of 82, time-this-age:5.8643
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:74 of 82, time-this-age:6.3012
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:73 of 82, time-this-age:5.9394
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:72 of 82, time-this-age:6.142
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:71 of 82, time-this-age:5.978
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:70 of 82, time-this-age:6.107
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:69 of 82, time-this-age:6.0351
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:68 of 82, time-this-age:6.1776
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:67 of 82, time-this-age:6.0961
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:66 of 82, time-this-age:6.0576
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:65 of 82, time-this-age:5.9533
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:64 of 82, time-this-age:6.2248
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:63 of 82, time-this-age:5.965
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:62 of 82, time-this-age:6.8146
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:61 of 82, time-this-age:7.1769
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:60 of 82, time-this-age:6.3121
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:59 of 82, time-this-age:6.3631
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:58 of 82, time-this-age:6.1827
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:57 of 82, time-this-age:6.0925
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:56 of 82, time-this-age:6.3425
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:55 of 82, time-this-age:6.3273
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:54 of 82, time-this-age:6.3841
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:53 of 82, time-this-age:6.6004
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:52 of 82, time-this-age:6.2408
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:51 of 82, time-this-age:6.2591
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:50 of 82, time-this-age:6.2645
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:49 of 82, time-this-age:6.3071
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:48 of 82, time-this-age:6.373
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:47 of 82, time-this-age:6.706
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:46 of 82, time-this-age:6.6956
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:45 of 82, time-this-age:6.4754
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:44 of 82, time-this-age:6.4592
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:43 of 82, time-this-age:6.5326
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:42 of 82, time-this-age:6.6336
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:41 of 82, time-this-age:6.485
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:40 of 82, time-this-age:7.2556
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:39 of 82, time-this-age:6.4432
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:38 of 82, time-this-age:6.3289
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:37 of 82, time-this-age:6.8039
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:36 of 82, time-this-age:6.5217
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:35 of 82, time-this-age:6.3529
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:34 of 82, time-this-age:6.8033
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:33 of 82, time-this-age:6.5971
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:32 of 82, time-this-age:6.3756
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:31 of 82, time-this-age:6.4397
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:30 of 82, time-this-age:6.534
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:29 of 82, time-this-age:6.5172
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:28 of 82, time-this-age:6.6958
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:27 of 82, time-this-age:7.0906
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:26 of 82, time-this-age:6.3312
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:25 of 82, time-this-age:6.6834
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:24 of 82, time-this-age:6.5892
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:23 of 82, time-this-age:6.7868
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:22 of 82, time-this-age:6.5247
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:21 of 82, time-this-age:6.5358
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:20 of 82, time-this-age:6.7883
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:19 of 82, time-this-age:6.3319
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:18 of 82, time-this-age:6.4524
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:17 of 82, time-this-age:6.4512
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:16 of 82, time-this-age:6.4475
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:15 of 82, time-this-age:6.4821
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:14 of 82, time-this-age:6.9806
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:13 of 82, time-this-age:6.744
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:12 of 82, time-this-age:6.353
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:11 of 82, time-this-age:6.3679
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:10 of 82, time-this-age:6.5458
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:9 of 82, time-this-age:6.3679
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:8 of 82, time-this-age:6.5428
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:7 of 82, time-this-age:6.8776
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:6 of 82, time-this-age:6.3248
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:5 of 82, time-this-age:6.2553
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:4 of 82, time-this-age:6.66
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:3 of 82, time-this-age:6.4982
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:2 of 82, time-this-age:6.3466
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:1 of 82, time-this-age:6.3944
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=535.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN          sum         mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -8.6673e+08    -19.834    28.17
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.4164e+09     32.412      36.
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05      2.131e+08     4.8764    8.326

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
              c1         c2         c3         c4         c5        c526496      c526497      c526498
            -------    -------    -------    -------    -------    ---------    ---------    --------

    r1      -376.05    -375.66    -373.17     -367.4    -358.05        -6.68      -6.5297      -6.379
    r2       -363.8    -363.41    -360.93    -355.25    -346.25      -6.4892      -6.3437      -6.197
    r3      -351.75    -351.36     -348.9    -343.44     -334.9      -6.2948      -6.1538      -6.011
    r4      -339.81    -339.45    -337.16    -332.06    -324.04       -6.095      -5.9584        -5.8
    r5      -328.99    -328.65    -326.51    -321.72    -314.17      -5.9054      -5.7725      -5.637
    r79     -14.033     -14.02    -13.926    -13.689    -13.287     -0.22848     -0.21775     -0.2076
    r80     -12.564     -12.55    -12.457     -12.22    -11.818     -0.17427     -0.16611     -0.1584
    r81     -10.778    -10.764    -10.671    -10.434    -10.032     -0.11927     -0.11368     -0.1084
    r82     -8.4226    -8.4089    -8.3155    -8.0786    -7.6766     -0.06597     -0.06284    -0.05992
    r83     -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.020968    -0.019972    -0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
            c1    c2        c3            c4            c5         c526496    c526497    c526498    c52
            --    --    ----------    ---------    ----------     -------    -------    -------    ---

    r1      0     0      0.0005656    0.0075134      0.022901     114.76     120.42     126.29     132
    r2      0     0     0.00051498    0.0065334      0.021549     114.87     120.54     126.42     132
    r3      0     0     0.00051498    0.0049294      0.019875     114.98     120.67     126.57     132
    r4      0     0     0.00051498    0.0047937      0.019672     115.74     121.44     127.36     133
    r5      0     0     0.00048517    0.0046683      0.019484     116.51     122.22     128.16     134
    r79     0     0              0            0    0.00051498     81.091      85.68     90.325     94.
    r80     0     0              0            0             0     76.669      80.55     84.292     88.
    r81     0     0              0            0             0     68.313      71.52     74.459     77.
    r82     0     0              0            0             0     50.126     53.467     56.953     58.
    r83     0     0              0            0             0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
               c1          c2          c3          c4          c5        c526496    c526497    c526498
            --------    --------    --------    --------    --------    -------    -------    -------

    r1      0.036717    0.037251    0.040426     0.04363    0.048012     9.6396     9.8066     9.9533
    r2      0.036717    0.037251    0.040477     0.04461    0.049364     9.8014     9.9571     10.088
    r3      0.036717    0.037251    0.040477    0.046214    0.051039     9.9664     10.108      10.22
    r4      0.038144    0.038678    0.041903    0.047776    0.052666     10.118     10.244     10.339
    r5      0.039534    0.040068    0.043323     0.04929    0.054241     10.258     10.369     10.446
```

r79	0.19737	0.19791	0.20163	0.21175	0.23093	35.811	37.046	38.418
r80	0.19737	0.19791	0.20163	0.21175	0.23145	40.207	42.15	44.426
r81	0.19737	0.19791	0.20163	0.21175	0.23145	48.541	51.158	54.236
r82	0.19737	0.19791	0.20163	0.21175	0.23145	66.71	69.193	71.724
r83	0.19737	0.19791	0.20163	0.21175	0.23145	116.82	122.65	128.66

Second, solve for the unemployment value, use the exact-bisec result code, call the snw_vfi_main_bisec_vec.m function with a third input of existing value. xi is the share of income lost during covid year given surprise covid shock, b is the share of income loss that is covered by unemployment insurance. xi=0.5 and b=0 means will lose 50 percent of income given COVID shocks, and the loss will not be covered at all by unemployment insurance. Calling the **snw_vfi_main_bisec_vec_stimulus** means households will receive positive amounts of stimulus given household structure (marital status and children count), as well as their total household income level.

```
mp_params('xi') = 0.5;
mp_params('b') = 0;
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
[V_VFI_wthtrumpchk,ap_VFI_wthtrumpchecks,cons_VFI_wthtrumpchk,mp_valpol_more_wthtrumpchk] = ...
    snw_vfi_main_bisec_vec_stimulus(mp_params, mp_controls, V_VFI_ss);
```

```
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 1 of 82, time-this-age:6.62
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 2 of 82, time-this-age:6.6881
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 3 of 82, time-this-age:6.8137
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 4 of 82, time-this-age:6.5643
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 5 of 82, time-this-age:6.6084
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 6 of 82, time-this-age:6.658
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 7 of 82, time-this-age:6.8349
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 8 of 82, time-this-age:6.5825
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 9 of 82, time-this-age:6.5439
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 10 of 82, time-this-age:6.6846
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 11 of 82, time-this-age:6.5489
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 12 of 82, time-this-age:6.6835
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 13 of 82, time-this-age:7.0851
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 14 of 82, time-this-age:6.7157
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 15 of 82, time-this-age:6.6425
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 16 of 82, time-this-age:6.8711
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 17 of 82, time-this-age:6.746
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 18 of 82, time-this-age:6.6717
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 19 of 82, time-this-age:6.9565
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 20 of 82, time-this-age:7.0231
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 21 of 82, time-this-age:6.6252
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 22 of 82, time-this-age:6.5266
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 23 of 82, time-this-age:6.6398
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 24 of 82, time-this-age:6.6428
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 25 of 82, time-this-age:6.5884
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 26 of 82, time-this-age:6.9986
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 27 of 82, time-this-age:6.58
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 28 of 82, time-this-age:6.3892
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 29 of 82, time-this-age:6.8717
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 30 of 82, time-this-age:6.5103
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 31 of 82, time-this-age:6.7253
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 32 of 82, time-this-age:7.0548
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 33 of 82, time-this-age:6.7513
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 34 of 82, time-this-age:6.7784
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 35 of 82, time-this-age:6.6639
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 36 of 82, time-this-age:6.8596
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 37 of 82, time-this-age:6.6734
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 38 of 82, time-this-age:6.5029
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 39 of 82, time-this-age:7.02
```

```
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 40 of 82, time-this-age:6.6714
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 41 of 82, time-this-age:6.5598
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 42 of 82, time-this-age:6.7617
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 43 of 82, time-this-age:6.5979
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 44 of 82, time-this-age:6.7787
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 45 of 82, time-this-age:7.1311
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 46 of 82, time-this-age:6.6511
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 47 of 82, time-this-age:6.826
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 48 of 82, time-this-age:6.6529
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 49 of 82, time-this-age:6.447
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 50 of 82, time-this-age:6.5167
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 51 of 82, time-this-age:6.5401
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 52 of 82, time-this-age:6.722
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 53 of 82, time-this-age:6.5741
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 54 of 82, time-this-age:6.9319
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 55 of 82, time-this-age:7.182
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 56 of 82, time-this-age:7.6984
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 57 of 82, time-this-age:6.4441
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 58 of 82, time-this-age:6.891
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 59 of 82, time-this-age:7.0068
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 60 of 82, time-this-age:7.0328
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 61 of 82, time-this-age:6.8954
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 62 of 82, time-this-age:6.9436
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 63 of 82, time-this-age:6.983
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 64 of 82, time-this-age:6.9707
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 65 of 82, time-this-age:6.9944
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 66 of 82, time-this-age:7.3946
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 67 of 82, time-this-age:7.0114
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 68 of 82, time-this-age:7.109
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 69 of 82, time-this-age:7.0221
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 70 of 82, time-this-age:7.0655
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 71 of 82, time-this-age:6.8624
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 72 of 82, time-this-age:6.8953
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 73 of 82, time-this-age:6.9772
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 74 of 82, time-this-age:6.9265
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 75 of 82, time-this-age:6.9556
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 76 of 82, time-this-age:6.8726
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 77 of 82, time-this-age:7.1171
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 78 of 82, time-this-age:7.0941
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 79 of 82, time-this-age:6.6593
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 80 of 82, time-this-age:6.504
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 81 of 82, time-this-age:6.3068
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 82 of 82, time-this-age:6.6383
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 83 of 82, time-this-age:8.1596
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN          sum         mean       std
                -    ---    ----    --------    ----    ---------    ----------    -------    ------

    V_VFI       1     1      6     4.37e+07     83     5.265e+05    -8.759e+08    -20.044    27.874
    ap_VFI      2     2      6     4.37e+07     83     5.265e+05    1.3809e+09     31.599    36.658
    cons_VFI    3     3      6     4.37e+07     83     5.265e+05     2.105e+08     4.8169    8.3229

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496      c526497      c526498
```

```
        -------    -------    -------    -------    -------    ---------    ---------    --------

r1      -369.35    -369.07    -367.11    -361.97    -353.39      -6.8096      -6.6548      -6.500
r2      -357.14    -356.86    -354.97    -349.98    -341.85       -6.618      -6.4683      -6.318
r3      -345.28    -344.99    -343.18    -338.45    -330.78      -6.4227       -6.278      -6.132
r4      -334.13    -333.86    -332.12    -327.68    -320.43      -6.2297      -6.0896      -5.948
r5      -324.01    -323.75    -322.06    -317.88    -310.99      -6.0467      -5.9107      -5.773
r79     -13.388    -13.378    -13.307     -13.12    -12.781      -0.2305     -0.21962     -0.2093
r80     -11.919    -11.908    -11.837    -11.653    -11.317     -0.17582     -0.16751     -0.1596
r81     -10.133    -10.122    -10.051    -9.8679    -9.5372     -0.12032     -0.11462     -0.1092
r82     -7.7774     -7.767    -7.6957    -7.5131    -7.1868    -0.066524    -0.063355    -0.06039
r83     -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.021146    -0.020134    -0.01918

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1           c2            c3            c4           c5       c526496    c526497    c
        ---------    ---------    ----------    ----------    --------    -------    -------    -

r1      0.0041199    0.0043268     0.0080703      0.013905    0.032959     110.07     115.71    1
r2      0.0041199    0.0041199     0.0070902      0.013905    0.032284     110.04     115.69    1
r3      0.0041199    0.0041199     0.0054862      0.013905    0.030609        110     115.66    1
r4      0.0041199    0.0041199     0.0046552      0.013905    0.029711     110.29     115.97    1
r5      0.0035447    0.0040795     0.0041199      0.013905    0.028846     110.59     116.28    1
r79             0            0    0.00041185     0.0041199    0.013905     81.091     85.231    8
r80             0            0             0     0.0041199    0.013905     75.865     79.539    8
r81             0            0             0     0.0020033    0.013501      67.78      70.52    7
r82             0            0             0    0.00051498    0.010558     50.126     53.467    5
r83             0            0             0             0           0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
           c1          c2          c3          c4          c5       c526496    c526497    c526498
        --------    --------    --------    --------    --------    -------    -------    -------

r1      0.043302     0.04363     0.04363    0.047957    0.048689     9.4621     9.6396     9.8066
r2      0.043302    0.043837     0.04461    0.047957    0.049364     9.6318     9.8014     9.9571
r3      0.043302    0.043837    0.046214    0.047957    0.051039     9.8074     9.9664     10.108
r4      0.044033    0.044568    0.047776    0.048687    0.052666      9.971     10.118     10.244
r5       0.04532     0.04532    0.049023    0.049398    0.054241     10.123     10.258     10.369
r79      0.22617      0.2267     0.23002     0.23643     0.24634     34.787     36.471     38.418
r80      0.22617      0.2267     0.23043     0.23643     0.24634     40.001      42.15     44.426
r81      0.22617      0.2267     0.23043     0.23854     0.24675     48.074     51.158     54.236
r82      0.22617      0.2267     0.23043     0.24003     0.24969     65.719     68.202     71.583
r83      0.19737     0.19791     0.20163     0.21175     0.23145     115.84     121.66     127.68
```

Difference Between Value and Choices In Unemployment and Future Periods

```
V_VFI_wthtrumpchk_drop = V_VFI_ss - V_VFI_wthtrumpchk;
ap_VFI_wthtrumpchk_drop = ap_VFI_ss - ap_VFI_wthtrumpchecks;
cons_VFI_wthtrumpchk_drop = cons_VFI_ss - cons_VFI_wthtrumpchk;
```

6.1.2 Define Parameter Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
```

```
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

6.1.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 15; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(VAL(A,Z) - VAL(A,Z|CARESActChecks)), MEAN(AP(A,Z) - AP(A,Z|CARESActChecks)), MEAN(C(A,Z) - C(A,Z|CARESActChecks))

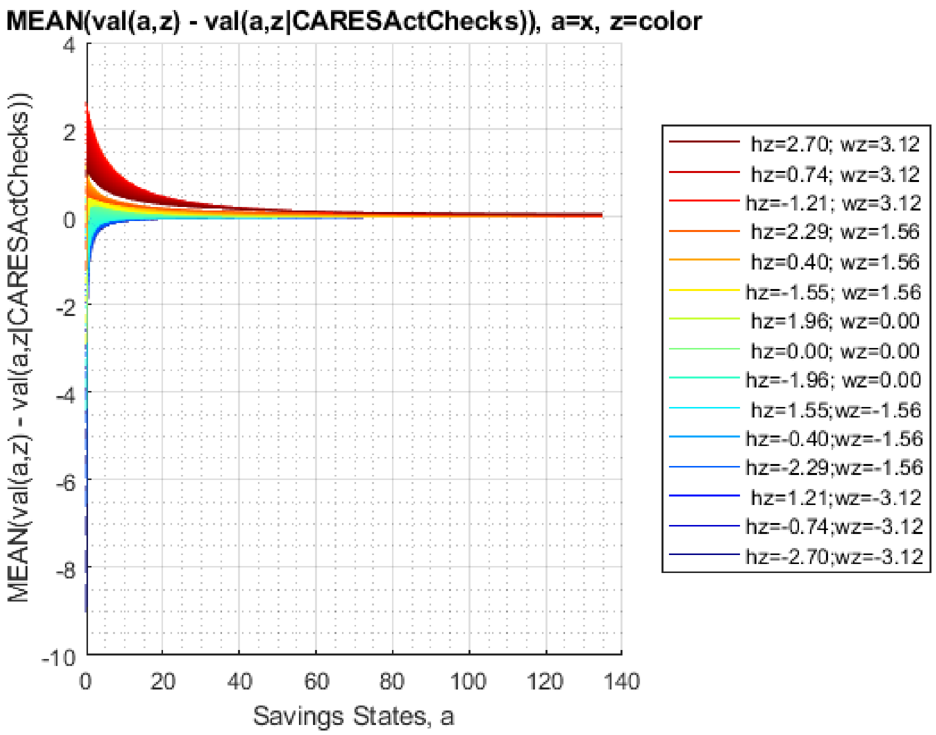
Tabulate value and policies along savings and shocks:

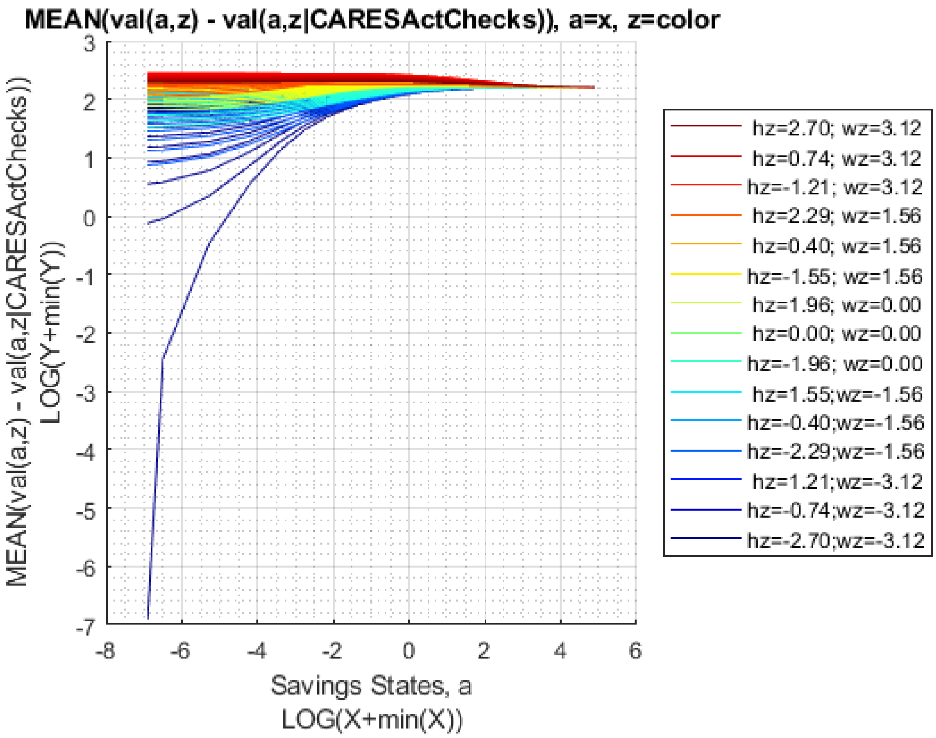
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(v(A,Z) - v(A,Z|CARESActChecks))", V_VFI_wthtrumpchk_drop, true, ["m

xxx  MEAN(v(A,Z) - v(A,Z|CARESActChecks))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5
    -----    ----------    -----------    -----------    -----------    -----------    ----------

      1              0        -9.0028        -8.1198        -7.2767        -6.4867       -5.7583
```

```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(A,Z) - C(A,Z|CARESActChecks))", cons_VFI_wthtrumpchk_drop, true,
```

```
xxx  MEAN(C(A,Z) - C(A,Z|CARESActChecks))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group      savings       mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5
    -----    ----------     -----------    -----------    -----------    -----------    -----------

      1               0      -0.048192      -0.047194      -0.046217      -0.045225      -0.044179
```

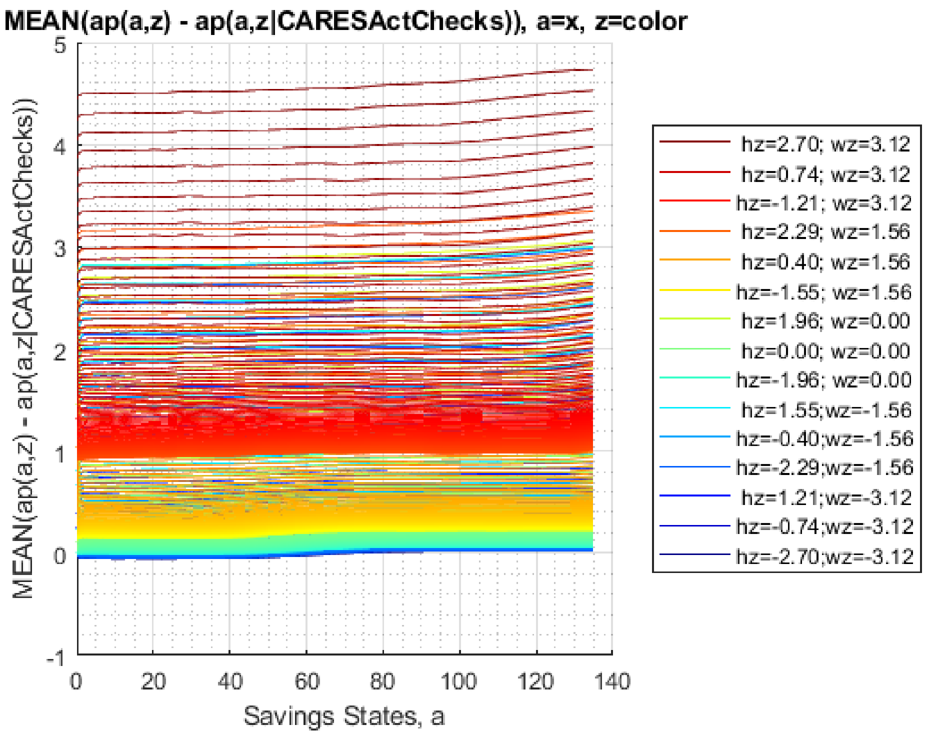
```
mp_support_graph('cl_st_graph_title') = {'MEAN(val(a,z) - val(a,z|CARESActChecks)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(val(a,z) - val(a,z|CARESActChecks))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

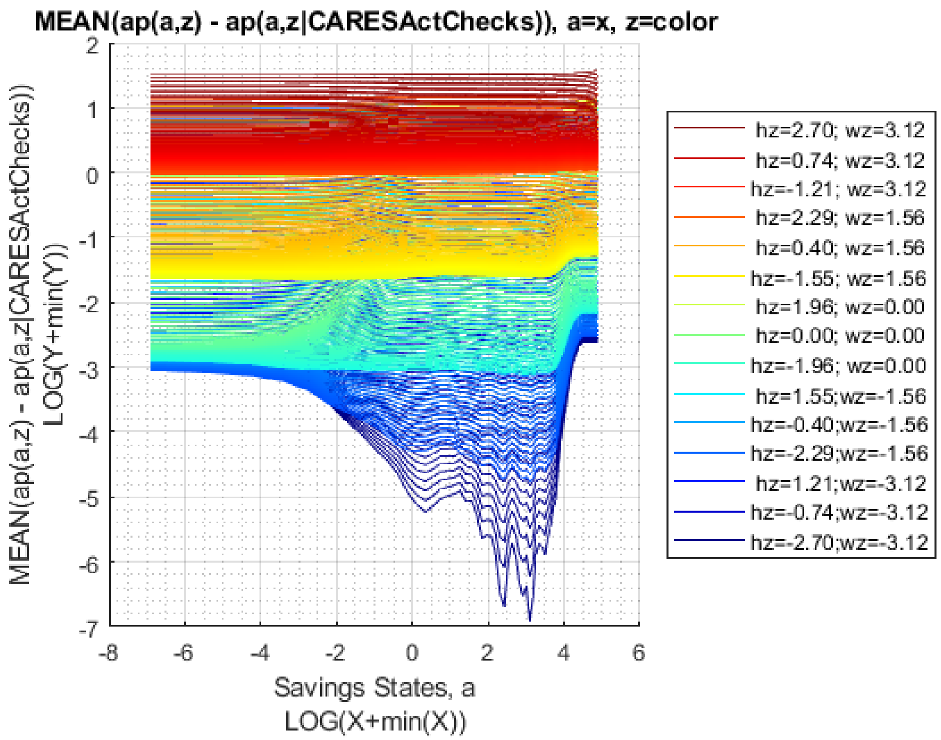




Graph Mean Savings Choices Change:

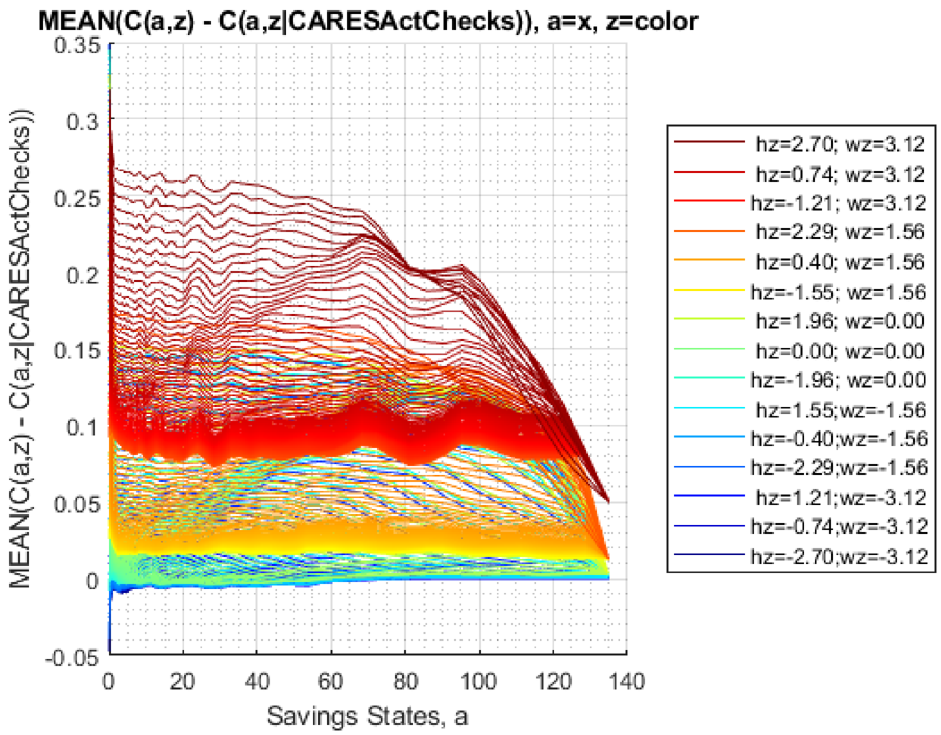
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(a,z) - ap(a,z|CARESActChecks)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(a,z) - ap(a,z|CARESActChecks))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

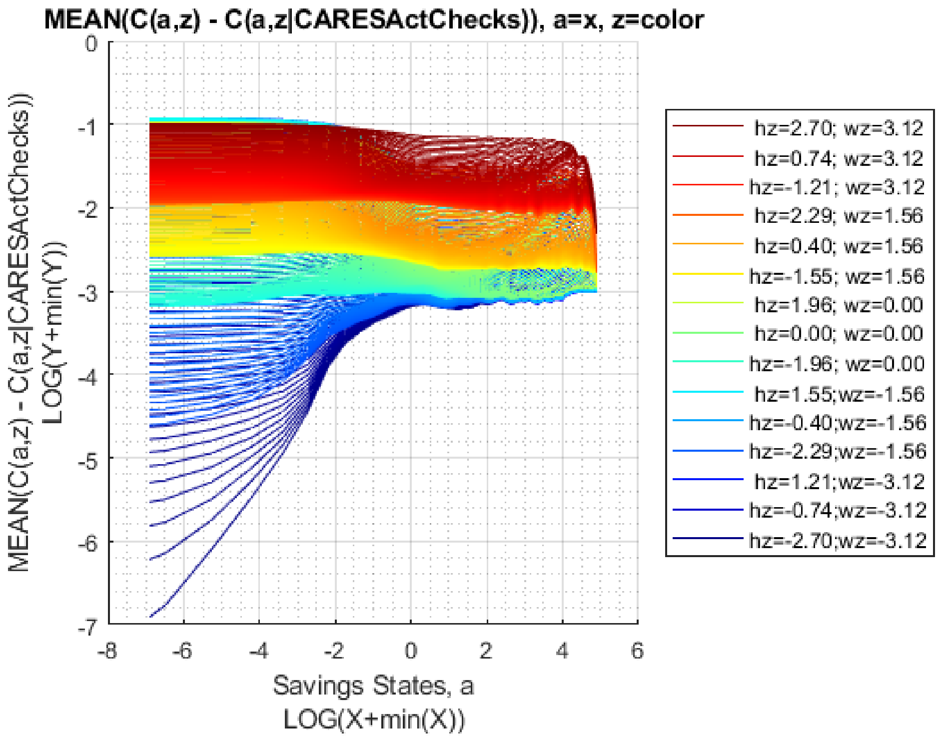




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z) - C(a,z|CARESActChecks)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z) - C(a,z|CARESActChecks))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





6.1.4 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(V(KM,J) - V(KM,J | CARESActChecks)), MEAN(ap(KM,J) - ap(KM,J | CARESActChecks)), MEAN(c(KM,J) - c(KM,J | CARESActChecks))

Tabulate value and policies:

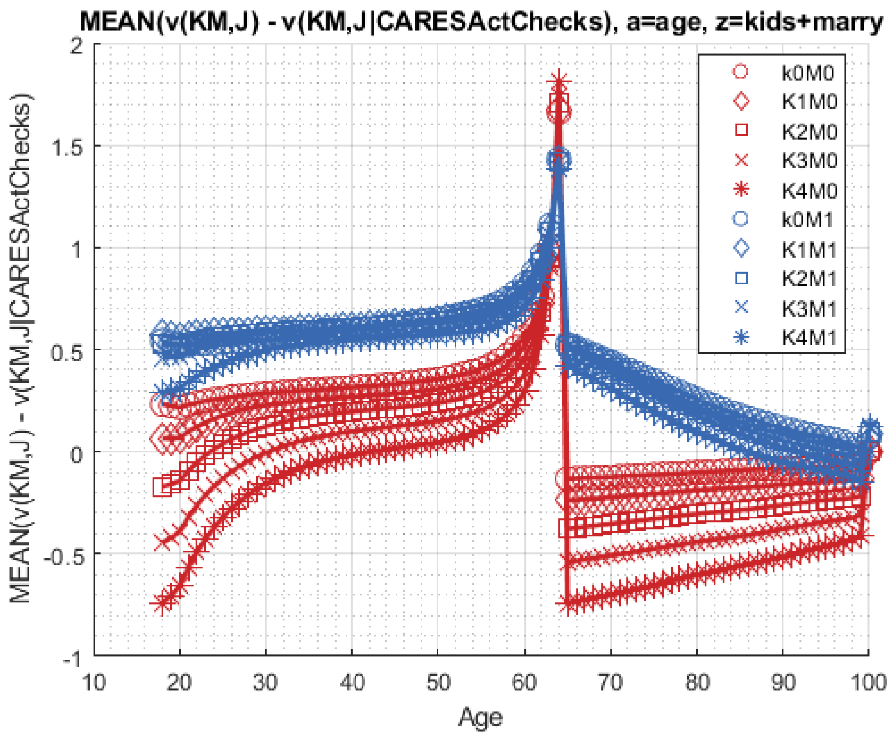
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(V(KM,J) - V(KM,J | CARESActChecks))", V_VFI_wthtrumpchk_drop, true,
```

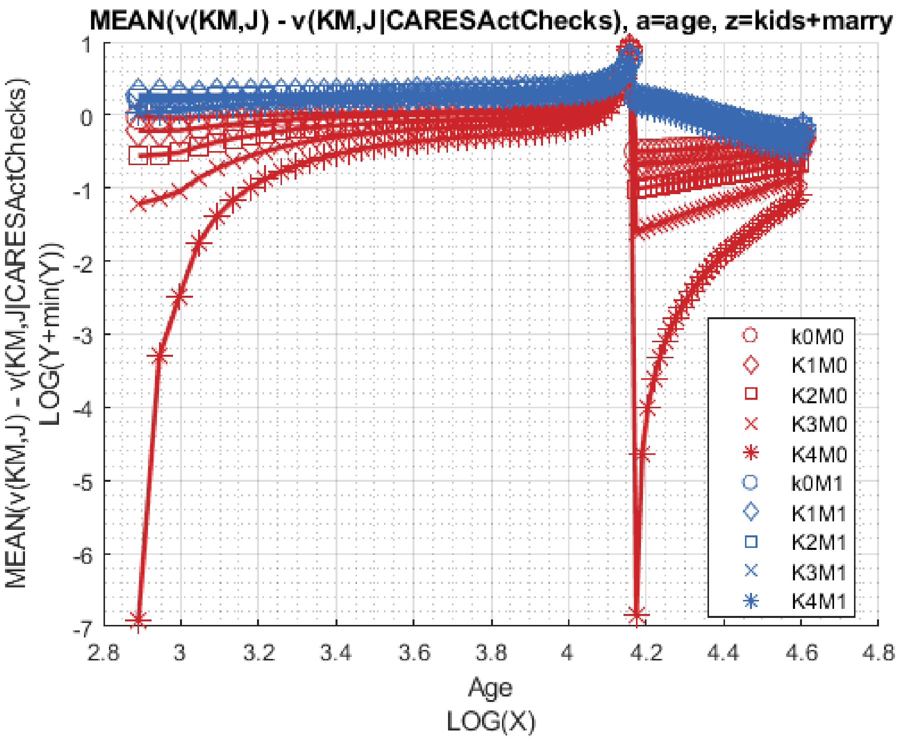
```
xxx  MEAN(V(KM,J) - V(KM,J | CARESActChecks))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1       1        0          0.2332        0.22485        0.21742        0.22953        0.23963
```

```
      2        2        0       0.066539       0.065877       0.068247        0.098285        0.12333
      3        3        0       -0.16926       -0.15841        -0.1423       -0.093192      -0.051804
      4        4        0       -0.44442       -0.42121       -0.39008        -0.32042       -0.26131
      5        5        0       -0.73932       -0.70328       -0.65682        -0.56749       -0.49136
      6        1        1        0.54283        0.53403        0.52485         0.52685        0.52859
      7        2        1          0.572        0.56458        0.55612         0.56246        0.56751
      8        3        1        0.52502        0.52141        0.51661         0.52951         0.5404
      9        4        1        0.46008        0.46188        0.46209         0.48277        0.50057
     10        5        1        0.29213        0.30189        0.31023         0.34193        0.36924

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(ap(KM,J) - ap(KM,J | CARESActChecks))", ap_VFI_wthtrumpchk_drop, t

xxx  MEAN(ap(KM,J) - ap(KM,J | CARESActChecks))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1        0       0.53409        0.53149        0.52842        0.56711        0.60568
      2        2        0       0.51704        0.51346        0.50927        0.54742        0.58546
      3        3        0       0.50183        0.49768        0.49316        0.53101        0.56887
      4        4        0       0.48856        0.48424        0.47955        0.51731        0.55508
      5        5        0       0.47499        0.47072        0.46598        0.50375        0.54161
      6        1        1        1.1088         1.1527         1.1974         1.2901         1.3837
      7        2        1        1.0065         1.0431         1.0802          1.164         1.2484
      8        3        1       0.92804        0.96014        0.99224         1.0702         1.1489
      9        4        1       0.84205        0.86965        0.89728        0.97107         1.0451
     10        5        1       0.71408        0.73273         0.7514         0.8152        0.87929

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(c(KM,J) - c(KM,J | CARESActChecks))", cons_VFI_wthtrumpchk_drop, tr

xxx  MEAN(c(KM,J) - c(KM,J | CARESActChecks))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1        0      0.047456        0.05006       0.053131       0.053501       0.053678
      2        2        0      0.051258       0.054838       0.059034       0.060117        0.06099
      3        3        0      0.055768       0.059917       0.064439       0.065978       0.067166
      4        4        0      0.057894       0.062211       0.066905       0.068696       0.070118
      5        5        0      0.059903       0.064171       0.068906       0.070851        0.07233
      6        1        1        0.0854       0.090837       0.096389        0.10046        0.10399
      7        2        1      0.079182       0.084761       0.090437       0.095109       0.099255
      8        3        1      0.078652       0.083563       0.089111       0.093893       0.098067
      9        4        1      0.082412       0.086456       0.091037       0.093958       0.096652
     10        5        1      0.085584       0.089889       0.094581       0.097555        0.10014
```

Graph Mean Values Change:

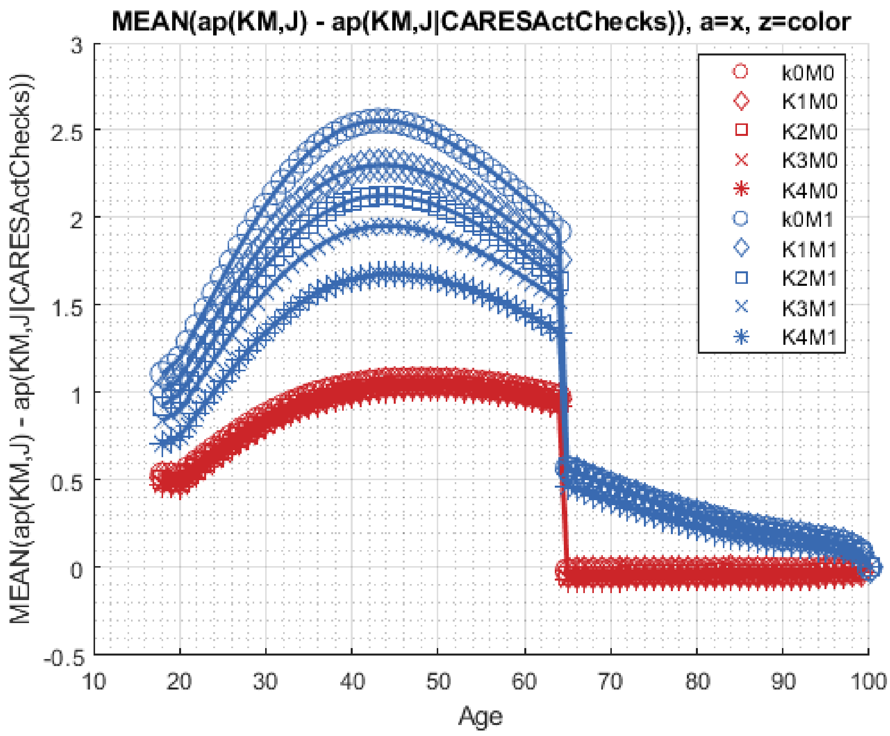
```
mp_support_graph('cl_st_graph_title') = {'MEAN(v(KM,J) - v(KM,J|CARESActChecks), a=age, z=kids+marry
mp_support_graph('cl_st_ytitle') = {'MEAN(v(KM,J) - v(KM,J|CARESActChecks)'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

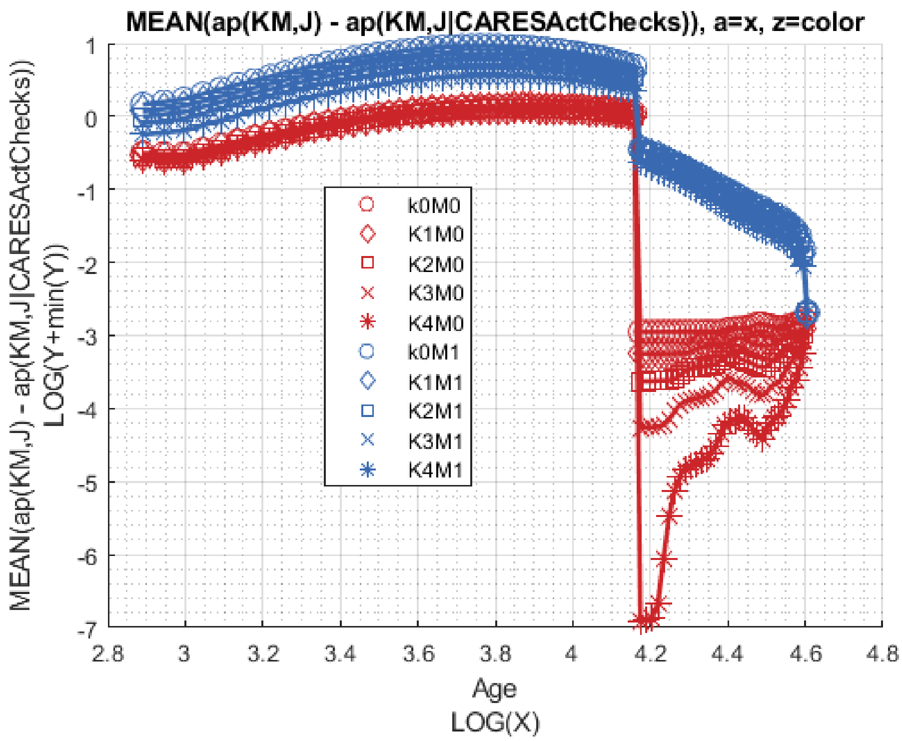




Graph Mean Savings Choices Change:

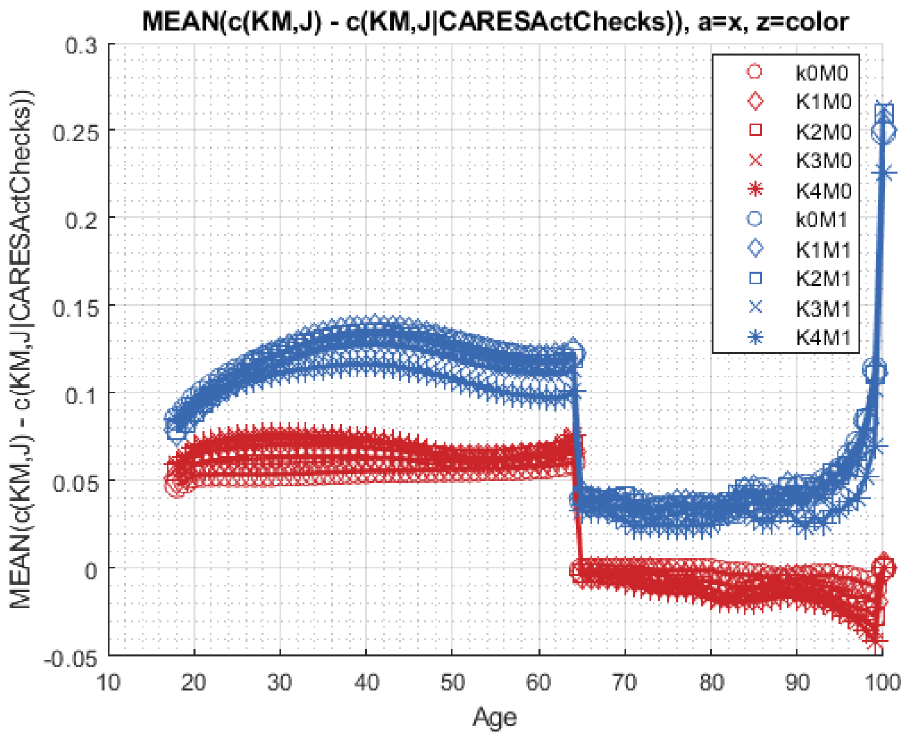
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(KM,J) - ap(KM,J|CARESActChecks)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(KM,J) - ap(KM,J|CARESActChecks))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

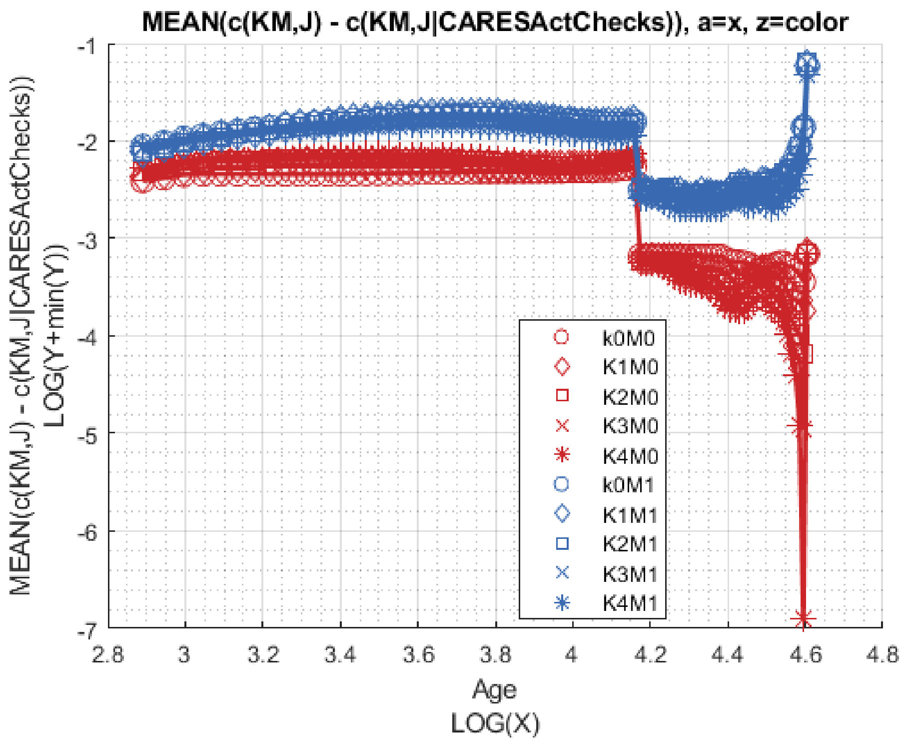




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(c(KM,J) - c(KM,J|CARESActChecks)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(c(KM,J) - c(KM,J|CARESActChecks))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





6.1.5 Analyze Education and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

```
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

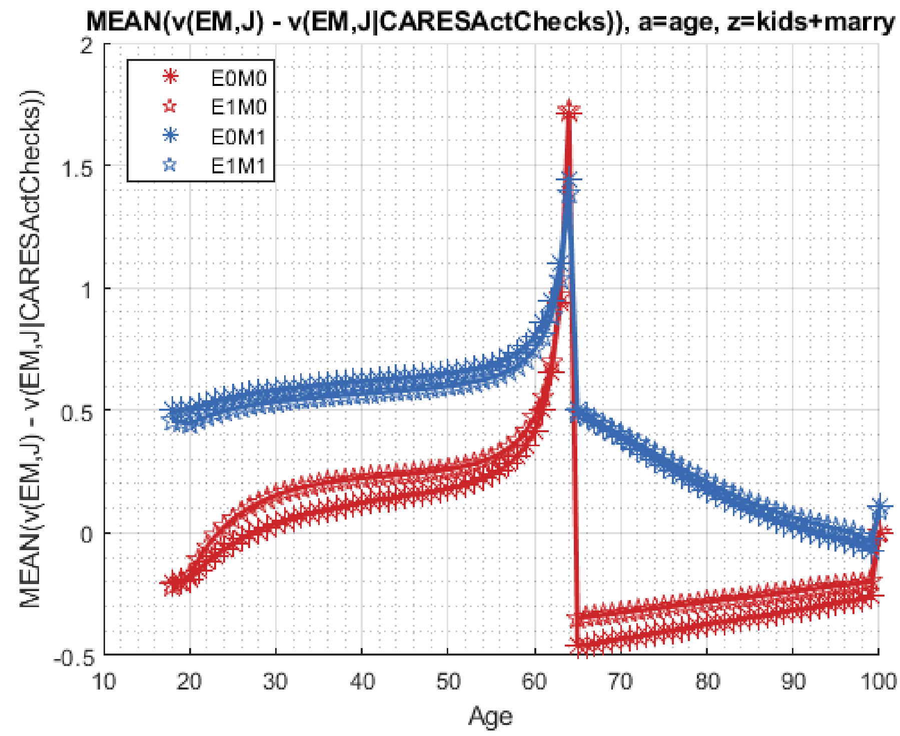
MEAN(v(EKM,J) - v(EKM,J|CARESActChecks)), MEAN(ap(EM,J) - ap(EM,J|CARESActChecks)), MEAN(c(EM,J) - c(EM,J|CARESActChecks))

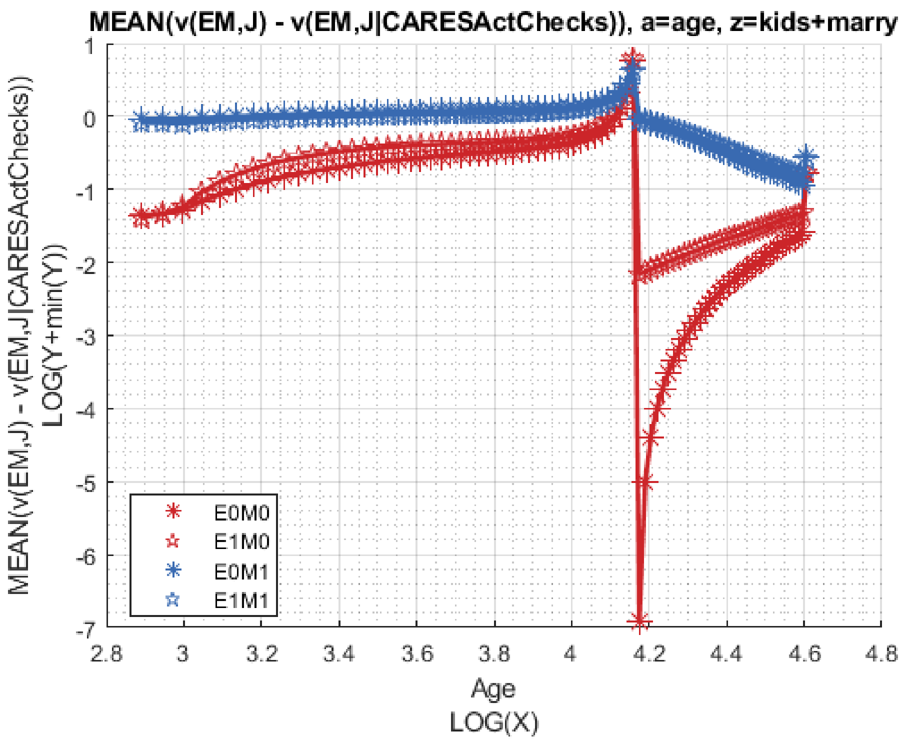
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(v(EM,J) - v(EM,J|CARESActChecks))", V_VFI_wthtrumpchk_drop, true, [

xxx  MEAN(v(EM,J) - v(EM,J|CARESActChecks))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0        -0.20644       -0.19866       -0.18731          -0.152       -0.12059
      2       1       0        -0.21487       -0.19821       -0.17411        -0.10932      -0.056018
      3       0       1         0.49975        0.50183        0.50278         0.51591        0.52735
      4       1       1         0.45707        0.45169        0.44517          0.4615        0.47518

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(ap(EM,J) - ap(EM,J|CARESActChecks))", ap_VFI_wthtrumpchk_drop, tru

xxx  MEAN(ap(EM,J) - ap(EM,J|CARESActChecks))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0         0.51372        0.51186        0.50965         0.53259        0.55496
      2       1       0         0.49288        0.48717         0.4809         0.53405        0.58772
      3       0       1         0.88501        0.91449        0.94426          1.0019         1.0594
      4       1       1         0.95477        0.98885         1.0231          1.1223         1.2227

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(c(EM,J) - c(EM,J|CARESActChecks))", cons_VFI_wthtrumpchk_drop, true

xxx  MEAN(c(EM,J) - c(EM,J|CARESActChecks))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0        0.044034       0.045896       0.048105        0.048949       0.049779
      2       1       0        0.064877       0.070583       0.076861        0.078708       0.079934
      3       0       1        0.068818       0.072249       0.075948        0.078738        0.08143
      4       1       1        0.095673        0.10195        0.10867         0.11365        0.11781
```

Graph Mean Values Change:

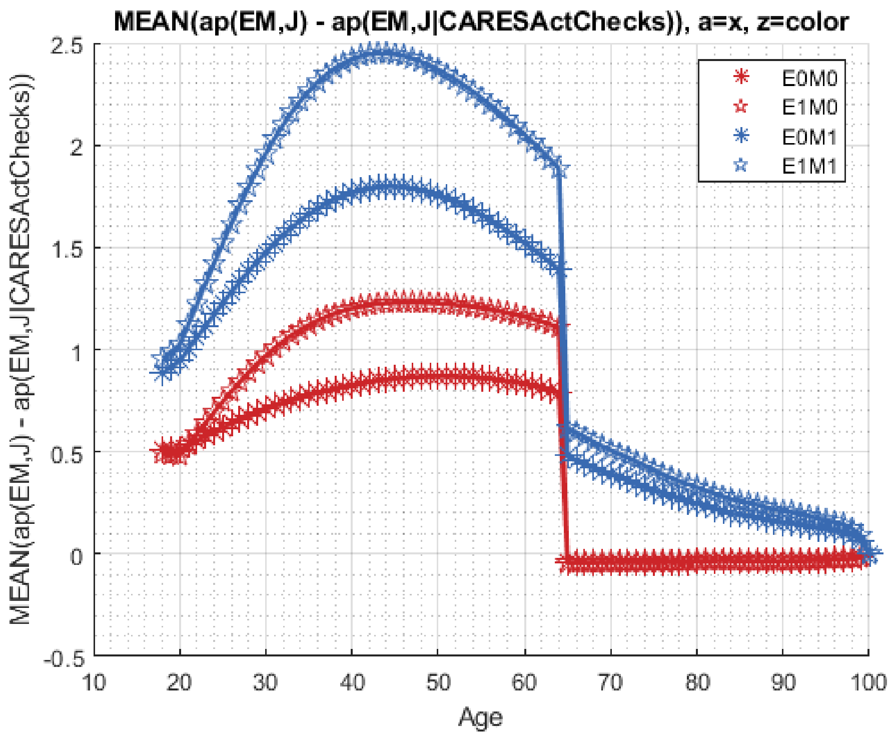
```
mp_support_graph('cl_st_graph_title') = {'MEAN(v(EM,J) - v(EM,J|CARESActChecks)), a=age, z=kids+marr
mp_support_graph('cl_st_ytitle') = {'MEAN(v(EM,J) - v(EM,J|CARESActChecks))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

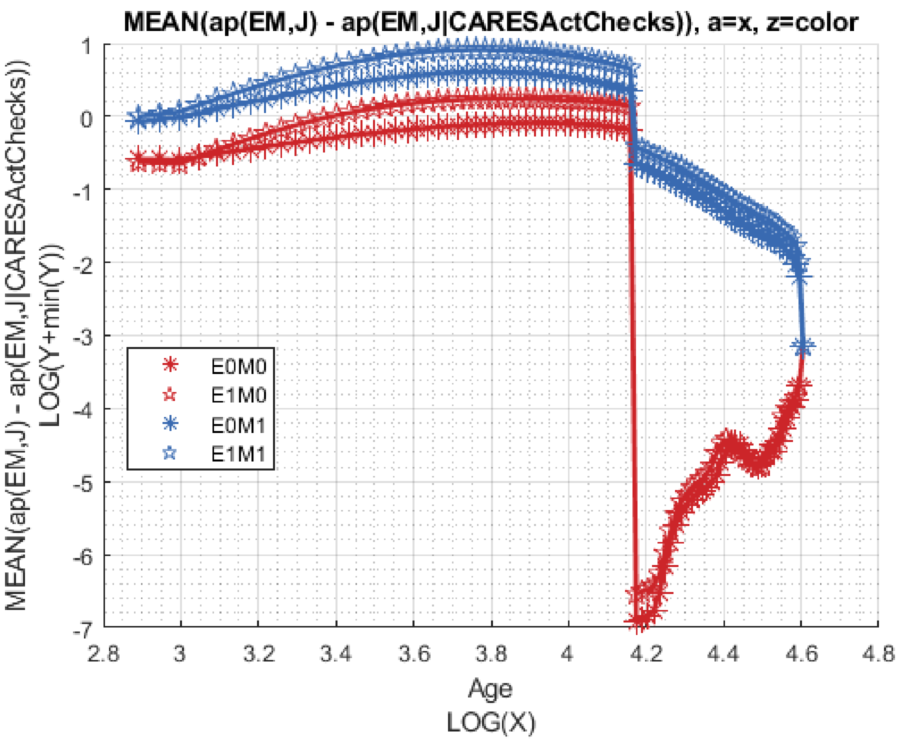




Graph Mean Savings Choices Change:

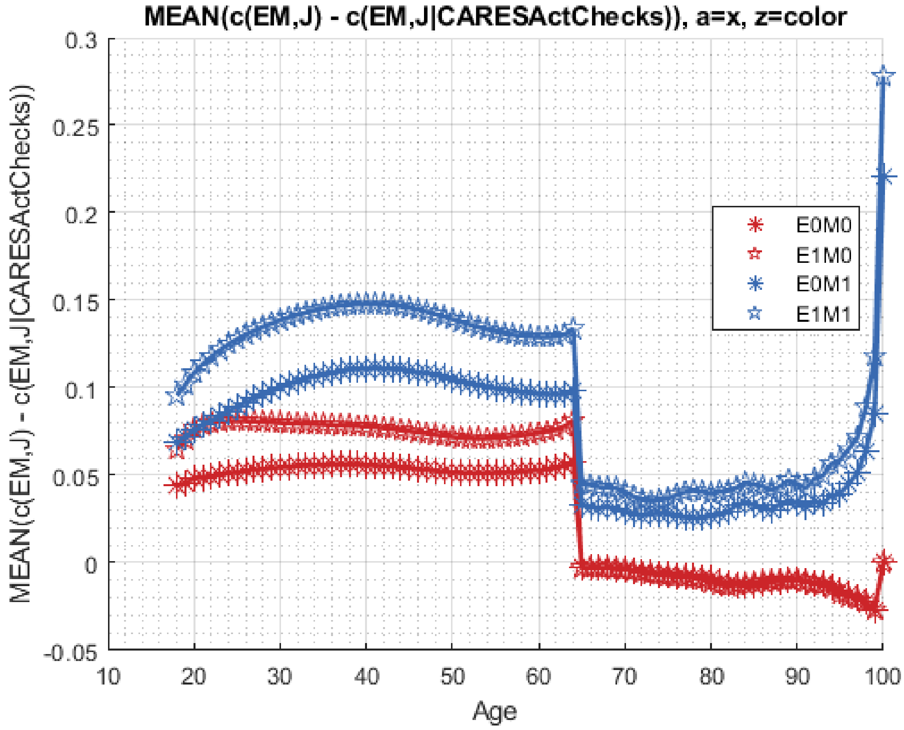
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(EM,J) - ap(EM,J|CARESActChecks)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(EM,J) - ap(EM,J|CARESActChecks))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

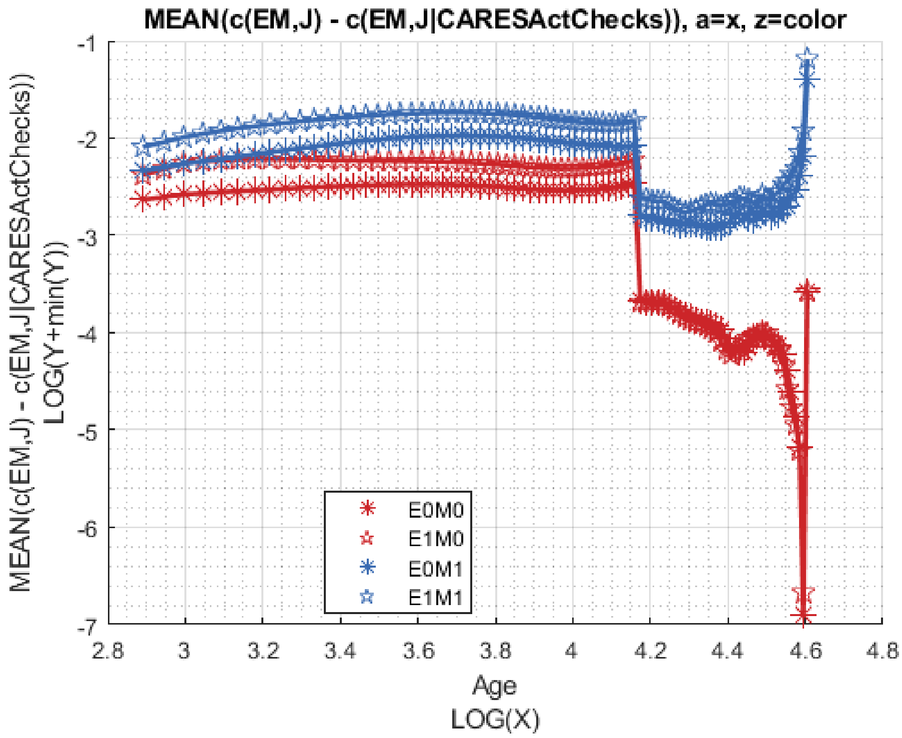




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(c(EM,J) - c(EM,J|CARESActChecks)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(c(EM,J) - c(EM,J|CARESActChecks))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```


MEAN(c(EM,J) - c(EM,J|CARESActChecks)), a=x, z=color
MEAN(c(EM,J) - c(EM,J|CARESActChecks))
Age
E0M0
E1M0
E0M1
E1M1
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
10
20
30
40
50
60
70
80
90
100


MEAN(c(EM,J) - c(EM,J|CARESActChecks)), a=x, z=color
MEAN(c(EM,J) - c(EM,J|CARESActChecks))
LOG(Y+min(Y))
Age
LOG(X)
E0M0
E1M0
E0M1
E1M1
-1
-2
-3
-4
-5
-6
-7
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
4.8

Chapter 7

Household Life Cycle Distribution

7.1 Distribution Grid Search

This is the example vignette for function: **snw_ds_main_grid_search** from the **PrjOptiSNW Package.** This function solves for vfi and gets distribution induced by policy functions and exogenous distributions. Grid Search for VFI and Grid Search also for Distribution. The results are illustrative of the differences between using grid search and exact solution. The grid search solution here is not fully vectorized but loops over the state-space.

7.1.1 Test SNW_DS_MAIN_GRID_SEARCH Defaults More Dense

Rather than solving for "docdense", this solves for "moredense", which has fewer shocks, in order to save time given the relatively slow speed of this algorithm.

```
mp_params = snw_mp_param('default_moredense');
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
[Phi_true,Phi_adj,A_agg,Y_inc_agg,it,mp_dsvfi_results] = snw_ds_main_grid_search(mp_params, mp_contr

Elapsed time is 13045.688258 seconds.
Completed SNW_VFI_MAIN_GRID_SEARCH;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test
Elapsed time is 13841.078692 seconds.
Completed SNW_DS_MAIN;SNW_MP_PARAM=;default_moredense;SNW_MP_CONTROL=;default_test

Phi_true = Phi_true/sum(Phi_true(:));
```

7.1.2 Show All Info in mp_dsvfi_results More Dense

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'))
```

	mean	unweighted_sum	sd	coefofvar	gini	min
a_ss	4.2955	5130.2	8.2965	1.9314	0.74079	0
ap_ss	33.948	11476	25.584	0.75362	0.43382	1
cons_ss	1.1795	1.585e+07	1.0182	0.86318	0.40848	0.035637
v_ss	-19.79	-1.1145e+07	35.654	-1.8016	-0.774	-868.79
n_ss	2.3554	21	1.4375	0.61029	0.3128	1
y_all	1.6272	2.3969e+07	1.8982	1.1665	0.50121	0.038108
y_head_inc	1.2682	5.6172e+05	1.5441	1.2175	0.50432	0.038108

```
    y_head_earn        1.0492          2628.2         1.4242       1.3574      0.60462            0
    y_spouse_inc      0.35895           55552        0.96039       2.6755      0.85307            0
    yshr_interest     0.11509      1.0971e+06        0.17681       1.5363      0.70728            0    0
    yshr_wage         0.78433      2.4004e+06        0.34004      0.43354      0.19505            0
    yshr_SS           0.10058           67295        0.23745       2.3608      0.91583            0
    yshr_tax          0.17694      7.7853e+05       0.040535      0.22909      0.13026     0.036506
    yshr_nttxss      0.076363      7.1123e+05        0.25868       3.3875       1.4024     -0.89715
```

7.1.3 More Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Probability mass matrixes, Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

7.1.4 Analyze Probability Mass Along Age Dimensions

Where are the mass at? Analyze mass given state space components.

```
% Get the Joint distribution over all states
% Define Graph Inputs
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = false; % do not log
```

Exogenous Permanent States Mass: Life Cycle, Edu and Marraige

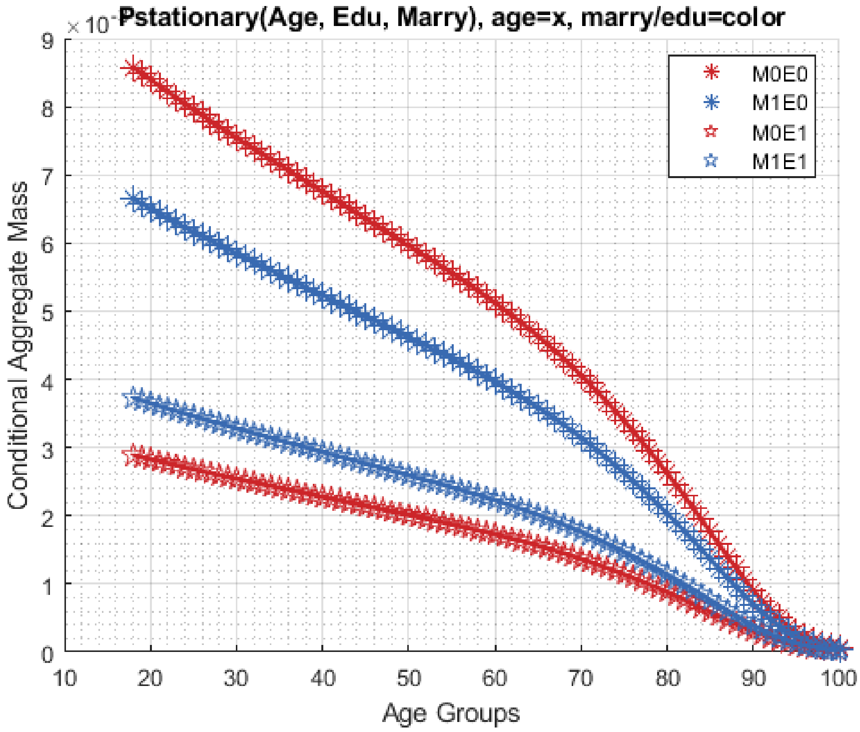
Tabulate value and policies along savings and shocks:

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,5,4];
% Value Function
tb_prob_aem = ff_summ_nd_array("P(Age, EDU, MARRY))", Phi_true, true, ["sum"], 3, 1, cl_mp_datasetde
```

```
xxx  P(Age, EDU, MARRY))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    marry    edu    sum_age_18    sum_age_19    sum_age_20    sum_age_21    sum_age_22    s
    _____    _____    ___    __________    __________    __________    __________    __________    _

      1        0       0     0.0085768     0.0084866     0.0083969     0.0083078     0.0082194     0
      2        1       0     0.0066438     0.0065739     0.0065044     0.0064354     0.0063669
      3        0       1     0.0028875     0.0028571      0.002827      0.002797     0.0027672     0
      4        1       1     0.0037292     0.0036899     0.0036509     0.0036122     0.0035738     0
```

```
mp_support_graph('cl_st_graph_title') = {'Pstationary(Age, Edu, Marry), age=x, marry/edu=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
```

```
ar_row_grid = ["M0E0", "M1E0", "M0E1", "M1E1"];
mp_support_graph('cl_st_xtitle') = {'Age Groups'};
mp_support_graph('cl_scatter_shapes') = {'*', '*', 'p', 'p' };
mp_support_graph('cl_colors') = {'red', 'blue', 'red', 'blue'};
ff_graph_grid((tb_prob_aem{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```



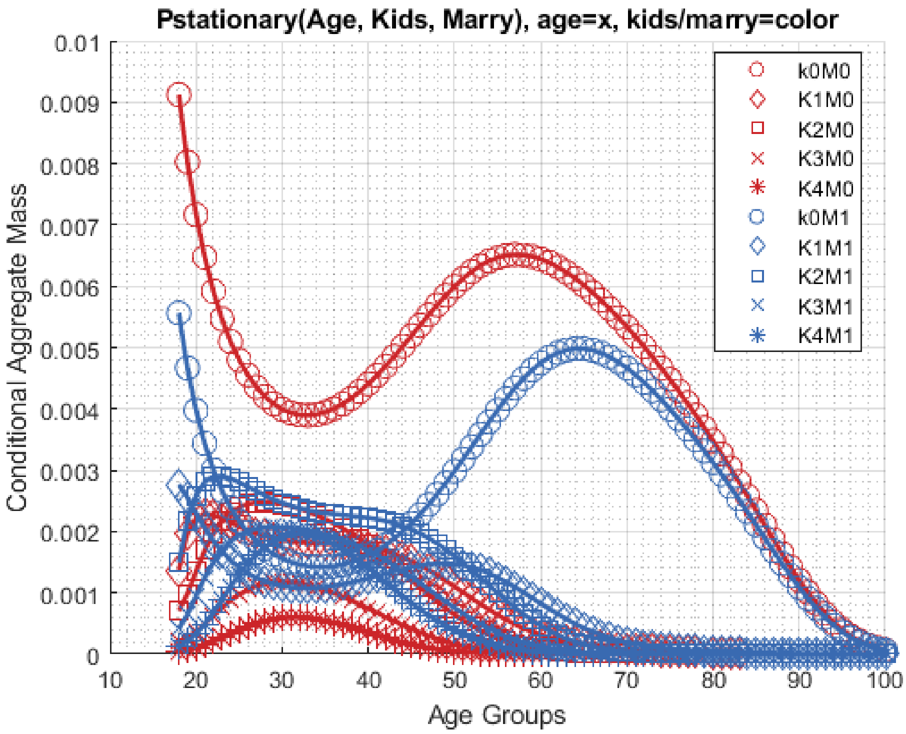
Kids and Marry By Age Mass

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_prob_amarrykids = ff_summ_nd_array("P(Age, Kids, Marry))", Phi_true, true, ["sum"], 3, 1, cl_mp_d
```

```
xxx  P(Age, Kids, Marry))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    sum_age_18    sum_age_19    sum_age_20    sum_age_21    sum_age_22
    _____    ____    _____    __________    __________    __________    __________    __________

      1        1       0       0.0091249     0.0080278     0.0071652     0.0064765     0.0059205
      2        2       0       0.0013699     0.0019743     0.0022187     0.0022858     0.0022687
      3        3       0      0.00071266    0.00098425     0.0013537     0.0016929     0.0019639
      4        4       0      0.00020622    0.00027865    0.00037326    0.00049476    0.00062818
      5        5       0      5.0761e-05    7.8715e-05      0.000113    0.00015485    0.00020534
      6        1       1       0.0055624     0.0046679     0.0039774     0.0034368     0.0030088
      7        2       1       0.0027682     0.0025539     0.0023005     0.0020611     0.0018525
      8        3       1       0.0014982     0.0021823     0.0025943     0.0028096      0.002896
      9        4       1      0.00041197    0.00064648    0.00095224     0.0012491     0.0015009
     10        5       1      0.00013221     0.0002132    0.00033097    0.00049097    0.00068255
```

```
mp_support_graph('cl_st_graph_title') = {'Pstationary(Age, Kids, Marry), age=x, kids/marry=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
```

```
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
mp_support_graph('cl_st_xtitle') = {'Age Groups'};
ff_graph_grid((tb_prob_amarrykids{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```



7.1.5 Analyze Probability Mass Asset and Shock Dimensions

Where are the mass at?

```
% Define Graph Inputs
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = false; % do not log
```

Asset and Shock Mass

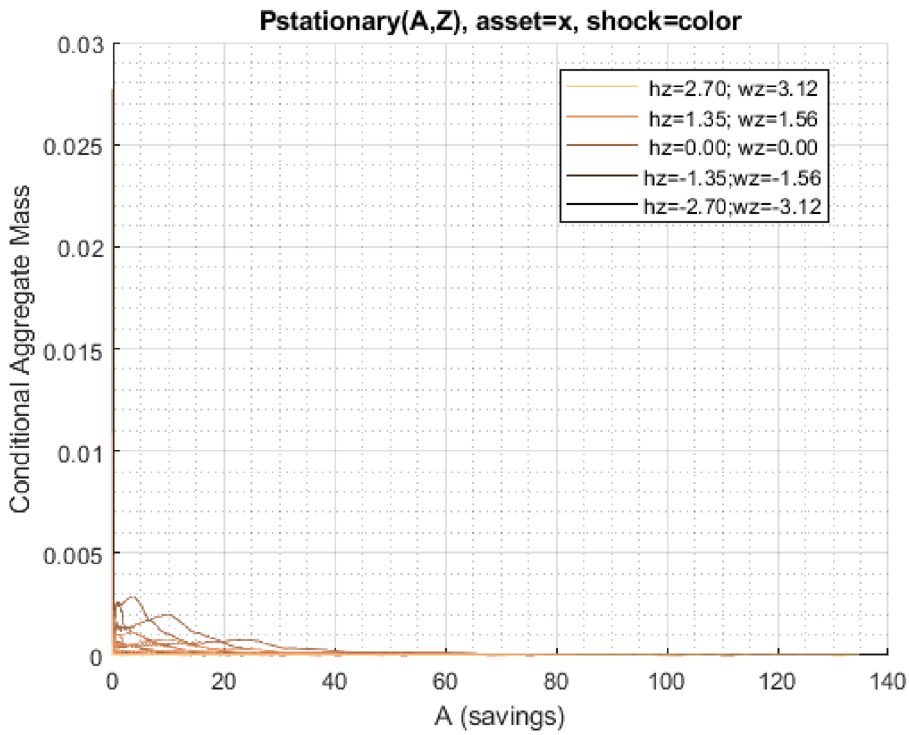
```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_prob_az = ff_summ_nd_array("P(A,Z))", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar_permut
```

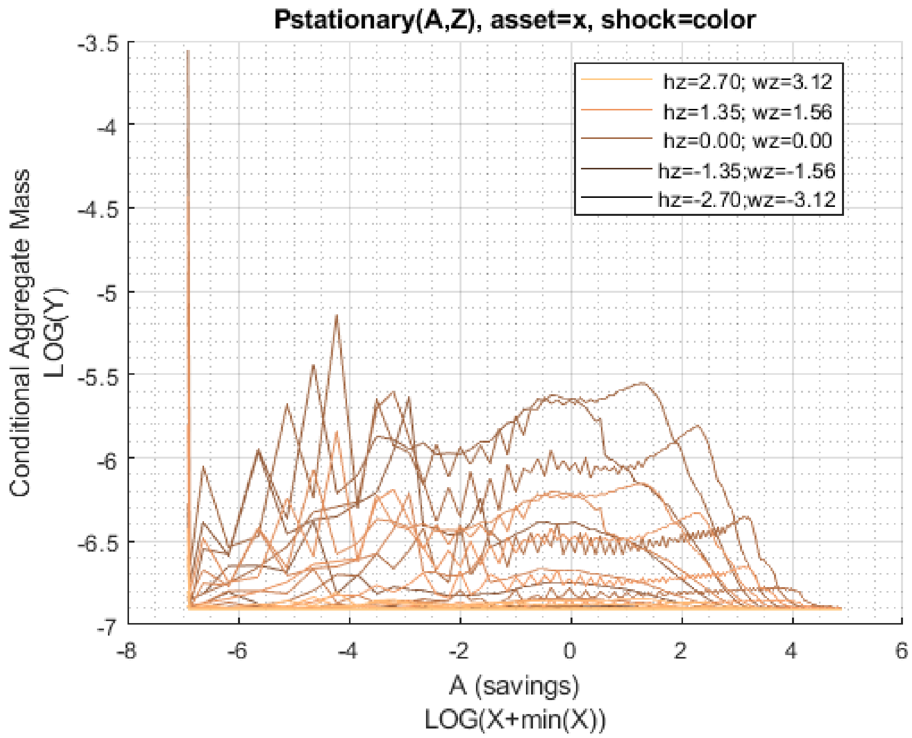
```
xxx  P(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    savings     sum_eta_1      sum_eta_2      sum_eta_3      sum_eta_4      sum_eta_5      sum_et
    -----    -------     ----------     ----------     ----------     ----------     ----------     ------

      1            0     1.7754e-05     0.00011362     0.00037708     0.00061913     0.00056244     0.0003
      2        4e-05     2.8763e-07     1.3442e-06     4.0195e-06     3.6795e-07     6.5205e-06     1.2822
      3      0.00032     8.5896e-07     2.1837e-06     1.5288e-05     9.7352e-06     3.0233e-05     2.2281
      4      0.00108     2.4168e-06     6.6439e-06     8.5514e-06      8.806e-06     1.0167e-05      3.836
      5      0.00256     7.7644e-07     6.7137e-06     3.5185e-05     3.6273e-05     1.2262e-05     6.4167
      6        0.005      1.615e-07     5.6988e-06     1.2805e-05     1.6079e-05     5.4165e-05     1.0241
      7      0.00864     2.9022e-07     1.5618e-05     1.5907e-05     7.5335e-05     2.1309e-05     1.7692
      8      0.01372     9.2058e-08     2.2512e-06     1.6629e-05     2.2447e-05     0.00010906     1.9013
```

9	0.02048	1.493e-07	5.139e-06	1.987e-05	2.7684e-05	1.8495e-05	2.0213
10	0.02916	2.2636e-07	2.1911e-06	2.2674e-05	4.0829e-05	5.6366e-05	5.2344
11	0.04	3.6847e-07	3.2622e-06	3.5384e-05	3.9524e-05	3.7428e-05	6.0069
12	0.05324	2.7682e-07	2.8958e-06	5.7484e-05	3.6034e-05	3.501e-05	4.0214
13	0.06912	3.353e-07	2.6767e-06	1.4572e-05	3.8273e-05	2.474e-05	2.563
14	0.08788	3.0394e-07	2.2742e-06	1.4027e-05	3.4255e-05	3.7729e-05	1.5417
15	0.10976	2.5858e-07	1.9422e-06	1.2587e-05	3.4742e-05	2.7058e-05	2.4059
16	0.135	2.7326e-07	1.9079e-06	1.2852e-05	3.438e-05	3.6783e-05	1.6552
17	0.16384	2.7552e-07	1.9749e-06	1.5066e-05	3.5948e-05	3.0858e-05	2.8702
18	0.19652	3.0673e-07	2.1238e-06	2.128e-05	3.877e-05	3.75e-05	1.7398
19	0.23328	3.2755e-07	2.1478e-06	1.0054e-05	3.8731e-05	3.1329e-05	2.6872
20	0.27436	3.5257e-07	2.2995e-06	1.0938e-05	4.2758e-05	4.3572e-05	1.8163
21	0.32	4.0641e-07	2.5518e-06	1.1635e-05	4.5919e-05	3.416e-05	3.1068
22	0.37044	4.4361e-07	2.8814e-06	1.2784e-05	4.688e-05	4.9116e-05	2.2826
23	0.42592	4.7158e-07	3.2535e-06	1.3383e-05	5.1616e-05	4.104e-05	2.9579
24	0.48668	5.1923e-07	3.4305e-06	1.432e-05	5.2283e-05	5.1981e-05	2.8417
25	0.55296	5.437e-07	3.6017e-06	1.5469e-05	5.6439e-05	5.2035e-05	3.2783
26	0.625	5.5347e-07	3.8425e-06	1.5479e-05	5.5965e-05	5.6994e-05	2.7605
27	0.70304	5.568e-07	3.7614e-06	1.5095e-05	5.8432e-05	5.4503e-05	3.5853
28	0.78732	5.7466e-07	3.7497e-06	1.502e-05	5.7903e-05	5.6109e-05	2.9314
29	0.87808	5.7563e-07	3.8005e-06	1.4892e-05	5.6949e-05	5.3853e-05	3.2364
30	0.97556	5.7132e-07	3.8705e-06	1.5095e-05	5.6906e-05	5.5758e-05	2.9176
31	1.08	5.726e-07	3.8645e-06	1.5349e-05	5.6413e-05	5.4339e-05	3.1568
32	1.1916	5.6625e-07	3.7872e-06	1.484e-05	5.448e-05	5.7108e-05	2.7934
33	1.3107	5.4238e-07	3.6971e-06	1.4269e-05	5.2724e-05	5.4906e-05	3.2967
34	1.4375	5.3278e-07	3.5724e-06	1.3626e-05	4.9663e-05	5.6325e-05	2.6723
35	1.5722	5.0996e-07	3.485e-06	1.3686e-05	4.6691e-05	5.5236e-05	3.069
36	1.715	5.0398e-07	3.4238e-06	1.3137e-05	4.7646e-05	5.6783e-05	2.8995
37	1.8662	4.767e-07	3.3615e-06	1.2844e-05	3.4086e-05	5.6434e-05	3.0261
38	2.0261	4.684e-07	3.2063e-06	1.2531e-05	3.2602e-05	5.8615e-05	2.8719
39	2.1949	4.5256e-07	3.1314e-06	1.2176e-05	3.2249e-05	5.8445e-05	3.0339
40	2.3728	4.2248e-07	3.093e-06	1.2003e-05	3.0763e-05	6.0571e-05	2.7865
41	2.56	4.0678e-07	2.958e-06	1.1808e-05	3.0172e-05	6.0706e-05	3.0549
42	2.7568	3.9214e-07	2.8541e-06	1.1601e-05	2.9569e-05	6.2065e-05	2.9269
43	2.9635	3.6841e-07	2.7289e-06	1.1074e-05	2.8387e-05	6.2402e-05	3.0802
44	3.1803	3.4717e-07	2.6124e-06	1.0683e-05	2.5913e-05	6.3791e-05	2.948
45	3.4074	3.2367e-07	2.453e-06	1.0352e-05	2.5365e-05	6.4051e-05	3.198
46	3.645	3.0387e-07	2.3261e-06	9.9363e-06	2.5054e-05	6.4312e-05	3.0105
47	3.8934	2.8153e-07	2.1577e-06	9.4355e-06	2.4465e-05	6.3912e-05	3.2461
48	4.1529	2.5915e-07	2.0525e-06	8.9454e-06	2.4289e-05	6.2333e-05	3.3681
49	4.4237	2.3743e-07	1.8857e-06	8.3087e-06	2.2777e-05	6.0276e-05	3.3448
50	4.706	2.1918e-07	1.7288e-06	7.9955e-06	2.2245e-05	5.8169e-05	3.4403
51	5	1.9952e-07	1.5877e-06	7.4469e-06	2.1203e-05	5.5849e-05	3.516
52	5.306	1.8097e-07	1.466e-06	6.9877e-06	2.0273e-05	5.2403e-05	3.5974
53	5.6243	1.6229e-07	1.3297e-06	6.4332e-06	1.9708e-05	4.7628e-05	3.6187
54	5.9551	1.4898e-07	1.2561e-06	6.0843e-06	1.9174e-05	4.4076e-05	3.7855
55	6.2986	1.3486e-07	1.128e-06	5.585e-06	1.771e-05	4.0303e-05	3.7916
56	6.655	1.2261e-07	1.0323e-06	5.2044e-06	1.6634e-05	3.8461e-05	3.8144
57	7.0246	1.0954e-07	9.419e-07	4.8168e-06	1.5965e-05	3.7804e-05	4.0109
58	7.4077	9.7875e-08	8.5816e-07	4.4452e-06	1.4933e-05	3.5109e-05	4.0919
59	7.8045	8.546e-08	7.7452e-07	4.0603e-06	1.4176e-05	3.2431e-05	4.1509
60	8.2152	7.3682e-08	6.8673e-07	3.7298e-06	1.328e-05	2.9425e-05	4.2677
61	8.64	6.477e-08	6.1848e-07	3.4011e-06	1.2396e-05	2.7206e-05	4.266
62	9.0792	5.531e-08	5.4066e-07	3.0693e-06	1.1543e-05	2.5521e-05	4.3977
63	9.5331	4.7541e-08	4.7103e-07	2.763e-06	1.0681e-05	2.4224e-05	4.4506
64	10.002	4.0248e-08	4.0902e-07	2.4495e-06	9.8848e-06	2.2907e-05	4.4746
65	10.486	3.4304e-08	3.5272e-07	2.1931e-06	9.3275e-06	2.2397e-05	4.3939
66	10.985	2.8818e-08	3.0078e-07	1.8744e-06	8.5509e-06	2.0402e-05	4.185

```
67       11.5     2.381e-08    2.5613e-07    1.6478e-06    7.8158e-06    1.8976e-05    4.0119
68     12.031    1.9337e-08    2.1722e-07    1.4123e-06    7.0946e-06    1.7273e-05    3.7509
69     12.577    1.5777e-08    1.8339e-07    1.2258e-06    6.3824e-06    1.6049e-05    3.4531
70      13.14    1.2851e-08    1.5121e-07    1.0425e-06    5.5804e-06    1.4547e-05    3.2353
71      13.72    1.0265e-08    1.2303e-07    8.8363e-07    4.9327e-06    1.3365e-05    2.8115
72     14.316    7.9662e-09    9.9864e-08    7.4205e-07    4.3605e-06    1.2054e-05    2.5938
73      14.93    6.2761e-09    8.0975e-08    6.2212e-07    3.8095e-06    1.0943e-05    2.5285
74     15.561    4.9435e-09    6.4741e-08    5.1199e-07    3.2737e-06    9.9099e-06     2.404
75     16.209    3.9203e-09    5.1535e-08    4.1842e-07    2.8741e-06    8.6393e-06    2.2512
76     16.875    3.0965e-09    4.1106e-08    3.4266e-07    2.4563e-06     7.837e-06    2.0909
77     17.559    2.4345e-09     3.245e-08    2.7852e-07    2.0728e-06    6.9943e-06    1.9281
78     18.261    1.8995e-09    2.5906e-08    2.2453e-07    1.8078e-06    6.1702e-06     1.838
79     18.982    1.4113e-09    2.0505e-08    1.7857e-07    1.4705e-06    5.4235e-06    1.6869
80     19.722    1.0589e-09    1.5945e-08    1.4273e-07    1.2186e-06    4.7175e-06    1.4272
81      20.48    7.4099e-10    1.2449e-08    1.1516e-07    1.0016e-06    4.0402e-06    1.2422
82     21.258    5.3869e-10    9.3255e-09    9.1737e-08    8.0834e-07     3.454e-06    1.0801
83     22.055    3.8509e-10    7.0555e-09     7.294e-08    6.5411e-07    2.9309e-06    9.4775
84     22.871    2.8297e-10    5.0721e-09    5.6197e-08    5.2093e-07    2.5225e-06    8.5883
85     23.708    2.0045e-10    3.6381e-09    4.3661e-08    4.1573e-07    2.1339e-06    7.5613
86     24.565    1.4573e-10    2.6214e-09    3.2964e-08    3.3317e-07    1.8413e-06    7.1167
87     25.442    1.0412e-10    1.9298e-09    2.4937e-08    2.6889e-07    1.5139e-06    6.1692
88      26.34    7.2935e-11    1.3869e-09    1.8357e-08    2.1245e-07    1.2556e-06    5.9791
```

```
mp_support_graph('cl_st_graph_title') = {'Pstationary(A,Z), asset=x, shock=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
mp_support_graph('cl_st_xtitle') = {'A (savings)'};
mp_support_graph('st_rowvar_name') = 'z=';
mp_support_graph('it_legend_select') = 5;
mp_support_graph('st_rounding') = '6.2f';
mp_support_graph('bl_graph_logy') = true;
mp_support_graph('cl_colors') = 'copper';
ff_graph_grid((tb_prob_az{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);% Consumption
```





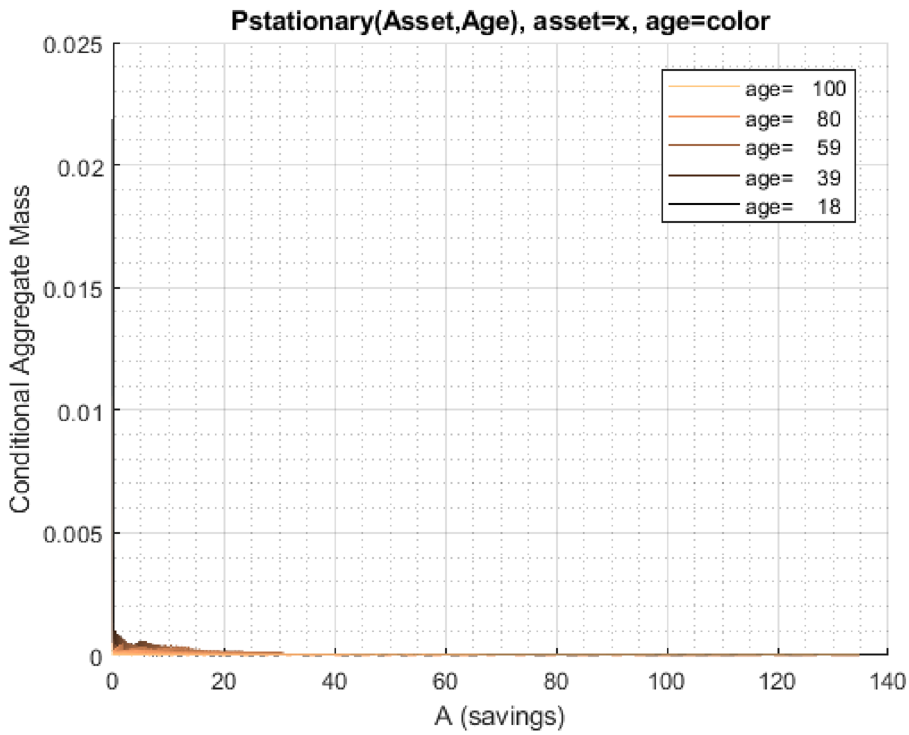
Asset Mass by Age

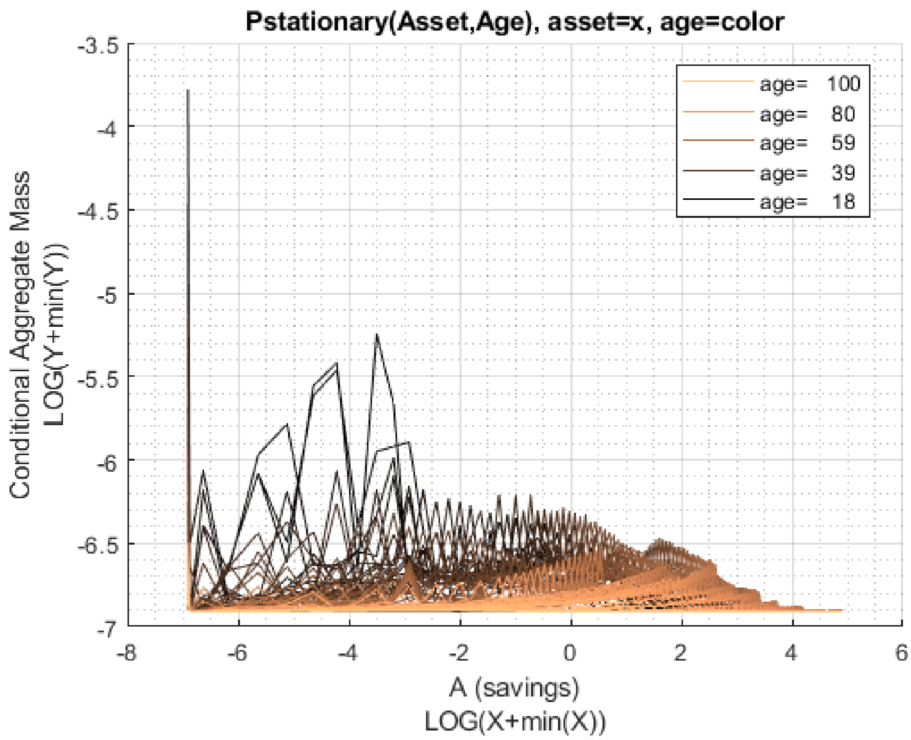
```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [3,4,5,6,1,2];
% Value Function
tb_prob_aage = ff_summ_nd_array("P(A,Z))", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar_perm
```

```
xxx  P(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    savings    sum_age_18    sum_age_19    sum_age_20    sum_age_21    sum_age_22    sum_ag
    -----    -------    ----------    ----------    ----------    ----------    ----------    ------

      1            0     0.021837      0.002388     0.0018389     0.0064409     0.0087881      0.01
      2        4e-05            0    2.3862e-06    2.8257e-06    1.5227e-05     0.0005064    7.2295
      3      0.00032            0     3.749e-05    3.8393e-05    0.00067452     0.0013202    3.5912
      4      0.00108            0    0.00031485     0.0003134    0.00027522    6.8839e-05    0.0001
      5      0.00256            0     0.0012853     0.0012851     0.0015569    8.2445e-05    6.9654
      6        0.005            0    0.00034215    0.00051426     0.0020793    0.00015903    0.0001
      7      0.00864            0     0.0028722     0.0026464    0.00031254    0.00033461    0.0001
      8      0.01372            0      0.003431      0.003249    0.00031772    0.00028917    0.0001
      9      0.02048            0    0.00028503    0.00067598    0.00043826    0.00042815    0.0003
     10      0.02916            0      0.004274     0.0016076    0.00077901    0.00039339    0.0003
     11         0.04            0     0.0024741     0.0016863     0.0015146     0.0012551    0.0003
     12      0.05324            0    0.00012193     0.0017565    0.00025805    0.00022615     0.001
     13      0.06912            0    0.00044563    0.00062939    0.00029172    0.00029416     0.000
     14      0.08788            0    2.7692e-05    0.00011258    0.00015714    0.00018719    0.0002
     15      0.10976            0    6.2377e-05    8.9179e-06    7.7238e-05    0.00018316    0.0001
     16        0.135            0    0.00067668    0.00016484    0.00010206    0.00022055    0.0002
     17      0.16384            0    5.8231e-06    5.0833e-05    0.00019203    0.00023346    0.0002
     18      0.19652            0    3.2338e-05    4.7739e-05    0.00021932    0.00024025    0.0002
     19      0.23328            0    2.7827e-05    0.00062964    0.00031839    0.00035773    0.0002
     20      0.27436            0    3.3098e-06    0.00010242    0.00072231     0.0003499    0.0003
     21         0.32            0    4.0326e-05    0.00030725    0.00040297    0.00025423    0.0003
     22      0.37044            0    0.00023294    0.00035244    0.00045546    0.00075563    0.0003
     23      0.42592            0    0.00029162    0.00046138    0.00031538    0.00034379     0.000
```

```
24     0.48668         0     0.0002901    0.00049108    0.00051896    0.00038724      0.000
25     0.55296         0    0.00034886    0.00054564     0.0003598    0.00041642     0.0004
26       0.625         0    0.00050916    0.00043448    0.00029056    0.00033958     0.0003
27     0.70304         0    0.00039586    0.00037749    0.00035324    0.00035324     0.0003
28     0.78732         0    0.00020681    0.00035156    0.00038514    0.00031557     0.0003
29     0.87808         0    1.4297e-05    5.5411e-05    0.00015561    0.00033422     0.0002
30     0.97556         0    1.5592e-05     6.288e-05    0.00029113    0.00022006     0.0002
31        1.08         0     2.009e-06    8.4115e-05    0.00014086    0.00019115     0.0002
32      1.1916         0    2.1045e-05    0.00010101    0.00015504    0.00012678     0.0001
33      1.3107         0    1.4435e-06    6.9572e-05    5.1203e-05    0.00020997     0.0001
34      1.4375         0    5.1689e-07     4.651e-05     7.846e-05     7.725e-05     0.0001
35      1.5722         0    4.7793e-07     4.484e-06    2.0179e-05     8.927e-05     0.0001
36       1.715         0    2.3446e-06    4.8601e-06     2.089e-05    4.4483e-05     0.0001
37      1.8662         0    2.6545e-07    5.0217e-06    2.8241e-05    3.3543e-05     5.1577
38      2.0261         0    5.4286e-07    3.5584e-06     1.983e-05    3.2623e-05     4.7562
39      2.1949         0    1.5332e-06    2.2585e-05    9.6032e-06    3.0291e-05      4.164
40      2.3728         0    4.1159e-06    1.2542e-05    1.5038e-05    2.1111e-05     4.7981
41        2.56         0    4.9992e-06    9.9161e-06    2.4244e-05    2.6564e-05     3.3403
42      2.7568         0    7.7981e-06    1.5369e-05    2.0348e-05    3.8876e-05     3.2557
43      2.9635         0    1.0694e-05    1.9867e-05    2.6661e-05    3.3652e-05     4.0382
44      3.1803         0    1.3309e-05    1.8776e-05    4.5294e-05    3.4882e-05     4.4248
45      3.4074         0    1.3226e-05    2.3002e-05    3.5636e-05    3.4395e-05     4.1287
46       3.645         0     3.533e-06    2.5709e-05    3.2868e-05    4.0509e-05     3.8809
47      3.8934         0    1.8503e-05    2.4946e-05    2.6065e-05    3.9174e-05     3.9923
```

```
mp_support_graph('cl_st_graph_title') = {'Pstationary(Asset,Age), asset=x, age=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
mp_support_graph('cl_st_xtitle') = {'A (savings)'};
mp_support_graph('st_rowvar_name') = 'age=';
mp_support_graph('it_legend_select') = 5;
mp_support_graph('st_rounding') = '6.0f';
mp_support_graph('bl_graph_logy') = true;
mp_support_graph('cl_colors') = 'copper';
ff_graph_grid((tb_prob_aage{1:end, 3:end})', age_grid, agrid, mp_support_graph);% Consumption Choice
```





7.1.6 Probability Statistics A, C and V Conditional on Ages

Where are the mass at?

```
ap_ss = mp_dsvfi_results('ap_ss');
c_ss = mp_dsvfi_results('cons_ss');
v_ss = mp_dsvfi_results('v_ss');
n_ss = mp_dsvfi_results('n_ss');

y_head_inc = mp_dsvfi_results('y_head_inc_ss');
y_spouse_inc = mp_dsvfi_results('y_spouse_inc_ss');

yshr_wage = mp_dsvfi_results('yshr_wage_ss');
yshr_SS = mp_dsvfi_results('yshr_SS_ss');
yshr_nttxss = mp_dsvfi_results('yshr_nttxss_ss');

for it_ctr=1:size(ap_ss, 1)
    if (ismember(it_ctr, round(linspace(1, size(ap_ss, 1), 3))))
        display(['age =' num2str(age_grid(it_ctr))]);

        % construct input data
        Phi_true_age = Phi_true(it_ctr, :, :, : ,: ,:);
        ap_ss_age = ap_ss(it_ctr, :, :, : ,: ,:);
        c_ss_age = c_ss(it_ctr, :, :, : ,: ,:);
        v_ss_age = v_ss(it_ctr, :, :, : ,: ,:);
        n_ss_age = n_ss(it_ctr, :, :, : ,: ,:);

        y_head_inc_age = y_head_inc(it_ctr, :, :, : ,: ,:);
        y_spouse_inc_age = y_spouse_inc(it_ctr, :, :, : ,: ,:);
        yshr_wage_age = yshr_wage(it_ctr, :, :, : ,: ,:);
        yshr_SS_age = yshr_SS(it_ctr, :, :, : ,: ,:);
        yshr_nttxss_age = yshr_nttxss(it_ctr, :, :, : ,: ,:);

        mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
```

```
        mp_cl_ar_xyz_of_s('ap_ss') = {ap_ss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('c_ss') = {c_ss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('v_ss') = {v_ss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('n_ss') = {n_ss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('y_head_inc') = {y_head_inc_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('y_spouse') = {y_spouse_inc_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('yshr_wage') = {yshr_wage_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('yshr_SS') = {yshr_SS_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('yshr_nttxss') = {yshr_nttxss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('ar_st_y_name') = ["ap_ss", "c_ss", "v_ss", "n_ss",...
            "y_head_inc", "y_spouse", "yshr_wage", "yshr_SS", "yshr_nttxss"];

        % controls
        mp_support = containers.Map('KeyType','char', 'ValueType','any');
        mp_support('ar_fl_percentiles') = [0.01 10 25 50 75 90 99.99];
        mp_support('bl_display_final') = true;
        mp_support('bl_display_detail') = false;
        mp_support('bl_display_drvm2outcomes') = false;
        mp_support('bl_display_drvstats') = false;
        mp_support('bl_display_drvm2covcor') = false;

        % Call Function
        mp_cl_mt_xyz_of_s = ff_simu_stats(Phi_true_age(:)/sum(Phi_true_age,'all'), mp_cl_ar_xyz_of_s
    end
end

age =18
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       ap_ss         c_ss          v_ss          n_ss        y_head_inc
    ______________________    __________    __________    ___________    ___________    ___________

    {'mean'              }        10.116       0.75737        -37.592         1.9854        0.84341
    {'unweighted_sum'    }         11476    2.4405e+05    -7.8955e+05             21         4422.1
    {'sd'                }        6.9537       0.67774         55.748         1.0848        0.90505
    {'coefofvar'         }       0.68742       0.89486         -1.483        0.54639         1.0731
    {'gini'              }       0.32034       0.41117       -0.64344          0.268        0.41353
    {'min'               }             1      0.035637        -868.79              1       0.038108
    {'max'               }           151        18.059         25.518              6         13.784
    {'pYis0'             }             0             0              0              0              0
    {'pYls0'             }             0             0         0.8166              0              0
    {'pYgr0'             }             1             1         0.1834              1              1
    {'pYisMINY'          }       0.11052     0.0014188     7.8342e-06        0.41786      0.0033703
    {'pYisMAXY'          }             0             0              0      0.0060544              0
    {'p0_01'             }             1      0.035637        -746.63              1       0.038108
    {'p10'               }             1       0.24578        -86.517              1        0.14676
    {'p25'               }             7        0.3161        -50.751              1        0.28802
    {'p50'               }             9       0.51551        -25.389              2        0.56523
    {'p75'               }            11       0.88958         -5.527              3         1.1092
    {'p90'               }            23        1.5797         6.0744              4         2.1768
    {'p99_99'            }            52        6.8857         23.692              6         8.3836
    {'fl_cov_ap_ss'      }        48.354        1.9167         116.57        0.29345         1.7747
    {'fl_cor_ap_ss'      }             1        0.4067         0.3007       0.038901        0.28199
    {'fl_cov_c_ss'       }        1.9167       0.45934         20.369       0.067217        0.59824
    {'fl_cor_c_ss'       }        0.4067             1         0.5391       0.091423         0.9753
    {'fl_cov_v_ss'       }        116.57        20.369         3107.8         2.9005         24.615
    {'fl_cor_v_ss'       }        0.3007        0.5391              1       0.047962        0.48787
    {'fl_cov_n_ss'       }       0.29345      0.067217         2.9005         1.1768      -1.236e-17
    {'fl_cor_n_ss'       }      0.038901      0.091423       0.047962              1    -1.2589e-17
```

```
    {'fl_cov_y_head_inc' }       1.7747       0.59824        24.615     -1.236e-17        0.81911
    {'fl_cor_y_head_inc' }      0.28199        0.9753       0.48787    -1.2589e-17              1
    {'fl_cov_y_spouse'   }       3.1074      0.081697        4.9476        0.13364       0.021751
    {'fl_cor_y_spouse'   }      0.77947       0.21026       0.15481        0.21488        0.04192
    {'fl_cov_yshr_wage'  }   3.7471e-30    2.4421e-31   -9.1828e-31     1.0754e-30     8.1847e-31
    {'fl_cor_yshr_wage'  }   4.0447e-16    2.7046e-16   -1.2364e-17     7.4411e-16      6.788e-16
    {'fl_cov_yshr_SS'    }            0             0             0              0              0
    {'fl_cor_yshr_SS'    }          NaN           NaN           NaN            NaN            NaN
    {'fl_cov_yshr_nttxss'}      0.16611      0.021334        1.8609      0.0077776       0.025219
    {'fl_cor_yshr_nttxss'}      0.58487       0.77071       0.81728        0.17554        0.68223
    {'fracByP0_01'       }     0.010925    6.6761e-05     0.0030452        0.21046     0.00015228
    {'fracByP10'         }     0.010925      0.050401       0.44014        0.21046       0.019229
    {'fracByP25'         }        0.148      0.072459       0.71161        0.21046       0.096342
    {'fracByP50'         }      0.28531       0.21889       0.94673        0.53024        0.29663
    {'fracByP75'         }      0.60536       0.47077        1.0363        0.77109        0.59361
    {'fracByP90'         }        0.758       0.70215        1.0323        0.92834        0.84502
    {'fracByP99_99'      }      0.99975       0.99993             1              1              1
age =59
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       ap_ss         c_ss          v_ss          n_ss        y_head_inc
    ______________________    __________    __________    ___________    __________    __________

    {'mean'              }       55.659         1.287       -12.919         1.7239         1.8545
    {'unweighted_sum'    }        11476    2.6894e+05   -1.1138e+05             21          13268
    {'sd'                }       23.095        1.0938        20.385        0.90777         2.0429
    {'coefofvar'         }      0.41494       0.84994       -1.5779        0.52659         1.1016
    {'gini'              }      0.22938       0.40011      -0.80515        0.23461        0.47957
    {'min'               }            1      0.055882       -235.34              1       0.059541
    {'max'               }          151         32.48        14.759              6          23.47
    {'pYis0'             }            0             0             0              0              0
    {'pYls0'             }            0             0       0.74277              0              0
    {'pYgr0'             }            1             1       0.25723              1              1
    {'pYisMINY'          }    0.0037896    2.9499e-05    3.9537e-07        0.48835     9.9096e-05
    {'pYisMAXY'          }   4.9199e-06    2.3292e-08             0      0.0036816     2.0186e-06
    {'p0_01'             }            1       0.05663       -137.64              1       0.059554
    {'p10'               }           28       0.31379       -41.113              1        0.39098
    {'p25'               }           41       0.59299       -18.867              1         0.6458
    {'p50'               }           55         1.065       -7.2226              2         1.1351
    {'p75'               }           70        1.6559       0.35778              2         2.1525
    {'p90'               }           85        2.4892         6.453              3           4.19
    {'p99_99'            }          146        15.179         14.69              6         22.847
    {'fl_cov_ap_ss'      }       533.38        21.832        417.21         2.9474         37.948
    {'fl_cor_ap_ss'      }            1       0.86423       0.88619        0.14059        0.80428
    {'fl_cov_c_ss'       }       21.832        1.1965        14.391        0.23796         2.0766
    {'fl_cor_c_ss'       }      0.86423             1       0.64539        0.23965        0.92925
    {'fl_cov_v_ss'       }       417.21        14.391        415.54         3.8082         23.854
    {'fl_cor_v_ss'       }      0.88619       0.64539             1         0.2058         0.5728
    {'fl_cov_n_ss'       }       2.9474       0.23796        3.8082        0.82404       0.062177
    {'fl_cor_n_ss'       }      0.14059       0.23965        0.2058              1       0.033527
    {'fl_cov_y_head_inc' }       37.948        2.0766        23.854       0.062177         4.1736
    {'fl_cor_y_head_inc' }      0.80428       0.92925        0.5728       0.033527              1
    {'fl_cov_y_spouse'   }       5.9801       0.27801        4.7175         0.2771         0.1726
    {'fl_cor_y_spouse'   }       0.2311       0.22684       0.20654        0.27244       0.075404
    {'fl_cov_yshr_wage'  }      -1.3386     -0.041425       -1.1118     -0.0063156      -0.054208
    {'fl_cor_yshr_wage'  }      -0.6493      -0.42426      -0.61101       -0.07794       -0.29725
    {'fl_cov_yshr_SS'    }            0             0             0              0              0
    {'fl_cor_yshr_SS'    }          NaN           NaN           NaN            NaN            NaN
    {'fl_cov_yshr_nttxss'}      0.76653      0.028255       0.72181      0.0084863       0.047728
```

```
    {'fl_cor_yshr_nttxss'}      0.88794        0.69107         0.94731          0.2501        0.62501
    {'fracByP0_01'       }    6.8085e-05     5.4514e-06       0.0013821        0.28329     4.1425e-06
    {'fracByP10'         }      0.031672       0.019903         0.46352         0.28329       0.013283
    {'fracByP25'         }       0.1219        0.075667         0.79865         0.28329       0.054393
    {'fracByP50'         }      0.34388        0.22765          1.0532          0.72028        0.1806
    {'fracByP75'         }      0.62522        0.48161          1.1105          0.72028       0.41873
    {'fracByP90'         }       0.8294        0.71297          1.0711          0.85389       0.65167
    {'fracByP99_99'      }       0.9998        0.99881               1                1       0.99935
age =100
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       ap_ss          c_ss          v_ss          n_ss        y_head_inc
    _____________________    ___________    __________    __________    __________    ___________

    {'mean'              }             1       0.33746       -3.2579        1.4797        0.23579
    {'unweighted_sum'    }             1    2.8049e+05        789.51            21          483.8
    {'sd'                }     1.088e-14       0.23923        1.2254       0.50567       0.022052
    {'coefofvar'         }     1.088e-14       0.70891      -0.37615       0.34173       0.093527
    {'gini'              }             0       0.29996      -0.20031       0.12034       0.044484
    {'min'               }             1       0.19737       -11.197             1           0.22
    {'max'               }             1        141.61       0.99282             6          5.666
    {'pYis0'             }             0             0             0             0              0
    {'pYls0'             }             0             0       0.99204             0              0
    {'pYgr0'             }             1             1      0.007965             1              1
    {'pYisMINY'          }             1       0.34474    1.4552e-10        0.5232        0.48869
    {'pYisMAXY'          }             1             0             0    4.2206e-08              0
    {'p0_01'             }             1       0.19737        -7.038             1           0.22
    {'p10'               }             1       0.19737       -4.0665             1           0.22
    {'p25'               }             1       0.19737       -4.0665             1           0.22
    {'p50'               }             1       0.23607       -3.7707             1         0.2202
    {'p75'               }             1       0.36676       -2.6758             2          0.266
    {'p90'               }             1       0.59408       -1.2803             2        0.26717
    {'p99_99'            }             1        2.9028       0.51281             4        0.31843
    {'fl_cov_ap_ss'      }    1.1838e-28    4.4139e-31    3.5754e-30     4.121e-29     2.8489e-30
    {'fl_cor_ap_ss'      }             1    1.6958e-16    2.6816e-16    7.4904e-15     1.1874e-14
    {'fl_cov_c_ss'       }    4.4139e-31      0.057229       0.23842      0.059118      0.0016668
    {'fl_cor_c_ss'       }    1.6958e-16             1       0.81327       0.48871        0.31595
    {'fl_cov_v_ss'       }    3.5754e-30       0.23842        1.5017       0.20689       0.012148
    {'fl_cor_v_ss'       }    2.6816e-16       0.81327             1       0.33387        0.44951
    {'fl_cov_n_ss'       }     4.121e-29      0.059118       0.20689        0.2557      0.0018516
    {'fl_cor_n_ss'       }    7.4904e-15       0.48871       0.33387             1        0.16604
    {'fl_cov_y_head_inc' }    2.8489e-30     0.0016668      0.012148     0.0018516      0.0004863
    {'fl_cor_y_head_inc' }    1.1874e-14       0.31595       0.44951       0.16604              1
    {'fl_cov_y_spouse'   }   -3.2619e-31      0.050594       0.17973      0.052581     0.00064389
    {'fl_cor_y_spouse'   }   -1.2267e-16       0.86534       0.60012       0.42546        0.11947
    {'fl_cov_yshr_wage'  }   -6.2277e-32      0.040102       0.18489      0.087766     0.00067273
    {'fl_cor_yshr_wage'  }   -2.5076e-17       0.73439       0.66096       0.76038        0.13365
    {'fl_cov_yshr_SS'    }    -3.036e-30     -0.041567      -0.19392     -0.089099    -0.00074396
    {'fl_cor_yshr_SS'    }   -1.2162e-15      -0.75734      -0.68973        -0.768       -0.14704
    {'fl_cov_yshr_nttxss'}   -3.8383e-30      0.045721       0.21331      0.096069     0.00089733
    {'fl_cor_yshr_nttxss'}   -1.4131e-15       0.76558       0.69726       0.76104          0.163
    {'fracByP0_01'       }             1       0.20164    0.00049502       0.35357        0.45597
    {'fracByP10'         }             1       0.20164        0.5347       0.35357        0.45597
    {'fracByP25'         }             1       0.20164        0.5347       0.35357        0.45597
    {'fracByP50'         }             1       0.31034       0.64676       0.35357        0.46775
    {'fracByP75'         }             1       0.52813       0.89014       0.99419        0.87014
    {'fracByP90'         }             1       0.73972       0.97784       0.99419        0.88578
    {'fracByP99_99'      }             1        0.9992             1       0.99999        0.99992
```

7.2 Distribution Exact Savings Choices

This is the example vignette for function: **snw_ds_main** from the **PrjOptiSNW Package.** This function solves for vfi and gets distribution induced by policy functions and exogenous distributions. Looped to get distribution, but uses bisect vec for VFI.

7.2.1 Test SNW_DS_MAIN Defaults

Call the function with testing defaults.

```
mp_params = snw_mp_param('default_docdense');
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
[Phi_true,Phi_adj,A_agg,Y_inc_agg,it,mp_dsvfi_results] = snw_ds_main(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=551.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN           sum          mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -8.6673e+08    -19.834    28.17
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.4164e+09     32.412      36.
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05      2.131e+08     4.8764    8.326

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
              c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    --------

    r1     -376.05    -375.66    -373.17     -367.4    -358.05        -6.68      -6.5297      -6.379
    r2      -363.8    -363.41    -360.93    -355.25    -346.25      -6.4892      -6.3437      -6.197
    r3     -351.75    -351.36     -348.9    -343.44     -334.9      -6.2948      -6.1538      -6.011
    r4     -339.81    -339.45    -337.16    -332.06    -324.04       -6.095      -5.9584        -5.8
    r5     -328.99    -328.65    -326.51    -321.72    -314.17      -5.9054      -5.7725      -5.637
    r79    -14.033     -14.02    -13.926    -13.689    -13.287     -0.22848     -0.21775     -0.2076
    r80    -12.564     -12.55    -12.457     -12.22    -11.818     -0.17427     -0.16611     -0.1584
    r81    -10.778    -10.764    -10.671    -10.434    -10.032     -0.11927     -0.11368     -0.1084
    r82    -8.4226    -8.4089    -8.3155    -8.0786    -7.6766     -0.06597     -0.06284    -0.05992
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.020968    -0.019972    -0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2        c3            c4             c5         c526496    c526497    c526498    c52
           --    --    ----------    ---------    ----------    -------    -------    -------    ---

    r1     0     0      0.0005656    0.0075134      0.022901     114.76     120.42     126.29     132
    r2     0     0     0.00051498    0.0065334      0.021549     114.87     120.54     126.42     132
    r3     0     0     0.00051498    0.0049294      0.019875     114.98     120.67     126.57     132
    r4     0     0     0.00051498    0.0047937      0.019672     115.74     121.44     127.36     133
    r5     0     0     0.00048517    0.0046683      0.019484     116.51     122.22     128.16     134
    r79    0     0              0            0    0.00051498     81.091      85.68     90.325     94.
    r80    0     0              0            0             0     76.669      80.55     84.292     88.
    r81    0     0              0            0             0     68.313      71.52     74.459     77.
    r82    0     0              0            0             0     50.126     53.467     56.953     58.
    r83    0     0              0            0             0          0          0          0
```

```
xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
            c1          c2          c3          c4          c5         c526496    c526497    c526498
         --------    --------    --------    --------    --------    -------    -------    -------

    r1   0.036717    0.037251    0.040426     0.04363    0.048012    9.6396     9.8066     9.9533
    r2   0.036717    0.037251    0.040477     0.04461    0.049364    9.8014     9.9571     10.088
    r3   0.036717    0.037251    0.040477    0.046214    0.051039    9.9664     10.108      10.22
    r4   0.038144    0.038678    0.041903    0.047776    0.052666    10.118     10.244     10.339
    r5   0.039534    0.040068    0.043323     0.04929    0.054241    10.258     10.369     10.446
    r79   0.19737     0.19791     0.20163     0.21175     0.23093    35.811     37.046     38.418
    r80   0.19737     0.19791     0.20163     0.21175     0.23145    40.207      42.15     44.426
    r81   0.19737     0.19791     0.20163     0.21175     0.23145    48.541     51.158     54.236
    r82   0.19737     0.19791     0.20163     0.21175     0.23145     66.71     69.193     71.724
    r83   0.19737     0.19791     0.20163     0.21175     0.23145    116.82     122.65     128.66

Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=2153.0961
```

```
% [Phi_true,Phi_adj] = snw_ds_main(mp_params, mp_controls);
Phi_true = Phi_true/sum(Phi_true(:));
```

7.2.2 Show All Info in mp_dsvfi_results More Dense

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'))
```

```
                       mean      unweighted_sum       sd        coefofvar      gini         min
                     --------    --------------    --------    ---------    --------    --------

    a_ss              4.3602            2228        6.8796       1.5778       0.6755           0
    ap_ss             4.4621      5.3216e+08        6.9169       1.5501      0.67638           0
    cons_ss           1.0635      5.0787e+07        0.6938      0.65237      0.33936    0.036717
    v_ss             -36.615     -4.0773e+08         24.55     -0.67049     -0.33945     -615.77
    n_ss              2.3554              21        1.4375      0.61029       0.3128           1
    y_all             1.4189       8.353e+07        1.4929       1.0521      0.47667           0
    y_head_inc        1.1081      1.9253e+06         1.013      0.91419      0.42164    0.038108
    y_head_earn      0.88655           19732       0.92804       1.0468      0.53121           0
    y_spouse_inc     0.35797       4.827e+05       0.95437       2.6661      0.85269           0
    yshr_interest    0.12865      3.8438e+06       0.17577       1.3663      0.65781           0
    yshr_wage        0.77402      8.8881e+06       0.33679      0.43512       0.2062           0
    yshr_SS         0.097329           29012        0.2266       2.3282      0.91382           0
    yshr_tax         0.17833      2.8338e+06      0.035661      0.19998      0.11386    0.036506
    yshr_nttxss     0.080996      2.8048e+06       0.24691       3.0485       1.2592    -0.89715
```

7.2.3 More Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Probability mass matrixes, Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
```

```
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

7.2.4 Analyze Probability Mass Along Age Dimensions

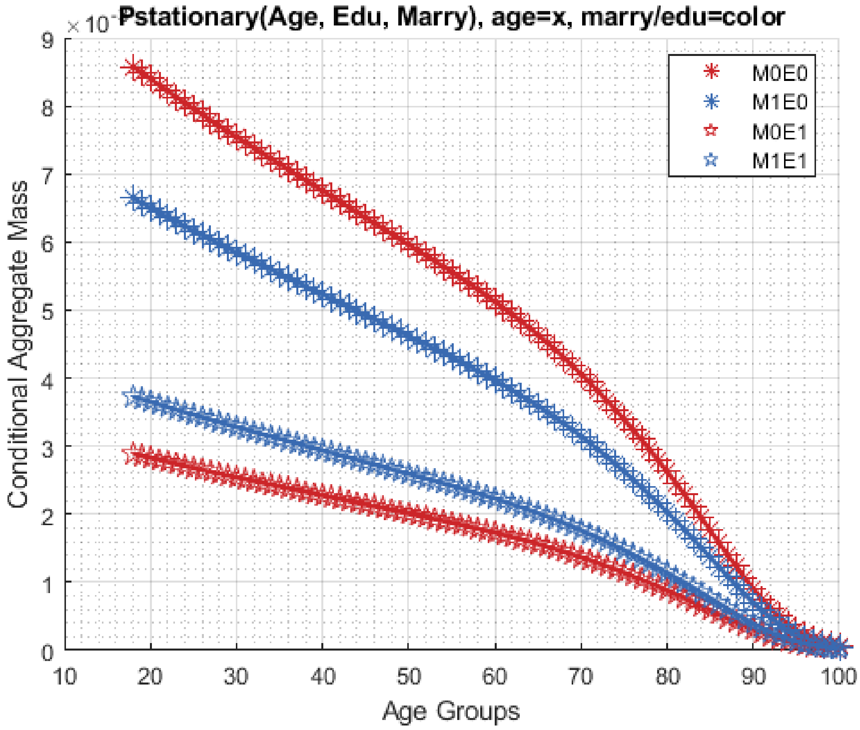
Where are the mass at? Analyze mass given state space components.

```
% Get the Joint distribution over all states
% Define Graph Inputs
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = false; % do not log
```

Exogenous Permanent States Mass: Life Cycle, Edu and Marraige

Tabulate value and policies along savings and shocks:

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,5,4];
% Value Function
tb_prob_aem = ff_summ_nd_array("P(Age, EDU, MARRY))", Phi_true, true, ["sum"], 3, 1, cl_mp_datasetde

xxx  P(Age, EDU, MARRY))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    marry    edu    sum_age_18    sum_age_19    sum_age_20    sum_age_21    sum_age_22    s
    -----    -----    ---    ----------    ----------    ----------    ----------    ----------    -

      1        0       0     0.0085768     0.0084866     0.0083969     0.0083078     0.0082194     0
      2        1       0     0.0066438     0.0065739     0.0065044     0.0064354     0.0063669
      3        0       1     0.0028875     0.0028571      0.002827      0.002797     0.0027672     0
      4        1       1     0.0037292     0.0036899     0.0036509     0.0036122     0.0035738     0

mp_support_graph('cl_st_graph_title') = {'Pstationary(Age, Edu, Marry), age=x, marry/edu=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
ar_row_grid = ["M0E0", "M1E0", "M0E1", "M1E1"];
mp_support_graph('cl_st_xtitle') = {'Age Groups'};
mp_support_graph('cl_scatter_shapes') = {'*', '*', 'p', 'p' };
mp_support_graph('cl_colors') = {'red', 'blue', 'red', 'blue'};
ff_graph_grid((tb_prob_aem{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```



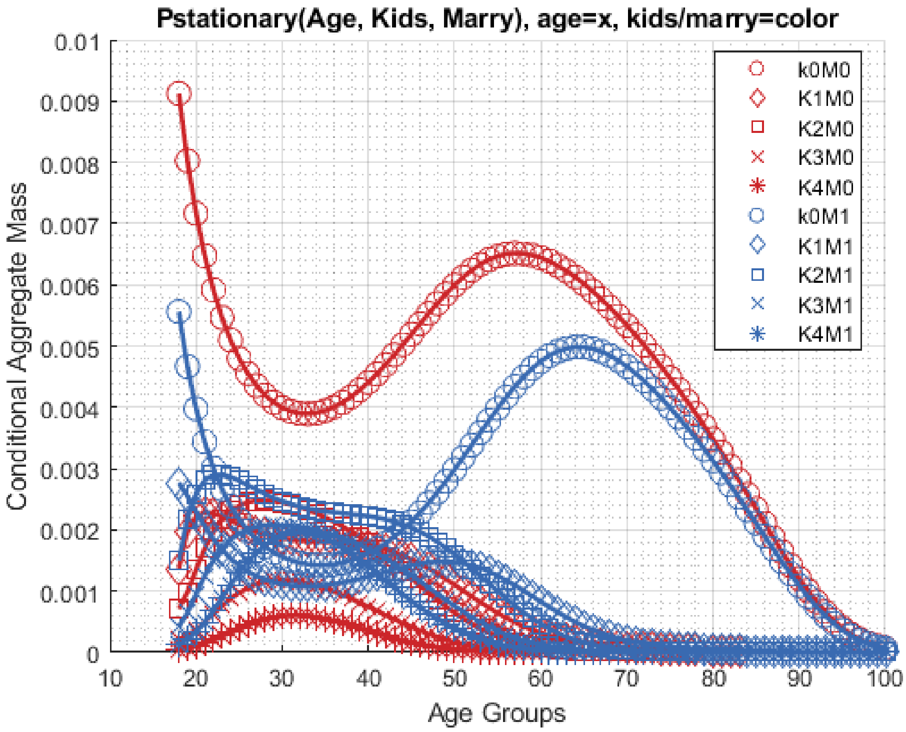
Kids and Marry By Age Mass

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_prob_amarrykids = ff_summ_nd_array("P(Age, Kids, Marry))", Phi_true, true, ["sum"], 3, 1, cl_mp_d
```

```
xxx  P(Age, Kids, Marry))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry     sum_age_18     sum_age_19     sum_age_20     sum_age_21     sum_age_22
    -----    ----    -----    ----------     ----------     ----------     ----------     ----------

      1        1       0        0.0091249      0.0080278      0.0071652      0.0064765      0.0059205
      2        2       0        0.0013699      0.0019743      0.0022187      0.0022858      0.0022687
      3        3       0       0.00071266     0.00098425      0.0013537      0.0016929      0.0019639
      4        4       0       0.00020622     0.00027865     0.00037326     0.00049476     0.00062818
      5        5       0       5.0761e-05     7.8715e-05       0.000113     0.00015485     0.00020534
      6        1       1        0.0055624      0.0046679      0.0039774      0.0034368      0.0030088
      7        2       1        0.0027682      0.0025539      0.0023005      0.0020611      0.0018525
      8        3       1        0.0014982      0.0021823      0.0025943      0.0028096       0.002896
      9        4       1       0.00041197     0.00064648     0.00095224      0.0012491      0.0015009
     10        5       1       0.00013221      0.0002132     0.00033097     0.00049097     0.00068255
```

```
mp_support_graph('cl_st_graph_title') = {'Pstationary(Age, Kids, Marry), age=x, kids/marry=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
mp_support_graph('cl_st_xtitle') = {'Age Groups'};
```

```
ff_graph_grid((tb_prob_amarrykids{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```



7.2.5 Analyze Probability Mass Asset and Shock Dimensions

Where are the mass at?

```
% Define Graph Inputs
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = false; % do not log
```

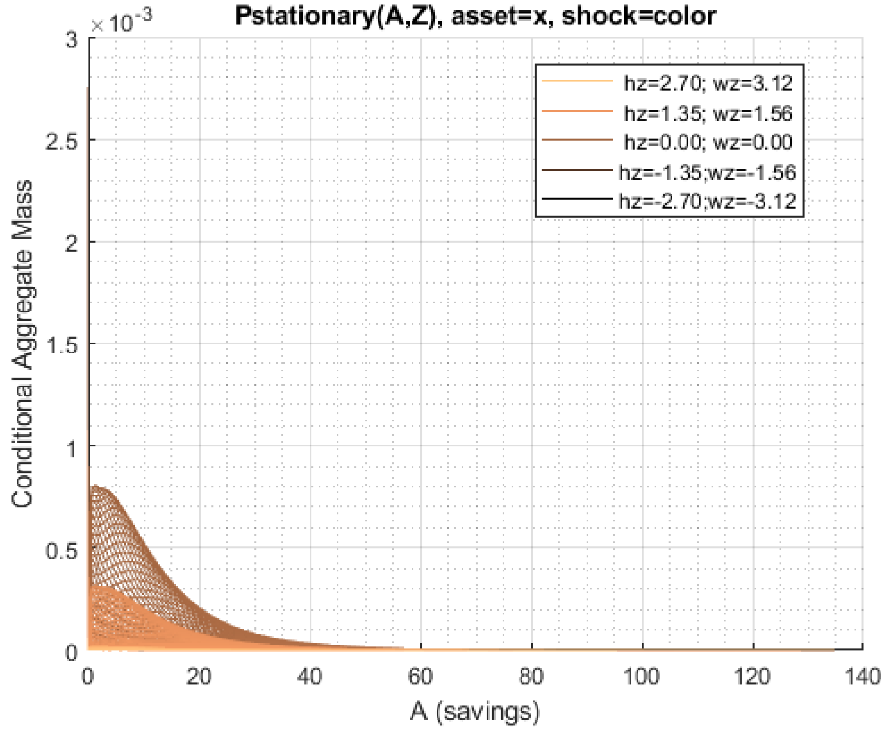
Asset and Shock Mass

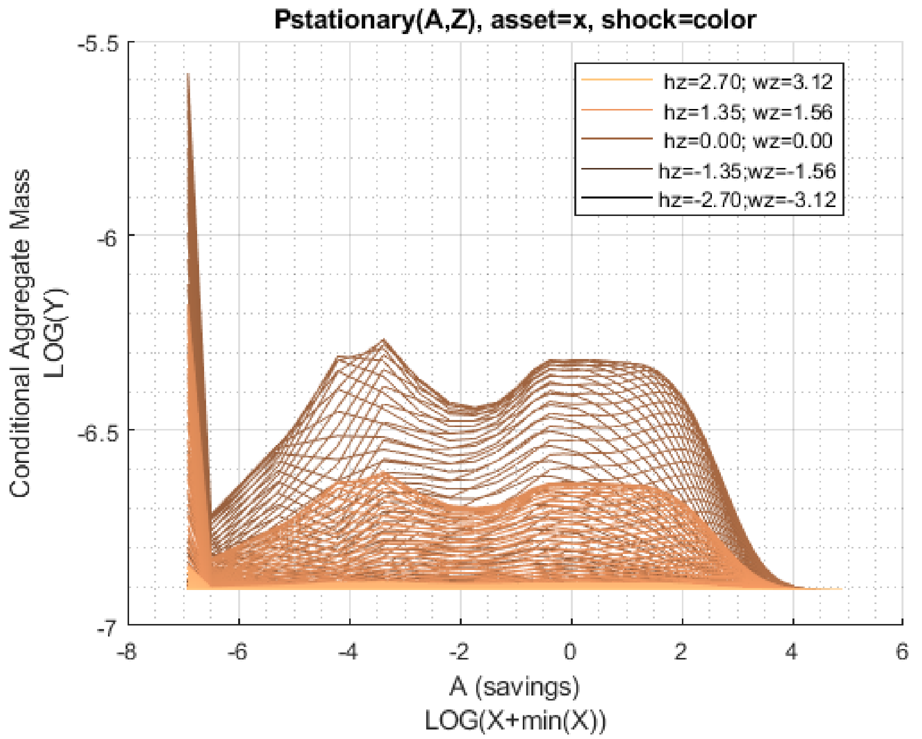
```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_prob_az = ff_summ_nd_array("P(A,Z))", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar_permut
```

```
xxx  P(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      sum_eta_1     sum_eta_2     sum_eta_3     sum_eta_4     sum_eta_5     sum
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1               0    1.6729e-07    1.4316e-07    2.1759e-07    3.1665e-07     4.509e-07    6.3
      2      0.00051498    3.7493e-10    3.6049e-10    6.2649e-10    1.1273e-09    2.1799e-09    4.4
      3       0.0041199    7.5745e-10    6.6694e-10    1.0474e-09    1.5976e-09    2.4182e-09    3.6
      4        0.013905    1.6314e-09    1.4169e-09    2.1927e-09    3.2778e-09    4.8429e-09    7.1
      5        0.032959    5.5034e-09    4.7405e-09     7.269e-09    1.0722e-08    1.5557e-08    2.2
      6        0.064373    6.5761e-09    5.6858e-09     8.729e-09    1.2871e-08    1.8634e-08    2.6
```

```
mp_support_graph('cl_st_graph_title') = {'Pstationary(A,Z), asset=x, shock=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
mp_support_graph('cl_st_xtitle') = {'A (savings)'};
mp_support_graph('st_rowvar_name') = 'z=';
mp_support_graph('it_legend_select') = 5;
mp_support_graph('st_rounding') = '6.2f';
```

```
mp_support_graph('bl_graph_logy') = true;
mp_support_graph('cl_colors') = 'copper';
ff_graph_grid((tb_prob_az{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);% Consumption
```



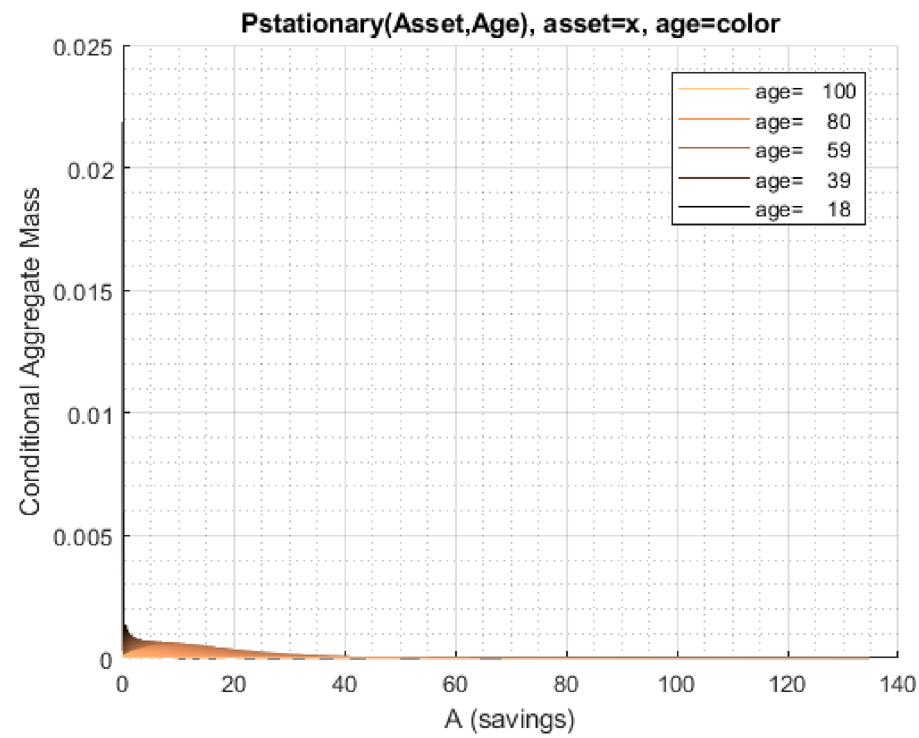


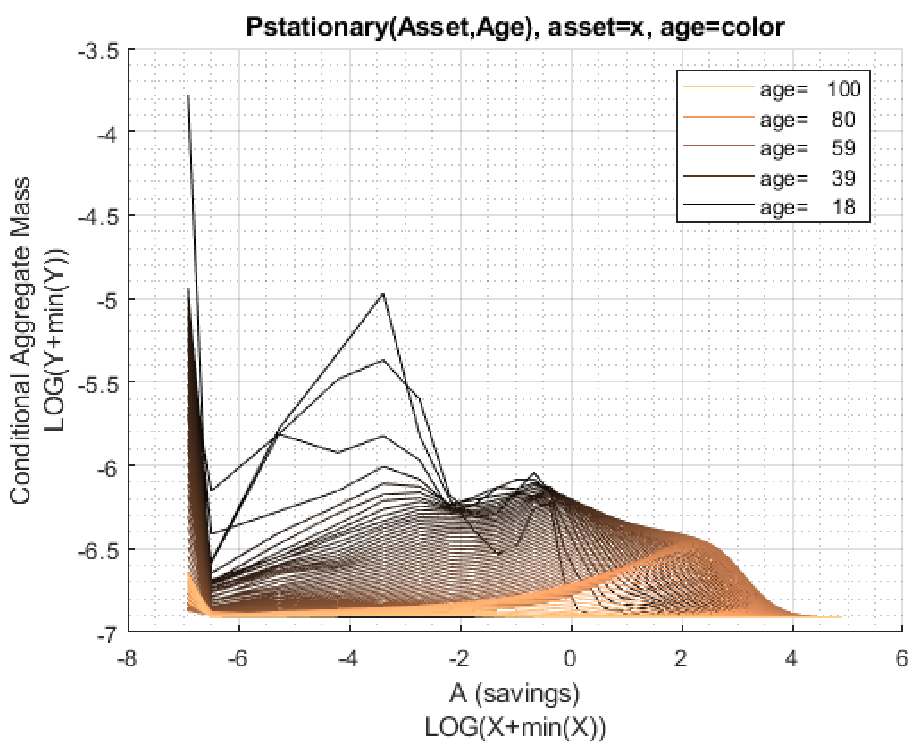
Asset Mass by Age

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [3,4,5,6,1,2];
% Value Function
tb_prob_aage = ff_summ_nd_array("P(A,Z))", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar_perm
```

```
xxx  P(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      sum_age_18    sum_age_19    sum_age_20    sum_age_21    sum_age_22    sum
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0     0.021837     0.0023506     0.0017989     0.0039364     0.0058286     0.
      2     0.00051498            0    0.00039478    0.00037898     0.0011264    0.00064408     0.0
      3      0.0041199            0     0.0020814     0.0019874     0.0019922    0.00088345     0.0
      4       0.013905            0     0.0038512     0.0031528     0.0016753     0.0011288     0.0
      5       0.032959            0     0.0059559     0.0036616     0.0019605     0.0014625     0.
      6       0.064373            0     0.0019624     0.0026806     0.0015589     0.0012782     0.
      7        0.11124            0     0.0010231     0.0010755     0.0008941     0.0009494     0.
      8        0.17664            0    0.00067266    0.00082141     0.0009652     0.0010565     0.
      9        0.26367            0    0.00045811    0.00086333     0.0011647     0.0011767     0.
     10        0.37542            0    0.00053656      0.001129     0.0012812     0.0011528     0.
     11        0.51498            0    0.00090841     0.0013729     0.0012822     0.0012345      0
     12        0.68544            0    0.00097635     0.0011151     0.0011024      0.001139     0.
     13        0.88989            0    0.00023558    0.00050592    0.00075138    0.00094942      0
     14         1.1314            0     4.591e-05    0.00027667    0.00049327    0.00061366     0.0
     15         1.4131            0    1.7547e-05    0.00019582    0.00030348    0.00040728     0.0
     16         1.7381            0    8.2268e-06    6.7346e-05    0.00015086    0.00025852     0.0
     17         2.1094            0    6.1503e-06    3.6395e-05    9.6376e-05    0.00016375     0.0
     18         2.5301            0     1.345e-05     3.725e-05    7.4002e-05    0.00012113     0.0
     19         3.0034            0    2.2526e-05    4.8231e-05    7.8125e-05    0.00011085     0.0
     20         3.5323            0    2.9888e-05    5.5596e-05    8.1028e-05    0.00010487     0.0
     21         4.1199            0    3.0433e-05    5.4594e-05    7.2792e-05    9.1925e-05     0.0
     22         4.7693            0    2.0409e-05    3.7846e-05    5.5558e-05    7.2536e-05     8.9
     23         5.4836            0    5.1452e-06    1.8425e-05    3.2883e-05    4.8468e-05     6.5
     24         6.2658            0    7.3282e-07    5.3334e-06    1.4182e-05    2.7049e-05     4.1
     25         7.1191            0     1.062e-07    1.2922e-06    4.9633e-06    1.2247e-05     2.3
     26         8.0466            0    1.7779e-08    5.0549e-07    2.0442e-06    5.3225e-06     1.1
     27         9.0514            0    3.0263e-09    3.0488e-07    1.0981e-06    2.7972e-06     5.7
     28         10.136            0    1.2227e-10    1.6491e-07    5.5442e-07    1.5926e-06     3.3
     29         11.305            0             0    4.8394e-08    2.2296e-07    8.1497e-07     1.9
     30          12.56            0             0    9.3997e-09     7.016e-08    3.1478e-07     1.0
     31         13.905            0             0     1.808e-09    2.0992e-08    9.9385e-08     4.7
     32         15.342            0             0    4.1404e-10    6.2716e-09    3.4866e-08     1.9
     33         16.875            0             0    9.9687e-11    1.6909e-09    1.3108e-08     6.2
     34         18.507            0             0    2.1381e-11    4.7515e-10    4.3065e-09     2.1
     35         20.241            0             0     8.897e-13    1.3917e-10    1.2185e-09     8.3
     36          22.08            0             0             0     2.966e-11    3.6653e-10     2.8
     37         24.027            0             0             0    3.6991e-12    1.1419e-10     8.4
     38         26.085            0             0             0    7.8046e-13    2.4029e-11     2.7
     39         28.258            0             0             0    1.7968e-13    4.0593e-12     8.1
     40         30.548            0             0             0    8.7684e-15    1.0642e-12     1.7
     41         32.959            0             0             0             0    1.9771e-13     3.6
     42         35.493            0             0             0             0    1.5011e-14     9.2
     43         38.154            0             0             0             0    2.3721e-15      1.
     44         40.945            0             0             0             0    3.0185e-16     1.7
     45         43.868            0             0             0             0    6.4297e-18     3.2
     46         46.928            0             0             0             0             0     3.4
     47         50.126            0             0             0             0              ...

mp_support_graph('cl_st_graph_title') = {'Pstationary(Asset,Age), asset=x, age=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
mp_support_graph('cl_st_xtitle') = {'A (savings)'};
mp_support_graph('st_rowvar_name') = 'age=';
mp_support_graph('it_legend_select') = 5;
mp_support_graph('st_rounding') = '6.0f';
```

```
mp_support_graph('bl_graph_logy') = true;
mp_support_graph('cl_colors') = 'copper';
ff_graph_grid((tb_prob_aage{1:end, 3:end})', age_grid, agrid, mp_support_graph);% Consumption Choice
```





7.2.6 Probability Statistics A, C and V Conditional on Ages

Where are the mass at?

```
ap_ss = mp_dsvfi_results('ap_ss');
c_ss = mp_dsvfi_results('cons_ss');
```

```
v_ss = mp_dsvfi_results('v_ss');
n_ss = mp_dsvfi_results('n_ss');

y_head_inc = mp_dsvfi_results('y_head_inc_ss');
y_spouse_inc = mp_dsvfi_results('y_spouse_inc_ss');

yshr_wage = mp_dsvfi_results('yshr_wage_ss');
yshr_SS = mp_dsvfi_results('yshr_SS_ss');
yshr_nttxss = mp_dsvfi_results('yshr_nttxss_ss');

for it_ctr=1:size(ap_ss, 1)
    if (ismember(it_ctr, round(linspace(1, size(ap_ss, 1), 3))))
        display(['age =' num2str(age_grid(it_ctr))]);

        % construct input data
        Phi_true_age = Phi_true(it_ctr, :, :, : ,: ,:);
        ap_ss_age = ap_ss(it_ctr, :, :, : ,: ,:);
        c_ss_age = c_ss(it_ctr, :, :, : ,: ,:);
        v_ss_age = v_ss(it_ctr, :, :, : ,: ,:);
        n_ss_age = n_ss(it_ctr, :, :, : ,: ,:);

        y_head_inc_age = y_head_inc(it_ctr, :, :, : ,: ,:);
        y_spouse_inc_age = y_spouse_inc(it_ctr, :, :, : ,: ,:);
        yshr_wage_age = yshr_wage(it_ctr, :, :, : ,: ,:);
        yshr_SS_age = yshr_SS(it_ctr, :, :, : ,: ,:);
        yshr_nttxss_age = yshr_nttxss(it_ctr, :, :, : ,: ,:);

        mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
        mp_cl_ar_xyz_of_s('ap_ss') = {ap_ss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('c_ss') = {c_ss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('v_ss') = {v_ss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('n_ss') = {n_ss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('y_head_inc') = {y_head_inc_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('y_spouse') = {y_spouse_inc_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('yshr_wage') = {yshr_wage_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('yshr_SS') = {yshr_SS_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('yshr_nttxss') = {yshr_nttxss_age(:), zeros(1)};
        mp_cl_ar_xyz_of_s('ar_st_y_name') = ["ap_ss", "c_ss", "v_ss", "n_ss",...
            "y_head_inc", "y_spouse", "yshr_wage", "yshr_SS", "yshr_nttxss"];

        % controls
        mp_support = containers.Map('KeyType','char', 'ValueType','any');
        mp_support('ar_fl_percentiles') = [0.01 10 25 50 75 90 99.99];
        mp_support('bl_display_final') = true;
        mp_support('bl_display_detail') = false;
        mp_support('bl_display_drvm2outcomes') = false;
        mp_support('bl_display_drvstats') = false;
        mp_support('bl_display_drvm2covcor') = false;

        % Call Function
        mp_cl_mt_xyz_of_s = ff_simu_stats(Phi_true_age(:)/sum(Phi_true_age,'all'), mp_cl_ar_xyz_of_s
    end
end
```

```
age =18
xxx tb_outcomes: all stats xxx
    OriginalVariableNames      ap_ss         c_ss          v_ss           n_ss        y_head_inc
    _____________________    __________    __________    ___________    __________    __________
```

```
    {'mean'               }      0.13209       0.63361         -59.72        1.9854        0.71265
    {'unweighted_sum'     }   1.0935e+07    8.5257e+05    -1.1176e+07            21          15541
    {'sd'                 }      0.34847       0.37861         29.967        1.0848        0.54567
    {'coefofvar'          }       2.6381       0.59754       -0.50178       0.54639        0.76569
    {'gini'               }       0.7705        0.3109       -0.25126         0.268        0.36259
    {'min'                }            0      0.036717        -615.77             1       0.038108
    {'max'                }       145.08        10.204        -3.7499             6         13.784
    {'pYis0'              }      0.10805             0              0             0              0
    {'pYls0'              }            0             0              1             0              0
    {'pYgr0'              }      0.89195             1              0             1              1
    {'pYisMINY'           }      0.10805    1.3288e-05     5.8837e-08       0.41786     2.5312e-05
    {'pYisMAXY'           }            0             0              0     0.0060544              0
    {'p0_01'              }            0      0.047727        -352.03             1       0.046651
    {'p10'                }            0       0.24819        -96.425             1        0.23528
    {'p25'                }     0.012214       0.36957        -70.656             1        0.35258
    {'p50'                }     0.032959       0.55272        -52.866             2        0.56523
    {'p75'                }     0.076248       0.80075        -39.739             3        0.90612
    {'p90'                }       0.4782        1.1197        -31.147             4         1.3579
    {'p99_99'             }       5.4534        3.6548        -10.999             6         6.8484
    {'fl_cov_ap_ss'       }      0.12143      0.055156         2.4756       0.02663       0.050357
    {'fl_cor_ap_ss'       }            1       0.41805        0.23707      0.070443        0.26483
    {'fl_cov_c_ss'        }     0.055156       0.14335         8.0725      0.076682        0.18653
    {'fl_cor_c_ss'        }      0.41805             1         0.7115        0.1867        0.90288
    {'fl_cov_v_ss'        }       2.4756        8.0725            898       0.45095          10.05
    {'fl_cor_v_ss'        }      0.23707        0.7115              1      0.013872        0.61462
    {'fl_cov_n_ss'        }      0.02663      0.076682        0.45095        1.1768      -4.12e-18
    {'fl_cor_n_ss'        }     0.070443        0.1867       0.013872             1      -6.96e-18
    {'fl_cov_y_head_inc'  }     0.050357       0.18653          10.05     -4.12e-18        0.29776
    {'fl_cor_y_head_inc'  }      0.26483       0.90288        0.61462     -6.96e-18              1
    {'fl_cov_y_spouse'    }      0.18246      0.071672         3.4951       0.13323       0.010455
    {'fl_cor_y_spouse'    }      0.91943       0.33241         0.2048       0.21565       0.033645
    {'fl_cov_yshr_wage'   }   7.6315e-33    1.7081e-32    -2.0646e-31    3.5437e-31     6.4579e-33
    {'fl_cor_yshr_wage'   }   9.8628e-17    2.0318e-16    -3.1028e-17    1.4712e-15     5.3299e-17
    {'fl_cov_yshr_SS'     }            0             0              0             0              0
    {'fl_cor_yshr_SS'     }          NaN           NaN            NaN           NaN            NaN
    {'fl_cov_yshr_nttxss'}     0.0057593      0.011163        0.86319      0.007516        0.01319
    {'fl_cor_yshr_nttxss'}       0.48714       0.86903        0.84902       0.20421        0.71249
    {'fracByP0_01'        }            0    7.1734e-06     0.00073274       0.21046      7.788e-06
    {'fracByP10'          }            0      0.030664        0.21367       0.21046       0.027495
    {'fracByP25'          }    0.0067284       0.10372         0.4286       0.21046       0.092606
    {'fracByP50'          }     0.046851       0.29072        0.67444       0.53024        0.26377
    {'fracByP75'          }      0.13176       0.54795        0.87012       0.77109         0.5245
    {'fracByP90'          }      0.35932       0.76949        0.95644       0.92834        0.74403
    {'fracByP99_99'       }      0.99576       0.99938        0.99998             1        0.99912
age =59
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       ap_ss         c_ss          v_ss          n_ss       y_head_inc
    ______________________    __________    __________    ___________    _________    __________

    {'mean'               }       9.6978        1.2003        -27.032       1.7239         1.6127
    {'unweighted_sum'     }   1.1254e+07    1.0744e+06    -5.7273e+06           21          45380
    {'sd'                 }       9.5091       0.76817          15.51      0.90777          1.276
    {'coefofvar'          }      0.98054          0.64       -0.57376      0.52659        0.79122
    {'gini'               }      0.47956       0.33158       -0.29874      0.23461        0.38177
    {'min'                }            0       0.05663        -230.76            1       0.059541
    {'max'                }       158.47        12.271         -1.694            6          23.47
    {'pYis0'              }     0.004589             0              0            0              0
```

```
    {'pYls0'             }            0             0             1            0             0
    {'pYgr0'             }      0.99541             1             0            1             1
    {'pYisMINY'          }     0.004589    9.8045e-06    2.9682e-09      0.48835     9.869e-06
    {'pYisMAXY'          }   9.1885e-09    2.1301e-11    5.3537e-07    0.0036816    1.4932e-06
    {'p0_01'             }            0       0.07838       -123.12            1       0.08341
    {'p10'               }       1.2229       0.40584       -47.779            1       0.49527
    {'p25'               }        3.196        0.6516       -33.261            1       0.77993
    {'p50'               }       7.0976        1.0499       -23.135            2        1.2719
    {'p75'               }       13.089         1.547       -16.274            2        2.0397
    {'p90'               }       21.159        2.1475       -11.776            3        3.1029
    {'p99_99'            }       112.62        8.4781       -2.7295            6        15.937
    {'fl_cov_ap_ss'      }       90.423        6.9267        101.69      0.81683        10.484
    {'fl_cor_ap_ss'      }            1       0.94827       0.68949     0.094628       0.86408
    {'fl_cov_c_ss'       }       6.9267       0.59008         8.839      0.23092       0.85409
    {'fl_cor_c_ss'       }      0.94827             1       0.74189      0.33116       0.87137
    {'fl_cov_v_ss'       }       101.69         8.839        240.55       2.5586        13.062
    {'fl_cor_v_ss'       }      0.68949       0.74189             1      0.18173       0.66001
    {'fl_cov_n_ss'       }      0.81683       0.23092        2.5586      0.82404      0.055266
    {'fl_cor_n_ss'       }     0.094628       0.33116       0.18173            1      0.047713
    {'fl_cov_y_head_inc' }       10.484       0.85409        13.062     0.055266        1.6281
    {'fl_cor_y_head_inc' }      0.86408       0.87137       0.66001     0.047713             1
    {'fl_cov_y_spouse'   }       2.2203       0.24459        3.6105      0.27625        0.1162
    {'fl_cor_y_spouse'   }      0.20978       0.28607       0.20915      0.27342       0.08182
    {'fl_cov_yshr_wage'  }     -0.53029     -0.034916      -0.87809    0.0021706     -0.036142
    {'fl_cor_yshr_wage'  }     -0.55378      -0.45137      -0.56221     0.023745      -0.28128
    {'fl_cov_yshr_SS'    }            0             0             0            0             0
    {'fl_cor_yshr_SS'    }          NaN           NaN           NaN          NaN           NaN
    {'fl_cov_yshr_nttxss'}      0.19504      0.017838       0.43624    0.0074857      0.026908
    {'fl_cor_yshr_nttxss'}      0.67474       0.76393       0.92529      0.27128       0.69373
    {'fracByP0_01'       }            0     6.875e-06    0.00049633      0.28329    5.7641e-06
    {'fracByP10'         }    0.0057779      0.026049       0.23158      0.28329      0.022494
    {'fracByP25'         }     0.040074      0.091752       0.44951      0.28329      0.082256
    {'fracByP50'         }      0.16977       0.26933       0.70658      0.72028       0.24076
    {'fracByP75'         }      0.42173       0.53549       0.88743      0.72028       0.48935
    {'fracByP90'         }      0.67785       0.76094       0.96591      0.85389       0.72071
    {'fracByP99_99'      }      0.99869       0.99925       0.99999            1       0.99889
age =100
xxx tb_outcomes: all stats xxx
    OriginalVariableNames     ap_ss       c_ss          v_ss          n_ss        y_head_inc      y_s
    ______________________    _____    __________    __________    __________    ___________    ____

    {'mean'              }      0         0.33019       -4.3147        1.4797         0.2355      0
    {'unweighted_sum'    }      0      1.2179e+05       -3630.5            21         209.83
    {'sd'                }      0         0.23351        1.1931       0.50567       0.021892
    {'coefofvar'         }    NaN          0.7072      -0.27653       0.34173       0.092957
    {'gini'              }    NaN          0.2934      -0.14462       0.12034       0.043711      0
    {'min'               }      0         0.19737       -12.197             1           0.22
    {'max'               }      0          141.61    -0.0071775             6          5.666
    {'pYis0'             }      1               0             0             0              0      0
    {'pYls0'             }      0               0             1             0              0
    {'pYgr0'             }      0               1             0             1              1      0
    {'pYisMINY'          }      1         0.35707    1.4848e-10        0.5232        0.50347      0
    {'pYisMAXY'          }      1               0             0    4.2206e-08              0   1.03
    {'p0_01'             }      0         0.19737        -8.038             1           0.22
    {'p10'               }      0         0.19737       -5.0665             1           0.22
    {'p25'               }      0         0.19737       -5.0665             1           0.22
    {'p50'               }      0         0.23607       -5.0103             1           0.22
    {'p75'               }      0         0.34876       -3.7889             2          0.266      0
```

```
    {'p90'              }     0       0.58892      -2.3842            2        0.26656      0
    {'p99_99'           }     0        2.8508     -0.49608            4        0.31126
    {'fl_cov_ap_ss'     }     0             0            0            0              0
    {'fl_cor_ap_ss'     }   NaN           NaN          NaN          NaN            NaN
    {'fl_cov_c_ss'      }     0      0.054526      0.22623     0.059682      0.0015233     0.
    {'fl_cor_c_ss'      }   NaN             1        0.812      0.50545          0.298      0
    {'fl_cov_v_ss'      }     0       0.22623       1.4236      0.21735       0.011521      0
    {'fl_cor_v_ss'      }   NaN         0.812            1      0.36026        0.44106      0
    {'fl_cov_n_ss'      }     0      0.059682      0.21735       0.2557      0.0018733     0.
    {'fl_cor_n_ss'      }   NaN       0.50545      0.36026            1        0.16923      0
    {'fl_cov_y_head_inc' }     0     0.0015233     0.011521    0.0018733     0.00047925   0.00
    {'fl_cor_y_head_inc' }   NaN         0.298      0.44106      0.16923              1      0
    {'fl_cov_y_spouse'  }     0      0.050691      0.18354     0.052581     0.00064761     0.
    {'fl_cor_y_spouse'  }   NaN       0.88823      0.62943      0.42546        0.12104
    {'fl_cov_yshr_wage' }     0      0.040339      0.19022     0.087809     0.00068213     0.
    {'fl_cor_yshr_wage' }   NaN       0.75656      0.69819      0.76049        0.13646      0
    {'fl_cov_yshr_SS'   }     0     -0.041495     -0.19728    -0.089315    -0.00073742     -0
    {'fl_cor_yshr_SS'   }   NaN      -0.77343     -0.71963     -0.76875       -0.14661     -0
    {'fl_cov_yshr_nttxss'}     0      0.045639      0.21697     0.096294     0.00089018      0
    {'fl_cor_yshr_nttxss'}   NaN       0.78185      0.72746      0.76178        0.16266      0
    {'fracByP0_01'      }   NaN       0.21345   0.00043423      0.35357        0.47033
    {'fracByP10'        }   NaN       0.21345      0.51425      0.35357        0.47033
    {'fracByP25'        }   NaN       0.21345      0.51425      0.35357        0.47033
    {'fracByP50'        }   NaN       0.33309       0.6114      0.35357        0.47033
    {'fracByP75'        }   NaN       0.53778       0.8525      0.99419        0.87579      0
    {'fracByP90'        }   NaN       0.74393      0.95864      0.99419         0.8936      0
    {'fracByP99_99'     }   NaN       0.99922      0.99999      0.99999        0.99991
```

7.3 Distribution Exact Savings Choices Vectorized

This is the example vignette for function: **snw_ds_main_vec** from the **PrjOptiSNW Package**. This function solves for vfi and gets distribution induced by policy functions and exogenous distributions. Vectorized vfi and distribution methods.

7.3.1 Test SNW_DS_MAIN_VEC

Call the function with testing defaults.

```
% mp_params = snw_mp_param('default_dense');
mp_params = snw_mp_param('default_docdense');
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_ds') = true;
mp_controls('bl_print_ds_verbose') = false;
[Phi_true,Phi_adj,A_agg,Y_inc_agg,it,mp_dsvfi_results] = snw_ds_main_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=543.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                 i    idx    ndim     numel      rowN      colN          sum          mean        std
                 -    ---    ----    --------    ----    ---------    -----------    -------     -----

    V_VFI        1     1      6      4.37e+07     83     5.265e+05    -8.6673e+08    -19.834     28.17
    ap_VFI       2     2      6      4.37e+07     83     5.265e+05     1.4164e+09     32.412       36.
    cons_VFI     3     3      6      4.37e+07     83     5.265e+05      2.131e+08     4.8764     8.326
```

```
xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1          c2          c3          c4          c5         c526496       c526497       c526498
          -------     -------     -------     -------     -------     ---------     ---------     --------

    r1    -376.05     -375.66     -373.17      -367.4     -358.05         -6.68       -6.5297        -6.379
    r2     -363.8     -363.41     -360.93     -355.25     -346.25       -6.4892       -6.3437        -6.197
    r3    -351.75     -351.36      -348.9     -343.44      -334.9       -6.2948       -6.1538        -6.011
    r4    -339.81     -339.45     -337.16     -332.06     -324.04        -6.095       -5.9584          -5.8
    r5    -328.99     -328.65     -326.51     -321.72     -314.17       -5.9054       -5.7725        -5.637
    r79   -14.033      -14.02     -13.926     -13.689     -13.287      -0.22848      -0.21775       -0.2076
    r80   -12.564      -12.55     -12.457      -12.22     -11.818      -0.17427      -0.16611       -0.1584
    r81   -10.778     -10.764     -10.671     -10.434     -10.032      -0.11927      -0.11368       -0.1084
    r82   -8.4226     -8.4089     -8.3155     -8.0786     -7.6766      -0.06597      -0.06284      -0.05992
    r83   -5.0665     -5.0529     -4.9595     -4.7226     -4.3206     -0.020968     -0.019972      -0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
          c1    c2        c3            c4             c5         c526496    c526497    c526498    c52
          --    --    ----------    ---------     ----------     -------    -------    -------    ---

    r1    0     0      0.0005656    0.0075134      0.022901     114.76     120.42     126.29     132
    r2    0     0     0.00051498    0.0065334      0.021549     114.87     120.54     126.42     132
    r3    0     0     0.00051498    0.0049294      0.019875     114.98     120.67     126.57     132
    r4    0     0     0.00051498    0.0047937      0.019672     115.74     121.44     127.36     133
    r5    0     0     0.00048517    0.0046683      0.019484     116.51     122.22     128.16     134
    r79   0     0              0            0    0.00051498     81.091      85.68     90.325     94.
    r80   0     0              0            0             0     76.669      80.55     84.292     88.
    r81   0     0              0            0             0     68.313      71.52     74.459     77.
    r82   0     0              0            0             0     50.126     53.467     56.953     58.
    r83   0     0              0            0             0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
             c1          c2          c3          c4          c5       c526496    c526497    c526498
          --------    --------    --------    --------    --------    -------    -------    -------

    r1    0.036717    0.037251    0.040426     0.04363    0.048012    9.6396     9.8066     9.9533
    r2    0.036717    0.037251    0.040477     0.04461    0.049364    9.8014     9.9571     10.088
    r3    0.036717    0.037251    0.040477    0.046214    0.051039    9.9664     10.108      10.22
    r4    0.038144    0.038678    0.041903    0.047776    0.052666    10.118     10.244     10.339
    r5    0.039534    0.040068    0.043323     0.04929    0.054241    10.258     10.369     10.446
    r79    0.19737     0.19791     0.20163     0.21175     0.23093    35.811     37.046     38.418
    r80    0.19737     0.19791     0.20163     0.21175     0.23145    40.207      42.15     44.426
    r81    0.19737     0.19791     0.20163     0.21175     0.23145    48.541     51.158     54.236
    r82    0.19737     0.19791     0.20163     0.21175     0.23145     66.71     69.193     71.724
    r83    0.19737     0.19791     0.20163     0.21175     0.23145    116.82     122.65     128.66

SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:1 of 82, time-this-age:0.3786
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:2 of 82, time-this-age:4.8096
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:3 of 82, time-this-age:5.9174
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:4 of 82, time-this-age:6.508
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:5 of 82, time-this-age:6.8927
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:6 of 82, time-this-age:7.5887
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:7 of 82, time-this-age:8.1099
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:8 of 82, time-this-age:9.0281
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:9 of 82, time-this-age:9.1355
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:10 of 82, time-this-age:11.9284
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:11 of 82, time-this-age:11.8537
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:12 of 82, time-this-age:14.3041
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:13 of 82, time-this-age:14.4545
```

```
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:14 of 82, time-this-age:14.7013
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:15 of 82, time-this-age:14.7272
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:16 of 82, time-this-age:14.7332
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:17 of 82, time-this-age:15.1111
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:18 of 82, time-this-age:14.5327
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:19 of 82, time-this-age:15.2564
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:20 of 82, time-this-age:15.4999
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:21 of 82, time-this-age:15.5015
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:22 of 82, time-this-age:16.8247
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:23 of 82, time-this-age:17.1551
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:24 of 82, time-this-age:16.2079
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:25 of 82, time-this-age:16.4377
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:26 of 82, time-this-age:16.5744
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:27 of 82, time-this-age:17.8337
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:28 of 82, time-this-age:17.647
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:29 of 82, time-this-age:17.601
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:30 of 82, time-this-age:17.5908
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:31 of 82, time-this-age:17.2374
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:32 of 82, time-this-age:17.3831
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:33 of 82, time-this-age:17.7541
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:34 of 82, time-this-age:17.2324
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:35 of 82, time-this-age:17.6096
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:36 of 82, time-this-age:17.8942
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:37 of 82, time-this-age:17.3612
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:38 of 82, time-this-age:16.873
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:39 of 82, time-this-age:16.9254
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:40 of 82, time-this-age:16.8782
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:41 of 82, time-this-age:16.5205
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:42 of 82, time-this-age:16.8312
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:43 of 82, time-this-age:16.23
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:44 of 82, time-this-age:16.1342
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:45 of 82, time-this-age:16.4806
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:46 of 82, time-this-age:15.8565
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:47 of 82, time-this-age:15.4775
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:48 of 82, time-this-age:15.4721
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:49 of 82, time-this-age:19.033
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:50 of 82, time-this-age:20.2225
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:51 of 82, time-this-age:20.5336
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:52 of 82, time-this-age:19.8387
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:53 of 82, time-this-age:20.2331
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:54 of 82, time-this-age:20.6319
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:55 of 82, time-this-age:20.6243
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:56 of 82, time-this-age:14.9386
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:57 of 82, time-this-age:14.0026
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:58 of 82, time-this-age:14.161
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:59 of 82, time-this-age:14.8816
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:60 of 82, time-this-age:10.6737
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:61 of 82, time-this-age:16.4232
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:62 of 82, time-this-age:10.5139
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:63 of 82, time-this-age:12.1908
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:64 of 82, time-this-age:11.9561
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:65 of 82, time-this-age:12.7329
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:66 of 82, time-this-age:13.7721
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:67 of 82, time-this-age:12.0024
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:68 of 82, time-this-age:11.9243
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:69 of 82, time-this-age:12.7582
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:70 of 82, time-this-age:12.1532
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:71 of 82, time-this-age:9.4885
```

```
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:72 of 82, time-this-age:10.7963
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:73 of 82, time-this-age:11.1703
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:74 of 82, time-this-age:10.3998
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:75 of 82, time-this-age:9.1922
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:76 of 82, time-this-age:11.3241
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:77 of 82, time-this-age:10.0886
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:78 of 82, time-this-age:9.2365
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:79 of 82, time-this-age:7.7439
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:80 of 82, time-this-age:7.7752
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:81 of 82, time-this-age:7.2503
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:82 of 82, time-this-age:6.2078
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:1 of 82, time-this-age:0.4434
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:2 of 82, time-this-age:0.057613
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:3 of 82, time-this-age:0.063844
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:4 of 82, time-this-age:0.050822
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:5 of 82, time-this-age:0.070678
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:6 of 82, time-this-age:0.060849
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:7 of 82, time-this-age:0.065418
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:8 of 82, time-this-age:0.061329
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:9 of 82, time-this-age:0.056986
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:10 of 82, time-this-age:0.063646
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:11 of 82, time-this-age:0.05255
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:12 of 82, time-this-age:0.057059
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:13 of 82, time-this-age:0.048215
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:14 of 82, time-this-age:0.063939
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:15 of 82, time-this-age:0.052633
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:16 of 82, time-this-age:0.056284
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:17 of 82, time-this-age:0.059784
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:18 of 82, time-this-age:0.053468
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:19 of 82, time-this-age:0.064496
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:20 of 82, time-this-age:0.067389
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:21 of 82, time-this-age:0.069029
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:22 of 82, time-this-age:0.051852
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:23 of 82, time-this-age:0.071656
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:24 of 82, time-this-age:0.055803
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:25 of 82, time-this-age:0.050012
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:26 of 82, time-this-age:0.056304
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:27 of 82, time-this-age:0.048197
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:28 of 82, time-this-age:0.048567
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:29 of 82, time-this-age:0.047927
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:30 of 82, time-this-age:0.055782
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:31 of 82, time-this-age:0.065655
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:32 of 82, time-this-age:0.056483
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:33 of 82, time-this-age:0.054022
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:34 of 82, time-this-age:0.048944
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:35 of 82, time-this-age:0.049292
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:36 of 82, time-this-age:0.057768
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:37 of 82, time-this-age:0.05061
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:38 of 82, time-this-age:0.054212
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:39 of 82, time-this-age:0.058645
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:40 of 82, time-this-age:0.04779
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:41 of 82, time-this-age:0.050251
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:42 of 82, time-this-age:0.053709
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:43 of 82, time-this-age:0.059374
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:44 of 82, time-this-age:0.049218
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:45 of 82, time-this-age:0.051461
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:46 of 82, time-this-age:0.046182
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:47 of 82, time-this-age:0.052269
```

```
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:48 of 82, time-this-age:0.046998
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:49 of 82, time-this-age:0.067955
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:50 of 82, time-this-age:0.063089
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:51 of 82, time-this-age:0.051776
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:52 of 82, time-this-age:0.049762
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:53 of 82, time-this-age:0.068838
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:54 of 82, time-this-age:0.056374
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:55 of 82, time-this-age:0.079779
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:56 of 82, time-this-age:0.07178
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:57 of 82, time-this-age:0.049449
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:58 of 82, time-this-age:0.062064
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:59 of 82, time-this-age:0.054074
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:60 of 82, time-this-age:0.066304
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:61 of 82, time-this-age:0.056874
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:62 of 82, time-this-age:0.057744
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:63 of 82, time-this-age:0.074837
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:64 of 82, time-this-age:0.054466
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:65 of 82, time-this-age:0.063535
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:66 of 82, time-this-age:0.057064
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:67 of 82, time-this-age:0.058795
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:68 of 82, time-this-age:0.052117
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:69 of 82, time-this-age:0.053227
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:70 of 82, time-this-age:0.065072
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:71 of 82, time-this-age:0.053119
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:72 of 82, time-this-age:0.062005
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:73 of 82, time-this-age:0.05542
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:74 of 82, time-this-age:0.060745
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:75 of 82, time-this-age:0.05168
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:76 of 82, time-this-age:0.051709
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:77 of 82, time-this-age:0.058973
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:78 of 82, time-this-age:0.059053
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:79 of 82, time-this-age:0.069633
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:80 of 82, time-this-age:0.046625
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:81 of 82, time-this-age:0.069697
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:82 of 82, time-this-age:0.049734
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:83 of 82, time-this-age:0.046755
SNW_DS_MAIN: Share of population with assets equal to upper bound on asset grid:6.8049e-06
SNW_DS_MAIN: Accidental bequests are thrown in the ocean
SNW_DS_MAIN_VEC tax and spend;it=1;err=0.014163
SNW_DS_MAIN_VEC tax and spend;it=2;err=0.011847
SNW_DS_MAIN_VEC tax and spend;it=3;err=0.0098976
SNW_DS_MAIN_VEC tax and spend;it=4;err=0.0082601
SNW_DS_MAIN_VEC tax and spend;it=5;err=0.0068875
SNW_DS_MAIN_VEC tax and spend;it=6;err=0.0057387
SNW_DS_MAIN_VEC tax and spend;it=7;err=0.0047786
SNW_DS_MAIN_VEC tax and spend;it=8;err=0.0039772
SNW_DS_MAIN_VEC tax and spend;it=9;err=0.0033087
SNW_DS_MAIN_VEC tax and spend;it=10;err=0.0027516
SNW_DS_MAIN_VEC tax and spend;it=11;err=0.0022876
SNW_DS_MAIN_VEC tax and spend;it=12;err=0.0019014
SNW_DS_MAIN_VEC tax and spend;it=13;err=0.0015801
SNW_DS_MAIN_VEC tax and spend;it=14;err=0.0013128
SNW_DS_MAIN_VEC tax and spend;it=15;err=0.0010906
SNW_DS_MAIN_VEC tax and spend;it=16;err=0.00090591
SNW_DS_MAIN_VEC tax and spend;it=17;err=0.00075241
SNW_DS_MAIN_VEC tax and spend;it=18;err=0.00062488
SNW_DS_MAIN_VEC tax and spend;it=19;err=0.00051892
SNW_DS_MAIN_VEC tax and spend;it=20;err=0.00043091
```

```
SNW_DS_MAIN_VEC tax and spend;it=21;err=0.00035781
SNW_DS_MAIN_VEC tax and spend;it=22;err=0.0002971
SNW_DS_MAIN_VEC tax and spend;it=23;err=0.00024668
SNW_DS_MAIN_VEC tax and spend;it=24;err=0.00020481
SNW_DS_MAIN_VEC tax and spend;it=25;err=0.00017004
SNW_DS_MAIN_VEC tax and spend;it=26;err=0.00014118
SNW_DS_MAIN_VEC tax and spend;it=27;err=0.00011721
SNW_DS_MAIN_VEC tax and spend;it=28;err=9.7309e-05
SNW_DS_MAIN_VEC: Number of a2-adjustments (for taxation) used to balance the government budget= 28
SNW_DS_MAIN_VEC: Old and updated value of a2=1.5286      1.4349
SNW_DS_MAIN_VEC: Aggregates: Cons., Gov. cons., Save, Assets, Income, Bequests 48.70063      11.4200
SNW_DS_MAIN_VEC: Resource constraint: C_t+A_{t+1}+G_t=A_t+Y_t 264.4537      264.6461
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1820.5957
xxx tb_outcomes: all stats xxx
     OriginalVariableNames          a_ss          ap_ss        cons_ss        n_ss          y_all
    _______________________      __________    __________    __________    _________    __________

    {'mean'                }         4.3602        4.4621        1.0635       2.3554        1.4661
    {'unweighted_sum'      }           2228    5.3216e+08    5.0787e+07           21    8.3558e+07
    {'sd'                  }         6.8796        6.9169        0.6938       1.4375        1.4665
    {'coefofvar'           }         1.5778        1.5501       0.65237      0.61029        1.0003
    {'gini'                }         0.6755       0.67638       0.33936       0.3128       0.44546
    {'min'                 }              0             0      0.036717            1      0.038108
    {'max'                 }            135        163.73        141.61            6        50.873
    {'pYis0'               }        0.11808      0.097968             0            0             0
    {'pYls0'               }              0             0             0            0             0
    {'pYgr0'               }        0.88192       0.90203             1            1             1
    {'pYisMINY'            }        0.11808      0.097968    8.6094e-07      0.36005    8.6094e-07
    {'pYisMAXY'            }     6.8049e-06    2.2836e-12             0     0.041101    2.2836e-12
    {'p0_01'               }              0             0      0.066342            1      0.069931
    {'p0_1'                }              0             0       0.10404            1       0.11208
    {'p1'                  }              0             0       0.18515            1       0.20327
    {'p5'                  }              0             0       0.27323            1       0.26924
    {'p10'                 }              0    0.00051498       0.35662            1       0.34663
    {'p20'                 }       0.064373      0.083002       0.49455            1       0.49947
    {'p25'                 }        0.17664       0.20935       0.56067            1       0.57738
    {'p30'                 }        0.37542       0.39434       0.62765            1       0.65537
    {'p40'                 }        0.88989       0.90689       0.76677            2       0.82916
    {'p50'                 }         1.7381        1.6906       0.91665            2        1.0312
    {'p60'                 }         3.0034        2.8912        1.0828            2        1.2832
    {'p70'                 }         4.7693        4.7054        1.2762            3        1.6179
    {'p75'                 }         5.4836        5.9044        1.3916            3        1.8375
    {'p80'                 }         7.1191        7.4694        1.5274            4        2.1165
    {'p90'                 }          12.56        12.521        1.9354            5        3.0585
    {'p95'                 }         16.875        17.769        2.3449            5        4.0476
    {'p99'                 }         30.548        31.792        3.4017            6        6.9072
    {'p99_9'               }         56.953        57.866        5.2893            6        14.815
    {'p99_99'              }         90.439        90.717        7.5592            6        21.023
    {'fl_cov_a_ss'         }         47.329        47.318         3.476       -1.478        4.5793
    {'fl_cor_a_ss'         }              1       0.99439       0.72826     -0.14945       0.45389
    {'fl_cov_ap_ss'        }         47.318        47.844        3.5799      -1.4406        5.4304
    {'fl_cor_ap_ss'        }        0.99439             1       0.74597     -0.14488       0.53534
    {'fl_cov_cons_ss'      }          3.476        3.5799       0.48135      0.23978        0.7718
    {'fl_cor_cons_ss'      }        0.72826       0.74597             1      0.24042       0.75854
    {'fl_cov_n_ss'         }         -1.478       -1.4406       0.23978       2.0664       0.36196
    {'fl_cor_n_ss'         }       -0.14945      -0.14488       0.24042            1        0.1717
    {'fl_cov_y_all'        }         4.5793        5.4304        0.7718      0.36196        2.1507
    {'fl_cor_y_all'        }        0.45389       0.53534       0.75854       0.1717             1
```

```
    {'fl_cov_y_head_inc'   }        3.9427          4.2301          0.5704       0.09459          1.1331
    {'fl_cor_y_head_inc'   }       0.56573         0.60369         0.81157      0.064956         0.76267
    {'fl_cov_y_head_earn'  }        1.8957           2.208         0.43323       0.19345         0.98441
    {'fl_cor_y_head_earn'  }       0.29692         0.34397         0.67286       0.14501          0.7233
    {'fl_cov_y_spouse_inc' }       0.63663          1.2003          0.2014       0.26737          1.0177
    {'fl_cor_y_spouse_inc' }      0.096963         0.18183         0.30417       0.19489          0.7271
    {'fl_cov_yshr_interest'}       0.80041         0.75231         0.03747     -0.072581       -0.014203
    {'fl_cor_yshr_interest'}        0.6619         0.61877         0.30725      -0.28725       -0.055097
    {'fl_cov_yshr_wage'    }      -0.80811        -0.71978      -0.0042544       0.17131         0.10973
    {'fl_cor_yshr_wage'    }      -0.34878        -0.30898       -0.018207       0.35386         0.22217
    {'fl_cov_yshr_SS'      }      0.007703        -0.03253       -0.033215     -0.098733       -0.095531
    {'fl_cor_yshr_SS'      }     0.0049412       -0.020754        -0.21127       -0.3031        -0.28746
    {'fl_cov_yshr_tax'     }       0.10041         0.11154        0.018848      0.013683        0.039306
    {'fl_cor_yshr_tax'     }       0.40929         0.45221         0.76179       0.26692         0.75157
    {'fl_cov_yshr_nttxss'  }       0.09271         0.14407        0.052063       0.11242         0.13484
    {'fl_cor_yshr_nttxss'  }      0.054578        0.084358         0.30392       0.31672         0.37237
    {'fracByP0_01'         }             0               0      5.4726e-06       0.15286      4.2573e-06
    {'fracByP0_1'          }             0               0      8.2626e-05       0.15286      6.3235e-05
    {'fracByP1'            }             0               0       0.0013773       0.15286       0.0010873
    {'fracByP5'            }             0               0        0.010075       0.15286       0.0076055
    {'fracByP10'           }             0      1.8925e-07        0.024991       0.15286        0.018124
    {'fracByP20'           }    0.00071111       0.0007016        0.065093       0.15286           0.047
    {'fracByP25'           }     0.0023543       0.0022849        0.089893       0.15286        0.065314
    {'fracByP30'           }     0.0065355       0.0056773         0.11782       0.15286        0.086292
    {'fracByP40'           }      0.023222        0.019937          0.1833       0.40183          0.1368
    {'fracByP50'           }      0.057104        0.048358         0.26234       0.40183         0.20002
    {'fracByP60'           }       0.11454        0.098974         0.35614       0.40183         0.27863
    {'fracByP70'           }        0.2027         0.18293         0.46671       0.56321         0.37699
    {'fracByP75'           }       0.23918         0.24222         0.52936       0.56321         0.43577
    {'fracByP80'           }       0.32171         0.31675         0.59788       0.75407         0.50297
    {'fracByP90'           }       0.55921         0.53294         0.75856        0.8953         0.67532
    {'fracByP95'           }       0.69427         0.69899         0.85803        0.8953         0.79442
    {'fracByP99'           }       0.90202          0.9039         0.96048             1         0.93109
    {'fracByP99_9'         }        0.9849         0.98396         0.99413             1           0.988
    {'fracByP99_99'        }       0.99794         0.99764         0.99921             1         0.99841
```

```
% [Phi_true,Phi_adj] = snw_ds_main(mp_params, mp_controls);
Phi_true = Phi_true/sum(Phi_true(:));
```

7.3.2 Show All Info in mp_dsvfi_results

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'))
```

```
                       mean      unweighted_sum       sd       coefofvar     gini        min
                     --------    --------------    --------    ---------    -------    --------    -

    a_ss               4.3602             2228      6.8796       1.5778      0.6755           0
    ap_ss              4.4621       5.3216e+08      6.9169       1.5501     0.67638           0
    cons_ss            1.0635       5.0787e+07      0.6938      0.65237     0.33936    0.036717
    n_ss               2.3554               21      1.4375      0.61029      0.3128           1
    y_all              1.4661       8.3558e+07      1.4665       1.0003     0.44546    0.038108
    y_head_inc         1.1081       1.9253e+06       1.013      0.91419     0.42164    0.038108
    y_head_earn       0.88655            19732     0.92804       1.0468     0.53121           0
    y_spouse_inc      0.35797        4.827e+05     0.95437       2.6661     0.85269           0
    yshr_interest     0.12865       3.8438e+06     0.17577       1.3663     0.65781           0    0
    yshr_wage         0.77402       8.8881e+06     0.33679      0.43512      0.2062           0
    yshr_SS          0.097329            29012      0.2266       2.3282     0.91382           0
```

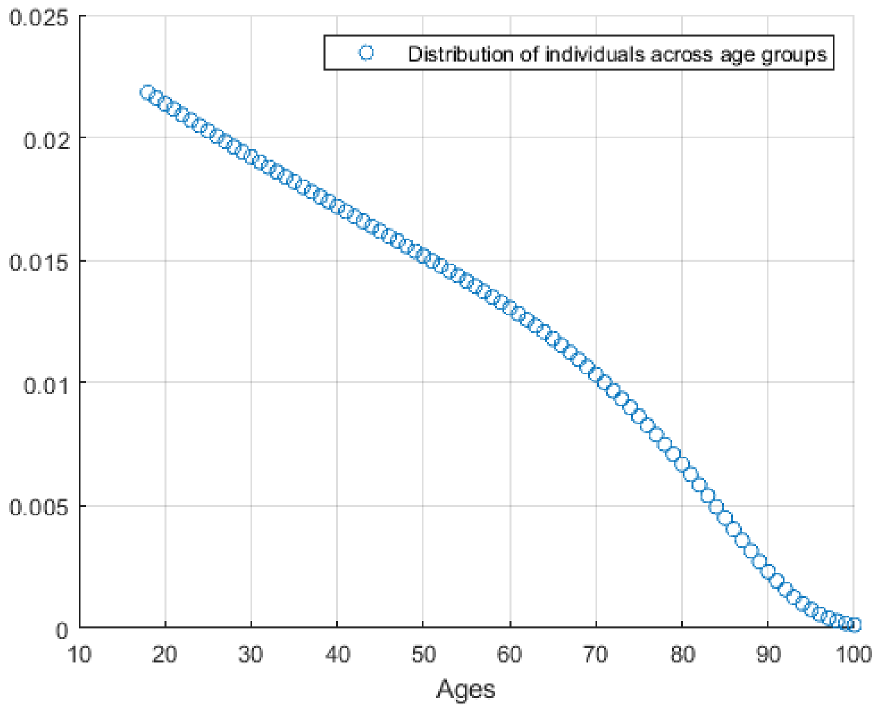
```
    yshr_tax          0.17833       2.8338e+06      0.035661      0.19998      0.11386     0.036506
    yshr_nttxss       0.080996      2.8048e+06       0.24691       3.0485       1.2592     -0.89715
```

7.3.3 Show Distribution by Age

Note that the age-distribution is exogenously determined by the age-specific survival rate. From: "load('Mortality_prob_by_age_18_99.mat','mort_prob') psi_full=1-mort_prob;".

First, we generate a vector for the age-specific mass, and visualize this.

```
ar_fl_phi_true_age = NaN([size(Phi_true,1),1]);
for it_age_ctr=1:size(Phi_true,1)
  ar_fl_phi_true_age(it_age_ctr,1) = sum(Phi_true(it_age_ctr,:));
end
ar_ages = 18:(18+size(Phi_true,1)-1);
% Graph
figure();
h1 = scatter(ar_ages, ar_fl_phi_true_age);
grid on;
legend1 = sprintf('Distribution of individuals across age groups');
legend({legend1});
xlabel("Ages");
```



Second, given some overall age span from age Xl to Xu, we consider G segments within, and the conditional probability of mass in each of the G segments within Xl to Xu bounds.

```
% Overall prime-age
it_age_min = 18;
it_age_max = 65;
ar_ages_18t65 = ((ar_ages<=it_age_max) & (ar_ages>=it_age_min));
ar_fl_phi_true_age_18t65 = ar_fl_phi_true_age(ar_ages_18t65);
fl_total_mass_18t65 = sum(ar_fl_phi_true_age_18t65);
% Sub-segments
ar_it_lower = [18, 25, 55]';
ar_it_higher = [24, 54, 65]';
ar_fl_p_agegrp_condi_18t65 = NaN([length(ar_it_lower),1]);
```

```
for it_age_ctr=1:length(ar_it_lower)
  it_age_lower = ar_it_lower(it_age_ctr);
  it_age_higher = ar_it_higher(it_age_ctr);
  ar_ages_subgrp = ((ar_ages<=it_age_higher) & (ar_ages>=it_age_lower));
  ar_fl_phi_true_age_subgrp = ar_fl_phi_true_age(ar_ages_subgrp);
  ar_fl_p_agegrp_condi_18t65(it_age_ctr) = sum(ar_fl_phi_true_age_subgrp)/fl_total_mass_18t65;
end
disp(['sum of ar_fl_p_agegrp_condi_18t65=' num2str(sum(ar_fl_p_agegrp_condi_18t65))]);
```

```
sum of ar_fl_p_agegrp_condi_18t65=1
```

```
disp(table(ar_fl_p_agegrp_condi_18t65, ar_it_lower, ar_it_higher));
```

```
    ar_fl_p_agegrp_condi_18t65    ar_it_lower    ar_it_higher
    __________________________    ___________    ____________

             0.18278                  18              24
             0.64041                  25              54
             0.17681                  55              65
```

7.4 Distribution with One Period Policy Shift

This is the example vignette for function: **snw_ds_main_vec** from the **PrjOptiSNW Package.**

7.4.1 One-period Deviation from Steady-State given Alternative Policy Function

In addition to solving for distribution given one policy function, **snw_ds_main_vec** can also solve for the distributional shift from "steady-state" with a one-period policy shift.

If a 6th parameter, PHI_ADJ_BASE, is provided to **snw_ds_main_vec**, solve for next-period forward distribution conditional on PHI_ADJ_BASE, using the policy function provided to **snw_ds_main_vec** as the 3rd and 4th parameters.

When PHI_ADJ_BASE is provided, if the AP_SS, CONS_SS policy functions inputs are from the same problem that generated PHI_ADJ_BASE, output PHI_ADJ will be identical to PHI_ADJ_BASE. However, if AP_SS, CONS_SS are different policy functions from those that induced PHI_ADJ_BASE, PHI_ADJ output will be different from PHI_ADJ_BASE input.

This allows for obtaining the distributional impact of a one period policy, allowing for deviation from "steady-state" distribution. This is used to solve for the distribution after one-period MIT shock, given stimulus checks provided in that period.

This is used to model the distributional effects of CARES Act, the two rounds of Trump Stimulus Checks, on household asset distribution when then receive the Biden stimulus checks from the the American Recovery Act. In effect, we have two MIT shock periods.

7.4.2 Solve for "Steady-State" Policy and Value Functions

Steady-state policy and value functions

```
% mp_params = snw_mp_param('default_dense');
mp_params = snw_mp_param('default_docdense');
% mp_params = snw_mp_param('default_moredense_a65zh133zs5_e2m2');
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
[V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=550.
```

7.4.3 Solve for "Steady-State" Distribution

Solve for steady-state distributions, using steady-state policy functions.

```
[Phi_true_ss,Phi_adj_ss,A_agg_ss,Y_inc_agg_ss,~,mp_dsvfi_results_ss] = ...
    snw_ds_main_vec(mp_params, mp_controls, ap_ss, cons_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=547.
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1786.6361
```

```
% [Phi_true,Phi_adj] = snw_ds_main(mp_params, mp_controls);
Phi_true_ss = Phi_true_ss/sum(Phi_true_ss(:));
```

Show distributional results.

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results_ss('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'));
```

	mean	unweighted_sum	sd	coefofvar	gini	min
a_ss	4.3602	2228	6.8796	1.5778	0.6755	0
ap_ss	4.4621	5.3216e+08	6.9169	1.5501	0.67638	0
cons_ss	1.0635	5.0787e+07	0.6938	0.65237	0.33936	0.036717
n_ss	2.3554	21	1.4375	0.61029	0.3128	1
y_all	1.4661	8.3558e+07	1.4665	1.0003	0.44546	0.038108
y_head_inc	1.1081	1.9253e+06	1.013	0.91419	0.42164	0.038108
y_head_earn	0.88655	19732	0.92804	1.0468	0.53121	0
y_spouse_inc	0.35797	4.827e+05	0.95437	2.6661	0.85269	0
yshr_interest	0.12865	3.8438e+06	0.17577	1.3663	0.65781	0
yshr_wage	0.77402	8.8881e+06	0.33679	0.43512	0.2062	0
yshr_SS	0.097329	29012	0.2266	2.3282	0.91382	0
yshr_tax	0.17833	2.8338e+06	0.035661	0.19998	0.11386	0.036506
yshr_nttxss	0.080996	2.8048e+06	0.24691	3.0485	1.2592	-0.89715

7.4.4 Solve for Policy Function Under Trump Stimulus

Same continuation value as prior (steady-state continuation), but now solve for new policy (one round) due to Trump stimulus. Same tax rate in covid and other years, manna-from-heaven. This calls the **snw_vfi_main_bisec_vec_stimulus** function, which provides the stimulus checks as a function of income and family status.

```
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
[~,ap_trumpchecks,cons_trumpchecks, mp_valpol_more_trumpchecks] = ...
    snw_vfi_main_bisec_vec_stimulus(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
```

7.4.5 Solve for Updated Distribution given Trump Stimulus

Fixing mass at their steady-state distribution, policy functions shift to the Trump stimulus policies, resolve for one-period forward distribution. The distributional code is almost identical, except uses steady-state distribution as the "base" distribution via parameter PHI_ADJ_SS.

```
[Phi_true_trumpchecks,Phi_adj_trumpchecks,...
    A_agg_trumpchecks,Y_inc_agg_trumpchecks,~,mp_dsvfi_results_trumpchecks] = snw_ds_main_vec(...
    mp_params, mp_controls, ...
    ap_trumpchecks, cons_trumpchecks, ...
```

```
    mp_valpol_more_trumpchecks, ...
    Phi_adj_ss);
```

```
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1546.0593
```

```
Phi_true_trumpchecks = Phi_true_trumpchecks/sum(Phi_true_trumpchecks(:));
```

Show distributional results.

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results_trumpchecks('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'));
```

	mean	unweighted_sum	sd	coefofvar	gini	min
a_ss	4.3988	2228	6.8731	1.5625	0.66733	0
ap_ss	4.5367	5.326e+08	6.9044	1.5219	0.66162	0
cons_ss	1.0761	5.0871e+07	0.68968	0.64091	0.33326	0.048012
n_ss	2.3554	21	1.4375	0.61029	0.3128	1
y_all	1.4676	8.3558e+07	1.4664	0.99915	0.44498	0.038108
y_head_inc	1.1097	1.9253e+06	1.0127	0.9126	0.42095	0.038108
y_head_earn	0.88655	19732	0.92804	1.0468	0.53121	0
y_spouse_inc	0.35797	4.827e+05	0.95437	2.6661	0.85269	0
yshr_interest	0.13035	3.8438e+06	0.1754	1.3457	0.64346	0
yshr_wage	0.77264	8.8881e+06	0.33616	0.43508	0.21167	0
yshr_SS	0.097017	29012	0.22582	2.3276	0.91391	0
yshr_tax	0.17842	2.8338e+06	0.035595	0.1995	0.11358	0.036506
yshr_nttxss	0.081403	2.8048e+06	0.24609	3.0231	1.249	-0.89715

7.4.6 Debug Check, SNW_DS_MAIN_VEC with Steady State Policies

This is to confirm that code is working properly. If we use steady-state policy functions and also provide as a sixth parameter the steady-state distribution, PHI_ADJ_SS, to **snw_ds_main_vec**, we should get back the same distribution, PHI_TRUE_SS_WITH_EXISTDIST_DEBUG, which is the same as PHI_ADJ_SS. See that the distributional outputs at the end of this subsection is the same as the distributional table before the table directly prior.

```
[Phi_true_ss_with_existdist_debug,~,~,~,~,mp_dsvfi_results_ss_with_existdist_debug] = snw_ds_main_ve
    mp_params, mp_controls, ...
    ap_ss, cons_ss, ...
    mp_valpol_more_ss, ...
    Phi_adj_ss);
```

```
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1848.7637
```

```
Phi_true_ss_with_existdist_debug = Phi_true_ss_with_existdist_debug/sum(Phi_true_ss_with_existdist_d
```

Show distributional results.

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results_ss_with_existdist_debug('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'));
```

	mean	unweighted_sum	sd	coefofvar	gini	min
a_ss	4.3602	2228	6.8796	1.5778	0.6755	0
ap_ss	4.4621	5.3216e+08	6.9169	1.5501	0.67638	0
cons_ss	1.0635	5.0787e+07	0.6938	0.65237	0.33936	0.036717
n_ss	2.3554	21	1.4375	0.61029	0.3128	1
y_all	1.4661	8.3558e+07	1.4665	1.0003	0.44546	0.038108
y_head_inc	1.1081	1.9253e+06	1.013	0.91419	0.42164	0.038108

y_head_earn	0.88655	19732	0.92804	1.0468	0.53121	0	
y_spouse_inc	0.35797	4.827e+05	0.95437	2.6661	0.85269	0	
yshr_interest	0.12865	3.8438e+06	0.17577	1.3663	0.65781	0	0
yshr_wage	0.77402	8.8881e+06	0.33679	0.43512	0.2062	0	
yshr_SS	0.097329	29012	0.2266	2.3282	0.91382	0	
yshr_tax	0.17833	2.8338e+06	0.035661	0.19998	0.11386	0.036506	
yshr_nttxss	0.080996	2.8048e+06	0.24691	3.0485	1.2592	-0.89715	

Chapter 8

Value of Each Check

8.1 2020 V and C without Unemployment

This is the example vignette for function: **snw_a4chk_wrk_bisec_vec** from the **PrjOptiSNW Package.** This function solves for the V(states, check) for individuals working. Dense solution. Bisection, most time for the test here taken to genereate the income matrixes. But these can be generated out of the check loops.

8.1.1 Test SNW_A4CHK_WRK_BISEC_VEC Defaults Dense

Call the function with default parameters. Solve first for non-covid value and policy. Then depending on 2020 taxes, solve for 2020 policy and value.

```
mp_params = snw_mp_param('default_docdense');
% mp_params = snw_mp_param('default_dense');
mp_params('beta') = 0.95;
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_timer') = true;
[V_ss,~,cons_ss,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=501.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN          sum         mean      std
                _    ___    ____    ________    ____    _________    ___________    _______    _____

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -6.6619e+08    -15.245    21.86
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.3967e+09     31.962    36.42
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05     2.3276e+08     5.3263    8.441

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxxx
              c1         c2         c3         c4         c5         c526496      c526497      c526498
           _______    _______    _______    _______    _______    _________    _________    _________

    r1     -293.96    -293.57    -291.09    -285.44    -276.41     -4.3584      -4.2643      -4.171
    r2     -284.42    -284.03    -281.55    -275.97    -267.24     -4.2519      -4.1612      -4.071
    r3     -274.87    -274.48    -272.03    -266.62    -258.33     -4.1429      -4.0559      -3.969
    r4     -265.22    -264.86    -262.58    -257.53    -249.74     -4.0309      -3.9475      -3.864
    r5     -256.51    -256.17    -254.04     -249.3    -241.96     -3.9252      -3.8452      -3.765
    r79    -13.642    -13.628    -13.535    -13.298    -12.896    -0.22092     -0.21058     -0.2008
```

```
    r80    -12.283    -12.269    -12.176    -11.939    -11.537     -0.16979     -0.16182      -0.154
    r81    -10.605    -10.591    -10.498    -10.261    -9.8589     -0.11712     -0.11163     -0.1064
    r82    -8.3494    -8.3358    -8.2424    -8.0055    -7.6035    -0.065333    -0.062242    -0.0593
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.020968    -0.019972    -0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2        c3           c4           c5       c526496    c526497    c526498    c5264
           --    --    ----------    ---------    --------    -------    -------    -------    -----

    r1     0     0     0.00051498    0.0066578    0.021589    112.13     117.67      123.4     129.3
    r2     0     0     0.00051498    0.0057684    0.020245    112.17     117.71     123.43     129.3
    r3     0     0     0.00020768    0.0041456    0.018539     112.2     117.73     123.45     129.3
    r4     0     0     0.00010346    0.0041199    0.018307    112.86     118.39     124.11     130.0
    r5     0     0     5.2907e-06    0.0041199    0.018091    113.53     119.07     124.79     130.7
    r79    0     0              0            0           0    81.091     85.364     89.335     93.25
    r80    0     0              0            0           0    76.124     79.747     83.431     86.98
    r81    0     0              0            0           0    67.945     70.639     73.673     76.99
    r82    0     0              0            0           0    50.126     53.467     56.302     57.88
    r83    0     0              0            0           0         0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.036717    0.037251    0.040477    0.044486    0.049324    12.265      12.55     12.844
    r2     0.036717    0.037251    0.040477    0.045375    0.050668    12.501     12.787     13.082
    r3     0.036717    0.037251    0.040784    0.046998    0.052374    12.755     13.042     13.337
    r4     0.038144    0.038678    0.042314    0.048449    0.054031        13     13.289     13.584
    r5     0.039534    0.040068    0.043802    0.049839    0.055635    13.236     13.525     13.821
    r79     0.19737     0.19791     0.20163     0.21175     0.23145    35.811     37.362     39.409
    r80     0.19737     0.19791     0.20163     0.21175     0.23145    40.752     42.953     45.286
    r81     0.19737     0.19791     0.20163     0.21175     0.23145    48.909     52.039     55.022
    r82     0.19737     0.19791     0.20163     0.21175     0.23145     66.71     69.193     72.375
    r83     0.19737     0.19791     0.20163     0.21175     0.23145    116.82     122.65     128.66
```

```
welf_checks = 2; % 2 checks is $200 dollar of welfare checks
xi=1; % xi=0 full income loss from covid shock, xi=1, no covid income losses
b=1; % when xi=1, b does not matter, no income losses
TR = 100/58056;
mp_params('TR') = TR;
mp_params('xi') = xi;
mp_params('b') = b;
% if = mp_params('a2_covidyr_manna_heaven'), V_emp_2020 same as V_ss if b=1
% or xi=1.
% if = mp_params('a2_covidyr_tax_fully_pay'), V_emp_2020 differ due to 2020
% tax differences
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
% mp_params('a2_covidyr') = mp_params('a2_covidyr_tax_fully_pay');
[V_emp_2020,~,cons_emp_2020,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN          sum         mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----
```

```
    V_VFI       1     1     6      4.37e+07     83     5.265e+05    -6.6619e+08    -15.245    21.86
    ap_VFI      2     2     6      4.37e+07     83     5.265e+05     1.3967e+09     31.962    36.42
    cons_VFI    3     3     6      4.37e+07     83     5.265e+05     2.3276e+08     5.3263    8.441

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    --------

    r1     -293.96    -293.57    -291.09    -285.44    -276.41      -4.3584      -4.2643      -4.171
    r2     -284.42    -284.03    -281.55    -275.97    -267.24      -4.2519      -4.1612      -4.071
    r3     -274.87    -274.48    -272.03    -266.62    -258.33      -4.1429      -4.0559      -3.969
    r4     -265.22    -264.86    -262.58    -257.53    -249.74      -4.0309      -3.9475      -3.864
    r5     -256.51    -256.17    -254.04     -249.3    -241.96      -3.9252      -3.8452      -3.765
    r79    -13.642    -13.628    -13.535    -13.298    -12.896     -0.22092     -0.21058     -0.2008
    r80    -12.283    -12.269    -12.176    -11.939    -11.537     -0.16979     -0.16182     -0.154
    r81    -10.605    -10.591    -10.498    -10.261    -9.8589     -0.11712     -0.11163     -0.1064
    r82    -8.3494    -8.3358    -8.2424    -8.0055    -7.6035    -0.065333    -0.062242    -0.0593
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.020968    -0.019972    -0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2        c3            c4           c5        c526496    c526497    c526498    c5264
           --    --    ----------    ---------    --------    -------    -------    -------    -----

    r1     0     0     0.00051498    0.0066578    0.021589     112.13     117.67      123.4      129.3
    r2     0     0     0.00051498    0.0057684    0.020245     112.17     117.71     123.43      129.3
    r3     0     0     0.00020768    0.0041456    0.018539      112.2     117.73     123.45      129.3
    r4     0     0     0.00010346    0.0041199    0.018307     112.86     118.39     124.11      130.0
    r5     0     0     5.2907e-06    0.0041199    0.018091     113.53     119.07     124.79      130.7
    r79    0     0              0            0           0     81.091     85.364     89.335      93.25
    r80    0     0              0            0           0     76.124     79.747     83.431      86.98
    r81    0     0              0            0           0     67.945     70.639     73.673      76.99
    r82    0     0              0            0           0     50.126     53.467     56.302      57.88
    r83    0     0              0            0           0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5        c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.036717    0.037251    0.040477    0.044486    0.049324     12.265      12.55     12.844
    r2     0.036717    0.037251    0.040477    0.045375    0.050668     12.501     12.787     13.082
    r3     0.036717    0.037251    0.040784    0.046998    0.052374     12.755     13.042     13.337
    r4     0.038144    0.038678    0.042314    0.048449    0.054031         13     13.289     13.584
    r5     0.039534    0.040068    0.043802    0.049839    0.055635     13.236     13.525     13.821
    r79     0.19737     0.19791     0.20163     0.21175     0.23145     35.811     37.362     39.409
    r80     0.19737     0.19791     0.20163     0.21175     0.23145     40.752     42.953     45.286
    r81     0.19737     0.19791     0.20163     0.21175     0.23145     48.909     52.039     55.022
    r82     0.19737     0.19791     0.20163     0.21175     0.23145      66.71     69.193     72.375
    r83     0.19737     0.19791     0.20163     0.21175     0.23145     116.82     122.65     128.66

[V_W_2020, C_W_2020] = snw_a4chk_wrk_bisec_vec(welf_checks, V_emp_2020, cons_emp_2020, mp_params, mp

Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=st_biden_or_trump_undefined;welf_checks=2;TR=0.001722
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_container_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                      i    idx    ndim     numel      rowN      colN          sum          mean
                      -    ---    ----    --------    ----    ---------    -----------    ----------
```

```
    C_W               1     1      6       4.37e+07      83      5.265e+05       2.3278e+08          5.3269
    C_W_minus_C_ss    2     2      6       4.37e+07      83      5.265e+05            25096      0.00057428
    V_W               3     3      6       4.37e+07      83      5.265e+05      -6.6561e+08         -15.231
    V_W_minus_V_ss    4     4      6       4.37e+07      83      5.265e+05       5.8108e+05        0.013297
    mn_MPC            5     5      6       4.37e+07      83      5.265e+05       7.2848e+06          0.1667
```

```
mn_V_W_gain_check = V_W_2020 - V_emp_2020;
mn_MPC_W_gain_share_check = (C_W_2020 - cons_emp_2020)./(welf_checks*mp_params('TR'));
```

8.1.2 Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

8.1.3 Analyze Difference in V and C with Check

The difference between V and V with Check, marginal utility gain given the check.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(MN_V_GAIN_CHECK(A,Z))

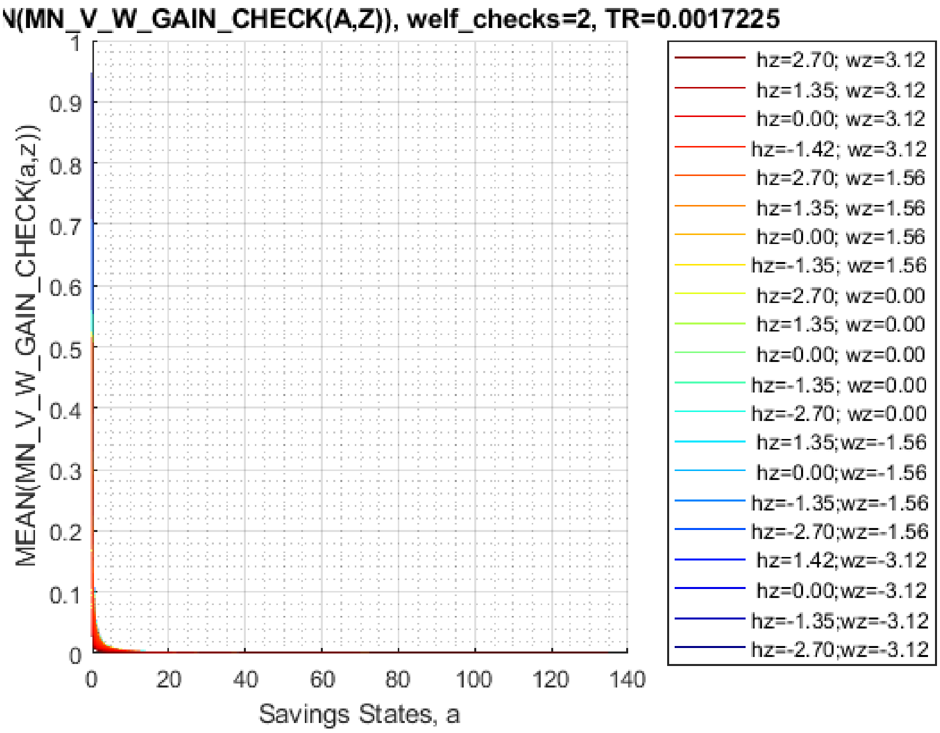
Tabulate value and policies along savings and shocks:

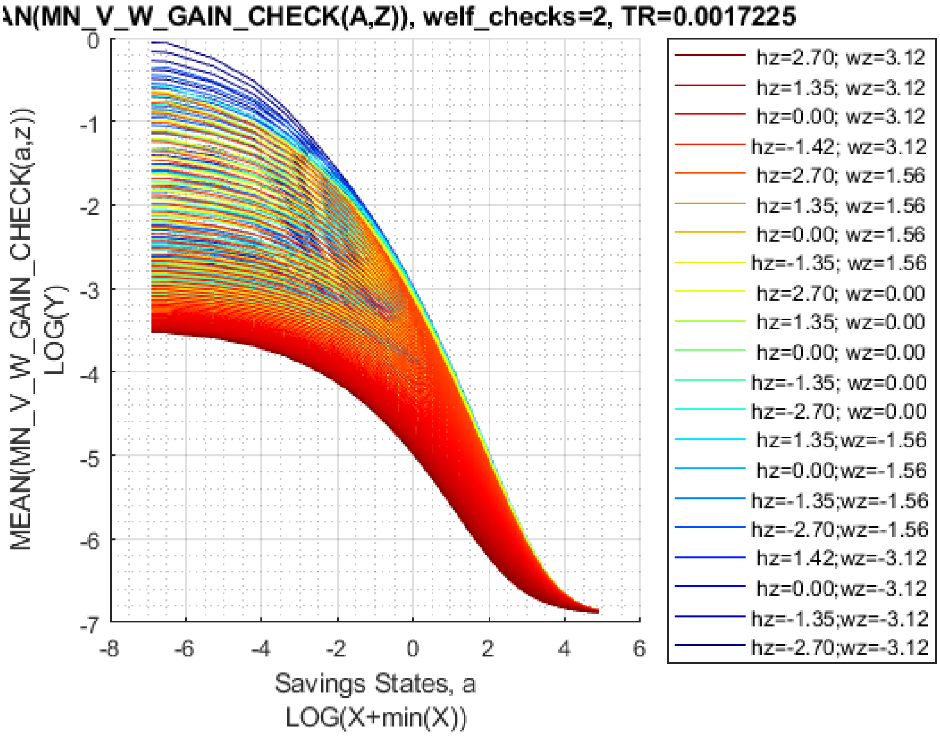
```
% Set
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_W_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_par
tb_az_v = ff_summ_nd_array(st_title, mn_V_W_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_
```

```
xxx  MEAN(MN_V_W_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0       0.94877       0.84952         0.761       0.68204       0.61169
```

```
3        0.0041199        0.78107        0.71106        0.6451        0.58402        0.52829
```

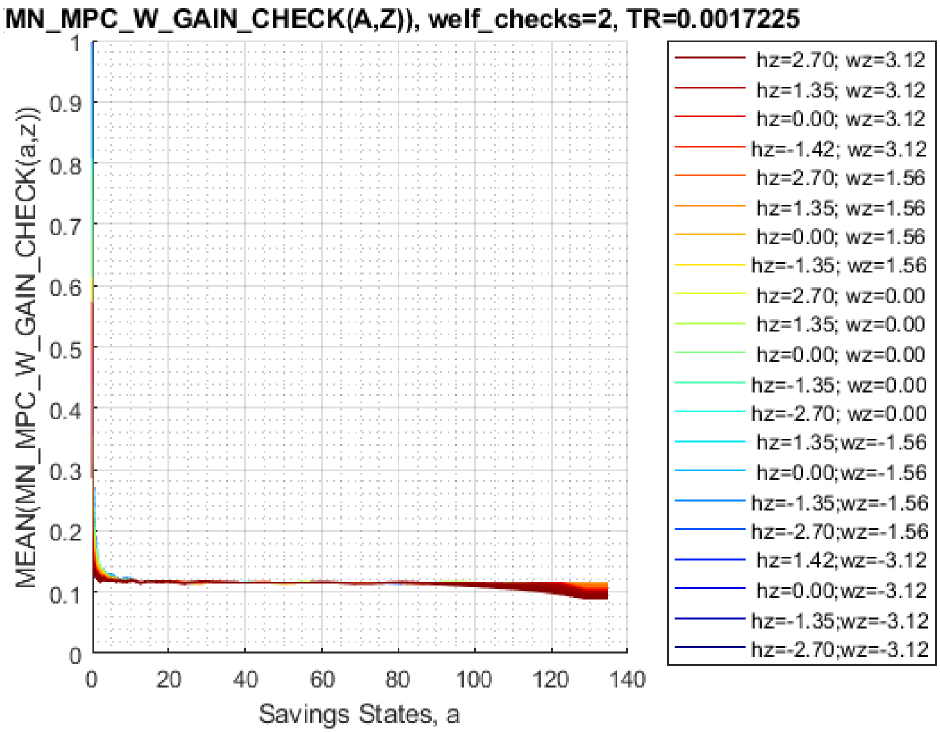
```
st_title = ['MEAN(MN\_V\_W\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_W\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

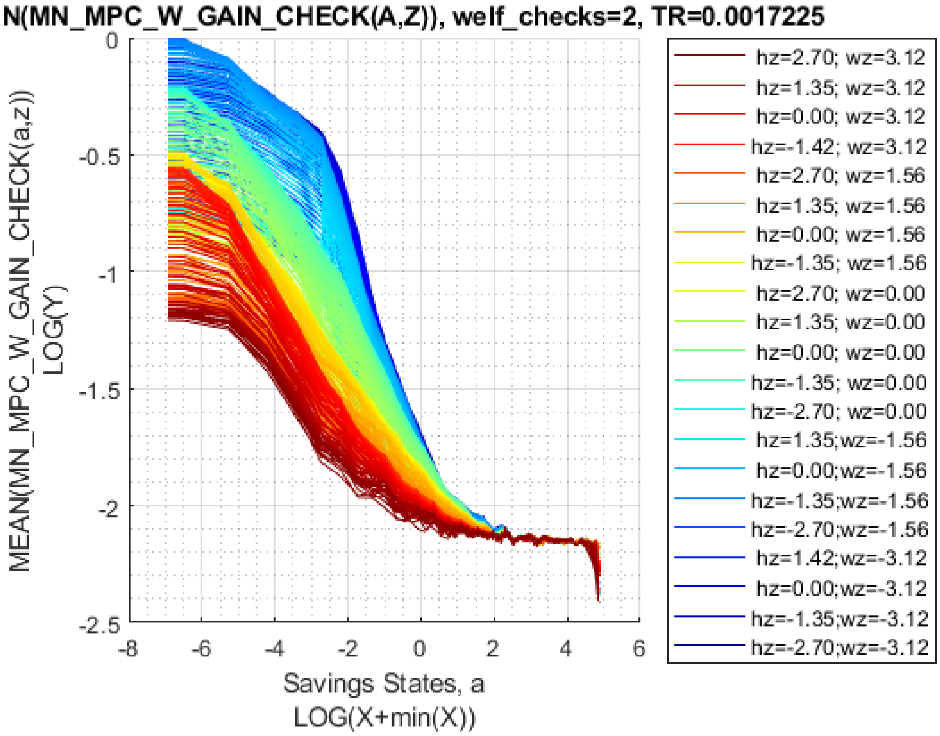




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_W\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_W\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





8.1.4 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
st_title = ['MEAN(MN_V_W_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_W_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_W_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0       0.028443       0.027382        0.02607       0.023829       0.021959
```

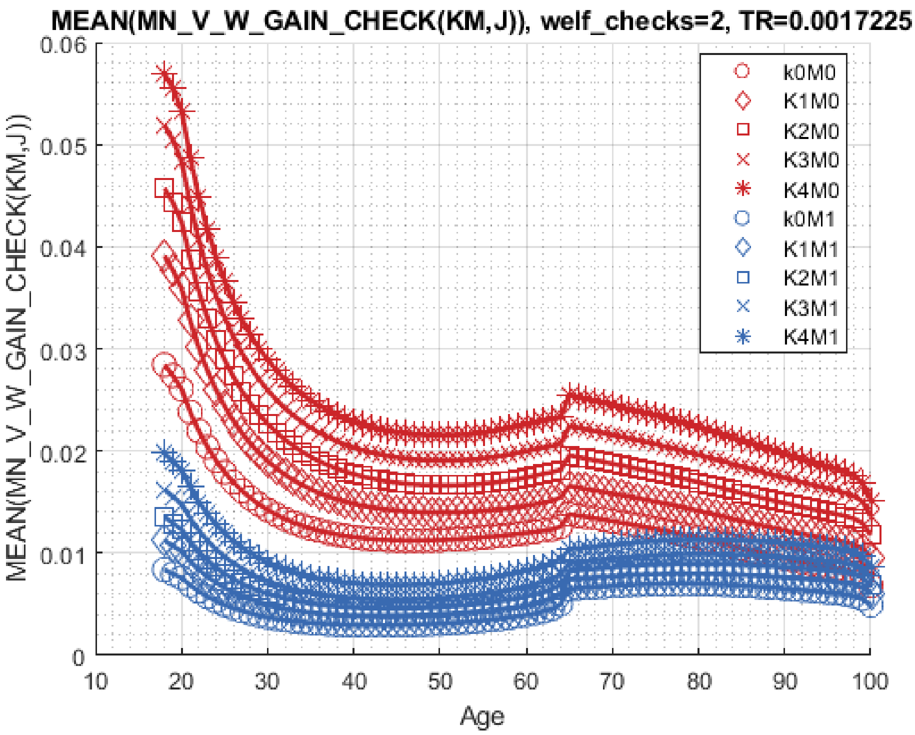
```
       2        2        0       0.039131       0.037712       0.035894       0.032743       0.030106
       3        3        0        0.04572        0.04432        0.04241       0.038719       0.035631
       4        4        0       0.051937       0.050449       0.048354       0.044164       0.040661
       5        5        0       0.056986         0.0555       0.053326        0.04875       0.044927
       6        1        1       0.008385      0.0079795      0.0075874      0.0068616      0.0062549
       7        2        1       0.011253       0.010708       0.010181      0.0092041      0.0083817
       8        3        1       0.013554       0.012928       0.012313       0.011136       0.010147
       9        4        1       0.016251       0.015529       0.014803       0.013404       0.012225
      10        5        1       0.019768       0.018969       0.018139       0.016444       0.015026
```

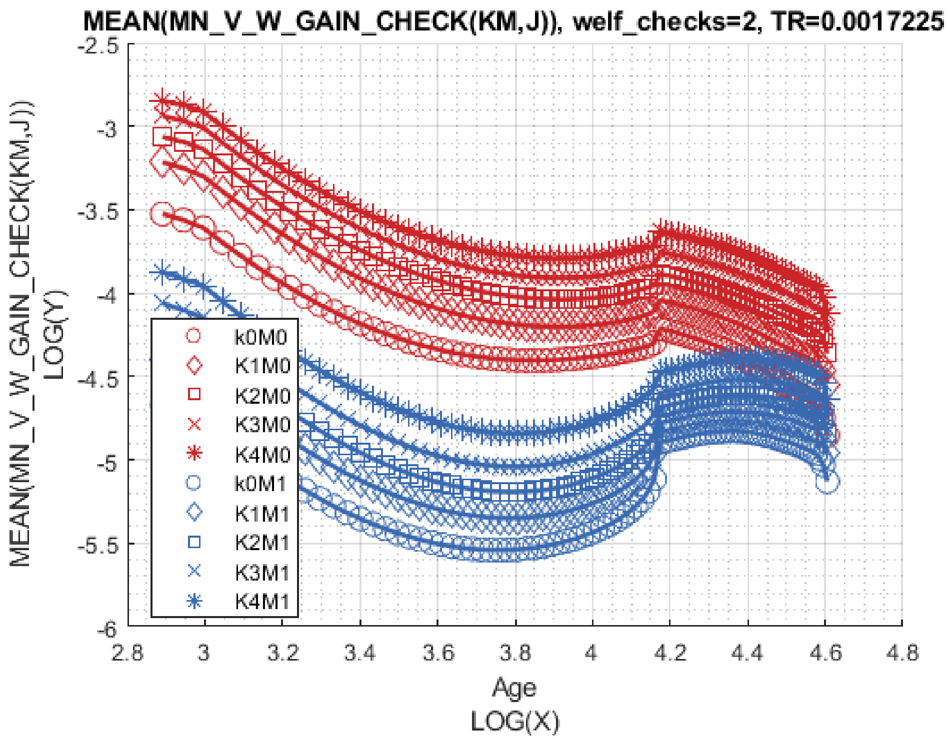
```
% Consumption Function
st_title = ['MEAN(MN_MPC_W_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_W_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd
```

```
xxx  MEAN(MN_MPC_W_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1        0       0.067542       0.074752       0.091075       0.088909       0.086942
      2        2        0       0.075256       0.083119        0.10165       0.099284       0.097581
      3        3        0       0.086542       0.095859        0.11593        0.11256        0.10948
      4        4        0       0.091496        0.10076        0.12129        0.11824        0.11514
      5        5        0       0.098346        0.10645        0.12728        0.12409        0.12073
      6        1        1        0.10277        0.10672         0.1125        0.11137        0.11019
      7        2        1        0.10343         0.1077        0.11433        0.11354        0.11208
      8        3        1        0.10875        0.11374        0.12309        0.11975        0.11861
      9        4        1        0.11014        0.11556        0.12324        0.12217        0.12228
     10        5        1         0.1166         0.1232        0.13246        0.13017        0.12661
```

Graph Mean Values:

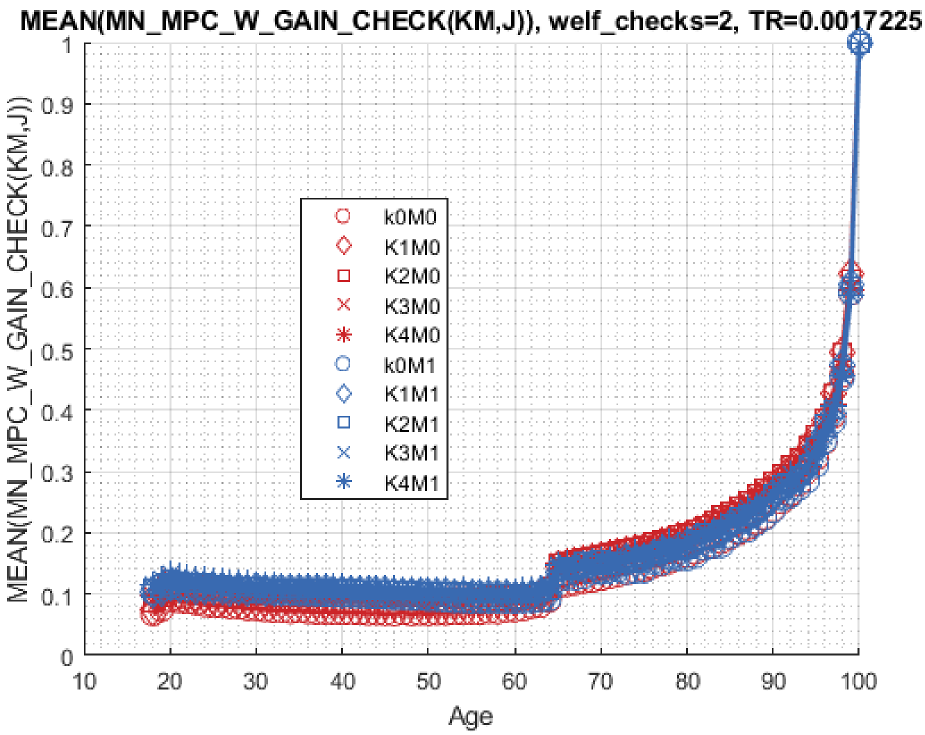
```
st_title = ['MEAN(MN\_V\_W\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_W\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

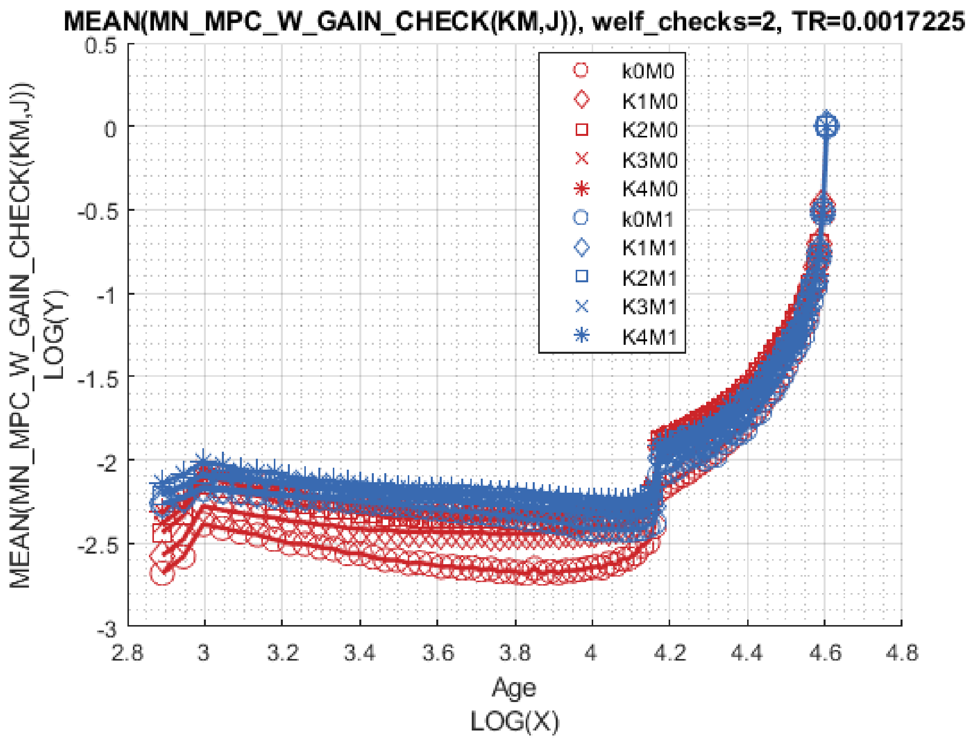




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_W\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_W\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





8.1.5 Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

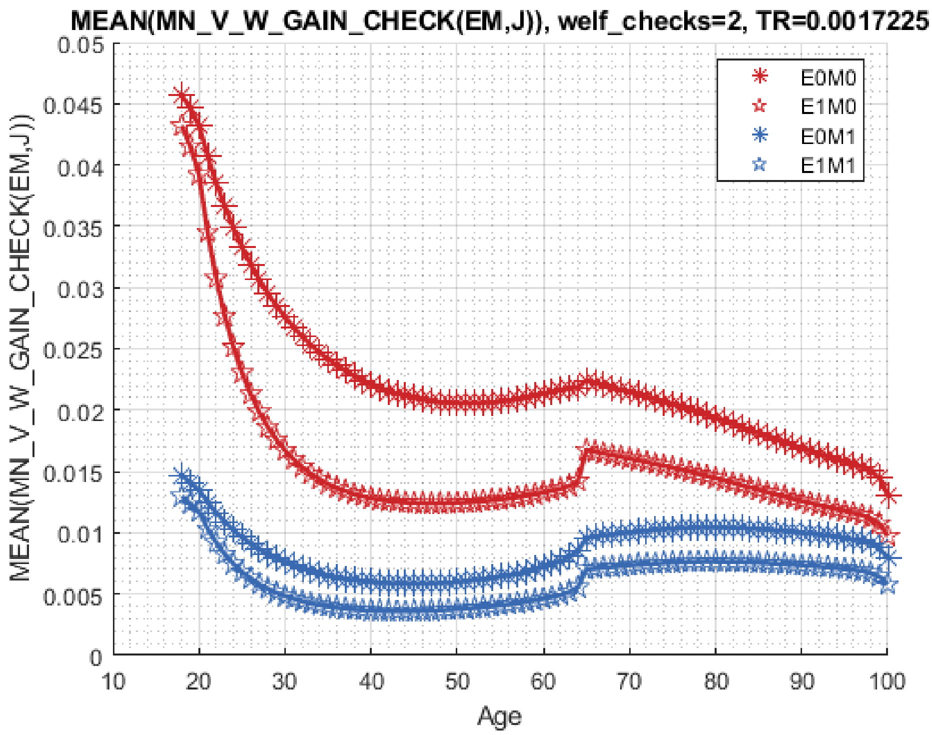
Tabulate value and policies:

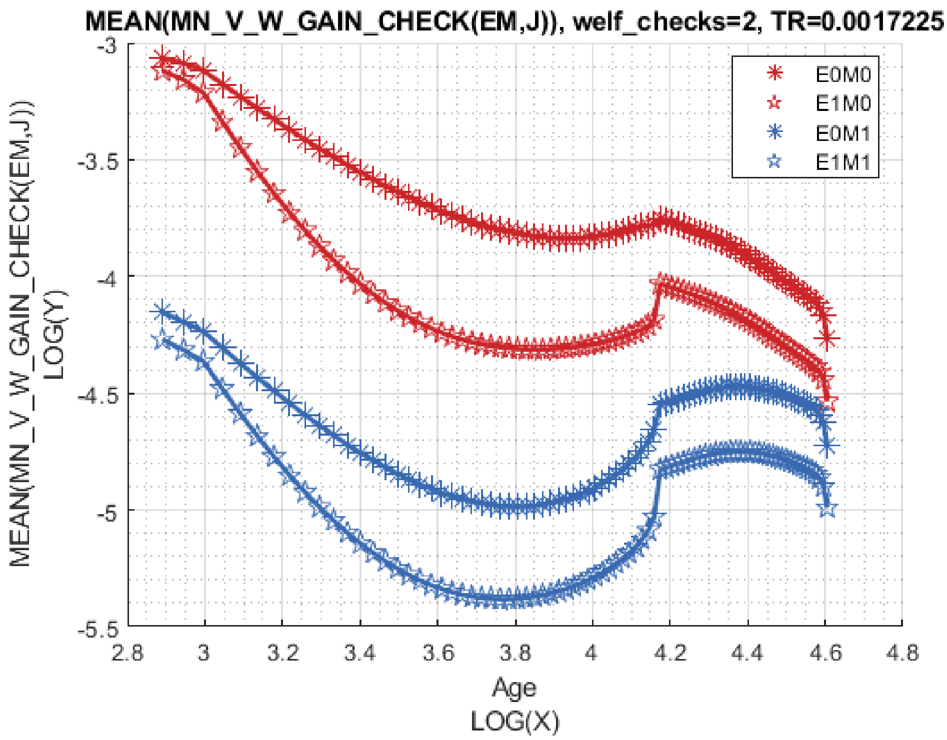
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_W_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_W_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_W_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0       0.045692       0.044619       0.043207       0.040746        0.03856
      2       1       0       0.043194       0.041526       0.039215       0.034536       0.030754
      3       0       1       0.014697       0.014079        0.01347       0.012491        0.01163
      4       1       1       0.012987       0.012367       0.011739       0.010329       0.009184

% Consumption
st_title = ['MEAN(MN_MPC_W_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
```

```
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_W_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd

xxx  MEAN(MN_MPC_W_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0       0.075296       0.080407       0.092505       0.091671       0.091522
      2       1       0       0.092377        0.10397        0.13038        0.12556        0.12042
      3       0       1       0.099842        0.10362        0.10816        0.10847        0.10824
      4       1       1        0.11684        0.12315        0.13408        0.13033        0.12767
```

Graph Mean Values:

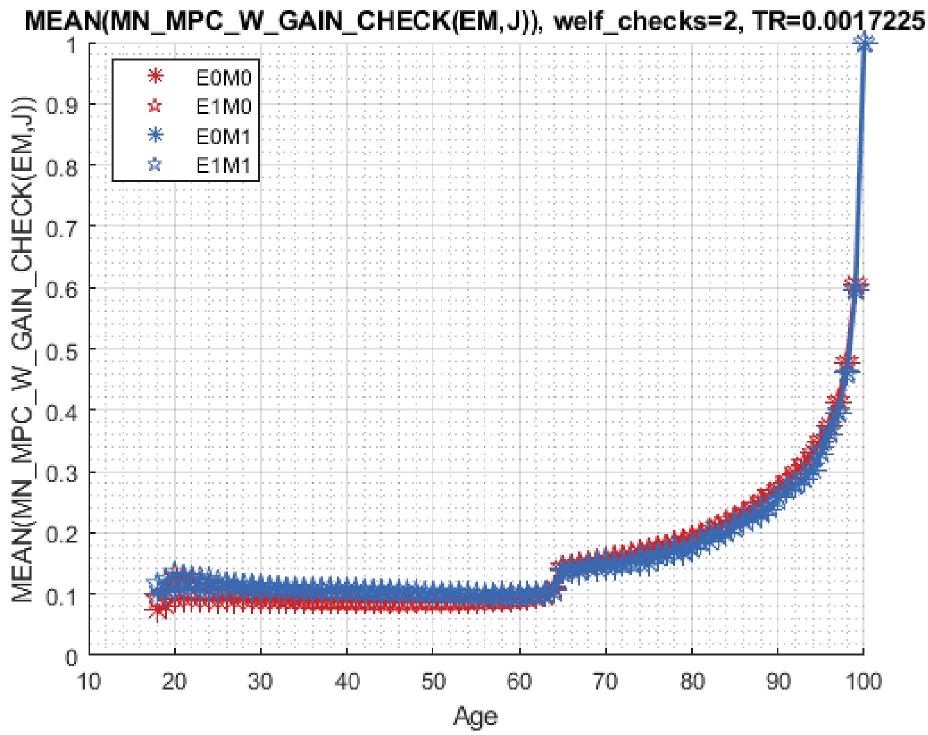
```
st_title = ['MEAN(MN\_V\_W\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_W\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

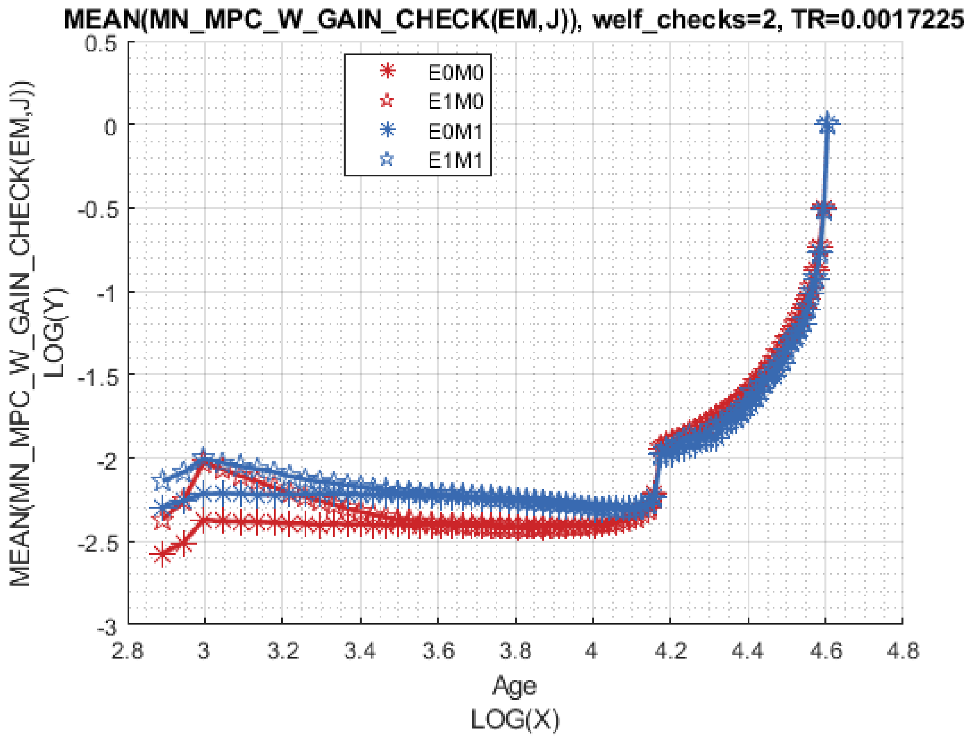




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_W\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_W\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





8.2 2020 V and C with Unemployment

This is the example vignette for function: **snw_a4chk_unemp_bisec_vec** from the **PrjOptiSNW Package.** This function solves for the V(states, check) for individuals working. Dense solution. Bisection, most time for the test here taken to genereate the income matrixes. But these can be generated out of the check loops.

8.2.1 Test SNW_A4CHK_UNEMP_BISEC_VEC Defaults

Solve for Value/Policy in non-COVID years, then solve for covid year value/policy given covid shocks. COVID lasts one period.

```
mp_params = snw_mp_param('default_docdense', false, 'tauchen', true);
mp_params('beta') = 0.95;
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_timer') = true;
[V_ss,~,cons_ss,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=503.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN          sum          mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -6.6619e+08    -15.245    21.86
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.3967e+09     31.962    36.42
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05     2.3276e+08     5.3263    8.441

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    --------
```

```
    r1     -293.96    -293.57    -291.09    -285.44    -276.41       -4.3584       -4.2643       -4.171
    r2     -284.42    -284.03    -281.55    -275.97    -267.24       -4.2519       -4.1612       -4.071
    r3     -274.87    -274.48    -272.03    -266.62    -258.33       -4.1429       -4.0559       -3.969
    r4     -265.22    -264.86    -262.58    -257.53    -249.74       -4.0309       -3.9475       -3.864
    r5     -256.51    -256.17    -254.04     -249.3    -241.96       -3.9252       -3.8452       -3.765
    r79    -13.642    -13.628    -13.535    -13.298    -12.896      -0.22092      -0.21058      -0.2008
    r80    -12.283    -12.269    -12.176    -11.939    -11.537      -0.16979      -0.16182      -0.154
    r81    -10.605    -10.591    -10.498    -10.261    -9.8589      -0.11712      -0.11163      -0.1064
    r82    -8.3494    -8.3358    -8.2424    -8.0055    -7.6035     -0.065333     -0.062242      -0.0593
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206     -0.020968     -0.019972     -0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2        c3            c4            c5        c526496    c526497    c526498    c5264
           --    --    ----------    ---------    --------    -------    -------    -------    -----

    r1     0     0    0.00051498    0.0066578    0.021589     112.13     117.67      123.4      129.3
    r2     0     0    0.00051498    0.0057684    0.020245     112.17     117.71     123.43      129.3
    r3     0     0    0.00020768    0.0041456    0.018539      112.2     117.73     123.45      129.3
    r4     0     0    0.00010346    0.0041199    0.018307     112.86     118.39     124.11      130.0
    r5     0     0    5.2907e-06    0.0041199    0.018091     113.53     119.07     124.79      130.7
    r79    0     0             0            0           0     81.091     85.364     89.335      93.25
    r80    0     0             0            0           0     76.124     79.747     83.431      86.98
    r81    0     0             0            0           0     67.945     70.639     73.673      76.99
    r82    0     0             0            0           0     50.126     53.467     56.302      57.88
    r83    0     0             0            0           0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.036717    0.037251    0.040477    0.044486    0.049324    12.265      12.55     12.844
    r2     0.036717    0.037251    0.040477    0.045375    0.050668    12.501     12.787     13.082
    r3     0.036717    0.037251    0.040784    0.046998    0.052374    12.755     13.042     13.337
    r4     0.038144    0.038678    0.042314    0.048449    0.054031        13     13.289     13.584
    r5     0.039534    0.040068    0.043802    0.049839    0.055635    13.236     13.525     13.821
    r79     0.19737     0.19791     0.20163     0.21175     0.23145    35.811     37.362     39.409
    r80     0.19737     0.19791     0.20163     0.21175     0.23145    40.752     42.953     45.286
    r81     0.19737     0.19791     0.20163     0.21175     0.23145    48.909     52.039     55.022
    r82     0.19737     0.19791     0.20163     0.21175     0.23145     66.71     69.193     72.375
    r83     0.19737     0.19791     0.20163     0.21175     0.23145    116.82     122.65     128.66
```

```
welf_checks = 2; % 2 checks is $200 dollar of welfare checks
xi=0.5; % xi=0 full income loss from covid shock, xi=1, no covid income losses
b=0; % b=0 means no UI benefits compensating COVID, b=1 if full income replacement
TR = 100/58056;
mp_params('TR') = TR;
mp_params('xi') = xi;
mp_params('b') = b;
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
% mp_params('a2_covidyr') = mp_params('a2_covidyr_tax_fully_pay');
[V_unemp_2020,~,cons_unemp_2020,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
                 i    idx    ndim     numel      rowN      colN          sum         mean       std
                 -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI        1     1      6      4.37e+07     83     5.265e+05    -6.8822e+08    -15.749    22.87
    ap_VFI       2     2      6      4.37e+07     83     5.265e+05     1.3605e+09     31.134    36.29
    cons_VFI     3     3      6      4.37e+07     83     5.265e+05     2.2887e+08     5.2375    8.443

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
              c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    --------

    r1     -320.42    -318.92    -310.39    -296.97    -284.58      -4.4406      -4.3429      -4.246
    r2     -310.88    -309.38    -300.85    -287.43    -275.14      -4.3331       -4.239      -4.146
    r3     -301.33    -299.83     -291.3    -277.88    -265.85      -4.2231      -4.1327      -4.043
    r4     -290.68    -289.29    -281.32     -268.6     -257.1      -4.1145      -4.0276      -3.941
    r5     -281.05    -279.76    -272.29     -260.2    -249.16      -4.0121      -3.9284      -3.845
    r79    -13.642    -13.628    -13.535    -13.298    -12.896     -0.22291     -0.21238     -0.2024
    r80    -12.283    -12.269    -12.176    -11.939    -11.537     -0.17128     -0.16316     -0.1555
    r81    -10.605    -10.591    -10.498    -10.261    -9.8589     -0.11815     -0.11254     -0.1072
    r82    -8.3494    -8.3358    -8.2424    -8.0055    -7.6035    -0.065887    -0.062757    -0.05982
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.021146    -0.020134    -0.01918

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2    c3    c4        c5        c526496    c526497    c526498    c526499    c526500
           --    --    --    --    ---------    -------    -------    -------    -------    -------

    r1     0     0     0     0     0.0083625    107.54     113.09     118.82     124.74     130.86
    r2     0     0     0     0     0.0074731    107.45     112.99     118.72     124.64     130.75
    r3     0     0     0     0     0.0058503    107.33     112.88     118.61     124.52     130.64
    r4     0     0     0     0     0.0049981    107.54     113.08     118.81     124.73     130.85
    r5     0     0     0     0      0.004174    107.76      113.3     119.03     124.95     131.07
    r79    0     0     0     0             0    80.462      84.34     88.311     92.234     96.324
    r80    0     0     0     0             0    75.113     78.736      82.42     85.975     90.439
    r81    0     0     0     0             0    66.945     69.639     72.673     76.669     81.091
    r82    0     0     0     0             0    50.126     53.467     55.311     56.953     60.587
    r83    0     0     0     0             0         0          0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
               c1          c2          c3          c4          c5        c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.018623    0.019158    0.022901    0.033062    0.044486    11.989     12.265      12.55
    r2     0.018623    0.019158    0.022901    0.033062    0.045375    12.223     12.501     12.787
    r3     0.018623    0.019158    0.022901    0.033062    0.046998    12.476     12.755     13.042
    r4     0.019354    0.019888    0.023632    0.033792    0.048579     12.72         13     13.289
    r5     0.020066    0.020601    0.024344    0.034504    0.050114    12.955     13.236     13.525
    r79     0.19737     0.19791     0.20163     0.21175     0.23145    35.417     37.362     39.409
    r80     0.19737     0.19791     0.20163     0.21175     0.23145    40.752     42.953     45.286
    r81     0.19737     0.19791     0.20163     0.21175     0.23145    48.909     52.039     55.022
    r82     0.19737     0.19791     0.20163     0.21175     0.23145    65.719     68.202     72.375
    r83     0.19737     0.19791     0.20163     0.21175     0.23145    115.84     121.66     127.68

[V_U_2020, C_U_2020] = snw_a4chk_unemp_bisec_vec(welf_checks, V_unemp_2020, cons_unemp_2020, mp_para

Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=2;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_docde
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_container_map ND Array (Matrix etc)
```

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                        i    idx    ndim     numel      rowN      colN           sum          mean
                        -    ---    ----    --------    ----    ---------    -----------    -------

    C_U                 1     1      6      4.37e+07     83     5.265e+05     2.2891e+08        5.2
    C_U_minus_C_unemp   2     2      6      4.37e+07     83     5.265e+05          31259    0.00071
    V_U                 3     3      6      4.37e+07     83     5.265e+05    -6.8723e+08       -15.
    V_U_minus_V_unemp   4     4      6      4.37e+07     83     5.265e+05     9.9746e+05      0.022
    mn_MPC_unemp        5     5      6      4.37e+07     83     5.265e+05     9.0737e+06       0.20
```

```
mn_V_U_gain_check = V_U_2020 - V_unemp_2020;
mn_MPC_U_gain_share_check = (C_U_2020 - cons_unemp_2020)./(welf_checks*mp_params('TR'));
```

8.2.2 Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

8.2.3 Analyze Difference in V and C with Check

The difference between V and V with Check, marginal utility gain given the check.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(MN_V_GAIN_CHECK(A,Z))

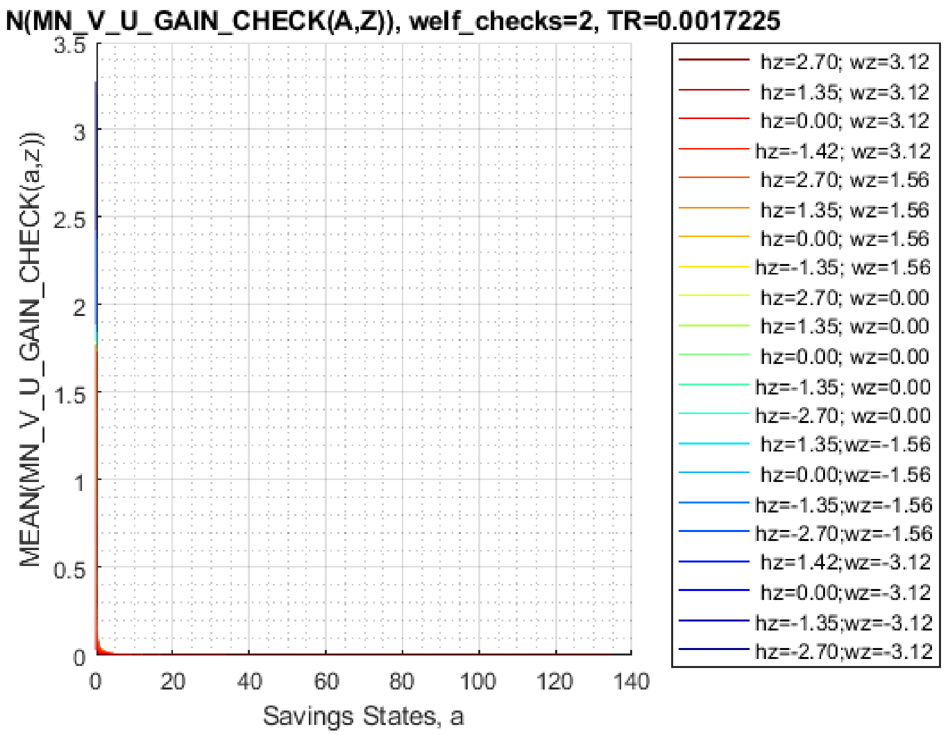
Tabulate value and policies along savings and shocks:

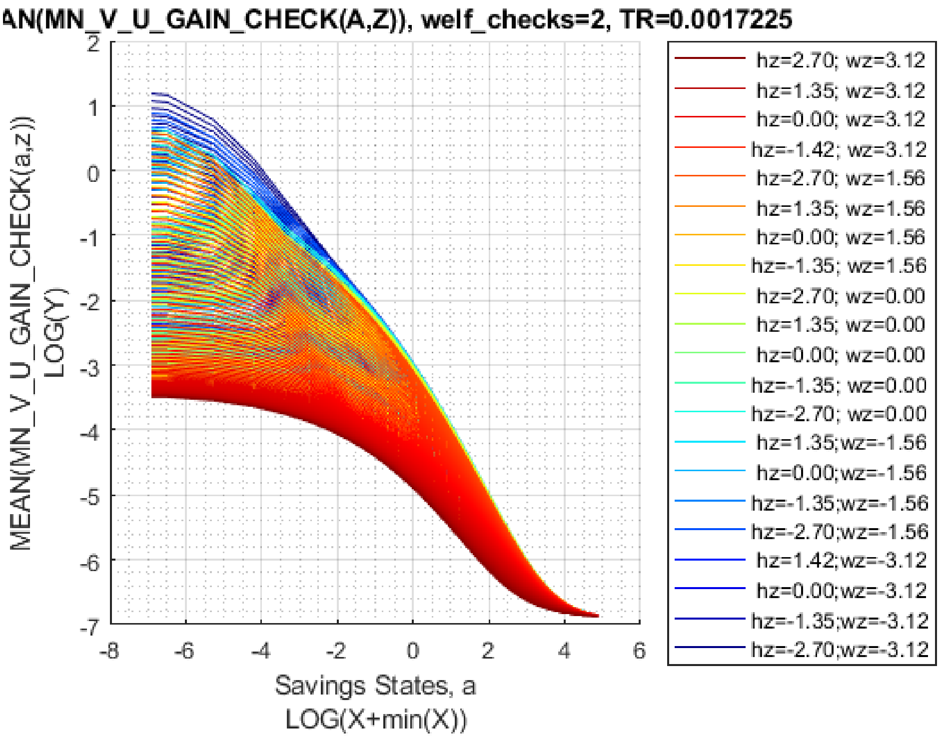
```
% Set
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_par
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_
```

```
xxx  MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---
```

```
     1                0        3.2784       2.9257        2.61       2.3278       2.0757
```

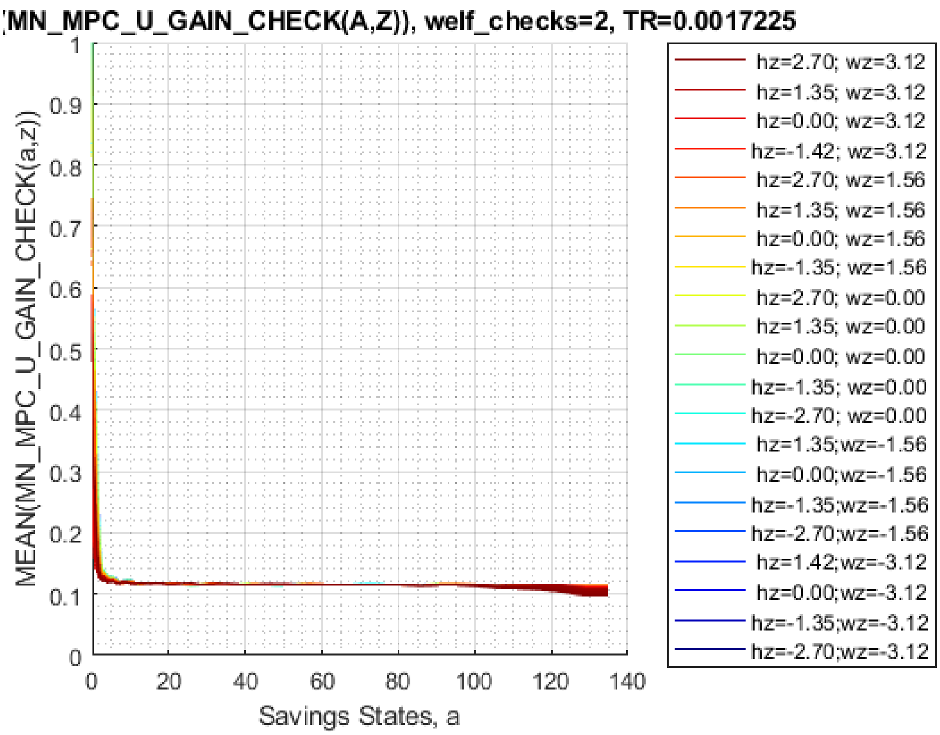
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

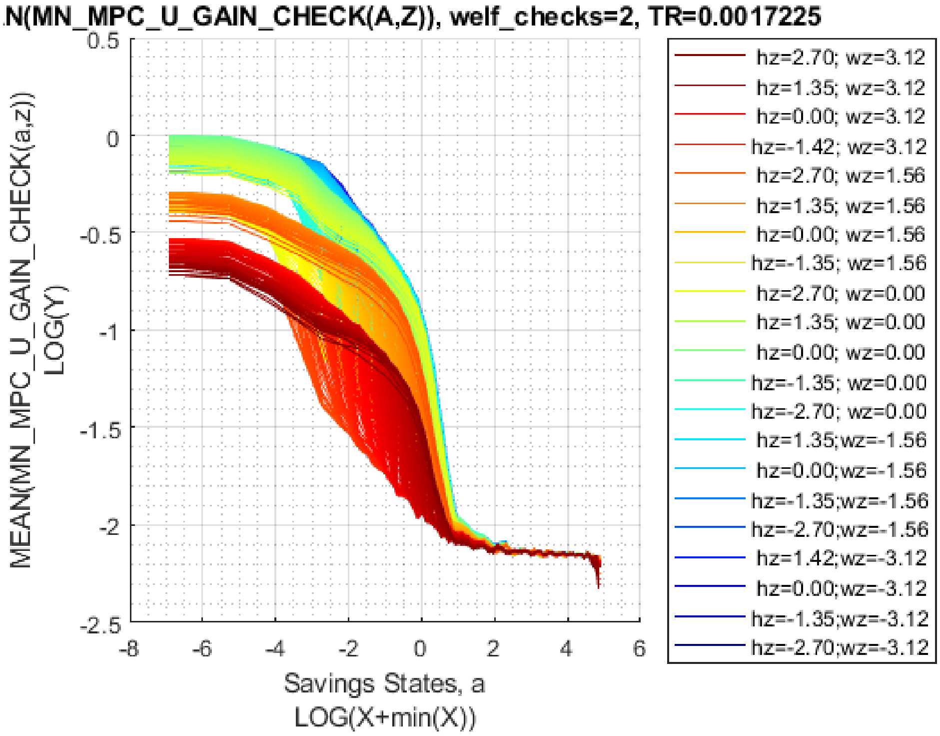




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





8.2.4 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0         0.05653       0.055715       0.054794        0.04982       0.045679
```

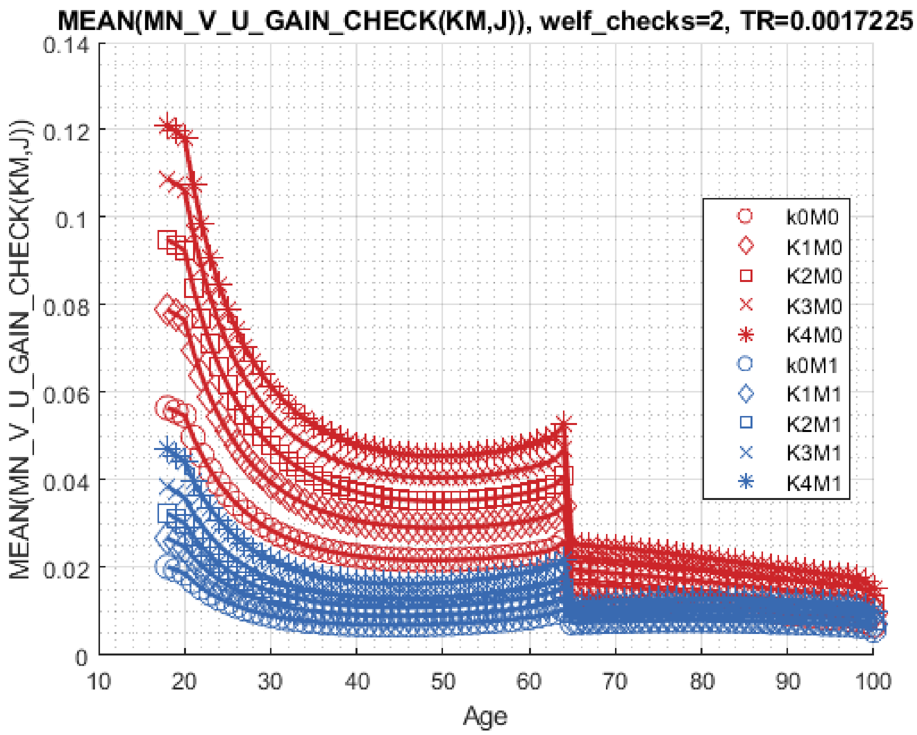
```
    2      2     0    0.078925    0.077839    0.076575    0.069564    0.063719
    3      3     0    0.094952    0.093819     0.09245    0.084005    0.076969
    4      4     0     0.10891      0.1077      0.1062     0.09651    0.088441
    5      5     0     0.12087     0.11965     0.11809     0.10735    0.098413
    6      1     1    0.020237    0.019467    0.018733    0.016911    0.015385
    7      2     1    0.026775    0.025778    0.024831     0.02242    0.020395
    8      3     1    0.032414    0.031263    0.030161    0.027239    0.024791
    9      4     1    0.038629    0.037309    0.036028    0.032548    0.029638
   10      5     1    0.047127    0.045665    0.044235    0.039997     0.03645
```

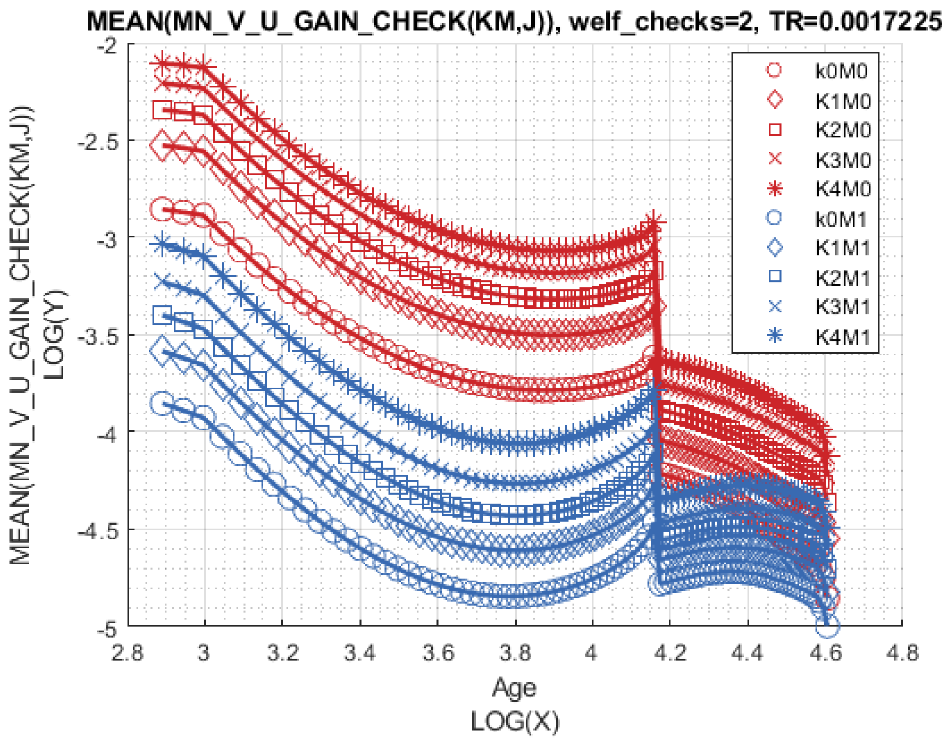
```
% Consumption Function
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd
```

```
xxx  MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        0.16534        0.16905        0.17306        0.17381        0.17437
      2        2       0         0.1737        0.17744        0.18164        0.18288        0.18395
      3        3       0        0.18114        0.18464        0.18867        0.18992        0.19103
      4        4       0        0.18485        0.18821        0.19215        0.19339        0.19447
      5        5       0        0.18839        0.19152        0.19526        0.19638        0.19735
      6        1       1        0.16189        0.16482         0.1704        0.16811        0.17036
      7        2       1        0.16399         0.1672         0.1718        0.17073        0.17219
      8        3       1        0.16993        0.17298        0.17649        0.17652        0.17818
      9        4       1        0.17337        0.17752        0.17974        0.17989        0.18201
     10        5       1        0.18363        0.18473        0.18799        0.18953        0.18894
```

Graph Mean Values:

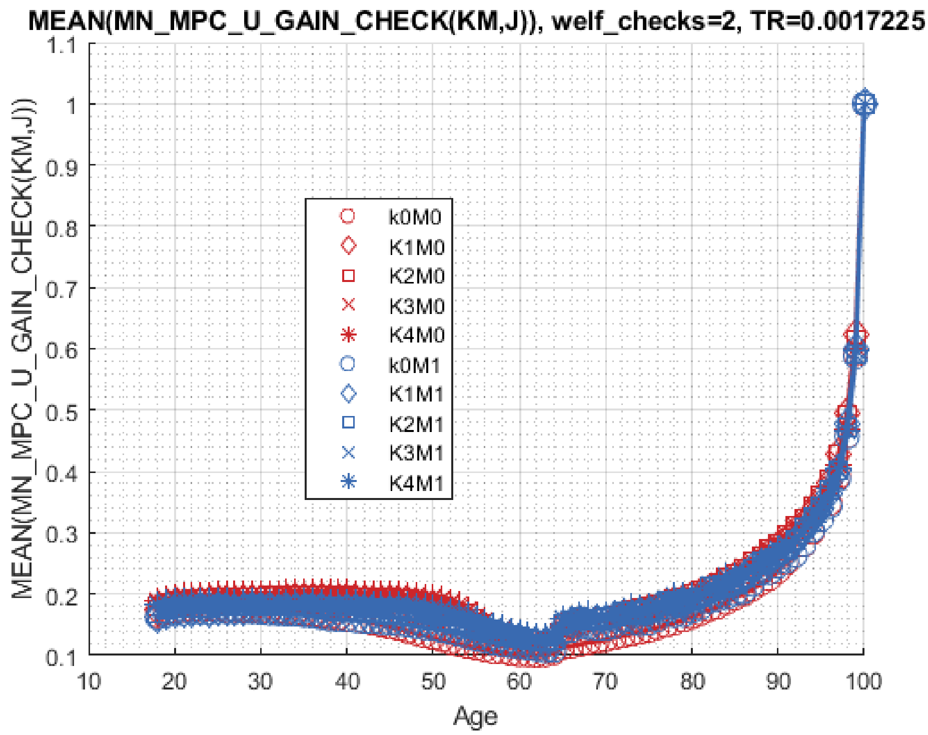
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

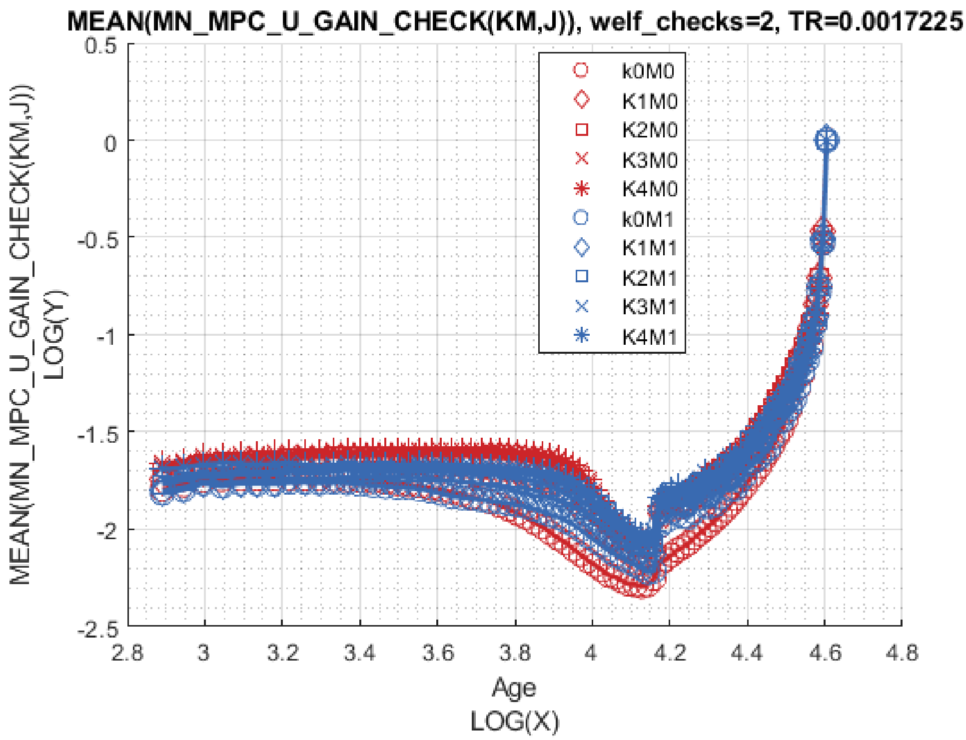




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





8.2.5 Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

Tabulate value and policies:

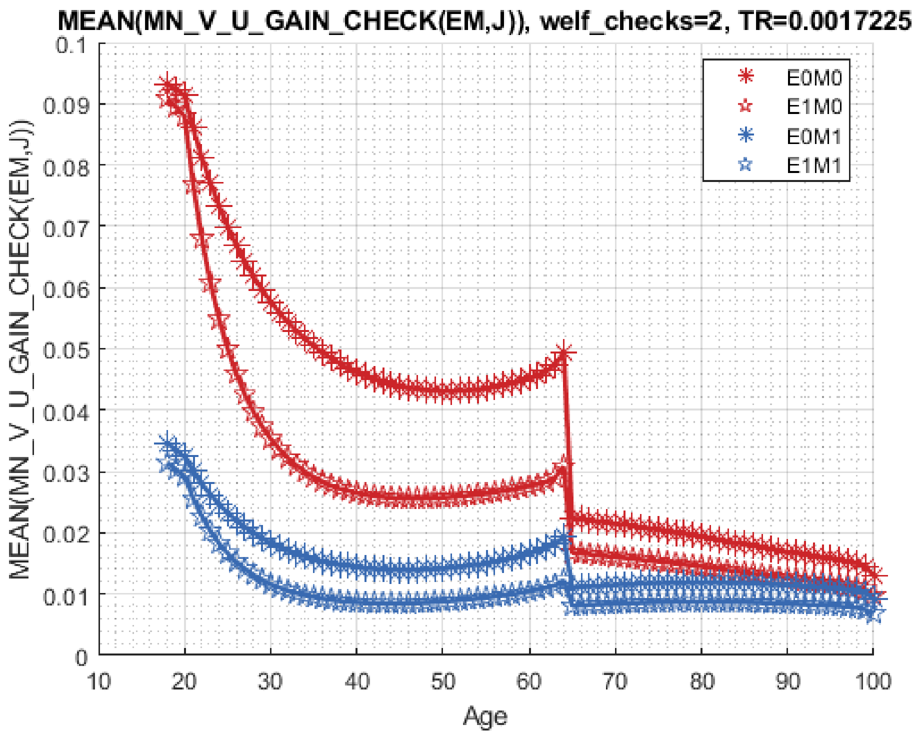
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0         0.0932       0.092347       0.091343       0.086009       0.081278
      2       1       0       0.090875       0.089539       0.087898       0.076892       0.068011
      3       0       1       0.034609       0.033465       0.032367       0.030037       0.027999
      4       1       1       0.031464       0.030328       0.029228       0.025609       0.022664

% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
```

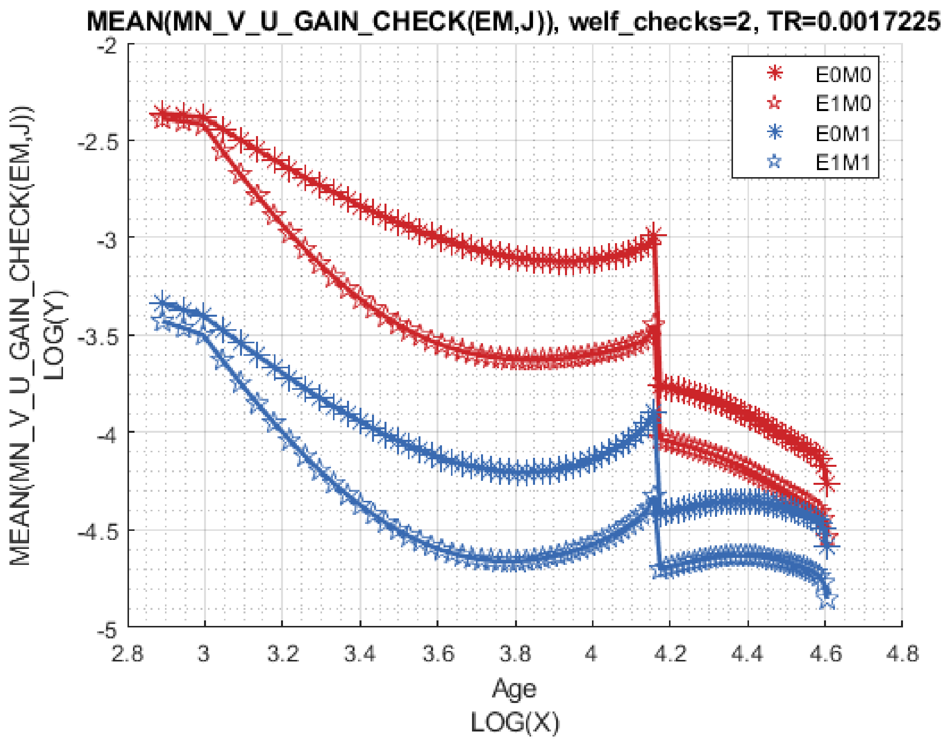
```
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd
```

```
xxx  MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        0.17201        0.17468        0.17758        0.17848        0.17931
      2       1       0        0.18536        0.18966        0.19474        0.19608        0.19716
      3       0       1        0.16432        0.16694        0.16991        0.16969        0.17077
      4       1       1         0.1768        0.17996        0.18466        0.18422         0.1859
```

Graph Mean Values:

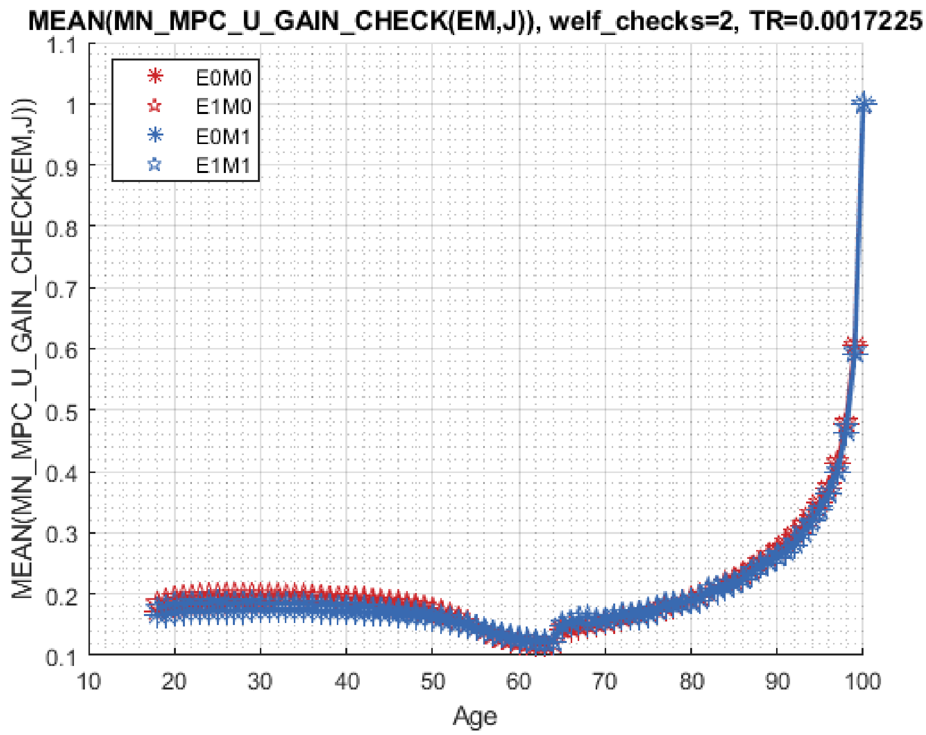
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

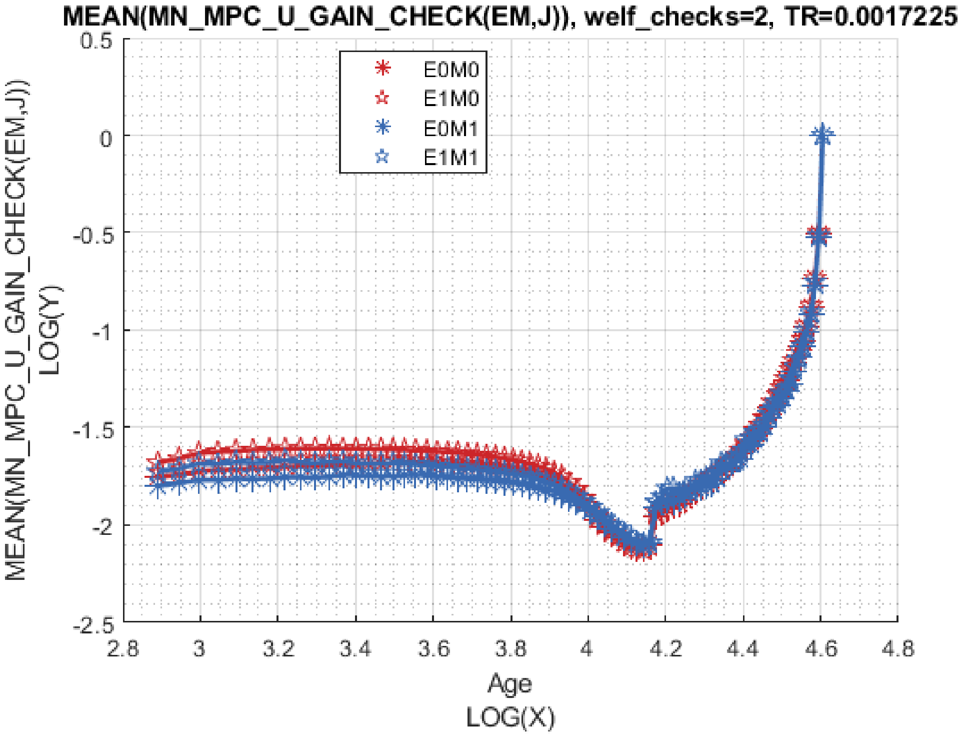




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```




MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225
MEAN(MN_MPC_U_GAIN_CHECK(EM,J))
LOG(Y)
Age
LOG(X)
E0M0
E1M0
E0M1
E1M1
0.5
0
-0.5
-1
-1.5
-2
-2.5
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
4.8

Chapter 9

2019/2020 and 2007/2008 V and C Full State Space

9.1 2020 Full States EV and EC of One Check

This is the example vignette for function: **snw_evuvw20_jaeemk** from the **PrjOptiSNW Package.** 2020 integrated over VU and VW. Average C or V given unemployment probabilities.

9.1.1 Test SNW_EVUVW20_JAEEMK Defaults

Call the function with defaults.

```
clear all;
st_solu_type = 'bisec_vec';

% Solve the VFI Problem and get Value Function
mp_params = snw_mp_param('default_docdense');
mp_params('beta') = 0.95;
mp_controls = snw_mp_control('default_test');

% set Unemployment Related Variables
xi=0.5; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=1
b=0; % Unemployment insurance replacement rate (b=0 refers to no UI benefits; b=1 refers to 100 perc
TR=100/58056; % Value of a welfare check (can receive multiple checks). TO DO: Update with alternati

mp_params('xi') = xi;
mp_params('b') = b;
mp_params('TR') = TR;

% Solve for Unemployment Values
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;
```

Solve the model:

```
%% A. Solve VFI
% 2. Solve VFI and Distributon
```

```
% Solve the Model to get V working and unemployed
% solved with calibrated regular a2
[V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=524.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim      numel      rowN      colN          sum          mean       std
                -    ---    ----    --------     ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07      83     5.265e+05    -6.6619e+08    -15.245    21.86
    ap_VFI      2     2      6      4.37e+07      83     5.265e+05     1.3967e+09     31.962    36.42
    cons_VFI    3     3      6      4.37e+07      83     5.265e+05     2.3276e+08     5.3263    8.441

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    --------

    r1     -293.96    -293.57    -291.09    -285.44    -276.41      -4.3584      -4.2643      -4.171
    r2     -284.42    -284.03    -281.55    -275.97    -267.24      -4.2519      -4.1612      -4.071
    r3     -274.87    -274.48    -272.03    -266.62    -258.33      -4.1429      -4.0559      -3.969
    r4     -265.22    -264.86    -262.58    -257.53    -249.74      -4.0309      -3.9475      -3.864
    r5     -256.51    -256.17    -254.04     -249.3    -241.96      -3.9252      -3.8452      -3.765
    r79    -13.642    -13.628    -13.535    -13.298    -12.896     -0.22092     -0.21058     -0.2008
    r80    -12.283    -12.269    -12.176    -11.939    -11.537     -0.16979     -0.16182      -0.154
    r81    -10.605    -10.591    -10.498    -10.261    -9.8589     -0.11712     -0.11163     -0.1064
    r82    -8.3494    -8.3358    -8.2424    -8.0055    -7.6035    -0.065333    -0.062242     -0.0593
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.020968    -0.019972    -0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2        c3            c4            c5        c526496    c526497    c526498    c5264
           --    --    ----------    ---------    --------    -------    -------    -------    -----

    r1     0     0     0.00051498    0.0066578    0.021589     112.13     117.67      123.4     129.3
    r2     0     0     0.00051498    0.0057684    0.020245     112.17     117.71     123.43     129.3
    r3     0     0     0.00020768    0.0041456    0.018539      112.2     117.73     123.45     129.3
    r4     0     0     0.00010346    0.0041199    0.018307     112.86     118.39     124.11     130.0
    r5     0     0     5.2907e-06    0.0041199    0.018091     113.53     119.07     124.79     130.7
    r79    0     0              0            0           0     81.091     85.364     89.335     93.25
    r80    0     0              0            0           0     76.124     79.747     83.431     86.98
    r81    0     0              0            0           0     67.945     70.639     73.673     76.99
    r82    0     0              0            0           0     50.126     53.467     56.302     57.88
    r83    0     0              0            0           0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5        c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.036717    0.037251    0.040477    0.044486    0.049324     12.265      12.55     12.844
    r2     0.036717    0.037251    0.040477    0.045375    0.050668     12.501     12.787     13.082
    r3     0.036717    0.037251    0.040784    0.046998    0.052374     12.755     13.042     13.337
    r4     0.038144    0.038678    0.042314    0.048449    0.054031         13     13.289     13.584
    r5     0.039534    0.040068    0.043802    0.049839    0.055635     13.236     13.525     13.821
    r79     0.19737     0.19791     0.20163     0.21175     0.23145     35.811     37.362     39.409
    r80     0.19737     0.19791     0.20163     0.21175     0.23145     40.752     42.953     45.286
    r81     0.19737     0.19791     0.20163     0.21175     0.23145     48.909     52.039     55.022
```

```
    r82     0.19737     0.19791     0.20163     0.21175     0.23145     66.71      69.193      72.375
    r83     0.19737     0.19791     0.20163     0.21175     0.23145    116.82     122.65      128.66

% COVID year tax
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
% 2020 V and C same as V_SS and cons_ss if tax the same
if (mp_params('a2_covidyr') == mp_params('a2'))
    % mana from heaven
    V_ss_2020 = V_ss;
    cons_ss_2020 = cons_ss;
else
    % change xi and b to for people without unemployment shock
    % solving for employed but 2020 tax results
    % a2_covidyr > a2, we increased tax in 2020 to pay for covid and other
    % costs resolve for both employed and unemployed
    xi = mp_params('xi');
    b = mp_params('b');
    mp_params('xi') = 1;
    mp_params('b') = 0;
    [V_ss_2020,~,cons_ss_2020,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
    mp_params('xi') = xi;
    mp_params('b') = b;
end

% Solve unemployment, with three input parameters, auto will use a2_covidyr
% as tax, similar for employed call above
[V_unemp_2020,~,cons_unemp_2020] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);

Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN          sum         mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -6.8822e+08    -15.749    22.87
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.3605e+09     31.134    36.29
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05     2.2887e+08     5.2375    8.443

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    --------

    r1     -320.42    -318.92    -310.39    -296.97    -284.58      -4.4406      -4.3429      -4.246
    r2     -310.88    -309.38    -300.85    -287.43    -275.14      -4.3331       -4.239      -4.146
    r3     -301.33    -299.83     -291.3    -277.88    -265.85      -4.2231      -4.1327      -4.043
    r4     -290.68    -289.29    -281.32     -268.6     -257.1      -4.1145      -4.0276      -3.941
    r5     -281.05    -279.76    -272.29     -260.2    -249.16      -4.0121      -3.9284      -3.845
    r79    -13.642    -13.628    -13.535    -13.298    -12.896     -0.22291     -0.21238     -0.2024
    r80    -12.283    -12.269    -12.176    -11.939    -11.537     -0.17128     -0.16316     -0.1555
    r81    -10.605    -10.591    -10.498    -10.261    -9.8589     -0.11815     -0.11254     -0.1072
    r82    -8.3494    -8.3358    -8.2424    -8.0055    -7.6035    -0.065887    -0.062757    -0.05982
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.021146    -0.020134    -0.01918

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2    c3    c4       c5        c526496    c526497    c526498    c526499    c526500
           --    --    --    --    ---------    -------    -------    -------    -------    -------
```

```
    r1     0      0      0      0     0.0083625    107.54      113.09      118.82      124.74      130.86
    r2     0      0      0      0     0.0074731    107.45      112.99      118.72      124.64      130.75
    r3     0      0      0      0     0.0058503    107.33      112.88      118.61      124.52      130.64
    r4     0      0      0      0     0.0049981    107.54      113.08      118.81      124.73      130.85
    r5     0      0      0      0      0.004174    107.76       113.3      119.03      124.95      131.07
    r79    0      0      0      0             0    80.462       84.34      88.311      92.234      96.324
    r80    0      0      0      0             0    75.113      78.736       82.42      85.975      90.439
    r81    0      0      0      0             0    66.945      69.639      72.673      76.669      81.091
    r82    0      0      0      0             0    50.126      53.467      55.311      56.953      60.587
    r83    0      0      0      0             0         0           0           0           0           0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
               c1          c2          c3          c4          c5       c526496    c526497    c526498
            --------    --------    --------    --------    --------    -------    -------    -------

    r1      0.018623    0.019158    0.022901    0.033062    0.044486    11.989     12.265      12.55
    r2      0.018623    0.019158    0.022901    0.033062    0.045375    12.223     12.501     12.787
    r3      0.018623    0.019158    0.022901    0.033062    0.046998    12.476     12.755     13.042
    r4      0.019354    0.019888    0.023632    0.033792    0.048579     12.72         13     13.289
    r5      0.020066    0.020601    0.024344    0.034504    0.050114    12.955     13.236     13.525
    r79      0.19737     0.19791     0.20163     0.21175     0.23145    35.417     37.362     39.409
    r80      0.19737     0.19791     0.20163     0.21175     0.23145    40.752     42.953     45.286
    r81      0.19737     0.19791     0.20163     0.21175     0.23145    48.909     52.039     55.022
    r82      0.19737     0.19791     0.20163     0.21175     0.23145    65.719     68.202     72.375
    r83      0.19737     0.19791     0.20163     0.21175     0.23145    115.84     121.66     127.68
```

```
%% B. Solve Dist
[Phi_true] = snw_ds_main_vec(mp_params, mp_controls, ap_ss, cons_ss);
```

```
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=929.8427
```

Previous code

```
% % Solve the Model to get V working and unemployed
% [V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
% % Solve unemployment
% [V_unemp,~,cons_unemp,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
% [Phi_true] = snw_ds_main(mp_params, mp_controls, ap_ss, cons_ss, mp_valpol_more_ss);
```

9.1.2 Precompute

```
inc_VFI = mp_valpol_more_ss('inc_VFI');
spouse_inc_VFI = mp_valpol_more_ss('spouse_inc_VFI');
total_inc_VFI = inc_VFI + spouse_inc_VFI;
% Get Matrixes
cl_st_precompute_list = {'a', ...
    'inc', 'inc_unemp', 'spouse_inc', 'spouse_inc_unemp', 'ref_earn_wageind_grid'};
mp_controls('bl_print_precompute_verbose') = false;
[mp_precompute_res] = snw_hh_precompute(mp_params, mp_controls, cl_st_precompute_list, ap_ss, Phi_tr
```

```
Wage quintile cutoffs=0.4645     0.71528      1.0335      1.5632
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time cost=274.
```

9.1.3 Solve for 2020 Evuvw With 0 and 2 Checks

```
% Call Function
welf_checks = 0;
[ev20_jaeemk_check0, ec20_jaeemk_check0] = snw_evuvw20_jaeemk(...
```

```
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss_2020, cons_ss_2020, V_unemp_2020, cons_unemp_2020, mp_precompute_res);
```

```
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=st_biden_or_trump_undefined;welf_checks=0;TR=0.001722
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=0;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_docde
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=8.01
```

```
% Call Function
welf_checks = 2;
[ev20_jaeemk_check2, ec20_jaeemk_check2] = snw_evuvw20_jaeemk(...
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss_2020, cons_ss_2020, V_unemp_2020, cons_unemp_2020, mp_precompute_res);
```

```
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=st_biden_or_trump_undefined;welf_checks=2;TR=0.001722
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=2;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_docde
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=7.97
```

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check = ev20_jaeemk_check2 - ev20_jaeemk_check0;
mn_MPC_U_gain_share_check = (ec20_jaeemk_check2 - ec20_jaeemk_check0)./(welf_checks*mp_params('TR'))
```

9.1.4 Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

9.1.5 Analyze Difference in V and C with Check

The difference between V and V with Check, marginal utility gain given the check.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

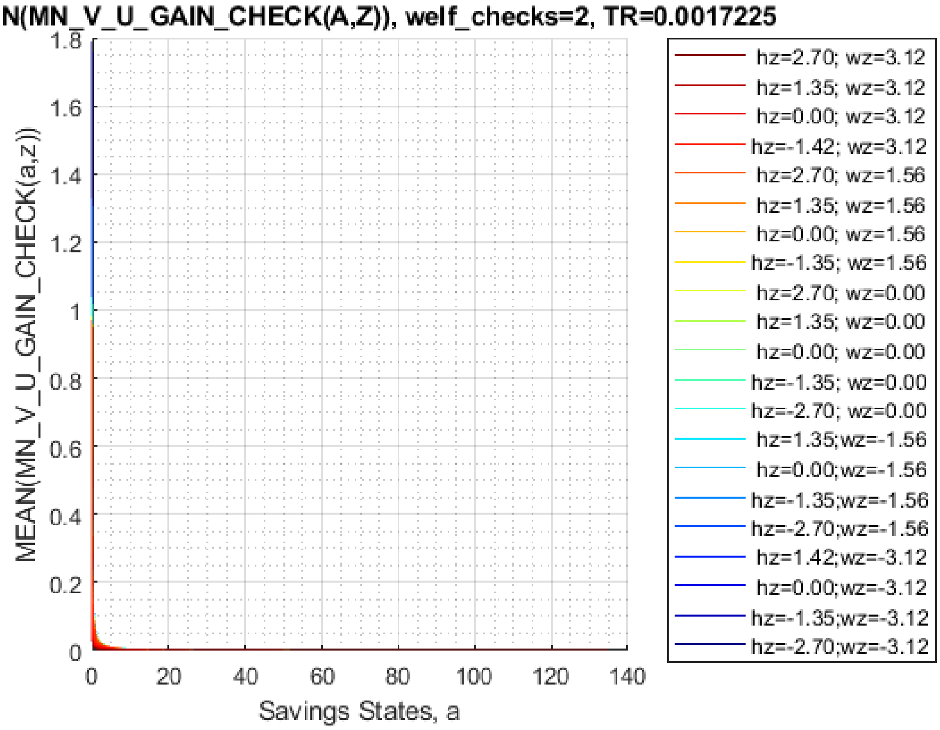
MEAN(MN_V_GAIN_CHECK(A,Z))

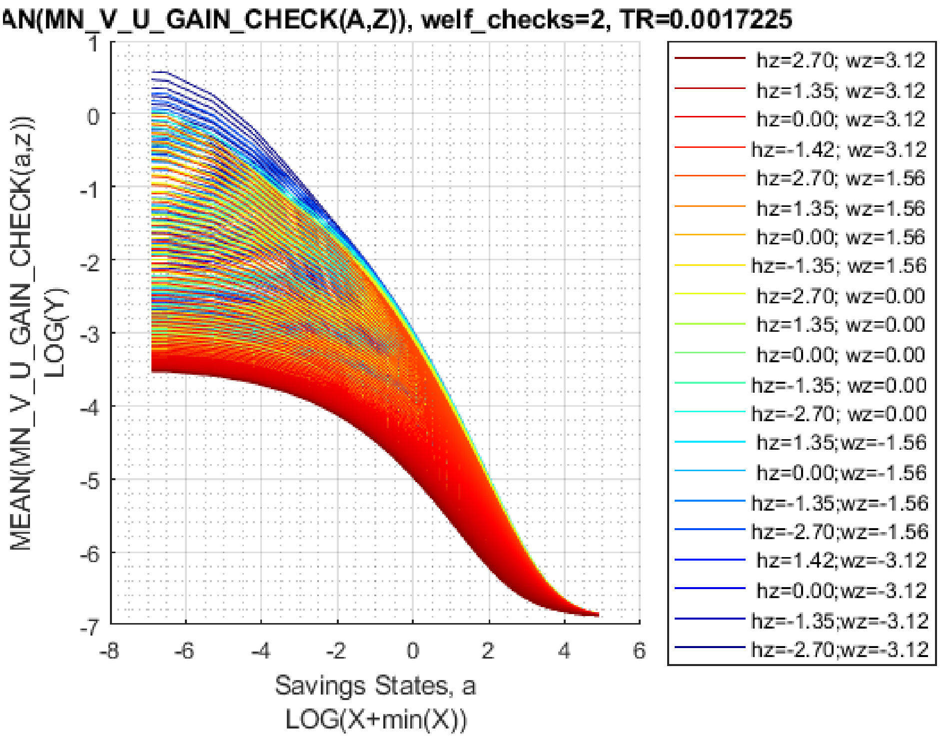
Tabulate value and policies along savings and shocks:

```
% Set
```

```
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_par
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1             0        1.7895        1.5987        1.4282        1.2759        1.1399
```

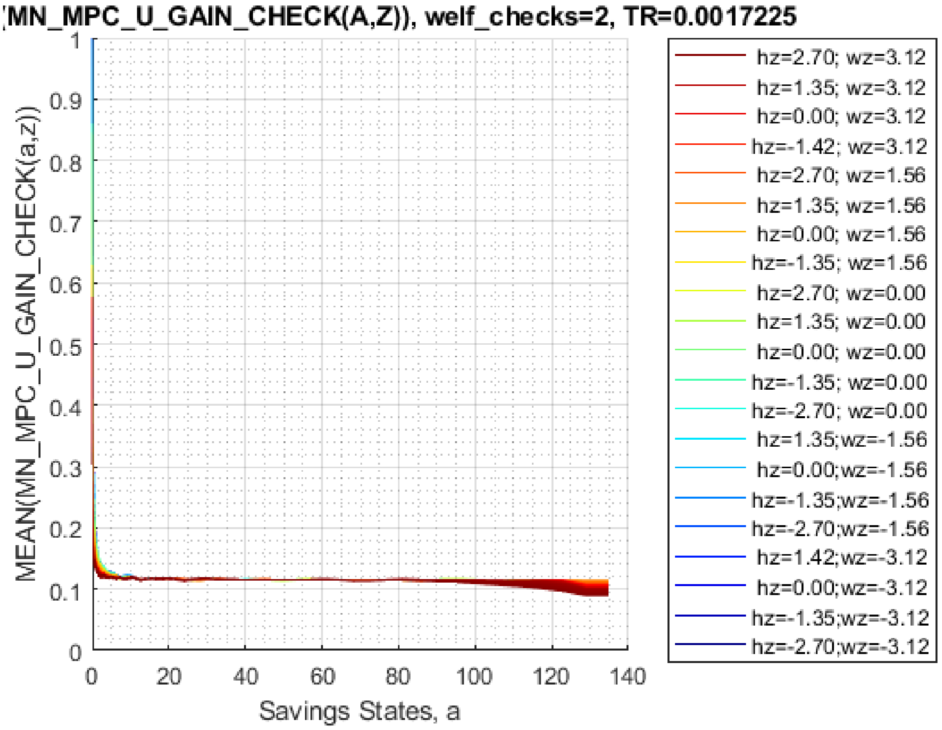
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

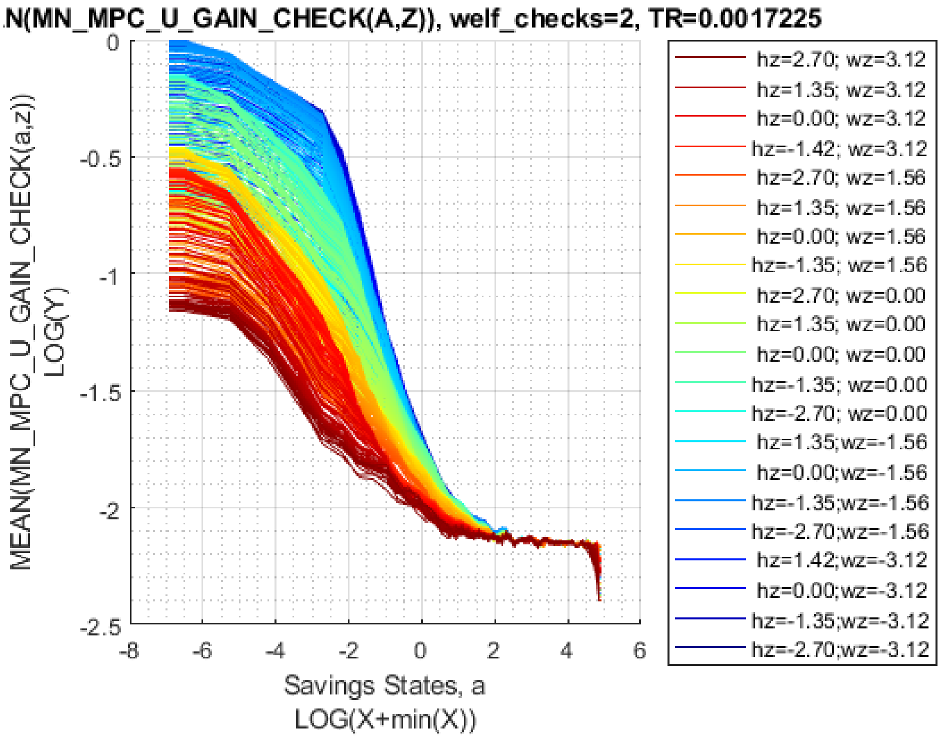




Graph Mean Consumption (MPC: Share of Check Consumed):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





9.1.6 Analyze Marginal Value and MPC over Y(a,eta), Conditional On Kids, Marry, Age, Education

Income is generated by savings and shocks, what are the income levels generated by all the shock and savings points conditional on kids, marital status, age and educational levels. Plot on the Y axis MPC, and plot on the X axis income levels, use colors to first distinguish between different a levels, then use colors to distinguish between different eta levles.

Set Up date, Select Age 38, unmarried, no kids, lower education:

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
% 38 year old, unmarried, no kids, lower educated
% Only Household Head Shock Matters so select up to 'n_eta_H_grid'
mn_total_inc_jemk = total_inc_VFI(20,:,1:mp_params('n_eta_H_grid'),1,1,1);
mn_V_W_gain_check_use = ev20_jaeemk_check2 - ev20_jaeemk_check0;
mn_C_W_gain_check_use = ec20_jaeemk_check2 - ec20_jaeemk_check0;
```

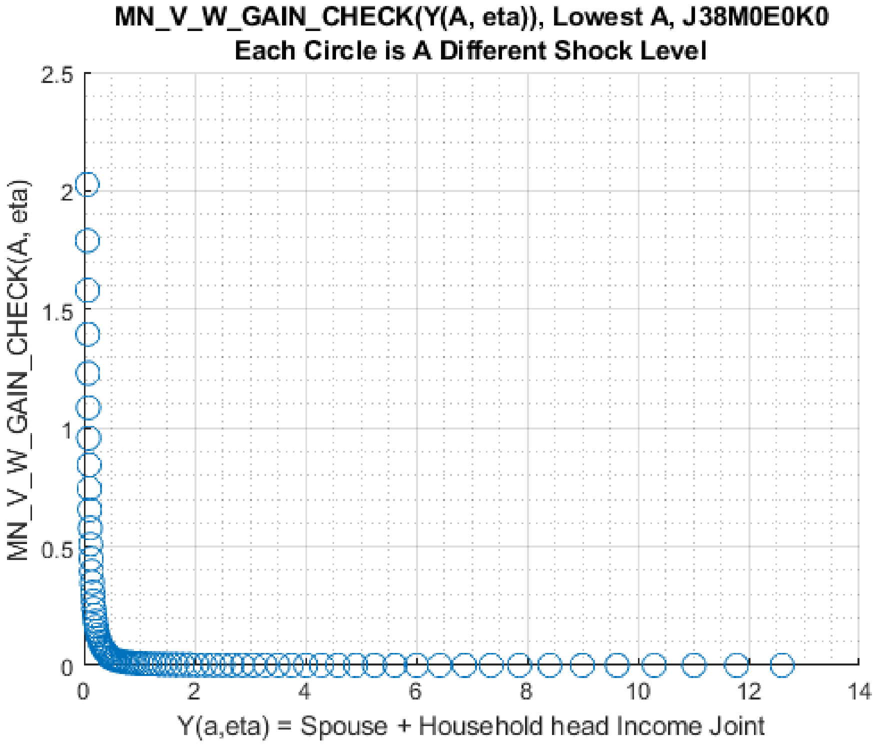
Select Age, Education, Marital, Kids Count:s

```
% Selections
it_age = 21; % +18
it_marital = 1; % 1 = unmarried
it_kids = 1; % 1 = kids is zero
it_educ = 1; % 1 = lower education
% Select: NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
mn_C_W_gain_check_jemk = mn_C_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ, it_m
mn_V_W_gain_check_jemk = mn_V_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ, it_m
% Reshape, so shock is the first dim, a is the second
mt_total_inc_jemk = permute(mn_total_inc_jemk,[3,2,1]);
mt_C_W_gain_check_jemk = permute(mn_C_W_gain_check_jemk,[3,2,1]);
mt_C_W_gain_check_jemk(mt_C_W_gain_check_jemk<=1e-10) = 1e-10;
mt_V_W_gain_check_jemk = permute(mn_V_W_gain_check_jemk,[3,2,1]);
mt_V_W_gain_check_jemk(mt_V_W_gain_check_jemk<=1e-10) = 1e-10;
% Generate meshed a and shock grid
[mt_eta_H, mt_a] = ndgrid(eta_H_grid(1:mp_params('n_eta_H_grid')), agrid);
```

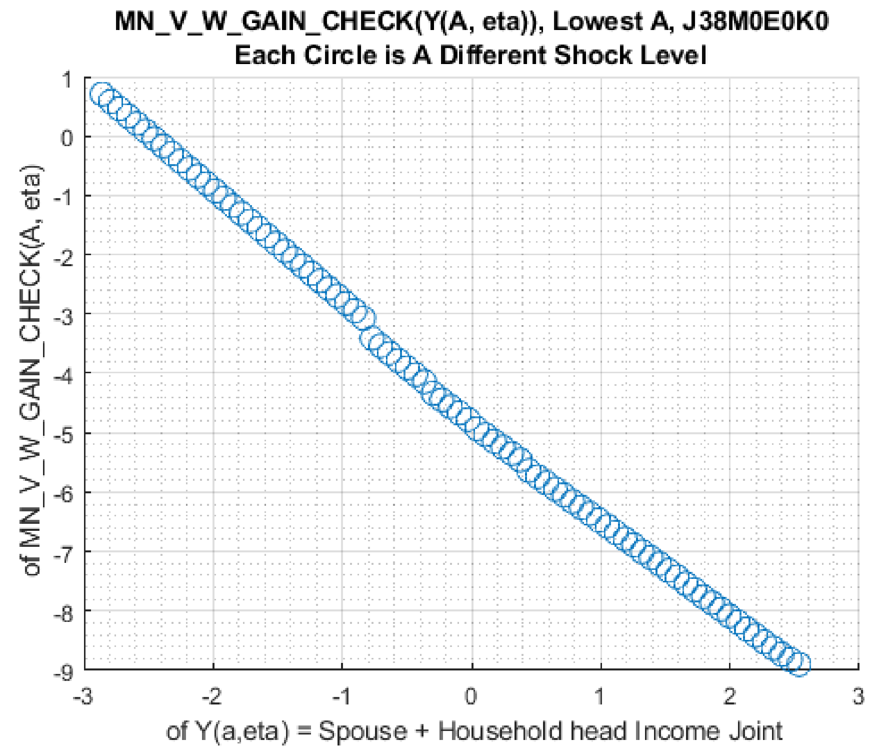
9.1.7 Marginal Value Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
it_a = 1;
scatter((mt_total_inc_jemk(:,it_a)), (mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

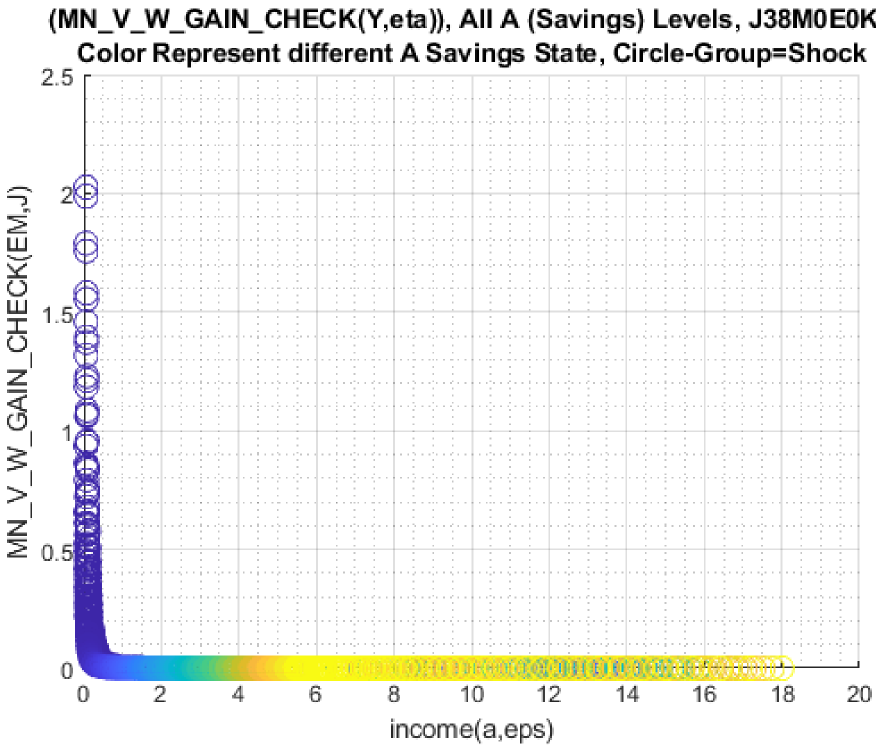



```
figure();
it_shock = 1;
scatter(log(mt_total_inc_jemk(:,it_a)), log(mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel(' of Y(a,eta) = Spouse + Household head Income Joint');
ylabel(' of MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

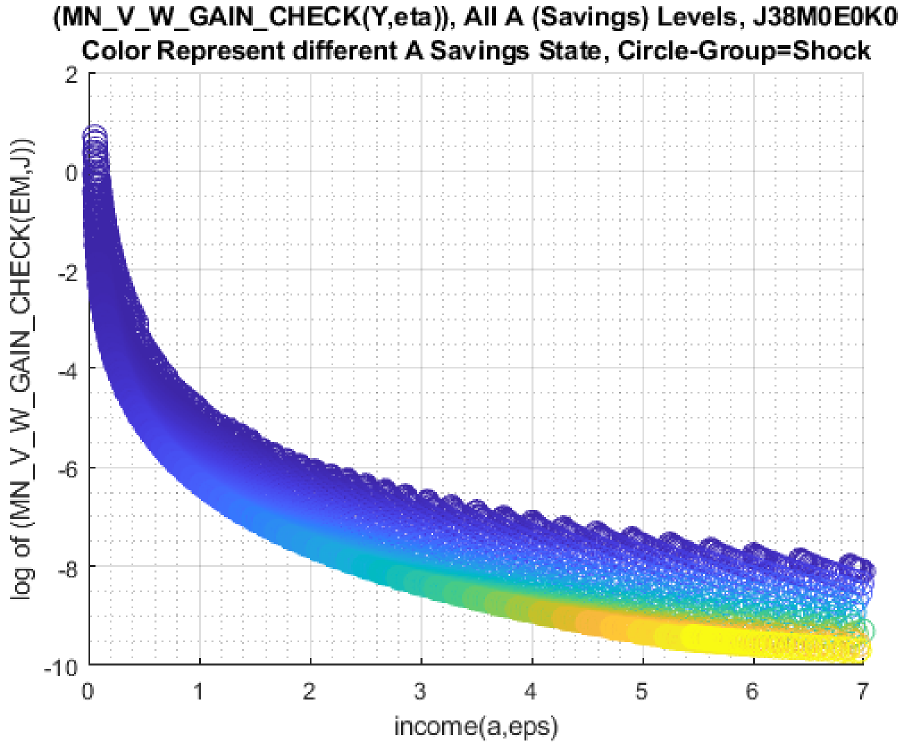


Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_V\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```



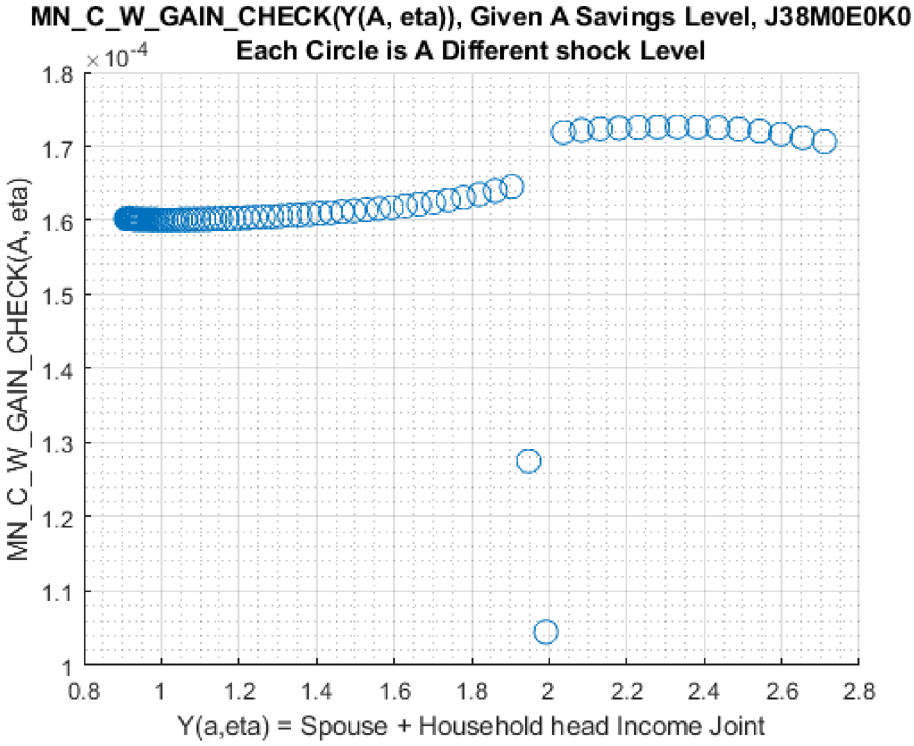

```
figure();
scatter((mt_total_inc_jemk(:)), log(mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
xlim([0,7]);
grid on;
grid minor;
```



9.1.8 Marginal Consumption Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

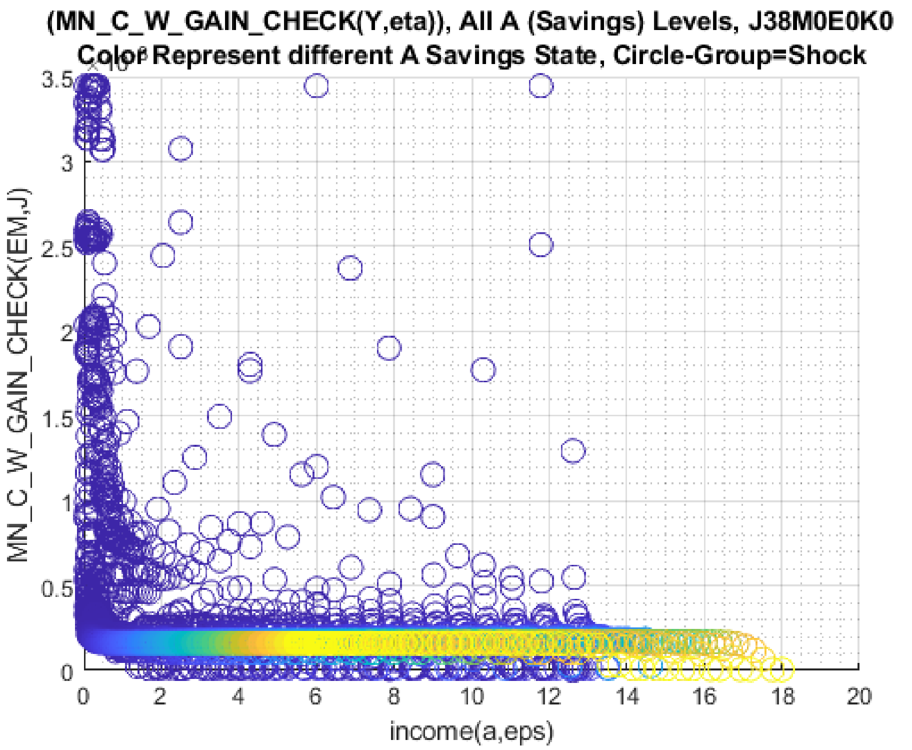
How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
it_a = 50;
scatter(log(mt_total_inc_jemk(:,it_a)), mt_C_W_gain_check_jemk(:,it_a), 100);
title({'MN\_C\_W\_GAIN\_CHECK(Y(A, eta)), Given A Savings Level, J38M0E0K0', ...
    'Each Circle is A Different shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_C\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

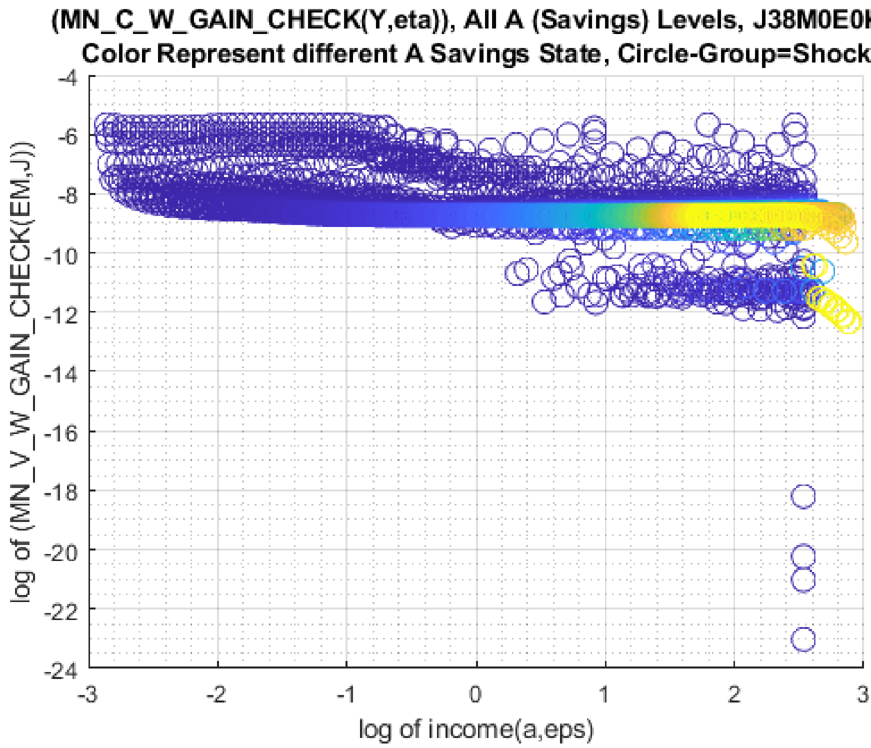


Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_C_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_C\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```



```
figure();
scatter(log(mt_total_inc_jemk(:)), log(mt_C_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('log of income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
grid on;
grid minor;
```



9.1.9 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

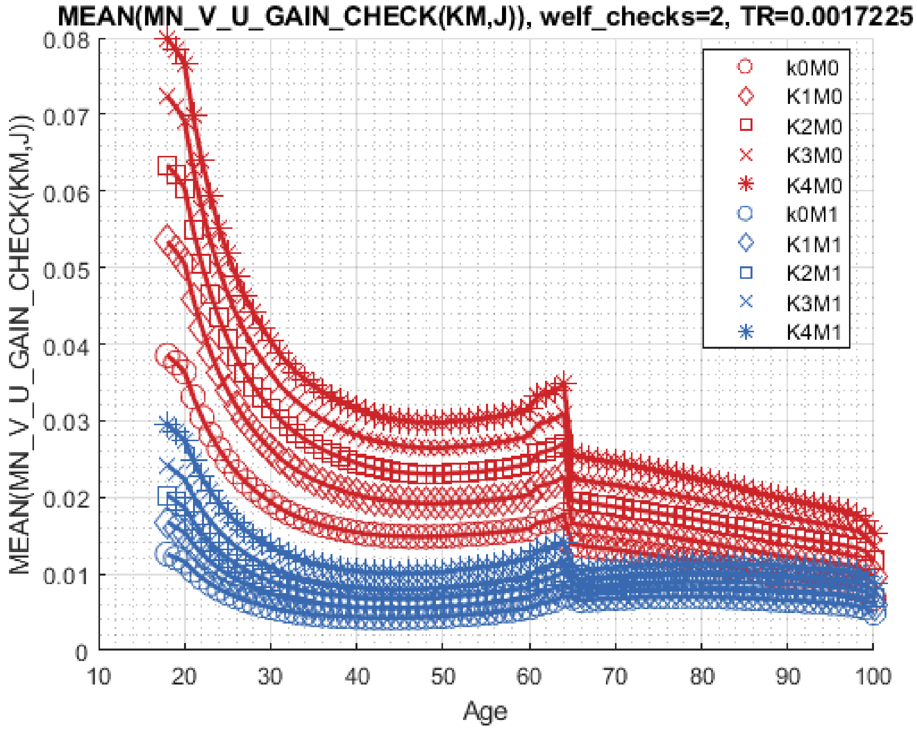
Tabulate value and policies:

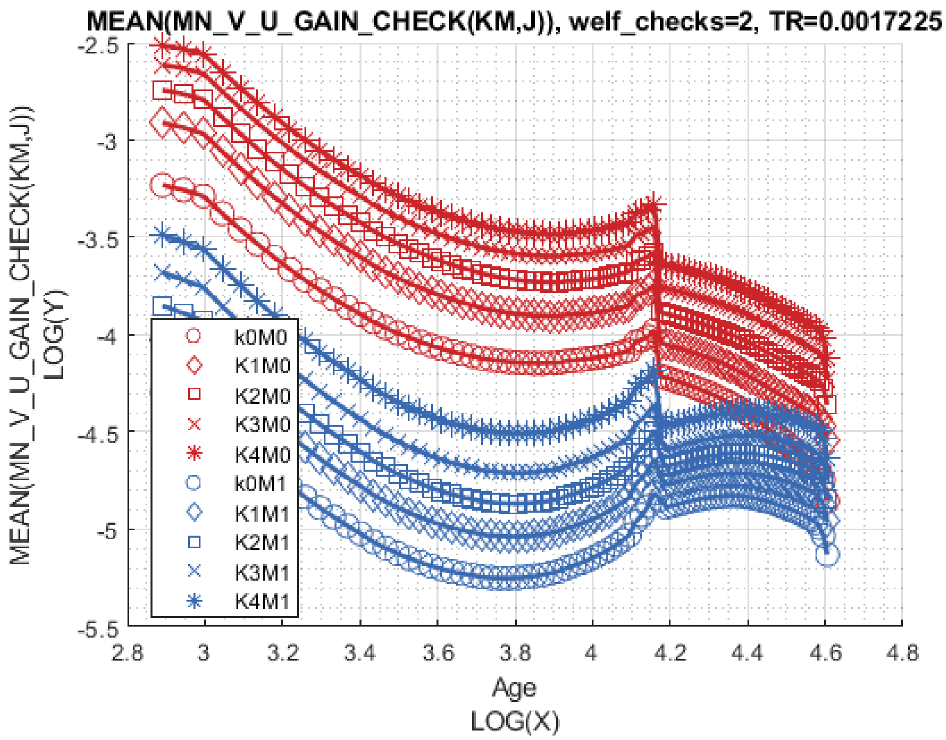
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
```

```
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        0.038527       0.037553       0.036379       0.033145       0.030452
      2        2       0        0.053415       0.052114       0.050493       0.045939       0.042139
      3        3       0        0.063389       0.062083       0.060366       0.054947       0.050428
      4        4       0        0.072383       0.070992       0.069108       0.062921       0.057762
      5        5       0        0.079913       0.078518       0.076562       0.069748        0.06407
      6        1       1        0.012602       0.012065        0.01155       0.010426      0.0094863
      7        2       1         0.01678       0.016072       0.015393       0.013895       0.012637
      8        3       1        0.020271       0.019456       0.018665       0.016854       0.015337
      9        4       1        0.024225       0.023287       0.022361       0.020206       0.018399
     10        5       1        0.029524       0.028487        0.02744        0.02482       0.022631

% Consumption Function
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd

xxx  MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        0.084608       0.090543        0.10335        0.10135       0.099555
      2        2       0         0.09227        0.09874         0.1136        0.11143        0.10989
      3        3       0         0.10204         0.1099        0.12674        0.12369        0.12091
      4        4       0         0.10652         0.1144        0.13184        0.12908         0.1263
      5        5       0          0.1125        0.11953        0.13744        0.13456        0.13155
      6        1       1         0.11122        0.11518        0.12131        0.11968        0.11893
      7        2       1         0.11206        0.11641        0.12306        0.12166        0.12056
      8        3       1          0.1176        0.12247         0.1311        0.12797        0.12718
      9        4       1         0.11929        0.12501        0.13176         0.1305        0.13114
     10        5       1          0.1264        0.13179         0.1402        0.13884        0.13503
```

Graph Mean Values:

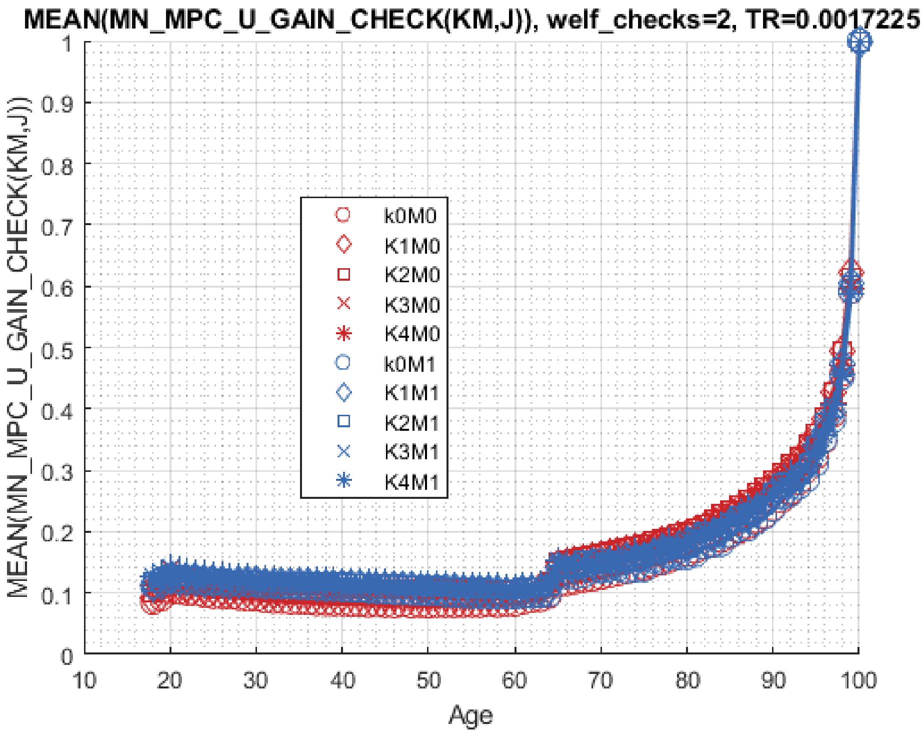
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

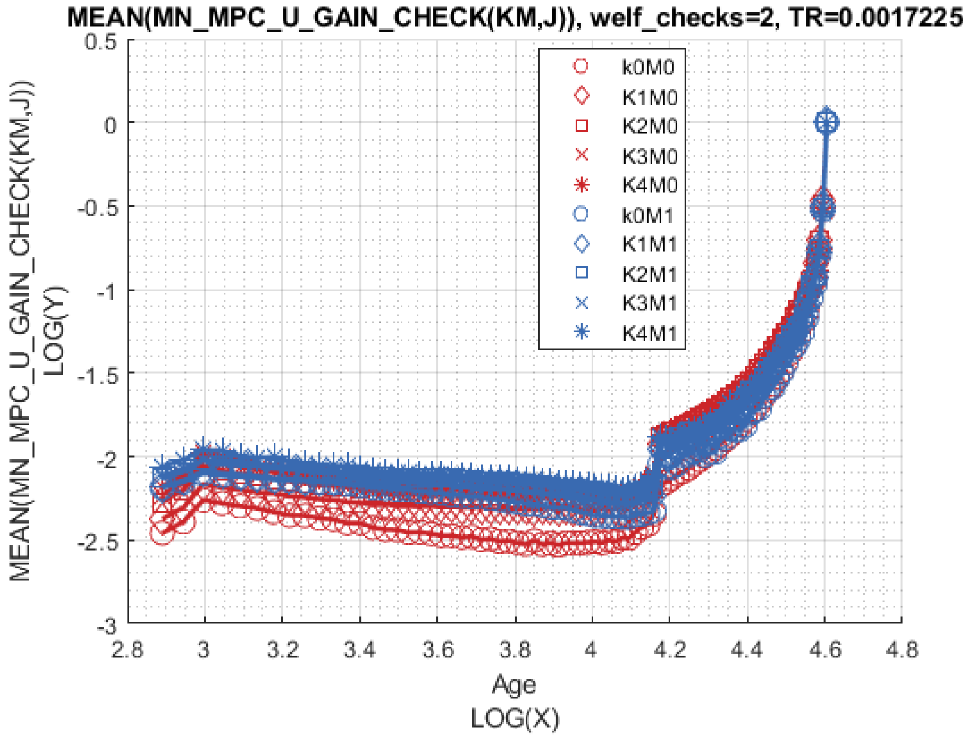




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





9.1.10 Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

```
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

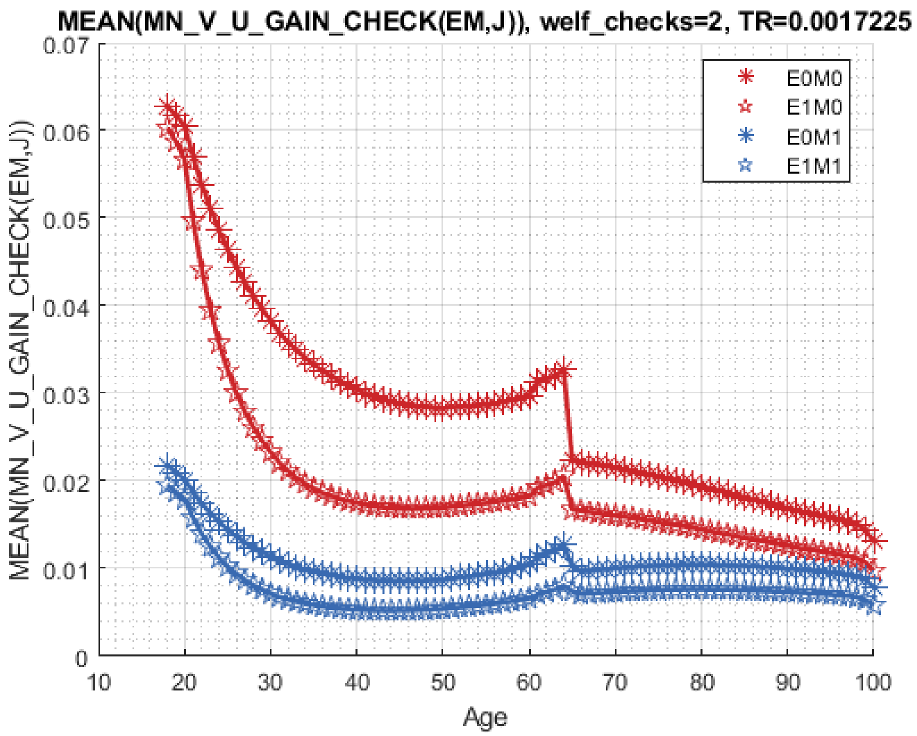
MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

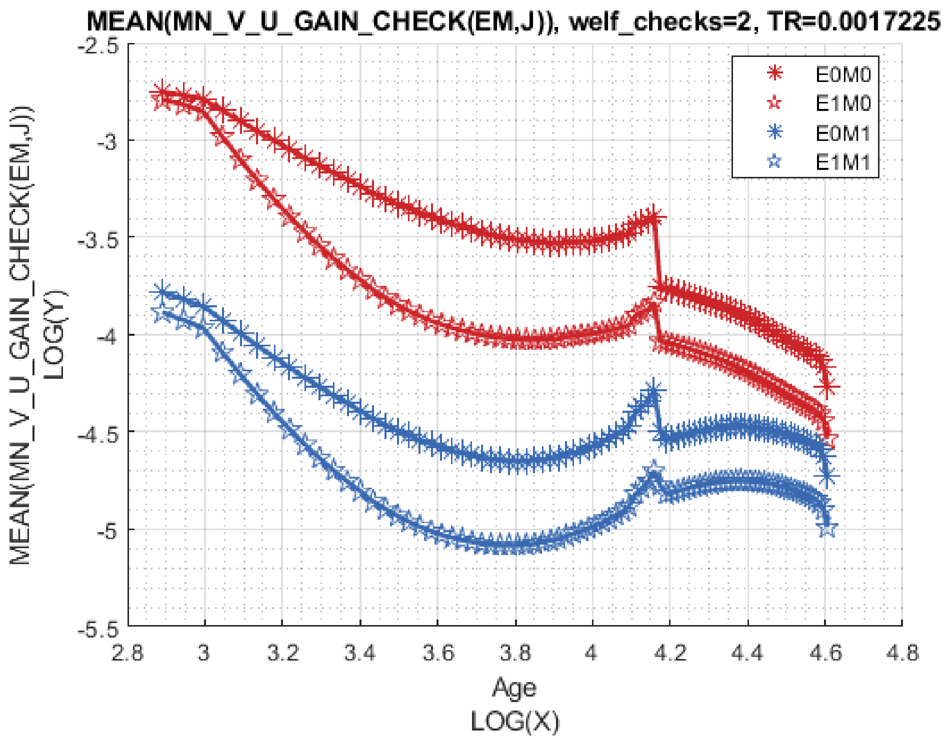
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0       0.062745        0.06175       0.060482       0.056976       0.053863
      2       1       0       0.060305       0.058754       0.056681       0.049704       0.044078
      3       0       1       0.021795       0.020987       0.020201       0.018731       0.017442
      4       1       1       0.019567        0.01876       0.017963       0.015749       0.013955

% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd

xxx  MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0       0.091431        0.09559        0.10516        0.10437        0.10421
      2       1       0        0.10775        0.11765        0.14003        0.13567        0.13107
      3       0       1         0.1091        0.11287         0.1172        0.11714        0.11697
      4       1       1        0.12553        0.13148        0.14177        0.13832        0.13616
```

Graph Mean Values:

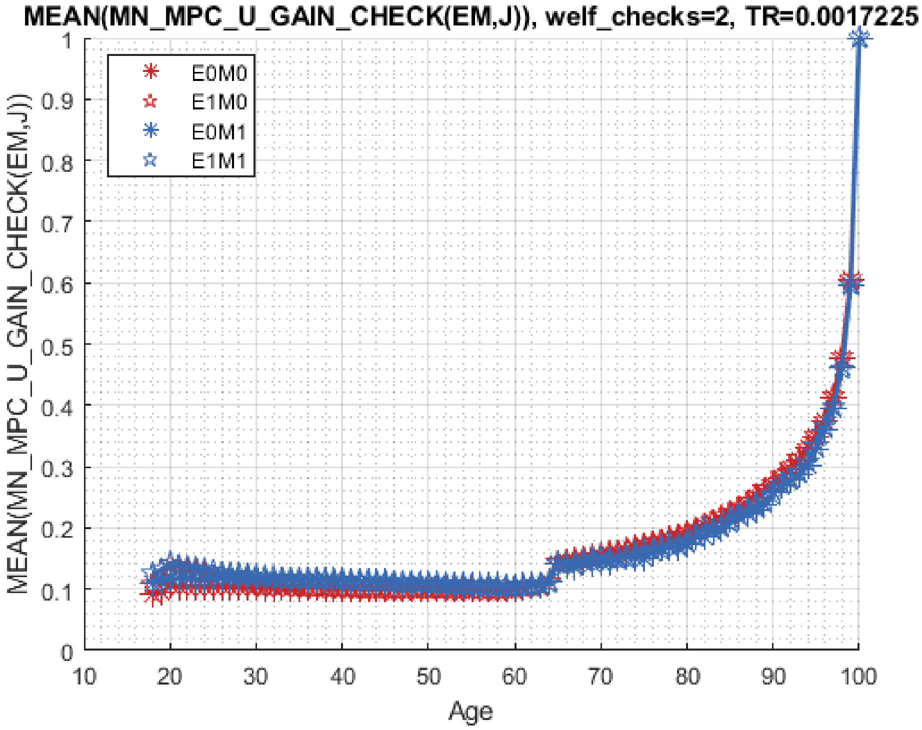
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

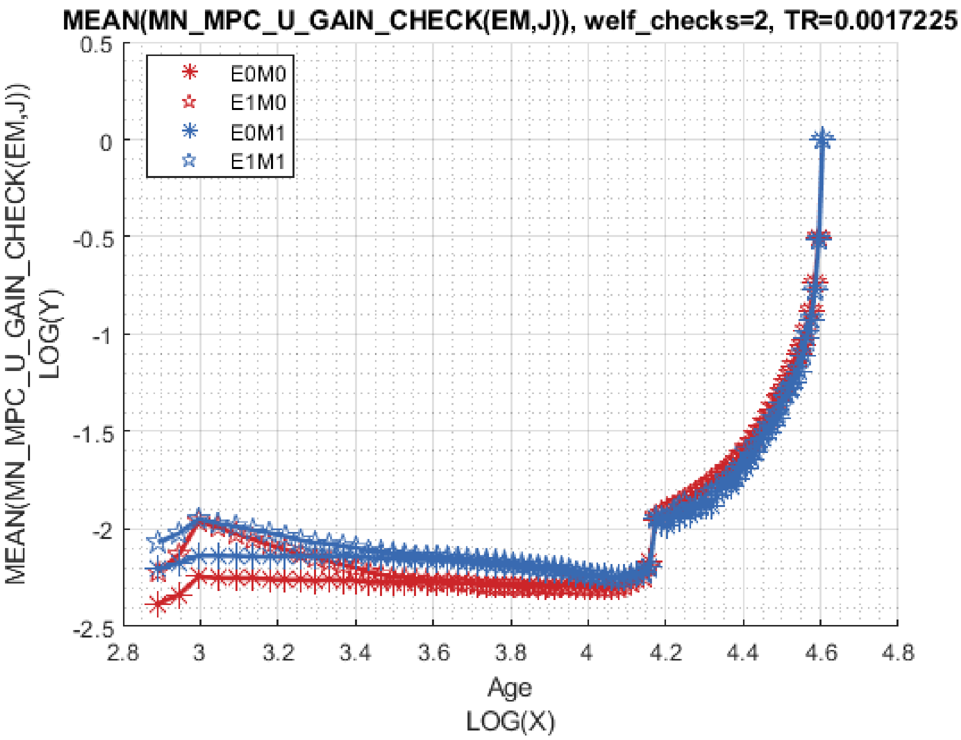




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





9.2 2019 (Biden/Trump Checks) Full States EV and EC of Two Checks

This is the example vignette for function: **snw_evuvw19_jaeemk_foc** from the **PrjOptiSNW Package.** 2019 integrated over VU and VW, given optimal savings choices, unemployment shocks and various expectations.

Given 2020 JAEEMK (age, endogenous savings, education, income shock, marital status, kids count), what is the expected value for the planner given 2020 JAEEMK and transition between 2019 to 2020

JAEEMK given some stimulus check assignment based on 2019 information? (Stimulus amount set by WELF_CHECKS). This is similar to **snw_evuvw19_jaeemk**, except the solution here, under **snw_evuvw19_jaeemk_foc**, relies on First Order Conditions, and are hence faster.

Despite the name, this function supports solving the 2019 looking into 2020 as well as the 2007 looking into 2008 problems. The idea is that the planner only has information from 2019 and from 2007, and must allocate using those information. Stimulus, however, is given in 2020 and in 2008. So the planner needs to consider expected values in consumption or welfare given the transition probabilities of states in 2007 to 2008 and in 2019 to 2020. The **snw_evuvw19_jmky** file then aggregates the full state-space results to just JMKY state-space, which is the extend of information available to the planner.

9.2.1 Test SNW_EVUVW19_JAEEMK Defaults for 2019

Call the function with defaults parameters.

```
clear all;
% Solution types
st_biden_or_trump = 'bidenchk';
st_solu_type = 'bisec_vec';

% Solve the VFI Problem and get Value Function
mp_params = snw_mp_param('default_docdense');
% mp_params = snw_mp_param('default_dense');
mp_params('beta') = 0.95;
mp_params('st_biden_or_trump') = st_biden_or_trump;
% mp_params = snw_mp_param('default_dense');
mp_controls = snw_mp_control('default_test');

% set Unemployment Related Variables
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
% mp_params('a2_covidyr') = mp_params('a2_covidyr_tax_fully_pay');

% Solve for Unemployment Values
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = true;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;

% Solve the Model to get V working and unemployed
[V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=519.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN         sum          mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -6.6619e+08    -15.245    21.86
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.3967e+09     31.962    36.42
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05     2.3276e+08     5.3263    8.441
```

```
xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
           c1         c2         c3         c4         c5        c526496      c526497      c526498
         -------    -------    -------    -------    -------    ---------    ---------    ---------

    r1   -293.96    -293.57    -291.09    -285.44    -276.41      -4.3584      -4.2643      -4.171
    r2   -284.42    -284.03    -281.55    -275.97    -267.24      -4.2519      -4.1612      -4.071
    r3   -274.87    -274.48    -272.03    -266.62    -258.33      -4.1429      -4.0559      -3.969
    r4   -265.22    -264.86    -262.58    -257.53    -249.74      -4.0309      -3.9475      -3.864
    r5   -256.51    -256.17    -254.04     -249.3    -241.96      -3.9252      -3.8452      -3.765
    r79  -13.642    -13.628    -13.535    -13.298    -12.896     -0.22092     -0.21058     -0.2008
    r80  -12.283    -12.269    -12.176    -11.939    -11.537     -0.16979     -0.16182      -0.154
    r81  -10.605    -10.591    -10.498    -10.261    -9.8589     -0.11712     -0.11163     -0.1064
    r82  -8.3494    -8.3358    -8.2424    -8.0055    -7.6035    -0.065333    -0.062242     -0.0593
    r83  -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.020968    -0.019972    -0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
         c1    c2        c3            c4           c5       c526496    c526497    c526498    c5264
         --    --    ----------    ---------    --------    -------    -------    -------    -----

    r1   0     0     0.00051498    0.0066578    0.021589    112.13     117.67      123.4     129.3
    r2   0     0     0.00051498    0.0057684    0.020245    112.17     117.71     123.43     129.3
    r3   0     0     0.00020768    0.0041456    0.018539     112.2     117.73     123.45     129.3
    r4   0     0     0.00010346    0.0041199    0.018307    112.86     118.39     124.11     130.0
    r5   0     0     5.2907e-06    0.0041199    0.018091    113.53     119.07     124.79     130.7
    r79  0     0              0            0           0    81.091     85.364     89.335     93.25
    r80  0     0              0            0           0    76.124     79.747     83.431     86.98
    r81  0     0              0            0           0    67.945     70.639     73.673     76.99
    r82  0     0              0            0           0    50.126     53.467     56.302     57.88
    r83  0     0              0            0           0         0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
            c1          c2          c3          c4          c5        c526496    c526497    c526498
         --------    --------    --------    --------    --------    -------    -------    -------

    r1   0.036717    0.037251    0.040477    0.044486    0.049324    12.265      12.55     12.844
    r2   0.036717    0.037251    0.040477    0.045375    0.050668    12.501     12.787     13.082
    r3   0.036717    0.037251    0.040784    0.046998    0.052374    12.755     13.042     13.337
    r4   0.038144    0.038678    0.042314    0.048449    0.054031        13     13.289     13.584
    r5   0.039534    0.040068    0.043802    0.049839    0.055635    13.236     13.525     13.821
    r79   0.19737     0.19791     0.20163     0.21175     0.23145    35.811     37.362     39.409
    r80   0.19737     0.19791     0.20163     0.21175     0.23145    40.752     42.953     45.286
    r81   0.19737     0.19791     0.20163     0.21175     0.23145    48.909     52.039     55.022
    r82   0.19737     0.19791     0.20163     0.21175     0.23145     66.71     69.193     72.375
    r83   0.19737     0.19791     0.20163     0.21175     0.23145    116.82     122.65     128.66
```

```
inc_VFI = mp_valpol_more_ss('inc_VFI');
spouse_inc_VFI = mp_valpol_more_ss('spouse_inc_VFI');
total_inc_VFI = inc_VFI + spouse_inc_VFI;
% Solve employment, same as 2020, except with possible change in tax
mp_params('xi') = 1;
mp_params('b') = 0;
[V_emp_2020,~,cons_emp_2020,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std
	-	---	----	--------	----	---------	-----------	-------	-----
V_VFI	1	1	6	4.37e+07	83	5.265e+05	-6.6619e+08	-15.245	21.86
ap_VFI	2	2	6	4.37e+07	83	5.265e+05	1.3967e+09	31.962	36.42
cons_VFI	3	3	6	4.37e+07	83	5.265e+05	2.3276e+08	5.3263	8.441

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
	-------	-------	-------	-------	-------	---------	---------	--------
r1	-293.96	-293.57	-291.09	-285.44	-276.41	-4.3584	-4.2643	-4.171
r2	-284.42	-284.03	-281.55	-275.97	-267.24	-4.2519	-4.1612	-4.071
r3	-274.87	-274.48	-272.03	-266.62	-258.33	-4.1429	-4.0559	-3.969
r4	-265.22	-264.86	-262.58	-257.53	-249.74	-4.0309	-3.9475	-3.864
r5	-256.51	-256.17	-254.04	-249.3	-241.96	-3.9252	-3.8452	-3.765
r79	-13.642	-13.628	-13.535	-13.298	-12.896	-0.22092	-0.21058	-0.2008
r80	-12.283	-12.269	-12.176	-11.939	-11.537	-0.16979	-0.16182	-0.154
r81	-10.605	-10.591	-10.498	-10.261	-9.8589	-0.11712	-0.11163	-0.1064
r82	-8.3494	-8.3358	-8.2424	-8.0055	-7.6035	-0.065333	-0.062242	-0.0593
r83	-5.0665	-5.0529	-4.9595	-4.7226	-4.3206	-0.020968	-0.019972	-0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c5264
	--	--	----------	---------	--------	-------	-------	-------	-----
r1	0	0	0.00051498	0.0066578	0.021589	112.13	117.67	123.4	129.3
r2	0	0	0.00051498	0.0057684	0.020245	112.17	117.71	123.43	129.3
r3	0	0	0.00020768	0.0041456	0.018539	112.2	117.73	123.45	129.3
r4	0	0	0.00010346	0.0041199	0.018307	112.86	118.39	124.11	130.0
r5	0	0	5.2907e-06	0.0041199	0.018091	113.53	119.07	124.79	130.7
r79	0	0	0	0	0	81.091	85.364	89.335	93.25
r80	0	0	0	0	0	76.124	79.747	83.431	86.98
r81	0	0	0	0	0	67.945	70.639	73.673	76.99
r82	0	0	0	0	0	50.126	53.467	56.302	57.88
r83	0	0	0	0	0	0	0	0	

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
	--------	--------	--------	--------	--------	-------	-------	-------
r1	0.036717	0.037251	0.040477	0.044486	0.049324	12.265	12.55	12.844
r2	0.036717	0.037251	0.040477	0.045375	0.050668	12.501	12.787	13.082
r3	0.036717	0.037251	0.040784	0.046998	0.052374	12.755	13.042	13.337
r4	0.038144	0.038678	0.042314	0.048449	0.054031	13	13.289	13.584
r5	0.039534	0.040068	0.043802	0.049839	0.055635	13.236	13.525	13.821
r79	0.19737	0.19791	0.20163	0.21175	0.23145	35.811	37.362	39.409
r80	0.19737	0.19791	0.20163	0.21175	0.23145	40.752	42.953	45.286
r81	0.19737	0.19791	0.20163	0.21175	0.23145	48.909	52.039	55.022
r82	0.19737	0.19791	0.20163	0.21175	0.23145	66.71	69.193	72.375
r83	0.19737	0.19791	0.20163	0.21175	0.23145	116.82	122.65	128.66

```
% Solve unemployment, different income than under ss due to income losses
mp_params('xi') = 0.50;
mp_params('b') = 0.50;
[V_unemp_2020,~,cons_unemp_2020,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
```

```
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN         sum         mean      std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -6.7567e+08    -15.462    22.25
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.3783e+09     31.541     36.3
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05     2.3114e+08     5.2893    8.440

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    ---------

    r1      -302.8    -302.11    -297.97     -290.4    -280.12      -4.3991      -4.3032      -4.208
    r2     -293.25    -292.57    -288.43    -280.86     -270.8      -4.2921      -4.1998      -4.108
    r3      -283.7    -283.02    -278.88    -271.34    -261.75      -4.1826       -4.094      -4.006
    r4     -273.72    -273.09    -269.23    -262.13     -253.1      -4.0721       -3.987      -3.902
    r5      -264.7    -264.11    -260.51    -253.79    -245.27      -3.9679      -3.8861      -3.805
    r79    -13.642    -13.628    -13.535    -13.298    -12.896     -0.22191     -0.21148     -0.2016
    r80    -12.283    -12.269    -12.176    -11.939    -11.537     -0.17053     -0.16249      -0.154
    r81    -10.605    -10.591    -10.498    -10.261    -9.8589     -0.11764     -0.11208     -0.1068
    r82    -8.3494    -8.3358    -8.2424    -8.0055    -7.6035    -0.065608    -0.062497    -0.05959
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.021056    -0.020052    -0.01911

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2    c3        c4            c5       c526496    c526497    c526498    c526499    c
           --    --    --    ----------    --------    -------    -------    -------    -------    -

    r1     0     0     0      0.0011815    0.013905     109.98     115.52     121.26     127.18     1
    r2     0     0     0     0.00090277    0.013905     109.95     115.49     121.22     127.14     1
    r3     0     0     0     0.00051498    0.013905      109.9     115.45     121.18      127.1     1
    r4     0     0     0     0.00051498    0.013905     110.34     115.88     121.61     127.53     1
    r5     0     0     0     0.00048777    0.013905     110.79     116.33     122.06     127.98
    r79    0     0     0              0           0     80.974     84.852     88.823     92.746     9
    r80    0     0     0              0           0     75.619     79.241     82.926     86.481     9
    r81    0     0     0              0           0     67.445     70.139     73.173     76.669     8
    r82    0     0     0              0           0     50.126     53.467     55.806     57.389     6
    r83    0     0     0              0           0          0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.027723    0.028258    0.031999    0.040974    0.048028     11.989     12.265      12.55
    r2     0.027723    0.028258    0.031999    0.041253    0.048028     12.223     12.501     12.787
    r3     0.027723    0.028258    0.031999    0.041641    0.048028     12.476     12.755     13.042
    r4     0.028805    0.029339    0.033081    0.042722    0.049108      12.72         13     13.289
    r5     0.029859    0.030394    0.034135    0.043802    0.050161     12.955     13.236     13.525
    r79     0.19737     0.19791     0.20163     0.21175     0.23145     35.417     37.362     39.409
    r80     0.19737     0.19791     0.20163     0.21175     0.23145     40.752     42.953     45.286
    r81     0.19737     0.19791     0.20163     0.21175     0.23145     48.909     52.039     55.022
    r82     0.19737     0.19791     0.20163     0.21175     0.23145     66.215     68.697     72.375
    r83     0.19737     0.19791     0.20163     0.21175     0.23145     116.33     122.15     128.17

[Phi_true] = snw_ds_main(mp_params, mp_controls, ap_ss, cons_emp_2020, mp_valpol_more_ss);
```

```
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1473.2344

% Get Matrixes
cl_st_precompute_list = {'a', ...
    'inc', 'inc_unemp', 'spouse_inc', 'spouse_inc_unemp', 'ref_earn_wageind_grid',...
    'ar_z_ctr_amz'};
% cl_st_precompute_list = {'a', ...
%     'inc', 'inc_unemp', 'spouse_inc', 'spouse_inc_unemp', 'ref_earn_wageind_grid',...
%     'ap_idx_lower_ss', 'ap_idx_higher_ss', 'ap_idx_lower_weight_ss'};
mp_controls('bl_print_precompute_verbose') = false;
[mp_precompute_res] = snw_hh_precompute(mp_params, mp_controls, cl_st_precompute_list, ap_ss, Phi_tr

Wage quintile cutoffs=0.4645     0.71528      1.0335      1.5632
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time cost=279.
```

9.2.2 Solve for 2019 Evuvw With 0 and 2 Checks

Solve for 0 and 2 checks, by finding the increase to asset state-space that is equivalent to the check increase, so that the problem can be solved without increasing the state-space.

```
% Call Function
welf_checks = 0;
[ev19_jaeemk_check0, ec19_jaeemk_check0, ev20_jaeemk_check0, ec20_jaeemk_check0] = snw_evuvw19_jaeem
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_emp_2020, ap_ss, cons_emp_2020, V_unemp_2020, cons_unemp_2020, mp_precompute_res);

Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bidenchk;welf_checks=0;TR=0.0015999;SNW_MP_PARAM=defa
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=0;TR=0.0015999;xi=0.5;b=0.5;SNW_MP_PARAM=default_doc
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=8.35
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bidenchk;SNW_MP_PARAM=default_docdense;SNW_MP_CON
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                  i    idx    ndim       numel       rowN      colN          sum         mean
                  -    ---    ----    ----------     ----    ---------    -----------    -------

    ec19_jaeemk   1     1      6      4.3173e+07      82     5.265e+05     1.9673e+08     4.5568
    ec20_jaeemk   2     2      6        4.37e+07      83     5.265e+05     2.3259e+08     5.3225
    ev19_jaeemk   3     3      6      4.3173e+07      82     5.265e+05    -6.4971e+08    -15.049
    ev20_jaeemk   4     4      6        4.37e+07      83     5.265e+05     -6.683e+08    -15.293

xxx TABLE:ec19_jaeemk xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.036494    0.036494    0.037029    0.041925    0.048857    12.017     12.289     12.569
    r2     0.036494    0.036494    0.037029    0.041745    0.049665    12.261     12.534     12.815
    r3     0.037912    0.037912    0.038127    0.041994    0.050655    12.495      12.77     13.052
    r4     0.039293    0.039293    0.039401    0.043382    0.052052    12.753     13.028     13.311
    r5     0.040635    0.040635     0.04064    0.044725    0.053494    13.002     13.278      13.56
    r78     0.19737     0.19737     0.19737     0.19737     0.19791     27.77     28.769      29.78
    r79     0.19737     0.19737     0.19737     0.19737     0.19737    30.426     31.659     32.732
    r80     0.19737     0.19737     0.19737     0.19737     0.19737    33.678     35.498     37.364
    r81     0.19737     0.19737     0.19737     0.19737     0.19737    40.112     41.394     43.173
    r82     0.19737     0.19737     0.19737     0.19737     0.19737    52.096     55.537     58.457

xxx TABLE:ec20_jaeemk xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498
```

```
        --------    --------    --------    --------    --------    -------    -------    -------

    r1  0.033462    0.033996    0.037408    0.043215    0.048855    12.242     12.527     12.819
    r2  0.033462    0.033996    0.037408    0.043883    0.049712    12.478     12.763     13.057
    r3  0.033462    0.033996    0.037604    0.045059    0.050801    12.731     13.018     13.313
    r4  0.034763    0.035298    0.038972    0.046376    0.052249    12.977     13.265      13.56
    r5  0.036032    0.036566    0.040303    0.047654    0.053654    13.213     13.501     13.796
    r79  0.19737     0.19791     0.20163     0.21175     0.23145    35.811     37.362     39.409
    r80  0.19737     0.19791     0.20163     0.21175     0.23145    40.752     42.953     45.286
    r81  0.19737     0.19791     0.20163     0.21175     0.23145    48.909     52.039     55.022
    r82  0.19737     0.19791     0.20163     0.21175     0.23145     66.71     69.193     72.375
    r83  0.19737     0.19791     0.20163     0.21175     0.23145    116.82     122.65     128.66

xxx TABLE:ev19_jaeemk xxxxxxxxxxxxxxxxxx
           c1         c2         c3         c4         c5        c526496      c526497      c526498
        -------    -------    -------    -------    -------    ---------    ---------    --------

    r1  -284.04    -284.04    -283.59    -279.25    -271.39      -4.3825       -4.288       -4.194
    r2  -273.99    -273.99    -273.54    -269.79    -262.27      -4.2743      -4.1834       -4.093
    r3  -263.81    -263.81    -263.65     -260.8    -253.57      -4.1639      -4.0766       -3.990
    r4  -254.62    -254.62    -254.55    -251.83    -245.12        -4.05      -3.9664       -3.883
    r5  -246.32    -246.32    -246.31     -243.7    -237.45      -3.9424      -3.8621       -3.782
    r78 -13.649    -13.649    -13.649    -13.649    -13.635     -0.27318     -0.26108      -0.2497
    r79  -12.29     -12.29     -12.29     -12.29     -12.29     -0.21858     -0.20783      -0.1987
    r80 -10.611    -10.611    -10.611    -10.611    -10.611     -0.16128     -0.15409      -0.1473
    r81 -8.3555    -8.3555    -8.3555    -8.3555    -8.3555     -0.10114    -0.097402     -0.09344
    r82 -5.0715    -5.0715    -5.0715    -5.0715    -5.0715    -0.044205    -0.041465     -0.03941

xxx TABLE:ev20_jaeemk xxxxxxxxxxxxxxxxxx
           c1         c2         c3         c4         c5        c526496      c526497      c526498
        -------    -------    -------    -------    -------    ---------    ---------    --------

    r1  -297.16    -296.66    -293.58    -287.23    -277.75      -4.3618      -4.2675       -4.174
    r2  -287.62    -287.12    -284.04    -277.74    -268.53      -4.2552      -4.1645       -4.074
    r3  -278.06    -277.57    -274.51    -268.33    -259.57      -4.1462       -4.059       -3.972
    r4   -268.3    -267.84    -264.99    -259.19    -250.96      -4.0343      -3.9508       -3.868
    r5  -259.47    -259.05    -256.38    -250.92    -243.16      -3.9288      -3.8486       -3.769
    r79 -13.642    -13.628    -13.535    -13.298    -12.896     -0.22092     -0.21058      -0.2008
    r80 -12.283    -12.269    -12.176    -11.939    -11.537     -0.16979     -0.16182       -0.154
    r81 -10.605    -10.591    -10.498    -10.261    -9.8589     -0.11712     -0.11163      -0.1064
    r82 -8.3494    -8.3358    -8.2424    -8.0055    -7.6035    -0.065333    -0.062242      -0.0593
    r83 -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.020968    -0.019972     -0.01903
```

```
% Call Function
welf_checks = 2;
[ev19_jaeemk_check2, ec19_jaeemk_check2, ev20_jaeemk_check2, ec20_jaeemk_check2] = snw_evuvw19_jaeem
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_emp_2020,  ap_ss, cons_emp_2020, V_unemp_2020, cons_unemp_2020, mp_precompute_res);
```

```
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bidenchk;welf_checks=2;TR=0.0015999;SNW_MP_PARAM=defa
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=2;TR=0.0015999;xi=0.5;b=0.5;SNW_MP_PARAM=default_doc
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=7.96
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bidenchk;SNW_MP_PARAM=default_docdense;SNW_MP_CON
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                   i    idx    ndim      numel       rowN      colN           sum          mean
```

ec19_jaeemk	1	1	6	4.3173e+07	82	5.265e+05	1.9675e+08	4.5574
ec20_jaeemk	2	2	6	4.37e+07	83	5.265e+05	2.3261e+08	5.323
ev19_jaeemk	3	3	6	4.3173e+07	82	5.265e+05	-6.4916e+08	-15.036
ev20_jaeemk	4	4	6	4.37e+07	83	5.265e+05	-6.6772e+08	-15.28

xxx TABLE:ec19_jaeemk xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	0.039371	0.039371	0.03984	0.043617	0.0497	12.017	12.289	12.569
r2	0.039522	0.039522	0.040022	0.043796	0.050541	12.261	12.534	12.815
r3	0.040961	0.040961	0.041164	0.044279	0.051563	12.496	12.77	13.052
r4	0.042361	0.042361	0.042463	0.045683	0.05298	12.753	13.028	13.311
r5	0.043704	0.043704	0.043709	0.04707	0.054441	13.002	13.278	13.561
r78	0.20057	0.20057	0.20057	0.20057	0.20111	27.771	28.77	29.781
r79	0.20057	0.20057	0.20057	0.20057	0.20057	30.427	31.66	32.733
r80	0.20057	0.20057	0.20057	0.20057	0.20057	33.68	35.5	37.365
r81	0.20057	0.20057	0.20057	0.20057	0.20057	40.114	41.396	43.176
r82	0.20057	0.20057	0.20057	0.20057	0.20057	52.099	55.54	58.46

xxx TABLE:ec20_jaeemk xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	0.036428	0.036915	0.039238	0.044128	0.049481	12.243	12.527	12.82
r2	0.036428	0.036915	0.039449	0.044827	0.050406	12.478	12.763	13.057
r3	0.036567	0.037083	0.039954	0.045988	0.051564	12.732	13.018	13.313
r4	0.037916	0.038441	0.041306	0.047327	0.053041	12.977	13.265	13.56
r5	0.03923	0.039763	0.04262	0.048625	0.054472	13.213	13.502	13.797
r79	0.20057	0.20111	0.20483	0.21495	0.23328	35.812	37.363	39.41
r80	0.20057	0.20111	0.20483	0.21495	0.23328	40.753	42.954	45.288
r81	0.20057	0.20111	0.20483	0.21495	0.23328	48.911	52.041	55.024
r82	0.20057	0.20111	0.20483	0.21495	0.23352	66.711	69.194	72.378
r83	0.20057	0.20111	0.20483	0.21495	0.23465	116.83	122.65	128.67

xxx TABLE:ev19_jaeemk xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	-281.59	-281.59	-281.18	-277.52	-270.11	-4.3824	-4.288	-4.194
r2	-271.55	-271.55	-271.16	-268.06	-261.04	-4.2742	-4.1833	-4.093
r3	-261.56	-261.56	-261.41	-259.11	-252.39	-4.1639	-4.0766	-3.990
r4	-252.52	-252.52	-252.45	-250.24	-244	-4.0499	-3.9663	-3.883
r5	-244.35	-244.35	-244.35	-242.21	-236.39	-3.9423	-3.8621	-3.782
r78	-13.568	-13.568	-13.568	-13.568	-13.555	-0.27317	-0.26107	-0.2497
r79	-12.209	-12.209	-12.209	-12.209	-12.209	-0.21857	-0.20783	-0.1987
r80	-10.531	-10.531	-10.531	-10.531	-10.531	-0.16127	-0.15408	-0.1473
r81	-8.2749	-8.2749	-8.2749	-8.2749	-8.2749	-0.10114	-0.097398	-0.0934
r82	-4.991	-4.991	-4.991	-4.991	-4.991	-0.044202	-0.041463	-0.03941

xxx TABLE:ev20_jaeemk xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	-294.47	-294.03	-291.58	-285.7	-276.56	-4.3618	-4.2675	-4.174
r2	-284.93	-284.49	-282.06	-276.25	-267.39	-4.2552	-4.1644	-4.074

```
    r3     -275.39    -274.95    -272.56    -266.91    -258.48      -4.1462       -4.059       -3.972
    r4     -265.81     -265.4    -263.16    -257.86    -249.93      -4.0343      -3.9507      -3.868
    r5     -257.15    -256.77    -254.66    -249.67    -242.18      -3.9287      -3.8486      -3.769
    r79    -13.561    -13.548     -13.46    -13.233    -12.841     -0.22091     -0.21058     -0.2008
    r80    -12.203    -12.189    -12.101    -11.874    -11.482     -0.16978     -0.16181     -0.154
    r81    -10.524    -10.511    -10.423    -10.196    -9.8048     -0.11712     -0.11163     -0.1064
    r82    -8.2689    -8.2556    -8.1674    -7.9402    -7.5499    -0.065331    -0.062241    -0.05935
    r83    -4.9861    -4.9727    -4.8846    -4.6573    -4.2682    -0.020967    -0.019972    -0.01903
```

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check = ev19_jaeemk_check2 - ev19_jaeemk_check0;
mn_MPC_U_gain_share_check = (ec19_jaeemk_check2 - ec19_jaeemk_check0)./(welf_checks*mp_params('TR'))
```

9.2.3 Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:99;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

9.2.4 Analyze Difference in V and C with Check

The difference between V and V with Check, marginal utility gain given the check.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

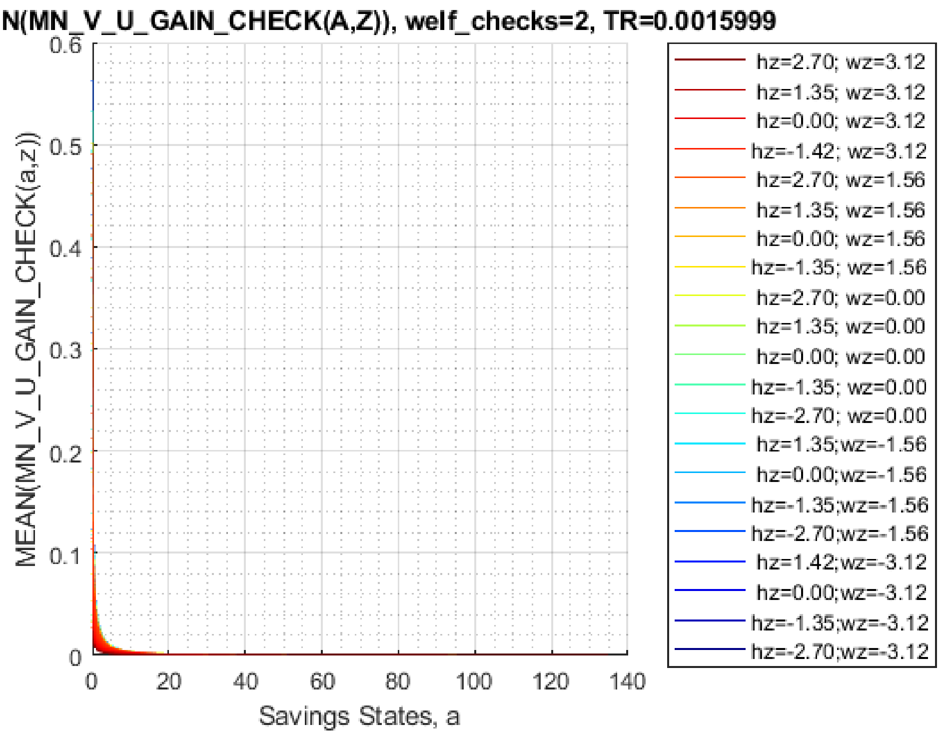
MEAN(MN_V_GAIN_CHECK(A,Z))

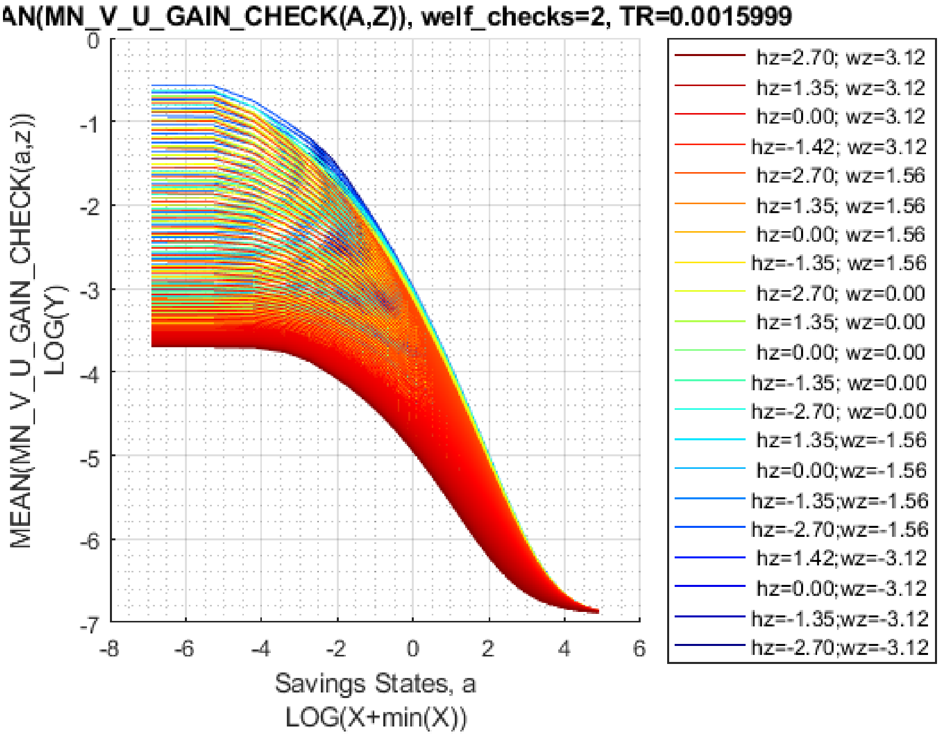
Tabulate value and policies along savings and shocks:

```
% Set
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_par
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0015999  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---
```

1	0	0.56295	0.52208	0.47698	0.43181	0.38933

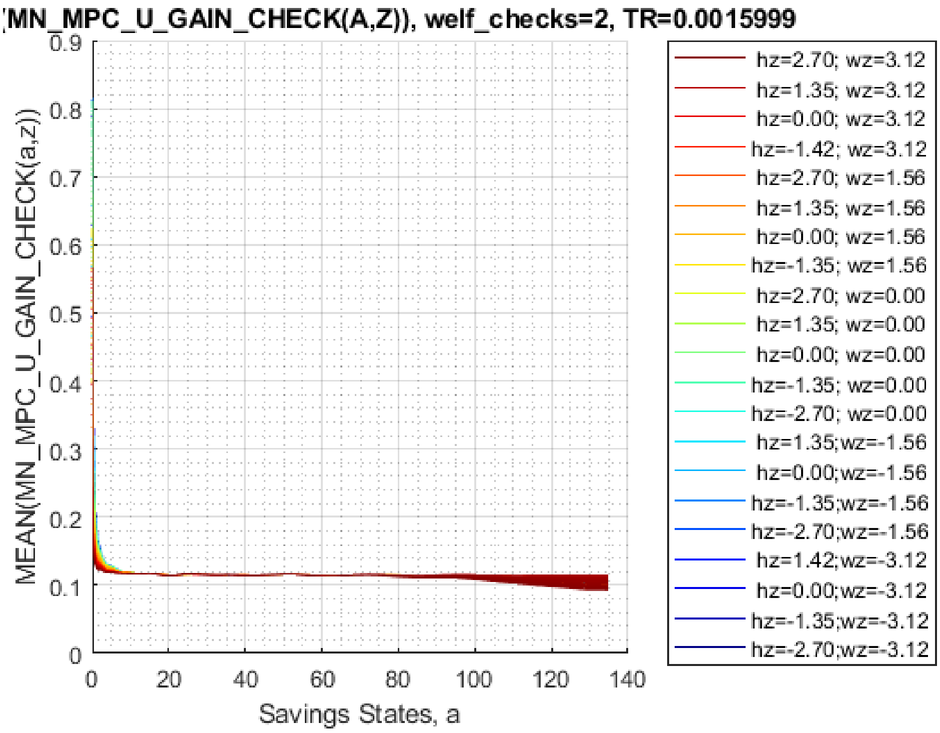
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

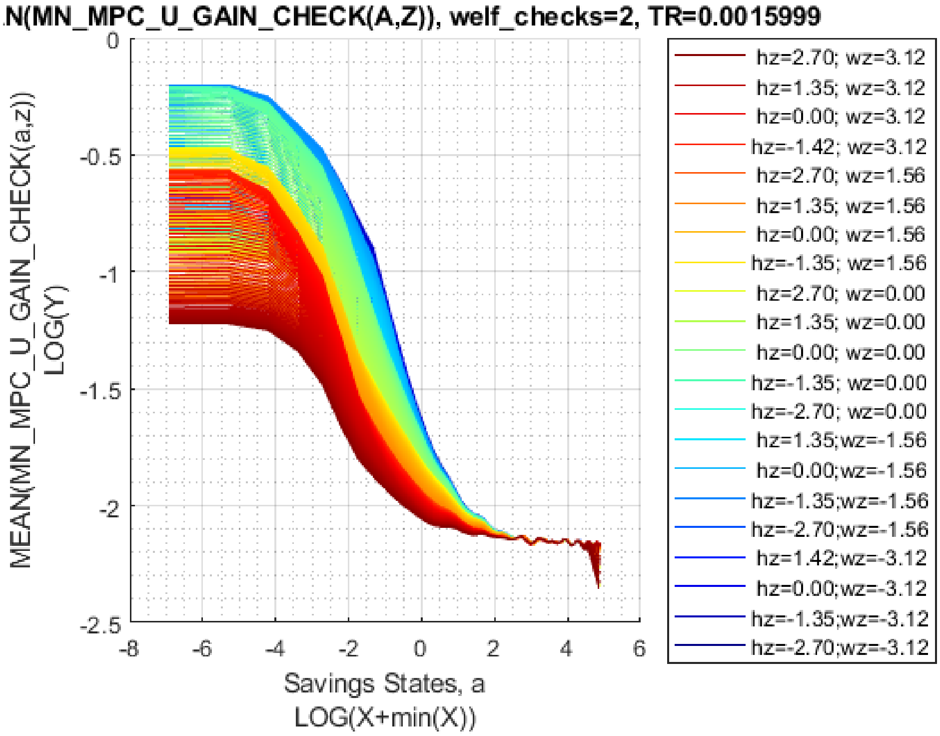




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





9.2.5 Analyze Marginal Value and MPC over Y(a,eta), Conditional On Kids, Marry, Age, Education

Income is generated by savings and shocks, what are the income levels generated by all the shock and savings points conditional on kids, marital status, age and educational levels. Plot on the Y axis MPC, and plot on the X axis income levels, use colors to first distinguish between different a levels, then use colors to distinguish between different eta levles.

Set Up date, Select Age 37vn

, unmarried, no kids, lower education:

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
% 38 year old, unmarried, no kids, lower educated
% Only Household Head Shock Matters so select up to 'n_eta_H_grid'
mn_total_inc_jemk = total_inc_VFI(19,:,1:mp_params('n_eta_H_grid'),1,1,1);
mn_V_W_gain_check_use = ev19_jaeemk_check2 - ev19_jaeemk_check0;
mn_C_W_gain_check_use = ec19_jaeemk_check2 - ec19_jaeemk_check0;
```

Select Age, Education, Marital, Kids Count:s

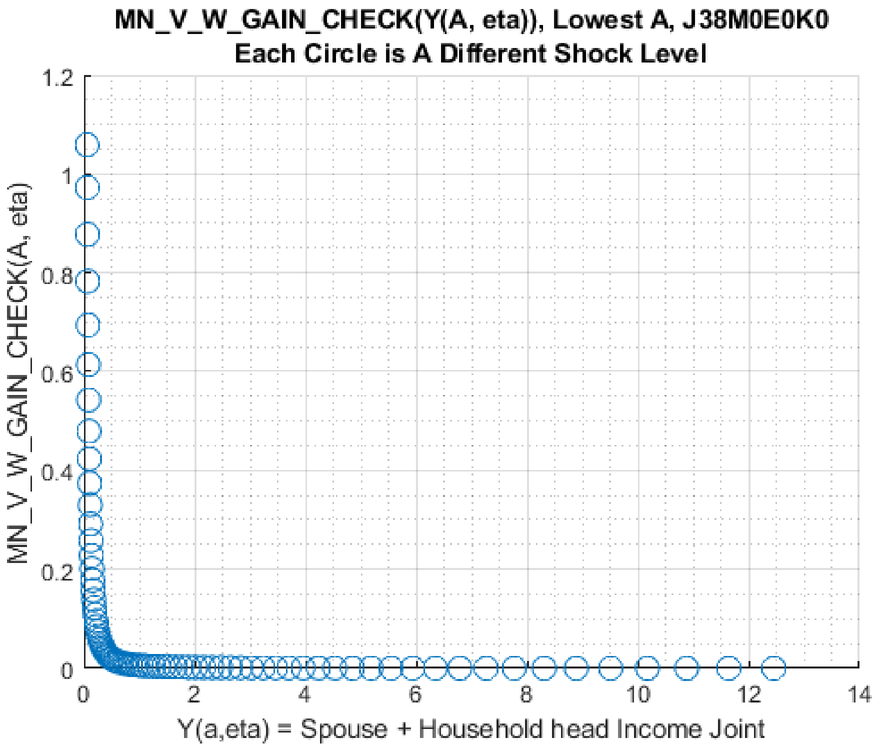
```
% Selections
it_age = 21; % +18
it_marital = 1; % 1 = unmarried
it_kids = 1; % 1 = kids is zero
it_educ = 1; % 1 = lower education
% Select: NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
mn_C_W_gain_check_jemk = mn_C_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ, it_m
mn_V_W_gain_check_jemk = mn_V_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ, it_m
% Reshape, so shock is the first dim, a is the second
mt_total_inc_jemk = permute(mn_total_inc_jemk,[3,2,1]);
mt_C_W_gain_check_jemk = permute(mn_C_W_gain_check_jemk,[3,2,1]);
mt_C_W_gain_check_jemk(mt_C_W_gain_check_jemk<=1e-10) = 1e-10;
mt_V_W_gain_check_jemk = permute(mn_V_W_gain_check_jemk,[3,2,1]);
mt_V_W_gain_check_jemk(mt_V_W_gain_check_jemk<=1e-10) = 1e-10;
% Generate meshed a and shock grid
```

```
[mt_eta_H, mt_a] = ndgrid(eta_H_grid(1:mp_params('n_eta_H_grid')), agrid);
```

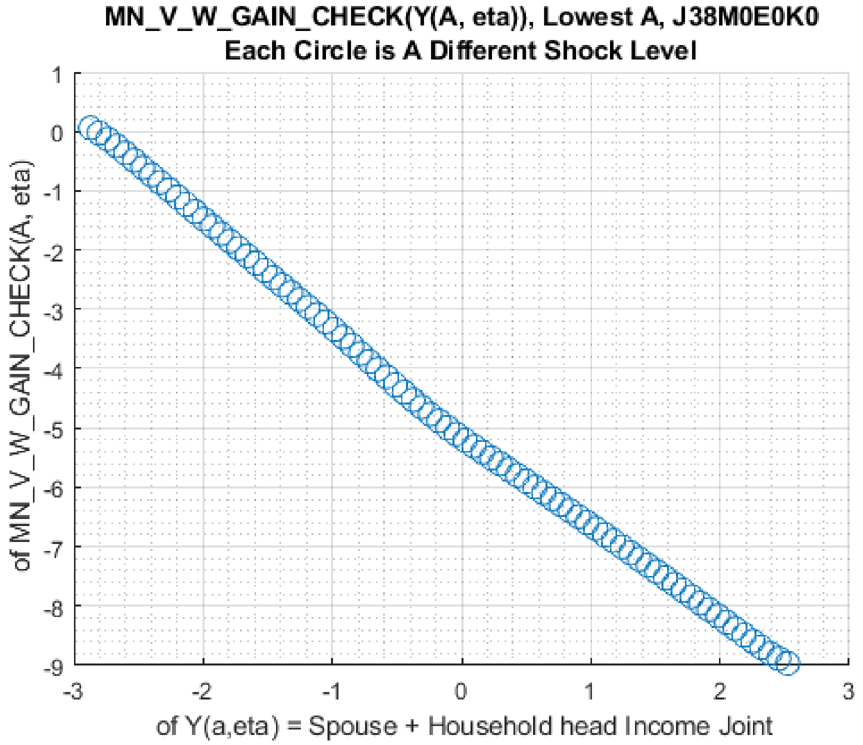
9.2.6 Marginal Value Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
it_a = 1;
scatter((mt_total_inc_jemk(:,it_a)), (mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

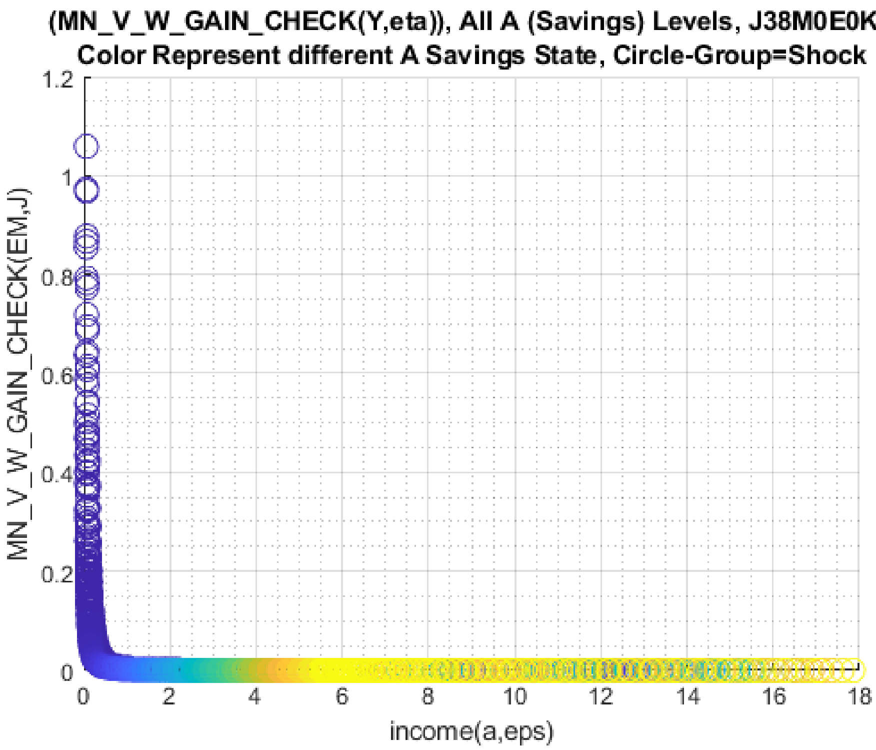



```
figure();
it_shock = 1;
scatter(log(mt_total_inc_jemk(:,it_a)), log(mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel(' of Y(a,eta) = Spouse + Household head Income Joint');
ylabel(' of MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

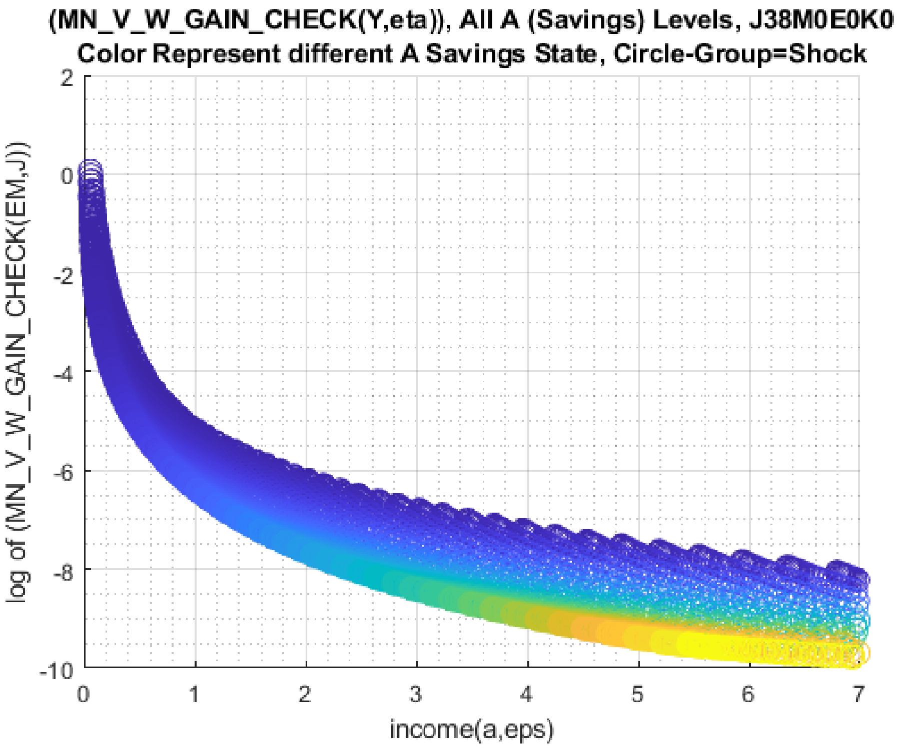


Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_V\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```



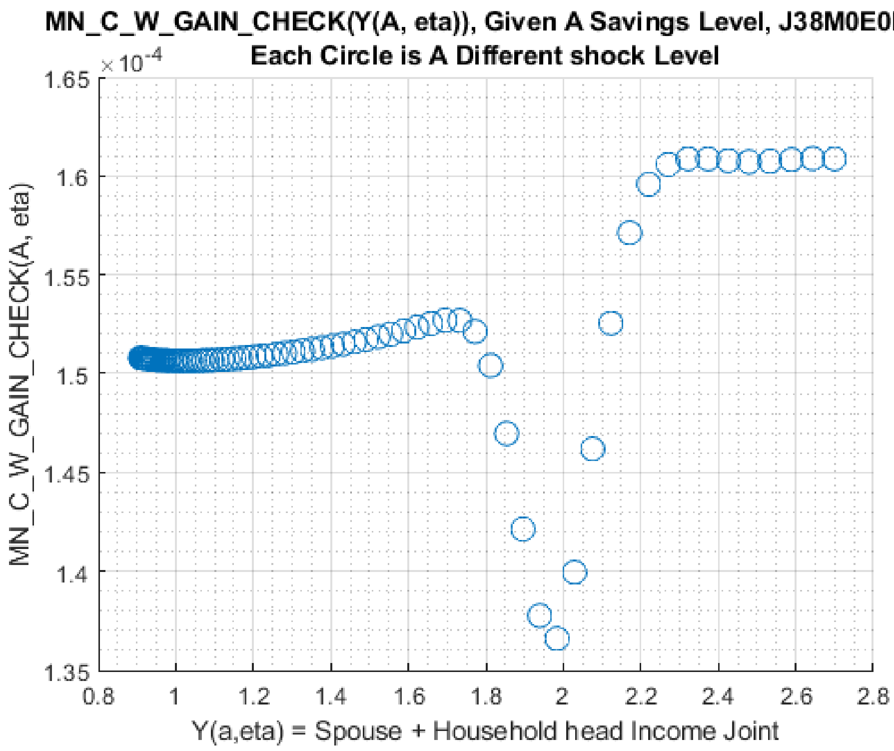
```
figure();
scatter((mt_total_inc_jemk(:)), log(mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
xlim([0,7]);
grid on;
grid minor;
```



9.2.7 Marginal Consumption Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

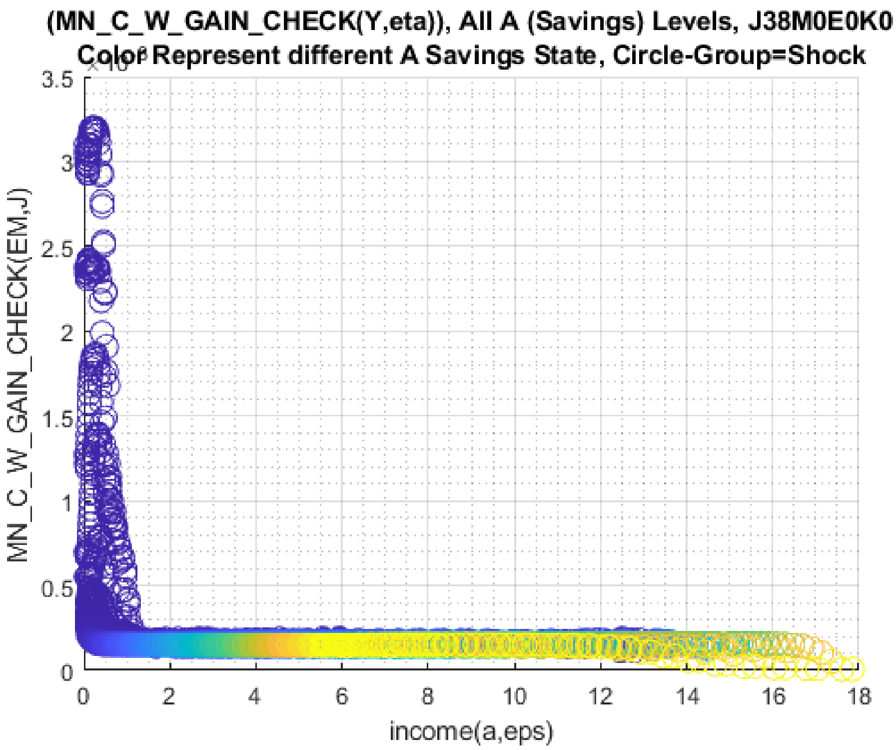
How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
it_a = 50;
scatter(log(mt_total_inc_jemk(:,it_a)), mt_C_W_gain_check_jemk(:,it_a), 100);
title({'MN\_C\_W\_GAIN\_CHECK(Y(A, eta)), Given A Savings Level, J38M0E0K0', ...
    'Each Circle is A Different shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_C\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

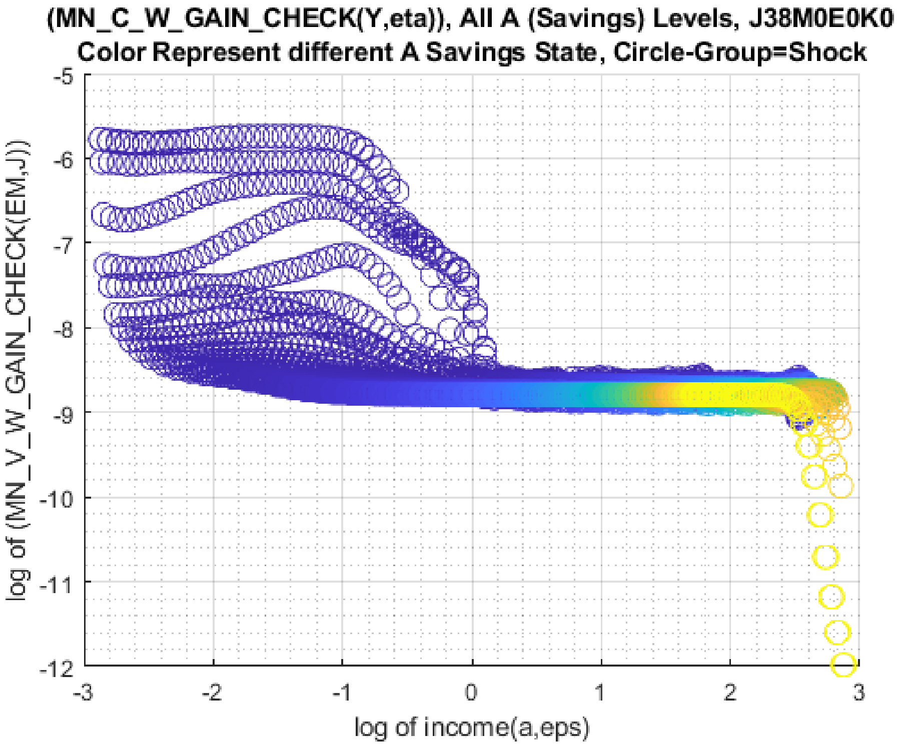


Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_C_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_C\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```



```
figure();
scatter(log(mt_total_inc_jemk(:)), log(mt_C_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('log of income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
grid on;
grid minor;
```



9.2.8 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
```

```
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_
```

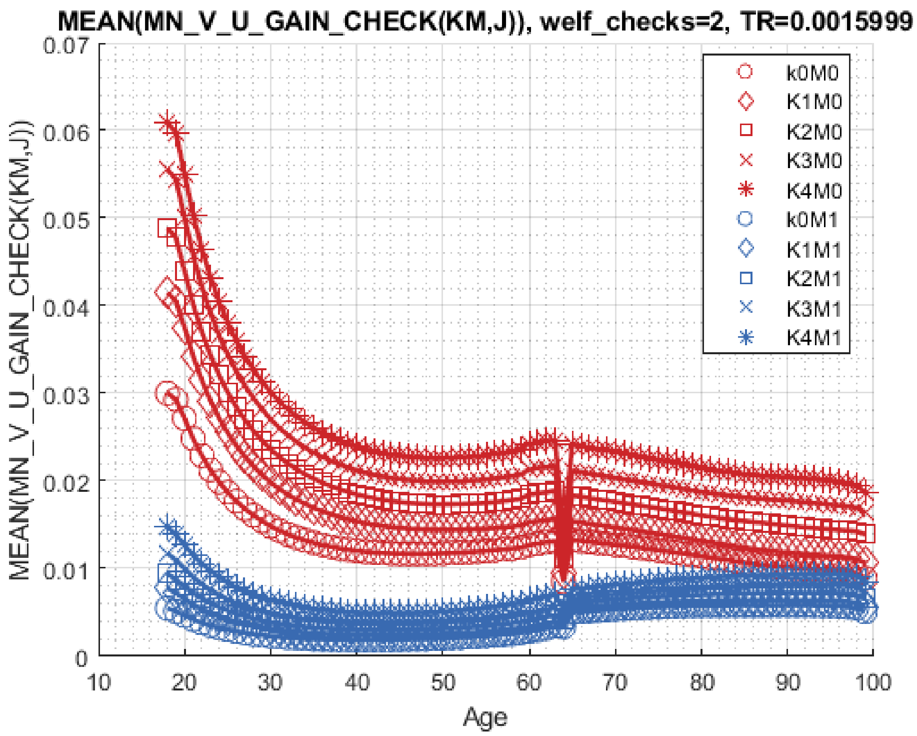
```
xxx  MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0015999  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0        0.030027       0.029309       0.027141       0.024832       0.022901
      2        2       0         0.04151        0.04055       0.037518       0.034247       0.031504
      3        3       0        0.048832       0.047813       0.043942       0.040179       0.037025
      4        4       0        0.055554       0.054443       0.050039       0.045784       0.042216
      5        5       0          0.0609        0.05977       0.054981        0.05038       0.046522
      6        1       1       0.0055093      0.0051081      0.0046334      0.0041967      0.0038272
      7        2       1       0.0077846      0.0072287      0.0065562      0.0059314      0.0054057
      8        3       1       0.0094266       0.008771      0.0079723      0.0072201      0.0065859
      9        4       1        0.011763       0.010976       0.009988      0.0090498      0.0082597
     10        5       1        0.014764       0.013879       0.012683       0.011539       0.010569
```

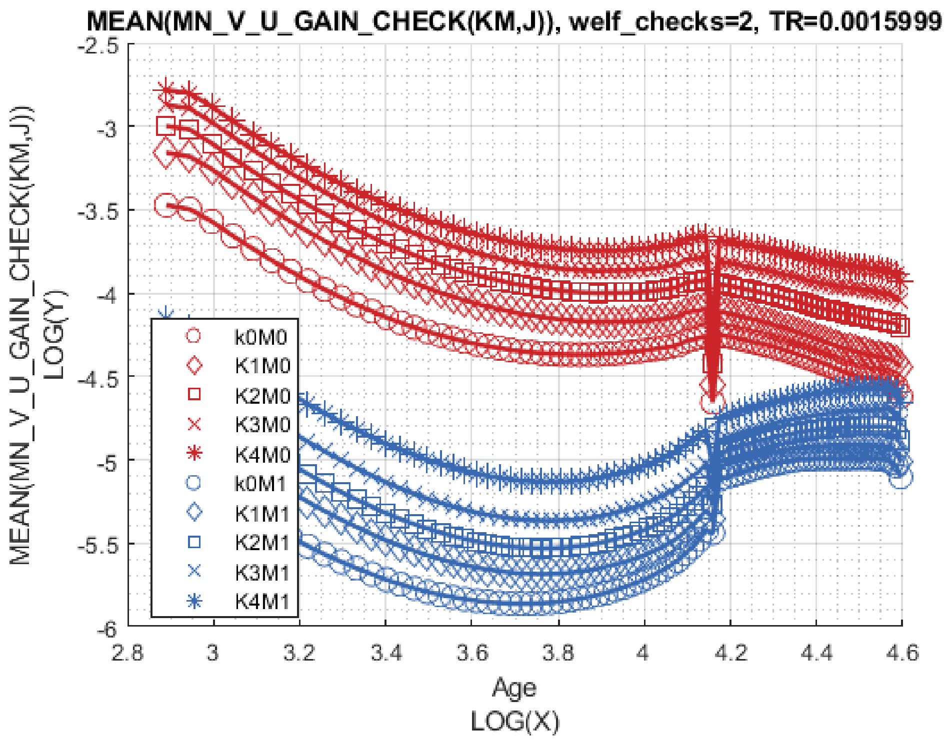
```
% Consumption Function
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd
```

```
xxx  MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0015999  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0        0.08445       0.099691        0.10757        0.10468        0.10247
      2        2       0       0.096015        0.11123        0.12103        0.11789        0.11535
      3        3       0        0.10769        0.12614        0.13451        0.13081        0.12755
      4        4       0        0.11389        0.13321        0.14167         0.1377        0.13399
      5        5       0         0.1198        0.14051        0.14851          0.144        0.13992
      6        1       1       0.096558        0.10433         0.1066        0.10427         0.1019
      7        2       1        0.10023        0.10921        0.11152        0.10928        0.10824
      8        3       1        0.10587        0.11747         0.1188        0.11732        0.11596
      9        4       1        0.11202        0.12194        0.12444         0.1225        0.11996
     10        5       1        0.12325        0.13304        0.13672        0.13148        0.12849
```

Graph Mean Values:

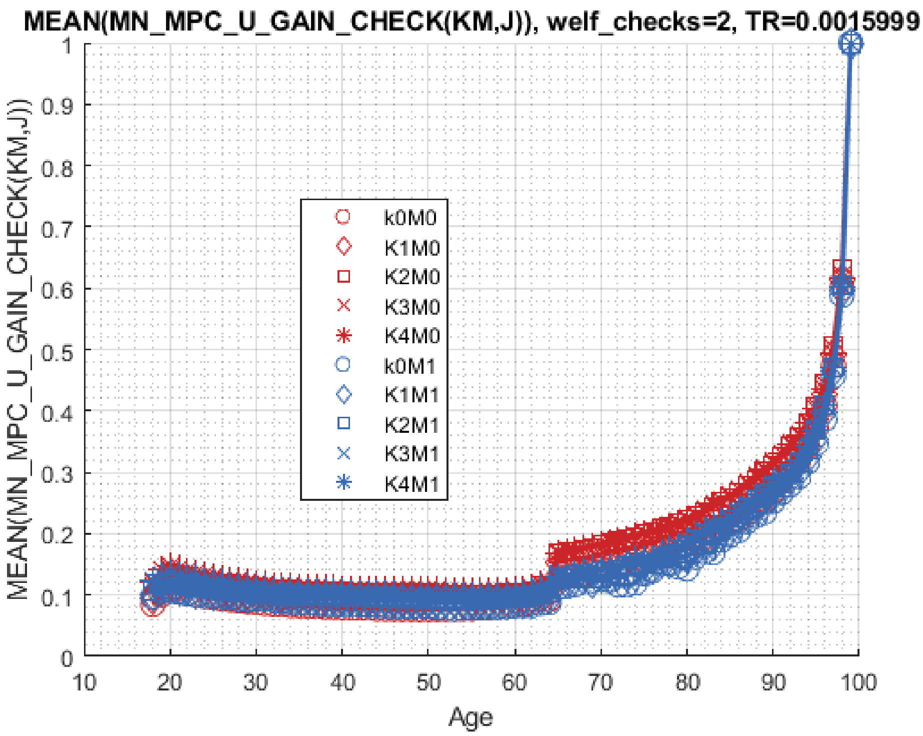
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

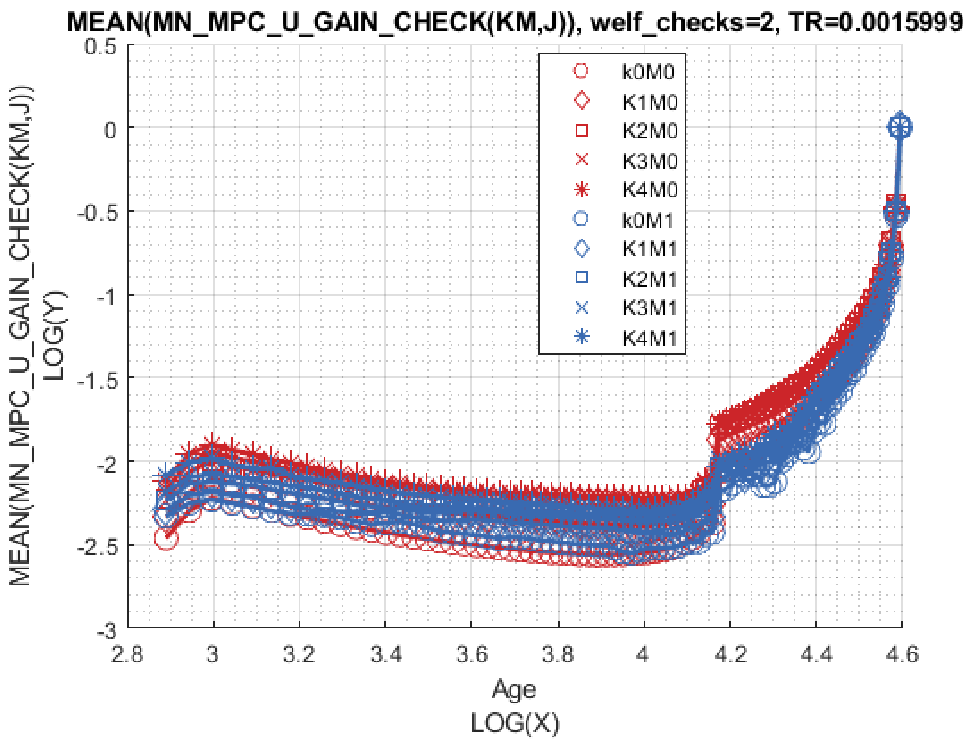




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





9.2.9 Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

```
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

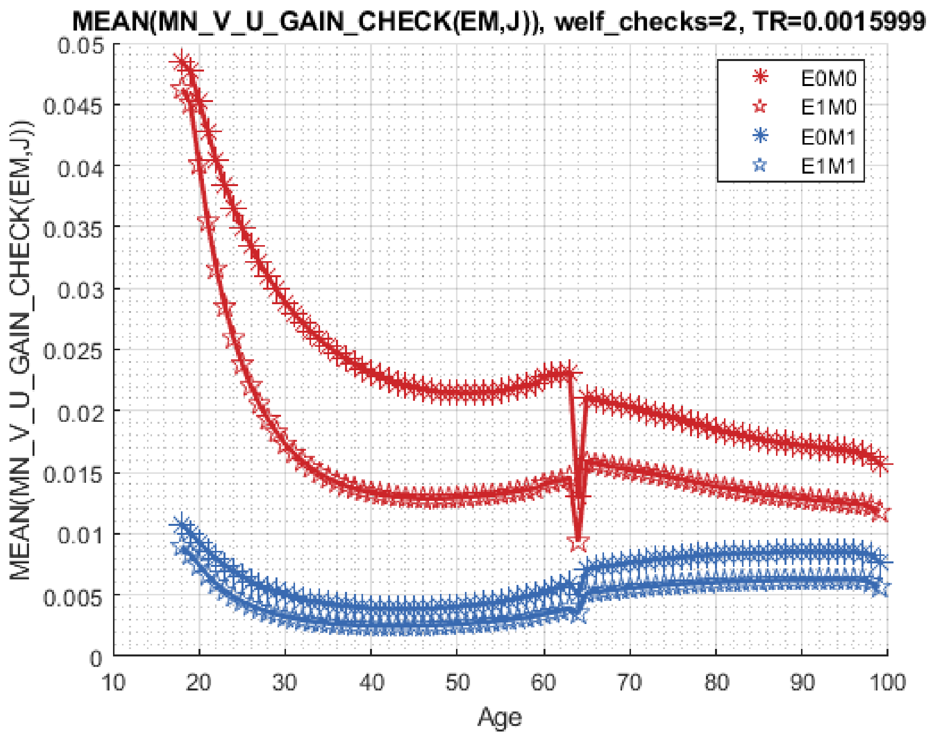
MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

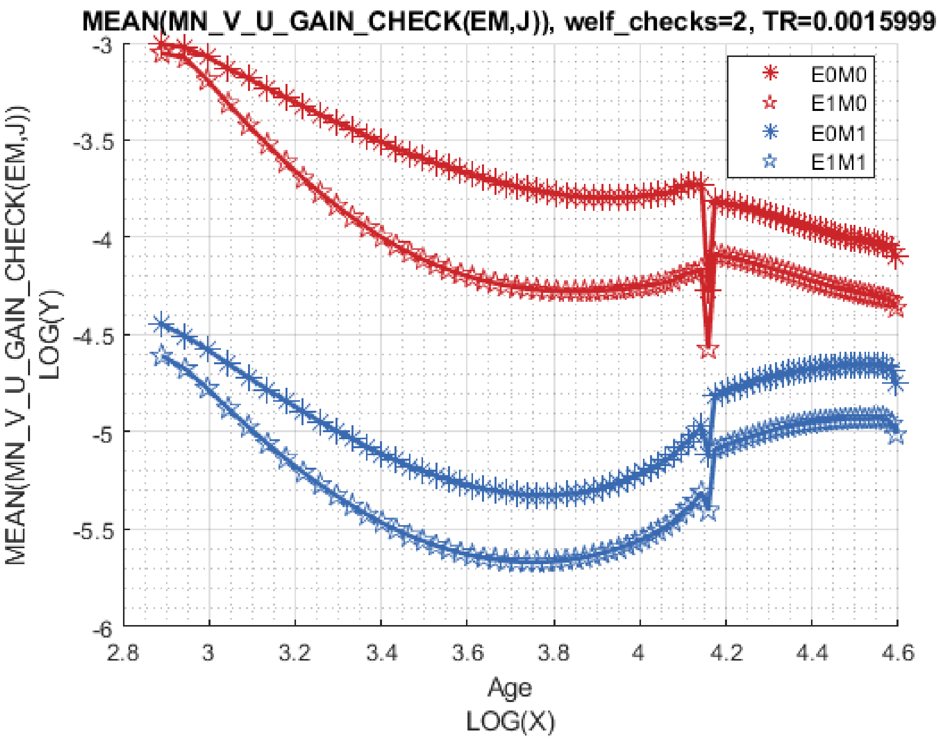
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0015999  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0         0.04847        0.04767       0.045314       0.042735       0.040445
      2       1       0         0.04626       0.045083       0.040135       0.035433       0.031622
      3       0       1        0.010726       0.010058      0.0092963      0.0085699      0.0079386
      4       1       1       0.0089734      0.0083275      0.0074368      0.0066048      0.0059203

% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd

xxx  MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0015999  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        0.092366        0.10262         0.1087        0.10789        0.10721
      2       1       0         0.11637         0.1417        0.15261        0.14614         0.1405
      3       0       1        0.098134        0.10328         0.1058        0.10505        0.10409
      4       1       1         0.11704        0.13112        0.13343        0.12889        0.12573
```

Graph Mean Values:

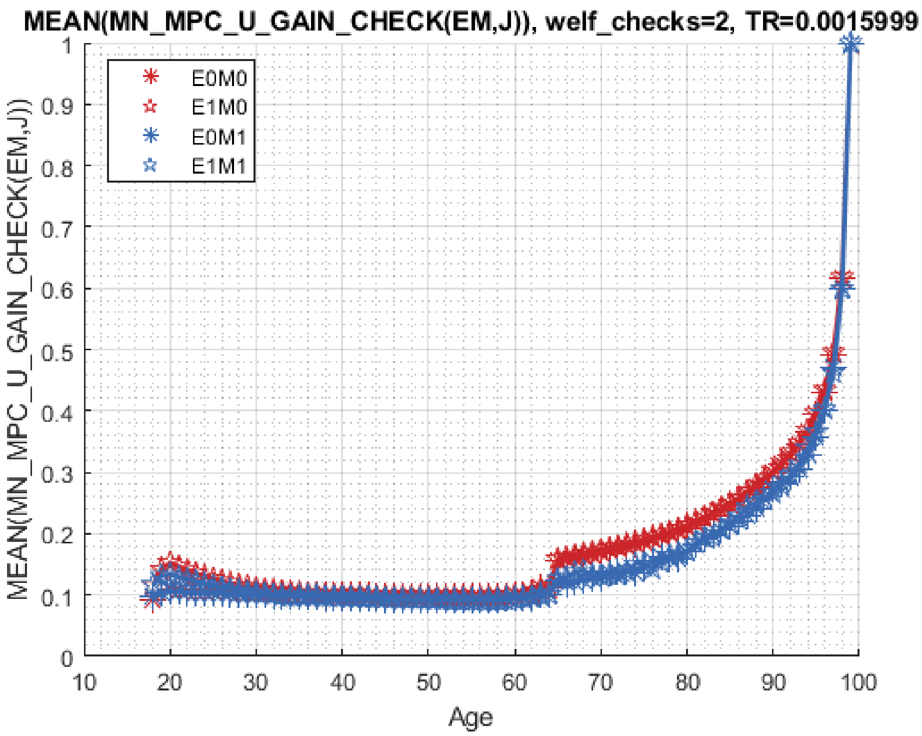
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

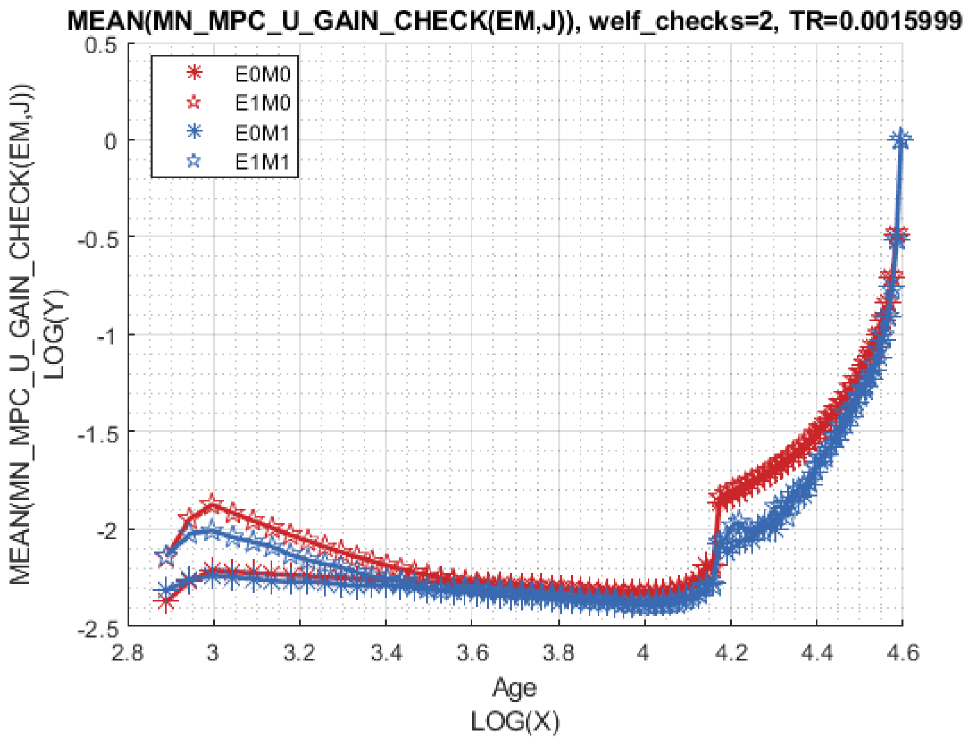




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





9.3 2008 Value and Optimal Savings and Consumption Given Stimulus

This is the example vignette for function: **snw_v08_jaeemk** from the **PrjOptiSNW Package.** This is simialr to **snw_evuvw20_jaeemk**, but for the 2008 Bush stimulus. **snw_v08p08_jaeemk** already solved for optimal policy and value functions in 2008, given expected unemployment shock in 2009. In this function, given some stimulus amount, we use **snw_a4chk_wrk_bisec_vec** to compute the updated optimal V and C in 2008 given the stimulus amount, based on the alues for V and C without stimulus computed by **snw_v08p08_jaeemk**.

Note that **snw_a4chk_wrk_bisec_vec** computes the adjustment in the savings state that would be equivalent to the increase in stimulus amount (which is not a state variable) to current resources, this is faster than resolving 2008 optimal V and C at specific stimulus check amount levels.

Note **snw_evuvw20_jaeemk** has EVUVW, but here, we only have V08, because in the 2020 problem, households receive checks ex-post of the COVID MIT shocks in 2020 and the EVUVW is the weighted average in V between the MIT unemployed and non-shock employed state. In 2008, however, there are no shocks yet, the state-space is the same as normal, the only difference is that households might receive stimulus checks from Bush. The Bush stimulus is provided ex-ante of the shock realization. The 2009 shocks due to the great recession is not a MIT shock, but expected shock. The effect of the 2009 shock on consumption, savings is solved by **snw_v08p08_jaeemk**. The expectation over shock, in another word, for the **snw_v08_jaeemk** is already included in EV' in 2008 for 2009.

9.3.1 Solve 2008 Value and Policy Function with SNW_V08p08_JAEEMK

Solve for the Value and Policy functions in 2008 given expected unemployment shock that is specific to age and education group in 2009, no stimulus amounts.

First, set various parameters

```
% 1. Paramters
% Parameters
mp_more_inputs = containers.Map('KeyType','char', 'ValueType','any');
mp_more_inputs('fl_ss_non_college') = 0.225;
mp_more_inputs('fl_ss_college') = 0.271;
fl_p50_hh_income_07 = 54831;
mp_more_inputs('fl_scaleconvertor') = fl_p50_hh_income_07;

% st_param_group = 'default_small';
% st_param_group = 'default_dense';
st_param_group = 'default_docdense';
mp_params = snw_mp_param(st_param_group, false, 'tauchen', false, 8, 8, mp_more_inputs);
mp_params('beta') = 0.95;

% Control parameters
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_v08p08_jaeemk') = false;
mp_controls('bl_print_v08p08_jaeemk_verbose') = false;
mp_controls('bl_print_v08_jaeemk') = true;
mp_controls('bl_print_v08_jaeemk_verbose') = true;
```

Second, solve the steady-state problem, same as employed results in 2009.

```
% 2. Solve value steady state (2009 employed)
[V_VFI_ss, ap_VFI_ss, cons_VFI_ss, mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_control

Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=530.

inc_VFI = mp_valpol_more_ss('inc_VFI');
spouse_inc_VFI = mp_valpol_more_ss('spouse_inc_VFI');
total_inc_VFI = inc_VFI + spouse_inc_VFI;
V_emp_2009 = V_VFI_ss;
% Solve for probability mass, needed for pre-compute
```

```
[Phi_true] = snw_ds_main_vec(mp_params, mp_controls, ap_VFI_ss, cons_VFI_ss, mp_valpol_more_ss);
```

```
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1271.649
```

```
% Solve for household head and spouse income and sum to total income.
inc_VFI = mp_valpol_more_ss('inc_VFI');
spouse_inc_VFI = mp_valpol_more_ss('spouse_inc_VFI');
total_inc_VFI = inc_VFI + spouse_inc_VFI;
```

Third, solve the unemployment problem in 2009. With 2009 specific unemployment parameters calibrated and found from data. Using b calibrated by **snw_calibrate_2009_b**.

```
% 3. Solve value unemployed 2009
% Set Unemployment Related Variables
mp_params('xi') = 0.532;
% Calibrated by snw_calibrate_2009_b
mp_params('b') = 0.37992;
mp_params('a2_covidyr') = mp_params('a2_greatrecession_2009');
mp_params('TR') = 100/fl_p50_hh_income_07; % Value of a stimulus check (can receive multiple checks)
[V_unemp_2009] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_VFI_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
```

Fourth, solve the 2008 problem, with 2008-specific value and policy functions.

```
% 4. Value and Optimal choice in 2009
[V_2008, ap_2008, cons_2008, ev_empshk_2009] = ...
    snw_v08p08_jaeemk(mp_params, mp_controls, V_emp_2009, V_unemp_2009);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
Completed SNW_V08P08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=522.6257
```

```
% 5. pre-compute
cl_st_precompute_list = {'a', ...
    'inc', 'inc_unemp', 'spouse_inc', 'spouse_inc_unemp', 'ref_earn_wageind_grid'};
mp_controls('bl_print_precompute_verbose') = false;
[mp_precompute_res] = snw_hh_precompute(mp_params, mp_controls, cl_st_precompute_list, ap_VFI_ss, Ph
```

```
Wage quintile cutoffs=0.4645     0.71528      1.0335      1.5632
SNW_HH_PRECOMPUTE: Finished Age Group:1 of 82, time-this-age:3.2746
SNW_HH_PRECOMPUTE: Finished Age Group:2 of 82, time-this-age:3.2852
SNW_HH_PRECOMPUTE: Finished Age Group:3 of 82, time-this-age:3.3869
SNW_HH_PRECOMPUTE: Finished Age Group:4 of 82, time-this-age:3.4589
SNW_HH_PRECOMPUTE: Finished Age Group:5 of 82, time-this-age:3.254
SNW_HH_PRECOMPUTE: Finished Age Group:6 of 82, time-this-age:3.3465
SNW_HH_PRECOMPUTE: Finished Age Group:7 of 82, time-this-age:3.4109
SNW_HH_PRECOMPUTE: Finished Age Group:8 of 82, time-this-age:3.4221
SNW_HH_PRECOMPUTE: Finished Age Group:9 of 82, time-this-age:3.4152
SNW_HH_PRECOMPUTE: Finished Age Group:10 of 82, time-this-age:3.4065
SNW_HH_PRECOMPUTE: Finished Age Group:11 of 82, time-this-age:3.1869
SNW_HH_PRECOMPUTE: Finished Age Group:12 of 82, time-this-age:3.3572
SNW_HH_PRECOMPUTE: Finished Age Group:13 of 82, time-this-age:3.4748
SNW_HH_PRECOMPUTE: Finished Age Group:14 of 82, time-this-age:3.3824
SNW_HH_PRECOMPUTE: Finished Age Group:15 of 82, time-this-age:3.4573
SNW_HH_PRECOMPUTE: Finished Age Group:16 of 82, time-this-age:3.3233
SNW_HH_PRECOMPUTE: Finished Age Group:17 of 82, time-this-age:3.3965
SNW_HH_PRECOMPUTE: Finished Age Group:18 of 82, time-this-age:3.3613
SNW_HH_PRECOMPUTE: Finished Age Group:19 of 82, time-this-age:3.3402
SNW_HH_PRECOMPUTE: Finished Age Group:20 of 82, time-this-age:3.479
SNW_HH_PRECOMPUTE: Finished Age Group:21 of 82, time-this-age:3.2899
```

```
SNW_HH_PRECOMPUTE: Finished Age Group:22 of 82, time-this-age:3.3222
SNW_HH_PRECOMPUTE: Finished Age Group:23 of 82, time-this-age:3.3025
SNW_HH_PRECOMPUTE: Finished Age Group:24 of 82, time-this-age:3.3682
SNW_HH_PRECOMPUTE: Finished Age Group:25 of 82, time-this-age:3.3555
SNW_HH_PRECOMPUTE: Finished Age Group:26 of 82, time-this-age:3.3701
SNW_HH_PRECOMPUTE: Finished Age Group:27 of 82, time-this-age:3.5453
SNW_HH_PRECOMPUTE: Finished Age Group:28 of 82, time-this-age:3.2218
SNW_HH_PRECOMPUTE: Finished Age Group:29 of 82, time-this-age:3.3947
SNW_HH_PRECOMPUTE: Finished Age Group:30 of 82, time-this-age:3.3024
SNW_HH_PRECOMPUTE: Finished Age Group:31 of 82, time-this-age:3.3699
SNW_HH_PRECOMPUTE: Finished Age Group:32 of 82, time-this-age:3.4931
SNW_HH_PRECOMPUTE: Finished Age Group:33 of 82, time-this-age:3.3584
SNW_HH_PRECOMPUTE: Finished Age Group:34 of 82, time-this-age:3.4127
SNW_HH_PRECOMPUTE: Finished Age Group:35 of 82, time-this-age:3.4113
SNW_HH_PRECOMPUTE: Finished Age Group:36 of 82, time-this-age:3.2095
SNW_HH_PRECOMPUTE: Finished Age Group:37 of 82, time-this-age:3.4244
SNW_HH_PRECOMPUTE: Finished Age Group:38 of 82, time-this-age:3.5012
SNW_HH_PRECOMPUTE: Finished Age Group:39 of 82, time-this-age:3.2675
SNW_HH_PRECOMPUTE: Finished Age Group:40 of 82, time-this-age:3.2625
SNW_HH_PRECOMPUTE: Finished Age Group:41 of 82, time-this-age:3.4011
SNW_HH_PRECOMPUTE: Finished Age Group:42 of 82, time-this-age:3.2533
SNW_HH_PRECOMPUTE: Finished Age Group:43 of 82, time-this-age:3.5132
SNW_HH_PRECOMPUTE: Finished Age Group:44 of 82, time-this-age:3.4771
SNW_HH_PRECOMPUTE: Finished Age Group:45 of 82, time-this-age:3.3133
SNW_HH_PRECOMPUTE: Finished Age Group:46 of 82, time-this-age:3.4673
SNW_HH_PRECOMPUTE: Finished Age Group:47 of 82, time-this-age:3.1794
SNW_HH_PRECOMPUTE: Finished Age Group:48 of 82, time-this-age:3.1958
SNW_HH_PRECOMPUTE: Finished Age Group:49 of 82, time-this-age:3.4545
SNW_HH_PRECOMPUTE: Finished Age Group:50 of 82, time-this-age:3.355
SNW_HH_PRECOMPUTE: Finished Age Group:51 of 82, time-this-age:3.2059
SNW_HH_PRECOMPUTE: Finished Age Group:52 of 82, time-this-age:3.2882
SNW_HH_PRECOMPUTE: Finished Age Group:53 of 82, time-this-age:3.3772
SNW_HH_PRECOMPUTE: Finished Age Group:54 of 82, time-this-age:3.3279
SNW_HH_PRECOMPUTE: Finished Age Group:55 of 82, time-this-age:3.5412
SNW_HH_PRECOMPUTE: Finished Age Group:56 of 82, time-this-age:3.504
SNW_HH_PRECOMPUTE: Finished Age Group:57 of 82, time-this-age:3.4961
SNW_HH_PRECOMPUTE: Finished Age Group:58 of 82, time-this-age:3.3629
SNW_HH_PRECOMPUTE: Finished Age Group:59 of 82, time-this-age:3.4105
SNW_HH_PRECOMPUTE: Finished Age Group:60 of 82, time-this-age:3.3755
SNW_HH_PRECOMPUTE: Finished Age Group:61 of 82, time-this-age:3.4102
SNW_HH_PRECOMPUTE: Finished Age Group:62 of 82, time-this-age:3.4844
SNW_HH_PRECOMPUTE: Finished Age Group:63 of 82, time-this-age:3.3864
SNW_HH_PRECOMPUTE: Finished Age Group:64 of 82, time-this-age:3.6674
SNW_HH_PRECOMPUTE: Finished Age Group:65 of 82, time-this-age:3.3549
SNW_HH_PRECOMPUTE: Finished Age Group:66 of 82, time-this-age:3.3543
SNW_HH_PRECOMPUTE: Finished Age Group:67 of 82, time-this-age:3.3775
SNW_HH_PRECOMPUTE: Finished Age Group:68 of 82, time-this-age:3.3574
SNW_HH_PRECOMPUTE: Finished Age Group:69 of 82, time-this-age:3.3706
SNW_HH_PRECOMPUTE: Finished Age Group:70 of 82, time-this-age:3.4841
SNW_HH_PRECOMPUTE: Finished Age Group:71 of 82, time-this-age:3.361
SNW_HH_PRECOMPUTE: Finished Age Group:72 of 82, time-this-age:3.3692
SNW_HH_PRECOMPUTE: Finished Age Group:73 of 82, time-this-age:3.4296
SNW_HH_PRECOMPUTE: Finished Age Group:74 of 82, time-this-age:3.4363
SNW_HH_PRECOMPUTE: Finished Age Group:75 of 82, time-this-age:3.391
SNW_HH_PRECOMPUTE: Finished Age Group:76 of 82, time-this-age:3.4927
SNW_HH_PRECOMPUTE: Finished Age Group:77 of 82, time-this-age:3.4073
SNW_HH_PRECOMPUTE: Finished Age Group:78 of 82, time-this-age:3.3416
SNW_HH_PRECOMPUTE: Finished Age Group:79 of 82, time-this-age:3.4042
```

```
SNW_HH_PRECOMPUTE: Finished Age Group:80 of 82, time-this-age:3.4844
SNW_HH_PRECOMPUTE: Finished Age Group:81 of 82, time-this-age:3.4572
SNW_HH_PRECOMPUTE: Finished Age Group:82 of 82, time-this-age:3.2411
SNW_HH_PRECOMPUTE: Finished Age Group:83 of 82, time-this-age:3.3428
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time cost=284.
```

9.3.2 Solve for 2008 Value and Consumption with 0 and 2 Checks

Now we use the **snw_v08_jaeemk** function, which takes as inputs the 2008-specific value and policy we have already found, to compute the 2008 value and consumption based on different stimulus amounts via asset-equivalent transformation given stimulus amounts.

FIrst, obtain V and C with zero stimulus.

```
% Call Function
welf_checks = 0;
[V_2008_check0, C_2008_check0] = snw_v08_jaeemk(...
    welf_checks, mp_params, mp_controls, V_2008, cons_2008, mp_precompute_res);
```

```
Solve for V_2008_check for 0 stimulus checks
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=st_biden_or_trump_undefined;welf_checks=0;TR=0.001823
Completed SNW_V08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=3e-05
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean
C_2008_check	1	1	6	4.37e+07	83	5.265e+05	2.3277e+08	5.3267
V_2008_check	2	2	6	4.37e+07	83	5.265e+05	-6.6426e+08	-15.201

xxx TABLE:C_2008_check xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	0.036218	0.036736	0.038184	0.042735	0.048545	12.256	12.541	12.835
r2	0.036271	0.036736	0.038385	0.043404	0.049852	12.491	12.778	13.072
r3	0.036717	0.037251	0.039845	0.044907	0.051515	12.744	13.032	13.327
r4	0.038144	0.038678	0.041269	0.046371	0.053128	12.989	13.277	13.573
r5	0.039534	0.040068	0.042653	0.047793	0.054687	13.224	13.513	13.809
r79	0.2016	0.20214	0.20586	0.21598	0.23568	35.82	37.367	39.414
r80	0.2016	0.20214	0.20586	0.21598	0.23568	40.755	42.955	45.289
r81	0.2016	0.20214	0.20586	0.21598	0.23568	48.912	52.041	55.022
r82	0.2016	0.20214	0.20586	0.21598	0.23568	66.719	69.201	72.373
r83	0.2016	0.20214	0.20586	0.21598	0.23568	116.83	122.65	128.67

xxx TABLE:V_2008_check xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	-295.66	-295.26	-292.66	-286.62	-277.22	-4.3615	-4.2673	-4.174
r2	-286.11	-285.71	-283.12	-277.16	-268.03	-4.2548	-4.1641	-4.074
r3	-276.49	-276.09	-273.59	-267.84	-259.11	-4.1461	-4.0589	-3.972
r4	-266.77	-266.41	-264.08	-258.7	-250.49	-4.0342	-3.9507	-3.86
r5	-257.99	-257.65	-255.48	-250.43	-242.69	-3.9287	-3.8485	-3.76
r79	-13.356	-13.343	-13.253	-13.025	-12.638	-0.22088	-0.21055	-0.2008
r80	-12.025	-12.012	-11.923	-11.695	-11.308	-0.16977	-0.1618	-0.1542
r81	-10.382	-10.369	-10.28	-10.052	-9.6651	-0.11711	-0.11162	-0.1064

```
    r82    -8.1742    -8.1611    -8.0716     -7.844     -7.457    -0.065329    -0.062239    -0.05935
    r83    -4.9602    -4.9471    -4.8576    -4.6301    -4.2431    -0.020966    -0.019971    -0.01903
```

```
% Call Function
```

Second, obtain V and C with two stimulus checks.

```
welf_checks = 2;
[V_2008_check2, C_2008_check2] = snw_v08_jaeemk(...
    welf_checks, mp_params, mp_controls, V_2008, cons_2008, mp_precompute_res);
```

```
Solve for V_2008_check for 2 stimulus checks
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=st_biden_or_trump_undefined;welf_checks=2;TR=0.001823
Completed SNW_V08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=2.16e-05
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                    i    idx    ndim     numel      rowN      colN           sum         mean
                    -    ---    ----    --------    ----    ---------    -----------    -------    -

    C_2008_check    1     1      6      4.37e+07     83     5.265e+05      2.328e+08     5.3273
    V_2008_check    2     2      6      4.37e+07     83     5.265e+05    -6.6365e+08    -15.187    2

xxx TABLE:C_2008_check xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.037941    0.038148    0.039819    0.043807    0.049244    12.256     12.541     12.835
    r2     0.038108    0.038344    0.040188    0.044594    0.050571    12.492     12.778     13.073
    r3      0.03941    0.039781    0.041664    0.046126    0.052244    12.745     13.032     13.327
    r4     0.040834    0.041205    0.043102    0.047618    0.053867    12.989     13.278     13.573
    r5      0.04222    0.042589      0.0445    0.049065    0.055435    13.224     13.513     13.809
    r79     0.20525     0.20579     0.20951     0.21963     0.23776    35.821     37.368     39.415
    r80     0.20525     0.20579     0.20951     0.21963     0.23776    40.756     42.957      45.29
    r81     0.20525     0.20579     0.20951     0.21963      0.2378    48.914     52.043     55.024
    r82     0.20525     0.20579     0.20951     0.21963     0.23814     66.72     69.203     72.375
    r83     0.20525     0.20579     0.20951     0.21963     0.23933    116.84     122.66     128.68

xxx TABLE:V_2008_check xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    ---------

    r1     -293.09    -292.72    -290.49    -284.88    -275.86      -4.3615      -4.2672      -4.174
    r2     -283.55    -283.18    -280.98    -275.47    -266.73      -4.2548       -4.164      -4.074
    r3     -274.01    -273.65    -271.52    -266.23    -257.88       -4.146      -4.0589      -3.972
    r4     -264.47    -264.13    -262.14    -257.19    -249.33      -4.0341      -3.9506      -3.867
    r5     -255.84    -255.53    -253.66       -249    -241.59      -3.9286      -3.8484       -3.76
    r79    -13.268    -13.255    -13.171    -12.954    -12.578     -0.22088     -0.21054     -0.2008
    r80    -11.937    -11.924    -11.841    -11.623    -11.248     -0.16976     -0.16179     -0.1542
    r81    -10.294    -10.281    -10.198    -9.9804    -9.6057     -0.11711     -0.11162     -0.1064
    r82    -8.0862    -8.0735    -7.9895    -7.7724    -7.3981    -0.065327    -0.062237    -0.05935
    r83    -4.8723    -4.8595    -4.7755    -4.5584    -4.1854    -0.020965     -0.01997    -0.01903
```

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check = V_2008_check2 - V_2008_check0;
mn_MPC_U_gain_share_check = (C_2008_check2 - C_2008_check0)./(welf_checks*mp_params('TR'));
```

9.3.3 Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

9.3.4 Analyze Difference in V and C with Check

The difference between V and V with Check, marginal utility gain given the check.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

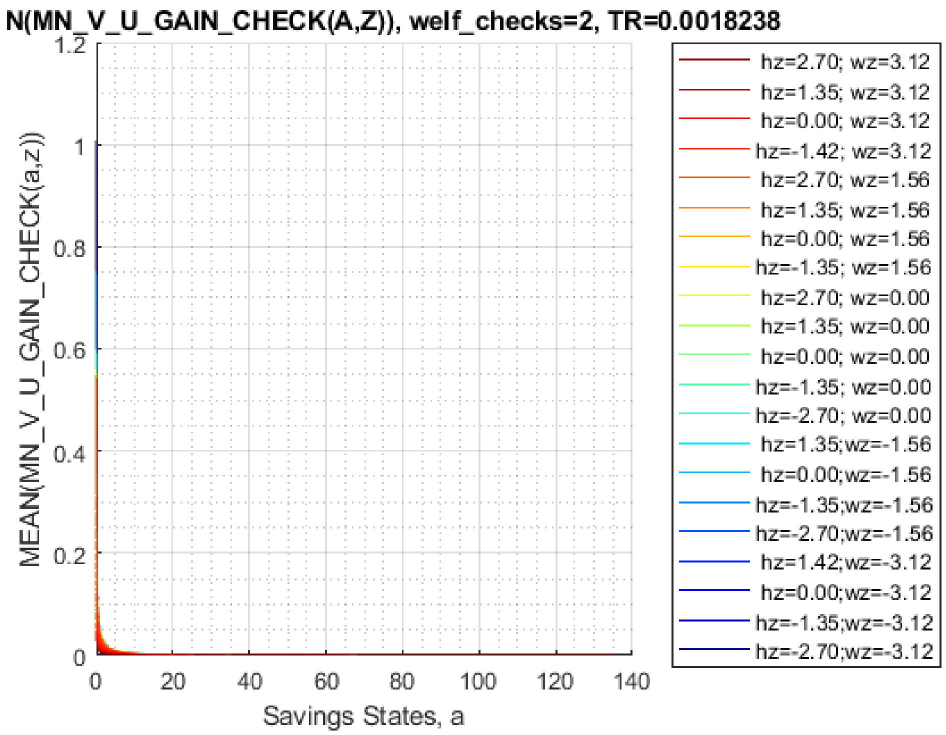
MEAN(MN_V_GAIN_CHECK(A,Z))

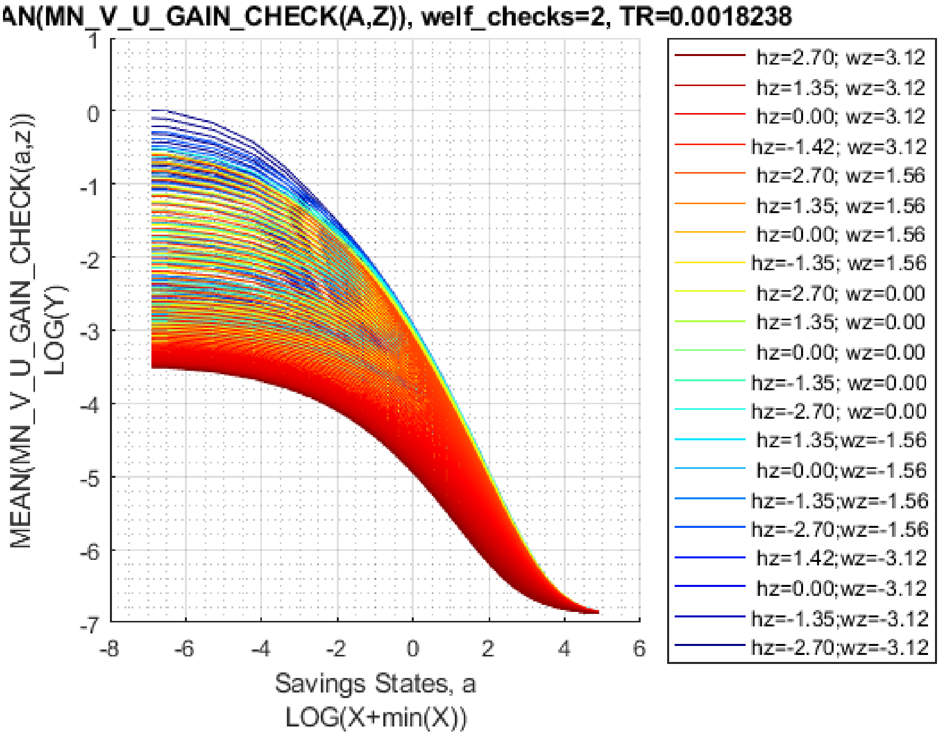
Tabulate value and policies along savings and shocks:

```
% Set
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_par
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0        1.0061       0.90365        0.8116       0.72825       0.65329
```

```
group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
-----    ----------    ----------    ----------    ----------    ----------    ----------    ---

  1              0      0.96418       0.92776       0.90436       0.88966       0.87994       0.
```

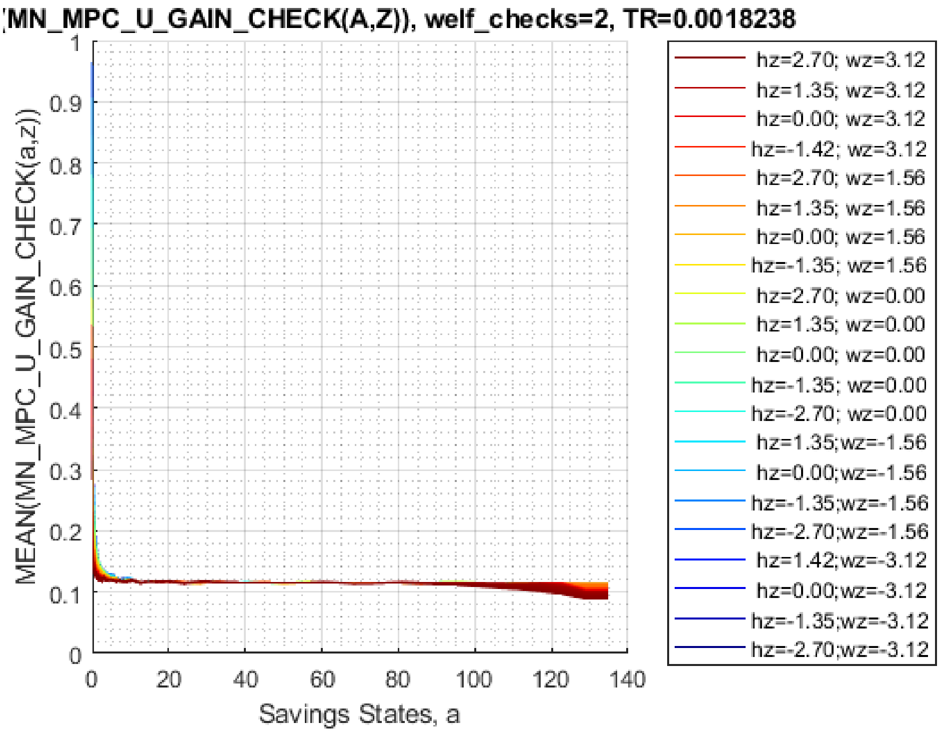
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

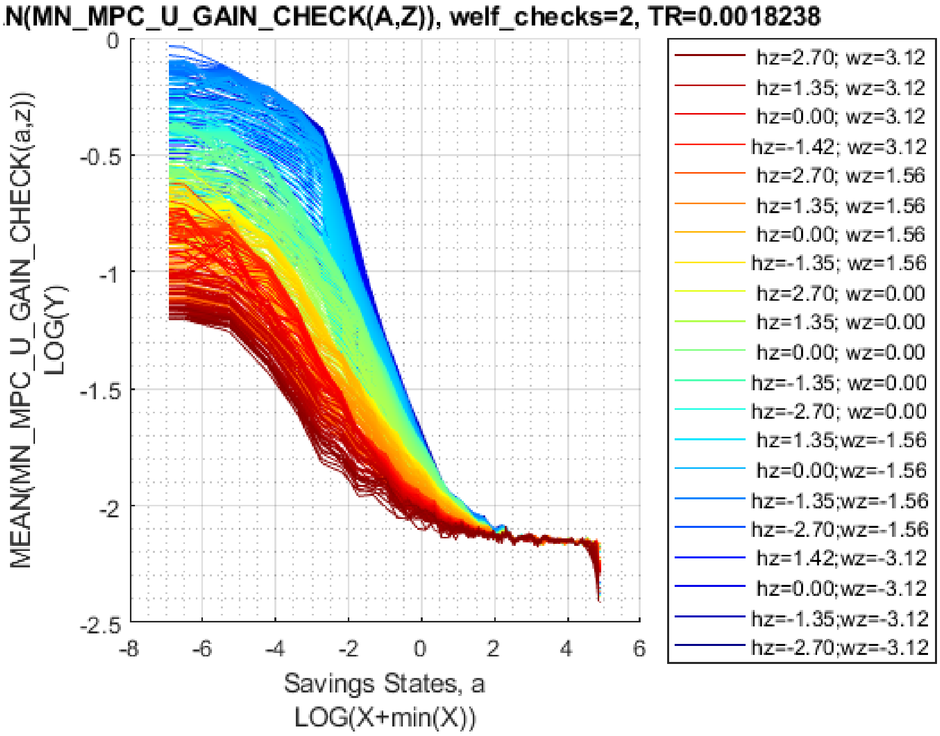




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





9.3.5 Analyze Marginal Value and MPC over Y(a,eta), Conditional On Kids, Marry, Age, Education

Income is generated by savings and shocks, what are the income levels generated by all the shock and savings points conditional on kids, marital status, age and educational levels. Plot on the Y axis MPC, and plot on the X axis income levels, use colors to first distinguish between different a levels, then use colors to distinguish between different eta levles.

Set Up date, Select Age, unmarried, no kids, lower education:

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
% 38 year old, unmarried, no kids, lower educated
% Only Household Head Shock Matters so select up to 'n_eta_H_grid'
mn_total_inc_jemk = total_inc_VFI(19,:,1:mp_params('n_eta_H_grid'),1,1,1);
mn_V_W_gain_check_use = V_2008_check2 - V_2008_check0;
mn_C_W_gain_check_use = C_2008_check2 - C_2008_check0;
```

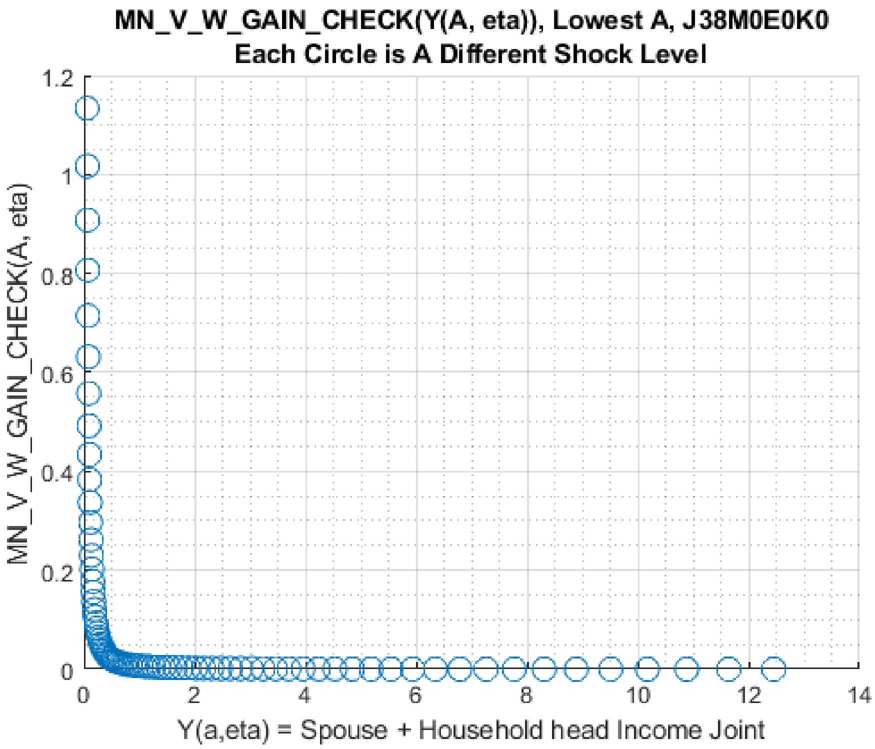
Select Age, Education, Marital, Kids Count:s

```
% Selections
it_age = 21; % +18
it_marital = 1; % 1 = unmarried
it_kids = 1; % 1 = kids is zero
it_educ = 1; % 1 = lower education
% Select: NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
mn_C_W_gain_check_jemk = mn_C_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ, it_m
mn_V_W_gain_check_jemk = mn_V_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ, it_m
% Reshape, so shock is the first dim, a is the second
mt_total_inc_jemk = permute(mn_total_inc_jemk,[3,2,1]);
mt_C_W_gain_check_jemk = permute(mn_C_W_gain_check_jemk,[3,2,1]);
mt_C_W_gain_check_jemk(mt_C_W_gain_check_jemk<=1e-10) = 1e-10;
mt_V_W_gain_check_jemk = permute(mn_V_W_gain_check_jemk,[3,2,1]);
mt_V_W_gain_check_jemk(mt_V_W_gain_check_jemk<=1e-10) = 1e-10;
% Generate meshed a and shock grid
[mt_eta_H, mt_a] = ndgrid(eta_H_grid(1:mp_params('n_eta_H_grid')), agrid);
```

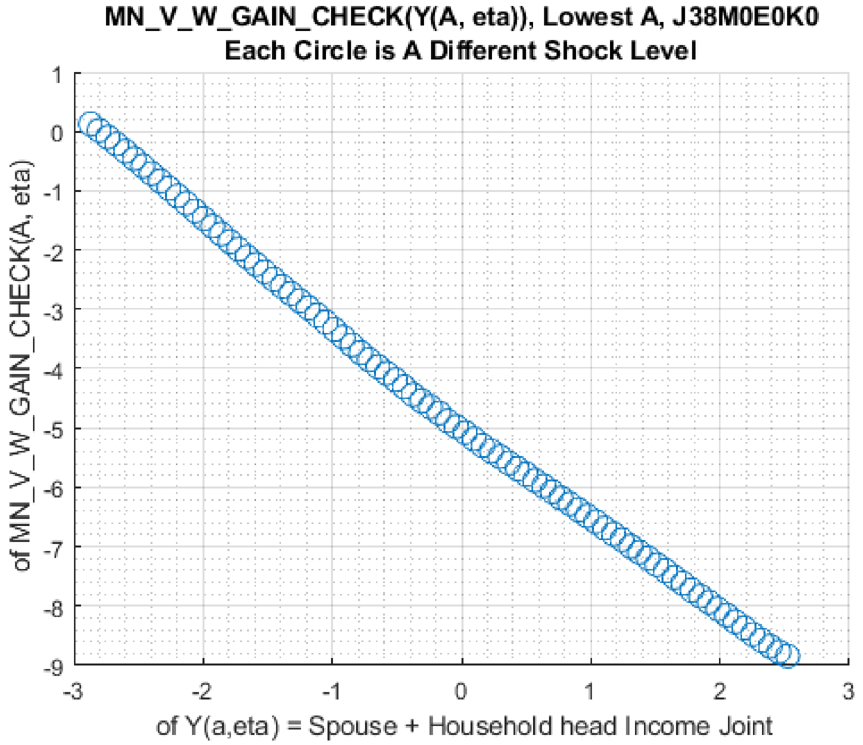
9.3.6 Marginal Value Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
it_a = 1;
scatter((mt_total_inc_jemk(:,it_a)), (mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

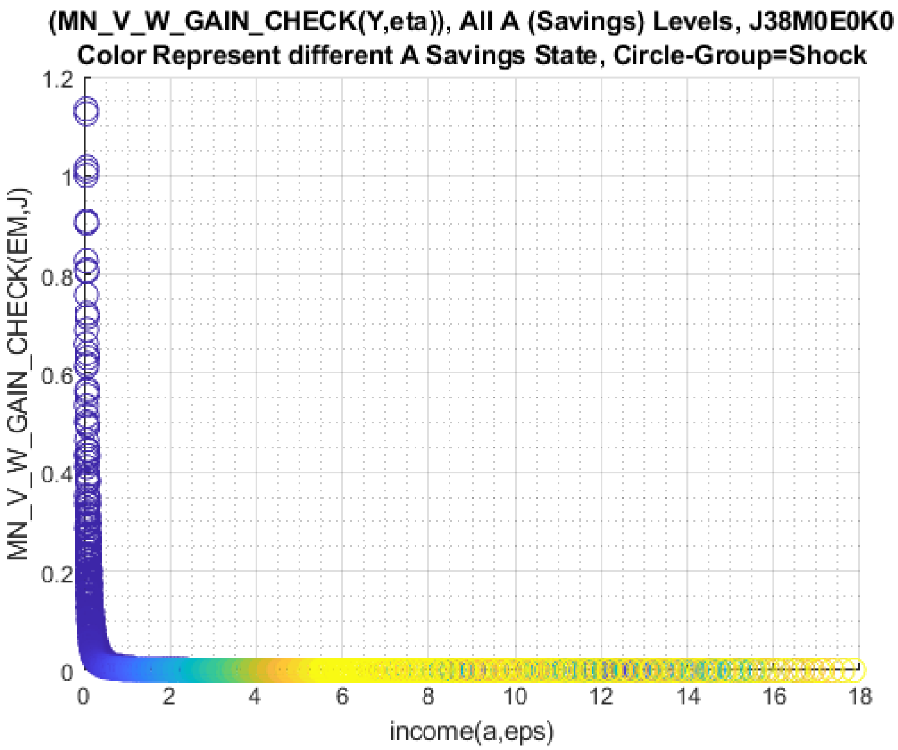



```
figure();
it_shock = 1;
scatter(log(mt_total_inc_jemk(:,it_a)), log(mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel(' of Y(a,eta) = Spouse + Household head Income Joint');
ylabel(' of MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

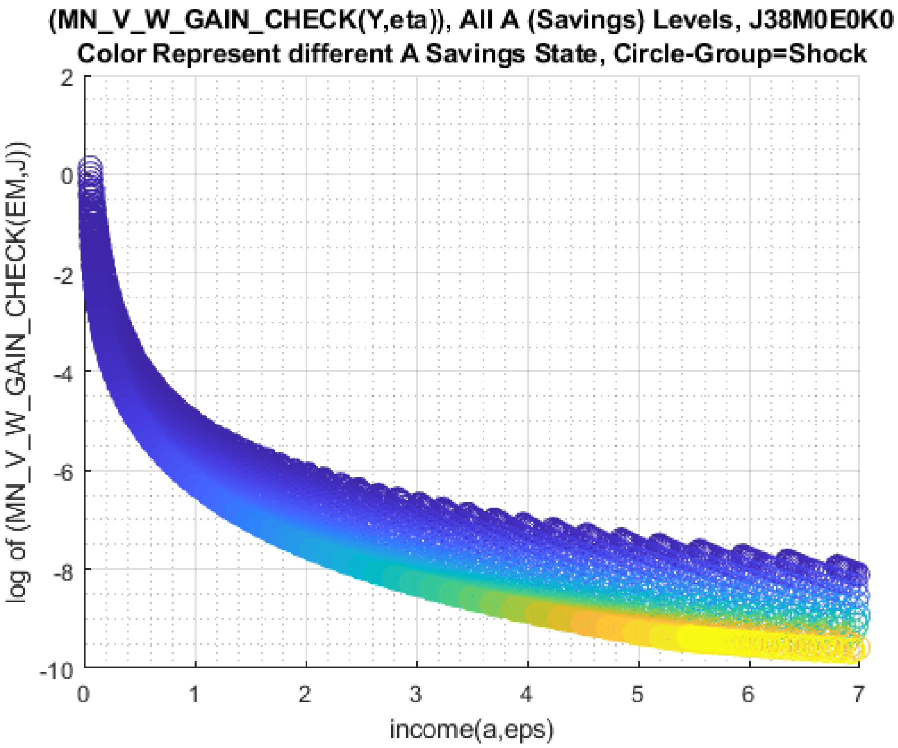


Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_V\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```



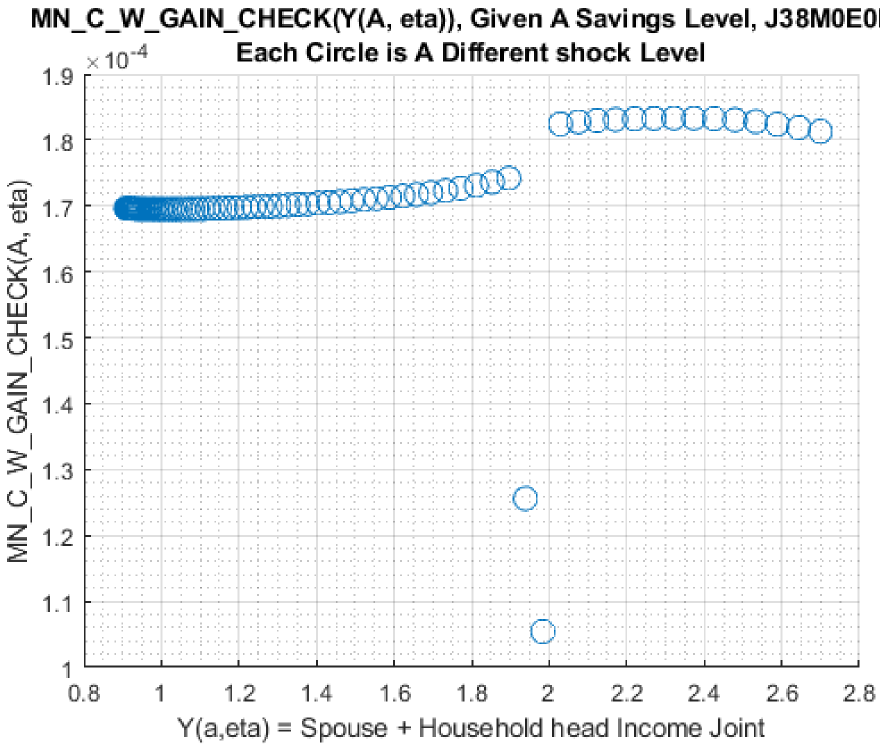
```
figure();
scatter((mt_total_inc_jemk(:)), log(mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
xlim([0,7]);
grid on;
grid minor;
```



9.3.7 Marginal Consumption Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

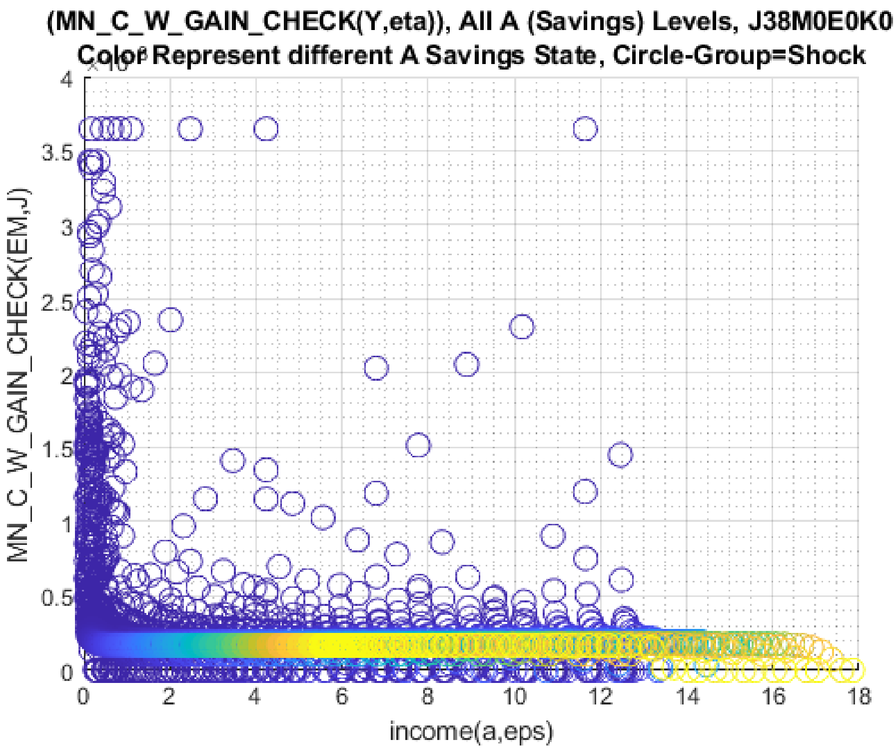
How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
it_a = 50;
scatter(log(mt_total_inc_jemk(:,it_a)), mt_C_W_gain_check_jemk(:,it_a), 100);
title({'MN\_C\_W\_GAIN\_CHECK(Y(A, eta)), Given A Savings Level, J38M0E0K0', ...
    'Each Circle is A Different shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_C\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

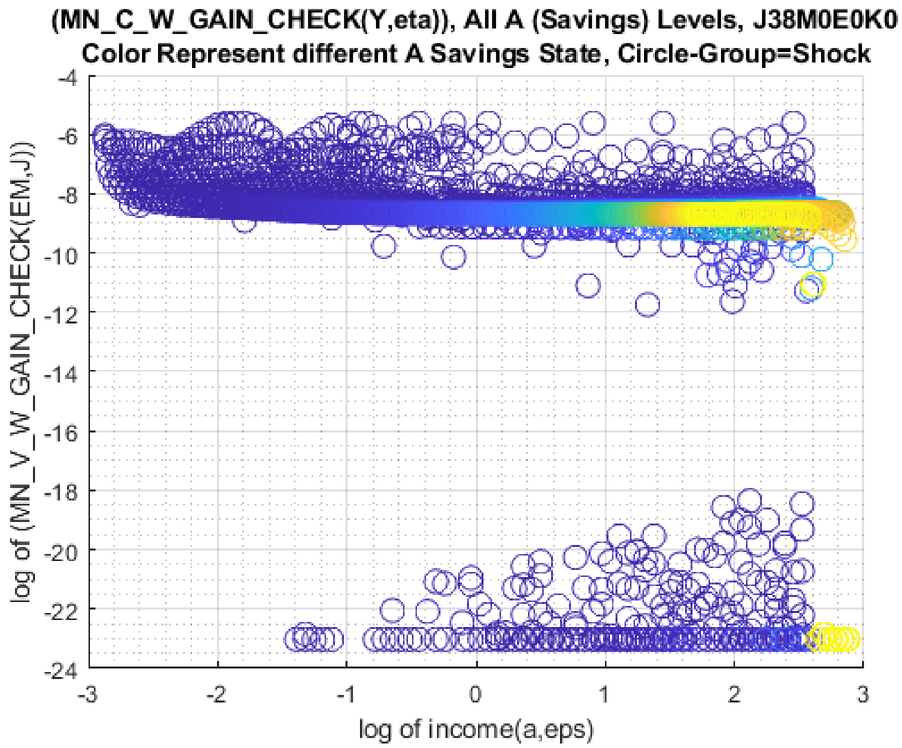


Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_C_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_C\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```



```
figure();
scatter(log(mt_total_inc_jemk(:)), log(mt_C_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('log of income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
grid on;
grid minor;
```



9.3.8 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

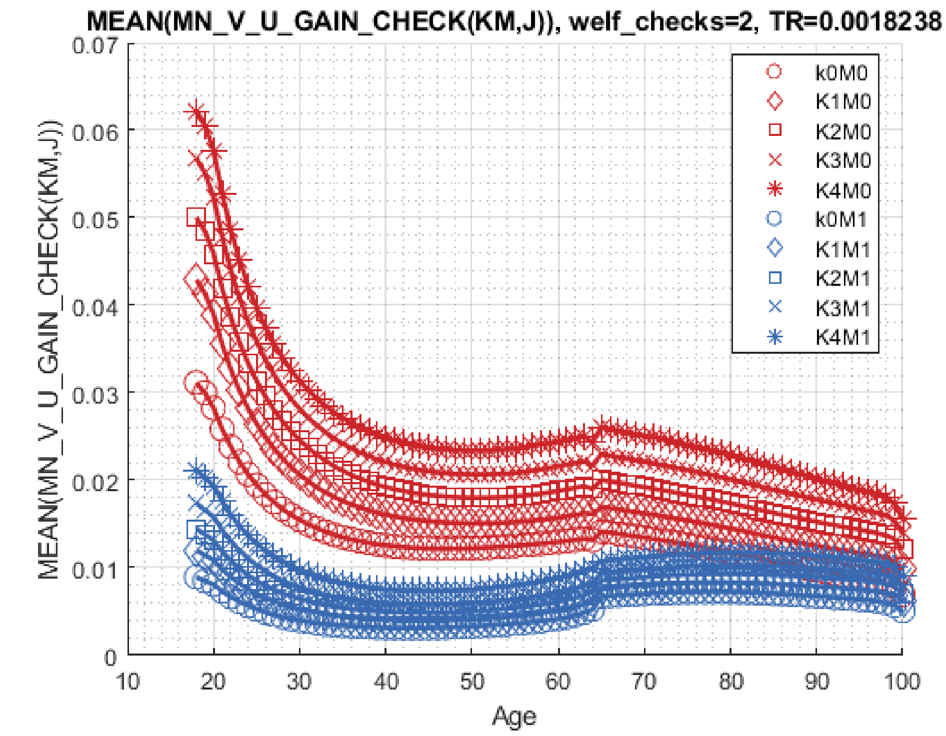
Tabulate value and policies:

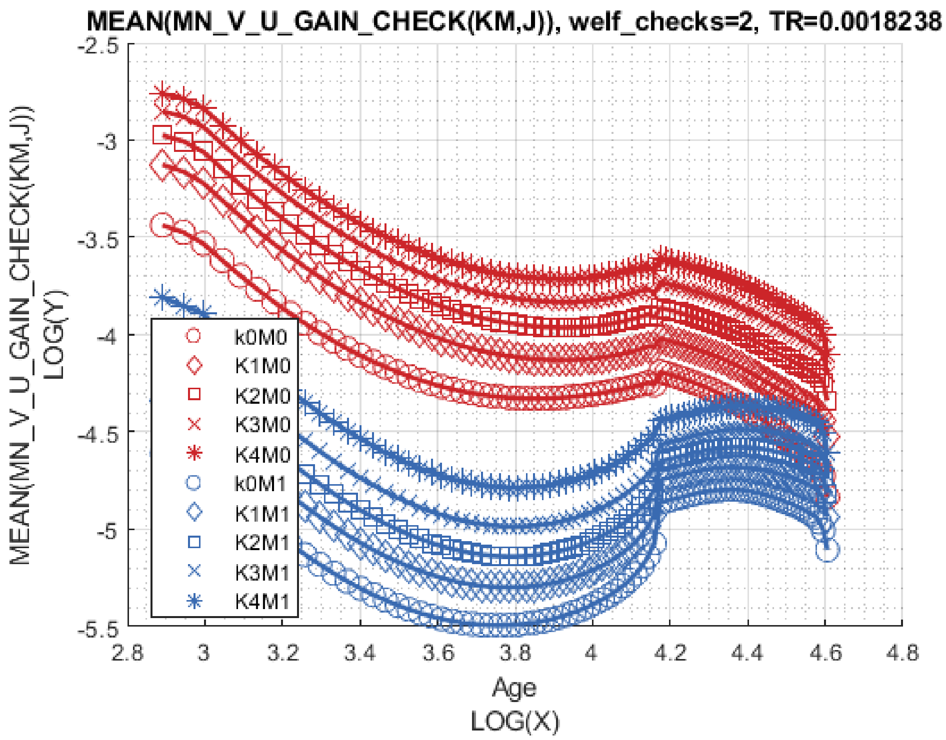
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
```

```
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0          0.03116       0.030004       0.028268       0.025846       0.023825
      2        2       0         0.042925       0.041372       0.038951       0.035539       0.032682
      3        3       0         0.050042       0.048477       0.045918       0.041935       0.038603
      4        4       0         0.056814       0.055153        0.05234       0.047822       0.044043
      5        5       0          0.06224       0.060592       0.057683       0.052755       0.048636
      6        1       1        0.0089468      0.0085141      0.0080936      0.0073219       0.006674
      7        2       1         0.012008       0.011426        0.01086      0.0098202      0.0089462
      8        3       1         0.014485        0.01381       0.013144       0.011882       0.010826
      9        4       1         0.017392       0.016611       0.015824       0.014325       0.013061
     10        5       1         0.021156       0.020287       0.019385       0.017573       0.016058

% Consumption Function
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd

xxx  MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0        0.071209       0.076265       0.082919       0.080484       0.079666
      2        2       0         0.08025       0.085444       0.092048       0.090298       0.089234
      3        3       0        0.087972       0.095508        0.10343        0.10119       0.099548
      4        4       0        0.092255       0.099859        0.10923        0.10651        0.10388
      5        5       0         0.09665         0.1041        0.11456        0.11152        0.10814
      6        1       1           0.101        0.10439        0.10978        0.10855        0.10822
      7        2       1         0.10297        0.10717        0.11242        0.11166        0.10915
      8        3       1         0.10827        0.11355        0.11922        0.11713        0.11645
      9        4       1         0.10932        0.11394        0.12031        0.11884        0.11867
     10        5       1         0.11555        0.12093         0.1289        0.12462        0.12328
```

Graph Mean Values:

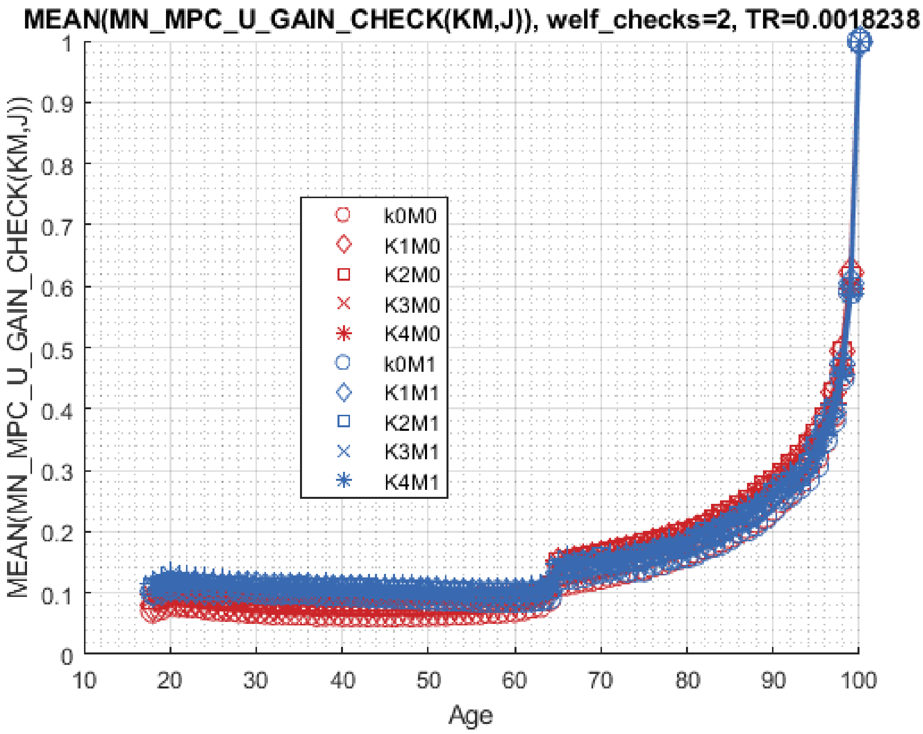
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

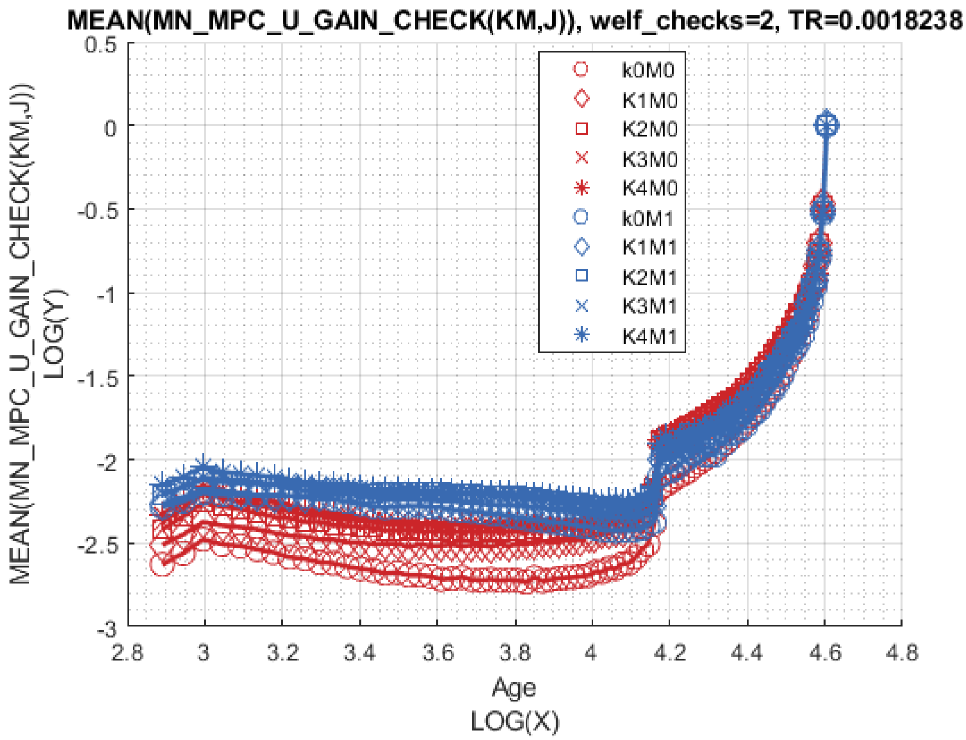




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





9.3.9 Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

```
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

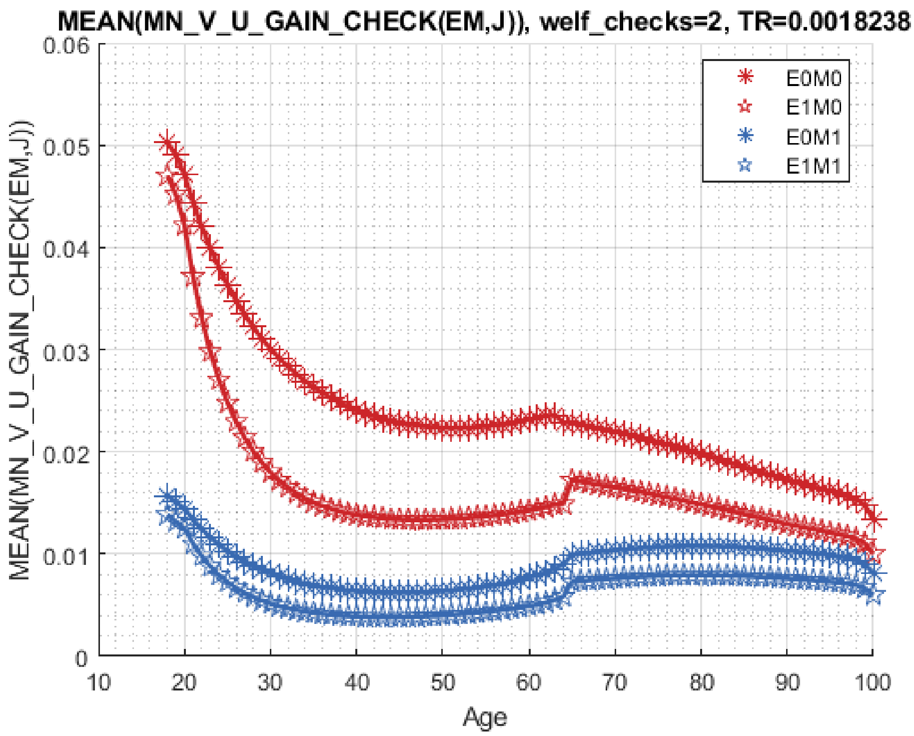
MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

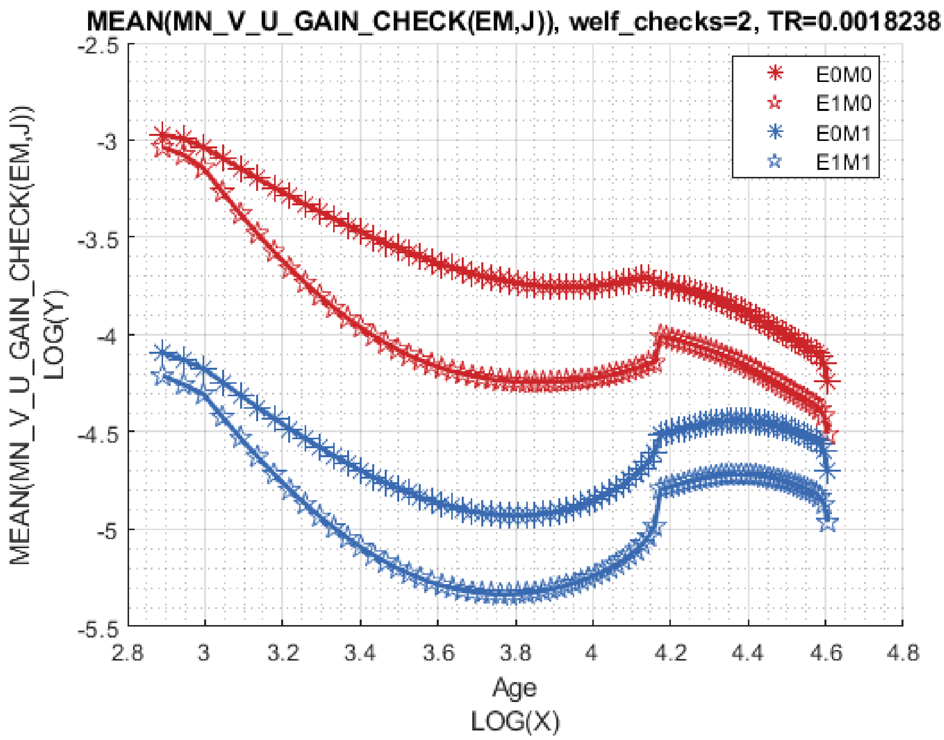
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0       0.050249       0.049057        0.04709       0.044412       0.042032
      2       1       0       0.047023       0.045182       0.042174       0.037147       0.033084
      3       0       1       0.015742       0.015072       0.014412        0.01336       0.012437
      4       1       1       0.013853       0.013188       0.012511       0.011009      0.0097884

% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd

xxx  MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0        0.07766       0.081478       0.084248       0.083711       0.083393
      2       1       0       0.093675        0.10299        0.11663        0.11229        0.10879
      3       0       1       0.099571        0.10294        0.10634        0.10623        0.10602
      4       1       1        0.11528        0.12105        0.12991        0.12609        0.12429
```

Graph Mean Values:

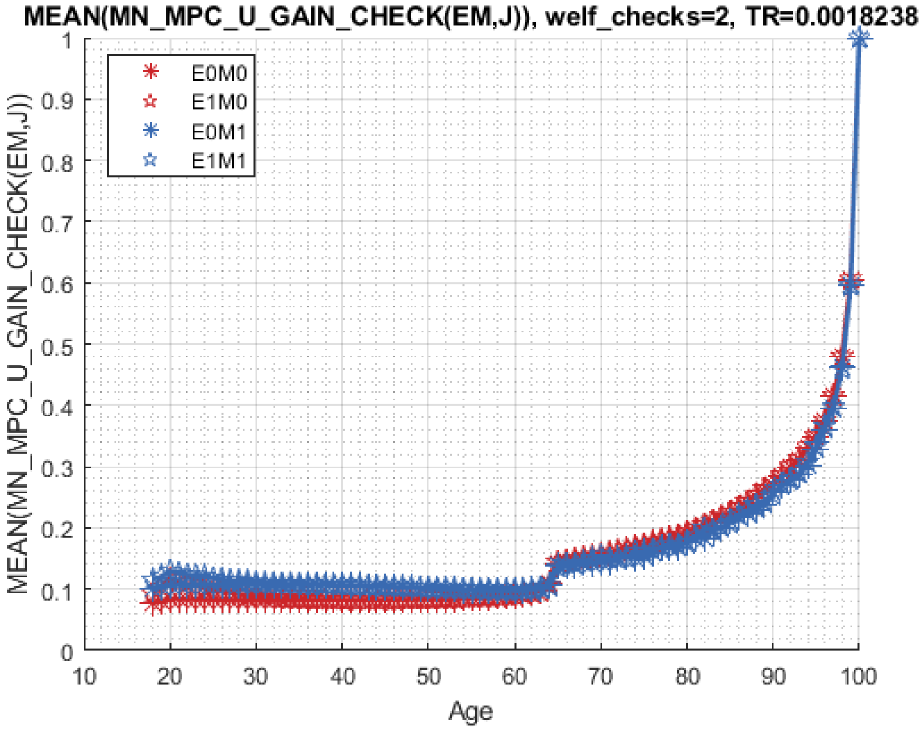
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

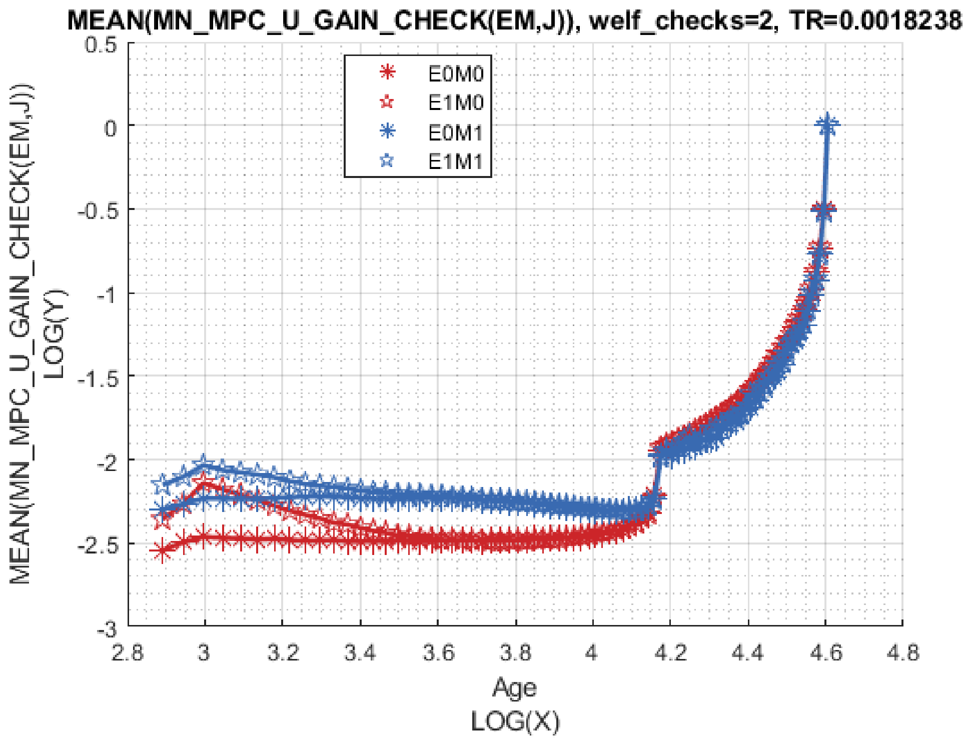




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





9.4 2007 (Bush 2008 Stimulus) Full States EV and EC of Two Checks

This is the example vignette for function: **snw_evuvw19_jaeemk_foc** from the **PrjOptiSNW Package.** 2008 integrated over VU and VW, given optimal savings choices, unemployment shocks and various expectations.

Given 2008 policy and value functions (given expectation of 2009 crisis unemployment shocks), call SNW_V08_JAEEMK to solve for value and consumption given stimulus checks. And then integrate over 08 JAEEMK states given 07 JAEEMK states (age, endogenous savings, education, income shock, marital status, kids count). The stimulus will be provisioned based on 07 JAEEMK states. Note that **snw_evuvw19_jaeemk** does not solve the 07/08 problem.

Despite the name, this function supports solving the 2019 looking into 2020 as well as the 2007 looking into 2008 problems. The idea is that the planner only has information from 2019 and from 2007, and must allocate using those information. Stimulus, however, is given in 2020 and in 2008. So the planner needs to consider expected values in consumption or welfare given the transition probabilities of states in 2007 to 2008 and in 2019 to 2020. The **snw_evuvw19_jmky** file then aggregates the full state-space results to just JMKY state-space, which is the extend of information available to the planner.

9.4.1 Test SNW_EVUVW19_JAEEMK Defaults for 2019

Call the function with defaults. First, set up some parameters.

```
clear all;
% Solution types
st_biden_or_trump = 'bushchck';

% Solve the VFI Problem and get Value Function
mp_controls = snw_mp_control('default_test');

% Solve for Unemployment Values
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;
mp_controls('bl_print_v08p08_jaeemk') = false;
mp_controls('bl_print_v08p08_jaeemk_verbose') = false;
mp_controls('bl_print_v08_jaeemk') = true;
mp_controls('bl_print_v08_jaeemk_verbose') = false;
```

Second, run initializing functions.

```
% 1. generate MP_PARAMS specific to 2008 stimulus
% Use non-default values for Bush Stimulus
mp_more_inputs = containers.Map('KeyType','char', 'ValueType','any');
mp_more_inputs('fl_ss_non_college') = 0.225;
mp_more_inputs('fl_ss_college') = 0.271;
fl_p50_hh_income_07 = 54831;
mp_more_inputs('fl_scaleconvertor') = fl_p50_hh_income_07;
% st_param_group = 'default_small';
st_param_group = 'default_dense';
st_param_group = 'default_docdense';
mp_params = snw_mp_param(st_param_group, false, 'tauchen', false, 8, 8, mp_more_inputs);
mp_params('st_biden_or_trump') = st_biden_or_trump;
mp_params('beta') = 0.95;
% 2. Solve value steady state (2009 employed)
[V_ss, ap_ss, cons_ss, mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=495.
```

```
V_emp_2009 = V_ss;
inc_ss = mp_valpol_more_ss('inc_VFI');
spouse_inc_ss = mp_valpol_more_ss('spouse_inc_VFI');
```

```
total_inc_ss = inc_ss + spouse_inc_ss;
% 3. Solve value unemployed 2009
mp_params('xi') = 0.532;
mp_params('b') = 0.37992;
mp_params('a2_covidyr') = mp_params('a2_greatrecession_2009');
mp_params('TR') = 100/fl_p50_hh_income_07;
[V_unemp_2009] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
```

```
% 4. Value and Optimal choice in 2008
[V_2008, ap_2008, cons_2008, ev_empshk_2009] = ...
    snw_v08p08_jaeemk(mp_params, mp_controls, V_emp_2009, V_unemp_2009);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
Completed SNW_V08P08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=497.3876
```

```
% 5. matrixes to pre-compute
% Only using the SNW_A4CHK_WRK_BISEC_VEC function, no unemployment
% related matrixes needed Also don't need REF_EARN_WAGEIND_GRID,
% become unemployment not conditional on wage in 2009.
cl_st_precompute_list = {'a', ...
    'inc', 'inc_unemp', 'spouse_inc',...
    'ar_z_ctr_amz'};
% Shared: Steady-State distribution
[Phi_true] = snw_ds_main(mp_params, mp_controls, ap_ss, cons_ss, mp_valpol_more_ss);
```

```
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1775.4241
```

```
% Shared: precompute, get Matrixes
% note, the mp_params inputs are based on unemployed in 2020 (MIT) or unemployed in 2009 (Expected)
% note, however, for the 2008/9 problem, only will use inc, inc_unemp, spouse_inc
mp_controls('bl_print_precompute_verbose') = false;
[mp_precompute_res] = snw_hh_precompute(mp_params, mp_controls, cl_st_precompute_list, ap_ss, Phi_tr
```

```
Wage quintile cutoffs=0.4645     0.71528      1.0335      1.5632
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time cost=254.
```

9.4.2 Solve for 2019 Evuvw With 0 and 2 Checks

Solve for 0 and 2 checks, by finding the increase to asset state-space that is equivalent to the check increase, so that the problem can be solved without increasing the state-space.

```
% Call Function
welf_checks = 0;
[ev07_jaeemk_check0, ec07_jaeemk_check0, ev08_jaeemk_check0, ec08_jaeemk_check0] = snw_evuvw19_jaeem
    welf_checks, mp_params, mp_controls, ...
    ap_ss, V_2008, cons_2008, mp_precompute_res);
```

```
Solve for V_2008_check for 0 stimulus checks
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bushchck;welf_checks=0;TR=0.0018238;SNW_MP_PARAM=defa
Completed SNW_V08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=6.99e-05
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bushchck;SNW_MP_PARAM=default_docdense;SNW_MP_CON
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                   i    idx    ndim      numel       rowN      colN            sum          mean
```

	-	---	----	----------	----	---------	-----------	-------
ec07_jaeemk	1	1	6	4.3173e+07	82	5.265e+05	1.9685e+08	4.5597
ec08_jaeemk	2	2	6	4.37e+07	83	5.265e+05	2.3277e+08	5.3267
ev07_jaeemk	3	3	6	4.3173e+07	82	5.265e+05	-6.4618e+08	-14.967
ev08_jaeemk	4	4	6	4.37e+07	83	5.265e+05	-6.6426e+08	-15.201

xxx TABLE:ec07_jaeemk xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	0.039543	0.039543	0.039978	0.042271	0.04854	12.009	12.281	12.561
r2	0.039889	0.039889	0.040323	0.043305	0.049735	12.251	12.524	12.806
r3	0.041432	0.041432	0.041608	0.043991	0.050734	12.485	12.759	13.042
r4	0.042935	0.042935	0.043023	0.045459	0.052199	12.742	13.017	13.3
r5	0.044395	0.044395	0.044399	0.046895	0.053615	12.99	13.266	13.548
r78	0.2016	0.2016	0.2016	0.2016	0.20214	27.775	28.774	29.785
r79	0.2016	0.2016	0.2016	0.2016	0.2016	30.43	31.663	32.736
r80	0.2016	0.2016	0.2016	0.2016	0.2016	33.68	35.501	37.368
r81	0.2016	0.2016	0.2016	0.2016	0.2016	40.118	41.397	43.175
r82	0.2016	0.2016	0.2016	0.2016	0.2016	52.1	55.541	58.464

xxx TABLE:ec08_jaeemk xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	0.036218	0.036736	0.038184	0.042735	0.048545	12.256	12.541	12.835
r2	0.036271	0.036736	0.038385	0.043404	0.049852	12.491	12.778	13.072
r3	0.036717	0.037251	0.039845	0.044907	0.051515	12.744	13.032	13.327
r4	0.038144	0.038678	0.041269	0.046371	0.053128	12.989	13.277	13.573
r5	0.039534	0.040068	0.042653	0.047793	0.054687	13.224	13.513	13.809
r79	0.2016	0.20214	0.20586	0.21598	0.23568	35.82	37.367	39.414
r80	0.2016	0.20214	0.20586	0.21598	0.23568	40.755	42.955	45.289
r81	0.2016	0.20214	0.20586	0.21598	0.23568	48.912	52.041	55.022
r82	0.2016	0.20214	0.20586	0.21598	0.23568	66.719	69.201	72.373
r83	0.2016	0.20214	0.20586	0.21598	0.23568	116.83	122.65	128.67

xxx TABLE:ev07_jaeemk xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	-282.59	-282.59	-282.23	-278.38	-270.8	-4.3833	-4.2888	-4.195
r2	-272.49	-272.49	-272.13	-268.93	-261.76	-4.2754	-4.1844	-4.094
r3	-262.36	-262.36	-262.23	-259.89	-253.05	-4.1651	-4.0777	-3.991
r4	-253.22	-253.22	-253.16	-250.92	-244.6	-4.0512	-3.9675	-3.884
r5	-244.95	-244.95	-244.95	-242.81	-236.93	-3.9436	-3.8633	-3.783
r78	-13.362	-13.362	-13.362	-13.362	-13.349	-0.27313	-0.26104	-0.2497
r79	-12.032	-12.032	-12.032	-12.032	-12.032	-0.21855	-0.20781	-0.1987
r80	-10.388	-10.388	-10.388	-10.388	-10.388	-0.16126	-0.15407	-0.1473
r81	-8.1801	-8.1801	-8.1801	-8.1801	-8.1801	-0.10114	-0.097396	-0.09343
r82	-4.9651	-4.9651	-4.9651	-4.9651	-4.9651	-0.044201	-0.041462	-0.03941

xxx TABLE:ev08_jaeemk xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	-295.66	-295.26	-292.66	-286.62	-277.22	-4.3615	-4.2673	-4.174
r2	-286.11	-285.71	-283.12	-277.16	-268.03	-4.2548	-4.1641	-4.074

```
    r3     -276.49    -276.09    -273.59    -267.84    -259.11      -4.1461      -4.0589      -3.972
    r4     -266.77    -266.41    -264.08     -258.7    -250.49      -4.0342      -3.9507       -3.86
    r5     -257.99    -257.65    -255.48    -250.43    -242.69      -3.9287      -3.8485       -3.76
    r79    -13.356    -13.343    -13.253    -13.025    -12.638     -0.22088     -0.21055     -0.2008
    r80    -12.025    -12.012    -11.923    -11.695    -11.308     -0.16977      -0.1618     -0.1542
    r81    -10.382    -10.369     -10.28    -10.052    -9.6651     -0.11711     -0.11162     -0.1064
    r82    -8.1742    -8.1611    -8.0716     -7.844     -7.457    -0.065329    -0.062239    -0.05935
    r83    -4.9602    -4.9471    -4.8576    -4.6301    -4.2431    -0.020966    -0.019971    -0.01903

% Call Function
welf_checks = 2;
[ev07_jaeemk_check2, ec07_jaeemk_check2, ev08_jaeemk_check2, ec08_jaeemk_check2] = snw_evuvw19_jaeem
    welf_checks, mp_params, mp_controls, ...
    ap_ss, V_2008, cons_2008, mp_precompute_res);

Solve for V_2008_check for 2 stimulus checks
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bushchck;welf_checks=2;TR=0.0018238;SNW_MP_PARAM=defa
Completed SNW_V08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=2.82e-05
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bushchck;SNW_MP_PARAM=default_docdense;SNW_MP_CON
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                   i    idx    ndim      numel       rowN      colN          sum         mean
                   -    ---    ----    ----------    ----    ---------    -----------    -------

    ec07_jaeemk    1     1      6      4.3173e+07     82     5.265e+05     1.9688e+08     4.5603
    ec08_jaeemk    2     2      6        4.37e+07     83     5.265e+05      2.328e+08     5.3273
    ev07_jaeemk    3     3      6      4.3173e+07     82     5.265e+05    -6.4561e+08    -14.954
    ev08_jaeemk    4     4      6        4.37e+07     83     5.265e+05    -6.6365e+08    -15.187

xxx TABLE:ec07_jaeemk xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.040827    0.040827    0.040972     0.04389    0.049551    12.009     12.281     12.561
    r2     0.042096    0.042096      0.0424    0.045024    0.050802    12.251     12.524     12.806
    r3     0.043624    0.043624    0.043746    0.045821    0.051869    12.485      12.76     13.042
    r4      0.04511     0.04511     0.04517    0.047304    0.053361    12.742     13.017       13.3
    r5      0.04655     0.04655    0.046553    0.048751    0.054802     12.99     13.266     13.549
    r78     0.20525     0.20525     0.20525     0.20525     0.20579    27.775     28.775     29.786
    r79     0.20525     0.20525     0.20525     0.20525     0.20525    30.431     31.664     32.737
    r80     0.20525     0.20525     0.20525     0.20525     0.20525    33.681     35.503     37.369
    r81     0.20525     0.20525     0.20525     0.20525     0.20525    40.119       41.4     43.178
    r82     0.20525     0.20525     0.20525     0.20525     0.20525    52.103     55.545     58.468

xxx TABLE:ec08_jaeemk xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.037941    0.038148    0.039819    0.043807    0.049244    12.256     12.541     12.835
    r2     0.038108    0.038344    0.040188    0.044594    0.050571    12.492     12.778     13.073
    r3      0.03941    0.039781    0.041664    0.046126    0.052244    12.745     13.032     13.327
    r4     0.040834    0.041205    0.043102    0.047618    0.053867    12.989     13.278     13.573
    r5      0.04222    0.042589      0.0445    0.049065    0.055435    13.224     13.513     13.809
    r79     0.20525     0.20579     0.20951     0.21963     0.23776    35.821     37.368     39.415
    r80     0.20525     0.20579     0.20951     0.21963     0.23776    40.756     42.957      45.29
    r81     0.20525     0.20579     0.20951     0.21963      0.2378    48.914     52.043     55.024
```

```
    r82     0.20525     0.20579     0.20951     0.21963     0.23814     66.72     69.203     72.375
    r83     0.20525     0.20579     0.20951     0.21963     0.23933    116.84     122.66     128.68

xxx TABLE:ev07_jaeemk xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5         c526496       c526497       c526498
            -------     -------     -------     -------     -------     ---------     ---------     ---------

    r1      -280.23     -280.23     -279.88      -276.5     -269.35       -4.3833       -4.2887        -4.195
    r2      -270.21     -270.21     -269.89     -267.08     -260.35       -4.2753       -4.1844        -4.094
    r3      -260.26     -260.26     -260.13      -258.1     -251.71        -4.165       -4.0777        -3.991
    r4      -251.26     -251.26      -251.2     -249.25     -243.33       -4.0511       -3.9674        -3.884
    r5      -243.12     -243.12     -243.12     -241.24     -235.73       -3.9436       -3.8633        -3.783
    r78     -13.274     -13.274     -13.274     -13.274     -13.262      -0.27312      -0.26103        -0.249
    r79     -11.944     -11.944     -11.944     -11.944     -11.944      -0.21854       -0.2078       -0.1987
    r80       -10.3       -10.3       -10.3       -10.3       -10.3      -0.16125      -0.15406       -0.1473
    r81     -8.0921     -8.0921     -8.0921     -8.0921     -8.0921      -0.10113     -0.097391      -0.09343
    r82     -4.8771     -4.8771     -4.8771     -4.8771     -4.8771     -0.044198      -0.04146       -0.0394

xxx TABLE:ev08_jaeemk xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5         c526496       c526497       c526498
            -------     -------     -------     -------     -------     ---------     ---------     ---------

    r1      -293.09     -292.72     -290.49     -284.88     -275.86       -4.3615       -4.2672        -4.174
    r2      -283.55     -283.18     -280.98     -275.47     -266.73       -4.2548        -4.164        -4.074
    r3      -274.01     -273.65     -271.52     -266.23     -257.88        -4.146       -4.0589        -3.972
    r4      -264.47     -264.13     -262.14     -257.19     -249.33       -4.0341       -3.9506        -3.867
    r5      -255.84     -255.53     -253.66        -249     -241.59       -3.9286       -3.8484         -3.76
    r79     -13.268     -13.255     -13.171     -12.954     -12.578      -0.22088      -0.21054       -0.2008
    r80     -11.937     -11.924     -11.841     -11.623     -11.248      -0.16976      -0.16179       -0.1542
    r81     -10.294     -10.281     -10.198     -9.9804     -9.6057      -0.11711      -0.11162       -0.1064
    r82     -8.0862     -8.0735     -7.9895     -7.7724     -7.3981     -0.065327     -0.062237      -0.05935
    r83     -4.8723     -4.8595     -4.7755     -4.5584     -4.1854     -0.020965      -0.01997      -0.01903
```

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check_07 = ev07_jaeemk_check2 - ev07_jaeemk_check0;
mn_MPC_U_gain_share_check_07 = (ec07_jaeemk_check2 - ec07_jaeemk_check0)./(welf_checks*mp_params('TR
```

9.4.3 Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:99;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

9.4.4 Analyze Difference in V and C with Check

The difference between V and V with Check, marginal utility gain given the check.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

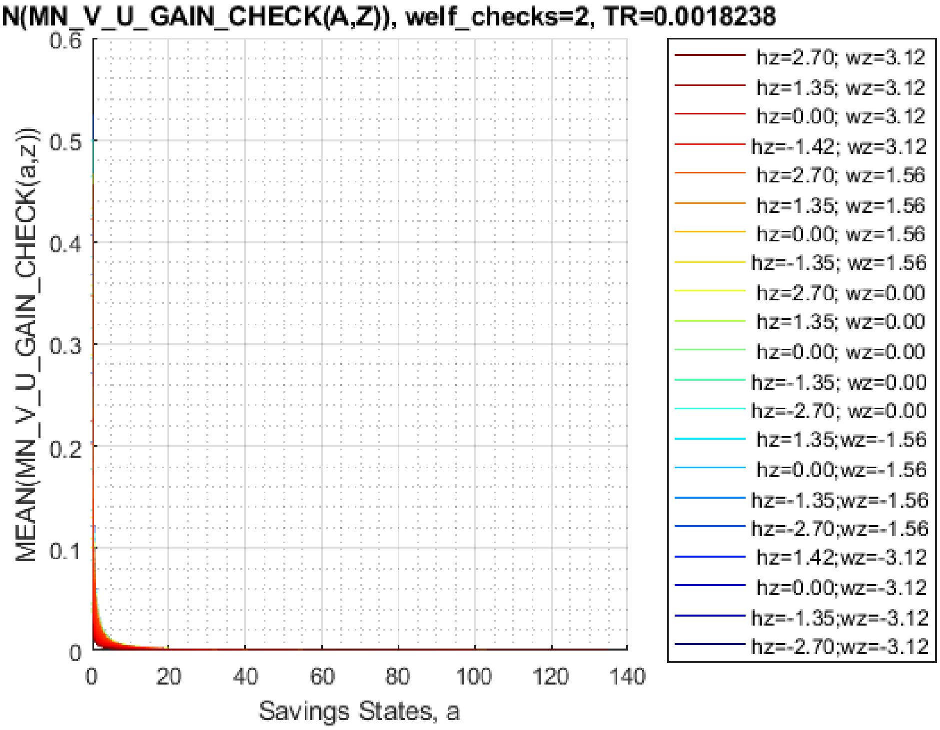
MEAN(MN_V_GAIN_CHECK(A,Z))

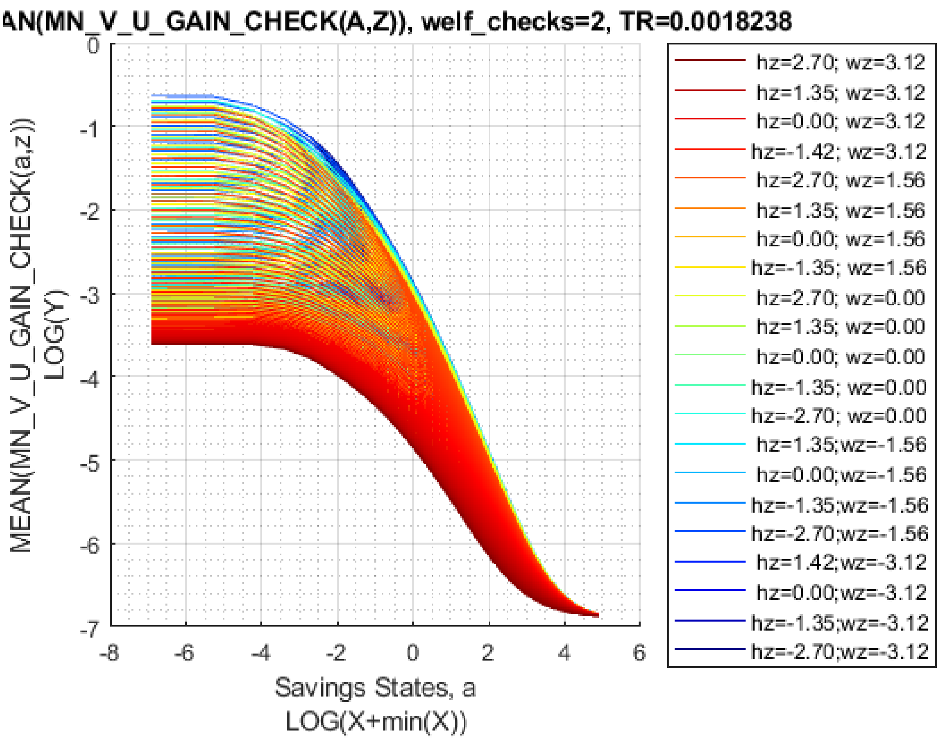
Tabulate value and policies along savings and shocks:

```
% Set
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_par
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check_07, true, ["mean"], 4, 1, cl_mp_datasetdesc,
```

```
xxx  MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0       0.52526       0.48873       0.44813       0.40715        0.3683
```

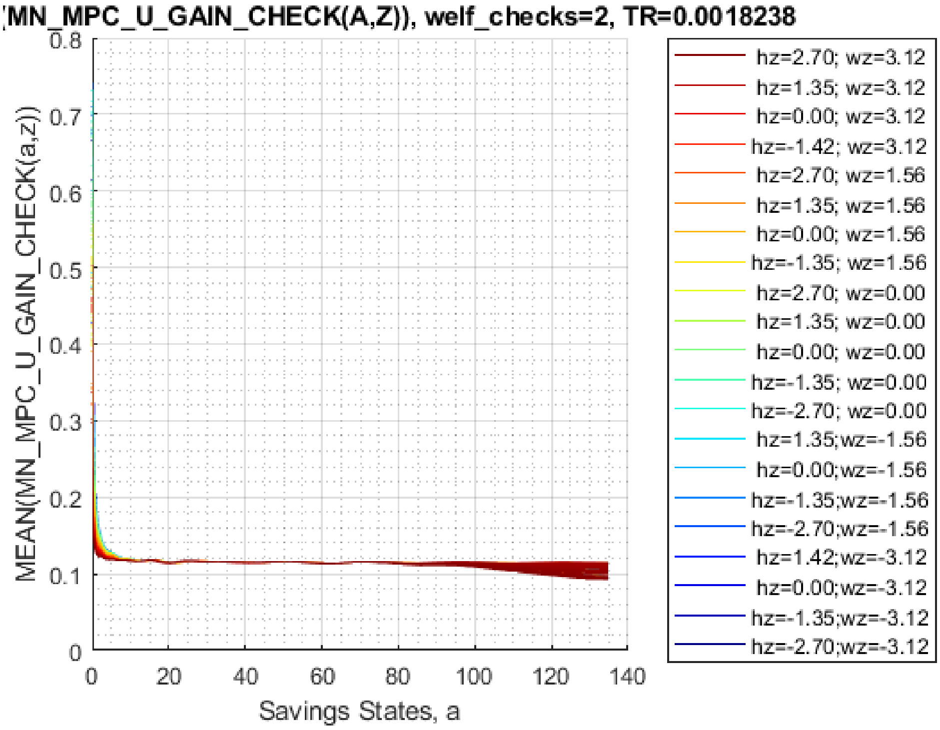
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

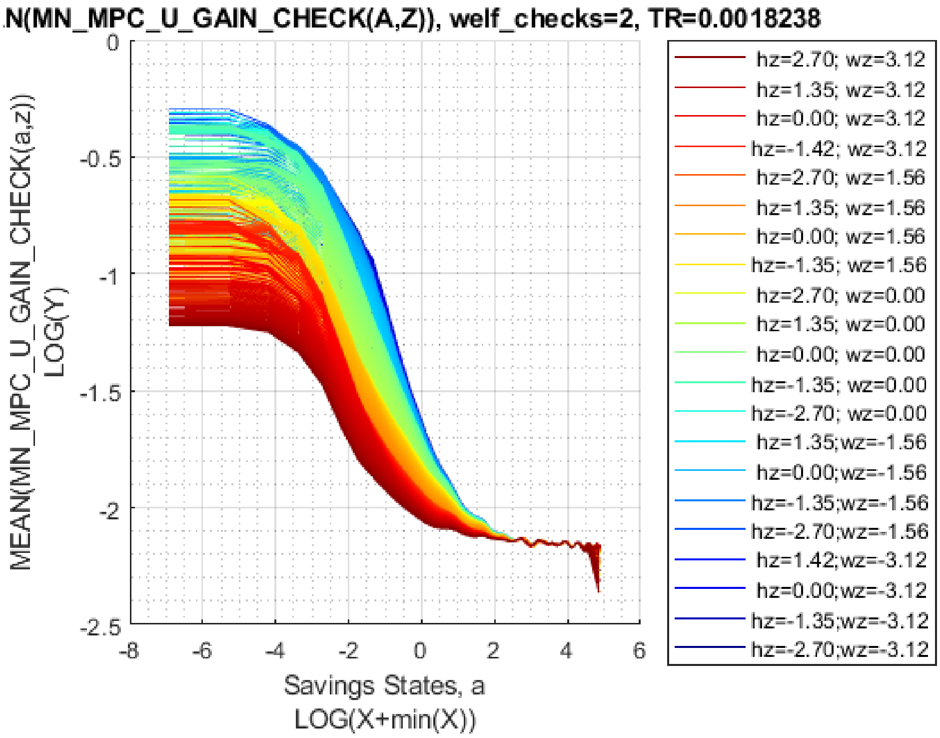




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





9.4.5 Analyze Marginal Value and MPC over Y(a,eta), Conditional On Kids, Marry, Age, Education

Income is generated by savings and shocks, what are the income levels generated by all the shock and savings points conditional on kids, marital status, age and educational levels. Plot on the Y axis MPC, and plot on the X axis income levels, use colors to first distinguish between different a levels, then use colors to distinguish between different eta levles.

Set Up date, Select Age 37vn

, unmarried, no kids, lower education:

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
% 38 year old, unmarried, no kids, lower educated
% Only Household Head Shock Matters so select up to 'n_eta_H_grid'
mn_total_inc_jemk_ss = total_inc_ss(19,:,1:mp_params('n_eta_H_grid'),1,1,1);
mn_V_W_gain_check_use_07 = ev07_jaeemk_check2 - ev07_jaeemk_check0;
mn_C_W_gain_check_use_07 = ec07_jaeemk_check2 - ec07_jaeemk_check0;
```

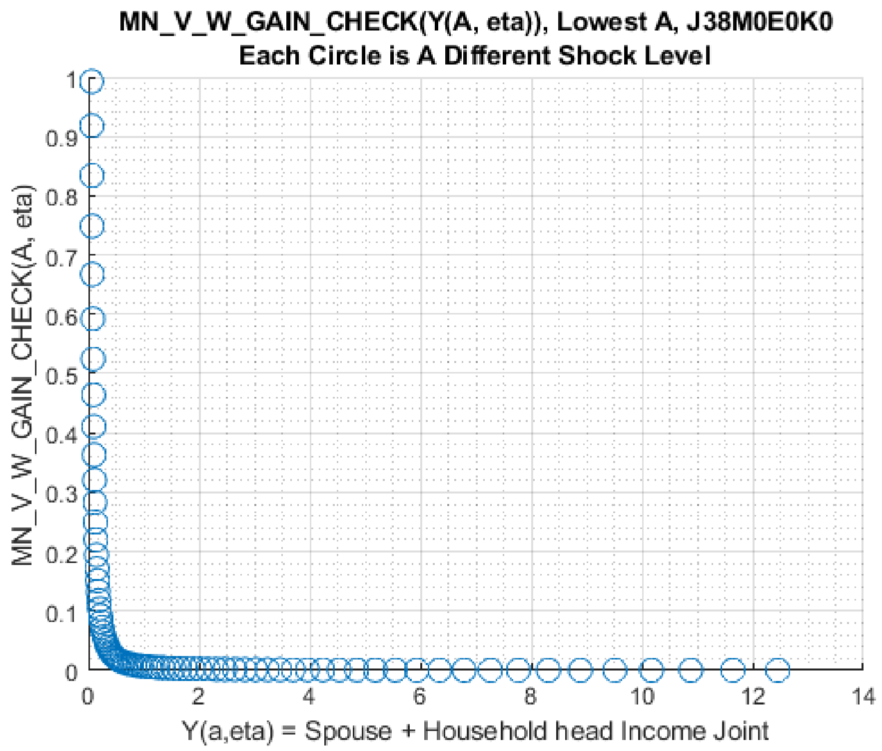
Select Age, Education, Marital, Kids Count:s

```
% Selections
it_age = 21; % +18
it_marital = 1; % 1 = unmarried
it_kids = 1; % 1 = kids is zero
it_educ = 1; % 1 = lower education
% Select: NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
mn_C_W_gain_check_jemk_07 = mn_C_W_gain_check_use_07(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ
mn_V_W_gain_check_jemk_07 = mn_V_W_gain_check_use_07(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ
% Reshape, so shock is the first dim, a is the second
mt_total_inc_jemk = permute(mn_total_inc_jemk_ss,[3,2,1]);
mt_C_W_gain_check_jemk_07 = permute(mn_C_W_gain_check_jemk_07,[3,2,1]);
mt_C_W_gain_check_jemk_07(mt_C_W_gain_check_jemk_07<=1e-10) = 1e-10;
mt_V_W_gain_check_jemk_07 = permute(mn_V_W_gain_check_jemk_07,[3,2,1]);
mt_V_W_gain_check_jemk_07(mt_V_W_gain_check_jemk_07<=1e-10) = 1e-10;
% Generate meshed a and shock grid
[mt_eta_H, mt_a] = ndgrid(eta_H_grid(1:mp_params('n_eta_H_grid')), agrid);
```

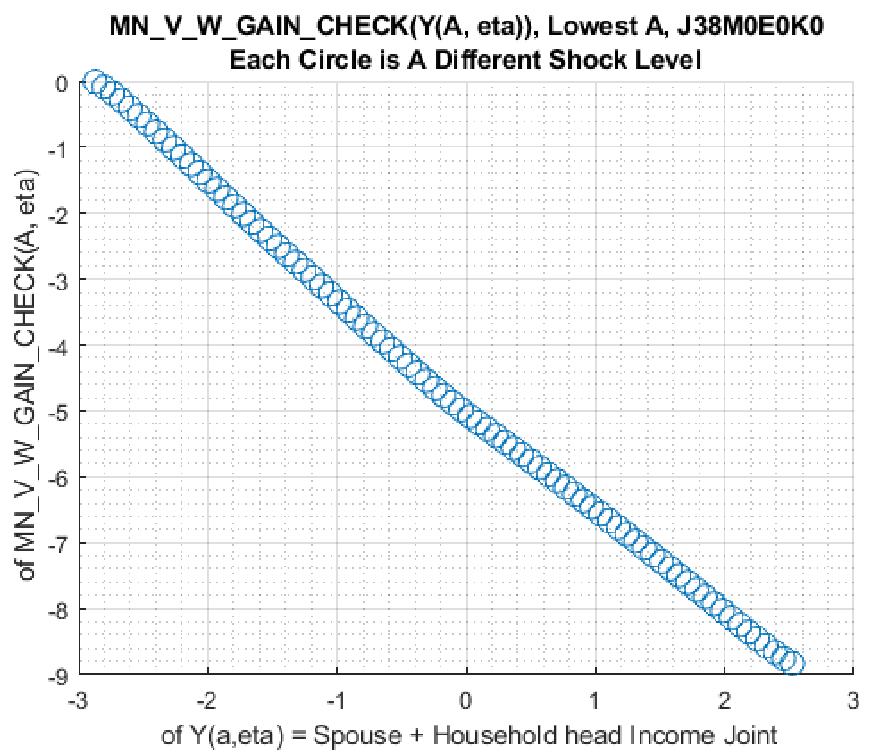
9.4.6 Marginal Value Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
it_a = 1;
scatter((mt_total_inc_jemk(:,it_a)), (mt_V_W_gain_check_jemk_07(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

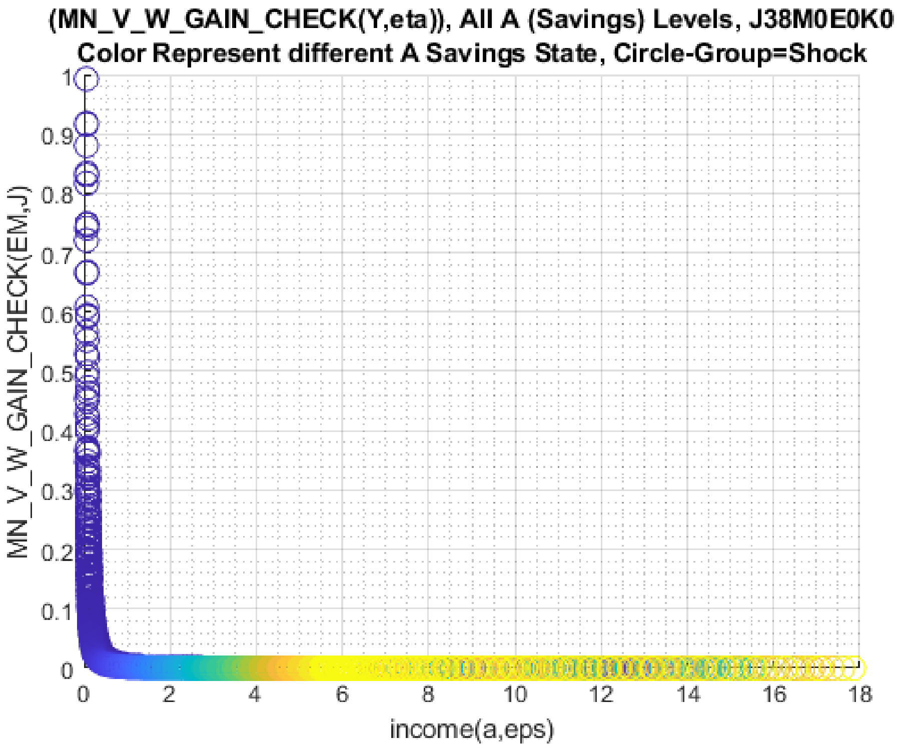



```
figure();
it_shock = 1;
scatter(log(mt_total_inc_jemk(:,it_a)), log(mt_V_W_gain_check_jemk_07(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel(' of Y(a,eta) = Spouse + Household head Income Joint');
ylabel(' of MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

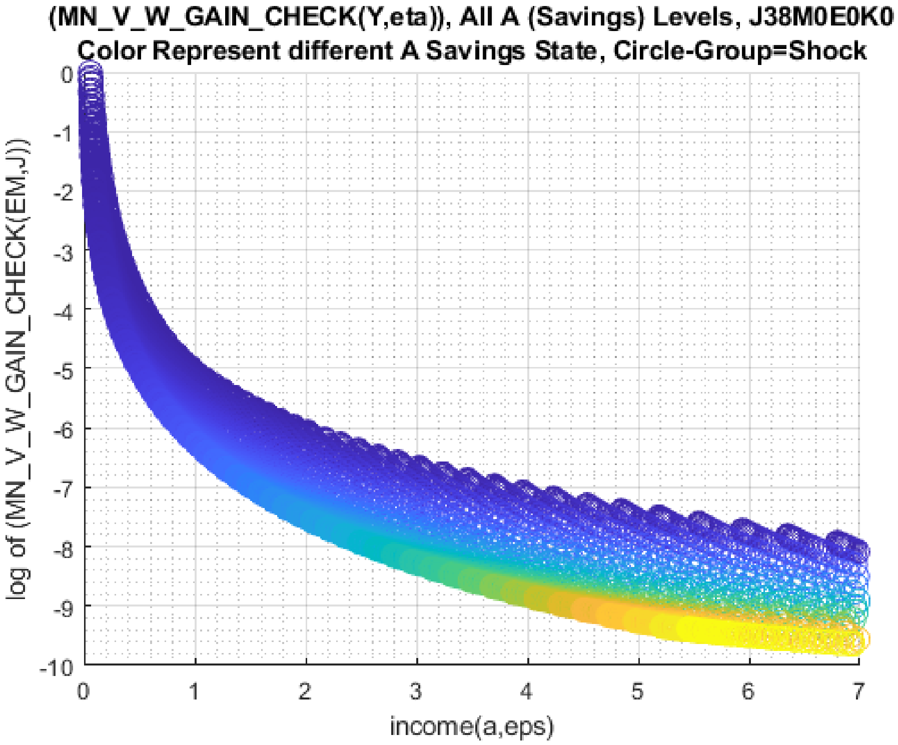


Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_V_W_gain_check_jemk_07(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_V\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```




```
figure();
scatter((mt_total_inc_jemk(:)), log(mt_V_W_gain_check_jemk_07(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
xlim([0,7]);
grid on;
grid minor;
```



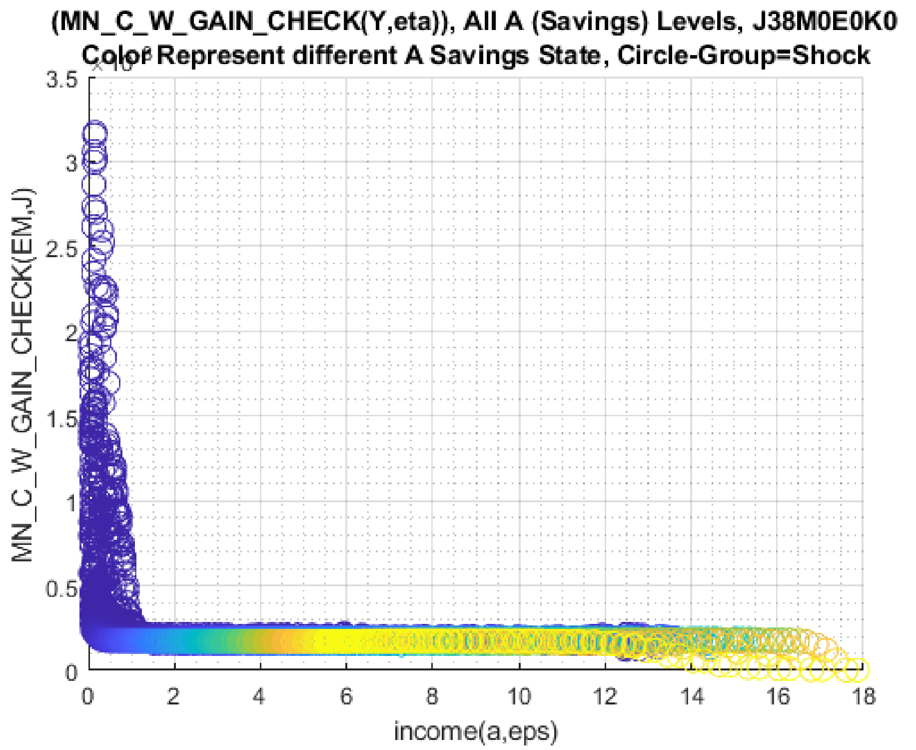
9.4.7 Marginal Consumption Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

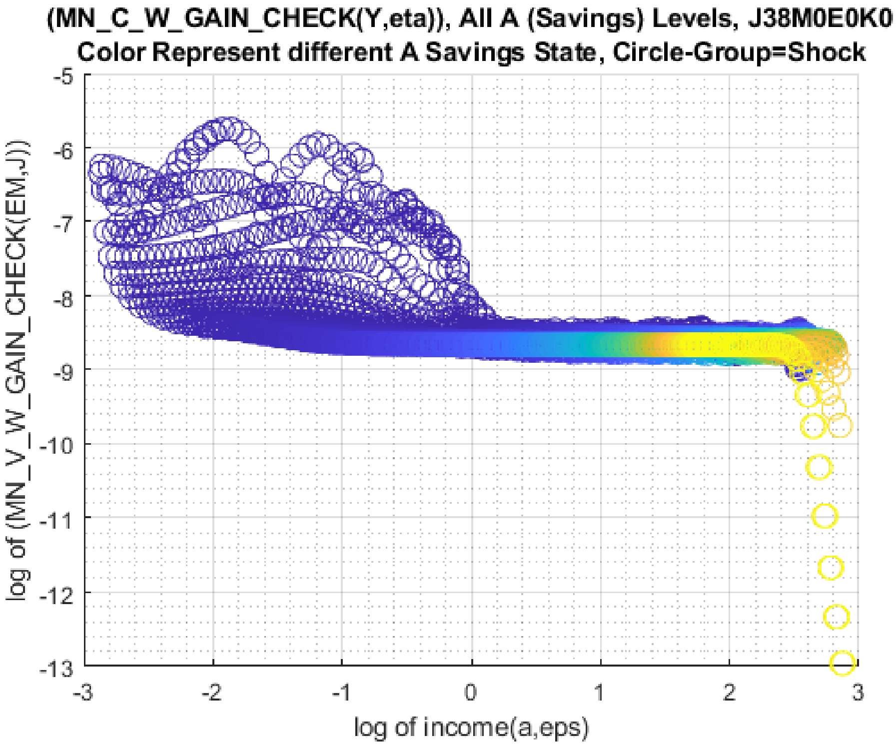
```
figure();
it_a = 50;
scatter(log(mt_total_inc_jemk(:,it_a)), mt_C_W_gain_check_jemk_07(:,it_a), 100);
title({'MN\_C\_W\_GAIN\_CHECK(Y(A, eta)), Given A Savings Level, J38M0E0K0', ...
    'Each Circle is A Different shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_C\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_C_W_gain_check_jemk_07(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_C\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```



```
figure();
scatter(log(mt_total_inc_jemk(:)), log(mt_C_W_gain_check_jemk_07(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('log of income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
grid on;
grid minor;
```



9.4.8 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

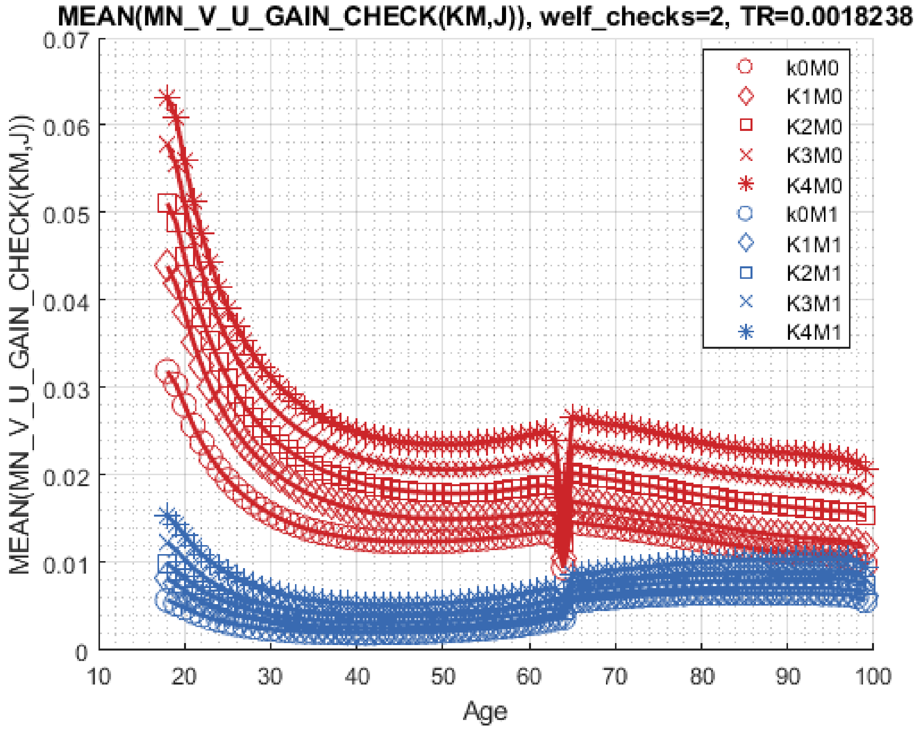
Tabulate value and policies:

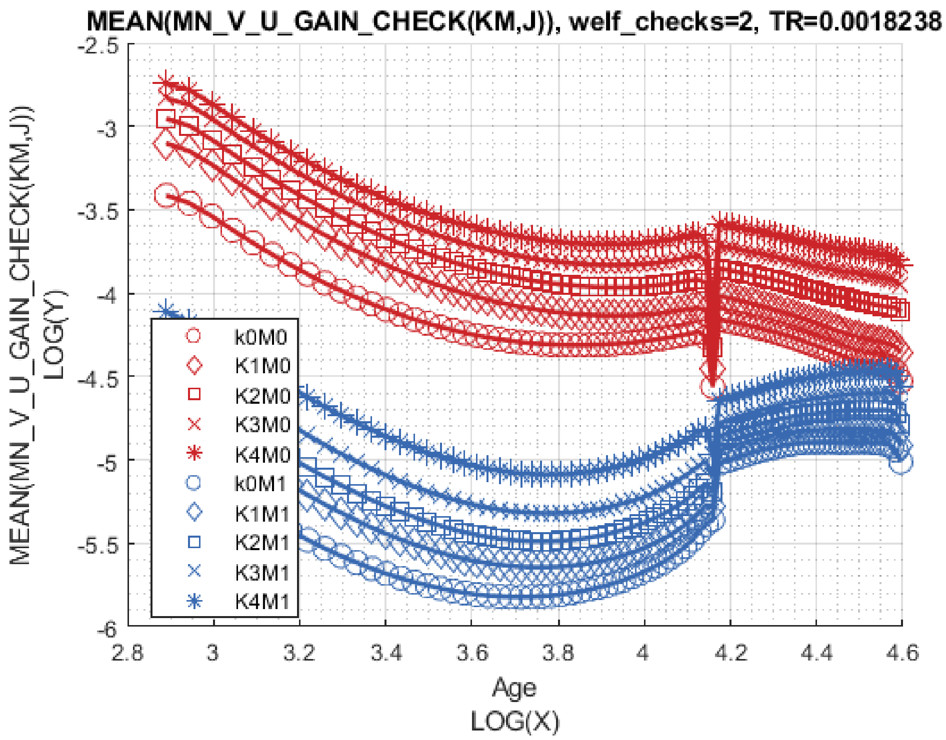
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
```

```
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check_07, true, ["mean"], 3, 1, cl_mp_datasetdesc,

xxx  MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0         0.031908       0.030377       0.028045       0.025708       0.023751
      2        2       0         0.043916        0.04186       0.038596       0.035292       0.032514
      3        3       0         0.051104        0.04891       0.044943        0.04117       0.038001
      4        4       0         0.057927       0.055555       0.051064       0.046812       0.043241
      5        5       0          0.06325       0.060842       0.055977       0.051401       0.047557
      6        1       1         0.005762      0.0053279      0.0048412      0.0043955      0.0040141
      7        2       1        0.0081705      0.0075587      0.0068473      0.0061989       0.005658
      8        3       1        0.0099025      0.0091765      0.0083275      0.0075396      0.0068765
      9        4       1         0.012339       0.011465        0.01043      0.0094589       0.008631
     10        5       1         0.015374       0.014388       0.013133        0.01196       0.010966

% Consumption Function
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check_07, true, ["mean"], 3, 1, cl_mp_datas

xxx  MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        0.078069       0.086102       0.087377       0.085109       0.082559
      2        2       0        0.088091       0.096475       0.099705       0.096801       0.094582
      3        3       0        0.096309        0.10647        0.11111        0.10818        0.10451
      4        4       0         0.10137        0.11245        0.11806        0.11424        0.11063
      5        5       0         0.10569        0.11834        0.12441        0.11996        0.11559
      6        1       1        0.091639       0.097614       0.095796        0.09274       0.094336
      7        2       1        0.098891        0.10394        0.10413        0.10026       0.097592
      8        3       1         0.10228        0.10811        0.10782        0.10839        0.10652
      9        4       1         0.10459        0.11283        0.11208        0.11053        0.10913
     10        5       1         0.11468        0.12506        0.12335        0.12035        0.11807
```

Graph Mean Values:

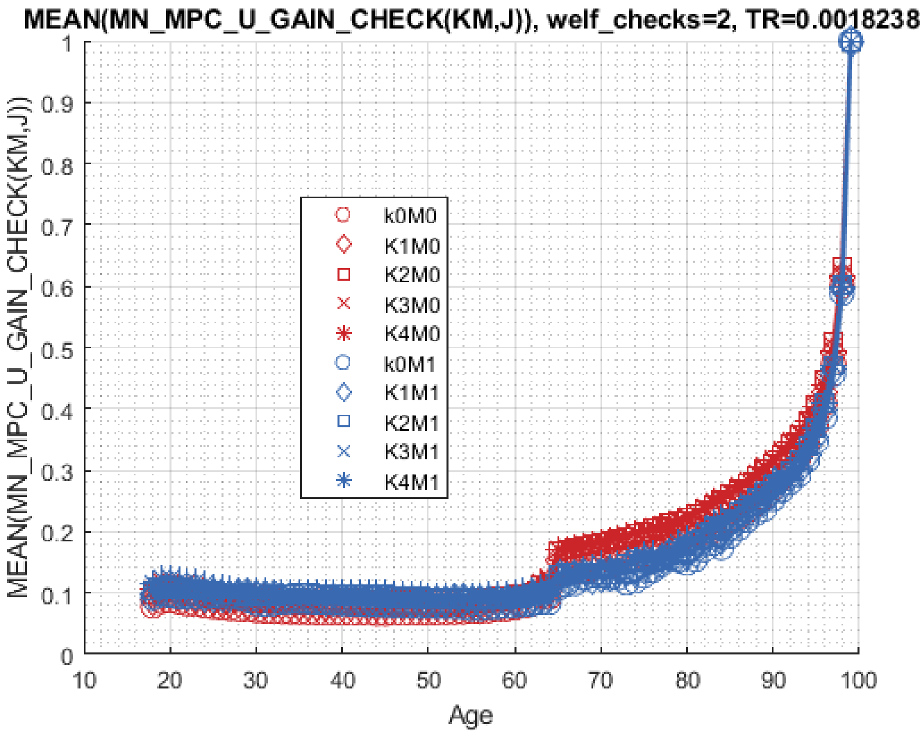
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

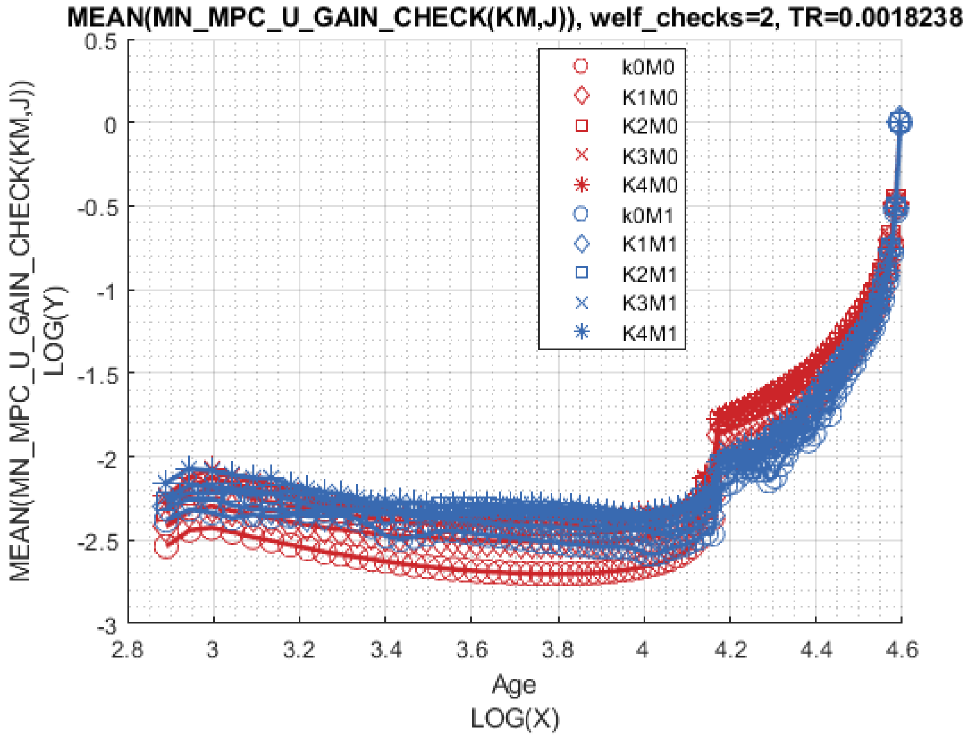




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





9.4.9 Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

```
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

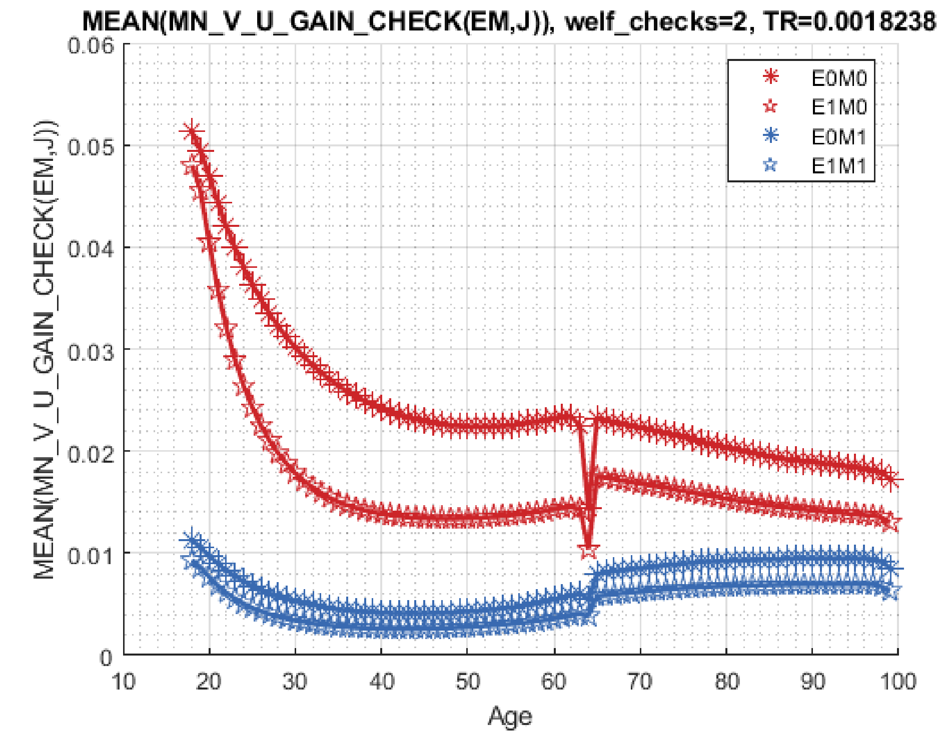
MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

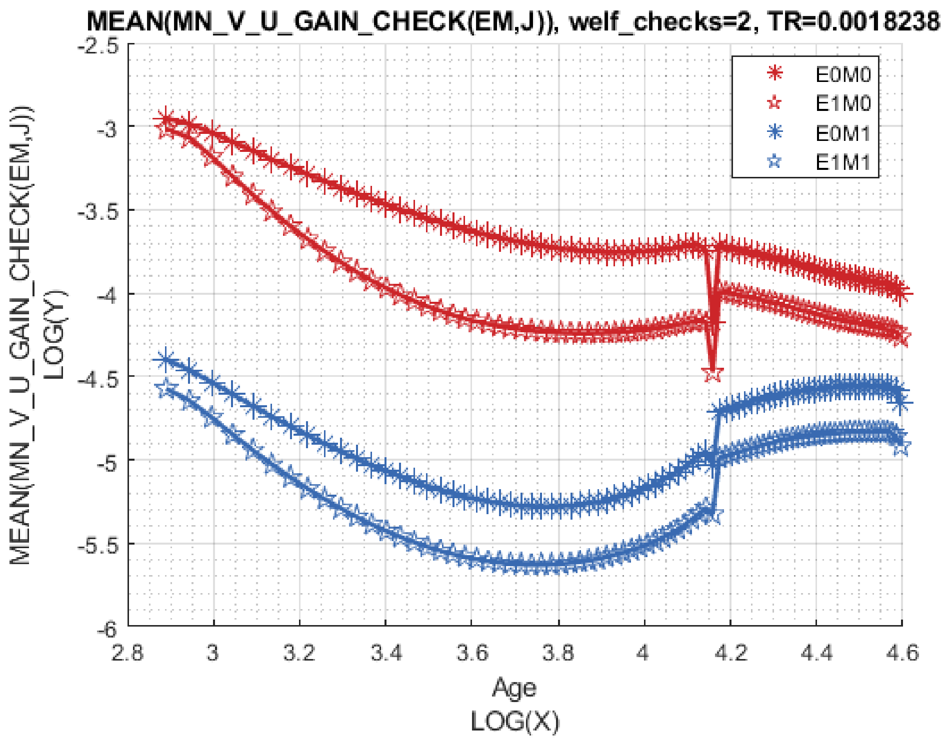
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check_07, true, ["mean"], 3, 1, cl_mp_datasetdesc,

xxx  MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        0.051256       0.049485       0.046969       0.044331       0.041985
      2       1       0        0.047986       0.045532       0.040481       0.035822       0.032041
      3       0       1         0.01129       0.010552      0.0097415      0.0089796      0.0083154
      4       1       1       0.0093296      0.0086141      0.0076904      0.0068414      0.0061429

% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check_07, true, ["mean"], 3, 1, cl_mp_datas

xxx  MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0018238  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0       0.081794       0.085652        0.08723        0.08665       0.085947
      2       1       0        0.10602        0.12228        0.12903        0.12307         0.1172
      3       0       1       0.093649       0.097104       0.096702       0.095946       0.095996
      4       1       1        0.11118        0.12192        0.12057        0.11696        0.11426
```

Graph Mean Values:

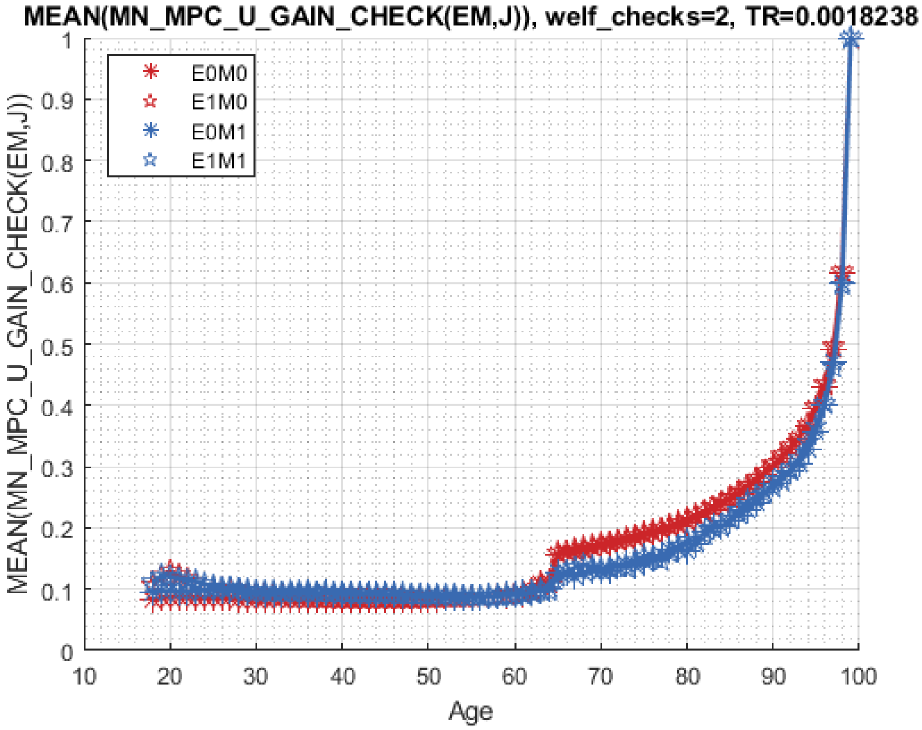
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

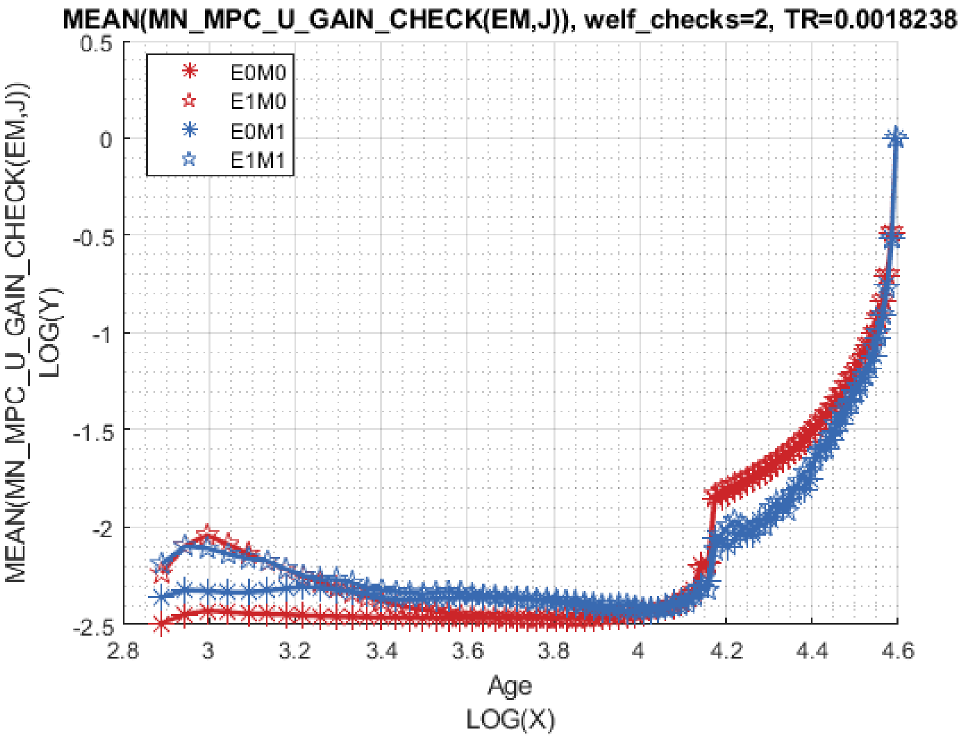




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```


MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0018238
E0M0
E1M0
E0M1
E1M1
MEAN(MN_MPC_U_GAIN_CHECK(EM,J))
Age


MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0018238
E0M0
E1M0
E0M1
E1M1
MEAN(MN_MPC_U_GAIN_CHECK(EM,J))
LOG(Y)
Age
LOG(X)

Chapter 10

2019 Expectations Given Income, Age, Kids and Marital Status

10.1 2019 Age, Income, Kids, Marry EV and EC of One Check

This is the example vignette for function: **snw_evuvw19_jmky** from the **PrjOptiSNW Package.** 2019 integrated over VU and VW

10.1.1 Test SNW_EVUVW19_JMKY Defaults Dense

Set Parameters

Call the function with defaults.

```
clear all;
st_solu_type = 'bisec_vec';

% Solve the VFI Problem and get Value Function
% mp_params = snw_mp_param('default_tiny');
% mp_params = snw_mp_param('default_dense');
mp_params = snw_mp_param('default_docdense');
mp_params('beta') = 0.95;
mp_controls = snw_mp_control('default_test');

% set Unemployment Related Variables
xi=0.5; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=1
b=0; % Unemployment insurance replacement rate (b=0 refers to no UI benefits; b=1 refers to 100 perc
TR=100/58056; % Value of a welfare check (can receive multiple checks). TO DO: Update with alternati

mp_params('xi') = xi;
mp_params('b') = b;
mp_params('TR') = TR;

% Check Numbers
% n_incgrid=201; % Number of income groups
% n_incgrid_aux=round(0.75*n_incgrid);
% inc_grid1=linspace(0,4,n_incgrid_aux)'; % 4 refers to 4*58056=232224 dollars in 2012USD
% inc_grid=[inc_grid1;linspace(4+((7-4)/(n_incgrid-n_incgrid_aux)),7,n_incgrid-n_incgrid_aux)']; % 7
n_incgrid=201; % Number of income groups
inc_grid=linspace(0,7,n_incgrid)';
mp_params('n_incgrid') = n_incgrid;
mp_params('inc_grid') = inc_grid;
```

```
% Solve for Unemployment Values
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jaeemk') = false;
mp_controls('bl_print_evuvw19_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jmky') = false;
```

10.1.2 Solve VFI and Distributon

```
% Solve the Model to get V working and unemployed
[V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);

Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=533.

inc_VFI = mp_valpol_more_ss('inc_VFI');
spouse_inc_VFI = mp_valpol_more_ss('spouse_inc_VFI');
total_inc_VFI = inc_VFI + spouse_inc_VFI;

% COVID year tax
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
% 2020 V and C same as V_SS and cons_ss if tax the same
if (mp_params('a2_covidyr') == mp_params('a2'))
    % mana from heaven
    V_ss_2020 = V_ss;
    cons_ss_2020 = cons_ss;
else
    % change xi and b to for people without unemployment shock
    % solving for employed but 2020 tax results
    % a2_covidyr > a2, we increased tax in 2020 to pay for covid and other
    % costs resolve for both employed and unemployed
    xi = mp_params('xi');
    b = mp_params('b');
    mp_params('xi') = 1;
    mp_params('b') = 0;
    [V_ss_2020,~,cons_ss_2020,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
    mp_params('xi') = xi;
    mp_params('b') = b;
end
% Solve unemployment
[V_unemp_2020,~,cons_unemp_2020] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);

Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d

[Phi_true] = snw_ds_main(mp_params, mp_controls, ap_ss, cons_ss, mp_valpol_more_ss);

Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1486.9836

% Get Matrixes
cl_st_precompute_list = {'a', ...
    'inc', 'inc_unemp', 'spouse_inc', 'spouse_inc_unemp', 'ref_earn_wageind_grid', ...
```

```
    'ap_idx_lower_ss', 'ap_idx_higher_ss', 'ap_idx_lower_weight_ss', ...
    'inc_tot_ygroup_grid'};
mp_controls('bl_print_precompute_verbose') = false;
```

10.1.3 Pre-Compute Matrixes and YMKY Mass

```
% Pre-compute
[mp_precompute_res] = snw_hh_precompute(mp_params, mp_controls, cl_st_precompute_list, ap_ss, Phi_tr

Wage quintile cutoffs=0.4645     0.71528     1.0335     1.5632
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time cost=405.

inc_tot_ygroup_grid = mp_precompute_res('inc_tot_ygroup_grid');
% YMKY Mass
[Phi_true_jmky] = snw_evuvw19_jmky_mass(mp_params, mp_controls, Phi_true, inc_tot_ygroup_grid);

SNW_EVUVW19_JMKY_MASS Start
Completed SNW_EVUVW19_JMKY_MASS;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=12.07
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                    i    idx    ndim      numel       rowN      colN        sum         mean
                    -    ---    ----    ----------    ----    ---------    ------    ----------

    Phi_true        1     1      6        4.37e+07     83     5.265e+05    45.793    1.0479e-06
    Phi_true_jmky   2     2      4      1.6482e+05     82          2010    45.787     0.0002778
```

10.1.4 Solve for 2019 Evuvw With 0 and 2 Checks

Zero checks:

```
% Solve ev 19 JAEEMK
welf_checks = 0;
[ev19_jaeemk_check0, ec19_jaeemk_check0, ev20_jaeemk_check0, ec20_jaeemk_check0] = ...
    snw_evuvw19_jaeemk(...
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss_2020, cons_ss_2020, V_unemp_2020, cons_unemp_2020, mp_precompute_res);

Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=st_biden_or_trump_undefined;welf_checks=0;TR=0.001722
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=0;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_docde
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=8.03
Completed SNW_EVUVW19_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=4262.196

% Solve ev 19 JMKY
[ev19_jmky_check0, ec19_jmky_check0] = snw_evuvw19_jmky(...
    mp_params, mp_controls, ...
    ev19_jaeemk_check0, ec19_jaeemk_check0, ...
    Phi_true, Phi_true_jmky, inc_tot_ygroup_grid);

Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=18.7645
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                    i    idx    ndim      numel       rowN      colN          sum            mean
                    -    ---    ----    ----------    ----    ---------    -----------    ---------

    Phi_true        1     1      6        4.37e+07     83     5.265e+05         45.793    1.0479e-0
```

```
    Phi_true_jmky    2      2      4      1.6482e+05      82         2010         45.787     0.000277
    ec19_jaeemk      3      3      6      4.3173e+07      82     5.265e+05     1.9659e+08       4.553
    ec19_jmky        4      4      4      1.6482e+05      82         2010     3.4206e+05       2.075
    ev19_jaeemk      5      5      6      4.3173e+07      82     5.265e+05     -6.521e+08      -15.10
    ev19_jmky        6      6      4      1.6482e+05      82         2010    -2.1785e+06      -13.21
```

Two checks:

```
% Solve ev 19 JAEEMK
welf_checks = 1;
[ev19_jaeemk_check2, ec19_jaeemk_check2, ev20_jaeemk_check2, ec20_jaeemk_check2] = ...
    snw_evuvw19_jaeemk(...
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss_2020, cons_ss_2020, V_unemp_2020, cons_unemp_2020, mp_precompute_res);

Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=st_biden_or_trump_undefined;welf_checks=1;TR=0.001722
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=1;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_docde
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=7.84
Completed SNW_EVUVW19_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=4279.329

% Solve ev 19 JMKY
[ev19_jmky_check2, ec19_jmky_check2] = snw_evuvw19_jmky(...
    mp_params, mp_controls, ...
    ev19_jaeemk_check2, ec19_jaeemk_check2, ...
    Phi_true, Phi_true_jmky, inc_tot_ygroup_grid);

Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=19.3794
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                     i     idx    ndim       numel       rowN      colN          sum          mean
                     -     ---    ----    ----------     ----    ---------    -----------    ---------

    Phi_true         1      1      6        4.37e+07      83     5.265e+05         45.793    1.0479e-0
    Phi_true_jmky    2      2      4      1.6482e+05      82         2010         45.787     0.000277
    ec19_jaeemk      3      3      6      4.3173e+07      82     5.265e+05      1.966e+08       4.553
    ec19_jmky        4      4      4      1.6482e+05      82         2010      3.421e+05       2.075
    ev19_jaeemk      5      5      6      4.3173e+07      82     5.265e+05    -6.5176e+08      -15.09
    ev19_jmky        6      6      4      1.6482e+05      82         2010    -2.1774e+06      -13.21
```

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check = ev19_jmky_check2 - ev19_jmky_check0;
mn_MPC_U_gain_share_check = (ec19_jmky_check2 - ec19_jmky_check0)./(welf_checks*mp_params('TR'));
```

10.1.5 Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:99;
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
inc_grid = mp_params('inc_grid');
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'ylower', inc_grid});
```

10.1.6 Analyze Marginal Value and MPC over Y(a,eta), Conditional On Kids, Marry, Age, Education

Income is generated by savings and shocks, what are the income levels generated by all the shock and savings points conditional on kids, marital status, age and educational levels. Plot on the Y axis MPC, and plot on the X axis income levels, use colors to first distinguish between different a levels, then use colors to distinguish between different eta levles.

Set Up date, Select Age 37, unmarried, no kids, lower education:

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
% 38 year old, unmarried, no kids, lower educated
% Only Household Head Shock Matters so select up to 'n_eta_H_grid'
mn_V_W_gain_check_use = ev19_jmky_check2 - ev19_jmky_check0;
mn_C_W_gain_check_use = ec19_jmky_check2 - ec19_jmky_check0;
```

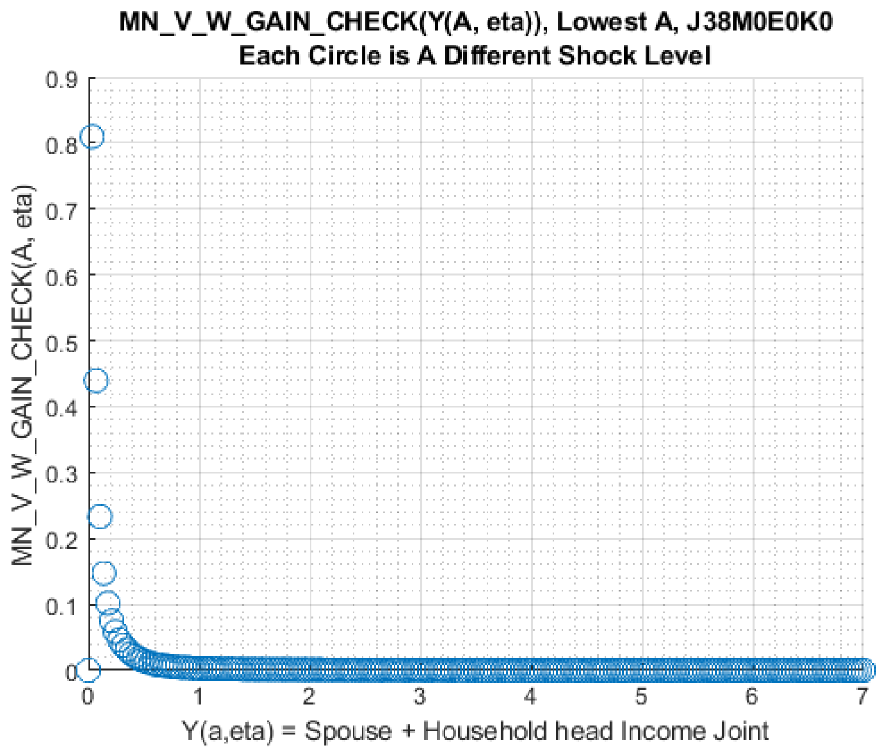
Select Age, Education, Marital, Kids Count:s

```
% Selections
it_age = 21; % +18
it_marital = 1; % 1 = unmarried
it_kids = 1; % 1 = kids is zero
% Select: NaN(n_jgrid-1,n_marriedgrid,n_kidsgrid,n_incgrid);
mn_C_W_gain_check_jemk = mn_C_W_gain_check_use(it_age, it_marital, it_kids, :);
mn_V_W_gain_check_jemk = mn_V_W_gain_check_use(it_age, it_marital, it_kids, :);
% Reshape, so shock is the first dim, a is the second
ar_C_W_gain_check_jemk = mn_C_W_gain_check_jemk(:);
ar_V_W_gain_check_jemk = mn_V_W_gain_check_jemk(:);
```

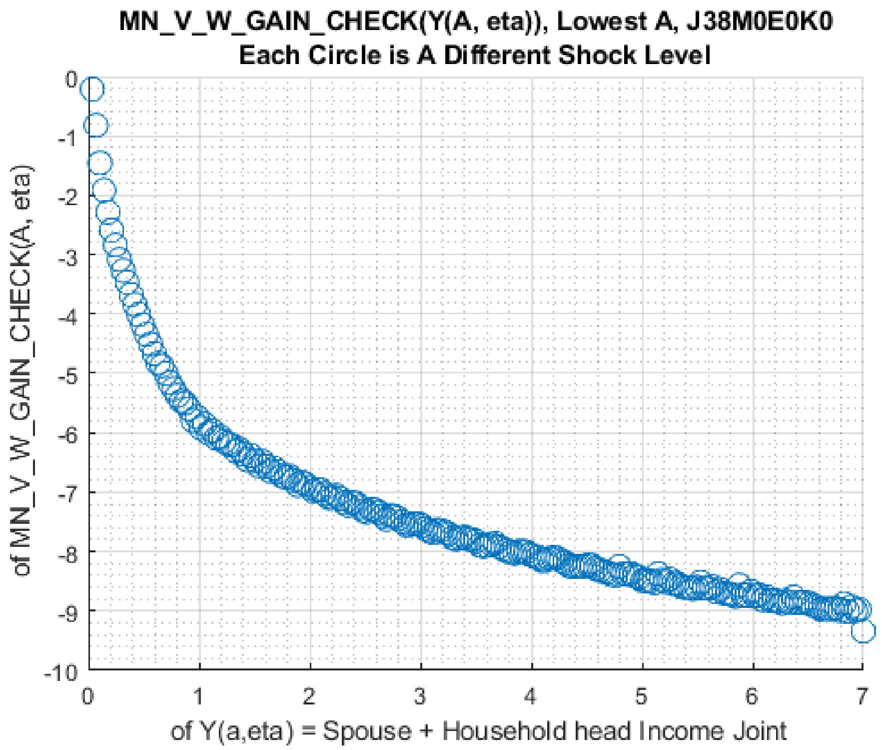
Marginal Value Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
scatter(inc_grid, ar_V_W_gain_check_jemk, 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```



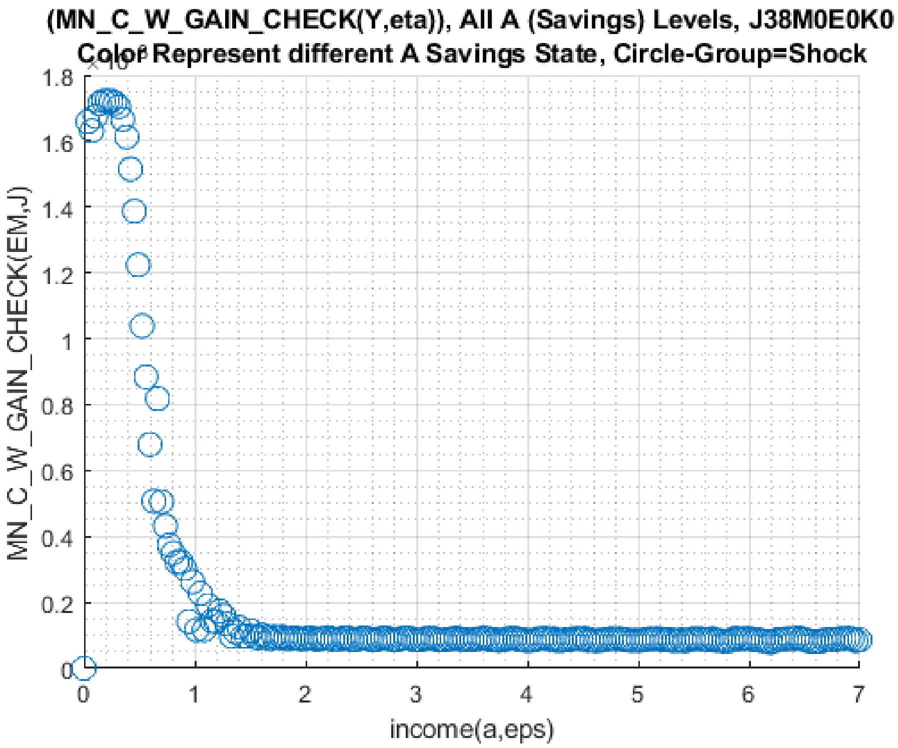

```
figure();
it_shock = 1;
scatter((inc_grid), log(ar_V_W_gain_check_jemk), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel(' of Y(a,eta) = Spouse + Household head Income Joint');
ylabel(' of MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```



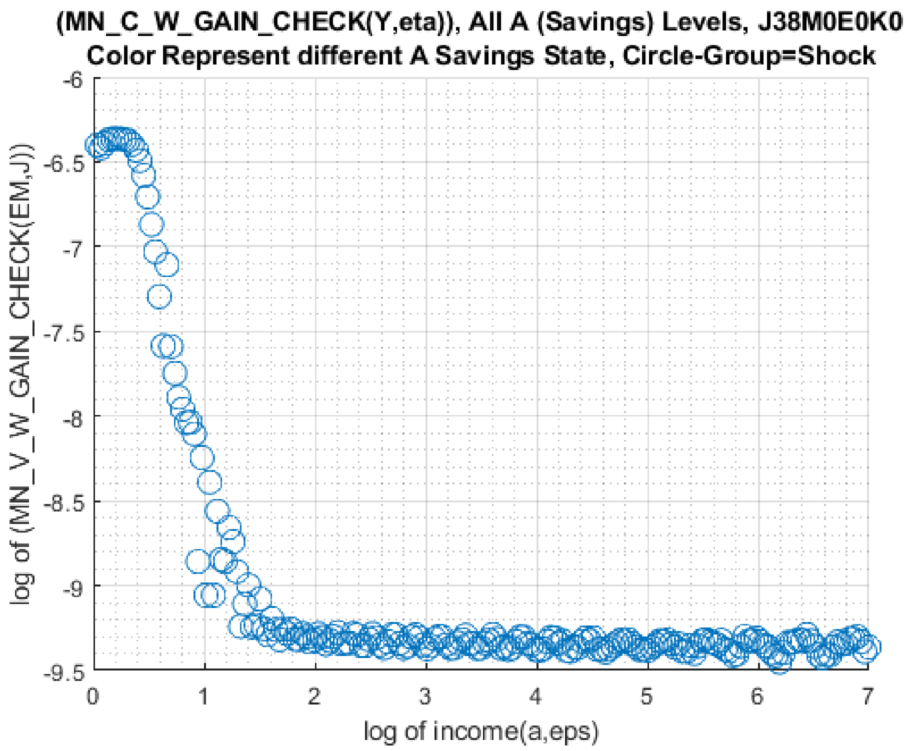
Marginal Consumption Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

Plot all asset levels:

```
figure();
scatter(inc_grid, ar_C_W_gain_check_jemk, 100);
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_C\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```




```
figure();
scatter((inc_grid), log(ar_C_W_gain_check_jemk), 100);
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('log of income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
grid on;
grid minor;
```



10.2 2019/2020 Age, Income, Kids, Marry EV and EC All Checks (Biden Checks)

This is the example vignette for function: **snw_evuvw19_jmky_allchecks** from the **PrjOptiSNW Package.** 2019 integrated over VU and VW

The key features of the Biden stimulus check are: i) determined based on 2020 information; ii) checks received in 2021, ex-post the realization of the second one-time MIT shock, which similar to the first MIT shock, is conditional on income and age groups; iii) state of the economy returns to steady-state in 2022; iv) trump checks were provided in 2020, which changes saves and consumption and distributions in 2020, and appropriate adjustments are made so that the biden check is conditional on the distributional changes in endogenous savings due to the Trump checks.

10.2.1 Test SNW_EVUVW19_JMKY_ALLCHECKS Parameters for Biden Checks

Save a result that is low in memory cost so that it can be loaded quickly for various allocation tests. Turn off Various Printing Controls. Call function with wide income bins to reduce memory storage and retrievel costs

```
clear all;
% Start mp contorls
mp_controls = snw_mp_control('default_test');
% Solve for Unemployment Values
mp_controls('bl_timer') = true;
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = true;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
```

```
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jaeemk') = false;
mp_controls('bl_print_evuvw19_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jmky') = false;
mp_controls('bl_print_evuvw19_jmky_verbose') = false;
```

Dense default, and unemployment parameters:

```
% default dense load
% mp_params = snw_mp_param('default_dense');
mp_params = snw_mp_param('default_docdense')

mp_params =
  Map with properties:

        Count: 64
      KeyType: char
    ValueType: any

mp_params('beta') = 0.95;
fl_scaleconvertor = 62502;
mp_more_inputs('fl_scaleconvertor') = fl_scaleconvertor;
% Unemployment
xi = 0.651; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; x
b=1; % Unemployment insurance replacement rate (b=0 refers to no UI benefits; b=1 refers to 100 perc
TR=100/fl_scaleconvertor; % Value of a wezlfare check (can receive multiple checks). TO DO: Update w
mp_params('pi_unemp') = mp_params('pi_unemp_2020_juneadj');
mp_params('xi') = xi;
mp_params('b') = b;
mp_params('TR') = TR;
% Check Count: 89 checks to allow for both the first and the second round
n_welfchecksgrid = 3;
mp_params('n_welfchecksgrid') = n_welfchecksgrid;
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
```

Income bins:

```
% Income Grid
% 4 refers to 4*58056=232224 dollars in 2012USD
% max 7 refers to 7*58056=406392 dollars in 2012USD
% all phase out = (4400/5)*100 + 150000 = 238000
% if 500 dollar interval, need 476 inc groups before 238000
% if have 85 percent of points betwen 238000,
fl_max_phaseout = 238000;
fl_multiple = fl_scaleconvertor;
it_bin_dollar_before_phaseout = 5000;
it_bin_dollar_after_phaseout = 25000;
fl_thres = fl_max_phaseout/fl_multiple;
inc_grid1 = linspace(0,fl_thres,(fl_max_phaseout)/it_bin_dollar_before_phaseout);
inc_grid2 = linspace(fl_thres, 7, (7*fl_multiple-fl_max_phaseout)/it_bin_dollar_after_phaseout);
inc_grid=sort(unique([inc_grid1 inc_grid2]'));
mp_params('n_incgrid') = length(inc_grid);
mp_params('inc_grid') = inc_grid;
```

10.2.2 SNW_EVUVW19_JMKY_ALLCHECKS Low Storage Invoke for Biden Checks

The simulation here (dense) requires less than 10 GB of memory with 8 workers (8 threads needed), simulating over 88 checks takes with 8 workers

```
st_biden_or_trump = 'bidenchk';
st_solu_type = 'bisec_vec';
bl_parfor = false;
it_workers = 1;
bl_export = false;
bl_load_mat = false;
snm_suffix = ['_test_ybin' num2str(it_bin_dollar_before_phaseout)];
[ev19_jmky_allchecks, ec19_jmky_allchecks, output] = ...
    snw_evuvw19_jmky_allchecks(mp_params, mp_controls, ...
    st_biden_or_trump, st_solu_type, ...
    bl_parfor, it_workers, ...
    bl_export, bl_load_mat, snm_suffix);

Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=338.
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
sum of Phi_adj:83
sum of Phi_true:45.7931
sum of Phiss:83
summ of diff of Phiss and Phi_adj:-3.4939e-12
summ of diff of Phiss and Phi_true:37.2069
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=951.9985
Biden Check, resolve for distributions given Trump check
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
sum of Phi_adj:83
sum of Phi_true:45.7931
sum of Phiss:83
summ of diff of Phiss and Phi_adj:-1.0845e-13
summ of diff of Phiss and Phi_true:37.2069
summ of diff of Phi_adj_base and Phiss:1.0838e-13
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1016.4233
Wage quintile cutoffs=0.4645     0.71528      1.0335      1.5632
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time cost=300.
SNW_EVUVW19_JMKY_MASS Start
Completed SNW_EVUVW19_JMKY_MASS;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=5.304
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                    i    idx    ndim     numel      rowN      colN        sum         mean
                    -    ---    ----    --------    ----    ---------    ------    ----------    --

    Phi_true        1     1      6      4.37e+07     83     5.265e+05    45.793    1.0479e-06    1.
    Phi_true_jmky   2     2      4         43460     82           530    45.787     0.0010535     0

SNW_EVUVW19_JMKY_ALLCHECKS Start
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bidenchk;welf_checks=0;TR=0.0015999;SNW_MP_PARAM=defa
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=0;TR=0.0015999;xi=0.651;b=1;SNW_MP_PARAM=default_doc
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=8.32
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bidenchk;SNW_MP_PARAM=default_docdense;SNW_MP_CON
Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=9.5927
SNW_EVUVW19_JMKY_ALLCHECKS: Finished Check 0 of 2, time=190.7959
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bidenchk;welf_checks=1;TR=0.0015999;SNW_MP_PARAM=defa
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=1;TR=0.0015999;xi=0.651;b=1;SNW_MP_PARAM=default_doc
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=7.75
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bidenchk;SNW_MP_PARAM=default_docdense;SNW_MP_CON
Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=9.57
```

```
SNW_EVUVW19_JMKY_ALLCHECKS: Finished Check 1 of 2, time=190.7479
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bidenchk;welf_checks=2;TR=0.0015999;SNW_MP_PARAM=defa
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=2;TR=0.0015999;xi=0.651;b=1;SNW_MP_PARAM=default_doc
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=7.98
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bidenchk;SNW_MP_PARAM=default_docdense;SNW_MP_CON
Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=9.4891
SNW_EVUVW19_JMKY_ALLCHECKS: Finished Check 2 of 2, time=191.4103
Completed SNW_EVUVW19_JMKY_ALLCHECKS;ST_BIDEN_OR_TRUMP=bidenchk;SNW_MP_PARAM=default_docdense;SNW_MP
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum
	-	---	----	----------	----------	-----	--------
Output	1	1	2	1.0453e+06	1.1615e+05	9	7.7287e
ec19_jmky_allchecks	2	2	5	1.3038e+05	3	43460	2.7167e
ec19_jmky_allchecks_posmass	3	3	2	1.1615e+05	1.1615e+05	1	2.7167e
ev19_jmky_allchecks	4	4	5	1.3038e+05	3	43460	-2.3026e
ev19_jmky_allchecks_posmass	5	5	2	1.1615e+05	1.1615e+05	1	17

xxx TABLE:Output xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c6	c7	c8	c9
	--	--	--	--	----------	--------	---------	--------
r1	18	0	0	0	2.1599e-05	-0.57722	0.0053864	0.068495
r2	18	0	0	1	2.1599e-05	-0.57722	0.0053978	0.069683
r3	18	0	0	2	2.1599e-05	-0.57722	0.005409	0.070689
r4	19	0	0	0	1.9002e-05	0.42278	0.0056805	0.075684
r5	19	0	0	1	1.9002e-05	0.42278	0.0056903	0.07644
r116141	86	1	4	1	3.9923e-49	4.2268	0.97814	13.904
r116142	86	1	4	2	3.9923e-49	4.2268	0.97816	13.904
r116143	87	1	4	0	7.923e-63	4.2413	1.071	14.588
r116144	87	1	4	1	7.923e-63	4.2413	1.071	14.588
r116145	87	1	4	2	7.923e-63	4.2413	1.0711	14.589

xxx TABLE:ec19_jmky_allchecks xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c43457	c43458	c43459	c43460
	--------	--------	--------	--------	------	------	------	------
r1	0.068495	0.075684	0.077039	0.075763	0	0	0	0
r2	0.069683	0.07644	0.077934	0.076696	0	0	0	0
r3	0.070689	0.077196	0.078828	0.077629	0	0	0	0

xxx TABLE:ec19_jmky_allchecks_posmass xxxxxxxxxxxxxxxxxx

	c1
	--------
r1	0.068495
r2	0.069683
r3	0.070689
r4	0.075684
r5	0.07644
r116141	13.904
r116142	13.904
r116143	14.588
r116144	14.588

```
    r116145      14.589

xxx TABLE:ev19_jmky_allchecks xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4        c43457    c43458    c43459    c43460
            -------     -------     -------     -------     ------    ------    ------    ------

    r1     -185.65     -176.04     -169.67     -170.69        0         0         0         0
    r2     -185.26     -175.74     -169.37     -170.39        0         0         0         0
    r3     -184.88     -175.43     -169.08     -170.09        0         0         0         0

xxx TABLE:ev19_jmky_allchecks_posmass xxxxxxxxxxxxxxxxxx
                    c1
               ---------

    r1         0.0053864
    r2         0.0053978
    r3          0.005409
    r4         0.0056805
    r5         0.0056903
    r116141      0.97814
    r116142      0.97816
    r116143        1.071
    r116144        1.071
    r116145       1.0711
```

10.3 2007 Age, Income, Kids, Marry EV and EC All Checks (Bush Checks)

This is the example vignette for function: **snw_evuvw19_jmky_allchecks** from the **PrjOptiSNW Package.** 2019 integrated over VU and VW

The function **snw_evuvw19_jmky_allchecks** was initiall designed to handle the COVID problem, the revised version of the program handles both the 2007/8/9 Bush stimulus check problem, and the 2019/20/21 Trump and Biden stimulus check problems.

The key features of the Bush stimulus checks are: i) determined based on 2007 information; ii) checks received in 2008, when the great recession has not arrived yet, but all expect it to in 2009; iii) the Great Recession hits in 2009, putting some people, based on education and age, into unemployment state with a shared unemployment duration and lost income and also UI benefits (calibrated to match overall UI share of wages); iv) the economy returns to steady-state in 2010.

10.3.1 Test SNW_EVUVW19_JMKY_ALLCHECKS Parameters for Bush Checks

Save a result that is low in memory cost so that it can be loaded quickly for various allocation tests. Turn off Various Printing Controls. Call function with wide income bins to reduce memory storage and retrievel costs

```
clear all;
% Start mp contorls
mp_controls = snw_mp_control('default_test');
% Solve for Unemployment Values
mp_controls('bl_timer') = true;
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = true;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
```

```
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_v08p08_jaeemk') = false;
mp_controls('bl_print_v08p08_jaeemk_verbose') = false;
mp_controls('bl_print_v08_jaeemk') = false;
mp_controls('bl_print_v08_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jaeemk') = false;
mp_controls('bl_print_evuvw19_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jmky') = false;
mp_controls('bl_print_evuvw19_jmky_verbose') = false;
```

Dense default, and unemployment parameters:

```
% default dense load
% 1. generate MP_PARAMS specific to 2008 stimulus
% Use non-default values for Bush Stimulus
mp_more_inputs = containers.Map('KeyType','char', 'ValueType','any');
mp_more_inputs('fl_ss_non_college') = 0.225;
mp_more_inputs('fl_ss_college') = 0.271;
fl_p50_hh_income_07 = 54831;
mp_more_inputs('fl_scaleconvertor') = fl_p50_hh_income_07;
% st_param_group = 'default_small';
% st_param_group = 'default_dense';
st_param_group = 'default_docdense';
mp_params = snw_mp_param(st_param_group, false, 'tauchen', false, 8, 8, mp_more_inputs);
% mp_params = snw_mp_param('default_docdense')
mp_params('beta') = 0.95;
fl_scaleconvertor = 62502;
mp_more_inputs('fl_scaleconvertor') = fl_scaleconvertor;
% Unemployment
mp_params('xi') = 0.532;
mp_params('b') = 0.37992;
mp_params('a2_covidyr') = mp_params('a2_greatrecession_2009');
mp_params('TR') = 100/fl_p50_hh_income_07;
% Check Count: 89 checks to allow for both the first and the second round
n_welfchecksgrid = 3;
mp_params('n_welfchecksgrid') = n_welfchecksgrid;
mp_params('a2_covidyr') = mp_params('a2_greatrecession_2009');
```

Income bins:

```
% Income Grid
% 4 refers to 4*58056=232224 dollars in 2012USD
% max 7 refers to 7*58056=406392 dollars in 2012USD
% all phase out = (4400/5)*100 + 150000 = 238000
% if 500 dollar interval, need 476 inc groups before 238000
% if have 85 percent of points betwen 238000,
fl_max_phaseout = 238000;
fl_multiple = fl_scaleconvertor;
it_bin_dollar_before_phaseout = 5000;
it_bin_dollar_after_phaseout = 25000;
fl_thres = fl_max_phaseout/fl_multiple;
inc_grid1 = linspace(0,fl_thres,(fl_max_phaseout)/it_bin_dollar_before_phaseout);
inc_grid2 = linspace(fl_thres, 7, (7*fl_multiple-fl_max_phaseout)/it_bin_dollar_after_phaseout);
inc_grid=sort(unique([inc_grid1 inc_grid2]'));
mp_params('n_incgrid') = length(inc_grid);
mp_params('inc_grid') = inc_grid;
```

10.3.2 SNW_EVUVW19_JMKY_ALLCHECKS Low Storage Invoke for Bush Checks

The simulation here (dense) requires less than 10 GB of memory with 8 workers (8 threads needed), simulating over 88 checks takes with 8 workers

```
st_biden_or_trump = 'bushchck';
st_solu_type = 'bisec_vec';
bl_parfor = false;
it_workers = 1;
bl_export = false;
bl_load_mat = false;
snm_suffix = ['_test_ybin' num2str(it_bin_dollar_before_phaseout)];
[ev19_jmky_allchecks, ec19_jmky_allchecks, output] = ...
    snw_evuvw19_jmky_allchecks(mp_params, mp_controls, ...
    st_biden_or_trump, st_solu_type, ...
    bl_parfor, it_workers, ...
    bl_export, bl_load_mat, snm_suffix);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=330.
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
Completed SNW_V08P08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=524.1155
sum of Phi_adj:83
sum of Phi_true:45.7931
sum of Phiss:83
summ of diff of Phiss and Phi_adj:-3.4775e-12
summ of diff of Phiss and Phi_true:37.2069
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1254.4898
Trump Check, do not need to resolve distribution
Wage quintile cutoffs=0.4645     0.71528      1.0335      1.5632
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time cost=265.
SNW_EVUVW19_JMKY_MASS Start
Completed SNW_EVUVW19_JMKY_MASS;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=5.264
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                     i    idx    ndim     numel      rowN      colN        sum         mean
                     -    ---    ----    --------    ----    ---------    ------    ----------    --

    Phi_true         1     1      6      4.37e+07     83     5.265e+05    45.793    1.0479e-06    1.
    Phi_true_jmky    2     2      4         43460     82           530    45.787     0.0010535    0

SNW_EVUVW19_JMKY_ALLCHECKS Start
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bushchck;welf_checks=0;TR=0.0018238;SNW_MP_PARAM=defa
Completed SNW_V08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=2.81e-05
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bushchck;SNW_MP_PARAM=default_docdense;SNW_MP_CON
Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=10.3545
SNW_EVUVW19_JMKY_ALLCHECKS: Finished Check 0 of 2, time=106.2974
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bushchck;welf_checks=1;TR=0.0018238;SNW_MP_PARAM=defa
Completed SNW_V08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=2.94e-05
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bushchck;SNW_MP_PARAM=default_docdense;SNW_MP_CON
Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=10.0897
SNW_EVUVW19_JMKY_ALLCHECKS: Finished Check 1 of 2, time=106.6849
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=bushchck;welf_checks=2;TR=0.0018238;SNW_MP_PARAM=defa
```

```
Completed SNW_V08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=2.1e-05
Completed SNW_EVUVW19_JAEEMK_FOC;st_biden_or_trump=bushchck;SNW_MP_PARAM=default_docdense;SNW_MP_CON
Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=10.1555
SNW_EVUVW19_JMKY_ALLCHECKS: Finished Check 2 of 2, time=106.7998
Completed SNW_EVUVW19_JMKY_ALLCHECKS;ST_BIDEN_OR_TRUMP=bushchck;SNW_MP_PARAM=default_docdense;SNW_MP
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum
Output	1	1	2	1.0483e+06	1.1648e+05	9	7.7619e
ec19_jmky_allchecks	2	2	5	1.3038e+05	3	43460	2.737e
ec19_jmky_allchecks_posmass	3	3	2	1.1648e+05	1.1648e+05	1	2.737e
ev19_jmky_allchecks	4	4	5	1.3038e+05	3	43460	-2.3148e
ev19_jmky_allchecks_posmass	5	5	2	1.1648e+05	1.1648e+05	1	17

xxx TABLE:Output xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c6	c7	c8	c9
r1	18	0	0	0	2.1599e-05	-0.57722	0.0053606	0.066639
r2	18	0	0	1	2.1599e-05	-0.57722	0.0053744	0.067396
r3	18	0	0	2	2.1599e-05	-0.57722	0.0053882	0.068355
r4	19	0	0	0	1.9002e-05	0.42278	0.0055828	0.068128
r5	19	0	0	1	1.9002e-05	0.42278	0.0055969	0.069089
r116474	86	1	4	1	3.937e-49	4.2268	0.97866	13.912
r116475	86	1	4	2	3.937e-49	4.2268	0.97868	13.912
r116476	87	1	4	0	1.0014e-58	4.2413	1.0716	14.596
r116477	87	1	4	1	1.0014e-58	4.2413	1.0716	14.597
r116478	87	1	4	2	1.0014e-58	4.2413	1.0716	14.597

xxx TABLE:ec19_jmky_allchecks xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c43457	c43458	c43459	c43460
r1	0.066639	0.068128	0.071766	0.071091	0	0	0	0
r2	0.067396	0.069089	0.072695	0.072031	0	0	0	0
r3	0.068355	0.069782	0.073378	0.07271	0	0	0	0

xxx TABLE:ec19_jmky_allchecks_posmass xxxxxxxxxxxxxxxxxx

	c1
r1	0.066639
r2	0.067396
r3	0.068355
r4	0.068128
r5	0.069089
r116474	13.912
r116475	13.912
r116476	14.596
r116477	14.597
r116478	14.597

xxx TABLE:ev19_jmky_allchecks xxxxxxxxxxxxxxxxxx

c1	c2	c3	c4	c43457	c43458	c43459	c43460

r1	-186.55	-179.12	-171.74	-172.54	0	0	0	0
r2	-186.07	-178.67	-171.33	-172.13	0	0	0	0
r3	-185.59	-178.22	-170.92	-171.72	0	0	0	0

xxx TABLE:ev19_jmky_allchecks_posmass xxxxxxxxxxxxxxxxxx

	c1
r1	0.0053606
r2	0.0053744
r3	0.0053882
r4	0.0055828
r5	0.0055969
r116474	0.97866
r116475	0.97868
r116476	1.0716
r116477	1.0716
r116478	1.0716

Chapter 11

Taxes and Stimulus

11.1 Compute for Equilibrium Tax

Taking advantage of **snw_find_tax_rate** from the **PrjOptiSNW Package**, this function solves for equilibrium tax rate.

11.1.1 Parameter Controls

```
clear all;
mp_params = snw_mp_param('default_docdense');
% mp_params = snw_mp_param('default_dense');
% mp_params = snw_mp_param('default_base');
% mp_params = snw_mp_param('default_small');
mp_params('beta') = 0.95;
xi=0.651; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=
b=1; % Unemployment insurance replacement rate (b=0 refers to no UI benefits; b=1 refers to 100 perc
mp_params('xi') = xi;
mp_params('b') = b;
mp_controls = snw_mp_control('default_test');
```

Parameters for COVID related Costs:

```
% Average check per household, given COVID actual policy payment schedule
% And given distribution. The number is from averaging over the actual
% allocations given distribution.
Covid_checks_per_capita = 18.7255856*100/62502;
% Covid_checks_per_capita = 0;
% which tax parameter to change a2 is the deafult, a0 shifts max tax rate
bl_adjust_a0 = false;
bl_load_existing = false;
```

Graph Controls etc:

```
mp_controls('bl_timer') = true;
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_find_tax_rate') = true;
mp_controls('bl_print_find_tax_rate_verbose') = true;
```

11.1.2 Solve for New Tax Rate

Solve for Equilibrium Tax rate that clears government costs and income. In the extreme bounding exercise, we assume the government will pay COVID costs all in one year. This is to test whether an extreme tax scenario will lead to changes in allocation results.

Given the checks that the government hands out and the taxes imposed, individual resources post-tax are different in 2020. Households' savings decisions in 2020 vary with taxes and checks. However, the policy function post 2020 shifts back to thte previous non-COVID world's policy function because the COVID shock is an one period surprise shock.

```
a2 = snw_find_tax_rate(mp_params, mp_controls, Covid_checks_per_capita, bl_adjust_a0, bl_load_existi
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=315.
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1056.1907
Wage quintile cutoffs=0.4645     0.71528      1.0335      1.5632
Y_inc_agg=61.7582
A_agg=119.2022
Y_inc_agg_per_capita_1=1.3486
A_per_capita=2.6031
Gov_cons_per_capita=0.23703
Covid_checks_share_of_GDP=0.022215
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:1 of 83, time-this-age:0.40805
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:2 of 83, time-this-age:0.33319
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:3 of 83, time-this-age:0.41771
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:4 of 83, time-this-age:0.34531
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:5 of 83, time-this-age:0.33489
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:6 of 83, time-this-age:0.35842
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:7 of 83, time-this-age:0.35655
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:8 of 83, time-this-age:0.35122
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:9 of 83, time-this-age:0.33076
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:10 of 83, time-this-age:0.33222
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:11 of 83, time-this-age:0.32783
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:12 of 83, time-this-age:0.32957
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:13 of 83, time-this-age:0.33172
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:14 of 83, time-this-age:0.37079
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:15 of 83, time-this-age:0.36121
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:16 of 83, time-this-age:0.34383
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:17 of 83, time-this-age:0.32866
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:18 of 83, time-this-age:0.31896
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:19 of 83, time-this-age:0.32354
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:20 of 83, time-this-age:0.35757
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:21 of 83, time-this-age:0.35162
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:22 of 83, time-this-age:0.33998
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:23 of 83, time-this-age:0.32237
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:24 of 83, time-this-age:0.32174
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:25 of 83, time-this-age:0.38317
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:26 of 83, time-this-age:0.342
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:27 of 83, time-this-age:0.32267
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:28 of 83, time-this-age:0.34224
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:29 of 83, time-this-age:0.34442
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:30 of 83, time-this-age:0.35185
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:31 of 83, time-this-age:0.32489
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:32 of 83, time-this-age:0.32887
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:33 of 83, time-this-age:0.33332
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:34 of 83, time-this-age:0.32678
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:35 of 83, time-this-age:0.37362
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:36 of 83, time-this-age:0.38589
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:37 of 83, time-this-age:0.35322
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:38 of 83, time-this-age:0.32684
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:39 of 83, time-this-age:0.34446
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:40 of 83, time-this-age:0.40304
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:41 of 83, time-this-age:0.36362
```

```
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:42 of 83, time-this-age:0.32294
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:43 of 83, time-this-age:0.34509
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:44 of 83, time-this-age:0.32763
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:45 of 83, time-this-age:0.32328
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:46 of 83, time-this-age:0.3229
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:47 of 83, time-this-age:0.32421
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:48 of 83, time-this-age:0.36399
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:49 of 83, time-this-age:0.37545
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:50 of 83, time-this-age:0.37864
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:51 of 83, time-this-age:0.3443
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:52 of 83, time-this-age:0.3398
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:53 of 83, time-this-age:0.34794
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:54 of 83, time-this-age:0.36094
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:55 of 83, time-this-age:0.34198
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:56 of 83, time-this-age:0.36405
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:57 of 83, time-this-age:0.34572
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:58 of 83, time-this-age:0.34234
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:59 of 83, time-this-age:0.34115
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:60 of 83, time-this-age:0.36025
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:61 of 83, time-this-age:0.34223
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:62 of 83, time-this-age:0.34256
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:63 of 83, time-this-age:0.34432
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:64 of 83, time-this-age:0.34369
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:65 of 83, time-this-age:0.34474
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:66 of 83, time-this-age:0.36666
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:67 of 83, time-this-age:0.34095
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:68 of 83, time-this-age:0.34424
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:69 of 83, time-this-age:0.34722
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:70 of 83, time-this-age:0.34031
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:71 of 83, time-this-age:0.35799
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:72 of 83, time-this-age:0.34436
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:73 of 83, time-this-age:0.34066
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:74 of 83, time-this-age:0.34077
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:75 of 83, time-this-age:0.34236
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:76 of 83, time-this-age:0.34346
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:77 of 83, time-this-age:0.37815
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:78 of 83, time-this-age:0.3403
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:79 of 83, time-this-age:0.34125
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:80 of 83, time-this-age:0.34101
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:81 of 83, time-this-age:0.33996
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:82 of 83, time-this-age:0.3415
SNW_FIND_TAX_RATE: Aggregation, Finished Age Group:83 of 83, time-this-age:0.36096
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=1.8141;a0=0.258;err=0.20414;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=2.0964;a0=0.258;err=0.17541;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=2.3751;a0=0.258;err=0.15331;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=2.65;a0=0.258;err=0.13583;fl_total_costs=16.3562;Tax_r
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=2.9208;a0=0.258;err=0.12169;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=3.1877;a0=0.258;err=0.11003;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=3.4506;a0=0.258;err=0.10028;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=3.7096;a0=0.258;err=0.091993;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=3.9648;a0=0.258;err=0.084884;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=4.2162;a0=0.258;err=0.078718;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=4.464;a0=0.258;err=0.073324;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=4.7083;a0=0.258;err=0.068568;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=4.9491;a0=0.258;err=0.064345;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=5.1866;a0=0.258;err=0.060571;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=5.4207;a0=0.258;err=0.05718;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=5.6517;a0=0.258;err=0.054118;fl_total_costs=16.3562;Ta
```

```
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=5.8796;a0=0.258;err=0.051339;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=6.1044;a0=0.258;err=0.048807;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=6.3263;a0=0.258;err=0.046491;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=6.5454;a0=0.258;err=0.044365;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=6.7616;a0=0.258;err=0.042406;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=6.975;a0=0.258;err=0.040597;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=7.1858;a0=0.258;err=0.03892;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=7.394;a0=0.258;err=0.037362;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=7.5996;a0=0.258;err=0.035911;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=7.8027;a0=0.258;err=0.034557;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=8.0034;a0=0.258;err=0.03329;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=8.2017;a0=0.258;err=0.032102;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=8.3976;a0=0.258;err=0.030987;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=8.5912;a0=0.258;err=0.029938;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=8.7826;a0=0.258;err=0.028948;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=8.9718;a0=0.258;err=0.028015;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=9.1588;a0=0.258;err=0.027132;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=9.3437;a0=0.258;err=0.026296;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=9.5265;a0=0.258;err=0.025504;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=9.7073;a0=0.258;err=0.024751;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=9.886;a0=0.258;err=0.024036;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=10.0628;a0=0.258;err=0.023355;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=10.2377;a0=0.258;err=0.022707;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=10.4106;a0=0.258;err=0.022089;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=10.5817;a0=0.258;err=0.021498;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=10.7509;a0=0.258;err=0.020934;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=10.9184;a0=0.258;err=0.020394;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=11.084;a0=0.258;err=0.019877;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=11.2479;a0=0.258;err=0.019382;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=11.4101;a0=0.258;err=0.018907;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=11.5706;a0=0.258;err=0.018451;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=11.7294;a0=0.258;err=0.018013;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=11.8866;a0=0.258;err=0.017592;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=12.0421;a0=0.258;err=0.017187;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=12.1961;a0=0.258;err=0.016797;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=12.3485;a0=0.258;err=0.016421;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=12.4994;a0=0.258;err=0.016059;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=12.6487;a0=0.258;err=0.01571;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=12.7965;a0=0.258;err=0.015373;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=12.9429;a0=0.258;err=0.015048;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=13.0877;a0=0.258;err=0.014733;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=13.2312;a0=0.258;err=0.014429;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=13.3732;a0=0.258;err=0.014135;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=13.5139;a0=0.258;err=0.013851;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=13.6531;a0=0.258;err=0.013576;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=13.791;a0=0.258;err=0.013309;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=13.9275;a0=0.258;err=0.01305;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=14.0628;a0=0.258;err=0.0128;fl_total_costs=16.3562;Tax
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=14.1967;a0=0.258;err=0.012557;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=14.3293;a0=0.258;err=0.012321;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=14.4606;a0=0.258;err=0.012092;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=14.5907;a0=0.258;err=0.01187;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=14.7195;a0=0.258;err=0.011654;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=14.8472;a0=0.258;err=0.011444;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=14.9736;a0=0.258;err=0.01124;fl_total_costs=16.3562;Ta
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=15.0988;a0=0.258;err=0.011041;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=15.2228;a0=0.258;err=0.010848;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=15.3456;a0=0.258;err=0.01066;fl_total_costs=16.3562;Ta
```

```
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=15.4673;a0=0.258;err=0.010477;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=15.5879;a0=0.258;err=0.010299;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=15.7073;a0=0.258;err=0.010126;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=15.8257;a0=0.258;err=0.0099566;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=15.9429;a0=0.258;err=0.0097918;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=16.059;a0=0.258;err=0.0096311;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=16.1741;a0=0.258;err=0.0094744;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=16.2881;a0=0.258;err=0.0093215;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=16.4011;a0=0.258;err=0.0091723;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=16.513;a0=0.258;err=0.0090267;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=16.6239;a0=0.258;err=0.0088845;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=16.7338;a0=0.258;err=0.0087457;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=16.8426;a0=0.258;err=0.0086101;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=16.9505;a0=0.258;err=0.0084776;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.0574;a0=0.258;err=0.0083482;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.1634;a0=0.258;err=0.0082216;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.2684;a0=0.258;err=0.0080979;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.3724;a0=0.258;err=0.0079769;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.4755;a0=0.258;err=0.0078586;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.5777;a0=0.258;err=0.0077428;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.6789;a0=0.258;err=0.0076295;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.7793;a0=0.258;err=0.0075186;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.8787;a0=0.258;err=0.0074101;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=17.9773;a0=0.258;err=0.0073038;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=18.075;a0=0.258;err=0.0071997;fl_total_costs=16.3562;T
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=18.1718;a0=0.258;err=0.0070978;fl_total_costs=16.3562;
SNW_FIND_TAX_RATE tax a2 or a0 adjustments;a2=18.2678;a0=0.258;err=0.006998;fl_total_costs=16.3562;T
--------------------------------
--- SNW_FIND_TAX_RATE finished -
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
SNW_FIND_TAX_RATE: Number of a2-adjustments (for taxation) used to balance the government budget= 10
SNW_FIND_TAX_RATE info A;a2_new=18.2678;a2_guess_orig(mana-heaven)=1.5286;a0_new=0.258;a0_guess_orig
SNW_FIND_TAX_RATE info B;Y_inc_agg=61.7582;Y_inc_agg_COVID=58.8608;GPD_COVID_CHANGE=0.046915;Y_inc_a
SNW_FIND_TAX_RATE info C;Covid_checks_per_capita=0.02996;Covid_checks_share_of_GDP=0.022215
SNW_FIND_TAX_RATE info D;UI_benefits=1.9702;SS_spend=2.1597;Gov_cons=10.8544;Covid_checks=1.372;Tax_
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
--------------------------------
```

11.2 Existing Stimulus as a Function of Income and Family Status

Taking advantage of **snw_stimulus_checks** from the **PrjOptiSNW Package**, this function presents stimulus checks at different income levels for households with different children count and martial status. The function considers the first as well as the second stimulus check.

11.2.1 Trump Stimulus Checks for Unmarried Households

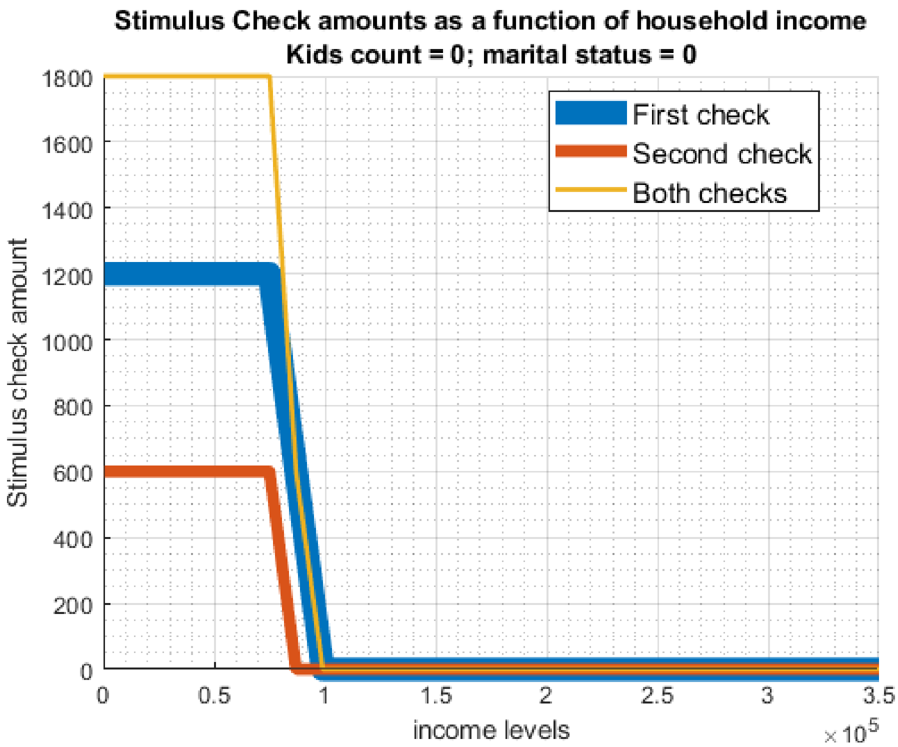
Check base amount per adult and per child for the first and second rounds.

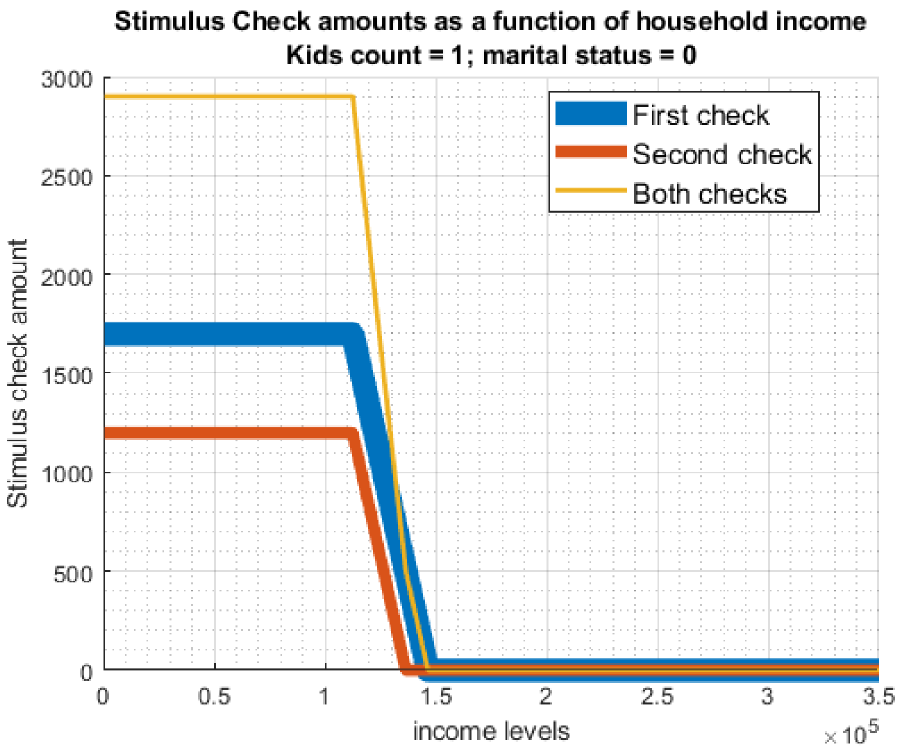
```
[fl_stimulus_adult_first, fl_stimulus_child_first] = deal(1200, 500);
[fl_stimulus_adult_second, fl_stimulus_child_second] = deal(600, 600);
bl_visualize = true;
```

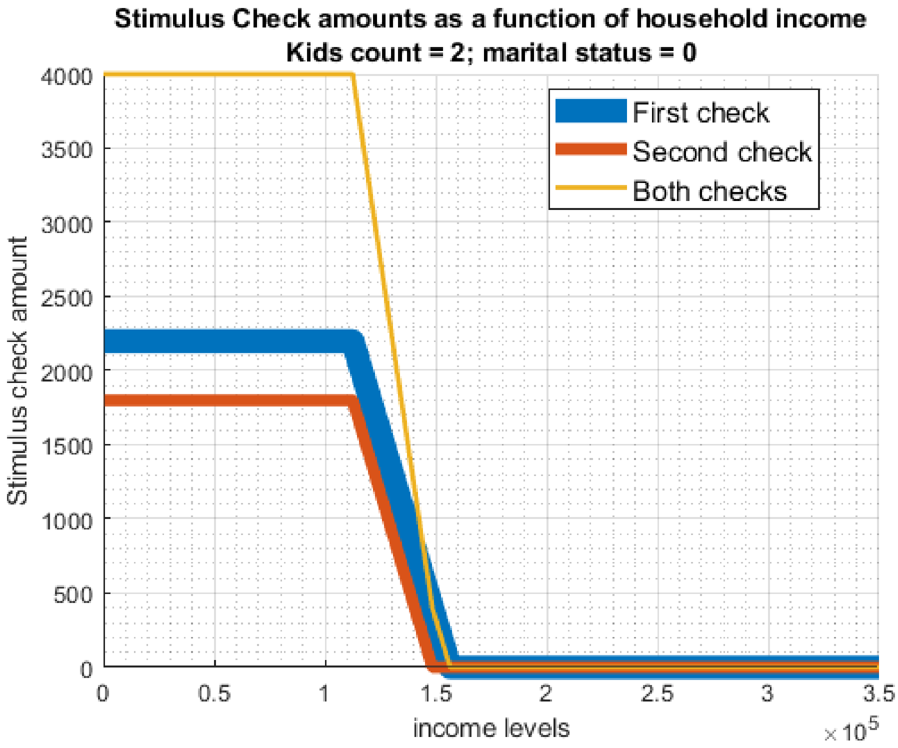
Visualize stimulus check amounts.

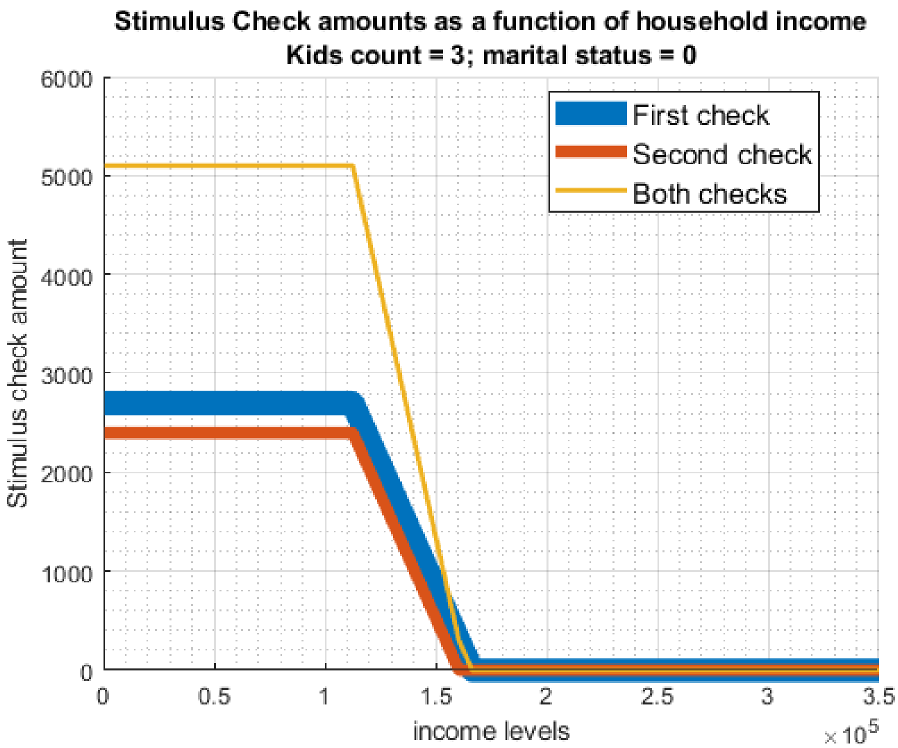
```
bl_marital = 0;
for it_kids=0:1:4
```

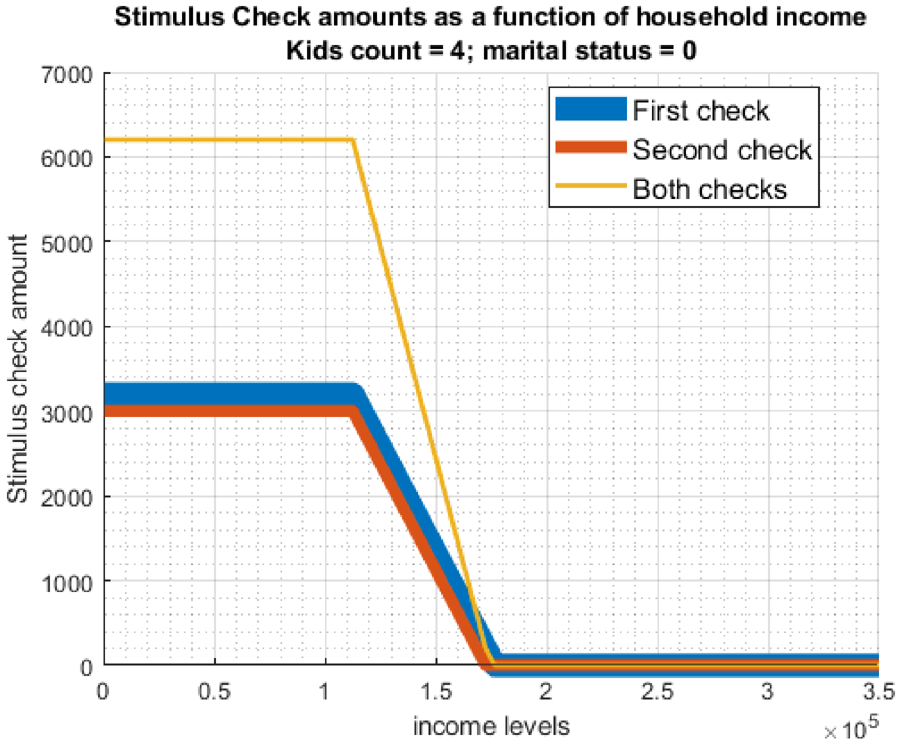
```
    snw_stimulus_checks(it_kids, bl_marital, ...
        fl_stimulus_adult_first, fl_stimulus_child_first, ...
        fl_stimulus_adult_second, fl_stimulus_child_second, ...
        bl_visualize);
end
```






Stimulus Check amounts as a function of household income
Kids count = 2; marital status = 0
First check
Second check
Both checks
Stimulus check amount
income levels
×10^5
0
500
1000
1500
2000
2500
3000
3500
4000
0.5
1
1.5
2
2.5
3
3.5

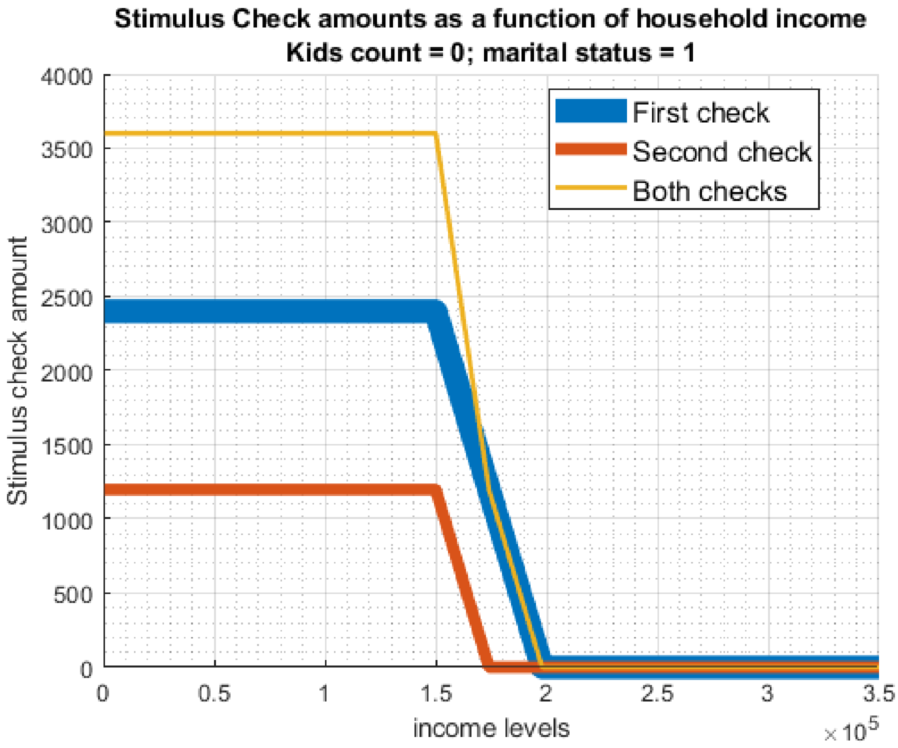

Stimulus Check amounts as a function of household income
Kids count = 3; marital status = 0
First check
Second check
Both checks
Stimulus check amount
income levels
×10^5
0
1000
2000
3000
4000
5000
6000
0.5
1
1.5
2
2.5
3
3.5

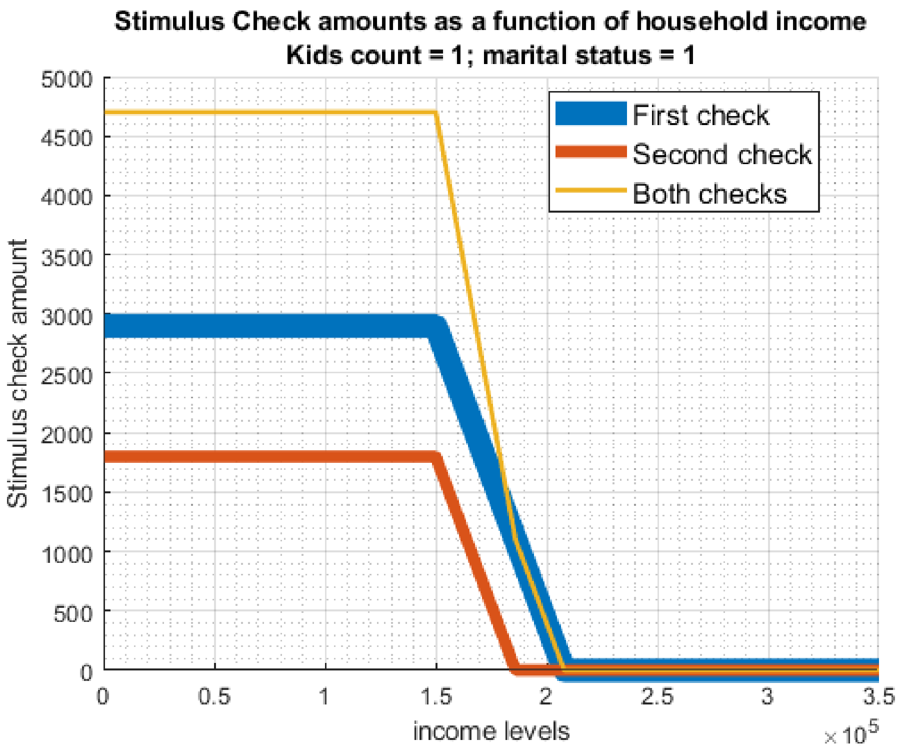


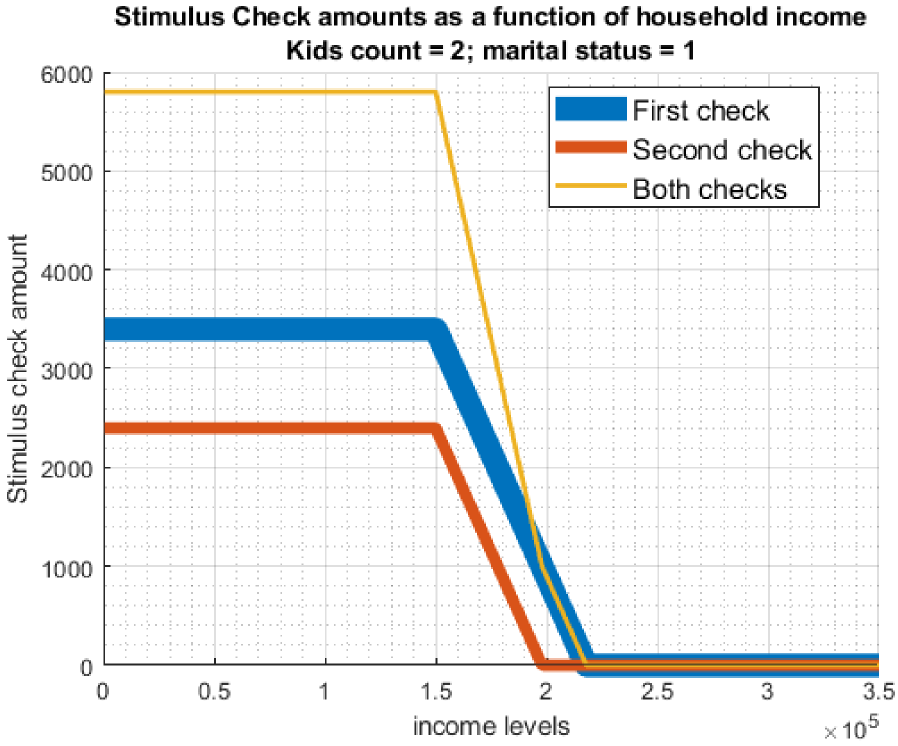
11.2.2 Trump Stimulus Checks for Married Households

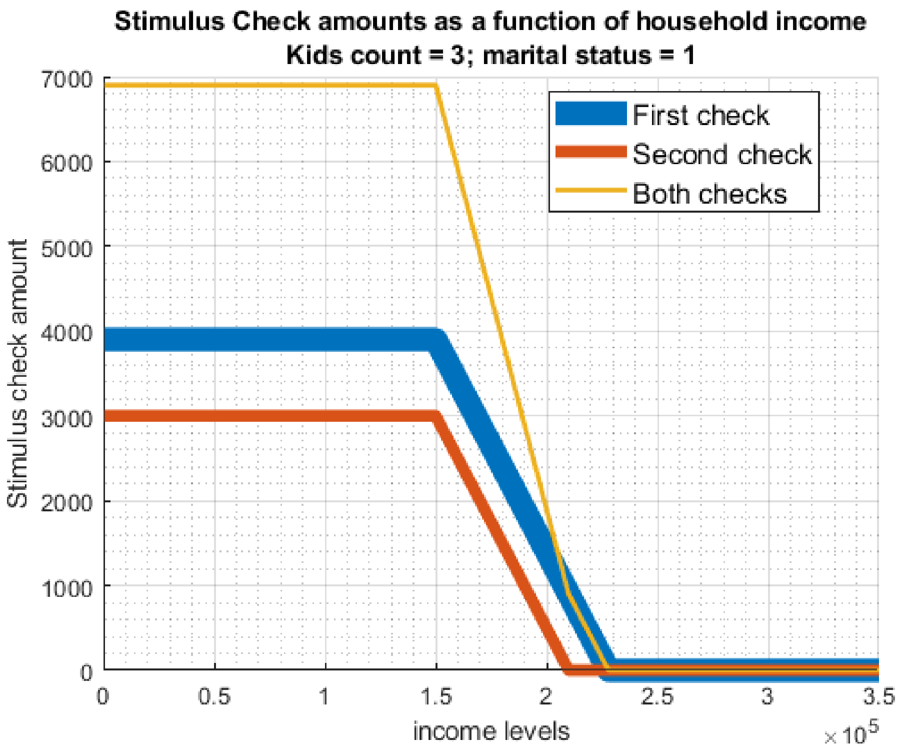
Visualize stimulus check amounts.

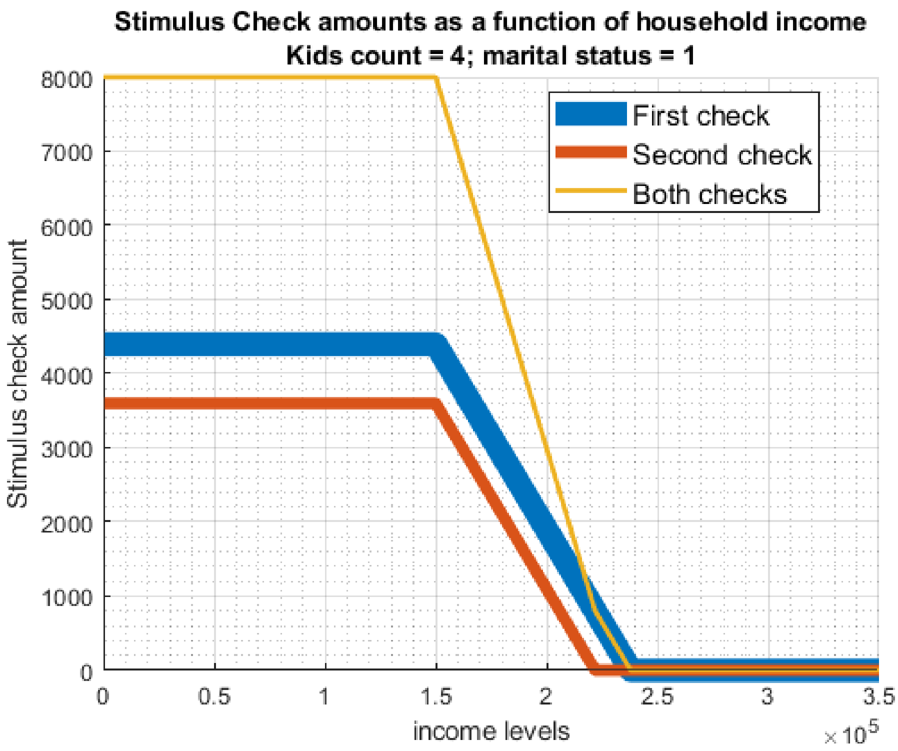
```
bl_marital = 1;
for it_kids=0:1:4
    snw_stimulus_checks(it_kids, bl_marital, ...
        fl_stimulus_adult_first, fl_stimulus_child_first, ...
        fl_stimulus_adult_second, fl_stimulus_child_second, ...
        bl_visualize);
end
```


Stimulus Check amounts as a function of household income
Kids count = 0; marital status = 1
Stimulus check amount
income levels
$\times 10^5$
First check
Second check
Both checks
0
500
1000
1500
2000
2500
3000
3500
4000
0.5
1
1.5
2
2.5
3
3.5


Stimulus Check amounts as a function of household income
Kids count = 1; marital status = 1
Stimulus check amount
income levels
$\times 10^5$
First check
Second check
Both checks
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
0.5
1
1.5
2
2.5
3
3.5


Stimulus Check amounts as a function of household income
Kids count = 2; marital status = 1
First check
Second check
Both checks
Stimulus check amount
income levels
6000
5000
4000
3000
2000
1000
0
0
0.5
1
1.5
2
2.5
3
3.5
×10^5


Stimulus Check amounts as a function of household income
Kids count = 3; marital status = 1
First check
Second check
Both checks
Stimulus check amount
income levels
7000
6000
5000
4000
3000
2000
1000
0
0
0.5
1
1.5
2
2.5
3
3.5
×10^5



11.3 Existing Stimulus as a Function of Income and Family Status

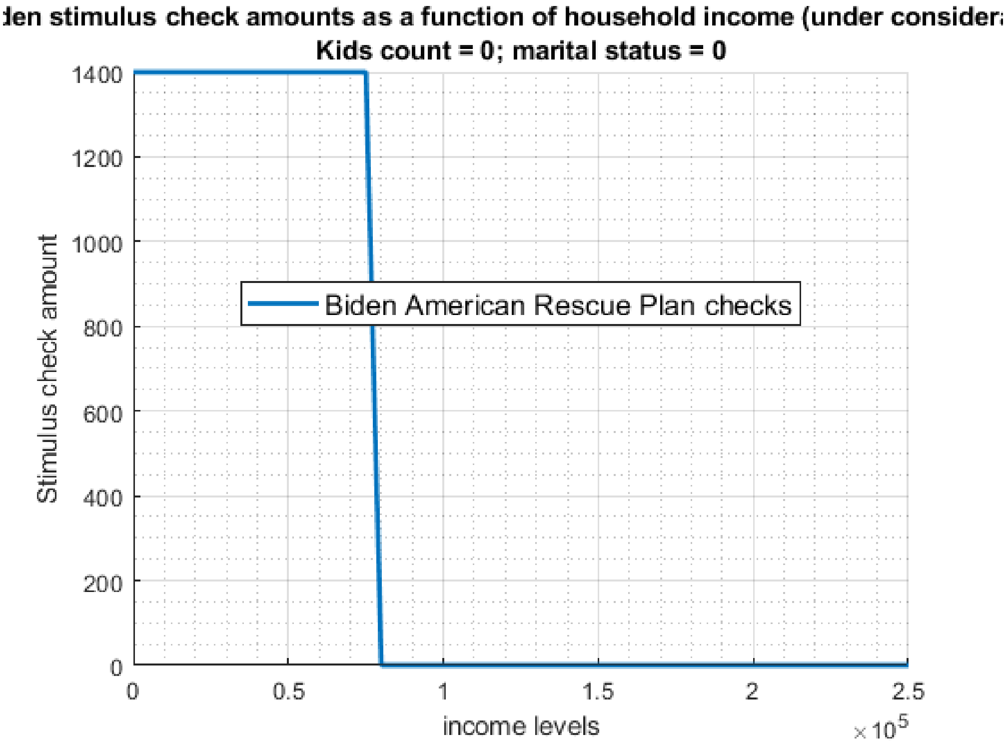
Taking advantage of **snw_stimulus_checks_biden** from the **PrjOptiSNW Package**, this function presents stimulus checks at different income levels for households with different children count and martial status. This is for the Biden Stimulus under the American Rescue Plan Act.

11.3.1 Biden Stimulus Checks for Unmarried Households

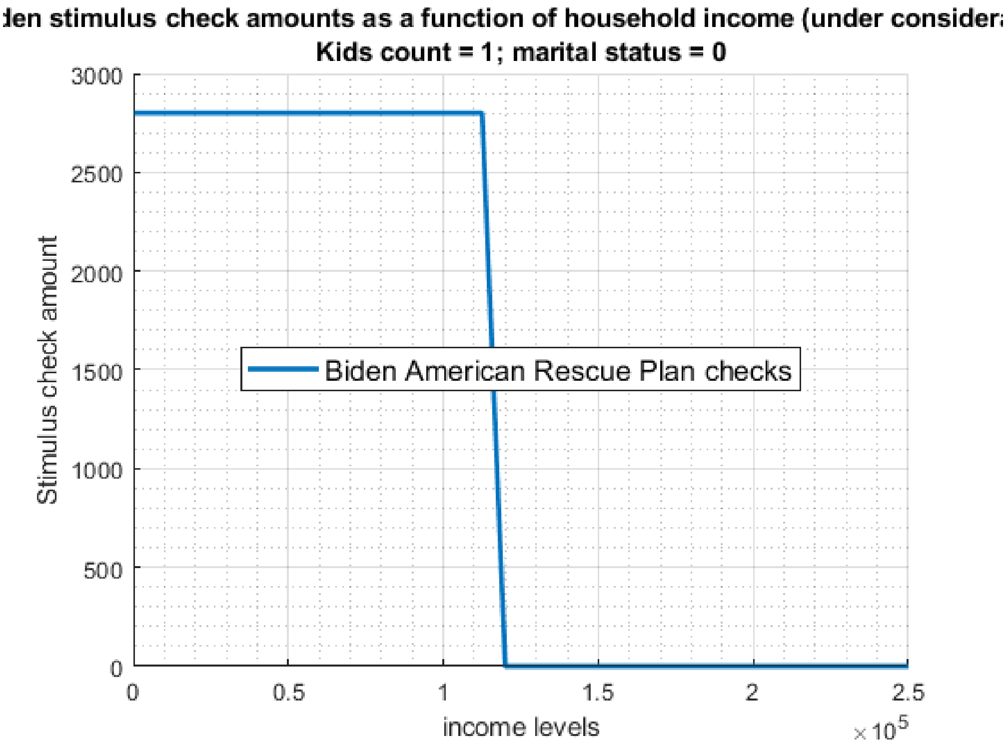
Check base amount per adult and per child for the first and second rounds.

Visualize stimulus check amounts.

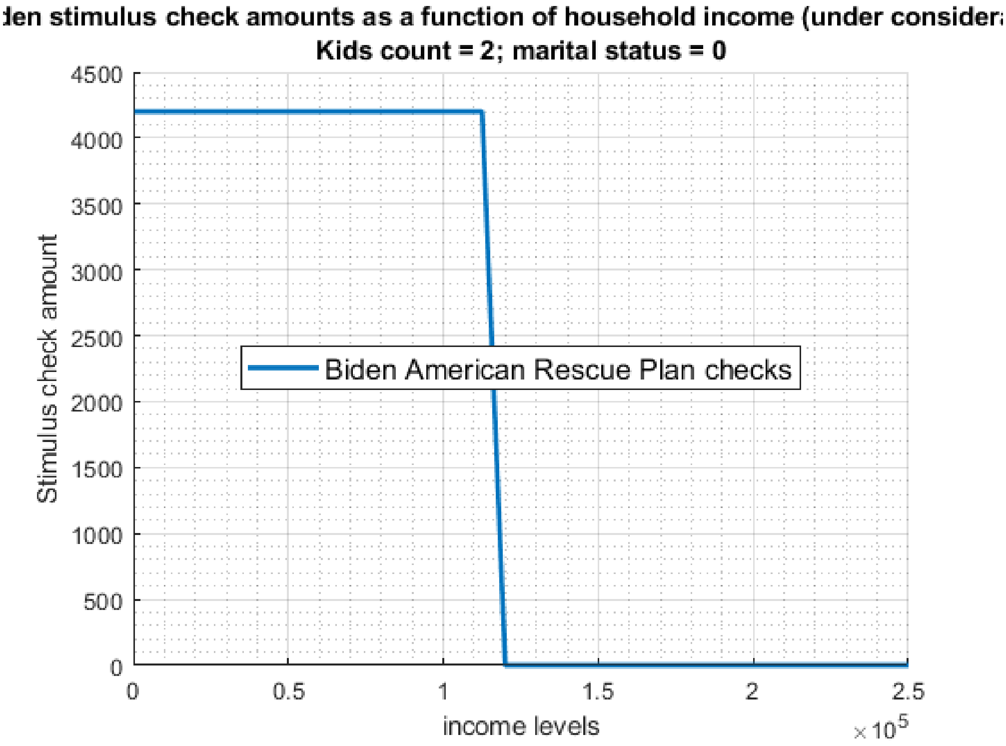
```
bl_visualize = true;
bl_marital = 0;
for it_kids=0:1:4
    snw_stimulus_checks_biden(it_kids, bl_marital, bl_visualize);
end
```



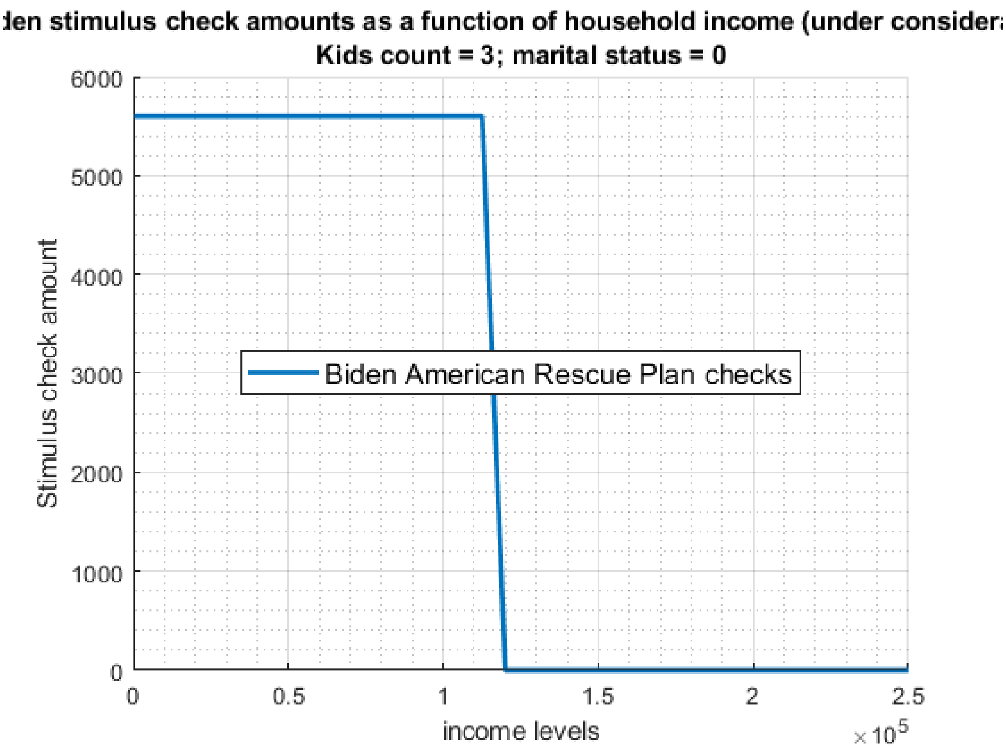
den stimulus check amounts as a function of household income (under considera
Kids count = 0; marital status = 0
Stimulus check amount
1400
1200
1000
800
600
400
200
0
Biden American Rescue Plan checks
0
0.5
1
1.5
2
2.5
income levels
×10^5



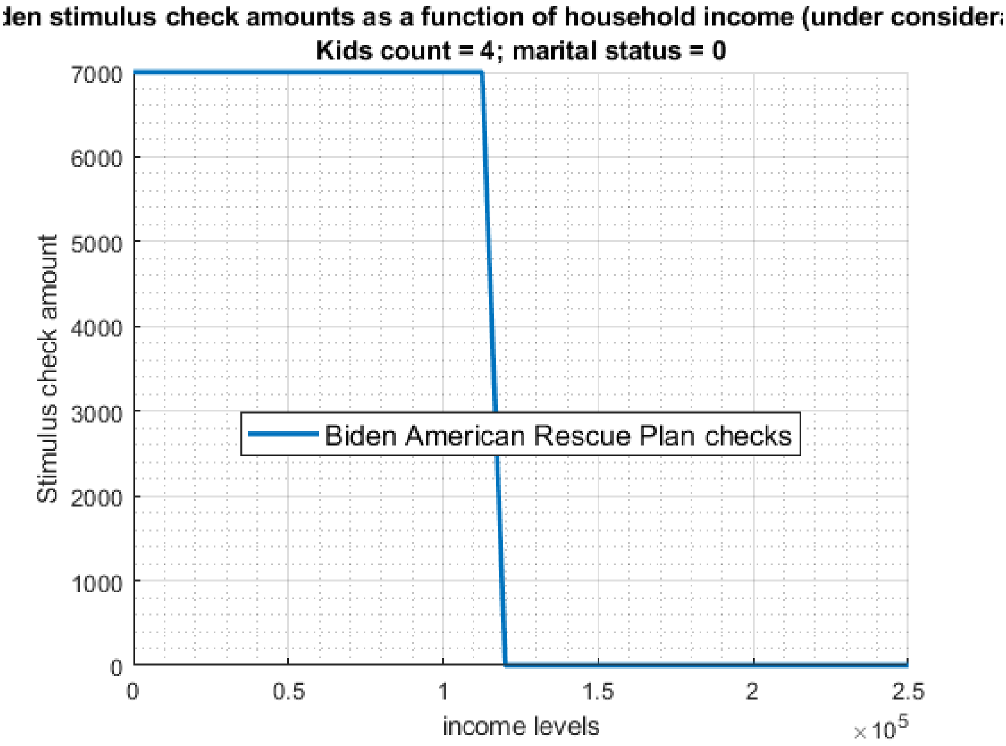
den stimulus check amounts as a function of household income (under considera
Kids count = 1; marital status = 0
Stimulus check amount
3000
2500
2000
1500
1000
500
0
Biden American Rescue Plan checks
0
0.5
1
1.5
2
2.5
income levels
×10^5



den stimulus check amounts as a function of household income (under considera
Kids count = 2; marital status = 0
Stimulus check amount
income levels
×10^5
Biden American Rescue Plan checks
0
500
1000
1500
2000
2500
3000
3500
4000
4500
0
0.5
1
1.5
2
2.5



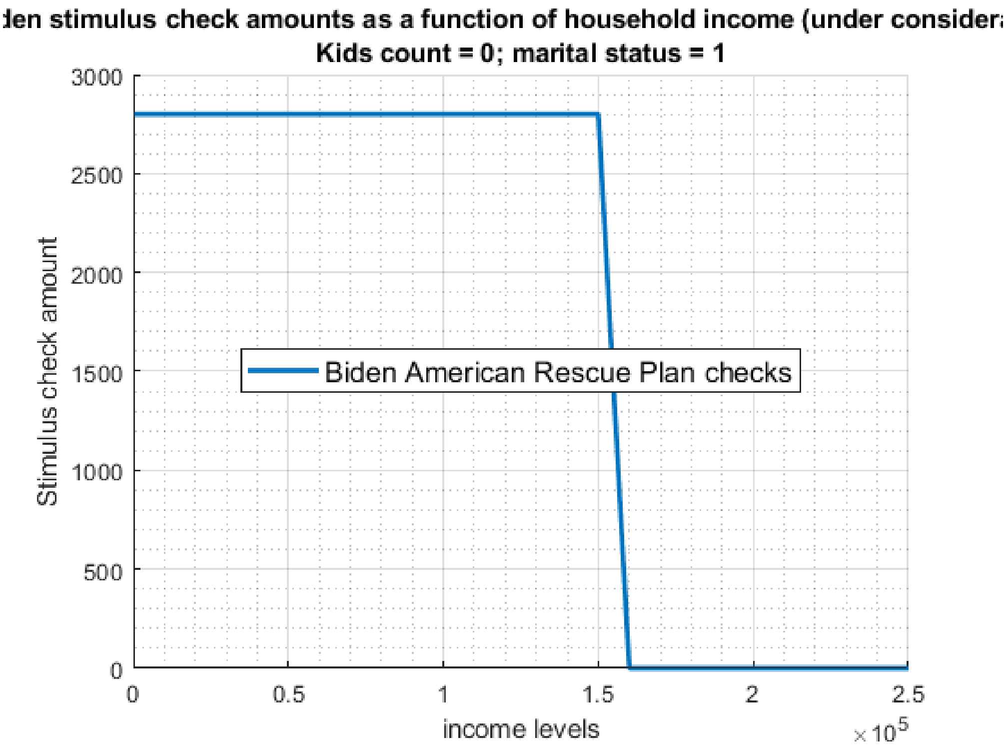
den stimulus check amounts as a function of household income (under considera
Kids count = 3; marital status = 0
Stimulus check amount
income levels
×10^5
Biden American Rescue Plan checks
0
1000
2000
3000
4000
5000
6000
0
0.5
1
1.5
2
2.5

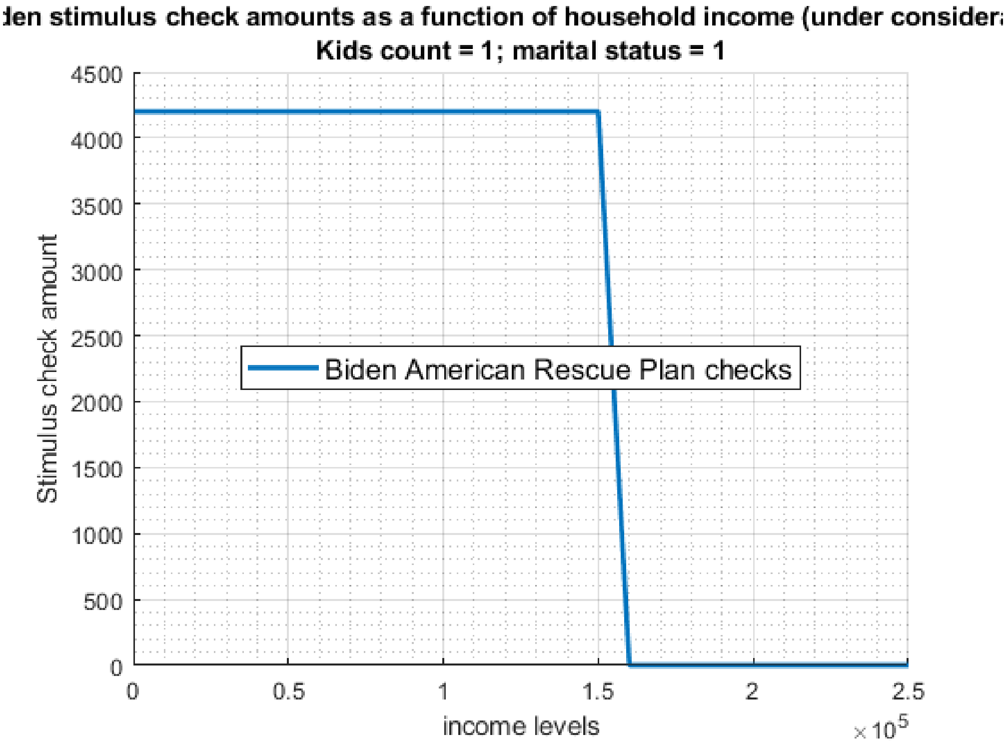


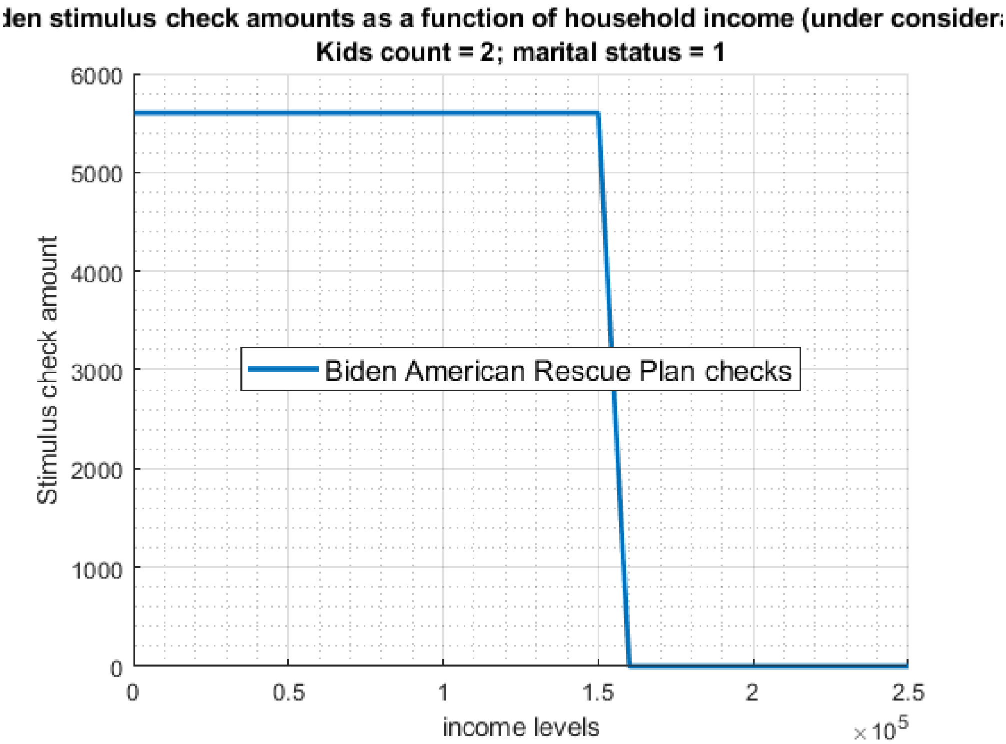
11.3.2 Biden Stimulus Checks for Married Households

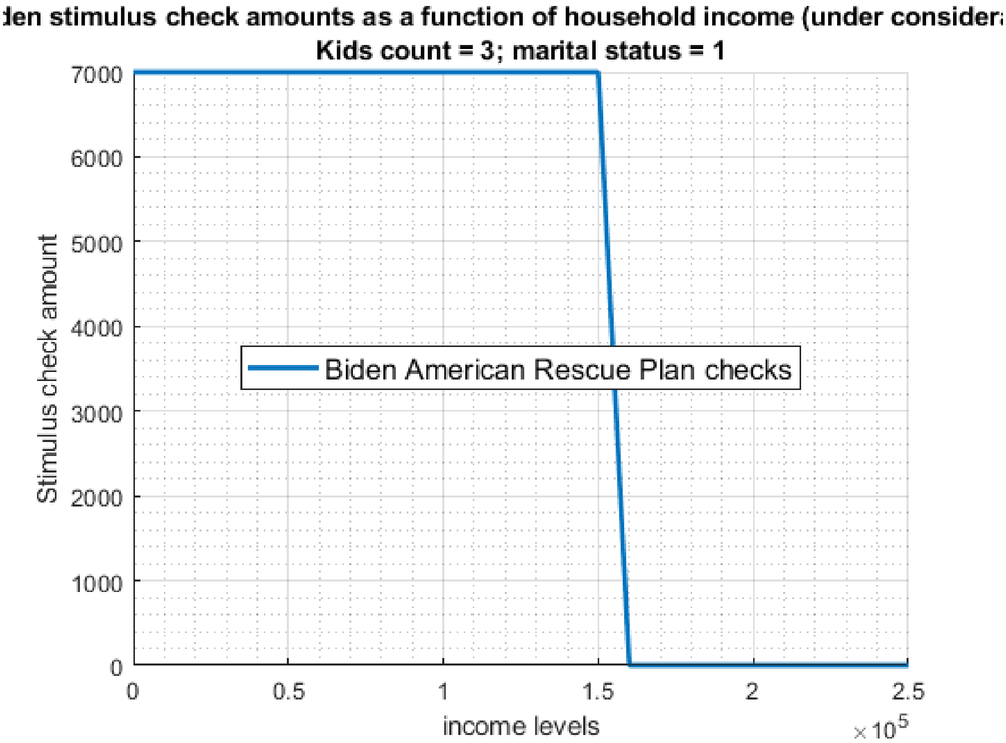
Visualize stimulus check amounts.

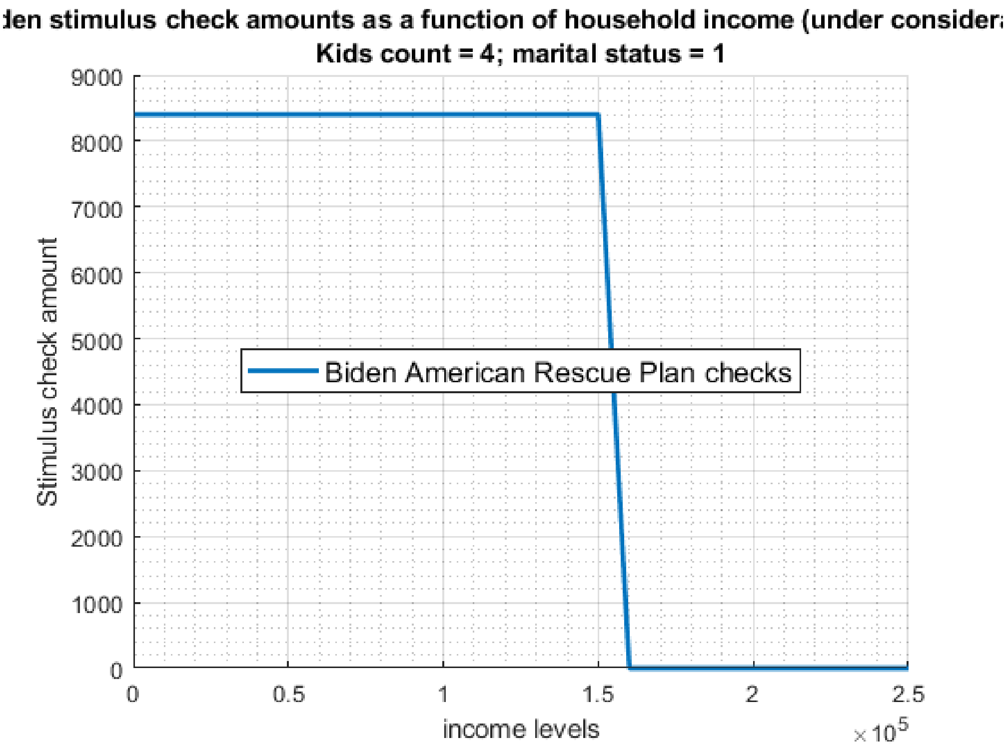
```
bl_marital = 1;
for it_kids=0:1:4
    snw_stimulus_checks_biden(it_kids, bl_marital, bl_visualize);
end
```




den stimulus check amounts as a function of household income (under consider
Kids count = 1; marital status = 1
Stimulus check amount
income levels
$\times 10^5$
Biden American Rescue Plan checks
0
500
1000
1500
2000
2500
3000
3500
4000
4500
0
0.5
1
1.5
2
2.5


den stimulus check amounts as a function of household income (under consider
Kids count = 2; marital status = 1
Stimulus check amount
income levels
$\times 10^5$
Biden American Rescue Plan checks
0
1000
2000
3000
4000
5000
6000
0
0.5
1
1.5
2
2.5





11.4 Taxable Income and Tax Liabilities in 2008

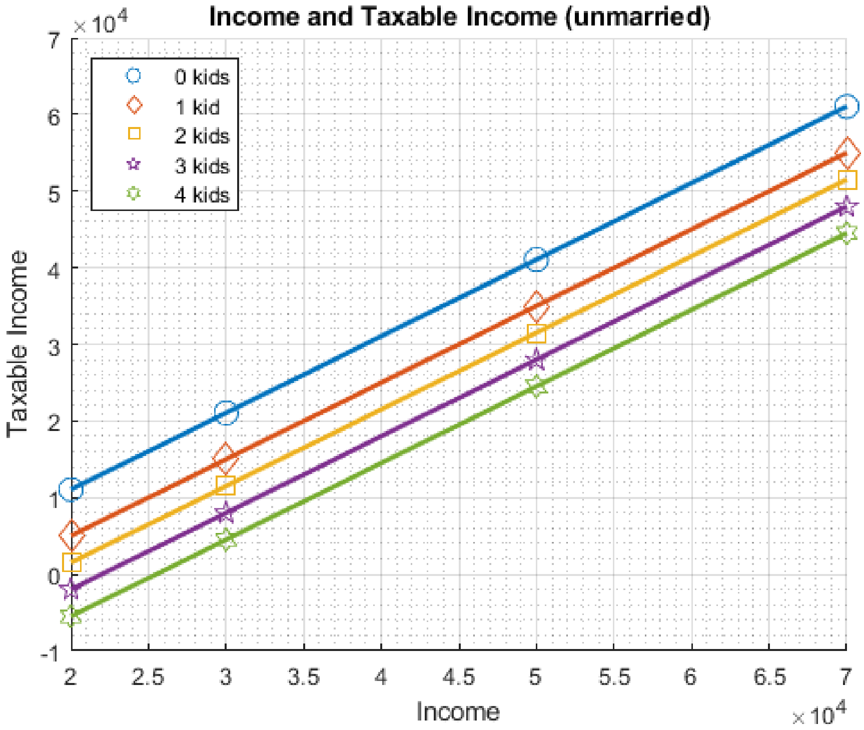
Taking advantage of **snw_tax_liability** from the **PrjOptiSNW Package**, the function solves for tax liability.

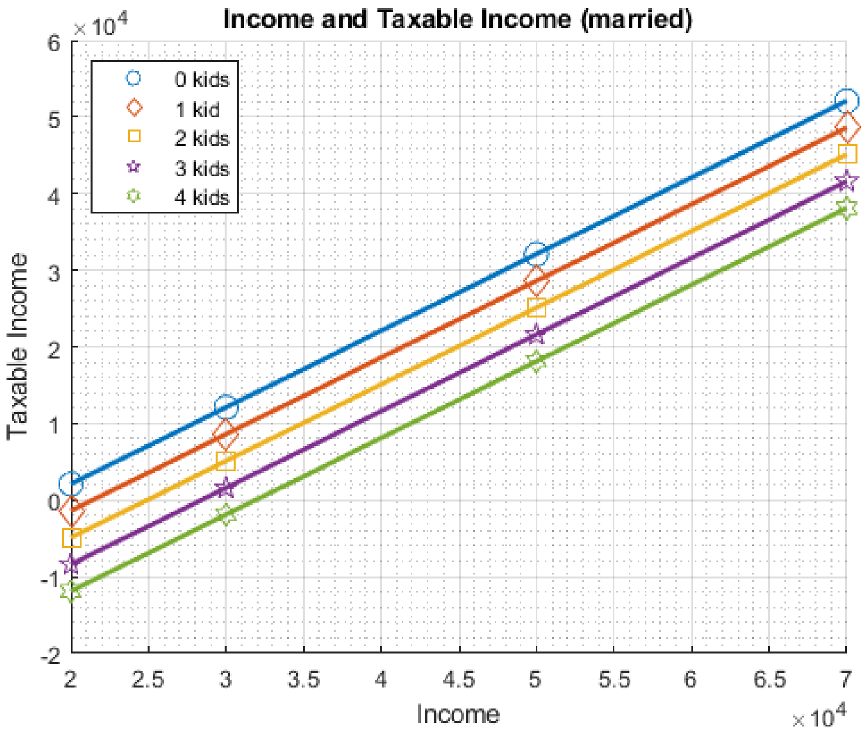
We can study the effects of the 2008 Tax Rebate. The Tax rebate is a rebate based on how much tax was paid, so we need to know taxable income and tax liability. These differ by income, household marital status, and the count of children. Given an array of pre-tax income values, we compute for from 0 to 4 kids and both married and unmarried taxable-income and tax-liability at all points along the income array. Deductions are from 2008 income (2008 IRS f1040). Tax brackets from 2008 are here (2008 IRS 1040tt).

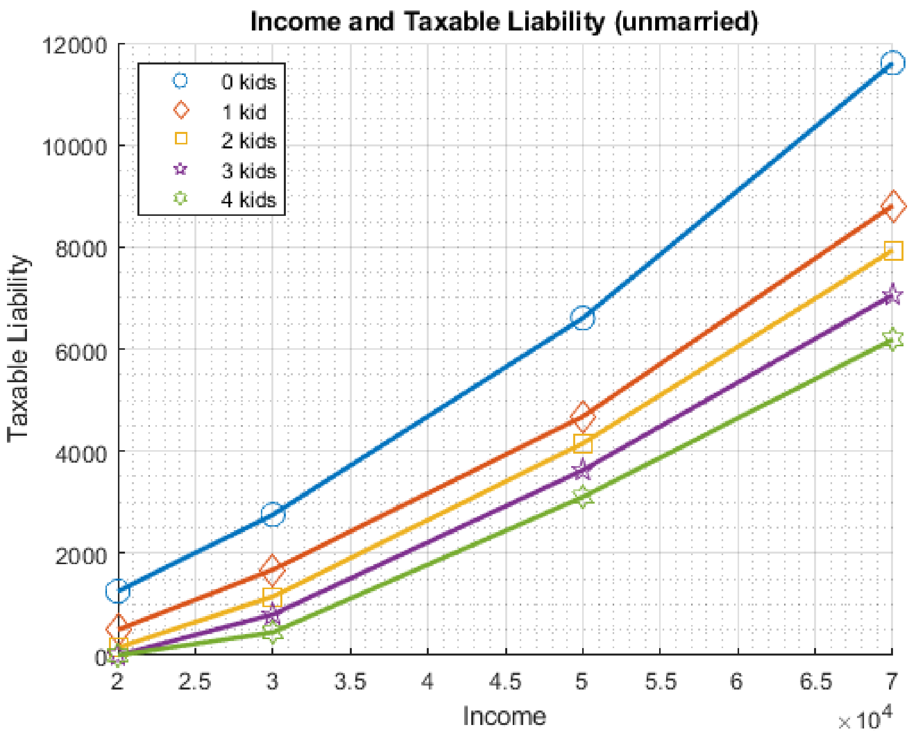
11.4.1 Taxable Income and Tax Liabilities in 2008 for 4 Income Levels

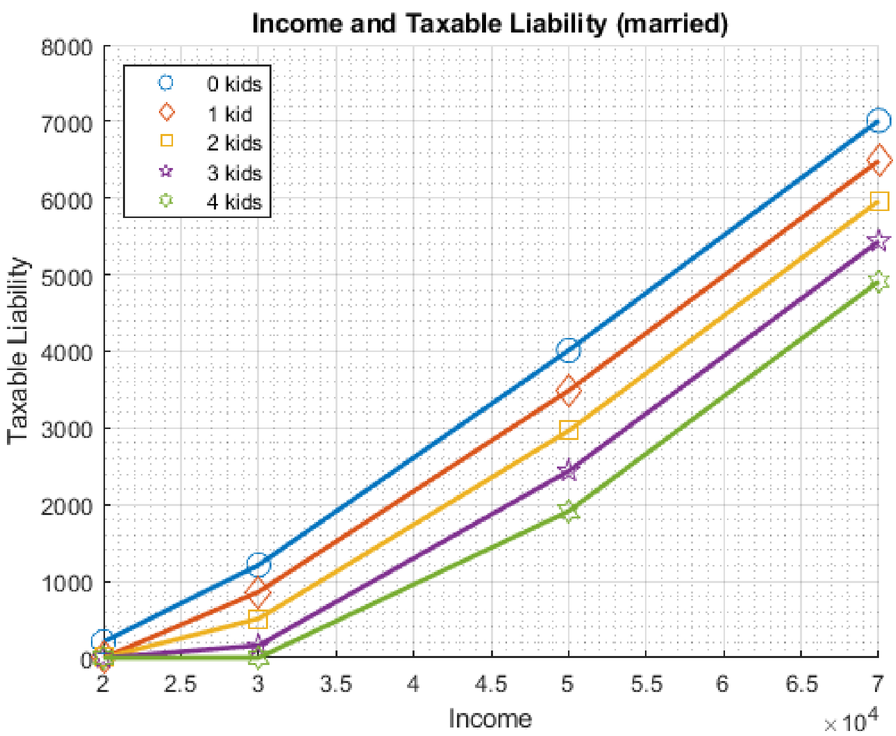
Call the function at four income levels. Solve for different kids count and by marital status.

```
bl_visualize = true;
ar_income = [20000, 30000, 50000, 70000];
snw_tax_liability(ar_income, true);
```





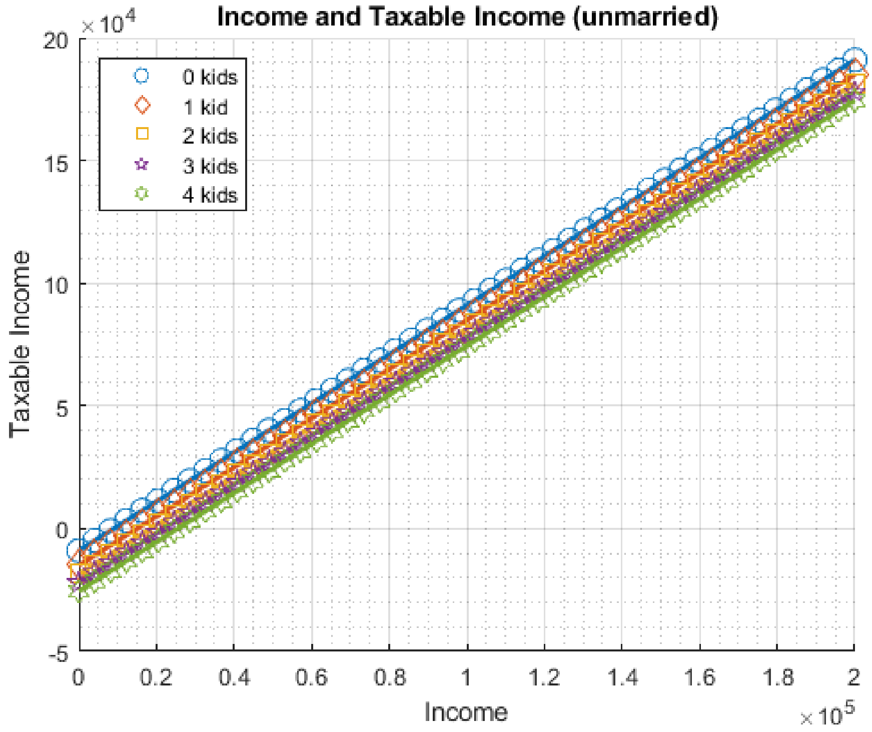


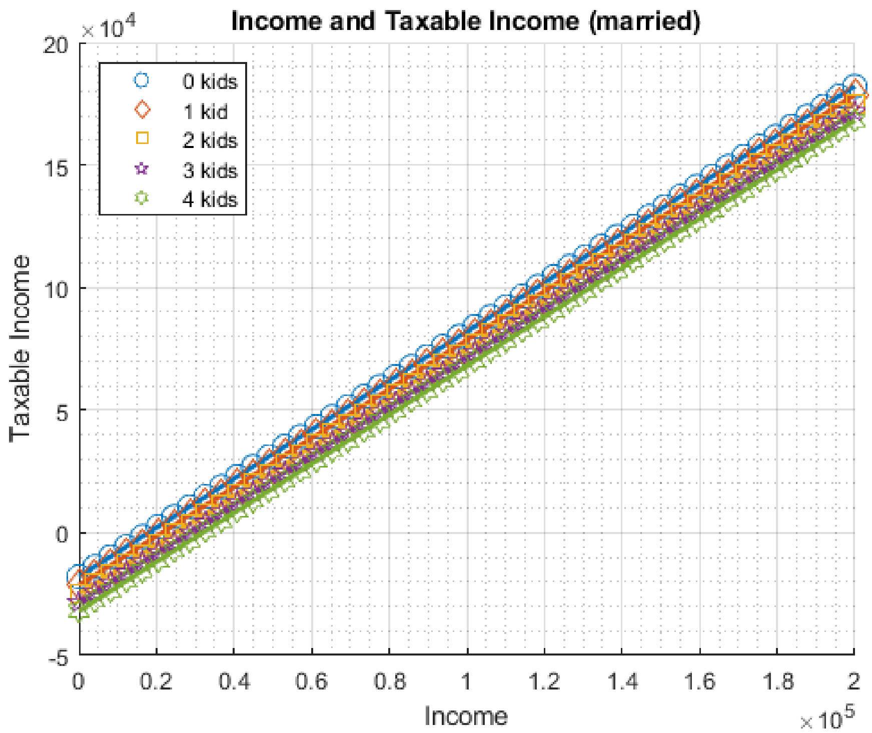


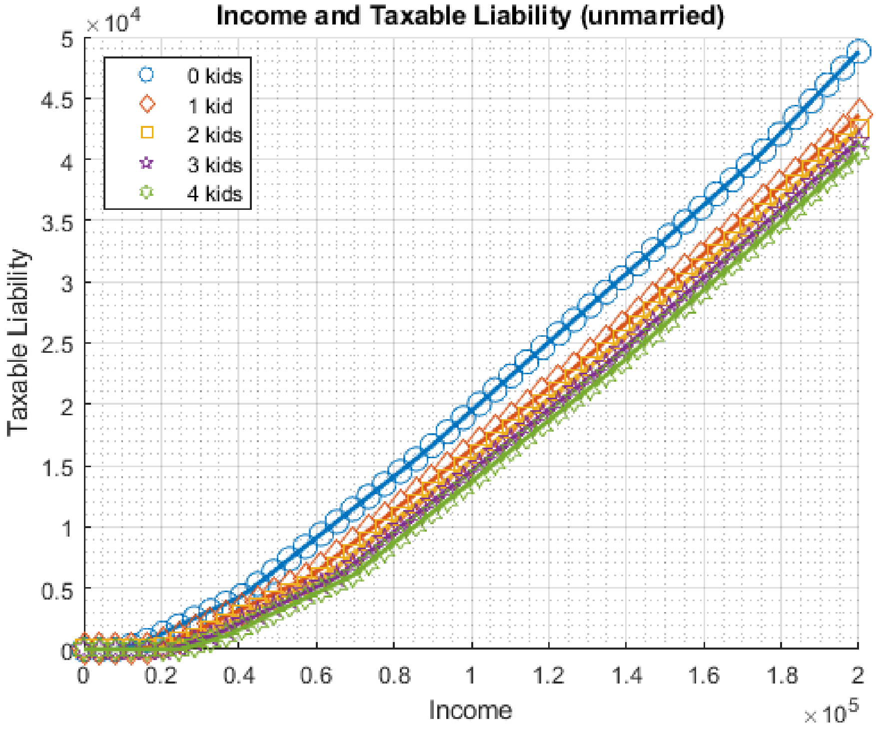
11.4.2 Taxable Income and Tax Liabilities in 2008 for 50 Income Levels

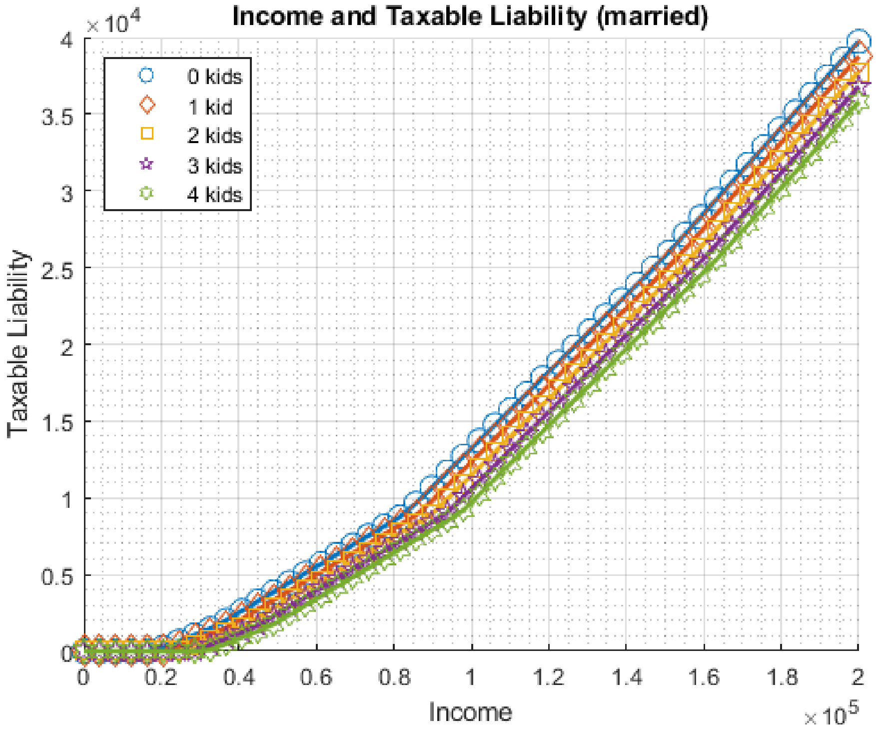
Call the function for incomes from 0 to 200k. Solve for different kids count and by marital status.

```
bl_visualize = true;
ar_income = linspace(0, 2e5, 50);
snw_tax_liability(ar_income, true);
```


Income and Taxable Income (unmarried)
×10^4
0 kids
1 kid
2 kids
3 kids
4 kids
Taxable Income
Income
×10^5
20
15
10
5
0
-5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


Income and Taxable Income (married)
×10^4
0 kids
1 kid
2 kids
3 kids
4 kids
Taxable Income
Income
×10^5
20
15
10
5
0
-5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2





11.5 Bush 2008 Stimulus as a Function of Income and Family Status

Taking advantage of **snw_stimulus_checks_bush** from the **PrjOptiSNW Package**, this function presents stimulus checks at different income levels for households with different children count and martial status. The function considers the bush stimulus checks from the Economic Stimulus Act of 2008. IRS information provides information at the Economic Stimulus Payment Information Center.

Note that the Bush stimulus is a tax rebate, so we compute tax liability based on snw_tax_liability.

The Stimulus policy is expressed in the following formula, first four components (M is marital status, equals to 1 for married, 0 for unmarried, K is the number of kids, Y is pre-tax income) :

- $\text{MinChk}(M) = 300 \cdot (1 + M)$
- $\text{CappedRebate}(Y, K, M) = \min\Big(\text{TaxLiability}\,(Y, K, M)\,, 600 \cdot (1 + M)\Big)$
- $\text{PhaseOut}(Y, M) = \max\Big(0, Y - 75000 \cdot (1 + M)\Big) \cdot 0.05$
- $\text{KidsChk}(K) = 300 \cdot K$

Overall Tax-rebate Stimulus is:

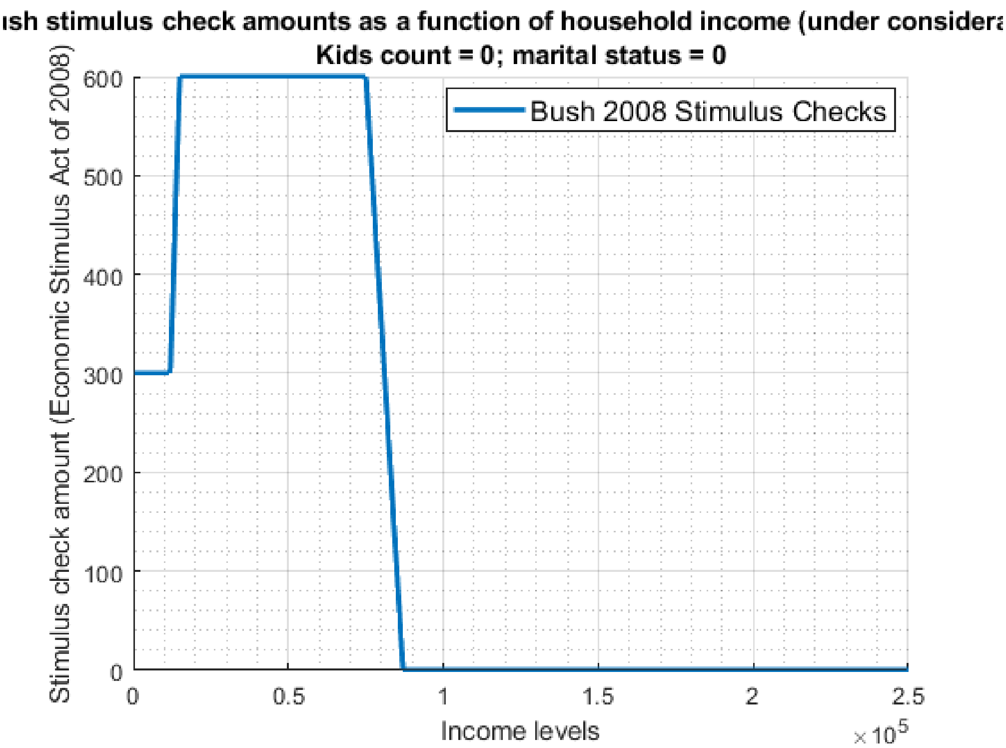
$$\text{Bush08StimulusTaxRebate} = \max\Bigg(\max\Bigg(\text{MinChk}(M), \text{CappedRebate}(Y, K, M)\Bigg) + \text{KidsChk}(K) - \text{PhaseOut}(Y, M), 0$$

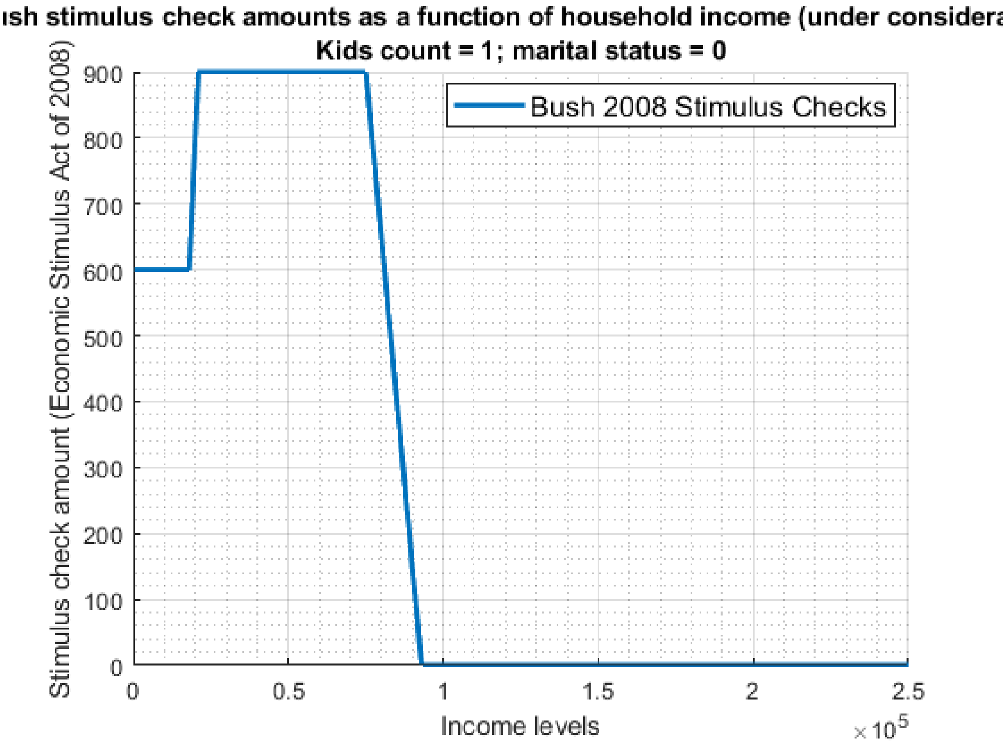
11.5.1 Bush Stimulus Checks for Unmarried Households

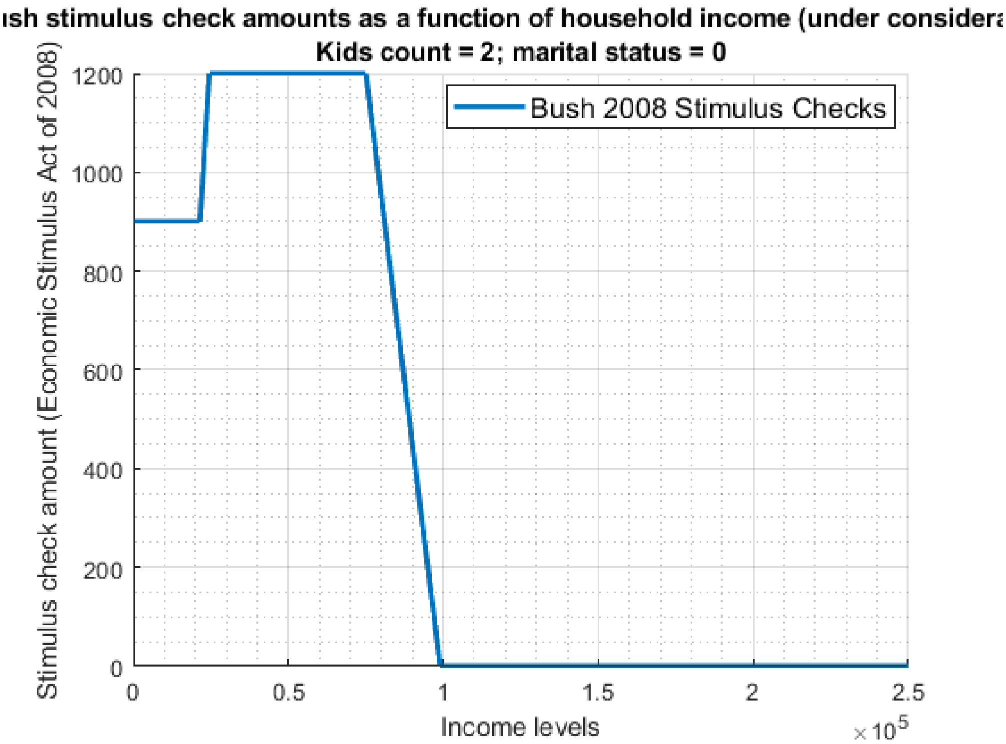
Check base amount per adult and per child for the first and second rounds.

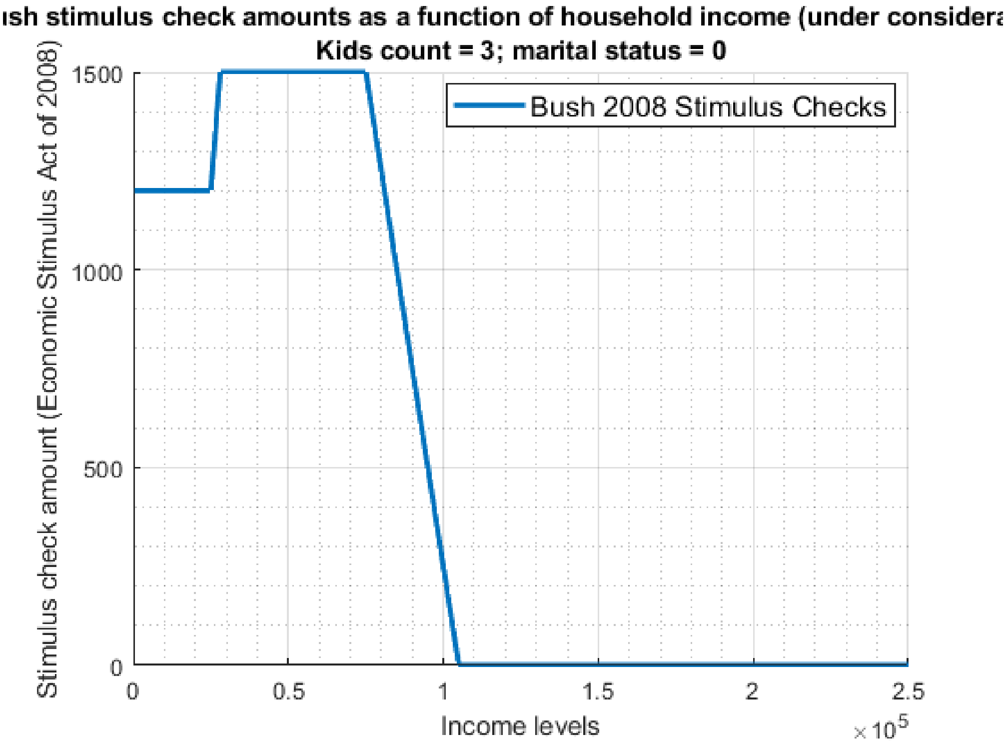
Visualize stimulus check amounts.

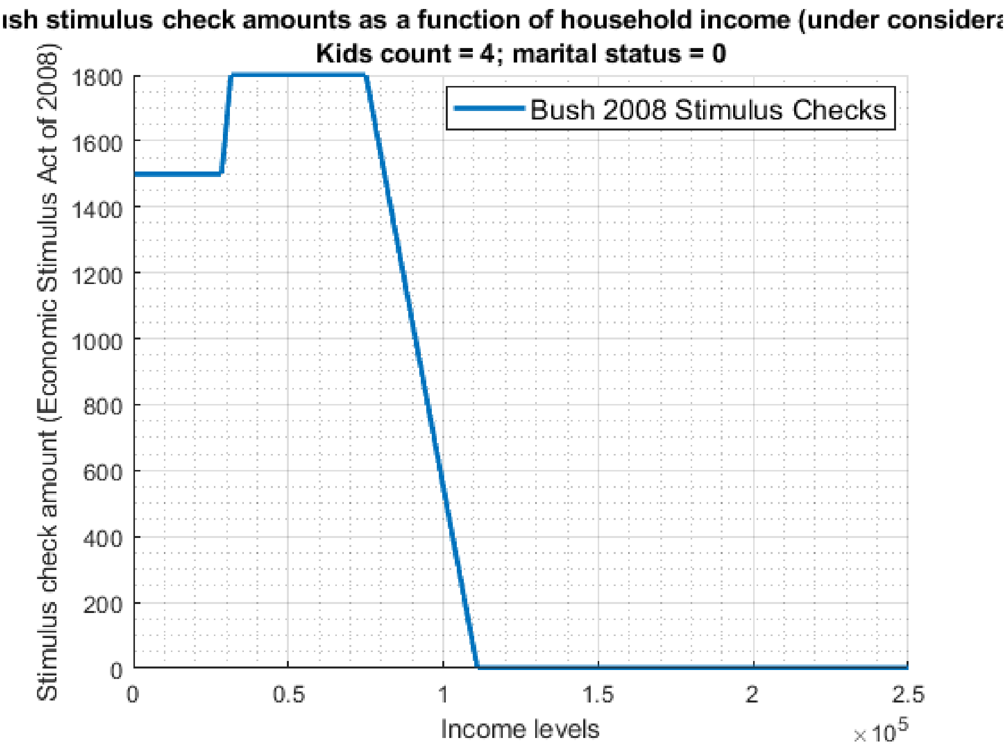
```
bl_visualize = true;
[fl_stimulus_adult, fl_stimulus_child] = deal(600, 300);
bl_marital = 0;
for it_kids=0:1:4
    snw_stimulus_checks_bush(it_kids, bl_marital, fl_stimulus_adult, fl_stimulus_child, bl_visualize
end
```




ısh stimulus check amounts as a function of household income (under considera
Kids count = 1; marital status = 0
Bush 2008 Stimulus Checks
Stimulus check amount (Economic Stimulus Act of 2008)
900
800
700
600
500
400
300
200
100
0
0
0.5
1
1.5
2
2.5
Income levels
$\times 10^5$


ısh stimulus check amounts as a function of household income (under considera
Kids count = 2; marital status = 0
Bush 2008 Stimulus Checks
Stimulus check amount (Economic Stimulus Act of 2008)
1200
1000
800
600
400
200
0
0
0.5
1
1.5
2
2.5
Income levels
$\times 10^5$

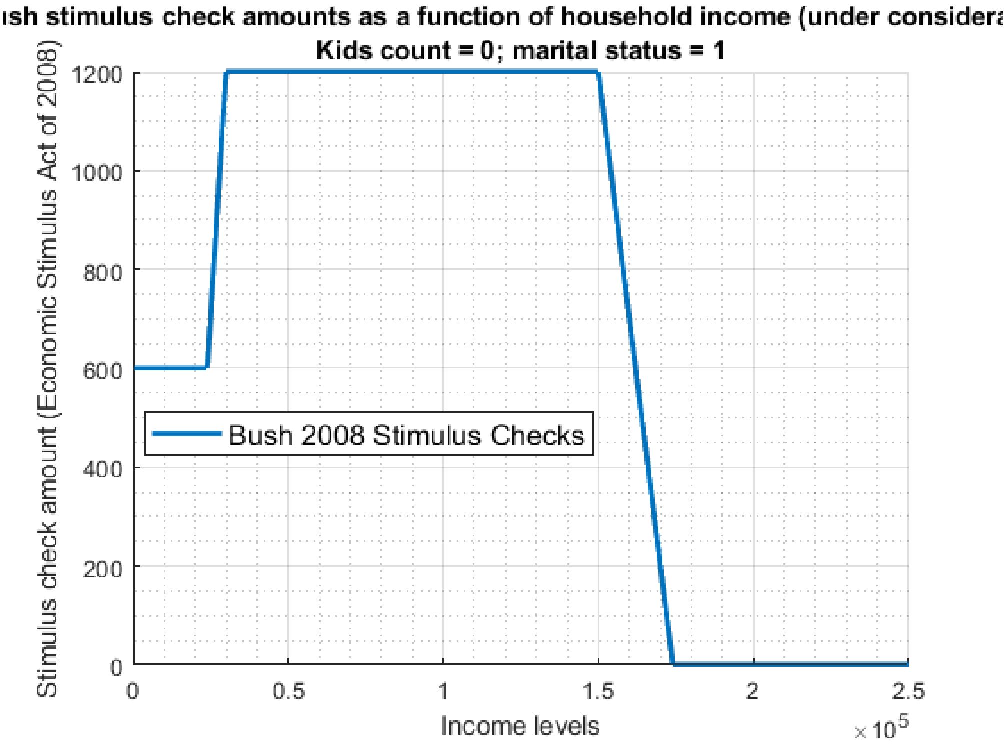


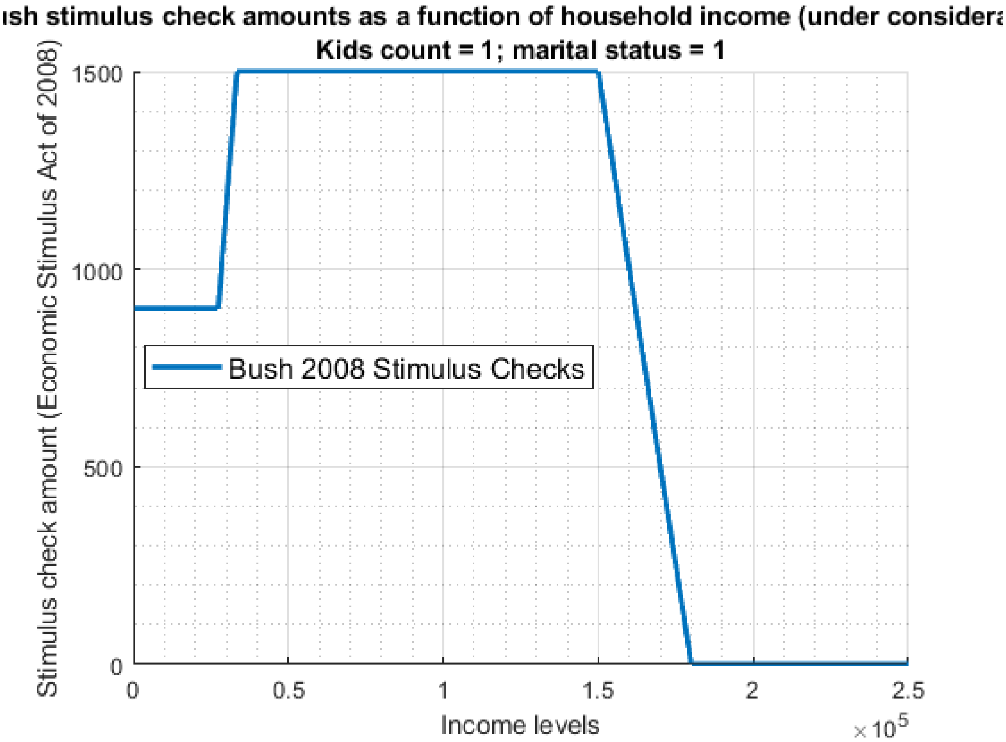


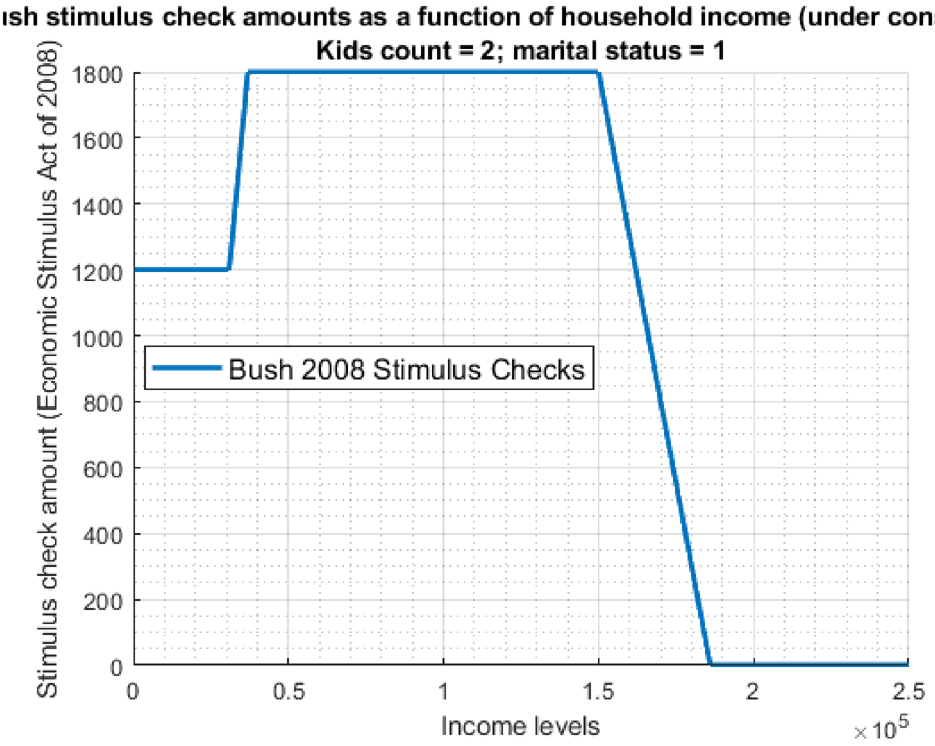
11.5.2 Bush Stimulus Checks for Married Households

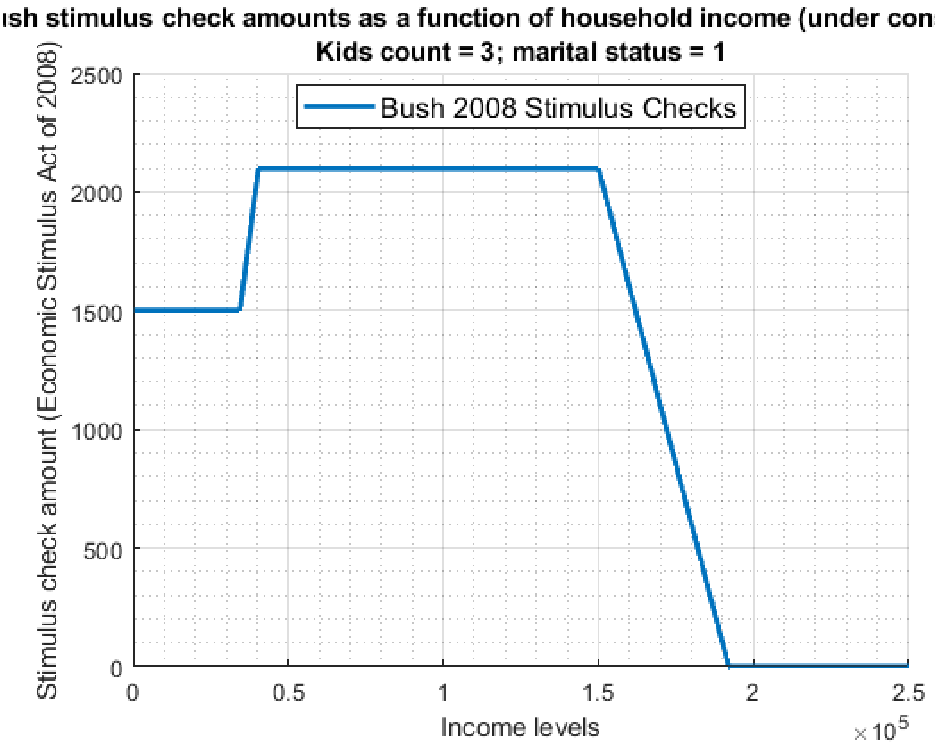
Visualize stimulus check amounts.

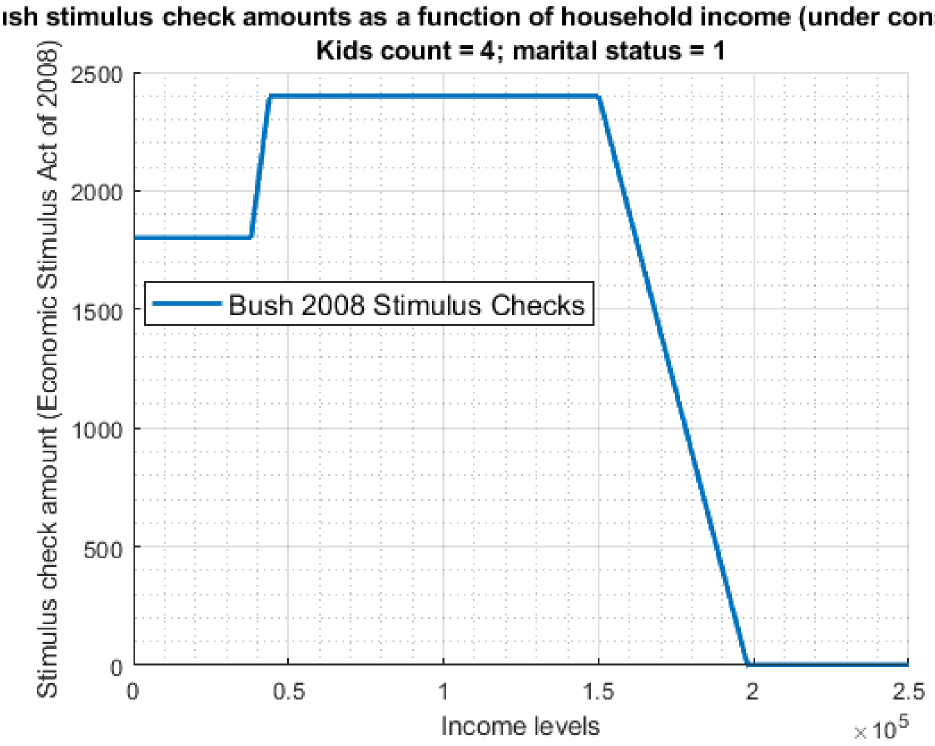
```
[fl_stimulus_adult, fl_stimulus_child] = deal(600, 300);
bl_marital = 1;
for it_kids=0:1:4
    snw_stimulus_checks_bush(it_kids, bl_marital, fl_stimulus_adult, fl_stimulus_child, bl_visualize
end
```


ısh stimulus check amounts as a function of household income (under considerа
Kids count = 0; marital status = 1
Stimulus check amount (Economic Stimulus Act of 2008)
1200
1000
800
600
400
200
0
0
0.5
1
1.5
2
2.5
Income levels
×10^5
Bush 2008 Stimulus Checks


ısh stimulus check amounts as a function of household income (under considerа
Kids count = 1; marital status = 1
Stimulus check amount (Economic Stimulus Act of 2008)
1500
1000
500
0
0
0.5
1
1.5
2
2.5
Income levels
×10^5
Bush 2008 Stimulus Checks


ısh stimulus check amounts as a function of household income (under considera
Kids count = 2; marital status = 1
Stimulus check amount (Economic Stimulus Act of 2008)
1800
1600
1400
1200
1000
800
600
400
200
0
0
0.5
1
1.5
2
2.5
Income levels
×10^5
Bush 2008 Stimulus Checks


ısh stimulus check amounts as a function of household income (under considera
Kids count = 3; marital status = 1
Stimulus check amount (Economic Stimulus Act of 2008)
2500
2000
1500
1000
500
0
0
0.5
1
1.5
2
2.5
Income levels
×10^5
Bush 2008 Stimulus Checks


ısh stimulus check amounts as a function of household income (under considera
Kids count = 4; marital status = 1
Stimulus check amount (Economic Stimulus Act of 2008)
2500
2000
1500
1000
500
0
0
0.5
1
1.5
2
2.5
Income levels
$\times 10^5$
Bush 2008 Stimulus Checks

Chapter 12

Calibration

12.1 Model Calibration

Taking advantage of **snw_calibrate_beta_norm_gdp** from the **PrjOptiSNW Package**, this function calibrates the discount factor and also solves for the normalizing constant.

12.1.1 Calibrate Parameter Controls for SNW Functions

Set up controls for shock process and tiny/small/dense/densemore

```
clear all;
bl_print_mp_params = false;
% st_shock_method = 'rouwenhorst';
st_shock_method = 'tauchen';
% st_param_group = 'default_tiny';
% st_param_group = 'default_small';
% st_param_group = 'default_base';
% st_param_group = 'default_dense';
% st_param_group = 'default_moredense';
st_param_group = 'default_docdense';
mp_params = snw_mp_param(st_param_group, bl_print_mp_params, st_shock_method);
Pop = mp_params('Pop');
```

Set up print defaults

```
mp_controls = snw_mp_control('default_test');
mp_controls('bl_timer') = true;
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
```

12.1.2 Calibrate Routine

Test this for 3 iterations

```
%% Calibration
err=1;
tol=0.005;
it_counter = 1;
while err>tol && it_counter <= 10
    disp('');
    it=1;

    while it>0
```

```
        % Solve optimization problem and get the distribution
        tm_start_a2 = tic;
        a2_old = mp_params('a2');
        [Phi_true,~,A_agg,Y_inc_agg,it,mp_dsvfi_results, a2] = snw_ds_main(mp_params, mp_controls);
        mp_params('a2') = a2;
        tm_end_a2 = toc(tm_start_a2);
        disp(['a2_old:' num2str(a2_old) ', a2_new:' num2str(a2) ', tm_end_a2:' num2str(tm_end_a2)])
    end

    % Get Stats
    mp_cl_mt_xyz_of_s = mp_dsvfi_results('mp_cl_mt_xyz_of_s');
    tb_outcomes = mp_cl_mt_xyz_of_s('tb_outcomes');
    A_agg_alt = tb_outcomes{'a_ss', 'mean'}*sum(Pop);
    Aprime_agg_alt = tb_outcomes{'ap_ss', 'mean'}*sum(Pop);
    Y_inc_agg_alt = tb_outcomes{'y_all', 'mean'}*sum(Pop);
    Y_inc_median = tb_outcomes{'y_all', 'p50'};

    % Comparison
    name='Median household income (target=1.0)=';
    name2=[name,num2str(Y_inc_median)];
    disp(name2);
    name='Aggregate wealth to aggregate income (target=3.0)=';
    name2=[name,num2str(A_agg/Y_inc_agg)];
    disp(name2);

    err1=abs(Y_inc_median-1.0); % Target: Median household income (normalized to 1 in the model)
    err2=abs((A_agg/Y_inc_agg)-3.0); % Target: Annual capital/income ratio of 3

    err=max(err1,err2);

    % Beta and Theta
    theta = mp_params('theta');
    beta = mp_params('beta');
    param_update=[theta;beta];

    if err>tol

        theta=theta*((1.0/Y_inc_median)^0.2); % Normalize theta such that median household income eq
        beta=beta*((3.0/(A_agg/Y_inc_agg))^0.025); % Calibrate beta such that annual capital/income

    end
    mp_params('theta') = theta;
    mp_params('beta') = beta;

    param_update=[param_update(1,1),theta;param_update(2,1),beta];

    it_counter = it_counter + 1;
    name='Old/updated theta:';
    st_theta=[name, num2str(param_update(1,:))];
    name='Old/updated beta:';
    st_beta=[name,num2str(param_update(2,:))];
    disp(['counter=' num2str(it_counter) ...
        ';beta=' num2str(beta) ...
        ';theta=' num2str(theta)]);
end

Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=489.
```

```
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1835.3351
a2_old:1.5286, a2_new:1.4349, tm_end_a2:2440.3793
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=487.
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1447.245
a2_old:1.4349, a2_new:1.4342, tm_end_a2:2051.784
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=483.
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1352.0741
a2_old:1.4342, a2_new:1.4342, tm_end_a2:1951.1421
Median household income (target=1.0)=1.0024
Aggregate wealth to aggregate income (target=3.0)=3.0822
counter=2;beta=0.97051;theta=0.56495
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=493.
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1570.3759
a2_old:1.4342, a2_new:1.4374, tm_end_a2:2178.0519
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=487.
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1348.532
a2_old:1.4374, a2_new:1.4374, tm_end_a2:1948.8124
Median household income (target=1.0)=0.99971
Aggregate wealth to aggregate income (target=3.0)=3.0379
counter=3;beta=0.9702;theta=0.56499
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=486.
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1503.2336
a2_old:1.4374, a2_new:1.439, tm_end_a2:2101.4027
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=485.
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1334.6119
a2_old:1.439, a2_new:1.439, tm_end_a2:1933.0606
Median household income (target=1.0)=0.99853
Aggregate wealth to aggregate income (target=3.0)=3.0176
counter=4;beta=0.97006;theta=0.56515
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=503.
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1423.8673
a2_old:1.439, a2_new:1.4393, tm_end_a2:2041.1993
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=497.
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1386.4679
a2_old:1.4393, a2_new:1.4393, tm_end_a2:1998.5793
Median household income (target=1.0)=0.99813
Aggregate wealth to aggregate income (target=3.0)=3.0084
counter=5;beta=0.96999;theta=0.56536
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=501.
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1375.2408
a2_old:1.4393, a2_new:1.4393, tm_end_a2:1990.0337
Median household income (target=1.0)=0.99829
Aggregate wealth to aggregate income (target=3.0)=3.0041
counter=6;beta=0.96999;theta=0.56536
```

12.2 Model Lockdown Calibration

Taking advantage of **snw_calibrate_lockdown_c** from the **PrjOptiSNW Package.** This function finds the proportional discount to current utility in one period to account for aggregate reducntions in consumption during lockdown.

12.2.1 Lock Down Impact on Consumption

"To calibrate the drop in marginal utility, we estimate that 10.9 percent of the goods that make up the consumer price index become highly undesirable, or simply unavailable, during the pandemic: food away from home, public transportation including airlines, and motor fuel. As we use a coefficient of risk aversion equal to one, we simply multiply utility from consumption during the period of the epidemic by a factor of 0.891"

There is one MIT shock period in which households face a one-period change in the current utility of consumption due to lock down. Solve the model and evaluate the effects on aggregate consumption (during the MIT shock period) with different proportional adjustments on consumption given differing intertemperal preference assumptions.

12.2.2 Graphical Illustraiton for Gamma=2 (docdense) and Gamma=1 (dense)

Solved Gamma=2 with docdense (81x5 shocks), and Gamma=1 with dense (7x5) shocks. Did not have time to resolve Gamma=1 with docdense.

Solved for gamma equals to 2, results visualized below.

```
% Percentage C drop desired
fl_c_drop_percent = 0.109;
```

First, grid of proportional one period current utility shifts:

```
% Grid of proportional one period current utility shifts.
[fl_invbtlock_min, fl_invbtlock_max, it_invbtlock_points] = deal(0, 1, 20);
st_grid_type = 'grid_powerspace';
mp_grid_control = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_grid_control('grid_powerspace_power') = 1.5;
[ar_fl_invbtlock] = ff_saveborr_grid(...
    fl_invbtlock_min, fl_invbtlock_max, it_invbtlock_points, ...
    st_grid_type, mp_grid_control);
ar_fl_invbtlock = 1-ar_fl_invbtlock;
% display
% disp(ar_fl_invbtlock);
```

Second, stored aggregate consumption values from docdense:

```
% From gamma=2 docdense
ar_fl_cons_mean_betaedu_innerwgt_gamma2_docdense = [...
    1.0466,1.0434,1.0374,1.0293,...
    1.0191,1.0067,0.99214,0.97543,...
    0.95642,0.93482,0.90995,0.88078,...
    0.847,0.80734,0.76126,0.70507,...
    0.635, 0.54343, 0.40755, 0.00013065];
% From gamma=1 dense
ar_fl_cons_mean_betaedu_innerwgt_gamma1_dense = [...
    1.2517,1.2467,1.237,1.2228,...
    1.2031,1.1783,1.1489,1.1147,...
    1.0754,1.0311,0.98169,0.92291,...
    0.85528,0.77933,0.69408,0.5964,...
    0.47976, 0.34416, 0.18672, 0.00013642];
```

Third, polynomial fit (4th order) between the 3rd and 14th point:

```
% Generate polynomial fit coefficients
ft_polynomial_gamma2_docdense = polyfit(ar_fl_invbtlock(3:13), ...
    ar_fl_cons_mean_betaedu_innerwgt_gamma2_docdense(3:13), 4);
ft_polynomial_gamma1_dense = polyfit(ar_fl_invbtlock(3:13), ...
    ar_fl_cons_mean_betaedu_innerwgt_gamma1_dense(3:13), 4);
% Evaluate polynomial fits
ar_fl_cons_mean_polyfit_gamma2_docdense = ...
    polyval(ft_polynomial_gamma2_docdense, ar_fl_invbtlock)';
ar_fl_cons_mean_polyfit_gamma1_dense = ...
    polyval(ft_polynomial_gamma1_dense, ar_fl_invbtlock)';
```

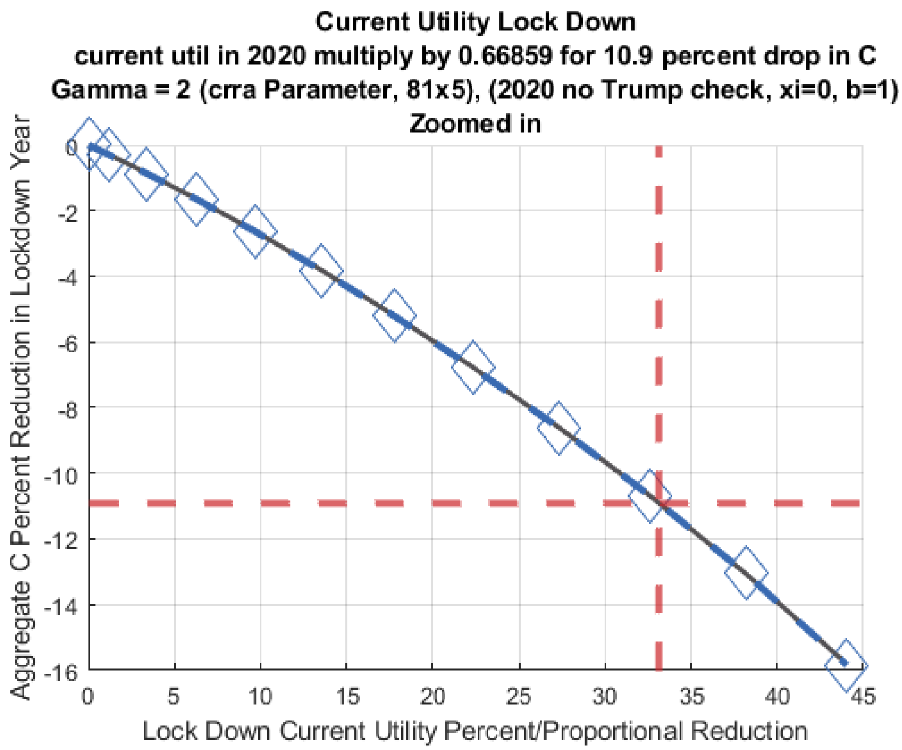
Fourth, find which proportional current utility change in 2020 matches a 10.9 percent drop in aggregate consumption in 2020:

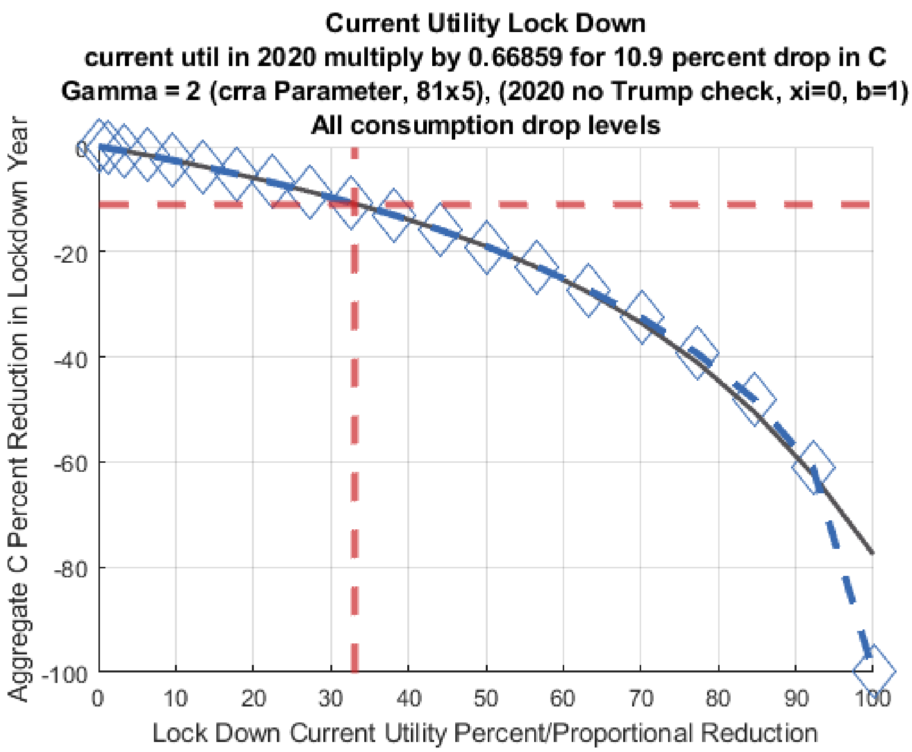
```
% Identify the exact point along polynomial where c drops by 10.9 percent
% Gamma = 2, docdense
ar_fl_invbtlock_interpgrid = linspace(0.50, 0.95, 100000);
ar_fit_reduce_grid = 1 - ...
    polyval(ft_polynomial_gamma2_docdense, ar_fl_invbtlock_interpgrid)...
    ./ar_fl_cons_mean_polyfit_gamma2_docdense(1);
[fl_mingap, it_minidx] = min(abs(ar_fit_reduce_grid-fl_c_drop_percent));
fl_fit_best_reduce = ar_fl_invbtlock_interpgrid(it_minidx);
fl_lockdown_u_prop_reduction_gamma2_docdense = (1-fl_fit_best_reduce)*100;
st_reduction_gamma2_docdense = ['current util in 2020 multiply by ', ...
    num2str(fl_fit_best_reduce), ...
    ' for 10.9 percent drop in C'];
% Gamma = 1, dense
ar_fl_invbtlock_interpgrid = linspace(0.50, 0.95, 100000);
ar_fit_reduce_grid = 1 - ...
    polyval(ft_polynomial_gamma1_dense, ar_fl_invbtlock_interpgrid)...
    ./ar_fl_cons_mean_polyfit_gamma1_dense(1);
[fl_mingap, it_minidx] = min(abs(ar_fit_reduce_grid-fl_c_drop_percent));
fl_fit_best_reduce = ar_fl_invbtlock_interpgrid(it_minidx);
fl_lockdown_u_prop_reduction_gamma1_dense = (1-fl_fit_best_reduce)*100;
st_reduction_gamma1_dense = ['current util in 2020 multiply by ', ...
    num2str(fl_fit_best_reduce), ...
    ' for 10.9 percent drop in C'];
```

Third, graphical illustration:

```
for it_graph_type=1:2
    if (it_graph_type == 1)
        st_title_add = 'Zoomed in';
    else
        st_title_add = 'All consumption drop levels';
    end

    for it_gamma=2:1:2
        figure();
        hold on;

        if (it_graph_type == 1 && it_gamma == 1)
            ar_fl_cons_mean_betaedu_innerwgt_select = ...
                ar_fl_cons_mean_betaedu_innerwgt_gamma1_dense(1:12);
            ar_fl_cons_mean_polyfit_select = ...
                ar_fl_cons_mean_polyfit_gamma1_dense(1:12);
            st_gamma = 'Gamma = 1 (crra Parameter, 7x5), (2020 no Trump check, xi=0, b=1)';
            fl_lockdown_u_prop_reduction = fl_lockdown_u_prop_reduction_gamma1_dense;
            st_reduction = st_reduction_gamma1_dense;
        elseif (it_graph_type == 1 && it_gamma == 2)
            ar_fl_cons_mean_betaedu_innerwgt_select = ...
                ar_fl_cons_mean_betaedu_innerwgt_gamma2_docdense(1:12);
            ar_fl_cons_mean_polyfit_select = ...
                ar_fl_cons_mean_polyfit_gamma2_docdense(1:12);
            st_gamma = 'Gamma = 2 (crra Parameter, 81x5), (2020 no Trump check, xi=0, b=1)';
            fl_lockdown_u_prop_reduction = fl_lockdown_u_prop_reduction_gamma2_docdense;
            st_reduction = st_reduction_gamma2_docdense;
        elseif (it_graph_type == 2 && it_gamma == 1)
            ar_fl_cons_mean_betaedu_innerwgt_select = ...
                ar_fl_cons_mean_betaedu_innerwgt_gamma1_dense;
            ar_fl_cons_mean_polyfit_select = ...
                ar_fl_cons_mean_polyfit_gamma1_dense;
            st_gamma = 'Gamma = 1 (crra Parameter, 7x5), (2020 no Trump check, xi=0, b=1)';
```

```
            fl_lockdown_u_prop_reduction = fl_lockdown_u_prop_reduction_gamma1_dense;
            st_reduction = st_reduction_gamma1_dense;
        elseif (it_graph_type == 2 && it_gamma == 2)
            ar_fl_cons_mean_betaedu_innerwgt_select = ...
                ar_fl_cons_mean_betaedu_innerwgt_gamma2_docdense;
            ar_fl_cons_mean_polyfit_select = ...
                ar_fl_cons_mean_polyfit_gamma2_docdense;
            st_gamma = 'Gamma = 2 (crra Parameter, 81x5), (2020 no Trump check, xi=0, b=1)';
            fl_lockdown_u_prop_reduction = fl_lockdown_u_prop_reduction_gamma2_docdense;
            st_reduction = st_reduction_gamma2_docdense;
        end

        ar_fl_invbtlock_select = ar_fl_invbtlock(1:length(ar_fl_cons_mean_betaedu_innerwgt_select));
        ar_x = 1-ar_fl_invbtlock_select;
        ar_y = -(1-ar_fl_cons_mean_betaedu_innerwgt_select/ar_fl_cons_mean_betaedu_innerwgt_select(1
        ar_y_poly = -(1-ar_fl_cons_mean_polyfit_select/ar_fl_cons_mean_polyfit_select(1));
        ar_x = ar_x*100;
        ar_y = ar_y*100;
        ar_y_poly = ar_y_poly*100;

        % Points scatter
        scatter(ar_x, ar_y, 300, [57 106 177]./255, 'd');

        % Points polynomial approximate
        pl_poly = plot(ar_x, ar_y_poly);
        pl_poly.Color = [83 81 84]./255;
        pl_poly.LineStyle = '-';
        pl_poly.LineWidth = 2;

        % Actual lines linearly connected
        line = plot(ar_x, ar_y);
        line.Color = [57 106 177]./255;
        line.LineStyle = '--';
        line.LineWidth = 3;

        % X-axis for -10.9 percent
        yline0 = yline(-10.9);
        yline0.HandleVisibility = 'off';
        yline0.Color = [204 37 41]./255;
        yline0.LineStyle = '--';
        yline0.LineWidth = 3;

        % Y-axis for -10.9 percent along the polynomial
        xline0 = xline(fl_lockdown_u_prop_reduction);
        xline0.HandleVisibility = 'off';
        xline0.Color = [204 37 41]./255;
        xline0.LineStyle = '--';
        xline0.LineWidth = 3;

        % labeling
        title({'Current Utility Lock Down',...
            st_reduction, st_gamma, ...
            st_title_add});
        ylabel('Aggregate C Percent Reduction in Lockdown Year');
        xlabel('Lock Down Current Utility Percent/Proportional Reduction');
        grid on;
    end
end
```





Fourth, tabular display for Gamma 2:

```
%% Table Dispaly
% Generate Table
tb_show = array2table([ar_x,ar_y',ar_y_poly']);
it_num_rows = length(ar_x);

% Generate Row and Column Names
cl_col_names = {'Lock Down Util Perc Reduce', 'Agg C Percent Reduc Lockdown Yr', 'Agg C Percent Redu
cl_row_names = strcat('lockdown_', string((1:it_num_rows)));
```

```
tb_show.Properties.VariableNames = matlab.lang.makeValidName(cl_col_names);
tb_show.Properties.RowNames = matlab.lang.makeValidName(cl_row_names);
disp(tb_show);
```

```
                   LockDownUtilPercReduce    AggCPercentReducLockdownYr    AggCPercentReducPolynomia
                   ______________________    __________________________    _________________________

    lockdown_1                 0                          0                            0
    lockdown_2            1.2075                   -0.30575                     -0.29672
    lockdown_3            3.4152                   -0.87904                     -0.86414
    lockdown_4            6.2741                     -1.653                       -1.643
    lockdown_5            9.6596                    -2.6276                      -2.6234
    lockdown_6              13.5                    -3.8123                      -3.8041
    lockdown_7            17.746                    -5.2035                      -5.1879
    lockdown_8            22.362                    -6.8001                      -6.7823
    lockdown_9            27.322                    -8.6165                      -8.6029
    lockdown_10           32.601                     -10.68                      -10.679
    lockdown_11           38.183                    -13.057                      -13.058
    lockdown_12           44.051                    -15.844                      -15.817
    lockdown_13           50.193                    -19.071                      -19.067
    lockdown_14           56.596                    -22.861                      -22.966
    lockdown_15            63.25                    -27.264                      -27.728
    lockdown_16           70.147                    -32.632                      -33.635
    lockdown_17           77.277                    -39.327                      -41.055
    lockdown_18           84.634                    -48.077                      -50.447
    lockdown_19            92.21                     -61.06                      -62.385
    lockdown_20              100                    -99.988                      -77.569
```

12.3 UI Benefit Unemployment Lost Wage Recovery Parameter b Calibration

Taking advantage of **snw__calibrate__2009__b** from the **PrjOptiSNW Package.**

The ratio of UI benefits to wages and salary is 2.1 percent in 2009. $\xi \in [0, 1]$ governs the duration of unemployment shock for those unemployed. This equals to 0.532 in 2009 ($\xi = 0$ no wages earned).

We solve for total wage earnings from unemployed and employed in 2009, for employed, same as under steady-state. For unemployed, they lose $(1 - \xi)$ share of the wage they would otherwise have earned. Our unemployment probability in 2009 is conditional on age and edu groups (SNW_UNEMP_2008.m) computed based on rectiilnear restriction.

We know total UI amount (multiply its share of total "Wages and salary" by total "wages and salary". We know how much wage was lost due to ξ. The ratio of these two levels is b, which is the parameter that is the share of lost-wage recovered. Note that this is based on exogenous wage earnings, so we do not have to worry about endogenous changes to savings. We will solve for the steady-state distribution, which generates mass of people by age, education, marital status, kids count, etc.

12.3.1 Calibrate b with 2.1% UI Benefits to Wages Ratio and $\xi = 0.532$

Using various default parameters, including the default unemployment in 2009 matrix, and the default $\xi = 0.532$ parameter, compute b.

```
% Solve parameters
mp_more_inputs = containers.Map('KeyType','char', 'ValueType','any');
mp_more_inputs('fl_ss_non_college') = 0.225;
mp_more_inputs('fl_ss_college') = 0.271;
mp_more_inputs('fl_scaleconvertor') = 54831;
% st_param_group = 'default_small';
```

```
st_param_group = 'default_docdense';
mp_params = snw_mp_param(st_param_group, false, 'tauchen', false, 8, 8, mp_more_inputs);
% Controls
mp_controls = snw_mp_control('default_test');

% no b, solving for b, b set to 0 when solving for wages
xi=0.532; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=
mp_params('xi') = xi;

% Solve for Unemployment Values
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_calibrate_2009') = true;
mp_controls('bl_print_calibrate_2009_verbose') = false;

% 2.1% UI Benefits to Wages and Salary Ratio
fl_ratio_ui_benefits_to_wage = 0.021;

% Solve
[fl_b_calibrated_by_ui_share, ...
    mp_stats_wage_ui_spending, ...
    mn_earn_tot_wgted, mn_earn_unemp_wgted, ...
    mn_earn_unemp_tot_wgted, mn_earn_unemp_weighted_wgted] = ...
    snw_calibrate_2009_b(mp_params, mp_controls, ...
    fl_ratio_ui_benefits_to_wage);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=310.
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1294.5472
Completed SNW_calibrate_2009;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1668.355
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_stats_wage_ui_spending Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                             i    idx     value
                                             -    ---    -------

    fl_b_calibrated_by_ui_share              1     1     0.37451
    fl_total_b_spending                      2     2      1.1333
    fl_total_wage                            3     3      53.969
    fl_total_wage_unemp_hhhead               4     4      3.4401
    fl_total_wage_unemp_hhhead_and_spouse    5     5      6.0062
    fl_total_wage_unemp_hhhead_lost          6     6      3.0262
```

12.3.2 Calibrate b with 5.68% UI Benefits to Wages Ratio and $\xi = 0.651$

Change the benefit share and ξ parameter to COVID values. The b we find below is not what should be used for COVID, the unemployment probability is based on 2009 crisis still. That is hard-coded into the **snw_calibrate_2009_b** function.

```
% Solve parameters
mp_more_inputs = containers.Map('KeyType','char', 'ValueType','any');
mp_more_inputs('fl_ss_non_college') = 0.225;
mp_more_inputs('fl_ss_college') = 0.271;
mp_more_inputs('fl_scaleconvertor') = 54831;
% st_param_group = 'default_small';
st_param_group = 'default_dense';
mp_params = snw_mp_param(st_param_group, false, 'tauchen', false, 8, 8, mp_more_inputs);
```

```
% Controls
mp_controls = snw_mp_control('default_test');

% no b, solving for b, b set to 0 when solving for wages
xi=0.651; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=
mp_params('xi') = xi;

% Solve for Unemployment Values
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_calibrate_2009') = true;
mp_controls('bl_print_calibrate_2009_verbose') = false;

% 2.1% UI Benefits to Wages and Salary Ratio
fl_ratio_ui_benefits_to_wage = 0.0568;

% Solve
[fl_b_calibrated_by_ui_share, ...
    mp_stats_wage_ui_spending, ...
    mn_earn_tot_wgted, mn_earn_unemp_wgted, ...
    mn_earn_unemp_tot_wgted, mn_earn_unemp_weighted_wgted] = ...
    snw_calibrate_2009_b(mp_params, mp_controls, ...
    fl_ratio_ui_benefits_to_wage);

Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time=14.9366
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time=37.6607
Completed SNW_calibrate_2009;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time=55.5689
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_stats_wage_ui_spending Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                             i    idx    value
                                             -    ---    ------

    fl_b_calibrated_by_ui_share              1     1     1.3973
    fl_total_b_spending                      2     2     3.8087
    fl_total_wage                            3     3     67.055
    fl_total_wage_unemp_hhhead               4     4     5.0843
    fl_total_wage_unemp_hhhead_and_spouse    5     5     8.3311
    fl_total_wage_unemp_hhhead_lost          6     6     2.7257
```

Chapter 13

Summary Statistics

13.1 2019 Full States MPC and Distributional Statistics by Marital, Kids, and Income Groups.

In the file here, we consider marital, kids and income groups, and summarize various statistics for each bin.

13.1.1 Test SNW_EVUVW19_JAEEMK Defaults Dense

VFI and Distribution

Call the function with defaults.

```
clear all;
st_solu_type = 'bisec_vec';
bl_save_csv = false;

% Solve the VFI Problem and get Value Function
% mp_params = snw_mp_param('default_dense');
% mp_params = snw_mp_param('default_docdense');
mp_params = snw_mp_param('default_moredense_a65zh133zs5_e2m2');
mp_controls = snw_mp_control('default_test');

% set Unemployment Related Variables
xi=0.5; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=1
b=1; % Unemployment insurance replacement rate (b=0 refers to no UI benefits; b=1 refers to 100 perc
TR=100/58056; % Value of a welfare check (can receive multiple checks). TO DO: Update with alternati

mp_params('xi') = xi;
mp_params('b') = b;
mp_params('TR') = TR;

% Solve for Unemployment Values
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = true;
mp_controls('bl_print_ds_verbose') = true;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;
```

```
mp_controls('bl_print_evuvw19_jaeemk') = false;
mp_controls('bl_print_evuvw19_jaeemk_verbose') = false;

% Solve the Model to get V working and unemployed
[V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);

Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_moredense_a65zh133zs5_e2m2;SNW_MP_CONTROL=defa

inc_VFI = mp_valpol_more_ss('inc_VFI');
spouse_inc_VFI = mp_valpol_more_ss('spouse_inc_VFI');
total_inc_VFI = inc_VFI + spouse_inc_VFI;
% tax during covid year
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
% Solve unemployment
[V_unemp,~,cons_unemp,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);

Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_moredense_a65zh133zs5_e2m

[Phi_true, Phi_adj, A_agg, Y_inc_agg, ~, mp_dsvfi_results] = snw_ds_main_vec(mp_params, mp_controls,

SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:1 of 82, time-this-age:1.074
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:2 of 82, time-this-age:20.5148
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:3 of 82, time-this-age:23.4908
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:4 of 82, time-this-age:28.525
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:5 of 82, time-this-age:33.2054
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:6 of 82, time-this-age:35.3197
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:7 of 82, time-this-age:37.5611
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:8 of 82, time-this-age:40.226
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:9 of 82, time-this-age:44.3653
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:10 of 82, time-this-age:48.3751
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:11 of 82, time-this-age:49.4182
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:12 of 82, time-this-age:50.6325
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:13 of 82, time-this-age:51.0802
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:14 of 82, time-this-age:52.1717
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:15 of 82, time-this-age:53.2068
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:16 of 82, time-this-age:53.6567
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:17 of 82, time-this-age:53.8811
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:18 of 82, time-this-age:55.0892
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:19 of 82, time-this-age:55.6717
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:20 of 82, time-this-age:56.2143
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:21 of 82, time-this-age:56.5704
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:22 of 82, time-this-age:57.0081
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:23 of 82, time-this-age:57.1682
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:24 of 82, time-this-age:57.3671
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:25 of 82, time-this-age:57.5453
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:26 of 82, time-this-age:57.8356
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:27 of 82, time-this-age:58.0491
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:28 of 82, time-this-age:57.9265
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:29 of 82, time-this-age:57.6332
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:30 of 82, time-this-age:58.1269
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:31 of 82, time-this-age:57.7606
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:32 of 82, time-this-age:57.5816
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:33 of 82, time-this-age:57.3361
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:34 of 82, time-this-age:57.7288
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:35 of 82, time-this-age:56.9154
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:36 of 82, time-this-age:57.2866
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:37 of 82, time-this-age:57.1634
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:38 of 82, time-this-age:57.0388
```

```
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:39 of 82, time-this-age:56.6859
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:40 of 82, time-this-age:56.7277
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:41 of 82, time-this-age:56.9976
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:42 of 82, time-this-age:56.6711
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:43 of 82, time-this-age:56.7355
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:44 of 82, time-this-age:56.6671
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:45 of 82, time-this-age:56.1114
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:46 of 82, time-this-age:55.9357
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:47 of 82, time-this-age:55.9514
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:48 of 82, time-this-age:55.4533
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:49 of 82, time-this-age:58.5505
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:50 of 82, time-this-age:59.402
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:51 of 82, time-this-age:59.5814
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:52 of 82, time-this-age:59.4987
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:53 of 82, time-this-age:59.3449
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:54 of 82, time-this-age:59.6498
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:55 of 82, time-this-age:59.3396
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:56 of 82, time-this-age:59.4903
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:57 of 82, time-this-age:59.4659
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:58 of 82, time-this-age:59.2382
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:59 of 82, time-this-age:58.2574
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:60 of 82, time-this-age:58.4884
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:61 of 82, time-this-age:58.2825
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:62 of 82, time-this-age:57.4508
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:63 of 82, time-this-age:56.9986
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:64 of 82, time-this-age:56.5337
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:65 of 82, time-this-age:55.94
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:66 of 82, time-this-age:54.1804
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:67 of 82, time-this-age:53.4807
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:68 of 82, time-this-age:52.222
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:69 of 82, time-this-age:51.6643
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:70 of 82, time-this-age:50.7393
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:71 of 82, time-this-age:49.5324
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:72 of 82, time-this-age:47.7517
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:73 of 82, time-this-age:45.9439
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:74 of 82, time-this-age:44.385
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:75 of 82, time-this-age:42.9
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:76 of 82, time-this-age:41.3804
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:77 of 82, time-this-age:35.089
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:78 of 82, time-this-age:33.9143
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:79 of 82, time-this-age:32.9597
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:80 of 82, time-this-age:26.3587
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:81 of 82, time-this-age:25.2198
SNW_DS_MAIN_VEC ACUMU MASS: Finished Age Group:82 of 82, time-this-age:22.8558
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:1 of 82, time-this-age:0.50074
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:2 of 82, time-this-age:0.078102
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:3 of 82, time-this-age:0.077705
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:4 of 82, time-this-age:0.077939
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:5 of 82, time-this-age:0.07796
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:6 of 82, time-this-age:0.078664
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:7 of 82, time-this-age:0.077012
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:8 of 82, time-this-age:0.077566
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:9 of 82, time-this-age:0.076968
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:10 of 82, time-this-age:0.076874
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:11 of 82, time-this-age:0.07674
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:12 of 82, time-this-age:0.07736
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:13 of 82, time-this-age:0.07804
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:14 of 82, time-this-age:0.077614
```

```
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:15 of 82, time-this-age:0.076794
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:16 of 82, time-this-age:0.077524
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:17 of 82, time-this-age:0.077125
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:18 of 82, time-this-age:0.076024
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:19 of 82, time-this-age:0.074863
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:20 of 82, time-this-age:0.07631
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:21 of 82, time-this-age:0.073418
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:22 of 82, time-this-age:0.073802
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:23 of 82, time-this-age:0.073525
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:24 of 82, time-this-age:0.073143
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:25 of 82, time-this-age:0.073793
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:26 of 82, time-this-age:0.073628
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:27 of 82, time-this-age:0.074547
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:28 of 82, time-this-age:0.074412
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:29 of 82, time-this-age:0.074226
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:30 of 82, time-this-age:0.073869
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:31 of 82, time-this-age:0.072886
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:32 of 82, time-this-age:0.074226
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:33 of 82, time-this-age:0.073236
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:34 of 82, time-this-age:0.074885
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:35 of 82, time-this-age:0.073514
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:36 of 82, time-this-age:0.073964
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:37 of 82, time-this-age:0.07539
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:38 of 82, time-this-age:0.073973
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:39 of 82, time-this-age:0.073071
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:40 of 82, time-this-age:0.073901
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:41 of 82, time-this-age:0.073413
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:42 of 82, time-this-age:0.074556
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:43 of 82, time-this-age:0.07319
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:44 of 82, time-this-age:0.073482
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:45 of 82, time-this-age:0.073433
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:46 of 82, time-this-age:0.073574
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:47 of 82, time-this-age:0.073129
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:48 of 82, time-this-age:0.073958
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:49 of 82, time-this-age:0.07419
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:50 of 82, time-this-age:0.07333
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:51 of 82, time-this-age:0.073146
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:52 of 82, time-this-age:0.07345
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:53 of 82, time-this-age:0.073098
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:54 of 82, time-this-age:0.073673
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:55 of 82, time-this-age:0.072706
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:56 of 82, time-this-age:0.073966
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:57 of 82, time-this-age:0.073423
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:58 of 82, time-this-age:0.073912
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:59 of 82, time-this-age:0.073841
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:60 of 82, time-this-age:0.073261
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:61 of 82, time-this-age:0.073155
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:62 of 82, time-this-age:0.074912
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:63 of 82, time-this-age:0.076206
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:64 of 82, time-this-age:0.073746
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:65 of 82, time-this-age:0.072696
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:66 of 82, time-this-age:0.073178
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:67 of 82, time-this-age:0.073645
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:68 of 82, time-this-age:0.073697
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:69 of 82, time-this-age:0.073477
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:70 of 82, time-this-age:0.075493
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:71 of 82, time-this-age:0.073607
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:72 of 82, time-this-age:0.073559
```

```
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:73 of 82, time-this-age:0.073002
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:74 of 82, time-this-age:0.073612
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:75 of 82, time-this-age:0.073039
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:76 of 82, time-this-age:0.073474
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:77 of 82, time-this-age:0.073582
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:78 of 82, time-this-age:0.076234
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:79 of 82, time-this-age:0.073668
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:80 of 82, time-this-age:0.073745
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:81 of 82, time-this-age:0.073108
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:82 of 82, time-this-age:0.072892
SNW_DS_MAIN NORMALIZE MASS: Finished Age Group:83 of 82, time-this-age:0.073316
SNW_DS_MAIN: Share of population with assets equal to upper bound on asset grid:6.0111e-06
SNW_DS_MAIN: Accidental bequests are thrown in the ocean
SNW_DS_MAIN_VEC tax and spend;it=1;err=0.0010205
SNW_DS_MAIN_VEC tax and spend;it=2;err=0.0008547
SNW_DS_MAIN_VEC tax and spend;it=3;err=0.0007159
SNW_DS_MAIN_VEC tax and spend;it=4;err=0.00059969
SNW_DS_MAIN_VEC tax and spend;it=5;err=0.00050237
SNW_DS_MAIN_VEC tax and spend;it=6;err=0.00042087
SNW_DS_MAIN_VEC tax and spend;it=7;err=0.00035261
SNW_DS_MAIN_VEC tax and spend;it=8;err=0.00029542
SNW_DS_MAIN_VEC tax and spend;it=9;err=0.00024752
SNW_DS_MAIN_VEC tax and spend;it=10;err=0.0002074
SNW_DS_MAIN_VEC tax and spend;it=11;err=0.00017378
SNW_DS_MAIN_VEC tax and spend;it=12;err=0.00014561
SNW_DS_MAIN_VEC tax and spend;it=13;err=0.00012201
SNW_DS_MAIN_VEC tax and spend;it=14;err=0.00010224
SNW_DS_MAIN_VEC tax and spend;it=15;err=8.567e-05
SNW_DS_MAIN_VEC: Number of a2-adjustments (for taxation) used to balance the government budget= 15
SNW_DS_MAIN_VEC: Old and updated value of a2=1.5286      1.5353
SNW_DS_MAIN_VEC: Aggregates: Cons., Gov. cons., Save, Assets, Income, Bequests 48.78871      11.3586
SNW_DS_MAIN_VEC: Resource constraint: C_t+A_{t+1}+G_t=A_t+Y_t 258.0346      258.0206
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_moredense_a65zh133zs5_e2m2;SNW_MP_CONTROL=default_tes
pos = 19 ; key = mp_controls
  Map with properties:

        Count: 37
      KeyType: char
    ValueType: any

pos = 20 ; key = mp_params
  Map with properties:

        Count: 52
      KeyType: char
    ValueType: any

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_dsvfi_results ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                        i     idx    ndim      numel       rowN      colN          sum         mean
                        --    ---    ----    ----------    ----    ---------    ----------    ------

    SS_ss               1     11      6      7.1754e+07     83     8.645e+05    8.3556e+06     0.116
    a_ss                2     16      6      7.1754e+07     83     8.645e+05    2.4595e+09      34.2
    ap_ss               3     17      6      7.1754e+07     83     8.645e+05    2.3245e+09      32.3
    cons_ss             4     18      6      7.1754e+07     83     8.645e+05    3.5119e+08      4.89
```

```
    n_ss                5     21      6      7.1754e+07     83      8.645e+05    2.5114e+08         3
    tax_ss              6     22      6      7.1754e+07     83      8.645e+05    6.6049e+07      0.92
    y_all_ss            7     23      6      7.1754e+07     83      8.645e+05    2.8219e+08      3.93
    y_head_earn_ss      8     24      6      7.1754e+07     83      8.645e+05     1.078e+08      1.50
    y_head_inc_ss       9     25      6      7.1754e+07     83      8.645e+05    2.1454e+08        2.
    y_spouse_inc_ss    10     26      6      7.1754e+07     83      8.645e+05    6.7646e+07     0.942
    yshr_SS_ss         11     27      6      7.1754e+07     83      8.645e+05    1.0586e+07     0.147
    yshr_interest_ss   12     28      6      7.1754e+07     83      8.645e+05    3.0079e+07      0.41
    yshr_nttxss_ss     13     29      6      7.1754e+07     83      8.645e+05    3.7387e+06    0.0521
    yshr_tax_ss        14     30      6      7.1754e+07     83      8.645e+05    1.4324e+07     0.199
    yshr_wage_ss       15     31      6      7.1754e+07     83      8.645e+05    3.1088e+07     0.433

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_dsvfi_results Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                          i     idx     value
                          --    ---    --------

    A_agg                  1     1       193.39
    A_agg_perhh            2     2       4.2232
    Aprime_agg             3     3       197.89
    Aprime_agg_perhh       4     4       4.3213
    Bequests_aux           5     5       2.5593
    Bequests_aux_perhh     6     6     0.055887
    C_agg                  7     7       48.789
    C_agg_perhh            8     8       1.0654
    SS_spend               9     9       2.3908
    SS_spend_perhh        10    10     0.052208
    Tax_revenues          11    12       13.735
    Tax_revenues_perhh    12    13      0.29994
    Y_inc_agg             13    14       64.627
    Y_inc_agg_perhh       14    15       1.4113

xxx tb_outcomes: all stats xxx
     OriginalVariableNames         a_ss          ap_ss         cons_ss         n_ss          y_all
    ________________________    __________    __________    __________    _________    __________

    {'mean'               }         4.2232        4.3213        1.0654       2.3554        1.4635
    {'unweighted_sum'     }           2228    8.7064e+08    8.2948e+07           21    1.3652e+08
    {'sd'                 }         6.7417         6.779        0.6899       1.4375        1.4563
    {'coefofvar'          }         1.5964        1.5687       0.64754      0.61029       0.99508
    {'gini'               }        0.68027       0.68124       0.33738       0.3128       0.44246
    {'min'                }              0             0      0.036717            1      0.038108
    {'max'                }            135         163.7        141.66            6        50.873
    {'pYis0'              }        0.12293       0.10299             0            0             0
    {'pYls0'              }              0             0             0            0             0
    {'pYgr0'              }        0.87707       0.89701             1            1             1
    {'pYisMINY'           }        0.12293       0.10299    6.7731e-07      0.36005    6.7731e-07
    {'pYisMAXY'           }     6.0111e-06    1.6708e-12             0     0.041101    1.6708e-12
    {'p0_01'              }              0             0      0.067181            1       0.07102
    {'p0_1'               }              0             0       0.10544            1       0.11346
    {'p1'                 }              0             0       0.18623            1       0.20359
    {'p5'                 }              0             0       0.27747            1       0.28173
    {'p10'                }              0             0       0.36103            1       0.35688
    {'p20'                }       0.064373      0.068222       0.49773            1       0.50299
    {'p25'                }        0.11124       0.17983       0.56413            1       0.57911
    {'p30'                }        0.26367       0.37542       0.63091            1       0.65753
```

```
{'p40'                 }       0.68544       0.84816       0.77012              2       0.83048
{'p50'                 }        1.4131        1.5883       0.91942              2        1.0325
{'p60'                 }        2.5301        2.7569        1.0845              2        1.2817
{'p70'                 }        4.1199        4.4885        1.2781              3         1.613
{'p75'                 }        5.4836        5.7144        1.3935              3        1.8306
{'p80'                 }        7.1191        7.2197        1.5293              4        2.1079
{'p90'                 }         12.56        12.096        1.9344              5        3.0419
{'p95'                 }        16.875        17.457        2.3404              5        4.0251
{'p99'                 }        30.548        31.377         3.384              6        6.8588
{'p99_9'               }        56.953        56.953        5.2437              6        14.778
{'p99_99'              }        90.439        88.534        7.4817              6        20.971
{'fl_cov_a_ss'         }        45.451        45.439        3.3942        -1.4049        4.4679
{'fl_cor_a_ss'         }             1       0.99423       0.72975       -0.14496       0.45507
{'fl_cov_ap_ss'        }        45.439        45.955        3.4956        -1.3685        5.3067
{'fl_cor_ap_ss'        }       0.99423             1       0.74743       -0.14043       0.53754
{'fl_cov_cons_ss'      }        3.3942        3.4956       0.47596        0.23909       0.76142
{'fl_cor_cons_ss'      }       0.72975       0.74743             1        0.24109       0.75787
{'fl_cov_n_ss'         }       -1.4049       -1.3685       0.23909         2.0664       0.35987
{'fl_cor_n_ss'         }      -0.14496      -0.14043       0.24109              1       0.17191
{'fl_cov_y_all'        }        4.4679        5.3067       0.76142        0.35987        2.1208
{'fl_cor_y_all'        }       0.45507       0.53754       0.75787        0.17191             1
{'fl_cov_y_head_inc'   }        3.8282        4.1045       0.55948       0.092667        1.1039
{'fl_cor_y_head_inc'   }       0.56819       0.60585       0.81146       0.064504       0.75851
{'fl_cov_y_head_earn'  }        1.8477        2.1508       0.42576        0.19287       0.96246
{'fl_cor_y_head_earn'  }       0.29785       0.34482       0.67071        0.14582       0.71827
{'fl_cov_y_spouse_inc' }       0.63967        1.2022       0.20194         0.2672        1.0169
{'fl_cor_y_spouse_inc' }       0.09937       0.18573       0.30656        0.19467       0.73129
{'fl_cov_yshr_interest'}       0.76424       0.71927      0.037996      -0.066731    -0.0094215
{'fl_cor_yshr_interest'}       0.67572       0.63246        0.3283       -0.27671     -0.038564
{'fl_cov_yshr_wage'    }      -0.77528      -0.68855    -0.0042957        0.17055       0.10767
{'fl_cor_yshr_wage'    }      -0.34062      -0.30085     -0.018443        0.35142       0.21899
{'fl_cov_yshr_SS'      }      0.011037     -0.030725     -0.033701       -0.10382      -0.09825
{'fl_cor_yshr_SS'      }     0.0069239     -0.019169       -0.2066       -0.30546      -0.28534
{'fl_cov_yshr_tax'     }      0.098159       0.10896      0.018583        0.01337      0.038535
{'fl_cor_yshr_tax'     }       0.41485       0.45797       0.76748        0.26501       0.75395
{'fl_cov_yshr_nttxss'  }      0.087122       0.13969      0.052284        0.11719       0.13679
{'fl_cor_yshr_nttxss'  }      0.050539      0.080586       0.29639        0.31882       0.36733
{'fracByP0_01'         }             0             0    5.5188e-06        0.15286    4.2239e-06
{'fracByP0_1'          }             0             0    8.2593e-05        0.15286     6.444e-05
{'fracByP1'            }             0             0     0.0013857        0.15286     0.0010994
{'fracByP5'            }             0             0      0.010292        0.15286     0.0079949
{'fracByP10'           }             0             0      0.025341        0.15286      0.018888
{'fracByP20'           }    0.00074832    0.00060951      0.065753        0.15286      0.048269
{'fracByP25'           }     0.0014123     0.0020285      0.090679        0.15286      0.066791
{'fracByP30'           }     0.0041719     0.0051595       0.11872        0.15286      0.087944
{'fracByP40'           }      0.016751       0.01877        0.1844        0.40183       0.13867
{'fracByP50'           }      0.045326      0.046338       0.26358        0.40183       0.20207
{'fracByP60'           }      0.095502      0.095716        0.3575        0.40183       0.28072
{'fracByP70'           }       0.17466       0.17847       0.46813        0.56321       0.37901
{'fracByP75'           }       0.24517       0.23715       0.53078        0.56321       0.43771
{'fracByP80'           }       0.32852       0.31134       0.59927        0.75407       0.50477
{'fracByP90'           }       0.56651       0.52814       0.75975         0.8953       0.67658
{'fracByP95'           }       0.70071        0.6954       0.85893         0.8953       0.79526
{'fracByP99'           }       0.90524       0.90259       0.96084              1       0.93132
{'fracByP99_9'         }       0.98567       0.98372       0.99419              1       0.98801
{'fracByP99_99'        }       0.99808        0.9976       0.99922              1       0.99841
```

```
% Get Matrixes
```

```
cl_st_precompute_list = {'a', 'ar_z_ctr_amz', ...
    'inc', 'inc_unemp', 'spouse_inc', 'spouse_inc_unemp', 'ref_earn_wageind_grid',...
    'ap_idx_lower_ss', 'ap_idx_higher_ss', 'ap_idx_lower_weight_ss'};
mp_controls('bl_print_precompute_verbose') = false;
[mp_precompute_res] = snw_hh_precompute(mp_params, mp_controls, cl_st_precompute_list, ap_ss, Phi_tr
```

```
Wage quintile cutoffs=0.47017     0.71433      1.0293      1.5654
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_moredense_a65zh133zs5_e2m2;SNW_MP_CONTROL=default_t
```

13.1.2 Solve for 2019 Evuvw With 0 and 1 Checks

```
% Call Function
welf_checks = 0;
[ev19_jaeemk_check0, ec19_jaeemk_check0, ev20_jaeemk_check0, ec20_jaeemk_check0] = snw_evuvw19_jaeem
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss, ap_ss, cons_ss, V_unemp, cons_unemp, mp_precompute_res);
```

```
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=0;TR=0.0017225;SNW_MP_PARAM=default_moredense_a65zh133
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=0;TR=0.0017225;xi=0.5;b=1;SNW_MP_PARAM=default_mored
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_moredense_a65zh133zs5_e2m2;SNW_MP_CONTROL=default_
Completed SNW_EVUVW19_JAEEMK_FOC;SNW_MP_PARAM=default_moredense_a65zh133zs5_e2m2;SNW_MP_CONTROL=defa
```

```
% Call Function
welf_checks = 1;
[ev19_jaeemk_check2, ec19_jaeemk_check2, ev20_jaeemk_check2, ec20_jaeemk_check2] = snw_evuvw19_jaeem
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss, ap_ss, cons_ss, V_unemp, cons_unemp, mp_precompute_res);
```

```
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=1;TR=0.0017225;SNW_MP_PARAM=default_moredense_a65zh133
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=1;TR=0.0017225;xi=0.5;b=1;SNW_MP_PARAM=default_mored
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_moredense_a65zh133zs5_e2m2;SNW_MP_CONTROL=default_
Completed SNW_EVUVW19_JAEEMK_FOC;SNW_MP_PARAM=default_moredense_a65zh133zs5_e2m2;SNW_MP_CONTROL=defa
```

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check = ev19_jaeemk_check2 - ev19_jaeemk_check0;
mn_MPC_C_gain_share_check = (ec19_jaeemk_check2 - ec19_jaeemk_check0)./(welf_checks*mp_params('TR'))
```

13.1.3 Additional Variables

Create additional Staet-Spac Arrays

```
% (n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
% Children Array
ar_kids = (1:mp_params('n_kidsgrid')) - 1;
mn_kids = zeros(1,1,1,1,1,length(ar_kids));
mn_kids(1,1,1,1,1,:) = ar_kids;
kids_ss = repmat(mn_kids, [mp_params('n_jgrid'), mp_params('n_agrid'), mp_params('n_etagrid'), ...
    mp_params('n_educgrid'), mp_params('n_marriedgrid'), 1]);
% Marital Status Arrays
ar_marital = (1:mp_params('n_marriedgrid')) - 1;
mn_marital = zeros(1,1,1,1,length(ar_marital),1);
mn_marital(1,1,1,1,:,1) = ar_marital;
marital_ss = repmat(mn_marital, [mp_params('n_jgrid'), mp_params('n_agrid'), mp_params('n_etagrid'),
    mp_params('n_educgrid'), 1, mp_params('n_kidsgrid')]);
% Educational Status Arrays
ar_educ = (1:mp_params('n_educgrid')) - 1;
mn_educ = zeros(1,1,1,length(ar_educ),1,1);
mn_educ(1,1,1,:,1,1) = ar_educ;
educ_ss = repmat(mn_educ, [mp_params('n_jgrid'), mp_params('n_agrid'), mp_params('n_etagrid'), ...
```

```
    1, mp_params('n_marriedgrid'), mp_params('n_kidsgrid')]);
% Age Array
ar_age = (1:mp_params('n_jgrid')) + 18;
mn_age = zeros(length(ar_age),1,1,1,1,1);
mn_age(:,1,1,1,1,1) = ar_age;
age_ss = repmat(mn_age, [1, mp_params('n_agrid'), mp_params('n_etagrid'), ...
    mp_params('n_educgrid'), mp_params('n_marriedgrid'), mp_params('n_kidsgrid')]);
```

13.1.4 Adjust to Probability Mass Function

```
Phi_true_1 = Phi_true./sum(Phi_true,'all');
```

13.1.5 Age Bounds

```
% 1 = 18
min_age = 1

min_age = 1

% retirement, 46+18=64, the year prior to retirement year.
max_age = 46;
```

13.1.6 Scale Statistics to Thousands of Dollars

```
a_ss = mp_dsvfi_results('a_ss')*58.056;
ap_ss = mp_dsvfi_results('ap_ss')*58.056;
c_ss = mp_dsvfi_results('cons_ss')*58.056;
n_ss = mp_dsvfi_results('n_ss');
% household head + spousal (realized) income
y_all = mp_dsvfi_results('y_all_ss')*58.056;
y_head_inc = mp_dsvfi_results('y_head_inc_ss')*58.056;
y_spouse_inc = mp_dsvfi_results('y_spouse_inc_ss')*58.056;

yshr_wage = mp_dsvfi_results('yshr_wage_ss');
yshr_SS = mp_dsvfi_results('yshr_SS_ss');
yshr_nttxss = mp_dsvfi_results('yshr_nttxss_ss');
```

13.1.7 Distributional Statistics Overall All Ages

```
% construct input data
marital_grp = marital_ss(min_age:82, :, :, : ,: ,:);
y_all_grp = y_all(min_age:82, :, :, : ,: ,:);
age_ss_grp = age_ss(min_age:82, :, :, : ,: ,:);
educ_ss_grp = educ_ss(min_age:82, :, :, : ,: ,:);
a_ss_grp = a_ss(min_age:82, :, :, : ,: ,:);
ap_ss_grp = ap_ss(min_age:82, :, :, : ,: ,:);
mn_MPC_C_gain_share_check_grp = mn_MPC_C_gain_share_check(min_age:82, :, :, :, :, :);
Phi_true_grp = Phi_true_1(min_age:82, :, :, : ,: ,:);
c_ss_grp = c_ss(min_age:82, :, :, : ,: ,:);
y_head_inc_grp = y_head_inc(min_age:82, :, :, : ,: ,:);
y_spouse_inc_grp = y_spouse_inc(min_age:82, :, :, : ,: ,:);
yshr_nttxss_grp = yshr_nttxss(min_age:82, :, :, : ,: ,:);

mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
mp_cl_ar_xyz_of_s('married') = {marital_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_all') = {y_all_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('age_ss') = {age_ss_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('educ_ss') = {educ_ss_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('a_ss') = {a_ss_grp(:), zeros(1)};
```

```
mp_cl_ar_xyz_of_s('ap_ss') = {ap_ss_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('MPC') = {mn_MPC_C_gain_share_check_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('Mass') = {Phi_true_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('c_ss') = {c_ss_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_head_inc') = {y_head_inc_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_spouse') = {y_spouse_inc_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('yshr_nttxss') = {yshr_nttxss_grp(:), zeros(1)};

mp_cl_ar_xyz_of_s('ar_st_y_name') = ["married", "y_all", "age_ss", "educ_ss", "a_ss", "ap_ss", "MPC"

% controls
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('ar_fl_percentiles') = [0.01 10 25 50 75 90 99.99];
mp_support('bl_display_final') = true;
mp_support('bl_display_detail') = false;
mp_support('bl_display_drvm2outcomes') = false;
mp_support('bl_display_drvstats') = false;
mp_support('bl_display_drvm2covcor') = false;

% Call Function
mp_cl_mt_xyz_of_s = ff_simu_stats(Phi_true_grp(:)/sum(Phi_true_grp,'all'), mp_cl_ar_xyz_of_s, mp_sup

xxx tb_outcomes: all stats xxx
    OriginalVariableNames       married         y_all        age_ss       educ_ss         a_ss
    ______________________    ___________    ___________    __________    ___________    ___________

    {'mean'              }        0.47501        84.974        47.129           0.303         245.22
    {'unweighted_sum'    }              1    7.9255e+09          4879               1     1.2935e+05
    {'sd'                }        0.49938        84.549        19.231         0.45956         391.42
    {'coefofvar'         }         1.0513         0.995       0.40805          1.5167         1.5962
    {'gini'              }        0.36718       0.44243       0.23101         0.61588        0.68023
    {'min'               }              0        2.2124            19               0              0
    {'max'               }              1        2953.5           100               1         7837.6
    {'pYis0'             }        0.52499             0             0           0.697        0.12285
    {'pYls0'             }              0             0             0               0              0
    {'pYgr0'             }        0.47501             1             1           0.303        0.87715
    {'pYisMINY'          }        0.52499     6.774e-07       0.02184           0.697        0.12285
    {'pYisMAXY'          }        0.47501     1.671e-12    0.00020326           0.303     6.0119e-06
    {'p0_01'             }              0        4.1232            19               0              0
    {'p10'               }              0        20.726            23               0              0
    {'p25'               }              0        33.631            31               0          6.458
    {'p50'               }              0        59.948            45               0          82.04
    {'p75'               }              1        106.28            62               1         318.35
    {'p90'               }              1        176.61            75               1         729.18
    {'p99_99'            }              1        1217.5           100               1         5250.6
    {'fl_cov_married'    }        0.24938        12.618    2.9987e-13        0.026842         31.201
    {'fl_cor_married'    }              1       0.29884    3.1225e-14         0.11697        0.15962
    {'fl_cov_y_all'      }         12.618        7148.6       -105.85          6.7259          15059
    {'fl_cor_y_all'      }        0.29884             1     -0.065099          0.1731        0.45504
    {'fl_cov_age_ss'     }     2.9987e-13       -105.85        369.84      5.7371e-13           2902
    {'fl_cor_age_ss'     }     3.1225e-14     -0.065099             1      6.4916e-14        0.38553
    {'fl_cov_educ_ss'    }       0.026842        6.7259    5.7371e-13         0.21119          20.13
    {'fl_cor_educ_ss'    }        0.11697        0.1731    6.4916e-14               1        0.11191
    {'fl_cov_a_ss'       }         31.201         15059          2902           20.13     1.5321e+05
    {'fl_cor_a_ss'       }        0.15962       0.45504       0.38553         0.11191              1
    {'fl_cov_ap_ss'      }          31.93         17886        2762.7          20.615     1.5316e+05
    {'fl_cor_ap_ss'      }        0.16246       0.53751       0.36501         0.11398        0.99423
    {'fl_cov_MPC'        }      -0.016733       -6.6507       -1.2778       0.0049583        -30.154
```

```
    {'fl_cor_MPC'        }       -0.13011        -0.30544         -0.258        0.041894        -0.29913
    {'fl_cov_Mass'       }    -5.1035e-07     -7.3196e-05     -2.691e-05     -2.0525e-07     -0.00031586
    {'fl_cor_Mass'       }       -0.19258        -0.16313       -0.26368       -0.084158        -0.15206
    {'fl_cov_c_ss'       }         8.8909          2566.3         57.161          4.6211           11440
    {'fl_cor_c_ss'       }        0.44452         0.75784       0.074211         0.25106         0.72974
    {'fl_cov_y_head_inc' }         1.6909          3720.9        -73.542          4.2898           12903
    {'fl_cor_y_head_inc' }       0.058359         0.75849      -0.065909         0.16088         0.56816
    {'fl_cov_y_spouse'   }         10.927          3427.7        -32.308           2.436          2155.8
    {'fl_cor_y_spouse'   }         0.3947         0.73129      -0.030304        0.095619         0.09935
    {'fl_cov_yshr_nttxss'}       0.022689           7.935        -3.2573       0.0058708          5.0323
    {'fl_cor_yshr_nttxss'}         0.1778         0.36727       -0.66283        0.049993        0.050313
    {'fracByP0_01'       }              0       4.224e-06      0.0088049               0               0
    {'fracByP10'         }              0        0.018881       0.047593               0               0
    {'fracByP25'         }              0        0.066793        0.14054               0       0.0014119
    {'fracByP50'         }              0         0.20209        0.34194               0        0.045325
    {'fracByP75'         }              1         0.43774        0.62344               1         0.24517
    {'fracByP90'         }              1          0.6766        0.82958               1         0.56651
    {'fracByP99_99'      }              1         0.99841              1               1         0.99808
```

```
tb_dist_stats_all = mp_cl_mt_xyz_of_s('tb_outcomes');
```

13.1.8 Distributional Statistics Overall 18 to 64

Statistics overall distributionally for 18 to 64 year olds.

```
% construct input data
marital_grp = marital_ss(min_age:max_age, :, :, : ,: ,:);
y_all_grp = y_all(min_age:max_age, :, :, : ,: ,:);
age_ss_grp = age_ss(min_age:max_age, :, :, : ,: ,:);
educ_ss_grp = educ_ss(min_age:max_age, :, :, : ,: ,:);
a_ss_grp = a_ss(min_age:max_age, :, :, : ,: ,:);
ap_ss_grp = ap_ss(min_age:max_age, :, :, : ,: ,:);
mn_MPC_C_gain_share_check_grp = mn_MPC_C_gain_share_check(min_age:max_age, :, :, :, :, :);
Phi_true_grp = Phi_true_1(min_age:max_age, :, :, : ,: ,:);
c_ss_grp = c_ss(min_age:max_age, :, :, : ,: ,:);
y_head_inc_grp = y_head_inc(min_age:max_age, :, :, : ,: ,:);
y_spouse_inc_grp = y_spouse_inc(min_age:max_age, :, :, : ,: ,:);
yshr_nttxss_grp = yshr_nttxss(min_age:max_age, :, :, : ,: ,:);

mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
mp_cl_ar_xyz_of_s('married') = {marital_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_all') = {y_all_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('age_ss') = {age_ss_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('educ_ss') = {educ_ss_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('a_ss') = {a_ss_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('ap_ss') = {ap_ss_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('MPC') = {mn_MPC_C_gain_share_check_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('Mass') = {Phi_true_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('c_ss') = {c_ss_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_head_inc') = {y_head_inc_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_spouse') = {y_spouse_inc_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('yshr_nttxss') = {yshr_nttxss_grp(:), zeros(1)};

mp_cl_ar_xyz_of_s('ar_st_y_name') = ["married", "y_all", "age_ss", "educ_ss", "a_ss", "ap_ss", "MPC"

% controls
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('ar_fl_percentiles') = [0.01 10 25 50 75 90 99.99];
mp_support('bl_display_final') = true;
```

```
mp_support('bl_display_detail') = false;
mp_support('bl_display_drvm2outcomes') = false;
mp_support('bl_display_drvstats') = false;
mp_support('bl_display_drvm2covcor') = false;

% Call Function
mp_cl_mt_xyz_of_s = ff_simu_stats(Phi_true_grp(:)/sum(Phi_true_grp,'all'), mp_cl_ar_xyz_of_s, mp_sup

xxx tb_outcomes: all stats xxx
    OriginalVariableNames       married         y_all        age_ss        educ_ss         a_ss
    ______________________    ___________    ___________    ___________    ___________    __________

    {'mean'              }        0.47501         95.246         39.372          0.303          194.
    {'unweighted_sum'    }              1     7.7487e+09           1909              1     1.2935e+0
    {'sd'                }        0.49938         89.631         13.105        0.45956          344.
    {'coefofvar'         }         1.0513        0.94104        0.33285         1.5167         1.771
    {'gini'              }        0.36718        0.42428        0.18859        0.61588        0.7157
    {'min'               }              0         2.2124             19              0
    {'max'               }              1         2953.5             64              1         7837.
    {'pYis0'             }        0.52499              0              0          0.697        0.1462
    {'pYls0'             }              0              0              0              0
    {'pYgr0'             }        0.47501              1              1          0.303        0.8537
    {'pYisMINY'          }        0.52499     8.6135e-07       0.027771          0.697        0.1462
    {'pYisMAXY'          }        0.47501     2.1248e-12       0.015675          0.303     5.4766e-0
    {'p0_01'             }              0         3.9581             19              0
    {'p10'               }              0         25.069             22              0
    {'p25'               }              0         40.654             28              0         3.737
    {'p50'               }              0          69.57             38              0         51.66
    {'p75'               }              1         119.76             50              1         239.1
    {'p90'               }              1          192.9             58              1         588.4
    {'p99_99'            }              1         1249.3             64              1         4707.
    {'fl_cov_married'    }        0.24938         13.756      2.335e-13       0.026842          25.2
    {'fl_cor_married'    }              1        0.30733     3.5679e-14        0.11697        0.1468
    {'fl_cov_y_all'      }         13.756         8033.6         270.03         7.5617          1785
    {'fl_cor_y_all'      }        0.30733              1        0.22988        0.18358        0.5781
    {'fl_cov_age_ss'     }      2.335e-13         270.03         171.75     4.3386e-15         2241.
    {'fl_cor_age_ss'     }     3.5679e-14        0.22988              1      7.204e-16        0.4964
    {'fl_cov_educ_ss'    }       0.026842         7.5617     4.3386e-15        0.21119         15.47
    {'fl_cor_educ_ss'    }        0.11697        0.18358      7.204e-16              1       0.09776
    {'fl_cov_a_ss'       }          25.27          17852         2241.5         15.478     1.1868e+0
    {'fl_cor_a_ss'       }        0.14689        0.57814        0.49648       0.097766
    {'fl_cov_ap_ss'      }         26.783          20993         2328.9         16.562     1.2238e+0
    {'fl_cor_ap_ss'      }        0.15001        0.65507        0.49704         0.1008        0.9935
    {'fl_cov_MPC'        }      -0.017248        -8.3845        -1.4685      0.0073384        -27.85
    {'fl_cor_MPC'        }       -0.12735       -0.34491       -0.41317       0.058877       -0.2981
    {'fl_cov_Mass'       }    -6.2681e-07    -0.00010581    -2.2759e-05    -2.2235e-07    -0.0003165
    {'fl_cor_Mass'       }       -0.21171       -0.19912       -0.29292      -0.081609         -0.15
    {'fl_cov_c_ss'       }         8.9405         2911.4          117.7         4.6429         9782.
    {'fl_cor_c_ss'       }        0.44676        0.81058        0.22412        0.25211        0.7086
    {'fl_cov_y_head_inc' }         1.5449         4083.5         215.29         4.8213          1513
    {'fl_cor_y_head_inc' }       0.050457        0.74307        0.26794        0.17111        0.7164
    {'fl_cov_y_spouse'   }         12.211         3950.1         54.733         2.7405         2719.
    {'fl_cor_y_spouse'   }        0.40608         0.7319       0.069359       0.099033         0.131
    {'fl_cov_yshr_nttxss'}      0.0064334         2.2345        0.12412      0.0029398         5.703
    {'fl_cor_yshr_nttxss'}        0.38567        0.74633        0.28352         0.1915        0.4956
    {'fracByP0_01'       }              0     3.6432e-06       0.013402              0
    {'fracByP10'         }              0       0.018969       0.056893              0
    {'fracByP25'         }              0       0.070975        0.15748              0      0.001135
```

```
    {'fracByP50'        }              0        0.21374        0.35932              0       0.03404
    {'fracByP75'        }              1        0.45357        0.64274              1        0.2134
    {'fracByP90'        }              1        0.69054        0.84608              1        0.5149
    {'fracByP99_99'     }              1        0.99855              1              1        0.9971
```

```
tb_dist_stats_all_18to64 = mp_cl_mt_xyz_of_s('tb_outcomes');
```

13.1.9 Distributional Statistics By Kids Count

Various statistics, including MPC (of the first check) by Children Count

```
it_row_ctr = 0;
for it_ctr=1:mp_params('n_kidsgrid')
    display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);
    display(['kids =' num2str(ar_kids(it_ctr))]);
    display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);

    % construct input data
    marital_grp = marital_ss(min_age:max_age, :, :, : ,: ,it_ctr);
    y_all_grp = y_all(min_age:max_age, :, :, : ,: ,it_ctr);
    age_ss_grp = age_ss(min_age:max_age, :, :, : ,: ,it_ctr);
    educ_ss_grp = educ_ss(min_age:max_age, :, :, : ,: ,it_ctr);
    a_ss_grp = a_ss(min_age:max_age, :, :, : ,: ,it_ctr);
    ap_ss_grp = ap_ss(min_age:max_age, :, :, : ,: ,it_ctr);
    mn_MPC_C_gain_share_check_grp = mn_MPC_C_gain_share_check(min_age:max_age, :, :, :, :, it_ctr);
    Phi_true_grp = Phi_true_1(min_age:max_age, :, :, : ,: ,it_ctr);
    c_ss_grp = c_ss(min_age:max_age, :, :, : ,: ,it_ctr);
    y_head_inc_grp = y_head_inc(min_age:max_age, :, :, : ,: ,it_ctr);
    y_spouse_inc_grp = y_spouse_inc(min_age:max_age, :, :, : ,: ,it_ctr);
    yshr_nttxss_grp = yshr_nttxss(min_age:max_age, :, :, : ,: ,it_ctr);

    mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
    mp_cl_ar_xyz_of_s('married') = {marital_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('y_all') = {y_all_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('age_ss') = {age_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('educ_ss') = {educ_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('a_ss') = {a_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('ap_ss') = {ap_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('MPC') = {mn_MPC_C_gain_share_check_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('Mass') = {Phi_true_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('c_ss') = {c_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('y_head_inc') = {y_head_inc_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('y_spouse') = {y_spouse_inc_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('yshr_nttxss') = {yshr_nttxss_grp(:), zeros(1)};

    mp_cl_ar_xyz_of_s('ar_st_y_name') = ["married", "y_all", "age_ss", "educ_ss", "a_ss", "ap_ss", "

    % controls
    mp_support = containers.Map('KeyType','char', 'ValueType','any');
    mp_support('ar_fl_percentiles') = [0.01 10 25 50 75 90 99.99];
    mp_support('bl_display_final') = true;
    mp_support('bl_display_detail') = false;
    mp_support('bl_display_drvm2outcomes') = false;
    mp_support('bl_display_drvstats') = false;
    mp_support('bl_display_drvm2covcor') = false;

    % Call Function
    mp_cl_mt_xyz_of_s = ff_simu_stats(Phi_true_grp(:)/sum(Phi_true_grp,'all'), mp_cl_ar_xyz_of_s, mp
```

```
    it_kids = ar_kids(it_ctr);

    tb_dist_stats = mp_cl_mt_xyz_of_s('tb_outcomes');

    fl_married_mean = tb_dist_stats{"married", "mean"};

    fl_age_mean = tb_dist_stats{"age_ss", "mean"};
    fl_age_p50 = tb_dist_stats{"age_ss", "p50"};

    fl_educ_mean = tb_dist_stats{"educ_ss", "mean"};

    fl_a_mean = tb_dist_stats{"a_ss", "mean"};
    fl_a_p50 = tb_dist_stats{"a_ss", "p50"};

    fl_ap_mean = tb_dist_stats{"ap_ss", "mean"};
    fl_ap_p50 = tb_dist_stats{"ap_ss", "p50"};

    fl_y_all_mean = tb_dist_stats{"y_all", "mean"};
    fl_y_all_p50 = tb_dist_stats{"y_all", "p50"};

    fl_mpc_mean = tb_dist_stats{"MPC", "mean"};
    fl_mpc_p50 = tb_dist_stats{"MPC", "p50"};

    fl_mass = tb_dist_stats{"Mass", "unweighted_sum"};

    fl_c_ss_mean = tb_dist_stats{"c_ss", "mean"};
    fl_c_ss_p50 = tb_dist_stats{"c_ss", "p50"};

    fl_y_head_inc_mean = tb_dist_stats{"y_head_inc", "mean"};
    fl_y_spouse_mean = tb_dist_stats{"y_spouse", "mean"};

    ar_store_stats = [it_kids, fl_married_mean, ...
        fl_age_mean, fl_age_p50, fl_educ_mean, ...
        fl_a_mean, fl_a_p50, fl_ap_mean, fl_ap_p50, ...
        fl_y_all_mean, fl_y_all_p50, ...
        fl_mpc_mean, fl_mpc_p50, ...
        fl_mass, ...
        fl_c_ss_mean, fl_c_ss_p50, ...
        fl_y_head_inc_mean, fl_y_spouse_mean];

    it_row_ctr = it_row_ctr + 1;

    if (it_row_ctr>1)
        mt_store_stats_by_k = [mt_store_stats_by_k;ar_store_stats];
    else
        mt_store_stats_by_k = [ar_store_stats];
    end
end

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
kids =0
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames      married        y_all        age_ss       educ_ss        a_ss
    ______________________    ___________    ___________    ___________    __________    ___________

    {'mean'              }       0.34092         95.696         42.81       0.29837         267.84
    {'unweighted_sum'    }             1     1.9045e+09          1909             1     1.2935e+05
```

{'sd' }	0.47402	93.188	14.55	0.45754	413.66
{'coefofvar' }	1.3904	0.97379	0.33987	1.5335	1.5444
{'gini' }	0.56028	0.43559	0.18997	0.62263	0.66933
{'min' }	0	2.2124	19	0	0
{'max' }	1	2953.5	64	1	7837.6
{'pYis0' }	0.65908	0	0	0.70163	0.10437
{'pYls0' }	0	0	0	0	0
{'pYgr0' }	0.34092	1	1	0.29837	0.89563
{'pYisMINY' }	0.65908	1.2783e-06	0.038791	0.70163	0.10437
{'pYisMAXY' }	0.34092	4.4127e-12	0.029551	0.29837	1.0023e-05
{'p0_01' }	0	3.8399	19	0	0
{'p10' }	0	23.75	21	0	0
{'p25' }	0	39.249	29	0	10.255
{'p50' }	0	68.775	45	0	100.91
{'p75' }	1	119.82	56	1	363.77
{'p90' }	1	198.36	61	1	729.18
{'p99_99' }	1	1317.3	64	1	5250.6
{'fl_cov_married' }	0.22469	15.781	0.41952	0.027901	47.935
{'fl_cor_married' }	1	0.35725	0.060827	0.12864	0.24447
{'fl_cov_y_all' }	15.781	8684	314.15	6.9889	24515
{'fl_cor_y_all' }	0.35725	1	0.23169	0.16391	0.63596
{'fl_cov_age_ss' }	0.41952	314.15	211.7	-0.40705	2895.3
{'fl_cor_age_ss' }	0.060827	0.23169	1	-0.061144	0.48104
{'fl_cov_educ_ss' }	0.027901	6.9889	-0.40705	0.20934	17.081
{'fl_cor_educ_ss' }	0.12864	0.16391	-0.061144	1	0.090246
{'fl_cov_a_ss' }	47.935	24515	2895.3	17.081	1.7111e+05
{'fl_cor_a_ss' }	0.24447	0.63596	0.48104	0.090246	1
{'fl_cov_ap_ss' }	50.8	28037	2996.5	18.304	1.7633e+05
{'fl_cor_ap_ss' }	0.25021	0.70243	0.48082	0.093396	0.9952
{'fl_cov_MPC' }	-0.0040817	-5.3961	-1.3578	0.016362	-22.104
{'fl_cor_MPC' }	-0.040228	-0.27052	-0.43598	0.16707	-0.24964
{'fl_cov_Mass' }	-6.3926e-07	-0.00016151	-4.6365e-05	-2.596e-07	-0.00069632
{'fl_cor_Mass' }	-0.16893	-0.2171	-0.39918	-0.071073	-0.21087
{'fl_cov_c_ss' }	9.1002	3017.9	137.75	4.0645	13300
{'fl_cor_c_ss' }	0.48715	0.82178	0.24023	0.22541	0.81588
{'fl_cov_y_head_inc' }	2.4027	4493.9	256.65	4.356	20579
{'fl_cor_y_head_inc' }	0.078733	0.74906	0.27398	0.14788	0.77274
{'fl_cov_y_spouse' }	13.378	4190.1	57.502	2.6329	3936
{'fl_cor_y_spouse' }	0.45539	0.72551	0.063767	0.092849	0.15353
{'fl_cov_yshr_nttxss'}	0.0065704	2.3727	0.15463	0.0025679	8.1118
{'fl_cor_yshr_nttxss'}	0.40207	0.73855	0.30827	0.1628	0.56882
{'fracByP0_01' }	0	3.5026e-06	0.017216	0	0
{'fracByP10' }	0	0.017915	0.047317	0	0
{'fracByP25' }	0	0.067497	0.14043	0	0.002351
{'fracByP50' }	0	0.20658	0.35779	0	0.050647
{'fracByP75' }	1	0.44409	0.66951	1	0.26381
{'fracByP90' }	1	0.6831	0.86922	1	0.54208
{'fracByP99_99' }	1	0.99848	1	1	0.99785

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
kids =1
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx

OriginalVariableNames	married	y_all	age_ss	educ_ss	a_ss
{'mean' }	0.48303	94.687	37.46	0.31392	163.0
{'unweighted_sum' }	1	1.7814e+09	1909	1	1.2935e+0

```
    {'sd'                 }         0.49971          90.675          12.413         0.46408          298.8
    {'coefofvar'          }          1.0345         0.95763         0.33137          1.4784          1.833
    {'gini'               }         0.35621         0.42949         0.18779         0.59992         0.7288
    {'min'                }               0          2.2124              19               0
    {'max'                }               1          2715.2              64               1          7837.
    {'pYis0'              }         0.51697               0               0         0.68608         0.1663
    {'pYls0'              }               0               0               0               0
    {'pYgr0'              }         0.48303               1               1         0.31392         0.8336
    {'pYisMINY'           }         0.51697      8.7082e-07        0.032554         0.68608         0.1663
    {'pYisMAXY'           }         0.48303      2.5175e-12       0.0061116         0.31392     2.7005e-0
    {'p0_01'              }               0          3.9202              19               0
    {'p10'                }               0          24.541              21               0
    {'p25'                }               0          39.852              26               0          1.913
    {'p50'                }               0          68.409              37               0          39.79
    {'p75'                }               1          118.73              47               1          205.0
    {'p90'                }               1          193.79              55               1          467.1
    {'p99_99'             }               1          1212.2              64               1          4203.
    {'fl_cov_married'     }         0.24971          14.899         0.69339        0.029149            35.
    {'fl_cor_married'     }               1         0.32882         0.11178         0.12569         0.2376
    {'fl_cov_y_all'       }          14.899          8221.9           296.2          7.6969           1569
    {'fl_cor_y_all'       }         0.32882               1         0.26316         0.18291         0.5789
    {'fl_cov_age_ss'      }         0.69339           296.2          154.09         0.15644          1774.
    {'fl_cor_age_ss'      }         0.11178         0.26316               1        0.027156         0.4783
    {'fl_cov_educ_ss'     }        0.029149          7.6969         0.15644         0.21537          16.18
    {'fl_cor_educ_ss'     }         0.12569         0.18291        0.027156               1         0.1166
    {'fl_cov_a_ss'        }            35.5           15691          1774.9          16.186           8933
    {'fl_cor_a_ss'        }         0.23769         0.57898         0.47838         0.11669
    {'fl_cov_ap_ss'       }          38.015           18932          1842.5          17.321           9190
    {'fl_cor_ap_ss'       }         0.24516         0.67285         0.47835         0.12028          0.990
    {'fl_cov_MPC'         }       -0.029849         -9.6826         -1.3781       0.0095259         -25.74
    {'fl_cor_MPC'         }        -0.21465        -0.38374        -0.39896        0.073764        -0.3095
    {'fl_cov_Mass'        }     -1.9886e-07     -6.1897e-05     -1.3655e-05     -1.6868e-07     -0.0001802
    {'fl_cor_Mass'        }        -0.14586        -0.25019        -0.40317        -0.13322        -0.2210
    {'fl_cov_c_ss'        }          8.8069          2953.1          157.35          4.6956          9278.
    {'fl_cor_c_ss'        }         0.43769         0.80882         0.31481         0.25128         0.7709
    {'fl_cov_y_head_inc'  }           2.065            3941          206.32          4.8059           1277
    {'fl_cor_y_head_inc'  }        0.069081         0.72657         0.27785         0.17312         0.7142
    {'fl_cov_y_spouse'    }          12.834          4280.8          89.888           2.891          2920.
    {'fl_cor_y_spouse'    }          0.4103         0.75422         0.11568        0.099519          0.156
    {'fl_cov_yshr_nttxss'}        0.0069685          2.2798         0.13506       0.0030572          5.020
    {'fl_cor_yshr_nttxss'}          0.41262         0.74395         0.32195         0.19492         0.4970
    {'fracByP0_01'        }               0      3.7241e-06        0.016512               0
    {'fracByP10'          }               0        0.018692        0.055462               0
    {'fracByP25'          }               0        0.069801          0.1535               0     0.0006990
    {'fracByP50'          }               0         0.21065         0.37529               0       0.02806
    {'fracByP75'          }               1          0.4485          0.6393               1         0.2118
    {'fracByP90'          }               1         0.68715         0.85347               1         0.4747
    {'fracByP99_99'       }               1         0.99858               1               1         0.9968

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
kids =2
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       married          y_all          age_ss         educ_ss           a_ss
    ______________________    ___________     ___________     ___________     ___________     __________

    {'mean'               }         0.58436          95.419          35.807         0.30789         124.28
    {'unweighted_sum'     }               1      1.6966e+09            1909               1     1.2935e+05
```

```
    {'sd'                 }        0.49283         87.576         10.518        0.46162         238.29
    {'coefofvar'          }        0.84337        0.91781        0.29375         1.4993         1.9174
    {'gini'               }        0.22818        0.41656        0.16465        0.60873        0.73697
    {'min'                }              0         2.2124             19              0              0
    {'max'                }              1         2551.1             64              1         7837.6
    {'pYis0'              }        0.41564              0              0        0.69211         0.1963
    {'pYls0'              }              0              0              0              0              0
    {'pYgr0'              }        0.58436              1              1        0.30789         0.8037
    {'pYisMINY'           }        0.41564     4.0938e-07       0.014906        0.69211         0.1963
    {'pYisMAXY'           }        0.58436     1.0736e-12      0.0019534        0.30789     9.1954e-07
    {'p0_01'              }              0         4.1232             19              0              0
    {'p10'                }              0         26.204             23              0              0
    {'p25'                }              0         41.871             27              0        0.80724
    {'p50'                }              1         70.257             35              0         29.898
    {'p75'                }              1         120.84             43              1         146.89
    {'p90'                }              1         190.32             51              1         363.77
    {'p99_99'             }              1         1122.5             64              1         3737.2
    {'fl_cov_married'     }        0.24288         12.863        0.51579       0.025827         25.491
    {'fl_cor_married'     }              1        0.29802       0.099501        0.11352        0.21706
    {'fl_cov_y_all'       }         12.863         7669.6         228.36         8.3133          11413
    {'fl_cor_y_all'       }        0.29802              1         0.2479        0.20564        0.54689
    {'fl_cov_age_ss'      }        0.51579         228.36         110.63        0.45675         1116.6
    {'fl_cor_age_ss'      }       0.099501         0.2479              1       0.094068        0.44549
    {'fl_cov_educ_ss'     }       0.025827         8.3133        0.45675        0.21309         15.009
    {'fl_cor_educ_ss'     }        0.11352        0.20564       0.094068              1        0.13644
    {'fl_cov_a_ss'        }         25.491          11413         1116.6         15.009          56783
    {'fl_cor_a_ss'        }        0.21706        0.54689        0.44549        0.13644              1
    {'fl_cov_ap_ss'       }         27.147          14327         1160.8         16.023          58304
    {'fl_cor_ap_ss'       }        0.22214        0.65975        0.44505        0.13997         0.9867
    {'fl_cov_MPC'         }      -0.055633        -12.184        -1.1929     -0.0029873        -25.836
    {'fl_cor_MPC'         }       -0.35573       -0.43842        -0.3574      -0.020393       -0.34167
    {'fl_cov_Mass'        }    -4.6541e-07    -7.3755e-05    -9.0248e-06    -2.4395e-07     -0.0001563
    {'fl_cor_Mass'        }       -0.32688       -0.29151       -0.29699       -0.18292       -0.22704
    {'fl_cov_c_ss'        }         8.1321         2864.5         129.17         5.2868         7092.7
    {'fl_cor_c_ss'        }        0.40653        0.80585        0.30257        0.28216        0.73333
    {'fl_cov_y_head_inc'  }         1.6658         3681.5         154.09         5.3399         9372.8
    {'fl_cor_y_head_inc'  }       0.058412        0.72644        0.25315         0.1999        0.67972
    {'fl_cov_y_spouse'    }         11.197         3988.2         74.272         2.9734         2040.1
    {'fl_cor_y_spouse'    }        0.37578        0.75322        0.11679        0.10654         0.1416
    {'fl_cov_yshr_nttxss'}       0.0064177         2.1485        0.10113      0.0033688         3.5399
    {'fl_cor_yshr_nttxss'}         0.39977        0.75315        0.29517        0.22404        0.45606
    {'fracByP0_01'        }              0     3.7828e-06      0.0079094              0              0
    {'fracByP10'          }              0       0.019866       0.075143              0              0
    {'fracByP25'          }              0       0.073594        0.17417              0     0.00024523
    {'fracByP50'          }              1        0.21851        0.40471              0       0.027182
    {'fracByP75'          }              1         0.4596        0.65078              1        0.20572
    {'fracByP90'          }              1        0.69638        0.85987              1        0.47333
    {'fracByP99_99'       }              1        0.99869              1              1        0.99695

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
kids =3
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames      married         y_all         age_ss        educ_ss          a_ss
    _____________________    ___________    ___________    ___________    ___________    __________

    {'mean'               }        0.69032         96.012         35.356        0.30365          101.
    {'unweighted_sum'     }              1     1.6091e+09           1909              1     1.2935e+0
```

```
    {'sd'                 }       0.46236          83.53         9.1314        0.45983         196.5
    {'coefofvar'          }       0.66978        0.86999        0.25827         1.5143         1.940
    {'gini'               }       0.12198        0.40117        0.14344        0.61493         0.729
    {'min'                }             0         2.2124             19              0
    {'max'                }             1         2381.6             64              1         7837.
    {'pYis0'              }       0.30968              0              0        0.69635        0.1917
    {'pYls0'              }             0              0              0              0
    {'pYgr0'              }       0.69032              1              1        0.30365        0.8082
    {'pYisMINY'           }       0.30968      2.133e-07       0.007718        0.69635        0.1917
    {'pYisMAXY'           }       0.69032     3.4711e-13     0.00070368        0.30365    3.1947e-0
    {'p0_01'              }             0         4.4187             19              0
    {'p10'                }             0         28.136             24              0
    {'p25'                }             0         44.054             28              0        0.8072
    {'p50'                }             1         72.443             34              0         29.89
    {'p75'                }             1         122.12             42              1         100.9
    {'p90'                }             1          185.9             48              1         276.8
    {'p99_99'             }             1         1027.1             64              1         3306.
    {'fl_cov_married'     }       0.21378         9.9452        0.39867        0.02286         16.46
    {'fl_cor_married'     }             1        0.25751       0.094427        0.10752        0.1811
    {'fl_cov_y_all'       }        9.9452         6977.2         176.66         8.4101         8663.
    {'fl_cor_y_all'       }       0.25751              1        0.23161        0.21896         0.527
    {'fl_cov_age_ss'      }       0.39867         176.66         83.382        0.55101          713.
    {'fl_cor_age_ss'      }      0.094427        0.23161              1        0.13123         0.397
    {'fl_cov_educ_ss'     }       0.02286         8.4101        0.55101        0.21145         12.95
    {'fl_cor_educ_ss'     }       0.10752        0.21896        0.13123              1        0.1433
    {'fl_cov_a_ss'        }        16.463         8663.4          713.9         12.958          3864
    {'fl_cor_a_ss'        }       0.18113         0.5276         0.3977        0.14334
    {'fl_cov_ap_ss'       }        17.437          11197         743.03         13.851          3960
    {'fl_cor_ap_ss'       }         0.184        0.65402          0.397        0.14696        0.9828
    {'fl_cov_MPC'         }     -0.061242         -11.95       -0.93092     -0.0093462        -21.83
    {'fl_cor_MPC'         }      -0.42463       -0.45863       -0.32683      -0.065159       -0.3561
    {'fl_cov_Mass'        }   -2.6557e-07    -3.5455e-05    -3.3149e-06    -1.3715e-07   -6.3012e-0
    {'fl_cor_Mass'        }      -0.38696       -0.28596       -0.24457       -0.20093       -0.2159
    {'fl_cov_c_ss'        }        6.6057         2725.3            105         5.4818         5577.
    {'fl_cor_c_ss'        }       0.35578        0.81251        0.28636        0.29687         0.706
    {'fl_cov_y_head_inc'  }        1.3302         3539.8         118.35          5.592         7371.
    {'fl_cor_y_head_inc'  }       0.05051          0.744        0.22755         0.2135        0.6583
    {'fl_cov_y_spouse'    }        8.6149         3437.4         58.307         2.8181         1291.
    {'fl_cor_y_spouse'    }        0.3324        0.73415        0.11391        0.10933        0.1172
    {'fl_cov_yshr_nttxss'}      0.0052966         1.9936       0.078236      0.0034153         2.582
    {'fl_cor_yshr_nttxss'}        0.36767        0.76604        0.27499        0.23838        0.4216
    {'fracByP0_01'        }             0     4.0037e-06      0.0041476              0
    {'fracByP10'          }             0       0.021316       0.072166              0
    {'fracByP25'          }             0       0.078153        0.18337              0     0.0003122
    {'fracByP50'          }             1        0.22773        0.39789              0       0.03805
    {'fracByP75'          }             1        0.47322        0.69228              1        0.1876
    {'fracByP90'          }             1         0.7065           0.86              1        0.4594
    {'fracByP99_99'       }             1         0.9988              1              1        0.9967

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
kids =4
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       married          y_all         age_ss        educ_ss          a_ss
    ______________________    ___________    ___________    ___________    ___________    __________

    {'mean'               }       0.78724         91.676         35.383        0.29511         81.61
    {'unweighted_sum'     }             1     1.4702e+09           1909              1    1.2935e+0
```

```
    {'sd'                 }       0.40926         74.133          7.9178        0.45609          164.
    {'coefofvar'          }       0.51987        0.80864         0.22378         1.5455          2.013
    {'gini'               }      0.054374        0.38168         0.12297        0.62738         0.7274
    {'min'                }             0         2.2124              19              0
    {'max'                }             1         2113.2              64              1          7837.
    {'pYis0'              }       0.21276              0               0        0.70489         0.1891
    {'pYls0'              }             0              0               0              0
    {'pYgr0'              }       0.78724              1               1        0.29511         0.8108
    {'pYisMINY'           }       0.21276    9.2536e-08       0.0035072        0.70489         0.1891
    {'pYisMAXY'           }       0.78724    2.0254e-13      0.00027556        0.29511     1.1672e-0
    {'p0_01'              }             0         4.7807              19              0
    {'p10'                }             0          29.13              26              0
    {'p25'                }             1          44.24              29              0         0.8072
    {'p50'                }             1           71.8              35              0          29.89
    {'p75'                }             1         115.49              41              1           82.0
    {'p90'                }             1         172.56              46              1          239.1
    {'p99_99'             }             1         888.01              64              1          2910.
    {'fl_cov_married'     }       0.16749         5.9174         0.25239       0.018555          8.620
    {'fl_cor_married'     }             1        0.19504        0.077888       0.099404          0.128
    {'fl_cov_y_all'       }        5.9174         5495.7          126.24         7.9495          6630.
    {'fl_cor_y_all'       }       0.19504              1         0.21507        0.23511         0.5443
    {'fl_cov_age_ss'      }       0.25239         126.24          62.692        0.61699           463.
    {'fl_cor_age_ss'      }      0.077888        0.21507               1        0.17085         0.3564
    {'fl_cov_educ_ss'     }      0.018555         7.9495         0.61699        0.20802          10.80
    {'fl_cor_educ_ss'     }      0.099404        0.23511         0.17085              1         0.1442
    {'fl_cov_a_ss'        }        8.6206         6630.3           463.7         10.809           2699
    {'fl_cor_a_ss'        }        0.1282        0.54435         0.35644        0.14424
    {'fl_cov_ap_ss'       }        8.7102         8295.7          479.98         11.425           2761
    {'fl_cor_ap_ss'       }       0.12468        0.65556         0.35513        0.14675         0.9845
    {'fl_cov_MPC'         }      -0.04739        -10.199        -0.82379      -0.011367         -17.74
    {'fl_cor_MPC'         }      -0.39132       -0.46494         -0.3516      -0.084227        -0.3649
    {'fl_cov_Mass'        }   -1.0116e-07    -1.7179e-05     -1.6822e-06    -8.6576e-08     -3.0647e-0
    {'fl_cor_Mass'        }      -0.30657       -0.28741        -0.26349       -0.23543        -0.2313
    {'fl_cov_c_ss'        }        4.4325         2483.7          79.686          5.416          4382.
    {'fl_cor_c_ss'        }       0.27632        0.85476         0.25676        0.30296          0.680
    {'fl_cov_y_head_inc'  }       0.92362         3408.6          90.849         5.7717          5998.
    {'fl_cor_y_head_inc'  }       0.03994        0.81373         0.20306        0.22396         0.6461
    {'fl_cov_y_spouse'    }        4.9938         2087.1          35.394         2.1778          631.8
    {'fl_cor_y_spouse'    }       0.28208        0.65082         0.10334        0.11038        0.08889
    {'fl_cov_yshr_nttxss'}      0.0035289         1.7418        0.059073      0.0033945          1.929
    {'fl_cor_yshr_nttxss'}        0.29013        0.79058         0.25104        0.25043         0.3950
    {'fracByP0_01'        }             0    4.5191e-06       0.0018833              0
    {'fracByP10'          }             0      0.023539         0.08684              0
    {'fracByP25'          }             1      0.083914         0.18322              0      0.0003800
    {'fracByP50'          }             1       0.24018         0.44948              0        0.05909
    {'fracByP75'          }             1       0.48959         0.71178              1         0.2009
    {'fracByP90'          }             1       0.71753         0.86506              1         0.4875
    {'fracByP99_99'       }             1        0.9989               1              1         0.9960
```

13.1.10 Distributional Statistics By Marital Status and Kids Count

Various statistics, including MPC (of the first check) by Marital Status and Kids COunt

```
it_row_ctr = 0;
for it_marry_ctr=1:mp_params('n_marriedgrid')

    display(['']);
    display(['']);
    display(['-------------------------------']);
```

```
display(['-------------------------------']);
display(['-------------------------------']);
display(['-------------------------------']);
display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);
display(['Marital =' num2str(ar_marital(it_marry_ctr))]);
display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);
display(['-------------------------------']);
display(['-------------------------------']);

for it_kids_ctr=1:mp_params('n_kidsgrid')
    display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);
    display(['Marital =' num2str(ar_marital(it_marry_ctr)) ' and kids =' num2str(ar_kids(it_kids
    display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);

    % construct input data
    y_all_grp = y_all(min_age:max_age, :, :, : ,it_marry_ctr ,it_ctr);
    age_ss_grp = age_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
    educ_ss_grp = educ_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
    a_ss_grp = a_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
    ap_ss_grp = ap_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
    mn_MPC_C_gain_share_check_grp = mn_MPC_C_gain_share_check(min_age:max_age, :, :, :, it_marry
    Phi_true_grp = Phi_true_1(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
    c_ss_grp = c_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
    y_head_inc_grp = y_head_inc(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
    y_spouse_inc_grp = y_spouse_inc(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
    yshr_nttxss_grp = yshr_nttxss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);

    mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
    mp_cl_ar_xyz_of_s('y_all') = {y_all_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('age_ss') = {age_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('educ_ss') = {educ_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('a_ss') = {a_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('ap_ss') = {ap_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('MPC') = {mn_MPC_C_gain_share_check_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('Mass') = {Phi_true_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('c_ss') = {c_ss_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('y_head_inc') = {y_head_inc_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('y_spouse') = {y_spouse_inc_grp(:), zeros(1)};
    mp_cl_ar_xyz_of_s('yshr_nttxss') = {yshr_nttxss_grp(:), zeros(1)};

    mp_cl_ar_xyz_of_s('ar_st_y_name') = ["y_all", "age_ss", "educ_ss", "a_ss", "ap_ss", "MPC", "

    % controls
    mp_support = containers.Map('KeyType','char', 'ValueType','any');
    mp_support('ar_fl_percentiles') = [0.01 10 25 50 75 90 99.99];
    mp_support('bl_display_final') = true;
    mp_support('bl_display_detail') = false;
    mp_support('bl_display_drvm2outcomes') = false;
    mp_support('bl_display_drvstats') = false;
    mp_support('bl_display_drvm2covcor') = false;

    % Call Function
    mp_cl_mt_xyz_of_s = ff_simu_stats(Phi_true_grp(:)/sum(Phi_true_grp,'all'), mp_cl_ar_xyz_of_s

    it_marital = ar_marital(it_marry_ctr);
    it_kids = ar_kids(it_kids_ctr);

    tb_dist_stats = mp_cl_mt_xyz_of_s('tb_outcomes');
```

```
        fl_age_mean = tb_dist_stats{"age_ss", "mean"};
        fl_age_p50 = tb_dist_stats{"age_ss", "p50"};

        fl_educ_mean = tb_dist_stats{"educ_ss", "mean"};

        fl_a_mean = tb_dist_stats{"a_ss", "mean"};
        fl_a_p50 = tb_dist_stats{"a_ss", "p50"};

        fl_ap_mean = tb_dist_stats{"ap_ss", "mean"};
        fl_ap_p50 = tb_dist_stats{"ap_ss", "p50"};

        fl_y_all_mean = tb_dist_stats{"y_all", "mean"};
        fl_y_all_p50 = tb_dist_stats{"y_all", "p50"};

        fl_mpc_mean = tb_dist_stats{"MPC", "mean"};
        fl_mpc_p50 = tb_dist_stats{"MPC", "p50"};

        fl_mass = tb_dist_stats{"Mass", "unweighted_sum"};

        fl_c_ss_mean = tb_dist_stats{"c_ss", "mean"};
        fl_c_ss_p50 = tb_dist_stats{"c_ss", "p50"};

        fl_y_head_inc_mean = tb_dist_stats{"y_head_inc", "mean"};
        fl_y_spouse_mean = tb_dist_stats{"y_spouse", "mean"};

        ar_store_stats = [it_marital, it_kids, ...
            fl_age_mean, fl_age_p50, fl_educ_mean, ...
            fl_a_mean, fl_a_p50, fl_ap_mean, fl_ap_p50, ...
            fl_y_all_mean, fl_y_all_p50, ...
            fl_mpc_mean, fl_mpc_p50, ...
            fl_mass, ...
            fl_c_ss_mean, fl_c_ss_p50, ...
            fl_y_head_inc_mean, fl_y_spouse_mean];

        it_row_ctr = it_row_ctr + 1;

        if (it_row_ctr>1)
            mt_store_stats_by_mk = [mt_store_stats_by_mk;ar_store_stats];
        else
            mt_store_stats_by_mk = [ar_store_stats];
        end
    end
end
```

```
  0x0 empty char array

  0x0 empty char array

-------------------------------
-------------------------------
-------------------------------
-------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-------------------------------
-------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
Marital =0 and kids =0
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       y_all        age_ss       educ_ss         a_ss          ap_ss
    ______________________    __________    __________    __________    __________    __________

    {'mean'               }       71.752        42.174       0.25604        195.11        208.1
    {'unweighted_sum'     }   1.7831e+08          1909             1    1.2935e+05    1.6068e+0
    {'sd'                 }       62.288        14.196       0.43644        339.77        352.4
    {'coefofvar'          }       0.8681       0.33661        1.7046        1.7414        1.693
    {'gini'               }      0.40852        0.1892       0.68372        0.7026        0.6989
    {'min'                }       2.2124            19             0             0
    {'max'                }       1414.1            64             1        7837.6        8386.
    {'pYis0'              }            0             0       0.74396       0.11911        0.08185
    {'pYls0'              }            0             0             0             0
    {'pYgr0'              }            1             1       0.25604       0.88089        0.9181
    {'pYisMINY'           }   1.9394e-06      0.036566       0.74396       0.11911        0.08185
    {'pYisMAXY'           }   1.1947e-09      0.024953       0.25604    5.4117e-06    5.9148e-1
    {'p0_01'              }       3.6063            19             0             0
    {'p10'                }       20.129            22             0             0        0.2008
    {'p25'                }       31.931            29             0        3.7372        6.563
    {'p50'                }        53.79            44             0        65.686        70.53
    {'p75'                }       90.494            55             1        239.18        258.9
    {'p90'                }       143.31            61             1        525.49        583.8
    {'p99_99'             }       816.36            64             1        4974.3        5033.
    {'fl_cov_y_all'       }       3879.8        217.85        3.8458         17148         1823
    {'fl_cor_y_all'       }            1       0.24637       0.14147       0.81024        0.8304
    {'fl_cov_age_ss'      }       217.85        201.53      -0.25515        2124.1        2205.
    {'fl_cor_age_ss'      }      0.24637             1     -0.041181       0.44036        0.4407
    {'fl_cov_educ_ss'     }       3.8458      -0.25515       0.19048        8.5838        9.279
    {'fl_cor_educ_ss'     }      0.14147     -0.041181             1      0.057885        0.06032
    {'fl_cov_a_ss'        }        17148        2124.1        8.5838    1.1544e+05    1.1966e+0
    {'fl_cor_a_ss'        }      0.81024       0.44036      0.057885             1        0.9992
    {'fl_cov_ap_ss'       }        18233        2205.5        9.2793    1.1966e+05    1.2423e+0
    {'fl_cor_ap_ss'       }      0.83049       0.44078      0.060323       0.99922
    {'fl_cov_MPC'         }      -4.5809       -1.3124      0.020725       -17.446        -18.64
    {'fl_cor_MPC'         }     -0.30887      -0.38826       0.19943      -0.21564        -0.2221
    {'fl_cov_Mass'        }  -0.00012497   -5.6686e-05   -3.0201e-07   -0.00063245    -0.0006678
    {'fl_cor_Mass'        }     -0.21994      -0.43773     -0.075858      -0.20405        -0.2077
    {'fl_cov_c_ss'        }       1859.9        85.521        2.2319        8778.5        9253.
    {'fl_cor_c_ss'        }      0.97519       0.19675       0.16702       0.84382        0.857
    {'fl_cov_y_head_inc'  }       3879.8        217.85        3.8458         17148         1823
    {'fl_cor_y_head_inc'  }            1       0.24637       0.14147       0.81024        0.8304
    {'fl_cov_y_spouse'    }            0             0             0             0
    {'fl_cor_y_spouse'    }          NaN           NaN           NaN           NaN            Na
    {'fl_cov_yshr_nttxss'}        1.7036       0.14382     0.0019345        6.9206        7.370
    {'fl_cor_yshr_nttxss'}       0.79998       0.29631       0.12964       0.59576        0.6116
    {'fracByP0_01'        }   4.4303e-06      0.016474             0             0
    {'fracByP10'          }     0.020773      0.059565             0             0    5.8512e-0
    {'fracByP25'          }     0.075454       0.14379             0     0.0011338      0.001658
    {'fracByP50'          }      0.22298        0.3659             0      0.043403       0.03929
    {'fracByP75'          }      0.46643       0.67037             1       0.22326        0.2188
    {'fracByP90'          }      0.70129       0.88673             1       0.49372        0.5010
    {'fracByP99_99'       }      0.99869             1             1       0.99759        0.9971

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0 and kids =1
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
xxx tb_outcomes: all stats xxx
    OriginalVariableNames        y_all         age_ss       educ_ss         a_ss          ap_ss
    ______________________    ___________    ___________    __________    ___________    __________

    {'mean'                }         65.867         36.118       0.25753         94.382        101.22
    {'unweighted_sum'      }     1.7831e+08           1909             1     1.2935e+05    1.5913e+09
    {'sd'                  }         56.932         11.182       0.43728         214.92        223.55
    {'coefofvar'           }        0.86435         0.3096        1.6979         2.2771        2.2087
    {'gini'                }        0.40364        0.17438       0.68158        0.79318       0.79099
    {'min'                 }         2.2124             19             0              0             0
    {'max'                 }         1414.1             64             1         7837.6        8291.1
    {'pYis0'               }              0              0       0.74247        0.23563       0.21309
    {'pYls0'               }              0              0             0              0             0
    {'pYgr0'               }              1              1       0.25753        0.76437       0.78691
    {'pYisMINY'            }     1.6845e-06       0.020845       0.74247        0.23563       0.21309
    {'pYisMAXY'            }     3.4305e-10      0.0031122       0.25753      8.262e-07    1.6379e-10
    {'p0_01'               }         3.5188             19             0              0             0
    {'p10'                 }         19.292             22             0              0             0
    {'p25'                 }         30.023             26             0       0.029898       0.23918
    {'p50'                 }         49.454             35             0         10.255        14.159
    {'p75'                 }         82.311             45             1          82.04        100.97
    {'p90'                 }         130.49             52             1         276.88        293.79
    {'p99_99'              }         764.17             64             1         3737.2        3751.1
    {'fl_cov_y_all'        }         3241.3         152.53        4.1103         9427.3         10125
    {'fl_cor_y_all'        }              1        0.23959       0.16511        0.77047       0.79555
    {'fl_cov_age_ss'       }         152.53         125.05       0.19904         967.92          1008
    {'fl_cor_age_ss'       }        0.23959              1      0.040704        0.40274       0.40323
    {'fl_cov_educ_ss'      }         4.1103        0.19904       0.19121         5.7853        6.3576
    {'fl_cor_educ_ss'      }        0.16511       0.040704             1       0.061559      0.065037
    {'fl_cov_a_ss'         }         9427.3         967.92        5.7853          46190         47995
    {'fl_cor_a_ss'         }        0.77047        0.40274      0.061559              1       0.99895
    {'fl_cov_ap_ss'        }          10125           1008        6.3576          47995         49975
    {'fl_cor_ap_ss'        }        0.79555        0.40323      0.065037        0.99895             1
    {'fl_cov_MPC'          }        -8.9788        -1.3659      0.021532        -20.591        -22.11
    {'fl_cor_MPC'          }       -0.45848       -0.35509       0.14315       -0.27853      -0.28753
    {'fl_cov_Mass'         }    -4.8054e-05    -1.3242e-05    -1.409e-07    -0.00013616    -0.0001451
    {'fl_cor_Mass'         }       -0.33303       -0.46722      -0.12714       -0.24996      -0.25609
    {'fl_cov_c_ss'         }         1766.6         76.831        2.5674         5341.5        5695.3
    {'fl_cor_c_ss'         }        0.98556        0.21823       0.18649         0.7894       0.80918
    {'fl_cov_y_head_inc'   }         3241.3         152.53        4.1103         9427.3         10125
    {'fl_cor_y_head_inc'   }              1        0.23959       0.16511        0.77047       0.79555
    {'fl_cov_y_spouse'     }              0              0             0              0             0
    {'fl_cor_y_spouse'     }            NaN            NaN           NaN            NaN           NaN
    {'fl_cov_yshr_nttxss'}           1.5555        0.10317     0.0024166         3.8467        4.1371
    {'fl_cor_yshr_nttxss'}          0.80522         0.2719       0.16288         0.5275       0.54542
    {'fracByP0_01'         }     4.7258e-06       0.010966             0              0             0
    {'fracByP10'           }       0.021712       0.068419             0              0             0
    {'fracByP25'           }       0.078097        0.15825             0     6.9352e-06    3.5474e-05
    {'fracByP50'           }        0.22713        0.37923             0      0.0099825      0.010737
    {'fracByP75'           }        0.46939        0.67298             1        0.11689       0.12619
    {'fracByP90'           }        0.70221        0.85638             1        0.39992       0.38891
    {'fracByP99_99'        }        0.99866              1             1        0.99622       0.99554

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0 and kids =2
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames        y_all         age_ss        educ_ss         a_ss          ap_ss
```

```
    ______________________    ___________    ___________    ___________    ___________    __________

    {'mean'              }         64.473         34.566        0.24576         62.952         67.62
    {'unweighted_sum'    }     1.7831e+08           1909              1     1.2935e+05     1.5826e+0
    {'sd'                }         54.982         9.1574        0.43053         161.65         168.9
    {'coefofvar'         }        0.85279        0.26492         1.7519         2.5678         2.498
    {'gini'              }        0.39845        0.14726        0.69833        0.83755        0.8405
    {'min'               }         2.2124             19              0              0
    {'max'               }         1414.1             64              1         7837.6           822
    {'pYis0'             }              0              0        0.75424        0.36638        0.3665
    {'pYls0'             }              0              0              0              0
    {'pYgr0'             }              1              1        0.24576        0.63362        0.6334
    {'pYisMINY'          }     9.8494e-07        0.01156        0.75424        0.36638        0.3665
    {'pYisMAXY'          }     6.7773e-11     0.00057855        0.24576        1.4e-07      9.363e-1
    {'p0_01'             }         3.5915             19              0              0
    {'p10'               }         19.324             23              0              0
    {'p25'               }         29.888             27              0              0
    {'p50'               }         48.811             34              0         1.9135         3.284
    {'p75'               }         80.448             41              0         51.664         56.74
    {'p90'               }         126.75             47              1         174.36         199.5
    {'p99_99'            }         740.25             64              1         2910.1         3038.
    {'fl_cov_y_all'      }           3023         108.56         4.4029         6789.1         7342.
    {'fl_cor_y_all'      }              1        0.21562          0.186        0.76387         0.790
    {'fl_cov_age_ss'     }         108.56         83.858        0.38657         516.18         539.9
    {'fl_cor_age_ss'     }        0.21562              1        0.09805         0.3487        0.3489
    {'fl_cov_educ_ss'    }         4.4029        0.38657        0.18536         4.3837         4.932
    {'fl_cor_educ_ss'    }          0.186        0.09805              1       0.062988       0.06779
    {'fl_cov_a_ss'       }         6789.1         516.18         4.3837          26131          2728
    {'fl_cor_a_ss'       }        0.76387         0.3487       0.062988              1        0.9987
    {'fl_cov_ap_ss'      }         7342.2         539.94         4.9328          27284          2855
    {'fl_cor_ap_ss'      }         0.7902        0.34891       0.067799        0.99878
    {'fl_cov_MPC'        }        -12.648          -1.27      0.0077811        -21.879        -23.53
    {'fl_cor_MPC'        }       -0.55643       -0.33546       0.043715       -0.32737       -0.3368
    {'fl_cov_Mass'       }    -7.6259e-05    -1.1065e-05    -2.8216e-07    -0.00014947    -0.0001602
    {'fl_cor_Mass'       }       -0.36259       -0.31587       -0.17133       -0.24172       -0.2479
    {'fl_cov_c_ss'       }         1746.8         59.484          2.819         3991.5         4289.
    {'fl_cor_c_ss'       }        0.98869        0.20214        0.20376        0.76839        0.7899
    {'fl_cov_y_head_inc' }           3023         108.56         4.4029         6789.1         7342.
    {'fl_cor_y_head_inc' }              1        0.21562          0.186        0.76387         0.790
    {'fl_cov_y_spouse'   }              0              0              0              0
    {'fl_cor_y_spouse'   }            NaN            NaN            NaN            NaN            Na
    {'fl_cov_yshr_nttxss'}         1.4898       0.073937      0.0027409           2.72         2.940
    {'fl_cor_yshr_nttxss'}         0.8086        0.24094        0.18997        0.50211        0.5192
    {'fracByP0_01'       }      4.972e-06      0.0063542              0              0
    {'fracByP10'         }        0.02237       0.067607              0              0
    {'fracByP25'         }       0.079911         0.1793              0              0
    {'fracByP50'         }        0.23057        0.42619              0      0.0018299      0.002042
    {'fracByP75'         }        0.47337        0.67985              0       0.091755       0.08396
    {'fracByP90'         }        0.70508        0.85252              1        0.32859        0.3322
    {'fracByP99_99'      }        0.99867              1              1        0.99463        0.9945

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0 and kids =3
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames        y_all         age_ss         educ_ss          a_ss          ap_ss
    ______________________    ___________    ___________    ___________    ___________    __________
```

{'mean' }	63.898	34.068	0.22983	48.134	51.95
{'unweighted_sum' }	1.7831e+08	1909	1	1.2935e+05	1.5774e+0
{'sd' }	54.001	7.9772	0.42073	134.32	141.0
{'coefofvar' }	0.84511	0.23415	1.8306	2.7906	2.715
{'gini' }	0.39521	0.12909	0.72073	0.86678	0.8694
{'min' }	2.2124	19	0	0	
{'max' }	1414.1	64	1	7837.6	8183.
{'pYis0' }	0	0	0.77017	0.45656	0.4513
{'pYls0' }	0	0	0	0	
{'pYgr0' }	1	1	0.22983	0.54344	0.5486
{'pYisMINY' }	6.8879e-07	0.0083137	0.77017	0.45656	0.4513
{'pYisMAXY' }	1.1096e-11	0.00013776	0.22983	2.4752e-08	1.1431e-1
{'p0_01' }	3.6125	19	0	0	
{'p10' }	19.479	24	0	0	
{'p25' }	29.928	28	0	0	
{'p50' }	48.607	33	0	0.23918	0.498
{'p75' }	79.715	39	0	29.898	37.19
{'p90' }	125.03	45	1	146.89	152.5
{'p99_99' }	727.36	64	1	2546.8	263
{'fl_cov_y_all' }	2916.1	86.208	4.4997	5525	6009.
{'fl_cor_y_all' }	1	0.20012	0.19805	0.76171	0.7889
{'fl_cov_age_ss' }	86.208	63.635	0.47515	327.14	343.3
{'fl_cor_age_ss' }	0.20012	1	0.14157	0.30531	0.3051
{'fl_cov_educ_ss' }	4.4997	0.47515	0.17701	3.7005	4.224
{'fl_cor_educ_ss' }	0.19805	0.14157	1	0.065482	0.07117
{'fl_cov_a_ss' }	5525	327.14	3.7005	18042	1892
{'fl_cor_a_ss' }	0.76171	0.30531	0.065482	1	0.9986
{'fl_cov_ap_ss' }	6009.3	343.36	4.2241	18921	1989
{'fl_cor_ap_ss' }	0.78891	0.30515	0.071177	0.99863	
{'fl_cov_MPC' }	-14.294	-1.1063	-0.0020988	-20.435	-22.07
{'fl_cor_MPC' }	-0.611	-0.32013	-0.011516	-0.35119	-0.3612
{'fl_cov_Mass' }	-4.6648e-05	-5.1607e-06	-2.1548e-07	-7.4065e-05	-7.9832e-0
{'fl_cor_Mass' }	-0.3662	-0.27425	-0.21712	-0.23375	-0.2399
{'fl_cov_c_ss' }	1734.9	49.892	2.92	3306.4	3575.
{'fl_cor_c_ss' }	0.99009	0.19275	0.21389	0.75862	0.7812
{'fl_cov_y_head_inc' }	2916.1	86.208	4.4997	5525	6009.
{'fl_cor_y_head_inc' }	1	0.20012	0.19805	0.76171	0.7889
{'fl_cov_y_spouse' }	0	0	0	0	
{'fl_cor_y_spouse' }	NaN	NaN	NaN	NaN	Na
{'fl_cov_yshr_nttxss'}	1.4552	0.059225	0.0028436	2.1663	2.353
{'fl_cor_yshr_nttxss'}	0.81051	0.2233	0.20329	0.4851	0.5018
{'fracByP0_01' }	5.1241e-06	0.0046366	0	0	
{'fracByP10' }	0.022798	0.072031	0	0	
{'fracByP25' }	0.081073	0.1985	0	0	
{'fracByP50' }	0.23275	0.41302	0	0.00019689	0.0001865
{'fracByP75' }	0.47593	0.67505	0	0.056161	0.05684
{'fracByP90' }	0.70691	0.86752	1	0.31708	0.2910
{'fracByP99_99' }	0.99868	1	1	0.99425	0.9938

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0 and kids =4
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx

OriginalVariableNames	y_all	age_ss	educ_ss	a_ss	ap_ss
{'mean' }	63.864	34.197	0.2079	41.099	44.67
{'unweighted_sum' }	1.7831e+08	1909	1	1.2935e+05	1.5749e+0

```
    {'sd'                 }        53.609         7.1538         0.4058         120.68         127.2
    {'coefofvar'          }       0.83942         0.2092         1.9519         2.9363         2.849
    {'gini'               }       0.39289        0.11447        0.75112        0.88249        0.8837
    {'min'                }        2.2124             19              0              0
    {'max'                }        1414.1             64              1         7837.6         8164.
    {'pYis0'              }             0              0         0.7921        0.50485        0.4925
    {'pYls0'              }             0              0              0              0
    {'pYgr0'              }             1              1         0.2079        0.49515        0.5074
    {'pYisMINY'           }    4.3493e-07      0.0045732         0.7921        0.50485        0.4925
    {'pYisMAXY'           }    1.4837e-12     4.6124e-05         0.2079      5.166e-09     1.5175e-1
    {'p0_01'              }        3.6887             19              0              0
    {'p10'                }        19.685             25              0              0
    {'p25'                }        30.153             29              0              0
    {'p50'                }         48.75             34              0              0       0.02989
    {'p75'                }        79.548             39              0         21.796         27.79
    {'p90'                }        124.62             44              1         122.46         130.0
    {'p99_99'             }        721.01             63              1         2377.1         2423.
    {'fl_cov_y_all'       }        2873.9         71.941         4.4963         4930.8           539
    {'fl_cor_y_all'       }             1        0.18759        0.20668        0.76216        0.7900
    {'fl_cov_age_ss'      }        71.941         51.176         0.5437         238.82         251.8
    {'fl_cor_age_ss'      }       0.18759              1        0.18729        0.27663        0.2765
    {'fl_cov_educ_ss'     }        4.4963         0.5437        0.16468         3.6463         4.175
    {'fl_cor_educ_ss'     }       0.20668        0.18729              1       0.074456       0.08084
    {'fl_cov_a_ss'        }        4930.8         238.82         3.6463          14564          1533
    {'fl_cor_a_ss'        }       0.76216        0.27663       0.074456              1        0.9985
    {'fl_cov_ap_ss'       }          5391         251.85         4.1759          15338          1620
    {'fl_cor_ap_ss'       }       0.79005        0.27658       0.080846        0.99852
    {'fl_cov_MPC'         }       -14.976        -1.0328      -0.010238        -18.883        -20.53
    {'fl_cor_MPC'         }      -0.63881       -0.33014      -0.057692       -0.35781       -0.3689
    {'fl_cov_Mass'        }   -2.7333e-05    -2.6204e-06    -1.3997e-07    -3.8707e-05    -4.2009e-0
    {'fl_cor_Mass'        }      -0.36608         -0.263       -0.24765        -0.2303       -0.2369
    {'fl_cov_c_ss'        }          1727         42.146         2.9107         2959.6         3218.
    {'fl_cor_c_ss'        }        0.9908         0.1812        0.22061        0.75427        0.7777
    {'fl_cov_y_head_inc'  }        2873.9         71.941         4.4963         4930.8           539
    {'fl_cor_y_head_inc'  }             1        0.18759        0.20668        0.76216        0.7900
    {'fl_cov_y_spouse'    }             0              0              0              0
    {'fl_cor_y_spouse'    }           NaN            NaN            NaN            NaN            Na
    {'fl_cov_yshr_nttxss'}          1.437       0.049376      0.0028207         1.8938         2.069
    {'fl_cor_yshr_nttxss'}        0.81163        0.20899        0.21047        0.47516        0.4922
    {'fracByP0_01'        }    5.1325e-06      0.0025409              0              0
    {'fracByP10'          }      0.022996       0.072385              0              0
    {'fracByP25'          }      0.081963        0.20856              0              0
    {'fracByP50'          }       0.23436        0.45949              0              0     4.2228e-0
    {'fracByP75'          }       0.47776        0.70775              0       0.041779       0.04305
    {'fracByP90'          }       0.70826        0.87861              1         0.2851         0.266
    {'fracByP99_99'       }       0.99869        0.99991              1        0.99427        0.9933

  0x0 empty char array

  0x0 empty char array

-------------------------------
-------------------------------
-------------------------------
-------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =1
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
-------------------------------
-------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =1 and kids =0
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames        y_all         age_ss        educ_ss         a_ss          ap_ss
    ______________________    ___________    ___________    __________    ___________    ___________

    {'mean'                }         109.57         44.041       0.38021         408.45          434.2
    {'unweighted_sum'      }     1.2919e+09           1909             1     1.2935e+05     8.4753e+09
    {'sd'                  }         83.584         15.135       0.48544         498.74          514.1
    {'coefofvar'           }        0.76282        0.34365        1.2768         1.2211          1.184
    {'gini'                }         0.3692        0.18905       0.50257        0.57853        0.57485
    {'min'                 }         2.4223             19             0              0              0
    {'max'                 }         2113.2             64             1         7837.6         9503.9
    {'pYis0'               }              0              0       0.61979        0.07588       0.036618
    {'pYls0'               }              0              0             0              0              0
    {'pYgr0'               }              1              1       0.38021        0.92412        0.96338
    {'pYisMINY'            }     5.1901e-09       0.043093       0.61979        0.07588       0.036618
    {'pYisMAXY'            }     2.4329e-11       0.038439       0.38021     1.8938e-05     1.2944e-11
    {'p0_01'               }          6.597             19             0              0              0
    {'p10'                 }         35.716             21             0         1.9135         3.7372
    {'p25'                 }         54.783             29             0         51.664         59.412
    {'p50'                 }         88.064             48             0         239.18         270.71
    {'p75'                 }          137.9             58             1         588.48          634.2
    {'p90'                 }         204.71             62             1         1074.4         1074.4
    {'p99_99'              }         967.74             64             1         5833.4         5977.2
    {'fl_cov_y_all'        }         6986.3         361.95        5.7657          27155          31359
    {'fl_cor_y_all'        }              1        0.28612        0.1421        0.65142        0.72978
    {'fl_cov_age_ss'       }         361.95         229.06      -0.85351         4123.7         4247.6
    {'fl_cor_age_ss'       }        0.28612              1      -0.11617         0.5463        0.54592
    {'fl_cov_educ_ss'      }         5.7657       -0.85351       0.23565         16.048         17.247
    {'fl_cor_educ_ss'      }         0.1421       -0.11617             1       0.066283       0.069108
    {'fl_cov_a_ss'         }          27155         4123.7        16.048     2.4874e+05      2.541e+05
    {'fl_cor_a_ss'         }        0.65142         0.5463      0.066283              1        0.99103
    {'fl_cov_ap_ss'        }          31359         4247.6        17.247      2.541e+05      2.643e+05
    {'fl_cor_ap_ss'        }        0.72978        0.54592      0.069108        0.99103              1
    {'fl_cov_MPC'          }        -4.5371        -1.4233     0.0094147        -28.557        -30.413
    {'fl_cor_MPC'          }        -0.3463       -0.59994       0.12373       -0.36528       -0.37741
    {'fl_cov_Mass'         }    -7.4044e-05    -2.2911e-05    5.5246e-08    -0.00041977    -0.00044771
    {'fl_cor_Mass'         }       -0.19462       -0.33257      0.025002       -0.18491       -0.19132
    {'fl_cov_c_ss'         }         2941.3         188.87        4.2928          16348          17228
    {'fl_cor_c_ss'         }        0.86319        0.30611       0.21692        0.80402        0.82201
    {'fl_cov_y_head_inc'   }         4857.1         318.48        4.4672          25709          27033
    {'fl_cor_y_head_inc'   }        0.85851        0.31089       0.13595        0.76156        0.77684
    {'fl_cov_y_spouse'     }         4673.1           95.4        2.8501         3173.5         9494.5
    {'fl_cor_y_spouse'     }        0.59164       0.066703       0.06213       0.067335        0.19543
    {'fl_cov_yshr_nttxss'}           1.6494        0.13956     0.0013995         6.3032         7.2859
    {'fl_cor_yshr_nttxss'}           0.7665        0.35816       0.11198        0.49089        0.55047
    {'fracByP0_01'         }     5.3214e-06       0.018591             0              0              0
    {'fracByP10'           }       0.024266       0.049706             0     0.00015705     0.00034862
    {'fracByP25'           }       0.086535         0.1342             0       0.010018       0.010404
    {'fracByP50'           }        0.24796        0.35975             0       0.097271       0.098709
```

13.1.11 Distributional Statistics By Marital Status, Kids Count and Income Bins

Various statistics, including MPC (of the first check) by Marital Status and Kids COunt and income bins

```
it_row_ctr = 0;
for it_marry_ctr=1:mp_params('n_marriedgrid')

    display(['']);
    display(['']);
    display(['-------------------------------']);
    display(['-------------------------------']);
    display(['-------------------------------']);
    display(['-------------------------------']);
    display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);
    display(['Marital =' num2str(ar_marital(it_marry_ctr))]);
    display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);
    display(['-------------------------------']);
    display(['-------------------------------']);

    for it_kids_ctr=1:mp_params('n_kidsgrid')
        display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);
        display(['Marital =' num2str(ar_marital(it_marry_ctr)) ' and kids =' num2str(ar_kids(it_kids
        display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);

        % construct input data
        y_all_grp = y_all(min_age:max_age, :, :, : ,it_marry_ctr ,it_ctr);
        age_ss_grp = age_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
        educ_ss_grp = educ_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
        a_ss_grp = a_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
        ap_ss_grp = ap_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
        mn_MPC_C_gain_share_check_grp = mn_MPC_C_gain_share_check(min_age:max_age, :, :, :, it_marry
        Phi_true_grp = Phi_true_1(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
        c_ss_grp = c_ss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
        y_head_inc_grp = y_head_inc(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
        y_spouse_inc_grp = y_spouse_inc(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);
        yshr_nttxss_grp = yshr_nttxss(min_age:max_age, :, :, : ,it_marry_ctr, it_kids_ctr);

        % Income Bins
        ar_y_all = y_all_grp(:);
        ar_age_ss = age_ss_grp(:);
        ar_educ_ss = educ_ss_grp(:);
        ar_a_ss = a_ss_grp(:);
        ar_ap_ss = ap_ss_grp(:);
        ar_mn_MPC_C_gain_share_check = mn_MPC_C_gain_share_check_grp(:);
        ar_Phi_true = Phi_true_grp(:);
        ar_c_ss = c_ss_grp(:);
        ar_y_head_inc = y_head_inc_grp(:);
        ar_y_spouse_inc = y_spouse_inc_grp(:);
        ar_yshr_nttxss = yshr_nttxss_grp(:);

        % income bins loop
        for it_y_all_ctr=1:6

            % Current y group index
            % y is in thousands of dollars
            y_all_start = (it_y_all_ctr-1)*20;
            if (it_y_all_ctr == 6)
```

```
    y_all_end = max(ar_y_all);
else
    y_all_end = it_y_all_ctr*20;
end

display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);
display(['Marital =' num2str(ar_marital(it_marry_ctr)) ', kids =' num2str(ar_kids(it_kid
display(['xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx']);

ar_y_idx = (ar_y_all >= y_all_start & ar_y_all <y_all_end);

ar_mky_y_all = ar_y_all(ar_y_idx);
ar_mky_age_ss = ar_age_ss(ar_y_idx);
ar_mky_educ_ss = ar_educ_ss(ar_y_idx);
ar_mky_a_ss = ar_a_ss(ar_y_idx);
ar_mky_ap_ss = ar_ap_ss(ar_y_idx);
ar_mky_mn_MPC_C_gain_share_check = ar_mn_MPC_C_gain_share_check(ar_y_idx);
ar_mky_Phi_true = ar_Phi_true(ar_y_idx);
ar_mky_c_ss = ar_c_ss(ar_y_idx);
ar_mky_y_head_inc = ar_y_head_inc(ar_y_idx);
ar_mky_y_spouse_inc = ar_y_spouse_inc(ar_y_idx);
ar_mky_yshr_nttxss = ar_yshr_nttxss(ar_y_idx);

mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
mp_cl_ar_xyz_of_s('y_all') = {ar_mky_y_all(:), zeros(1)};
mp_cl_ar_xyz_of_s('age_ss') = {ar_mky_age_ss(:), zeros(1)};
mp_cl_ar_xyz_of_s('educ_ss') = {ar_mky_educ_ss(:), zeros(1)};
mp_cl_ar_xyz_of_s('a_ss') = {ar_mky_a_ss(:), zeros(1)};
mp_cl_ar_xyz_of_s('ap_ss') = {ar_mky_ap_ss(:), zeros(1)};
mp_cl_ar_xyz_of_s('MPC') = {ar_mky_mn_MPC_C_gain_share_check(:), zeros(1)};
mp_cl_ar_xyz_of_s('Mass') = {ar_mky_Phi_true(:), zeros(1)};
mp_cl_ar_xyz_of_s('c_ss') = {ar_mky_c_ss(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_head_inc') = {ar_mky_y_head_inc(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_spouse') = {ar_mky_y_spouse_inc(:), zeros(1)};
mp_cl_ar_xyz_of_s('yshr_nttxss') = {ar_mky_yshr_nttxss(:), zeros(1)};
mp_cl_ar_xyz_of_s('ar_st_y_name') = ["y_all", "age_ss", "educ_ss", "a_ss", "ap_ss", "MPC

% controls
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('ar_fl_percentiles') = [0.01 10 25 50 75 90 99.99];
mp_support('bl_display_final') = true;
mp_support('bl_display_detail') = false;
mp_support('bl_display_drvm2outcomes') = false;
mp_support('bl_display_drvstats') = false;
mp_support('bl_display_drvm2covcor') = false;

% Call Function
mp_cl_mt_xyz_of_s = ff_simu_stats(ar_mky_Phi_true(:)/sum(ar_mky_Phi_true,'all'), mp_cl_a

it_marital = ar_marital(it_marry_ctr);
it_kids = ar_kids(it_kids_ctr);
fl_y_all_start = y_all_start;
fl_y_all_end = y_all_end;

tb_dist_stats = mp_cl_mt_xyz_of_s('tb_outcomes');
fl_age_mean = tb_dist_stats{"age_ss", "mean"};
fl_age_p50 = tb_dist_stats{"age_ss", "p50"};
```

```
            fl_educ_mean = tb_dist_stats{"educ_ss", "mean"};

            fl_a_mean = tb_dist_stats{"a_ss", "mean"};
            fl_a_p50 = tb_dist_stats{"a_ss", "p50"};

            fl_ap_mean = tb_dist_stats{"ap_ss", "mean"};
            fl_ap_p50 = tb_dist_stats{"ap_ss", "p50"};

            fl_y_all_mean = tb_dist_stats{"y_all", "mean"};
            fl_y_all_p50 = tb_dist_stats{"y_all", "p50"};

            fl_mpc_mean = tb_dist_stats{"MPC", "mean"};
            fl_mpc_p50 = tb_dist_stats{"MPC", "p50"};

            fl_mass = tb_dist_stats{"Mass", "unweighted_sum"};

            fl_c_ss_mean = tb_dist_stats{"c_ss", "mean"};
            fl_c_ss_p50 = tb_dist_stats{"c_ss", "p50"};

            fl_y_head_inc_mean = tb_dist_stats{"y_head_inc", "mean"};
            fl_y_spouse_mean = tb_dist_stats{"y_spouse", "mean"};

            ar_store_stats = [it_marital, it_kids, fl_y_all_start, fl_y_all_end, ...
                fl_age_mean, fl_age_p50, fl_educ_mean, ...
                fl_a_mean, fl_a_p50, fl_ap_mean, fl_ap_p50, ...
                fl_y_all_mean, fl_y_all_p50, ...
                fl_mpc_mean, fl_mpc_p50, ...
                fl_mass, ...
                fl_c_ss_mean, fl_c_ss_p50, ...
                fl_y_head_inc_mean, fl_y_spouse_mean];

            it_row_ctr = it_row_ctr + 1;

            if (it_row_ctr>1)
                mt_store_stats_by_mky = [mt_store_stats_by_mky;ar_store_stats];
            else
                mt_store_stats_by_mky = [ar_store_stats];
            end

        end
    end
end
```

```
  0x0 empty char array

  0x0 empty char array

-------------------------------
-------------------------------
-------------------------------
-------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-------------------------------
-------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0 and kids =0
```

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0, kids =0, ybin =0 to 20
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       y_all        age_ss        educ_ss         a_ss          ap_ss
    ______________________    __________    ___________    ___________    ___________    ___________

    {'mean'              }        14.688         34.974        0.18182         2.7226         2.6703
    {'unweighted_sum'    }    8.4083e+05           1909              1         2690.8     5.2986e+06
    {'sd'                }        3.6764         14.501         0.3857         10.125         9.7397
    {'coefofvar'         }       0.25031        0.41462         2.1213         3.7189         3.6474
    {'gini'              }         0.141        0.23128        0.78641        0.92249        0.92536
    {'min'               }        2.2124             19              0              0              0
    {'max'               }            20             64              1         413.31         409.12
    {'pYis0'             }             0              0        0.81818        0.53967        0.49704
    {'pYls0'             }             0              0              0              0              0
    {'pYgr0'             }             1              1        0.18182        0.46033        0.50296
    {'pYisMINY'          }    1.9988e-05       0.084859        0.81818        0.53967        0.49704
    {'pYisMAXY'          }    4.7568e-12        0.01496        0.18182     1.4916e-11              0
    {'p0_01'             }        2.6052             19              0              0              0
    {'p10'               }         9.307             20              0              0              0
    {'p25'               }        12.172             22              0              0              0
    {'p50'               }        15.236             30              0              0       0.011132
    {'p75'               }        17.778             48              0        0.23918        0.48535
    {'p90'               }         19.14             58              1          6.458         6.0051
    {'p99_99'            }        19.999             64              1         174.36         166.76
    {'fl_cov_y_all'      }        13.516         5.6455      0.0023525         6.9774         7.1104
    {'fl_cor_y_all'      }             1         0.1059       0.001659        0.18744        0.19857
    {'fl_cov_age_ss'     }        5.6455         210.28        -1.0046         57.763         56.482
    {'fl_cor_age_ss'     }        0.1059              1       -0.17962        0.39342        0.39991
    {'fl_cov_educ_ss'    }     0.0023525        -1.0046        0.14876       -0.29328       -0.29618
    {'fl_cor_educ_ss'    }      0.001659       -0.17962              1        -0.0751      -0.078843
    {'fl_cov_a_ss'       }        6.9774         57.763       -0.29328         102.52         98.523
    {'fl_cor_a_ss'       }       0.18744        0.39342        -0.0751              1        0.99907
    {'fl_cov_ap_ss'      }        7.1104         56.482       -0.29618         98.523         94.862
    {'fl_cor_ap_ss'      }       0.19857        0.39991      -0.078843        0.99907              1
    {'fl_cov_MPC'        }      -0.53539        -2.7127       0.035745        -1.3183        -1.3069
    {'fl_cor_MPC'        }      -0.41675       -0.53535        0.26522       -0.37261       -0.38399
    {'fl_cov_Mass'       }    6.2968e-06    -7.1893e-05    -3.4162e-07    -1.6308e-05    -1.5421e-05
    {'fl_cor_Mass'       }       0.18379       -0.53201      -0.095046       -0.17284        -0.1699
    {'fl_cov_c_ss'       }        11.292         6.0576      0.0049126         9.8651         9.6441
    {'fl_cor_c_ss'       }       0.98267        0.13365      0.0040749        0.31172        0.31679
    {'fl_cov_y_head_inc' }        13.516         5.6455      0.0023525         6.9774         7.1104
    {'fl_cor_y_head_inc' }             1         0.1059       0.001659        0.18744        0.19857
    {'fl_cov_y_spouse'   }             0              0              0              0              0
    {'fl_cor_y_spouse'   }           NaN            NaN            NaN            NaN            NaN
    {'fl_cov_yshr_nttxss'}      0.052502       0.022189     1.0001e-05       0.025753       0.026154
    {'fl_cor_yshr_nttxss'}       0.99014        0.10609      0.0017978        0.17635        0.18618
    {'fracByP0_01'       }    1.8388e-05         0.0461              0              0              0
    {'fracByP10'         }      0.051228       0.088792              0              0              0
    {'fracByP25'         }       0.16212        0.16288              0              0              0
    {'fracByP50'         }       0.39701         0.3352              0              0     6.3754e-06
    {'fracByP75'         }       0.68371        0.60992              0       0.013023       0.018274
    {'fracByP90'         }       0.86704        0.84015              1        0.16055        0.12661
    {'fracByP99_99'      }             1              1              1        0.99661        0.99334

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
Marital =0, kids =0, ybin =20 to 40
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames          y_all          age_ss       educ_ss          a_ss           ap_ss
    ______________________     ___________     __________     __________     ___________     ___________

    {'mean'               }         29.952         38.525        0.20638          25.273          26.927
    {'unweighted_sum'     }     1.9087e+06           1909              1          7355.4      1.8539e+07
    {'sd'                 }         5.6854         14.456         0.4047          44.204          44.499
    {'coefofvar'          }        0.18982        0.37522          1.961          1.7491          1.6525
    {'gini'               }        0.10955         0.2128         0.7532         0.72521         0.71433
    {'min'                }             20             19              0               0               0
    {'max'                }             40             64              1          890.69          885.93
    {'pYis0'              }              0              0        0.79362         0.16854         0.10132
    {'pYls0'              }              0              0              0               0               0
    {'pYgr0'              }              1              1        0.20638         0.83146         0.89868
    {'pYisMINY'           }    8.0995e-288       0.055295        0.79362         0.16854         0.10132
    {'pYisMAXY'           }    4.4783e-07        0.018337        0.20638      8.9396e-13               0
    {'p0_01'              }         20.004             19              0               0               0
    {'p10'                }         22.011             20              0               0               0
    {'p25'                }         25.065             25              0         0.80724          1.0692
    {'p50'                }         30.008             37              0           6.458          6.7423
    {'p75'                }         34.798             52              0          29.898           33.25
    {'p90'                }         37.826             59              1           82.04          83.859
    {'p99_99'             }         39.999             64              1          413.31          407.97
    {'fl_cov_y_all'       }         32.324         7.6438       0.038871          72.305          78.269
    {'fl_cor_y_all'       }              1       0.093006       0.016894         0.28771         0.30937
    {'fl_cov_age_ss'      }         7.6438         208.96       -0.73238          386.27          403.49
    {'fl_cor_age_ss'      }       0.093006              1       -0.12519         0.60451         0.62727
    {'fl_cov_educ_ss'     }       0.038871       -0.73238        0.16379         -2.1597         -2.3263
    {'fl_cor_educ_ss'     }       0.016894       -0.12519              1        -0.12073        -0.12918
    {'fl_cov_a_ss'        }         72.305         386.27        -2.1597            1954          1964.7
    {'fl_cor_a_ss'        }        0.28771        0.60451       -0.12073               1         0.99885
    {'fl_cov_ap_ss'       }         78.269         403.49        -2.3263          1964.7          1980.1
    {'fl_cor_ap_ss'       }        0.30937        0.62727       -0.12918         0.99885               1
    {'fl_cov_MPC'         }       -0.30518        -1.6691       0.062231         -3.0143         -3.2123
    {'fl_cor_MPC'         }         -0.208        -0.4474        0.59583        -0.26423        -0.27972
    {'fl_cov_Mass'        }    -4.5536e-06     -9.611e-05     3.7003e-08     -0.00013723     -0.00014127
    {'fl_cor_Mass'        }      -0.059962       -0.49775      0.0068451        -0.23242        -0.23768
    {'fl_cov_c_ss'        }         19.905          -11.1        0.19764          47.044          47.213
    {'fl_cor_c_ss'        }        0.88176       -0.19338        0.12299         0.26804         0.26721
    {'fl_cov_y_head_inc'  }         32.324         7.6438       0.038871          72.305          78.269
    {'fl_cor_y_head_inc'  }              1       0.093006       0.016894         0.28771         0.30937
    {'fl_cov_y_spouse'    }              0              0              0               0               0
    {'fl_cor_y_spouse'    }            NaN            NaN            NaN             NaN             NaN
    {'fl_cov_yshr_nttxss'}        0.058292       0.013838     6.9651e-05         0.12878         0.13941
    {'fl_cor_yshr_nttxss'}         0.99551       0.092947       0.016711         0.28287         0.30419
    {'fracByP0_01'        }     0.00010275        0.02727              0               0               0
    {'fracByP10'          }       0.070196       0.052525              0               0               0
    {'fracByP25'          }        0.18834        0.15502              0       0.0035395       0.0030822
    {'fracByP50'          }         0.4181        0.33724              0        0.038073        0.032523
    {'fracByP75'          }        0.68834        0.64594              0         0.19439         0.19244
    {'fracByP90'          }        0.87021        0.84782              1         0.53295         0.49305
    {'fracByP99_99'       }        0.99996              1              1         0.99833         0.99827

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0, kids =0, ybin =40 to 60
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

xxx tb_outcomes: all stats xxx

OriginalVariableNames	y_all	age_ss	educ_ss	a_ss	ap_ss
{'mean' }	49.366	41.657	0.23457	81.386	87.44
{'unweighted_sum' }	2.5368e+06	1909	1	13261	2.8595e+0
{'sd' }	5.7595	14.091	0.42373	93.65	93.9
{'coefofvar' }	0.11667	0.33826	1.8064	1.1507	1.074
{'gini' }	0.067274	0.1907	0.71409	0.55942	0.5484
{'min' }	40	19	0	0	
{'max' }	60	64	1	1394.9	1389.
{'pYis0' }	0	0	0.76543	0.074319	0.03470
{'pYls0' }	0	0	0	0	
{'pYgr0' }	1	1	0.23457	0.92568	0.9652
{'pYisMINY' }	1.1988e-05	0.035852	0.76543	0.074319	0.03470
{'pYisMAXY' }	2.6918e-19	0.022889	0.23457	1.725e-14	
{'p0_01' }	40.004	19	0	0	
{'p10' }	41.738	22	0	1.9135	4.178
{'p25' }	44.289	28	0	10.255	15.82
{'p50' }	49.163	43	0	51.664	56.52
{'p75' }	54.155	54	0	122.46	130.2
{'p90' }	57.677	60	1	205.07	213.8
{'p99_99' }	59.997	64	1	729.18	718.6
{'fl_cov_y_all' }	33.172	5.7383	0.031749	121.07	129.8
{'fl_cor_y_all' }	1	0.070707	0.013009	0.22446	0.2398
{'fl_cov_age_ss' }	5.7383	198.55	-0.52991	911.38	944.6
{'fl_cor_age_ss' }	0.070707	1	-0.088752	0.69065	0.7133
{'fl_cov_educ_ss' }	0.031749	-0.52991	0.17955	-5.8166	-6.25
{'fl_cor_educ_ss' }	0.013009	-0.088752	1	-0.14658	-0.1569
{'fl_cov_a_ss' }	121.07	911.38	-5.8166	8770.3	8794.
{'fl_cor_a_ss' }	0.22446	0.69065	-0.14658	1	0.9991
{'fl_cov_ap_ss' }	129.84	944.69	-6.252	8794.3	883
{'fl_cor_ap_ss' }	0.23985	0.71331	-0.15698	0.99911	
{'fl_cov_MPC' }	-0.09046	-0.71366	0.029802	-2.9986	-3.239
{'fl_cor_MPC' }	-0.096943	-0.31261	0.43412	-0.19763	-0.2127
{'fl_cov_Mass' }	-4.8663e-06	-5.8353e-05	-1.5517e-07	-0.00023117	-0.0002379
{'fl_cor_Mass' }	-0.088148	-0.43205	-0.038205	-0.25753	-0.264
{'fl_cov_c_ss' }	17.041	-28.838	0.46008	70.248	61.26
{'fl_cor_c_ss' }	0.62733	-0.43394	0.23022	0.15905	0.1382
{'fl_cov_y_head_inc' }	33.172	5.7383	0.031749	121.07	129.8
{'fl_cor_y_head_inc' }	1	0.070707	0.013009	0.22446	0.2398
{'fl_cov_y_spouse' }	0	0	0	0	
{'fl_cor_y_spouse' }	NaN	NaN	NaN	NaN	Na
{'fl_cov_yshr_nttxss'}	0.032789	0.0057148	3.1401e-05	0.11935	0.1280
{'fl_cor_yshr_nttxss'}	0.99787	0.071088	0.012989	0.22339	0.2387
{'fracByP0_01' }	8.1986e-05	0.016353	0	0	
{'fracByP10' }	0.082731	0.059676	0	0.00080736	0.001552
{'fracByP25' }	0.21327	0.13822	0	0.013027	0.01831
{'fracByP50' }	0.44964	0.36454	0	0.12623	0.1133
{'fracByP75' }	0.7111	0.65402	0	0.38755	0.3665
{'fracByP90' }	0.88093	0.85769	1	0.66284	0.654
{'fracByP99_99' }	0.99988	1	1	0.99951	0.9991

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0, kids =0, ybin =60 to 80
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx

OriginalVariableNames	y_all	age_ss	educ_ss	a_ss	ap_ss

```
    ______________________    ___________    ___________    ___________    ___________    __________

    {'mean'               }         69.288         43.863        0.26092          158.1         169.5
    {'unweighted_sum'     }     3.0245e+06           1909              1          20103     3.6714e+0
    {'sd'                 }         5.7381          13.54        0.43913         144.91         144.6
    {'coefofvar'          }       0.082816        0.30869          1.683        0.91655        0.8531
    {'gini'               }       0.047737        0.17193        0.67675        0.46696        0.4577
    {'min'                }             60             19              0              0
    {'max'                }         79.999             64              1         1913.5         1907.
    {'pYis0'              }              0              0        0.73908       0.036459      0.006283
    {'pYls0'              }              0              0              0              0
    {'pYgr0'              }              1              1        0.26092        0.96354        0.9937
    {'pYisMINY'           }     7.0785e-08       0.024362        0.73908       0.036459      0.006283
    {'pYisMAXY'           }     1.0527e-08       0.027901        0.26092     8.7298e-17
    {'p0_01'              }         60.004             19              0              0
    {'p10'                }         61.586             23              0         10.255         18.69
    {'p25'                }         64.221             32              0         39.794         54.13
    {'p50'                }          68.93             46              0         122.46         134.9
    {'p75'                }         74.224             56              1         239.18         250.8
    {'p90'                }         77.547             61              1         363.77         373.
    {'p99_99'             }         79.989             64              1         1074.4         1050.
    {'fl_cov_y_all'       }         32.926         4.1151       0.027108         145.03         154.4
    {'fl_cor_y_all'       }              1       0.052966       0.010758        0.17442        0.1861
    {'fl_cov_age_ss'      }         4.1151         183.33       -0.37329         1400.5           143
    {'fl_cor_age_ss'      }       0.052966              1      -0.062782        0.71382        0.7342
    {'fl_cov_educ_ss'     }       0.027108       -0.37329        0.19284        -9.2359        -9.793
    {'fl_cor_educ_ss'     }       0.010758      -0.062782              1       -0.14514       -0.1541
    {'fl_cov_a_ss'        }         145.03         1400.5        -9.2359          20999          2094
    {'fl_cor_a_ss'        }        0.17442        0.71382       -0.14514              1         0.999
    {'fl_cov_ap_ss'       }         154.46           1438        -9.7931          20944          2091
    {'fl_cor_ap_ss'       }        0.18612        0.73429       -0.15419         0.9993
    {'fl_cov_MPC'         }      -0.032152        -0.1128      0.0061122       -0.71422       -0.7941
    {'fl_cor_MPC'         }       -0.10557       -0.15696        0.26224      -0.092863       -0.1034
    {'fl_cov_Mass'        }    -2.7754e-06    -2.3008e-05    -3.5426e-07    -0.00016592    -0.0001677
    {'fl_cor_Mass'        }      -0.091883       -0.32281       -0.15325       -0.21751       -0.2203
    {'fl_cov_c_ss'        }         15.819        -34.276        0.57801         165.89          143.
    {'fl_cor_c_ss'        }        0.46921       -0.43083        0.22402        0.19483        0.1688
    {'fl_cov_y_head_inc'  }         32.926         4.1151       0.027108         145.03         154.4
    {'fl_cor_y_head_inc'  }              1       0.052966       0.010758        0.17442        0.1861
    {'fl_cov_y_spouse'    }              0              0              0              0
    {'fl_cor_y_spouse'    }            NaN            NaN            NaN            NaN            Na
    {'fl_cov_yshr_nttxss'}        0.020855      0.0026153      1.697e-05       0.091765        0.0977
    {'fl_cor_yshr_nttxss'}         0.99874       0.053079       0.010619        0.17402         0.185
    {'fracByP0_01'        }     0.00013554       0.010553              0              0
    {'fracByP10'          }       0.087755       0.048541              0      0.0025063      0.004513
    {'fracByP25'          }         0.2241        0.14456              0       0.028727       0.03673
    {'fracByP50'          }        0.46426        0.38247              0        0.18419        0.1735
    {'fracByP75'          }        0.72225        0.68941              1        0.48388        0.4489
    {'fracByP90'          }        0.88703        0.88024              1        0.74634        0.7160
    {'fracByP99_99'       }        0.99995              1              1        0.99979        0.9993

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0, kids =0, ybin =80 to 100
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames        y_all         age_ss         educ_ss          a_ss          ap_ss
    ______________________    ___________    ___________    ___________    ___________    __________
```

```
    {'mean'                 }        89.313          45.289         0.28587          244.08          261.1
    {'unweighted_sum'       }    3.4629e+06            1909               1           26756      4.4356e+0
    {'sd'                   }        5.7791          13.023         0.45183           194.1          192.8
    {'coefofvar'            }      0.064706         0.28755          1.5805         0.79522         0.7385
    {'gini'                 }      0.037295         0.15841          0.6408          0.4125         0.4039
    {'min'                  }        80.001              19               0               0
    {'max'                  }           100              64               1          2377.1         2370.
    {'pYis0'                }             0               0         0.71413        0.020853      1.1851e-1
    {'pYls0'                }             0               0               0               0
    {'pYgr0'                }             1               1         0.28587         0.97915
    {'pYisMINY'             }             0        0.018972         0.71413        0.020853      1.1851e-1
    {'pYisMAXY'             }    3.7911e-06         0.02925         0.28587      1.1813e-15
    {'p0_01'                }        80.012              19               0               0         0.7904
    {'p10'                  }         81.54              25               0          29.898          43.52
    {'p25'                  }         84.27              35               0          100.91          114.2
    {'p50'                  }        88.922              48               0          205.07          224.4
    {'p75'                  }        94.198              56               1          363.77          378.8
    {'p90'                  }        97.585              61               1          525.49          536.5
    {'p99_99'               }           100              64               1          1281.9          1273.
    {'fl_cov_y_all'         }        33.398          2.1956        0.039297          150.26          159.6
    {'fl_cor_y_all'         }             1        0.029174         0.01505         0.13396         0.1432
    {'fl_cov_age_ss'        }        2.1956          169.59        -0.29823          1813.9          1849.
    {'fl_cor_age_ss'        }      0.029174               1       -0.050684         0.71759         0.7365
    {'fl_cov_educ_ss'       }      0.039297        -0.29823         0.20415         -12.356         -12.92
    {'fl_cor_educ_ss'       }       0.01505       -0.050684               1        -0.14089         -0.148
    {'fl_cov_a_ss'          }        150.26          1813.9         -12.356           37675           3741
    {'fl_cor_a_ss'          }       0.13396         0.71759        -0.14089               1         0.9994
    {'fl_cov_ap_ss'         }        159.68          1849.7         -12.922           37410           3719
    {'fl_cor_ap_ss'         }       0.14327         0.73653         -0.1483         0.99942
    {'fl_cov_MPC'           }    -0.0031027        0.051957       0.0006815         0.63872         0.6402
    {'fl_cor_MPC'           }      -0.05432         0.40366         0.15261         0.33294         0.3358
    {'fl_cov_Mass'          }   -1.4164e-06     -8.0012e-06     -3.1745e-07     -9.5448e-05     -9.4333e-0
    {'fl_cor_Mass'          }     -0.083209        -0.20859        -0.23854        -0.16695        -0.1660
    {'fl_cov_c_ss'          }        15.988         -34.182         0.59583          378.89          341.3
    {'fl_cor_c_ss'          }       0.39229         -0.3722           0.187         0.27681         0.2509
    {'fl_cov_y_head_inc'    }        33.398          2.1956        0.039297          150.26          159.6
    {'fl_cor_y_head_inc'    }             1        0.029174         0.01505         0.13396         0.1432
    {'fl_cov_y_spouse'      }             0               0               0               0
    {'fl_cor_y_spouse'      }           NaN             NaN             NaN             NaN             Na
    {'fl_cov_yshr_nttxss'}         0.014829       0.0010034      1.7131e-05        0.066952         0.07114
    {'fl_cor_yshr_nttxss'}          0.99914        0.030003        0.014763         0.13431         0.1436
    {'fracByP0_01'          }    0.00042007       0.0079591               0               0      5.0103e-0
    {'fracByP10'            }      0.090622         0.05099               0       0.0059303       0.008103
    {'fracByP25'            }       0.22976         0.15254               0        0.060679        0.05259
    {'fracByP50'            }       0.47206         0.40219               0         0.22278          0.212
    {'fracByP75'            }       0.72831         0.67181               1         0.53644         0.4926
    {'fracByP90'            }       0.88939         0.87436               1         0.78432         0.7455
    {'fracByP99_99'         }             1               1               1         0.99942         0.9994

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Marital =0, kids =0, ybin =100 to 1414.0634
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx tb_outcomes: all stats xxx
    OriginalVariableNames        y_all          age_ss          educ_ss          a_ss           ap_ss
    _____________________     ___________     ___________     ___________     ___________     __________

    {'mean'                 }        164.25          47.879         0.35462          603.16          641.9
    {'unweighted_sum'       }    1.6654e+08            1909               1      1.2935e+05      1.4733e+0
```

```
{'sd'                 }        74.664        11.785        0.4784        524.59        535.2
{'coefofvar'          }       0.45456       0.24615         1.349       0.86973       0.8337
{'gini'               }       0.20972       0.13265       0.54013       0.41585       0.4099
{'min'                }           100            19             0             0        2.660
{'max'                }        1413.7            64             1        7837.6        8386.
{'pYis0'              }             0             0       0.64538      0.008464
{'pYls0'              }             0             0             0             0
{'pYgr0'              }             1             1       0.35462       0.99154
{'pYisMINY'           }   8.0846e-15     0.0084319       0.64538      0.008464
{'pYisMAXY'           }    9.784e-09      0.035671       0.35462    2.5784e-05    2.8187e-0
{'p0_01'              }        100.01            19             0             0        8.390
{'p10'                }        105.91            30             0        122.46        146.1
{'p25'                }        116.38            40             0        239.18        290.4
{'p50'                }        140.36            50             0        467.15        508.3
{'p75'                }        184.67            58             1        807.24        835.4
{'p90'                }         250.3            62             1        1175.1        1271.
{'p99_99'             }        1005.7            64             1        6140.4        6451.
{'fl_cov_y_all'       }        5574.6        82.616        3.9383         27029         2868
{'fl_cor_y_all'       }             1      0.093888       0.11026        0.6901       0.7178
{'fl_cov_age_ss'      }        82.616         138.9     -0.051378        3187.5        3233.
{'fl_cor_age_ss'      }      0.093888             1    -0.0091126       0.51557       0.5126
{'fl_cov_educ_ss'     }        3.9383     -0.051378       0.22886        1.1548        1.862
{'fl_cor_educ_ss'     }       0.11026    -0.0091126             1     0.0046015     0.007275
{'fl_cov_a_ss'        }         27029        3187.5        1.1548    2.7519e+05    2.8051e+0
{'fl_cor_a_ss'        }        0.6901       0.51557     0.0046015             1       0.9990
{'fl_cov_ap_ss'       }         28687        3233.7        1.8629    2.8051e+05    2.8647e+0
{'fl_cor_ap_ss'       }       0.71786       0.51265     0.0072754       0.99906
{'fl_cov_MPC'         }   -0.0039422      0.067815   -2.0374e-06        1.5699        1.574
{'fl_cor_MPC'         }   -0.0078548       0.85602   -0.00063355       0.44519       0.4376
{'fl_cov_Mass'        }  -3.2407e-05   -4.7599e-07   -1.5824e-07   -0.00016835   -0.0001761
{'fl_cor_Mass'        }     -0.36338     -0.033813      -0.27693      -0.26869      -0.2755
{'fl_cov_c_ss'        }          2511         15.67        2.2388         14895         1549
{'fl_cor_c_ss'        }       0.94083      0.037196       0.13092       0.79429       0.8098
{'fl_cov_y_head_inc'  }        5574.6        82.616        3.9383         27029         2868
{'fl_cor_y_head_inc'  }             1      0.093888       0.11026        0.6901       0.7178
{'fl_cov_y_spouse'    }             0             0             0             0
{'fl_cor_y_spouse'    }           NaN           NaN           NaN           NaN           Na
{'fl_cov_yshr_nttxss'}        0.64609      0.011641    0.00050086        3.1347        3.324
{'fl_cor_yshr_nttxss'}        0.90808       0.10365       0.10987       0.62707       0.6518
{'fracByP0_01'        }   7.5135e-05      0.003346             0             0     1.097e-0
{'fracByP10'          }     0.062666      0.056801             0      0.013271      0.01293
{'fracByP25'          }      0.16403       0.16984             0      0.057688      0.06415
{'fracByP50'          }      0.35792       0.40929             0         0.223       0.2196
{'fracByP75'          }      0.60086       0.72112             1       0.50267       0.4758
{'fracByP90'          }      0.79433       0.90509             1       0.70917       0.7129
{'fracByP99_99'       }      0.99932             1             1       0.99885       0.9988
```

13.1.12 Store Aggregate To File

Store Several Files:

1. Overall Aggregate Statistics All Distribution
2. Aggregate Statistics Only for 18 to 64 year olds
3. Group Statistics by Kids
4. Group Statistics by Marital + Kids
5. Group Statistics by Marital + Kids + Income Bins

```
if (bl_save_csv)
    % All Stats All Ages
    mp_path = snw_mp_path('fan');
    spt_simu_results_csv = mp_path('spt_simu_results_csv');
    writetable(tb_dist_stats_all, [spt_simu_results_csv 'stats_all_allages.csv'], 'WriteRowNames', t
    % All Stats 18 to 64 Year old
    mp_path = snw_mp_path('fan');
    spt_simu_results_csv = mp_path('spt_simu_results_csv');
    writetable(tb_dist_stats_all_18to64, [spt_simu_results_csv 'stats_all_18t64.csv'], 'WriteRowName
    % Group by K: Kids only
    tb_store_stats_by_k = array2table(mt_store_stats_by_k, 'VariableNames', ...
        {'kids', 'married_mean' ...
        'age_mean', 'age_p50', 'educ_mean', ...
        'a_mean', 'a_p50', 'ap_mean', 'ap_p50', ...
        'y_all_mean', 'y_all_p50', ...
        'mpc_mean', 'mpc_p50', ...
        'mass',...
        'c_ss_mean', 'c_ss_p50', ...
        'y_head_inc_mean', 'y_spouse_mean'});
    mp_path = snw_mp_path('fan');
    spt_simu_results_csv = mp_path('spt_simu_results_csv');
    writetable(tb_store_stats_by_k, [spt_simu_results_csv 'stats_by_kids.csv']);
    % Group by MK: marry + kids only
    tb_store_stats_by_mk = array2table(mt_store_stats_by_mk, 'VariableNames', ...
        {'marital', 'kids', ...
        'age_mean', 'age_p50', 'educ_mean', ...
        'a_mean', 'a_p50', 'ap_mean', 'ap_p50', ...
        'y_all_mean', 'y_all_p50', ...
        'mpc_mean', 'mpc_p50', ...
        'mass',...
        'c_ss_mean', 'c_ss_p50', ...
        'y_head_inc_mean', 'y_spouse_mean'});
    mp_path = snw_mp_path('fan');
    spt_simu_results_csv = mp_path('spt_simu_results_csv');
    writetable(tb_store_stats_by_mk, [spt_simu_results_csv 'stats_by_marital_kids.csv']);
    % Group by MKY
    tb_store_stats_by_mky = array2table(mt_store_stats_by_mky, 'VariableNames', ...
        {'marital', 'kids', 'y_all_start', 'y_all_end', ...
        'age_mean', 'age_p50', 'educ_mean', ...
        'a_mean', 'a_p50', 'ap_mean', 'ap_p50', ...
        'y_all_mean', 'y_all_p50', ...
        'mpc_mean', 'mpc_p50', ...
        'mass',...
        'c_ss_mean', 'c_ss_p50', ...
        'y_head_inc_mean', 'y_spouse_mean'});
    mp_path = snw_mp_path('fan');
    spt_simu_results_csv = mp_path('spt_simu_results_csv');
    writetable(tb_store_stats_by_mky, [spt_simu_results_csv 'stats_by_marital_kids_20kincbins.csv'])
end
```

13.1.13 Store Key Stats to Compare to Key US Distributional Statistics

Earning, income and Wealth.

Income = interest earnings + Social Security + labor income + spousal income. This is equal to y_all.

Earnings = labor income + spousal income.

```
% Income Variable
if (min(abs(total_inc_VFI*58.056 - y_all), [], 'all')>0)
```

```
    error('someothing is wrong, total_inc_VFI should be equal to y_all');
end
income = y_all;
% Earning variable
% earn*fl_earn_ratio generated earn_VFI
earning = (mp_valpol_more_ss('earn_VFI') + spouse_inc_VFI)*58.056;
% Wealth Varaible
wealth = a_ss;
```

Generate Key Statistics for these three variables only, distributional Statistics Overall All Ages:

```
% construct input data
income_grp = income(min_age:82, :, :, : ,: ,:);
earning_grp = earning(min_age:82, :, :, : ,: ,:);
wealth_grp = wealth(min_age:82, :, :, : ,: ,:);
Phi_true_grp = Phi_true_1(min_age:82, :, :, : ,: ,:);

mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
mp_cl_ar_xyz_of_s('earning') = {earning_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('income') = {income_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('wealth') = {wealth_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('earninglog') = {log(earning_grp(:)), zeros(1)};
mp_cl_ar_xyz_of_s('incomelog') = {log(income_grp(:)), zeros(1)};
mp_cl_ar_xyz_of_s('wealthlog') = {log(wealth_grp(:)), zeros(1)};
mp_cl_ar_xyz_of_s('ar_st_y_name') = ["earning", "income", "wealth", "earninglog", "incomelog", "weal

% controls
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('ar_fl_percentiles') = [20 30 40 60 50 80 90 95 99];
mp_support('bl_display_final') = true;
mp_support('bl_display_detail') = false;
mp_support('bl_display_drvm2outcomes') = false;
mp_support('bl_display_drvstats') = false;
mp_support('bl_display_drvm2covcor') = false;

% Call Function
mp_cl_mt_xyz_of_s = ff_simu_stats(Phi_true_grp(:)/sum(Phi_true_grp,'all'), mp_cl_ar_xyz_of_s, mp_sup

xxx tb_outcomes: all stats xxx
    OriginalVariableNames     earning       income        wealth      earninglog    incomelog     w
    _____________________    __________    __________    __________    __________    __________    _

    {'mean'             }        72.136        84.974        245.22          -Inf        4.1042
    {'unweighted_sum'   }    9.5943e+07    7.9255e+09    1.2935e+05          -Inf    1.1455e+08
    {'sd'               }        80.749        84.549        391.42           NaN       0.81216
    {'coefofvar'        }        1.1194         0.995        1.5962           NaN       0.19789
    {'gini'             }       0.51369       0.44243       0.68023           NaN       0.11243
    {'min'              }             0        2.2124             0          -Inf       0.79408
    {'max'              }          2640        2953.5        7837.6        7.8785        7.9907
    {'pYis0'            }       0.10578             0       0.12285             0             0
    {'pYls0'            }             0             0             0       0.10695             0
    {'pYgr0'            }       0.89422             1       0.87715       0.89305             1
    {'pYisMINY'         }       0.10578     6.774e-07       0.12285       0.10578     6.774e-07
    {'pYisMAXY'         }    1.5964e-10     1.671e-12    6.0119e-06    1.5964e-10     1.671e-12    6
    {'p20'              }        15.969        29.216        3.7372        2.7707        3.3747
    {'p30'              }        29.464        38.184        15.308        3.3832        3.6424
    {'p40'              }        40.761        48.225        39.794        3.7077        3.8759
    {'p60'              }        65.423        74.426        146.89        4.1809        4.3098
    {'p50'              }        52.252        59.948         82.04        3.9561        4.0935
```

```
    {'p80'              }       108.96         122.39         413.31          4.691        4.8072
    {'p90'              }        159.7         176.61         729.18         5.0733        5.1739
    {'p95'              }       211.84         233.69         979.69         5.3558         5.454
    {'p99'              }       356.31         398.22         1773.5         5.8758         5.987
    {'fl_cov_earning'   }       6520.5         6671.7         8382.5            NaN        53.875
    {'fl_cor_earning'   }            1        0.97721        0.26521            NaN       0.82149
    {'fl_cov_income'    }       6671.7         7148.6          15059            NaN        57.878
    {'fl_cor_income'    }      0.97721              1        0.45504            NaN       0.84286
    {'fl_cov_wealth'    }       8382.5          15059     1.5321e+05            NaN        141.72
    {'fl_cor_wealth'    }      0.26521        0.45504              1            NaN        0.4458
    {'fl_cov_earninglog'}          NaN            NaN            NaN            NaN           NaN
    {'fl_cor_earninglog'}          NaN            NaN            NaN            NaN           NaN
    {'fl_cov_incomelog' }       53.875         57.878         141.72            NaN       0.65961
    {'fl_cor_incomelog' }      0.82149        0.84286         0.4458            NaN             1
    {'fl_cov_wealthlog' }          NaN            NaN            NaN            NaN           NaN
    {'fl_cor_wealthlog' }          NaN            NaN            NaN            NaN           NaN
    {'fracByP20'        }     0.012671        0.04827     0.00074821            NaN       0.14532
    {'fracByP30'        }     0.044498        0.08795      0.0041711            NaN       0.23096
    {'fracByP40'        }     0.093262        0.13869       0.016749            NaN       0.32262
    {'fracByP60'        }      0.23895        0.28076       0.095501            NaN       0.52207
    {'fracByP50'        }      0.15762        0.20209       0.045325            NaN       0.41971
    {'fracByP80'        }      0.47178        0.50479        0.32852            NaN       0.74357
    {'fracByP90'        }      0.65353         0.6766        0.56651            NaN       0.86486
    {'fracByP95'        }      0.78022        0.79527        0.70071            NaN       0.92947
    {'fracByP99'        }      0.92468        0.93132        0.90524            NaN       0.98459
```

```
tb_dist_stats_all = mp_cl_mt_xyz_of_s('tb_outcomes');
% Select columns
tb_dist_stats_all_save = tb_dist_stats_all(1:3,:);
ar_st_columns = ["coefofvar", "gini", "varianceoflog", ...
                 "p99p50ratio", "p90p50ratio", "meantomedian", "p50p30ratio", ...
                 "fracP0toP20", "fracP20toP40", "fracP40toP60", "fracP60toP80", "fracP80toP100", ...
                 "fracP90toP95", "fracP95toP99", "fracP99toP100"];

varianceoflog = tb_dist_stats_all{4:6,"sd"}.^2;

p99p50ratio = tb_dist_stats_all_save{:,"p99"}./tb_dist_stats_all_save{:,"p50"};
p90p50ratio = tb_dist_stats_all_save{:,"p90"}./tb_dist_stats_all_save{:,"p50"};
meantomedian = tb_dist_stats_all_save{:,"mean"}./tb_dist_stats_all_save{:,"p50"};
p50p30ratio = tb_dist_stats_all_save{:,"p50"}./tb_dist_stats_all_save{:,"p30"};
fracP0toP20 = tb_dist_stats_all_save{:,"fracByP20"};
fracP20toP40 = tb_dist_stats_all_save{:,"fracByP40"} - tb_dist_stats_all_save{:,"fracByP20"};
fracP40toP60 = tb_dist_stats_all_save{:,"fracByP60"} - tb_dist_stats_all_save{:,"fracByP40"};
fracP60toP80 = tb_dist_stats_all_save{:,"fracByP80"} - tb_dist_stats_all_save{:,"fracByP60"};
fracP80toP100 = 1 - tb_dist_stats_all_save{:,"fracByP80"};

fracP90toP95 = tb_dist_stats_all_save{:,"fracByP95"} - tb_dist_stats_all_save{:,"fracByP90"};
fracP95toP99 = tb_dist_stats_all_save{:,"fracByP99"} - tb_dist_stats_all_save{:,"fracByP95"};
fracP99toP100 = 1 - tb_dist_stats_all_save{:,"fracByP99"};

tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, varianceoflog, 'Before', 'gini');
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, p99p50ratio);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, p90p50ratio);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, meantomedian);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, p50p30ratio);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP0toP20);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP20toP40);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP40toP60);
```

```
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP60toP80);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP80toP100);

tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP90toP95);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP95toP99);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP99toP100);
disp(tb_dist_stats_all_save(:, ar_st_columns));
```

```
               coefofvar     gini      varianceoflog    p99p50ratio    p90p50ratio    meantomedian
               _________    _______    _____________    ___________    ___________    ____________

    earning     1.1194      0.51369            NaN          6.819         3.0563           1.3805
    income       0.995      0.44243        0.65961         6.6427          2.946           1.4174
    wealth      1.5962      0.68023            NaN         21.618         8.8881            2.989
```

```
% Core Stats Table
if (bl_save_csv)
    mp_path = snw_mp_path('fan');
    spt_simu_results_csv = mp_path('spt_simu_results_csv');
    writetable(tb_dist_stats_all_save(:, ar_st_columns), [spt_simu_results_csv 'stats_all_allages_vr
end
```

Statistics overall distributionally for 18 to 64 year olds.

```
% construct input data
income_grp = income(min_age:max_age, :, :, : ,: ,:);
earning_grp = earning(min_age:max_age, :, :, : ,: ,:);
wealth_grp = wealth(min_age:max_age, :, :, : ,: ,:);
Phi_true_grp = Phi_true_1(min_age:max_age, :, :, : ,: ,:);

mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
mp_cl_ar_xyz_of_s('income') = {income_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('earning') = {earning_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('wealth') = {wealth_grp(:), zeros(1)};
mp_cl_ar_xyz_of_s('earninglog') = {log(earning_grp(:)), zeros(1)};
mp_cl_ar_xyz_of_s('incomelog') = {log(income_grp(:)), zeros(1)};
mp_cl_ar_xyz_of_s('wealthlog') = {log(wealth_grp(:)), zeros(1)};
mp_cl_ar_xyz_of_s('ar_st_y_name') = ["earning", "income", "wealth", "earninglog", "incomelog", "weal

% controls
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('ar_fl_percentiles') = [20 30 40 60 50 80 90 95 99];
mp_support('bl_display_final') = true;
mp_support('bl_display_detail') = false;
mp_support('bl_display_drvm2outcomes') = false;
mp_support('bl_display_drvstats') = false;
mp_support('bl_display_drvm2covcor') = false;

% Call Function
mp_cl_mt_xyz_of_s = ff_simu_stats(Phi_true_grp(:)/sum(Phi_true_grp,'all'), mp_cl_ar_xyz_of_s, mp_sup
```

```
xxx tb_outcomes: all stats xxx
    OriginalVariableNames     earning        income        wealth      earninglog    incomelog     w
    _____________________    __________    __________    __________    __________    __________    _

    {'mean'             }        87.466        95.246         194.5        4.1711        4.2425
    {'unweighted_sum'   }     9.394e+07    7.7487e+09    1.2935e+05    1.5445e+06     1.116e+08
    {'sd'               }        82.434        89.631         344.5       0.76834       0.79264
    {'coefofvar'        }       0.94247       0.94104        1.7712        0.1842       0.18683
```

```
    {'gini'            }         0.417        0.42428        0.71579        0.10382         0.1055
    {'min'             }        2.2124         2.2124              0        0.79408        0.79408
    {'max'             }          2640         2953.5         7837.6         7.8785         7.9907
    {'pYis0'           }             0              0        0.14627              0              0
    {'pYls0'           }             0              0              0              0              0
    {'pYgr0'           }             1              1        0.85373              1              1
    {'pYisMINY'        }     8.617e-07     8.6135e-07        0.14627      8.617e-07     8.6135e-07
    {'pYisMAXY'        }    2.0299e-10     2.1248e-12     5.4766e-06     2.0299e-10     2.1248e-12    5
    {'p20'             }        34.093         35.624        0.80724         3.5291          3.573
    {'p30'             }        43.249         45.828          6.458          3.767         3.8249
    {'p40'             }        52.993         56.888         29.898         3.9702         4.0411
    {'p60'             }        77.857         85.184         100.91         4.3549         4.4448
    {'p50'             }         64.26          69.57         51.664         4.1629         4.2423
    {'p80'             }        124.43         137.12         318.35         4.8237         4.9209
    {'p90'             }        175.33          192.9         588.48         5.1667         5.2621
    {'p95'             }        227.34         250.34         890.69         5.4265         5.5228
    {'p99'             }        384.15         427.18         1640.6          5.951         6.0572
    {'fl_cov_earning'  }        6795.4         7319.6          13105           53.1         53.884
    {'fl_cor_earning'  }             1        0.99065        0.46144        0.83837        0.82467
    {'fl_cov_income'   }        7319.6         8033.6          17852         58.043         59.852
    {'fl_cor_income'   }       0.99065              1        0.57814        0.84283        0.84246
    {'fl_cov_wealth'   }         13105          17852     1.1868e+05         123.58          149.2
    {'fl_cor_wealth'   }       0.46144        0.57814              1        0.46687         0.5464
    {'fl_cov_earninglog'}         53.1         58.043         123.58        0.59034         0.6043
    {'fl_cor_earninglog'}      0.83837        0.84283        0.46687              1        0.99226
    {'fl_cov_incomelog' }       53.884         59.852          149.2         0.6043        0.62827
    {'fl_cor_incomelog' }      0.82467        0.84246         0.5464        0.99226              1
    {'fl_cov_wealthlog' }          NaN            NaN            NaN            NaN            NaN
    {'fl_cor_wealthlog' }          NaN            NaN            NaN            NaN            NaN
    {'fracByP20'       }      0.053802       0.050961     0.00014055        0.14882        0.14762
    {'fracByP30'       }      0.098055       0.093694      0.0021143        0.23646        0.23488
    {'fracByP40'       }         0.153        0.14753       0.015697         0.3292        0.32764
    {'fracByP60'       }       0.30069        0.29468       0.079605        0.52874        0.52766
    {'fracByP50'       }       0.21981        0.21374       0.034043        0.42667        0.42529
    {'fracByP80'       }       0.52452        0.52079        0.28918        0.74816         0.7478
    {'fracByP90'       }       0.69236        0.69054        0.51495        0.86758        0.86757
    {'fracByP95'       }       0.80576        0.80501        0.69371        0.93096        0.93099
    {'fracByP99'       }       0.93293        0.93437        0.90041        0.98483        0.98492

tb_dist_stats_all = mp_cl_mt_xyz_of_s('tb_outcomes');
% Select columns
tb_dist_stats_all_save = tb_dist_stats_all(1:3,:);
ar_st_columns = ["coefofvar", "gini", "varianceoflog", ...
                 "p99p50ratio", "p90p50ratio", "meantomedian", "p50p30ratio", ...
                 "fracP0toP20", "fracP20toP40", "fracP40toP60", "fracP60toP80", "fracP80toP100", ...
                 "fracP90toP95", "fracP95toP99", "fracP99toP100"];

varianceoflog = tb_dist_stats_all{4:6,"sd"}.^2;

p99p50ratio = tb_dist_stats_all_save{:,"p99"}./tb_dist_stats_all_save{:,"p50"};
p90p50ratio = tb_dist_stats_all_save{:,"p90"}./tb_dist_stats_all_save{:,"p50"};
meantomedian = tb_dist_stats_all_save{:,"mean"}./tb_dist_stats_all_save{:,"p50"};
p50p30ratio = tb_dist_stats_all_save{:,"p50"}./tb_dist_stats_all_save{:,"p30"};
fracP0toP20 = tb_dist_stats_all_save{:,"fracByP20"};
fracP20toP40 = tb_dist_stats_all_save{:,"fracByP40"} - tb_dist_stats_all_save{:,"fracByP20"};
fracP40toP60 = tb_dist_stats_all_save{:,"fracByP60"} - tb_dist_stats_all_save{:,"fracByP40"};
fracP60toP80 = tb_dist_stats_all_save{:,"fracByP80"} - tb_dist_stats_all_save{:,"fracByP60"};
fracP80toP100 = 1 - tb_dist_stats_all_save{:,"fracByP80"};
```

```
fracP90toP95 = tb_dist_stats_all_save{:,"fracByP95"} - tb_dist_stats_all_save{:,"fracByP90"};
fracP95toP99 = tb_dist_stats_all_save{:,"fracByP99"} - tb_dist_stats_all_save{:,"fracByP95"};
fracP99toP100 = 1 - tb_dist_stats_all_save{:,"fracByP99"};

tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, varianceoflog, 'Before', 'gini');
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, p99p50ratio);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, p90p50ratio);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, meantomedian);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, p50p30ratio);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP0toP20);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP20toP40);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP40toP60);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP60toP80);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP80toP100);

tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP90toP95);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP95toP99);
tb_dist_stats_all_save = addvars(tb_dist_stats_all_save, fracP99toP100);
disp(tb_dist_stats_all_save(:, ar_st_columns));

               coefofvar     gini      varianceoflog    p99p50ratio    p90p50ratio    meantomedian
               _________    _______    _____________    ___________    ___________    ____________

    earning     0.94247       0.417       0.59034          5.978         2.7285          1.3611
    income      0.94104     0.42428       0.62827         6.1403         2.7727          1.3691
    wealth       1.7712     0.71579           NaN         31.755         11.391          3.7648

% Core Stats Table
if (bl_save_csv)
    mp_path = snw_mp_path('fan');
    spt_simu_results_csv = mp_path('spt_simu_results_csv');
    writetable(tb_dist_stats_all_save(:, ar_st_columns), [spt_simu_results_csv 'stats_all_18t64_vrrc
end
```

Chapter 14

Selected Comparative Statistics

14.1 Value and Consumption Low vs Higher Interest Rates Results Comparison

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for the V(states) for individuals at lower and higher savings interest rates. Note that welfare improves for all when interest rates go up for savings in a model where borrowing is not allowed. However, a change in the interest rate generates an income effect (higher resources) and changes the relative price of consumption today vs. tomorrow. The change in income increases the incentive to consume, but the change in relative price depresses incentives to consume today. The combined effect of rising interest rate on savings on consumption/savings differs by the state-space; households might overall consume more or less depending on their state-space.

14.1.1 Solve Model at 4 Percent Interest Rate

Solve the benchmark model at 4 percent savings interest rate.

```
% mp_params = snw_mp_param('default_dense');
mp_params = snw_mp_param('default_docdense');
mp_params('beta') = 0.95;
fl_higher_r = 0.04;
fl_lower_r = 0.02;
mp_params('r') = fl_higher_r;
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_timer') = true;
[V_ss_r04,~,cons_ss_r04,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=496.
```

14.1.2 Solve Model at 2 Percent Interest Rate

Solve the benchmark model at 2 percent savings interest rate.

```
mp_params('r') = fl_lower_r;
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_timer') = true;
[V_ss_r02,~,cons_ss_r02,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=495.
```

14.1.3 Generate Interest Rate Comparison Matrixes

Take the difference between 4 percent and 2 percent savings interest rate results. When interest rates are higher, greater incentive to save, but leads to heterogeneous responses by income and other characteristics, note that this changes both relative prices as well as total resource/budget, so there is both income and price effects that differ in magnitudes depending on the individual's statespace. Welfare does improve for all higher higher r. Welfare is converted to units in fixed life-time consumption.

```
gamma = mp_params('gamma');
mn_V_gain_r = snw_hh_welfare(V_ss_r04, gamma) - snw_hh_welfare(V_ss_r02, gamma);
mn_C_gain_r = cons_ss_r04 - cons_ss_r02;
fl_r_gap = fl_higher_r - fl_lower_r;
```

14.1.4 Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

14.1.5 Analyze Difference in V and C with Higher and Lower Savings Interest Rate

The difference between V and C with higher and lower r.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(MN_V_GAIN(A,Z))

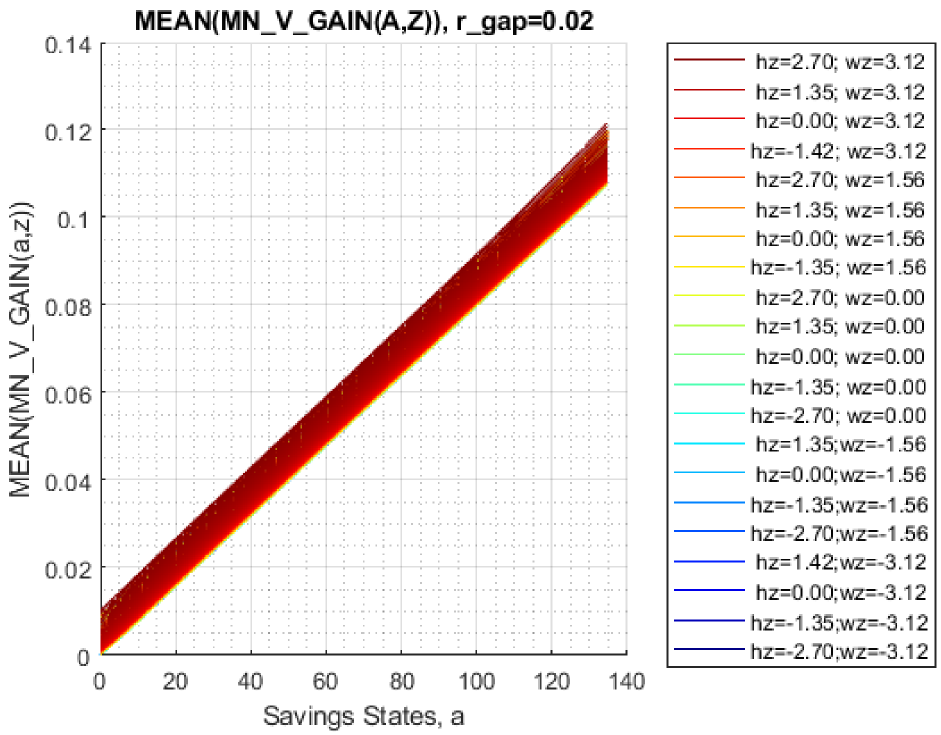
Tabulate value and policies along savings and shocks:

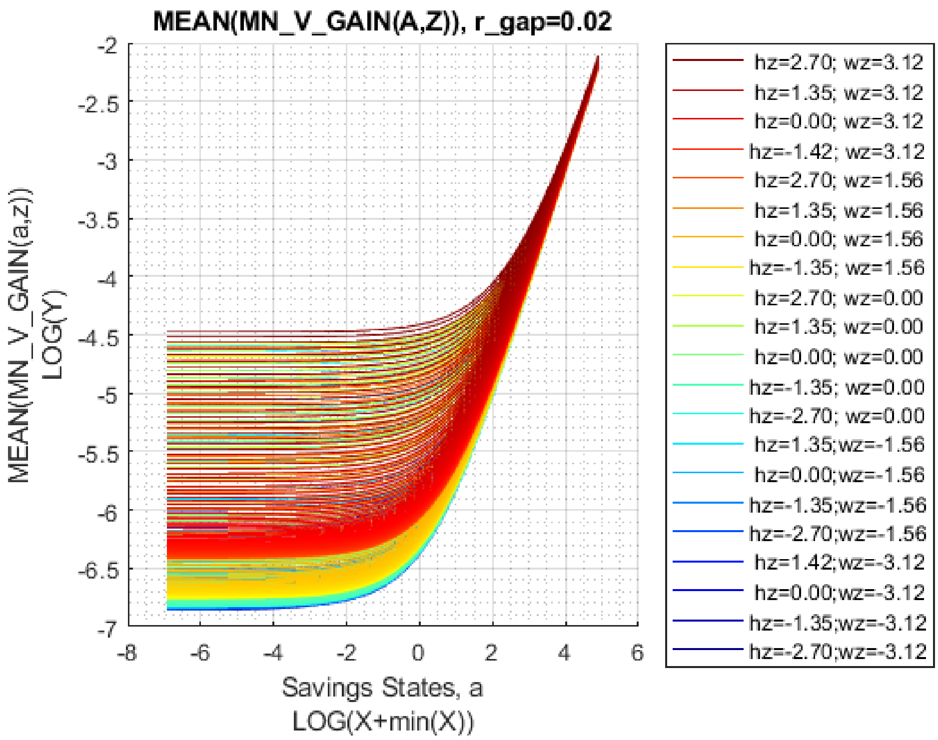
```
% Set
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_Gain(A,Z)), r_gap=' num2str(fl_r_gap) ];
tb_az_v = ff_summ_nd_array(st_title, mn_V_gain_r, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_permut
```

```
xxx  MEAN(MN_V_Gain(A,Z)), r_gap=0.02  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---
```

```
1                0    4.6652e-05     4.802e-05     4.9487e-05     5.1045e-05     5.2688e-05     5.4
```

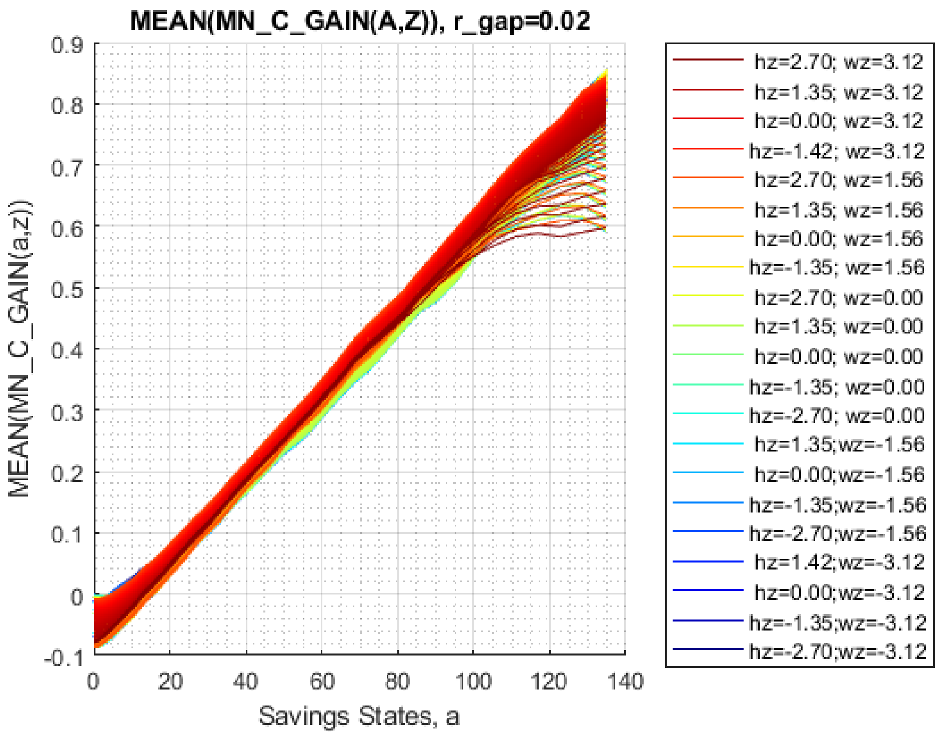
```
st_title = ['MEAN(MN\_V\_GAIN(A,Z)), r\_gap=' num2str(fl_r_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_GAIN(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

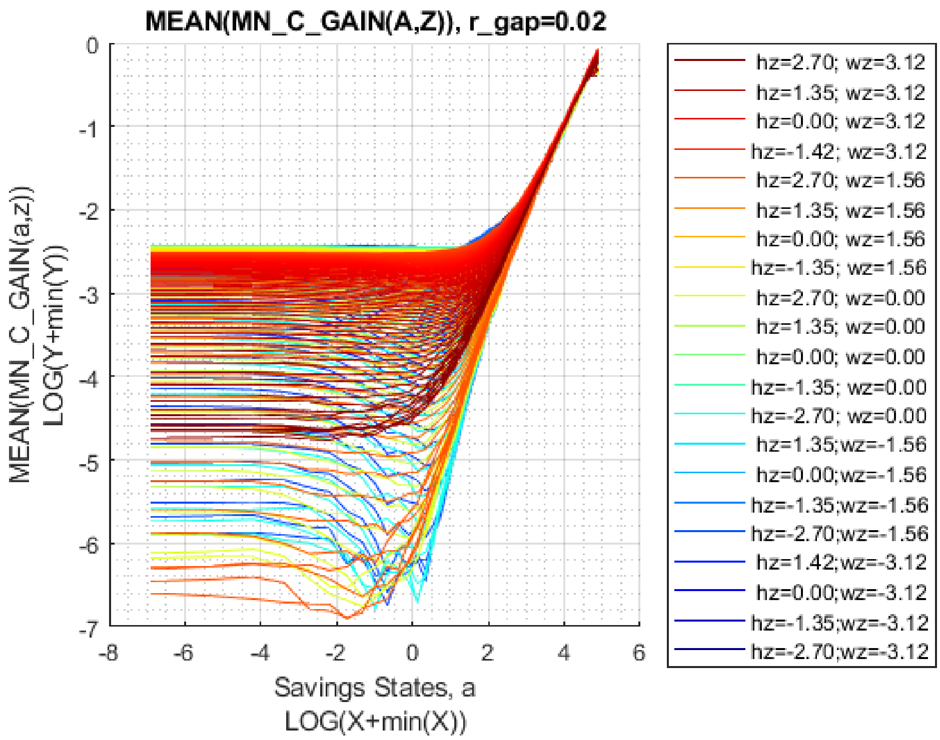




Graph Mean Consumption:

```
st_title = ['MEAN(MN\_C\_GAIN(A,Z)), r\_gap=' num2str(fl_r_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_C\_GAIN(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





14.1.6 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
st_title = ['MEAN(MN_V_Gain(KM,J)), r_gap=' num2str(fl_r_gap) ];
tb_az_v = ff_summ_nd_array(st_title, mn_V_gain_r, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permut
```

```
xxx  MEAN(MN_V_Gain(KM,J)), r_gap=0.02  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        0.015087       0.015546       0.016026        0.01653        0.017025
```

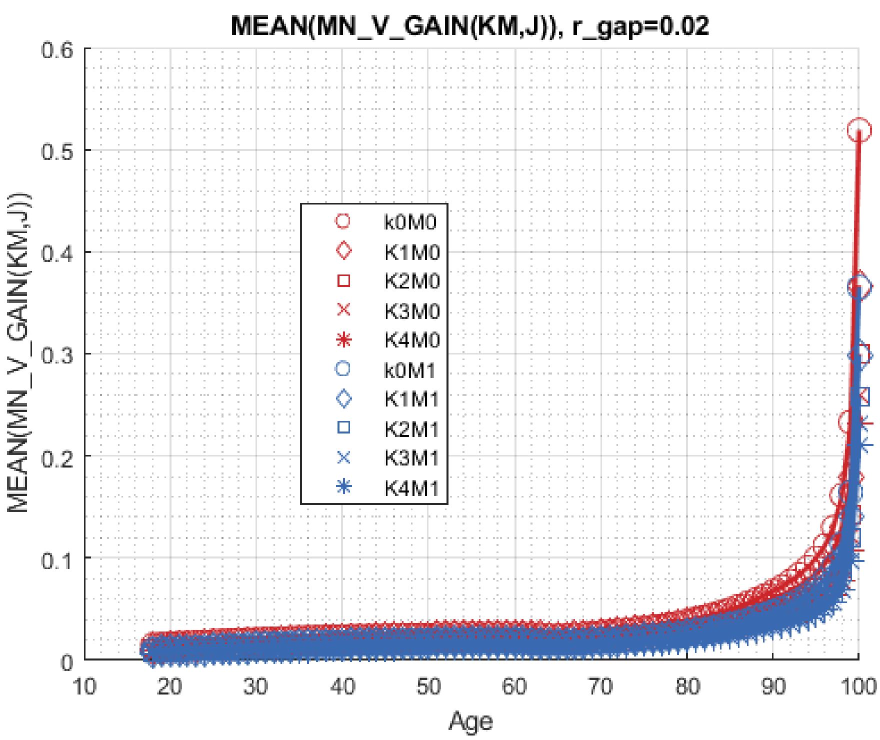
```
     2        2       0      0.011707      0.012061      0.012444      0.012858      0.013278
     3        3       0      0.010499      0.010777      0.011081      0.011412      0.011749
     4        4       0     0.0094894     0.0097107     0.0099547      0.010224        0.0105
     5        5       0       0.00886     0.0090319     0.0092234     0.0094371     0.0096558
     6        1       1      0.010229      0.010593      0.010984      0.011405      0.011831
     7        2       1      0.008528     0.0087978     0.0090932     0.0094182     0.0097531
     8        3       1     0.0079012     0.0081252     0.0083716     0.0086439     0.0089244
     9        4       1     0.0073246     0.0075062     0.0077075     0.0079319     0.0081637
    10        5       1       0.00696     0.0071075     0.0072717     0.0074553      0.007644
```

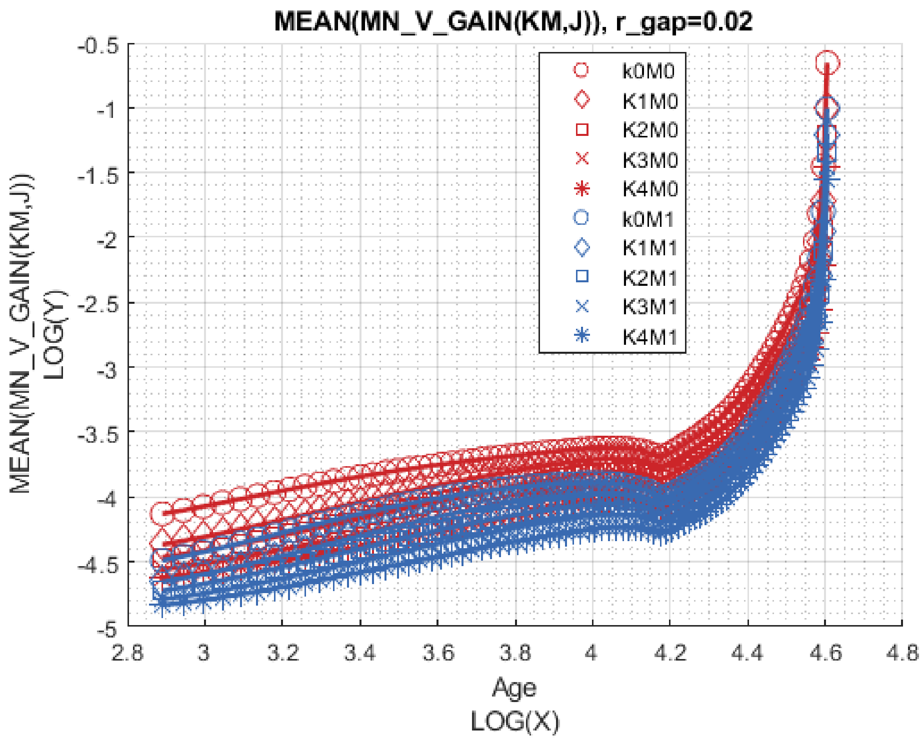
```
% Consumption Function
st_title = ['MEAN(MN_C_Gain(KM,J)), r_gap=' num2str(fl_r_gap) ];
tb_az_c = ff_summ_nd_array(st_title, mn_C_gain_r, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permut
```

```
xxx  MEAN(MN_C_Gain(KM,J)), r_gap=0.02  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0       0.055699       0.056856       0.058874       0.061289       0.064119
      2        2       0       0.051038       0.052145       0.054007       0.055436       0.057546
      3        3       0       0.050801        0.05338       0.054644       0.055147        0.05643
      4        4       0       0.050711       0.052919       0.054103       0.053992        0.05454
      5        5       0       0.052196       0.053297       0.054608       0.053912       0.053811
      6        1       1      -0.022203      -0.022187      -0.021707      -0.021466      -0.020507
      7        2       1      -0.019706      -0.020509      -0.020942       -0.02173      -0.021723
      8        3       1      -0.016291      -0.017215      -0.017899      -0.019714       -0.02062
      9        4       1      -0.013267      -0.013589      -0.013895      -0.016006      -0.017459
     10        5       1        -0.0131      -0.012701      -0.011796      -0.013633      -0.014917
```

Graph Mean Values:

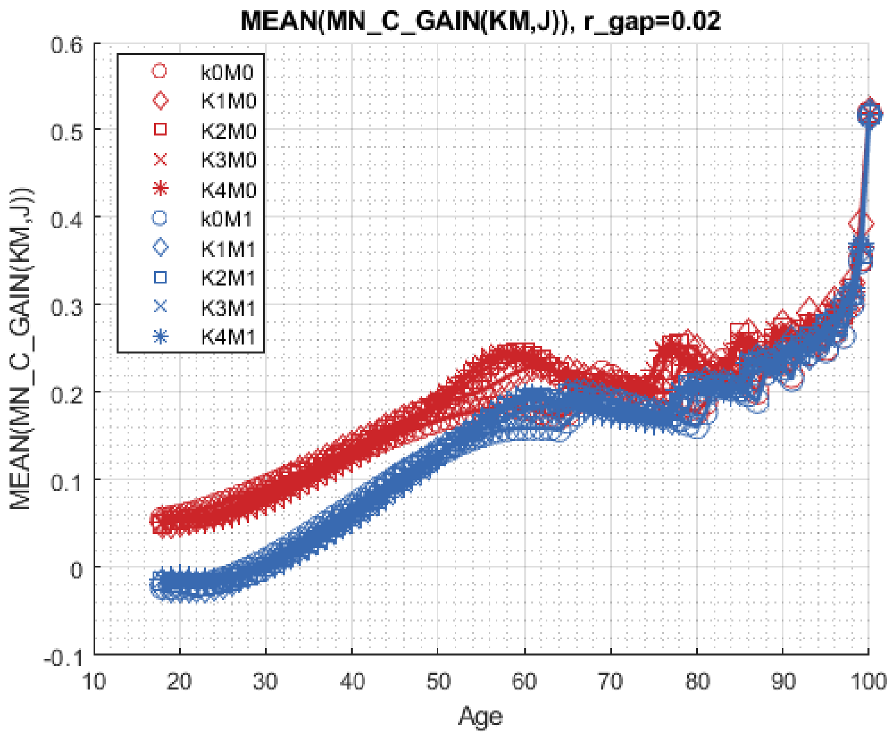
```
st_title = ['MEAN(MN\_V\_GAIN(KM,J)), r\_gap=' num2str(fl_r_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_GAIN(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

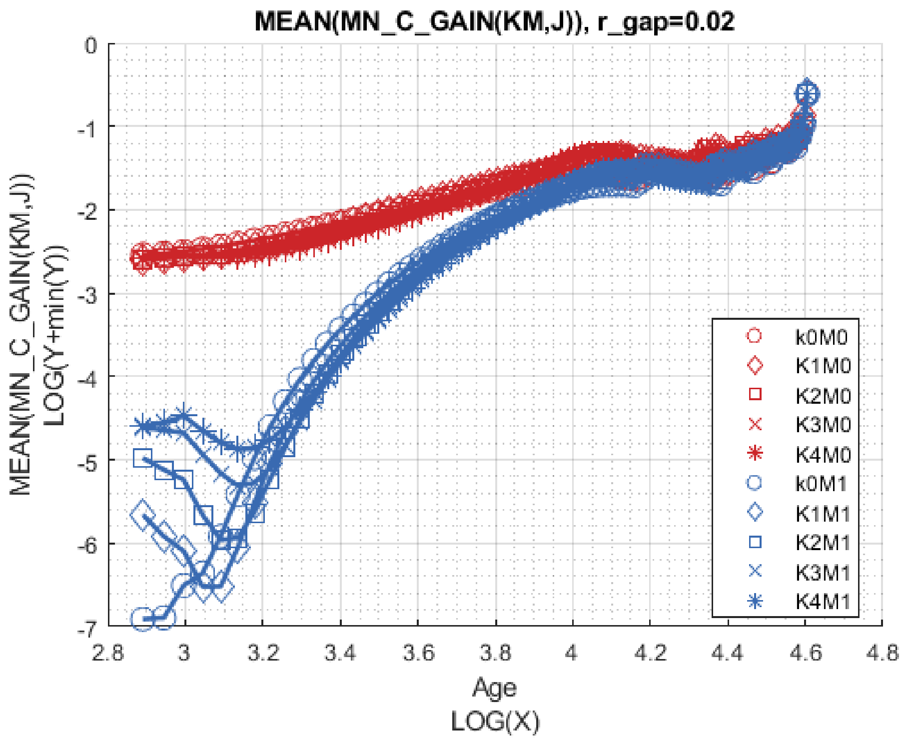




Graph Mean Consumption:

```
st_title = ['MEAN(MN\_C\_GAIN(KM,J)), r\_gap=' num2str(fl_r_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_C\_GAIN(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





14.1.7 Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_Gain(EM,J)), r_gap=' num2str(fl_r_gap) ];
tb_az_v = ff_summ_nd_array(st_title, mn_V_gain_r, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permut
```

```
xxx  MEAN(MN_V_Gain(EM,J)), r_gap=0.02  xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     edu     marry     mean_age_18     mean_age_19     mean_age_20     mean_age_21     mean_age_22
    -----     ---     -----     -----------     -----------     -----------     -----------     -----------

      1        0        0         0.011474        0.011736        0.012012        0.012305        0.012599
      2        1        0         0.010783        0.011115        0.011479        0.011879        0.012285
      3        0        1        0.0084099       0.0086195       0.0088444       0.0090861       0.0093328
      4        1        1        0.0079671       0.0082324       0.0085268       0.0088556       0.0091936
```

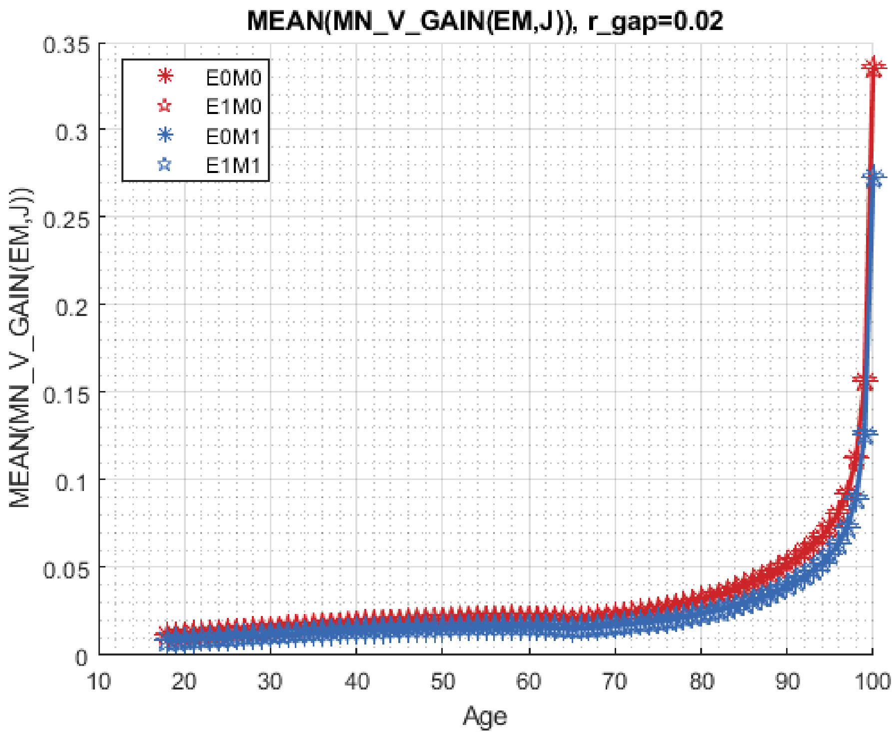
```
% Consumption
st_title = ['MEAN(MN_C_Gain(EM,J)), r_gap=' num2str(fl_r_gap) ];
```

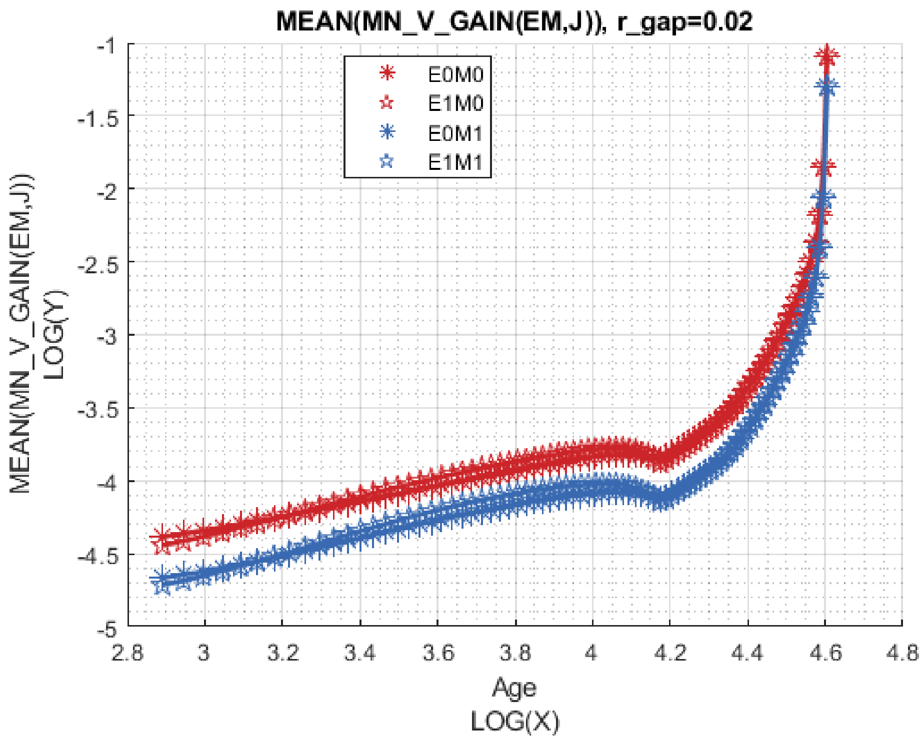
```
tb_az_c = ff_summ_nd_array(st_title, mn_C_gain_r, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permut
```

```
xxx  MEAN(MN_C_Gain(EM,J)), r_gap=0.02  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     edu     marry     mean_age_18     mean_age_19     mean_age_20     mean_age_21     mean_age_22
    _____     ___     _____     ___________     ___________     ___________     ___________     ___________

      1        0        0           0.06916        0.070507        0.072258         0.07446        0.076888
      2        1        0          0.035018        0.036932        0.038237         0.03745        0.037691
      3        0        1        -0.0014877       -0.001014     -0.00023405      0.00057466       0.0018134
      4        1        1         -0.032339       -0.033466       -0.034261       -0.037595       -0.039904
```

Graph Mean Values:

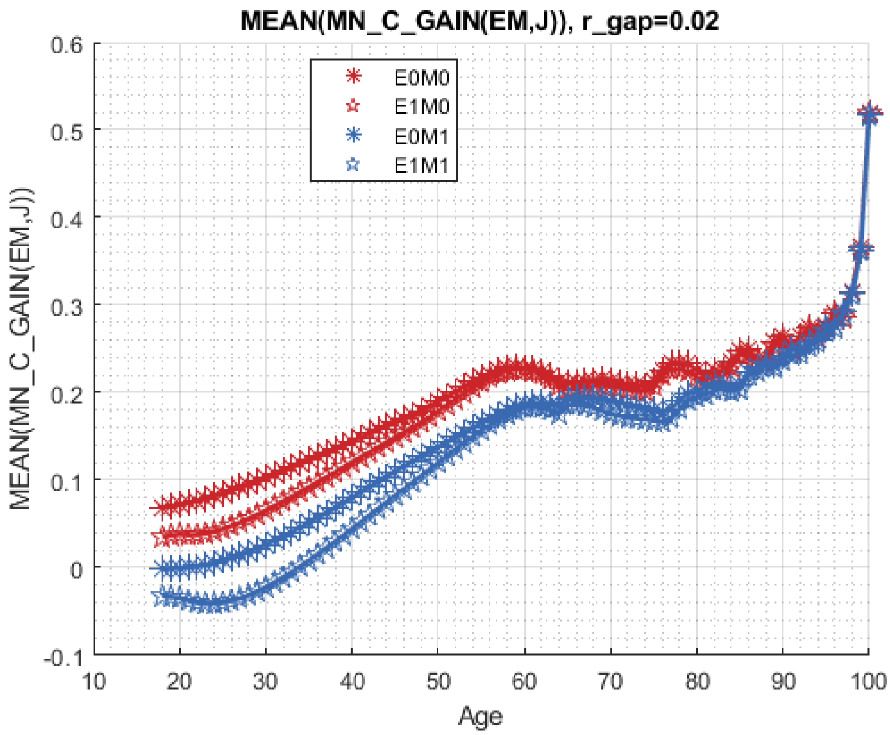
```
st_title = ['MEAN(MN\_V\_GAIN(EM,J)), r\_gap=' num2str(fl_r_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_GAIN(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

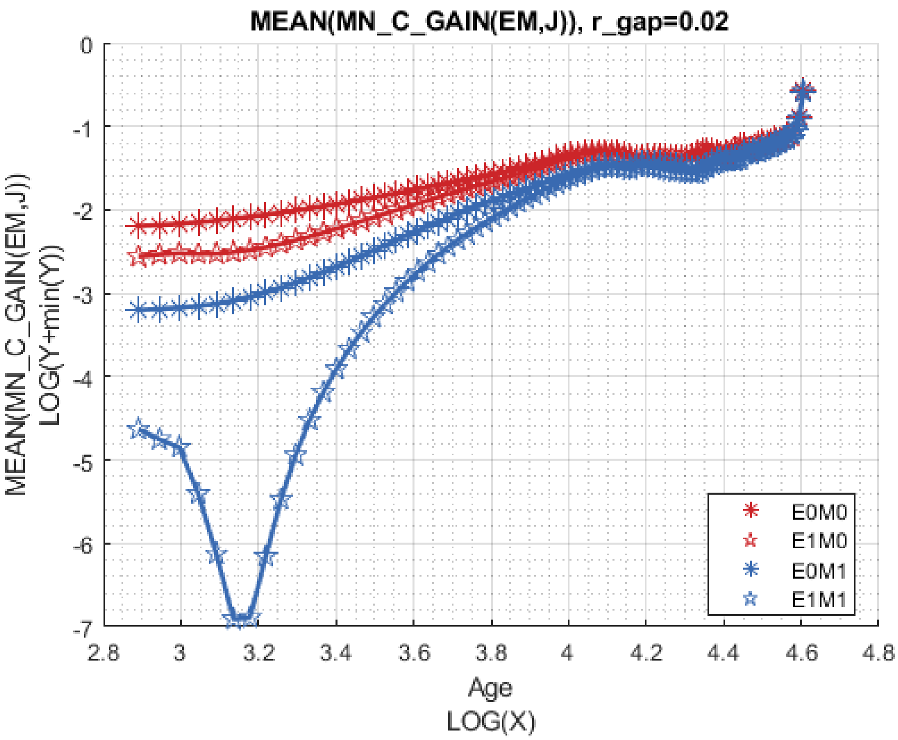




Graph Mean Consumption:

```
st_title = ['MEAN(MN\_C\_GAIN(EM,J)), r\_gap=' num2str(fl_r_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_C\_GAIN(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





14.2 Value and Consumption Low vs Higher Unemployment Insurance Comparison

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for the V(states, unemployed) assuming individuals suffer from unemployment spell, but with different UI (unemployment benefits). Higher UI benefits leads to value/welfare and also higher consumption.

14.2.1 Solve the Steady-State non-unemployment Problem

Solve for Value/Policy in non-COVID years, then solve for covid year value/policy given covid shocks. COVID lasts one period.

```
% mp_params = snw_mp_param('default_dense');
mp_params = snw_mp_param('default_docdense');
mp_params('beta') = 0.95;
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_timer') = true;
[V_ss,~,cons_ss,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=491.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN     colN          sum          mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -6.6619e+08    -15.245    21.86
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.3967e+09     31.962    36.42
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05     2.3276e+08     5.3263    8.441
```

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	-293.96	-293.57	-291.09	-285.44	-276.41	-4.3584	-4.2643	-4.171
r2	-284.42	-284.03	-281.55	-275.97	-267.24	-4.2519	-4.1612	-4.071
r3	-274.87	-274.48	-272.03	-266.62	-258.33	-4.1429	-4.0559	-3.969
r4	-265.22	-264.86	-262.58	-257.53	-249.74	-4.0309	-3.9475	-3.864
r5	-256.51	-256.17	-254.04	-249.3	-241.96	-3.9252	-3.8452	-3.765
r79	-13.642	-13.628	-13.535	-13.298	-12.896	-0.22092	-0.21058	-0.2008
r80	-12.283	-12.269	-12.176	-11.939	-11.537	-0.16979	-0.16182	-0.154
r81	-10.605	-10.591	-10.498	-10.261	-9.8589	-0.11712	-0.11163	-0.1064
r82	-8.3494	-8.3358	-8.2424	-8.0055	-7.6035	-0.065333	-0.062242	-0.0593
r83	-5.0665	-5.0529	-4.9595	-4.7226	-4.3206	-0.020968	-0.019972	-0.01903

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c5264
r1	0	0	0.00051498	0.0066578	0.021589	112.13	117.67	123.4	129.3
r2	0	0	0.00051498	0.0057684	0.020245	112.17	117.71	123.43	129.3
r3	0	0	0.00020768	0.0041456	0.018539	112.2	117.73	123.45	129.3
r4	0	0	0.00010346	0.0041199	0.018307	112.86	118.39	124.11	130.0
r5	0	0	5.2907e-06	0.0041199	0.018091	113.53	119.07	124.79	130.7
r79	0	0	0	0	0	81.091	85.364	89.335	93.25
r80	0	0	0	0	0	76.124	79.747	83.431	86.98
r81	0	0	0	0	0	67.945	70.639	73.673	76.99
r82	0	0	0	0	0	50.126	53.467	56.302	57.88
r83	0	0	0	0	0	0	0	0	

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
r1	0.036717	0.037251	0.040477	0.044486	0.049324	12.265	12.55	12.844
r2	0.036717	0.037251	0.040477	0.045375	0.050668	12.501	12.787	13.082
r3	0.036717	0.037251	0.040784	0.046998	0.052374	12.755	13.042	13.337
r4	0.038144	0.038678	0.042314	0.048449	0.054031	13	13.289	13.584
r5	0.039534	0.040068	0.043802	0.049839	0.055635	13.236	13.525	13.821
r79	0.19737	0.19791	0.20163	0.21175	0.23145	35.811	37.362	39.409
r80	0.19737	0.19791	0.20163	0.21175	0.23145	40.752	42.953	45.286
r81	0.19737	0.19791	0.20163	0.21175	0.23145	48.909	52.039	55.022
r82	0.19737	0.19791	0.20163	0.21175	0.23145	66.71	69.193	72.375
r83	0.19737	0.19791	0.20163	0.21175	0.23145	116.82	122.65	128.66

14.2.2 Solve Problem in MIT Unemployed State with High and Low Unemployment Insurance

Shared xi value, 50 percent income loss. This is a one-time MIT shock that changes choices today in the sense of changing the resource state-space, but does not change forward expectations.

```
xi=0.5; % xi=0 full income loss from covid shock, xi=1, no covid income losses
```

Solve for $b = 0$, no unemployment insurane.

```
fl_b_lower = 0.0;
fl_b_higher= 0.5;
b=fl_b_lower; % b=0 means no UI benefits compensating COVID, b=1 if full income replacement
mp_params('xi') = xi;
mp_params('b') = b;
```

```
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
% mp_params('a2_covidyr') = mp_params('a2_covidyr_tax_fully_pay');
[V_unemp_b_0p0,~,cons_unemp_b_0p0,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std
	-	---	----	--------	----	---------	-----------	-------	-----
V_VFI	1	1	6	4.37e+07	83	5.265e+05	-6.8822e+08	-15.749	22.87
ap_VFI	2	2	6	4.37e+07	83	5.265e+05	1.3605e+09	31.134	36.29
cons_VFI	3	3	6	4.37e+07	83	5.265e+05	2.2887e+08	5.2375	8.443

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
	-------	-------	-------	-------	-------	---------	---------	--------
r1	-320.42	-318.92	-310.39	-296.97	-284.58	-4.4406	-4.3429	-4.246
r2	-310.88	-309.38	-300.85	-287.43	-275.14	-4.3331	-4.239	-4.146
r3	-301.33	-299.83	-291.3	-277.88	-265.85	-4.2231	-4.1327	-4.043
r4	-290.68	-289.29	-281.32	-268.6	-257.1	-4.1145	-4.0276	-3.941
r5	-281.05	-279.76	-272.29	-260.2	-249.16	-4.0121	-3.9284	-3.845
r79	-13.642	-13.628	-13.535	-13.298	-12.896	-0.22291	-0.21238	-0.2024
r80	-12.283	-12.269	-12.176	-11.939	-11.537	-0.17128	-0.16316	-0.1555
r81	-10.605	-10.591	-10.498	-10.261	-9.8589	-0.11815	-0.11254	-0.1072
r82	-8.3494	-8.3358	-8.2424	-8.0055	-7.6035	-0.065887	-0.062757	-0.05982
r83	-5.0665	-5.0529	-4.9595	-4.7226	-4.3206	-0.021146	-0.020134	-0.01918

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c526499	c526500
	--	--	--	--	---------	-------	-------	-------	-------	-------
r1	0	0	0	0	0.0083625	107.54	113.09	118.82	124.74	130.86
r2	0	0	0	0	0.0074731	107.45	112.99	118.72	124.64	130.75
r3	0	0	0	0	0.0058503	107.33	112.88	118.61	124.52	130.64
r4	0	0	0	0	0.0049981	107.54	113.08	118.81	124.73	130.85
r5	0	0	0	0	0.004174	107.76	113.3	119.03	124.95	131.07
r79	0	0	0	0	0	80.462	84.34	88.311	92.234	96.324
r80	0	0	0	0	0	75.113	78.736	82.42	85.975	90.439
r81	0	0	0	0	0	66.945	69.639	72.673	76.669	81.091
r82	0	0	0	0	0	50.126	53.467	55.311	56.953	60.587
r83	0	0	0	0	0	0	0	0	0	0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498
	--------	--------	--------	--------	--------	-------	-------	-------
r1	0.018623	0.019158	0.022901	0.033062	0.044486	11.989	12.265	12.55
r2	0.018623	0.019158	0.022901	0.033062	0.045375	12.223	12.501	12.787
r3	0.018623	0.019158	0.022901	0.033062	0.046998	12.476	12.755	13.042
r4	0.019354	0.019888	0.023632	0.033792	0.048579	12.72	13	13.289
r5	0.020066	0.020601	0.024344	0.034504	0.050114	12.955	13.236	13.525
r79	0.19737	0.19791	0.20163	0.21175	0.23145	35.417	37.362	39.409
r80	0.19737	0.19791	0.20163	0.21175	0.23145	40.752	42.953	45.286
r81	0.19737	0.19791	0.20163	0.21175	0.23145	48.909	52.039	55.022

```
    r82     0.19737     0.19791     0.20163     0.21175     0.23145    65.719     68.202     72.375
    r83     0.19737     0.19791     0.20163     0.21175     0.23145    115.84     121.66     127.68
```

Solve for $b = 0.5$, 50 percent unemployment insurance, meaning 50 percent of lost income is recovered.

```
b=fl_b_higher; % b=0 means no UI benefits compensating COVID, b=1 if full income replacement
mp_params('xi') = xi;
mp_params('b') = b;
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
% mp_params('a2_covidyr') = mp_params('a2_covidyr_tax_fully_pay');
[V_unemp_b_0p5,~,cons_unemp_b_0p5,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN           sum          mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -6.7567e+08    -15.462    22.25
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.3783e+09     31.541     36.3
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05     2.3114e+08     5.2893    8.440

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496      c526497      c526498
           -------    -------    -------    -------    -------    ---------    ---------    --------

    r1      -302.8    -302.11    -297.97     -290.4    -280.12      -4.3991      -4.3032      -4.208
    r2     -293.25    -292.57    -288.43    -280.86     -270.8      -4.2921      -4.1998      -4.108
    r3      -283.7    -283.02    -278.88    -271.34    -261.75      -4.1826       -4.094      -4.006
    r4     -273.72    -273.09    -269.23    -262.13     -253.1      -4.0721       -3.987      -3.902
    r5      -264.7    -264.11    -260.51    -253.79    -245.27      -3.9679      -3.8861      -3.805
    r79    -13.642    -13.628    -13.535    -13.298    -12.896     -0.22191     -0.21148     -0.2016
    r80    -12.283    -12.269    -12.176    -11.939    -11.537     -0.17053     -0.16249      -0.154
    r81    -10.605    -10.591    -10.498    -10.261    -9.8589     -0.11764     -0.11208     -0.1068
    r82    -8.3494    -8.3358    -8.2424    -8.0055    -7.6035    -0.065608    -0.062497    -0.05959
    r83    -5.0665    -5.0529    -4.9595    -4.7226    -4.3206    -0.021056    -0.020052    -0.01911

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
            c1    c2    c3        c4           c5        c526496    c526497    c526498    c526499    c
            --    --    --    ----------    --------    -------    -------    -------    -------    -

    r1      0     0     0      0.0011815    0.013905     109.98     115.52     121.26     127.18     1
    r2      0     0     0     0.00090277    0.013905     109.95     115.49     121.22     127.14     1
    r3      0     0     0     0.00051498    0.013905      109.9     115.45     121.18      127.1     1
    r4      0     0     0     0.00051498    0.013905     110.34     115.88     121.61     127.53     1
    r5      0     0     0     0.00048777    0.013905     110.79     116.33     122.06     127.98
    r79     0     0     0              0           0     80.974     84.852     88.823     92.746     9
    r80     0     0     0              0           0     75.619     79.241     82.926     86.481     9
    r81     0     0     0              0           0     67.445     70.139     73.173     76.669     8
    r82     0     0     0              0           0     50.126     53.467     55.806     57.389     6
    r83     0     0     0              0           0          0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5        c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.027723    0.028258    0.031999    0.040974    0.048028     11.989     12.265      12.55
```

r2	0.027723	0.028258	0.031999	0.041253	0.048028	12.223	12.501	12.787
r3	0.027723	0.028258	0.031999	0.041641	0.048028	12.476	12.755	13.042
r4	0.028805	0.029339	0.033081	0.042722	0.049108	12.72	13	13.289
r5	0.029859	0.030394	0.034135	0.043802	0.050161	12.955	13.236	13.525
r79	0.19737	0.19791	0.20163	0.21175	0.23145	35.417	37.362	39.409
r80	0.19737	0.19791	0.20163	0.21175	0.23145	40.752	42.953	45.286
r81	0.19737	0.19791	0.20163	0.21175	0.23145	48.909	52.039	55.022
r82	0.19737	0.19791	0.20163	0.21175	0.23145	66.215	68.697	72.375
r83	0.19737	0.19791	0.20163	0.21175	0.23145	116.33	122.15	128.17

14.2.3 Generate UI Comparison Matrixes

Find the deviation in value and consumption between higher and lower UI world. Welfare is converted to units in fixed life-time consumption.

```
gamma = mp_params('gamma');
mn_V_U_gain_moreUI = snw_hh_welfare(V_unemp_b_0p5, gamma) - snw_hh_welfare(V_unemp_b_0p0, gamma);
mn_C_U_gain_moreUI = cons_unemp_b_0p5 - cons_unemp_b_0p0;
```

14.2.4 Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

14.2.5 Analyze Difference in V and C with Higher and Lower UI

The difference between V and C with higher and ower UI.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(MN_V_GAIN(A,Z))

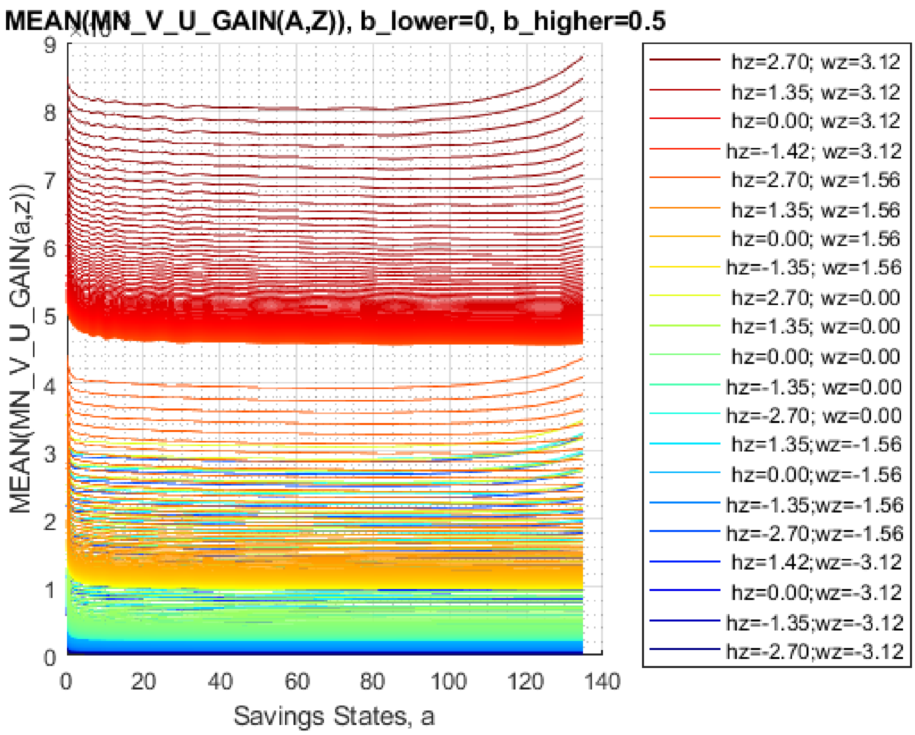
Tabulate value and policies along savings and shocks:

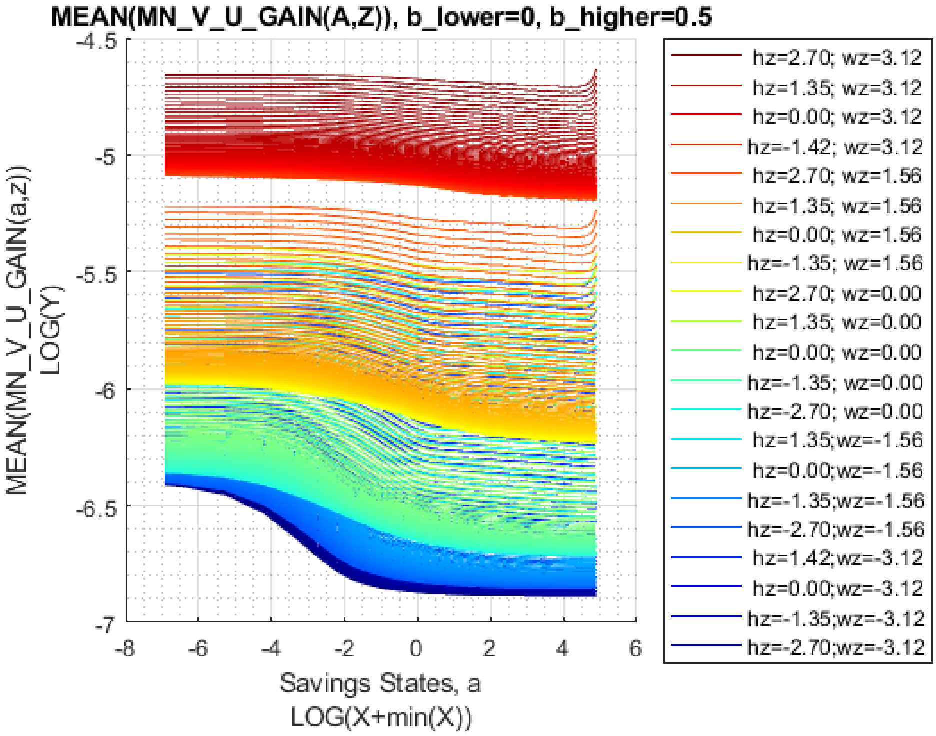
```
% Set
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN(A,Z)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_high
```

```
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_moreUI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar
```

```
xxx  MEAN(MN\_V\_U\_GAIN(A,Z)), b\_lower=0, b\_higher=0.5  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group      savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1               0    0.00064445    0.00064585    0.00064898     0.0006533    0.00065831    0.0
```

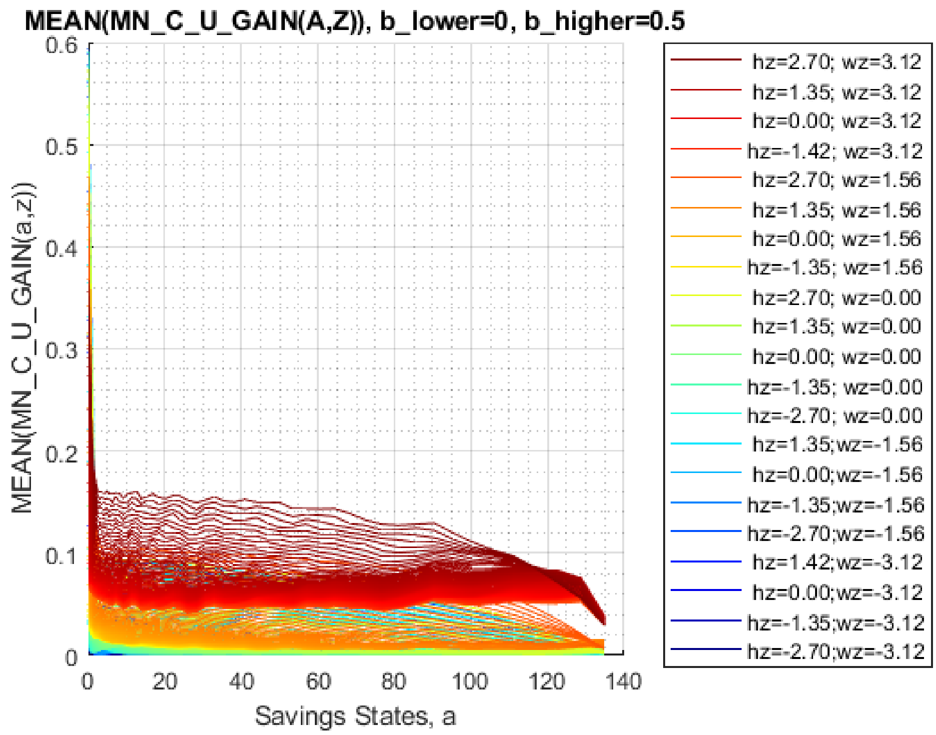
```
st_title = ['MEAN(MN\_V\_U\_GAIN(A,Z)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_h
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

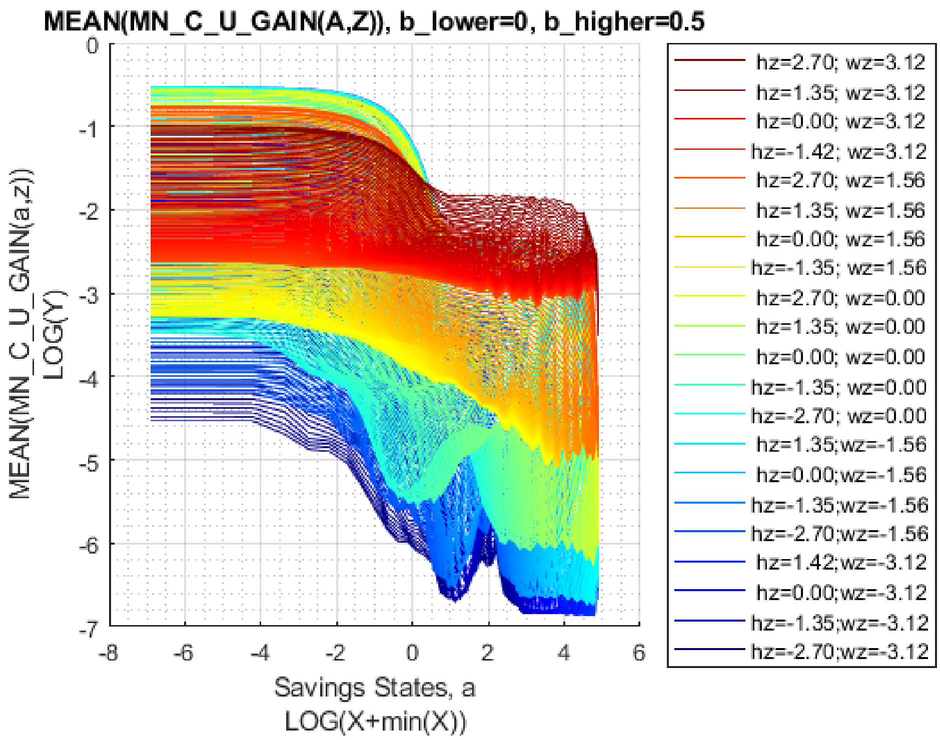




Graph Mean Consumption:

```
st_title = ['MEAN(MN\_C\_U\_GAIN(A,Z)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_h
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_C\_U\_GAIN(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





14.2.6 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN(KM,J)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_hig
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_moreUI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar

xxx  MEAN(MN_V_U_GAIN(KM,J)), b\_lower=0, b\_higher=0.5  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     kids     marry     mean_age_18     mean_age_19     mean_age_20     mean_age_21     mean_age_2
    -----     ----     -----     -----------     -----------     -----------     -----------     ----------

      1         1        0       0.00055672      0.00057552      0.00059522      0.00063544      0.00067622
```

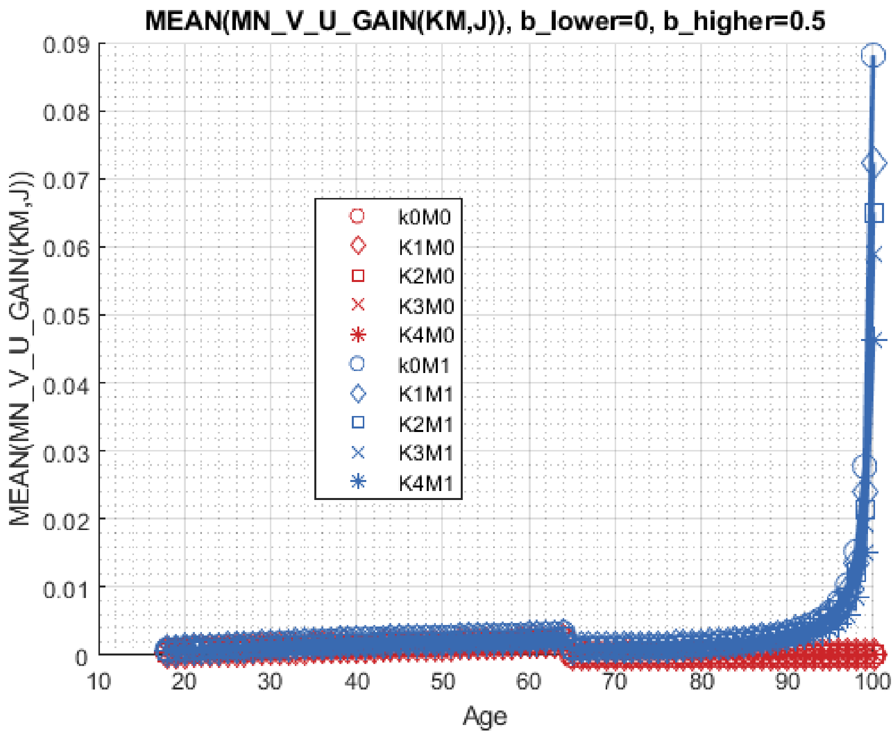
```
     2         2         0       0.00048244      0.00050035      0.00051966      0.00055508      0.00059141
     3         3         0       0.00046167      0.00047833      0.00049636      0.00052804      0.00056045
     4         4         0       0.00043771      0.00045259      0.00046877      0.00049707      0.00052603
     5         5         0       0.00042439      0.00043747      0.00045169       0.0004768      0.00050238
     6         1         1        0.0008562      0.00090669      0.00095849       0.0010295       0.0011022
     7         2         1       0.00071993      0.00075882      0.00079903      0.00085585      0.00091452
     8         3         1       0.00066704      0.00070046      0.00073484      0.00078384       0.0008343
     9         4         1       0.00061227      0.00064001      0.00066838      0.00070987      0.00075249
    10         5         1       0.00055284      0.00057485        0.000597      0.00063042      0.00066449
```

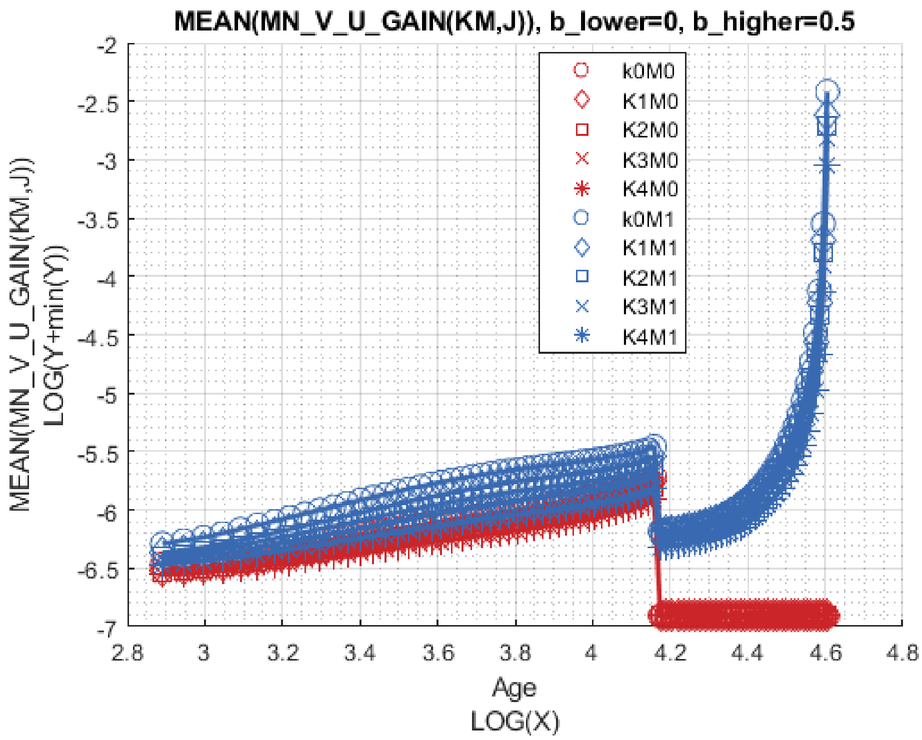
```
% Consumption Function
st_title = ['MEAN(MN_C_U_GAIN(KM,J)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_hig
tb_az_c = ff_summ_nd_array(st_title, mn_C_U_gain_moreUI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar
```

```
xxx  MEAN(MN_C_U_GAIN(KM,J)), b\_lower=0, b\_higher=0.5  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     kids     marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____     ____     _____    ___________    ___________    ___________    ___________    __________

      1         1        0       0.045263       0.048116       0.050981       0.053114       0.054711
      2         2        0       0.052245       0.055222       0.058311       0.061472       0.064326
      3         3        0        0.05831       0.061181       0.064294        0.06802       0.071488
      4         4        0       0.061725       0.064563       0.067634        0.07166       0.075434
      5         5        0       0.065016       0.067757       0.070896        0.07514       0.079088
      6         1        1        0.07018       0.073094       0.076227       0.079837       0.083882
      7         2        1       0.075602       0.078727       0.081843       0.085263       0.088392
      8         3        1       0.079512       0.083393       0.087172       0.091625       0.095741
      9         4        1       0.078346       0.082109       0.086013       0.091418       0.096514
     10         5        1       0.076598       0.080189       0.084089       0.089192        0.09394
```

Graph Mean Values:

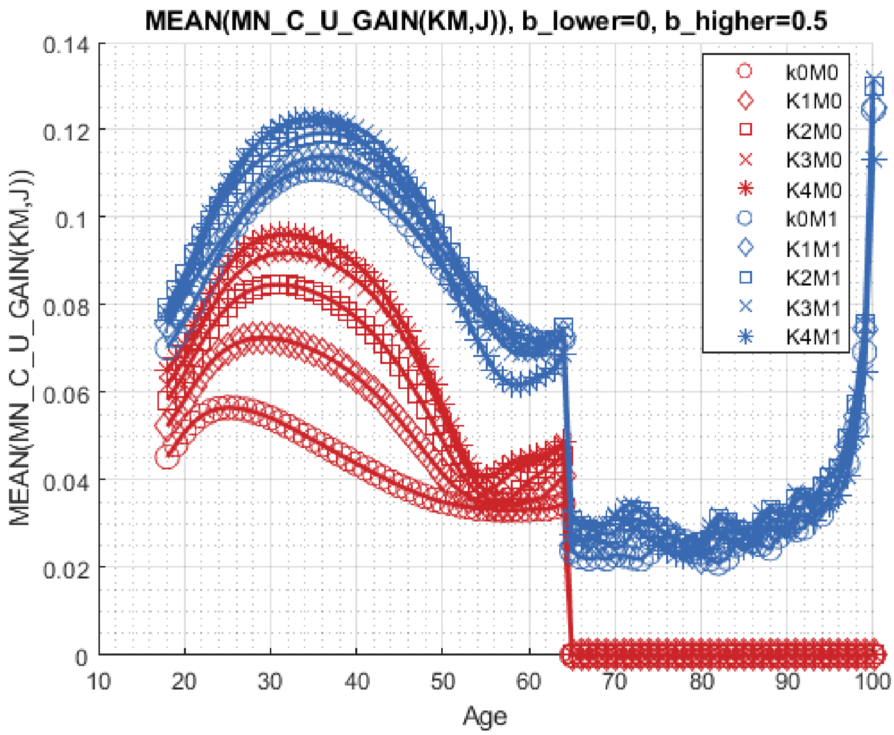
```
st_title = ['MEAN(MN\_V\_U\_GAIN(KM,J)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

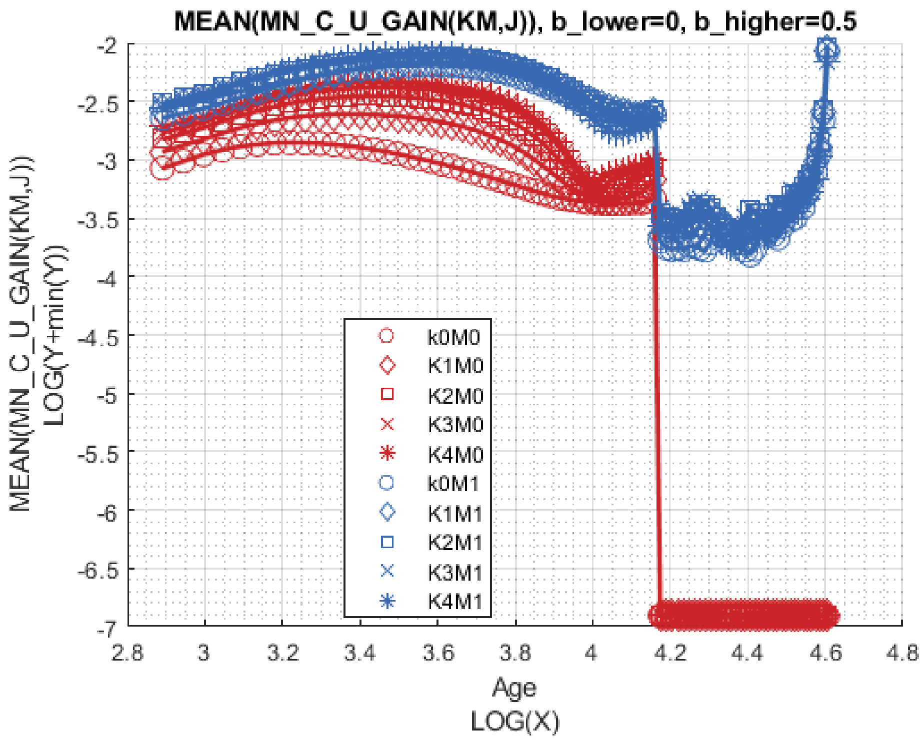




Graph Mean Consumption:

```
st_title = ['MEAN(MN\_C\_U\_GAIN(KM,J)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_C\_U\_GAIN(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





14.2.7 Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

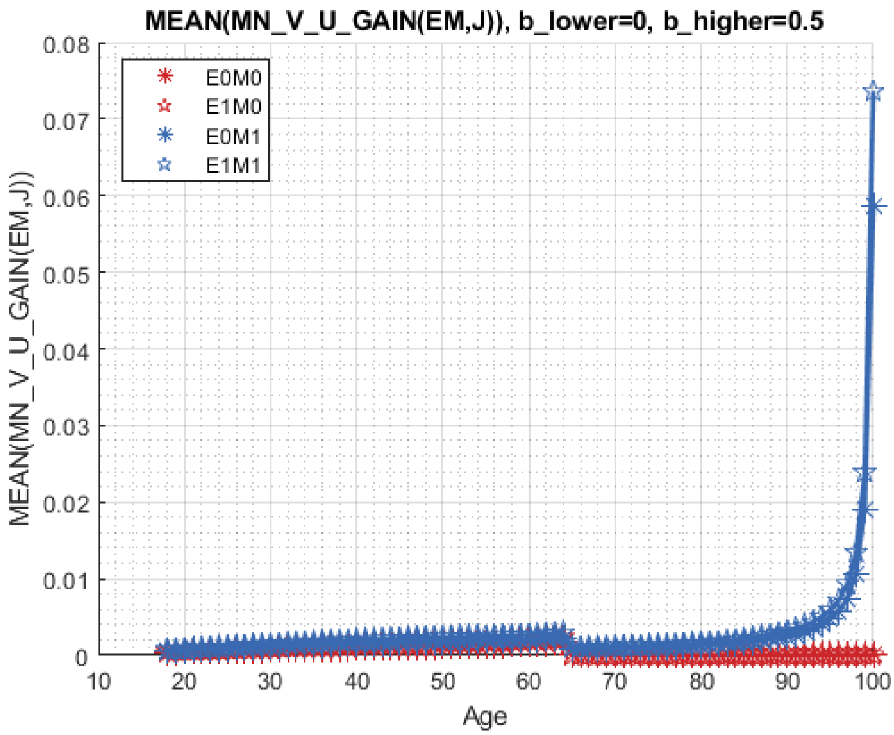
Tabulate value and policies:

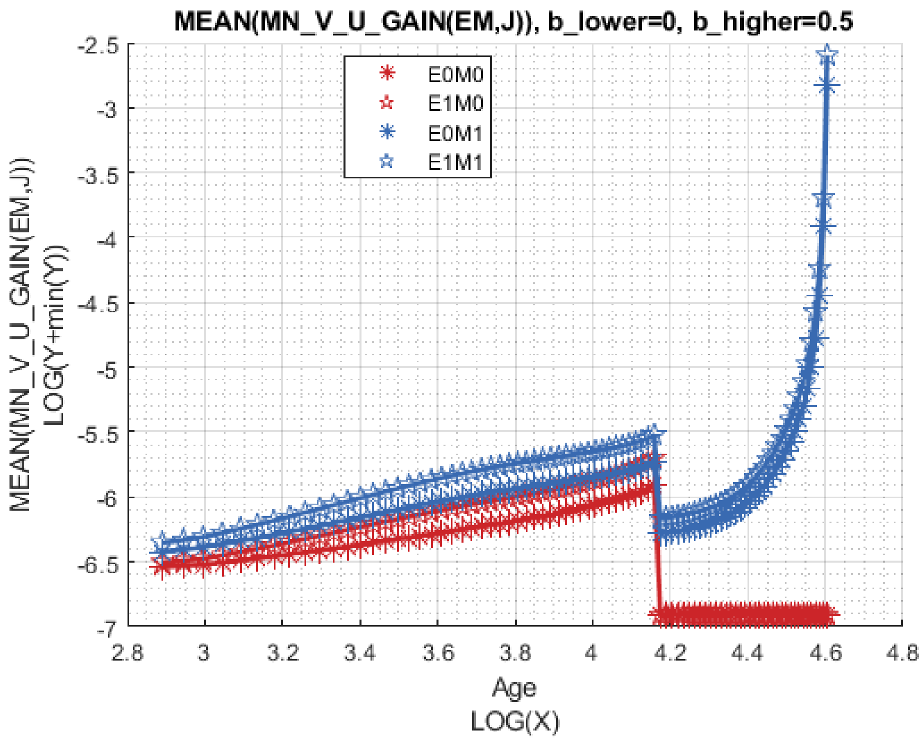
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN(EM,J)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_hig
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_moreUI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar

xxx  MEAN(MN_V_U_GAIN(EM,J)), b\_lower=0, b\_higher=0.5  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu     marry     mean_age_18     mean_age_19     mean_age_20     mean_age_21     mean_age_22
    _____    ___     _____     ___________     ___________     ___________     ___________     ___________

      1       0        0       0.00044711      0.00045939      0.00047231      0.00049626      0.00052045
      2       1        0       0.00049806      0.00051832      0.00054037      0.00058072      0.00062215
      3       0        1       0.00062246      0.00065265      0.00068378      0.00072512      0.00076731
      4       1        1       0.00074086      0.00077968      0.00081932      0.00087866       0.0009399

% Consumption
st_title = ['MEAN(MN_C_U_GAIN(EM,J)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_hig
```

```
tb_az_c = ff_summ_nd_array(st_title, mn_C_U_gain_moreUI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar

xxx  MEAN(MN_C_U_GAIN(EM,J)), b\_lower=0, b\_higher=0.5  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     edu     marry     mean_age_18     mean_age_19     mean_age_20     mean_age_21     mean_age_22
    -----     ---     -----     -----------     -----------     -----------     -----------     -----------

      1        0        0         0.047852        0.050213        0.052704        0.054124         0.05542
      2        1        0         0.065172        0.068522        0.072143        0.077638        0.082599
      3        0        1          0.06735        0.070027        0.072677        0.075227        0.077714
      4        1        1         0.084746        0.088978        0.093461        0.099707         0.10567
```

Graph Mean Values:

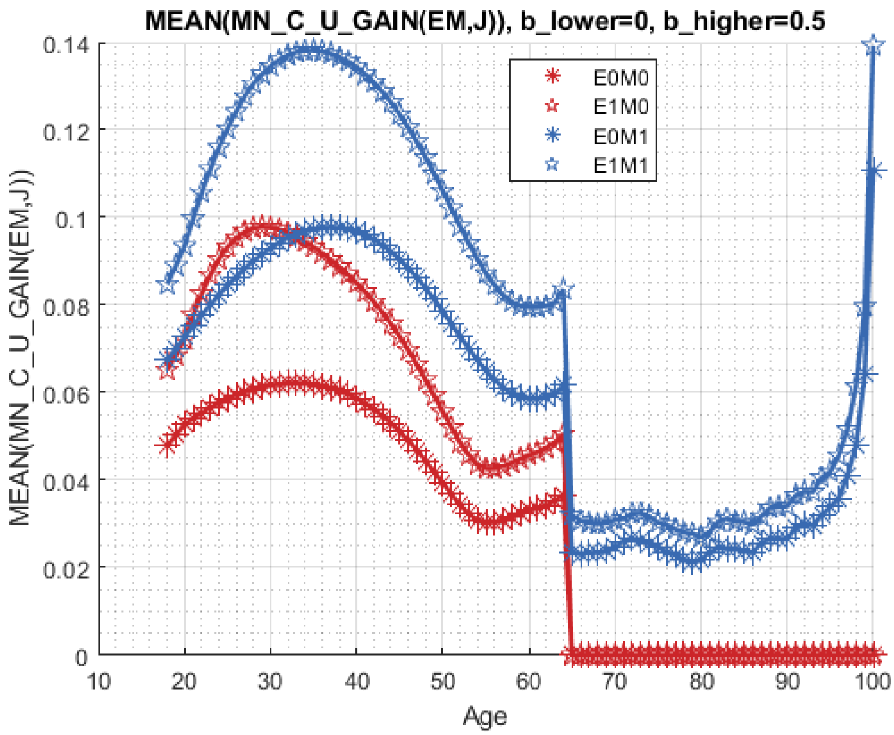
```
st_title = ['MEAN(MN\_V\_U\_GAIN(EM,J)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

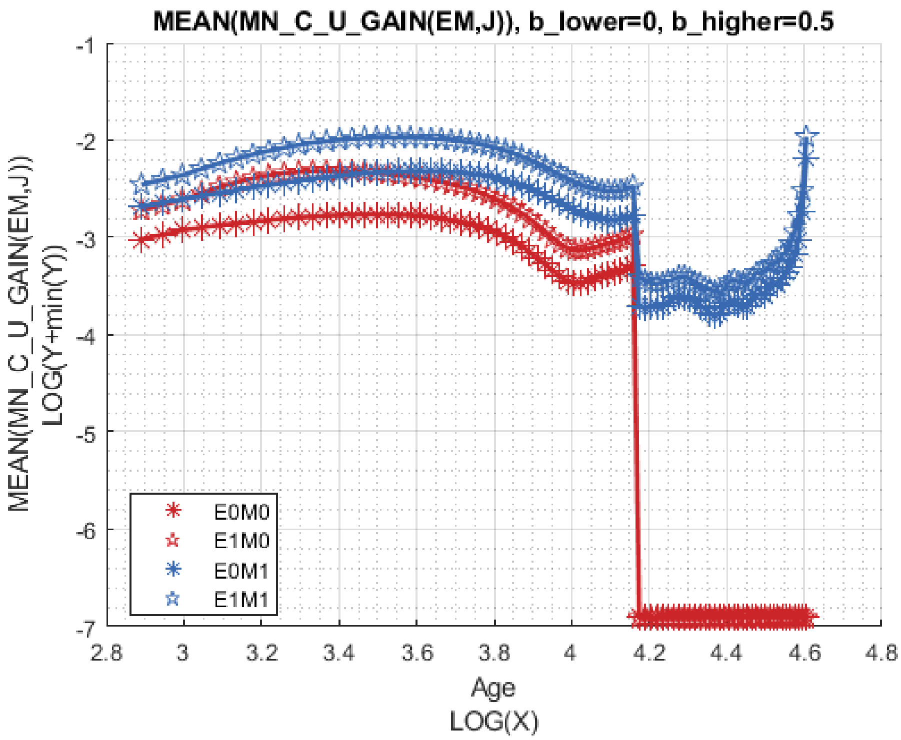




Graph Mean Consumption:

```
st_title = ['MEAN(MN\_C\_U\_GAIN(EM,J)), b\_lower=' num2str(fl_b_lower) ', b\_higher=' num2str(fl_b_
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_C\_U\_GAIN(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





14.3 Value and Consumption Low vs Higher Discount Factor Comparison

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for the V(states) for individuals at lower and higher discount factor β. We allow for β heterogeneity in the model to consider both patient and impatient households. The key difference is that patient households are more willing to save and will consume less.

14.3.1 Solve Model at $\beta = 0.95$

Our high type households have $\beta = 0.95$.

```
% mp_params = snw_mp_param('default_dense');
mp_params = snw_mp_param('default_docdense');
fl_higher_beta = 0.95;
fl_lower_beta = 0.60;
mp_params('beta') = fl_higher_beta;
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_timer') = true;
[V_ss_beta95,~,cons_ss_beta95,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=490.
```

14.3.2 Solve Model at $\beta = 0.60$

Our high type households have $\beta = 0.60$.

```
mp_params('beta') = fl_lower_beta;
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_timer') = true;
```

```
[V_ss_beta60,~,cons_ss_beta60,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=487.
```

14.3.3 Generate β Comparison Matrixes

Take the difference between $\beta = 0.95$ percent and $\beta = 0.60$ consumption and value n-dimensional matrixes. Welfare is converted to units in fixed life-time consumption. Note that for example: $\log(0.5) + 0 \cdot \log(0.5) > \log(0.5) + 0.99 \cdot \log(0.5)$, in another word, V is higher with lower β if utility per-period is a negative value. Note our $\gamma = 2$ for the CRRA parameter. We can compare V relatively across choices for the same individual, but less meaningfully across individuals with varying preferences.

```
gamma = mp_params('gamma');
mn_V_gain_beta = snw_hh_welfare(V_ss_beta95, gamma) - snw_hh_welfare(V_ss_beta60, gamma);
mn_C_gain_beta = cons_ss_beta95 - cons_ss_beta60;
fl_beta_gap = fl_higher_beta - fl_lower_beta;
```

14.3.4 Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

14.3.5 Analyze Difference in V and C with Higher and Lower β

The difference between V and C with higher and lower β.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

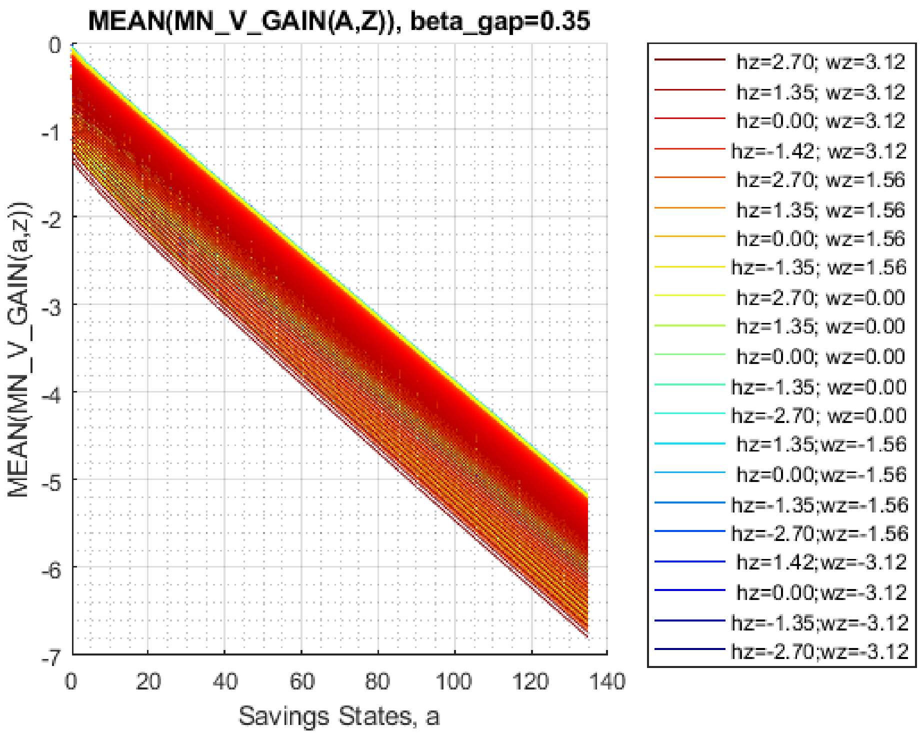
MEAN(MN_V_GAIN(A,Z))

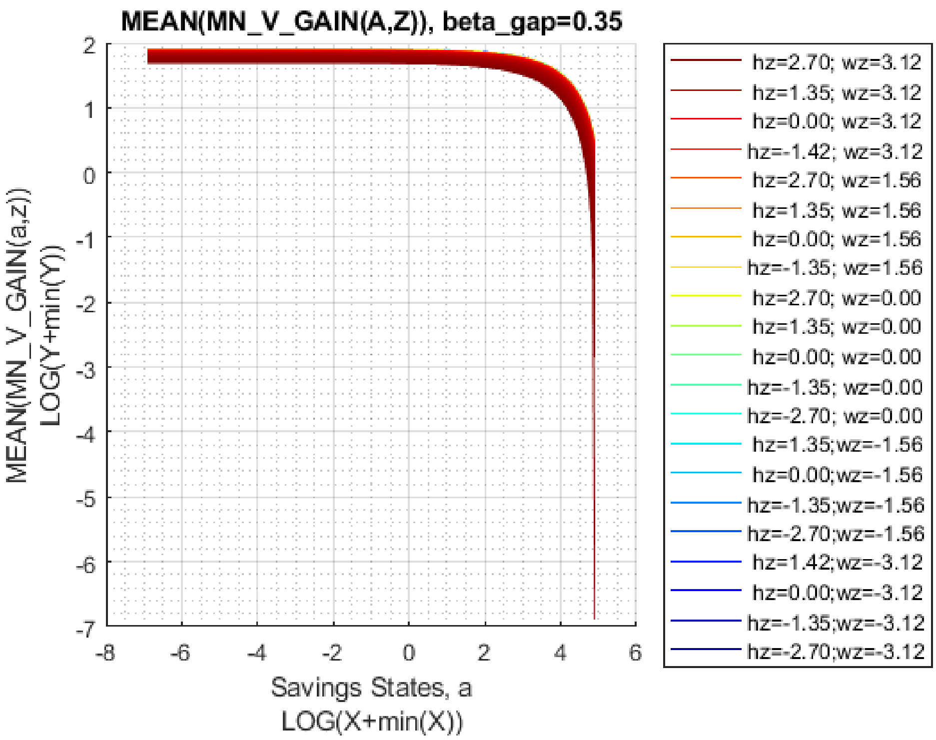
Tabulate value and policies along savings and shocks:

```
% Set
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_Gain(A,Z)), beta_gap=' num2str(fl_beta_gap) ];
tb_az_v = ff_summ_nd_array(st_title, mn_V_gain_beta, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per
```

```
xxx  MEAN(MN_V_Gain(A,Z)), beta_gap=0.35  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group      savings       mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5     mea
    -----     ----------     ----------     ----------     ----------     ----------     ----------     ---

      1                0     -0.024097      -0.024617      -0.025187      -0.025805      -0.026469      -0.
```

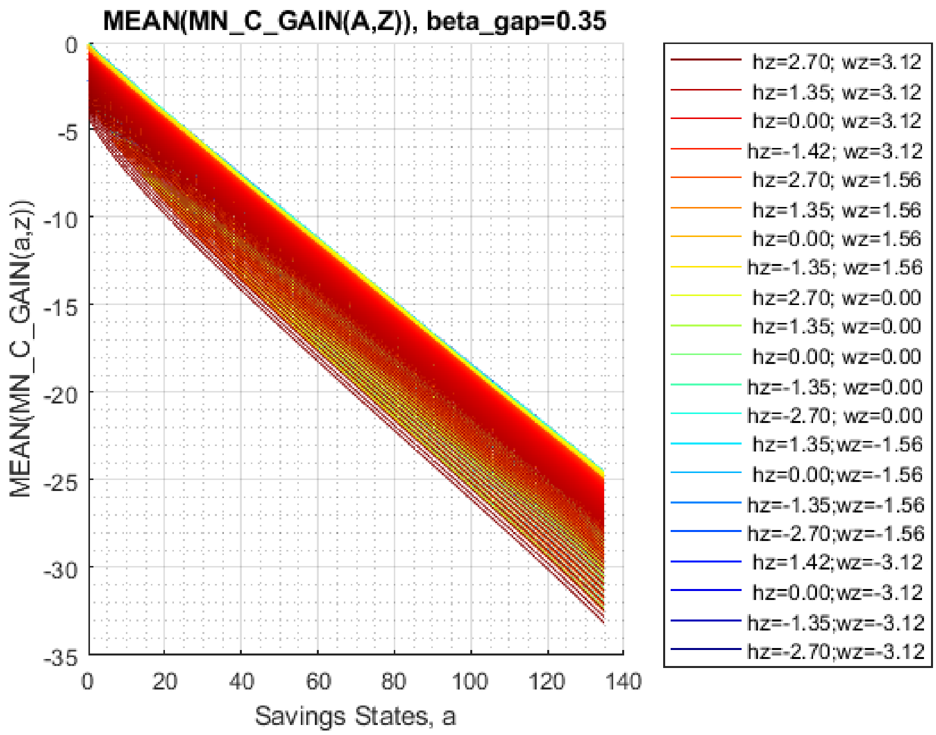
```
st_title = ['MEAN(MN\_V\_GAIN(A,Z)), beta\_gap=' num2str(fl_beta_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_GAIN(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

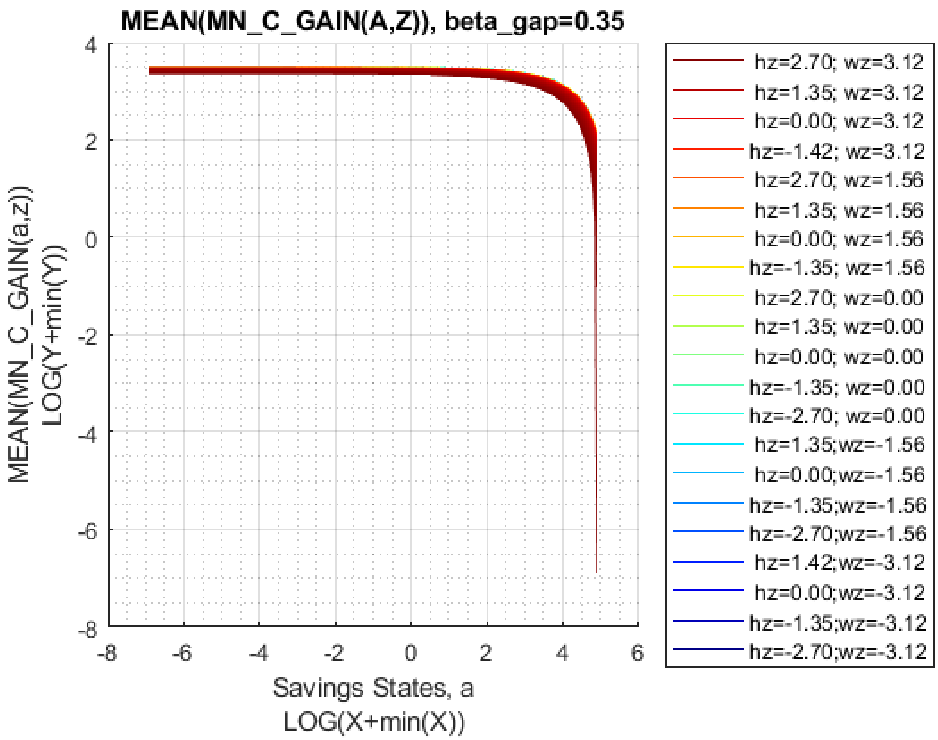




Graph Mean Consumption:

```
st_title = ['MEAN(MN\_C\_GAIN(A,Z)), beta\_gap=' num2str(fl_beta_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_C\_GAIN(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





14.3.6 Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd' ,'s', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
st_title = ['MEAN(MN_V_Gain(KM,J)), beta_gap=' num2str(fl_beta_gap) ];
tb_az_v = ff_summ_nd_array(st_title, mn_V_gain_beta, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(MN_V_Gain(KM,J)), beta_gap=0.35  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     kids     marry     mean_age_18     mean_age_19     mean_age_20     mean_age_21     mean_age_2
    _____     ____     _____     ___________     ___________     ___________     ___________     __________

      1         1        0         -2.3297         -2.3646         -2.4033         -2.4467         -2.4886
```

2	2	0	-1.6829	-1.7078	-1.7361	-1.7685	-1.8005
3	3	0	-1.4479	-1.4653	-1.4856	-1.5093	-1.5326
4	4	0	-1.2769	-1.2904	-1.3063	-1.3254	-1.3441
5	5	0	-1.1689	-1.1787	-1.1908	-1.2057	-1.2203
6	1	1	-1.8594	-1.8945	-1.9328	-1.9753	-2.0168
7	2	1	-1.4942	-1.5203	-1.5493	-1.5821	-1.6145
8	3	1	-1.3397	-1.3604	-1.3836	-1.4101	-1.4362
9	4	1	-1.2093	-1.2257	-1.2444	-1.2661	-1.2875
10	5	1	-1.1227	-1.1351	-1.1495	-1.1667	-1.1835

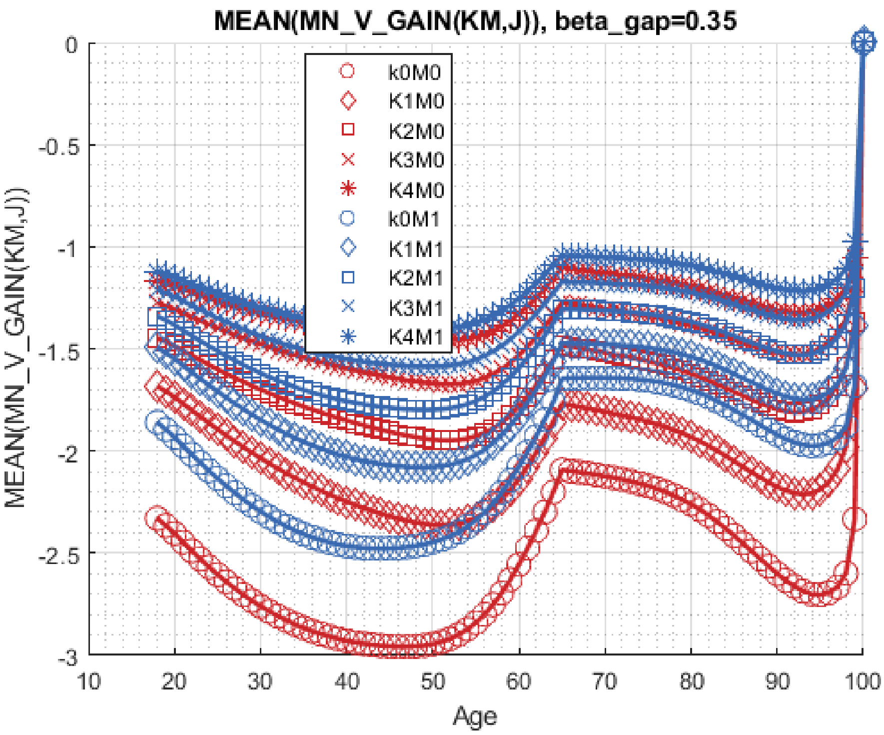
```
% Consumption Function
st_title = ['MEAN(MN_C_Gain(KM,J)), beta_gap=' num2str(fl_beta_gap) ];
tb_az_c = ff_summ_nd_array(st_title, mn_C_gain_beta, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per
```

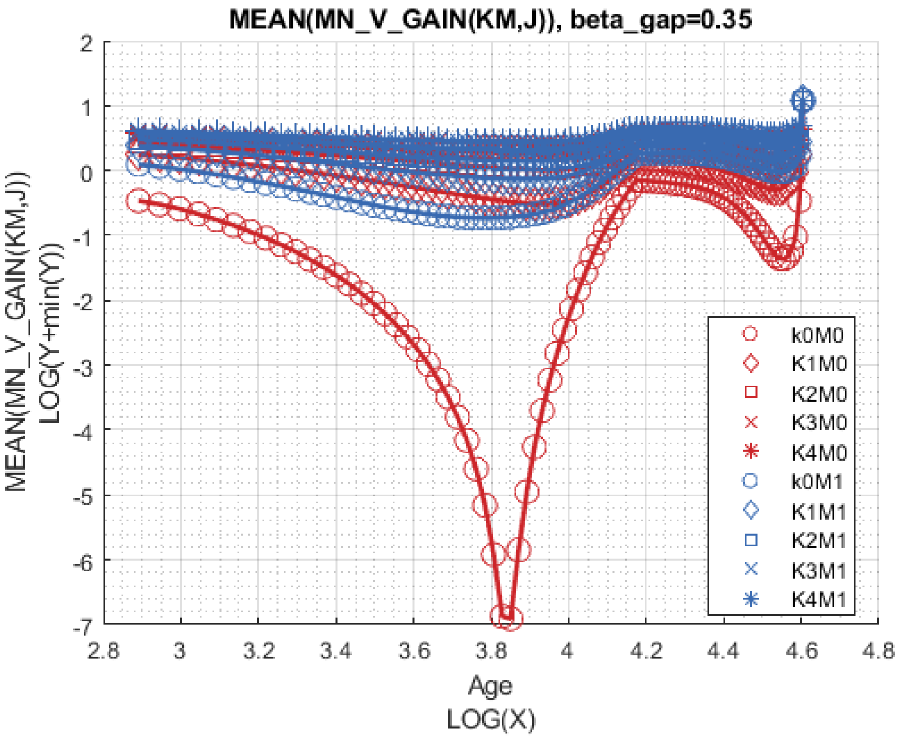
xxx MEAN(MN_C_Gain(KM,J)), beta_gap=0.35 xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	kids	marry	mean_age_18	mean_age_19	mean_age_20	mean_age_21	mean_age_2
1	1	0	-7.8284	-7.878	-7.942	-8.0313	-8.1185
2	2	0	-7.7474	-7.7919	-7.8514	-7.9374	-8.0227
3	3	0	-7.8502	-7.8841	-7.9298	-8.0042	-8.0781
4	4	0	-7.8498	-7.8791	-7.9197	-7.989	-8.0577
5	5	0	-7.882	-7.9053	-7.9401	-8.0028	-8.0648
6	1	1	-8.3054	-8.3892	-8.4885	-8.6164	-8.7429
7	2	1	-8.1032	-8.1784	-8.2681	-8.3867	-8.5041
8	3	1	-8.1357	-8.2006	-8.2817	-8.3905	-8.4971
9	4	1	-8.0781	-8.1343	-8.207	-8.3071	-8.4058
10	5	1	-8.051	-8.0944	-8.154	-8.2396	-8.3246

Graph Mean Values:

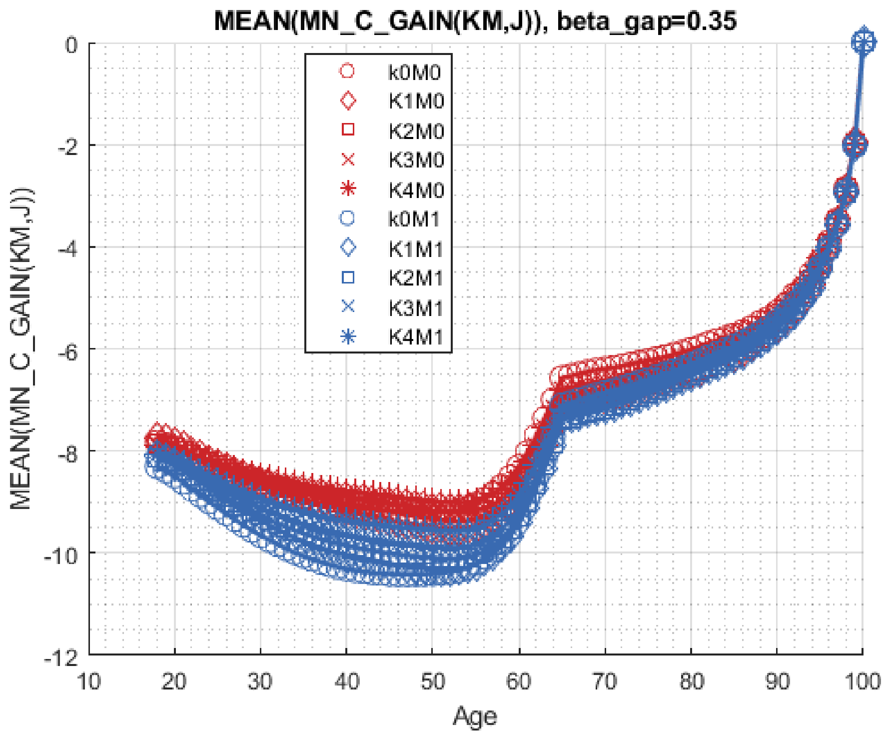
```
st_title = ['MEAN(MN\_V\_GAIN(KM,J)), beta\_gap=' num2str(fl_beta_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_GAIN(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

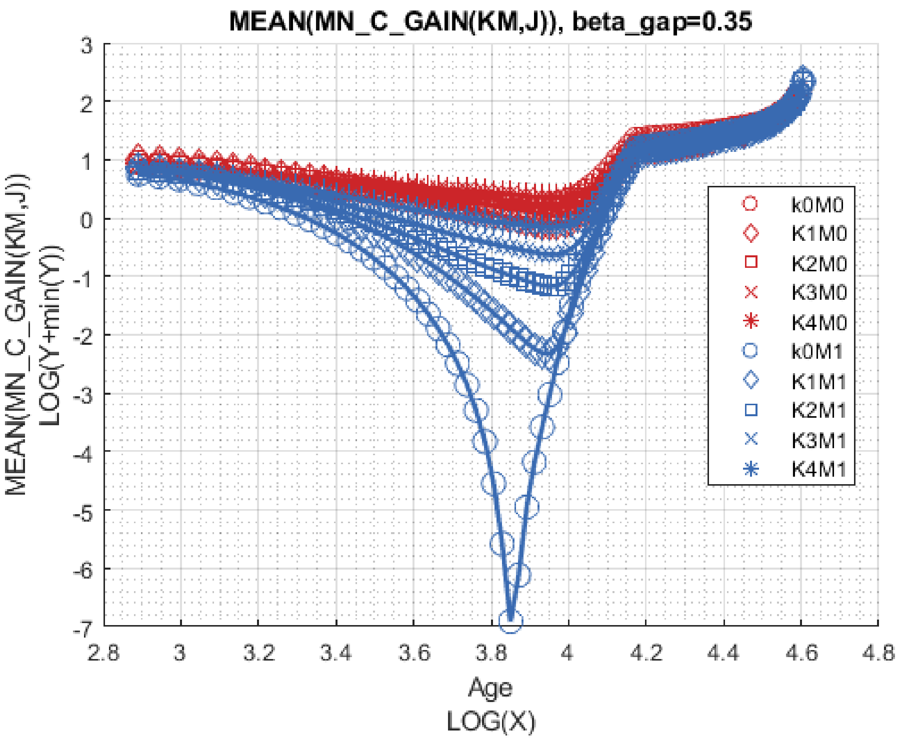




Graph Mean Consumption:

```
st_title = ['MEAN(MN\_C\_GAIN(KM,J)), beta\_gap=' num2str(fl_beta_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_C\_GAIN(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





14.3.7 Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

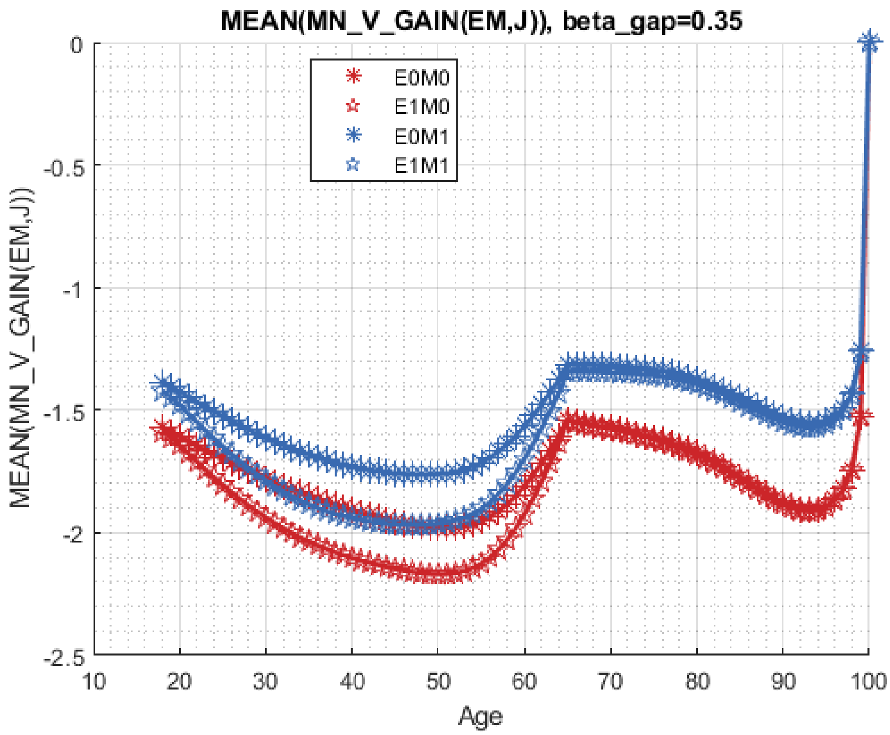
Tabulate value and policies:

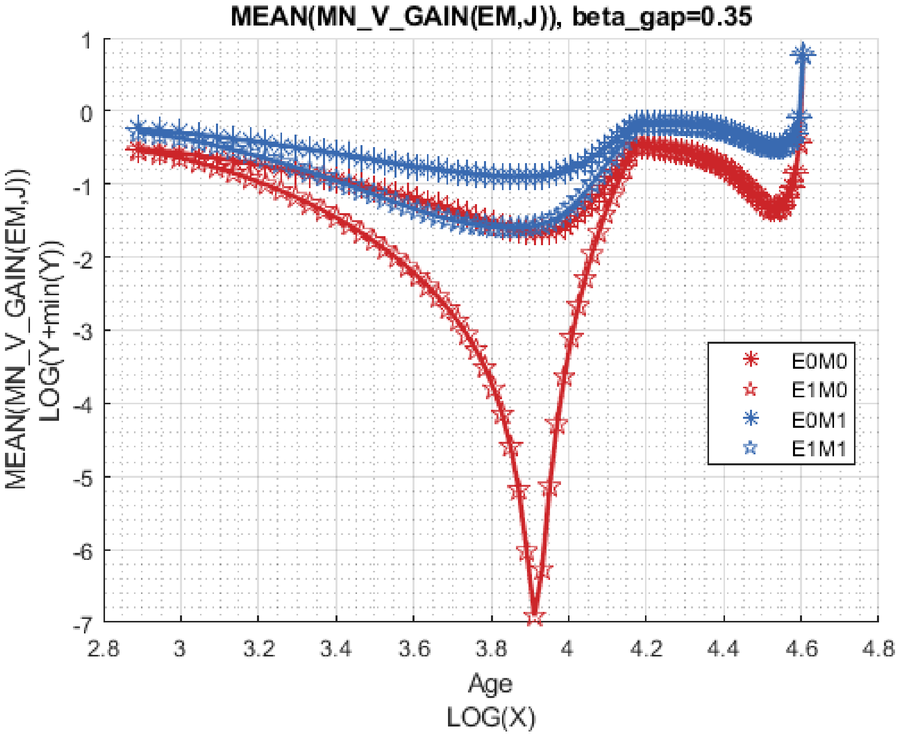
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_Gain(EM,J)), beta_gap=' num2str(fl_beta_gap) ];
tb_az_v = ff_summ_nd_array(st_title, mn_V_gain_beta, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(MN_V_Gain(EM,J)), beta_gap=0.35  xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        -1.5742        -1.5897        -1.6068        -1.6259        -1.6446
      2       1       0        -1.5882         -1.613         -1.642        -1.6763        -1.7098
      3       0       1        -1.3891        -1.4066        -1.4256        -1.4464        -1.4669
      4       1       1         -1.421        -1.4478        -1.4783        -1.5137        -1.5485

% Consumption
st_title = ['MEAN(MN_C_Gain(EM,J)), beta_gap=' num2str(fl_beta_gap) ];
```

```
tb_az_c = ff_summ_nd_array(st_title, mn_C_gain_beta, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(MN_C_Gain(EM,J)), beta_gap=0.35  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0        -7.8968        -7.9208        -7.9537        -8.0028        -8.0505
      2       1       0        -7.7663        -7.8145        -7.8794         -7.983        -8.0863
      3       0       1        -8.1784        -8.2294        -8.2887        -8.3652        -8.4396
      4       1       1         -8.091        -8.1694        -8.2711        -8.4109        -8.5502
```

Graph Mean Values:

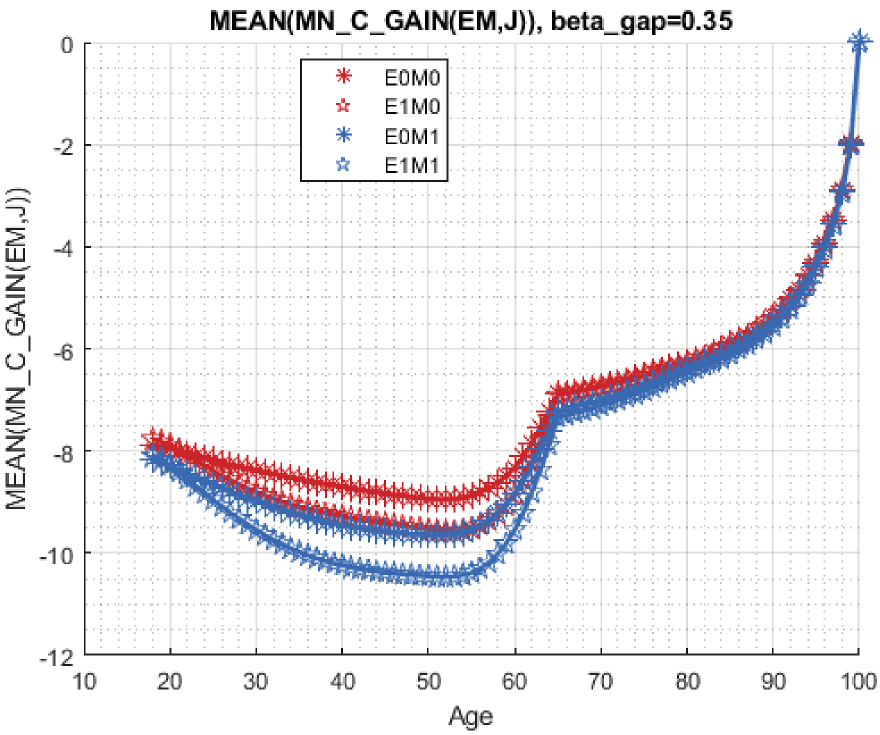
```
st_title = ['MEAN(MN\_V\_GAIN(EM,J)), beta\_gap=' num2str(fl_beta_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_GAIN(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

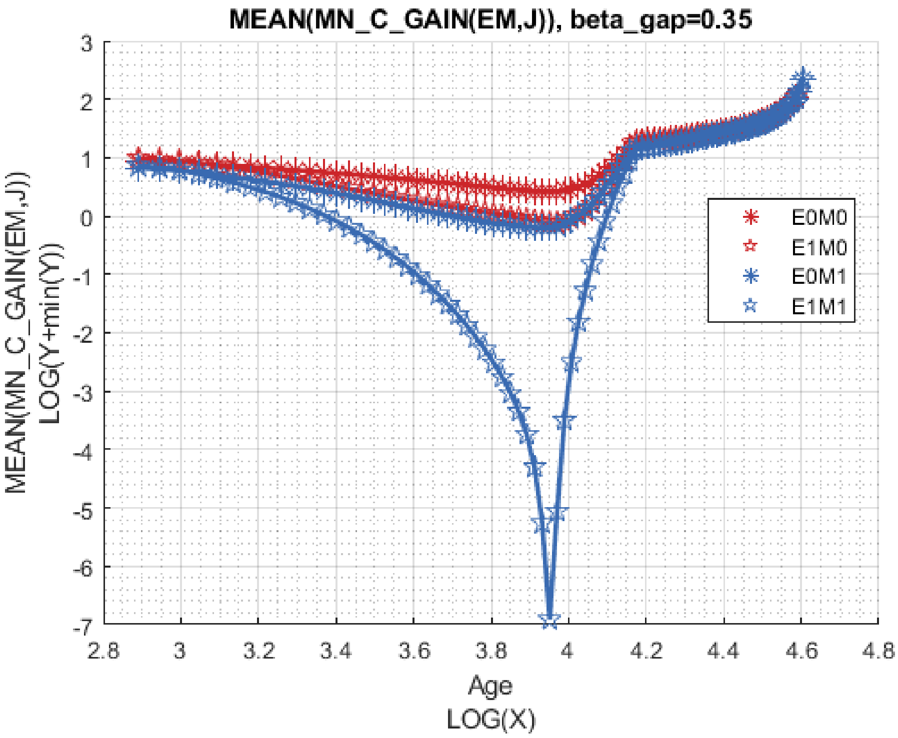




Graph Mean Consumption:

```
st_title = ['MEAN(MN\_C\_GAIN(EM,J)), beta\_gap=' num2str(fl_beta_gap)  ''];
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_C\_GAIN(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```




MEAN(MN_C_GAIN(EM,J)), beta_gap=0.35
MEAN(MN_C_GAIN(EM,J))
LOG(Y+min(Y))
Age
LOG(X)
E0M0
E1M0
E0M1
E1M1
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
4.8

Appendix A

Index and Code Links

A.1 ## Selected Comparative Statistics links links

Bibliography

The MathWorks Inc (2019). *MATLAB*. Matlab package version 2019b.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.18.