

# R, RTools, Rstudio Installation and Update with VSCode

Fan Wang

2022-07-09

## Contents

<b>1 R Installation and Set-Up</b>	<b>1</b>
1.1 Uninstall r . . . . .	1
1.2 Install R . . . . .	2
1.2.1 Install R for the First Time . . . . .	2
1.2.2 Linux R Install . . . . .	2
1.2.3 Update R on Windows . . . . .	3
1.2.4 R Add to Path . . . . .	3
1.3 R Package Installations . . . . .	4
1.3.1 Update R Install Directory . . . . .	4
1.3.2 Install vearious directory . . . . .	4
1.4 R Tests . . . . .	5
1.5 R with Radian . . . . .	5
1.6 Running R Inside VSCode . . . . .	5
1.6.1 RMD in VSCode . . . . .	5
1.6.2 RMD and MARKDOWN File Associations . . . . .	6

## 1 R Installation and Set-Up

Install R new, or update an existing R installation.

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools](#) research support package, [R4Econ](#) examples page, [PkgTestR](#) packaging guide, or [Stat4Econ](#) course page.

### 1.1 Uninstall r

Uninstall R, RStudio and RTools from Windows “Programs and Features” menu. After uninstaller finishes, check in the libPath folders to see if there are still stuff there, delete all, delete the folders.

If R was installed in a virtual environment, delete the environment. Otherwise, use system’s uninstaller. To check on this, from terminal/command-prompt for each virtual environment, type *R*, and *.libPaths()* to find paths. Do so inside conda main, outside of conda main, inside different environments, find all paths.

Inside R:

```
ls_spn_paths <- .libPaths()
print(ls_spn_paths)

## [1] "C:/Users/fan/AppData/Local/R/win-library/4.2"
## [2] "C:/Program Files/R/R-4.2.1/library"

# C:/Users/fan/Documents/R/win-library/3.6
# C:/Program Files/R/R-3.6.1/library
```

Check which conda env is installed, if there is an env installed for R.

```
# Show All installed environments
conda info --envs
```

For Linux and for unsintalling inside conda:

```
# Exit Conda
conda deactivate
# where is R installed outside of Conda
which R
# /usr/bin/R
# To remove all
sudo apt-get remove r-base
sudo apt-get remove r-base-core

# Inside Conda base
conda activate
# Conda r_env
conda activate r_envr
# Where is it installed?
which R
# /home/wangfanbng75/anaconda3/bin/R
conda uninstall r-base
```

## 1.2 Install R

### 1.2.1 Install R for the First Time

1. [download R](#)
  - for debian: [Johannes Ranke](#). For Linux/Debian installation, crucial to update the *source.list* to include sources that have more recent versions of R. If not, will get very old R versions that is not compatible with many packages.
  - add R to path for Windows. In Windows Path, add for example: *C:/Program Files/R/R-3.6.2/bin/x64/* and *C:/Rtools/bin*
2. [Install Rtools](#) for building R packages.
3. [download R-studio](#)
4. Open R-studio and auto-detect R
5. Install additional packages

### 1.2.2 Linux R Install

For linux/Debian, to Install latest R:

```
# Go to get latest debian latest r sources.list
cat /etc/apt/sources.list
# Install this First (should already be installed)
sudo apt install dirmngr

# Debian R is maintained by Johannes Ranke, copied from https://cran.r-project.org/bin/linux/debian/:
apt-key adv --keyserver keys.gnupg.net --recv-key 'E19F5F87128899B192B1A2C2AD5F960A256A04AF'
# Add to source.list, for debian stretch (9)
# sudo su added for security issue as super-user
sudo su -c "sudo echo 'deb http://cloud.r-project.org/bin/linux/debian stretch-cran35/' >> /etc/apt/sources.list"
# if added wrong lines, delete 3rd line
sudo sed '3d' /etc/apt/sources.list
```

```
# Update and Install R, should say updated from cloud.r
sudo apt-get update
sudo apt-get install r-base r-base-dev

# Also install these, otherwise r-packages do not install
# libxml2 seems need for tidymodels
sudo apt-get install libcurl4-openssl-dev
sudo apt-get install libssl-dev
sudo apt-get install libxml2-dev
```

### 1.2.3 Update R on Windows

First, use the `updateR()` function from the `installr` package.

1. On windows, install the `installr` package, and use `updateR()`
2. At the end, will ask if want to move all old packages to new R directory

New R will have new package directory, could keep all in old, should copy all to new, and not keep old. Can choose to copy all old packages to new folder, but still keep old packages in prior folder as they were

```
# https://www.r-project.org/nosvn/pandoc/installr.html
install.packages('installr')
# update R from inside R (not Rstudio)
require(installr)
# this will open dialog boxes to take you through the steps.
updateR()
# Set Rstudio to the Latest R
```

Second, after updating, might go into “Apps and Features” on Windows to unstaill the previous R version.

Third, update `RTools`. Uninstall `RTools`. Installation can be very large.

Fourth, update `RStudio`. Upon opening `RStudio`, if prior installations of R have been uninstalled, `RStudio` will auto-detect as ask if the latest version of R should be used. Choose yes. Upon entering `RStudio`, if there are updates, might prompt to `RStudio` website to download and update.

Fifth, follow the package installation directions below to update that.

### 1.2.4 R Add to Path

To be able to use R via command line, make sure Windows knows where the path to `R.exe` is.

First, find where the installed `R.exe` path is. Open up the R installation, and then check path as below.

```
ls_spn_paths <- .libPaths()
print(ls_spn_paths)
# "C:/Users/fan/AppData/Local/R/win-library/4.2"
# "C:/Program Files/R/R-4.2.1/library"
```

Second, given the path found, the `R.exe` is at “C:/Program Files/R/R-4.2.1/bin”. So now, in windows, System Properties -> Advanced -> Environment Variables -> System Variables -> Path -> Edit -> New -> Paste “C:/Program Files/R/R-4.2.1/bin”

Third, now open up command-prompt/terminal/git-bash, enter R, this will take us into the R console via command-line.

```
# To exit command line:
q()
```

## 1.3 R Package Installations

### 1.3.1 Update R Install Directory

After installing R, change the path sequence so that packages install for all users.

```
ls_spn_paths <- .libPaths()
print(ls_spn_paths)
# [1] "C:/Users/fan/AppData/Local/R/win-library/4.2" "C:/Program Files/R/R-4.2.1/library"
ls_spn_paths <- c(ls_spn_paths[2], ls_spn_paths[1])
.libPaths(ls_spn_paths)
ls_spn_paths <- .libPaths()
print(ls_spn_paths)
# [1] "C:/Program Files/R/R-4.2.1/library" "C:/Users/fan/AppData/Local/R/win-library/4.2"
```

### 1.3.2 Install various directory

After updating R, sometimes, old packages are not copied over to new directory, so need to reinstall all packages.

Having set the directories earlier so that packages do not install in user's personal folder, but the library folder where the R version is installed, we can find all installed packages in the *C:/Program Files/R/R-4.2.1/library* folder.

```
# Install RTools First!
# https://cran.r-project.org/bin/windows/Rtools/

# Install system tools
install.packages(c("backports"))

# Install tidyverse
install.packages(c("tidyverse", "tidymodels", "vroom"))

# Install Packaging tools
install.packages(c("devtools", "pkgdown", "roxygen2", "bookdown", "knitr", "kableExtra", "formatR", "reprex"))

# Install Statistics models
install.packages(c("AER", "minpack.lm"))
install.packages(c("quantreg"))

# Install Tools to Work with Other Packages
# matconv: converts matlab programs to R
install.packages(c("reticulate", "JuliaCall", "matconv"))
install.packages(c("matconv"))
# for reticulate errors, install directly from: devtools::install_github("rstudio/reticulate")

# Install Paralell Tools
install.packages(c("parallel", "doParallel", "foreach"))

# Install personal Packages
devtools::install_github("fanwangecon/REconTools")
devtools::install_github("fanwangecon/PrjOptiAlloc")

# Stata in Rmd
# devtools::install_github("Hemken/Statamarkdown")
```

```
# VSCode integration and also sublime r-ide
install.packages("languageserver")
```

Temp Installs:

```
# 2020-10-19
# Temp install development version due to but
# https://github.com/rstudio/reticulate/issues/831
devtools::install_github("rstudio/reticulate")
```

## 1.4 R Tests

Test the following file to see if we can execute a R file. Do it inside `r_env` and inside a `r` session.

```
## A simple file with summary statistics using tidyverse
# source('C:/Users/fan/R4Econ/summarize/dist/htmlpdr/fst_hist_onevar.R')
# source('G:/repos/R4Econ/summarize/dist/htmlpdr/fst_hist_onevar.R')
## Another simple file with summary statistics using tidyverse
# source('C:/Users/fan/R4Econ/support/tibble/htmlpdr/fs_tib_basics.R')
# source('G:/repos/R4Econ/support/tibble/htmlpdr/fs_tib_basics.R')
## A file involving estimation
# source('C:/Users/fan/R4Econ/optimization/cesloglin/htmlpdr/fst_ces_plan_linlog.R')
#
# C:/Users/fan/R4Econ/summarize/dist/fst_hist_onevar.Rmd
# C:/Users/fan/R4Econ/support/tibble/fs_tib_basics.Rmd
# C:/Users/fan/R4Econ/optimization/cesloglin/fst_ces_plan_linlog.Rmd
```

## 1.5 R with Radian

R with Radian:

- [Setup Visual Studio Code to run R on VSCode 2021](#)
- [Radian](#)

## 1.6 Running R Inside VSCode

Rstudio seems laggy sometimes, nice to be able to run R from VSCode, use VSCode as an alternative editor.

- [Writing R in VSCode: A Fresh Start](#)

To Run .R scripts from inside VSCode. Here is the official guide: [R in Visual Studio Code](#).

1. Install R following prior steps.
2. Make sure that the package `languageserver` is installed, check `require(languageserver)`.
3. Install [R Extension for Visual Studio Code](#)
  - change setting for `"r.terminal.windows"` to `"C:/Program Files/R/R-4.2.1/bin/R.exe"`
4. Click File -> Open Folder -> Select the folder where the ".R" file to run is located at, this way, the Terminal will be directed to the folder that is currently open, rather than home default for example.  
[By default, the terminal will open at the folder that is opened in the Explorer](#)
5. Run individual files in the folder just opened.

### 1.6.1 RMD in VSCode

Steps for RMD:

1. VSCode already has default markdown editor

2. Install Markdown Preview ehance, which generates a table of content bar on the side, and has math preview correctly
3. In file, F1, and type markdown preview and open up preview

## 1.6.2 RMD and MARKDOWN File Associations

### Change File Extension Association

- Working with RMD file, sometimes want to preview as MD file to view equations, sometimes want to view as RMD file to edit the R code. See [How to make VS Code to treat other file extensions as certain language?](#)
- [change extension association](#) between md and Rmd for Rmd files for example: “Ctrl + shift + p” and “change language mode” from one file association to another.
- Additionally, if there is a standard association we want for RMD, for it to be markdown for example, can add to JSON settings for *file.associations*.

```
"files.associations": {  
  "*.Rmd": "markdown"  
}
```