Encyclopedia of Quantitative Methods in R

Vol. 0: Setting up Your Computer

Sarah Schwartz & Tyson Barrett

Last updated: 2018-08-13

Contents

Introduction

Helpful Websites

Quick R: Basic Statistics

What is R?

R is a language and environment for statistical computing and graphics. (R Core Team, 2018)

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, …) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

What is R Markdown?

According to R Studio:

> "R Markdown is a format that enables easy authoring of reproducible web reports from R. It combines the core syntax of Markdown (an easy-to-write **plain text** format for web content) with embedded **R code chunks** that are run so their output can be included in the final document".

Dynamic Reporting

From Penn State Statistics:

The traditional way** to write a report**

1. Run your analysis in software, like SPSS or R and manually save our output
   - *i.e. saving the ANOVA table or using pdf() to save the graphs*
2. Type your your description and interpretation in a text editor like *Word*
   - *either drag/drop tables and figures, or worse copy-paste and retype all the numbers*

A report written in this way can be problematic. For instance, imagine your *Mentor/collaborator/journal reviewer* telling you that they want to use a sub-sample instead of the entire sample. Or to include a nother variable. You would have to redo all of your work!!

Therefore, in this way **dynamic also means reproducible**, in the sense that people who get the file from you can reproduce the entire work in the report.

How does R Markdown work out to be a .pdf or .html file?

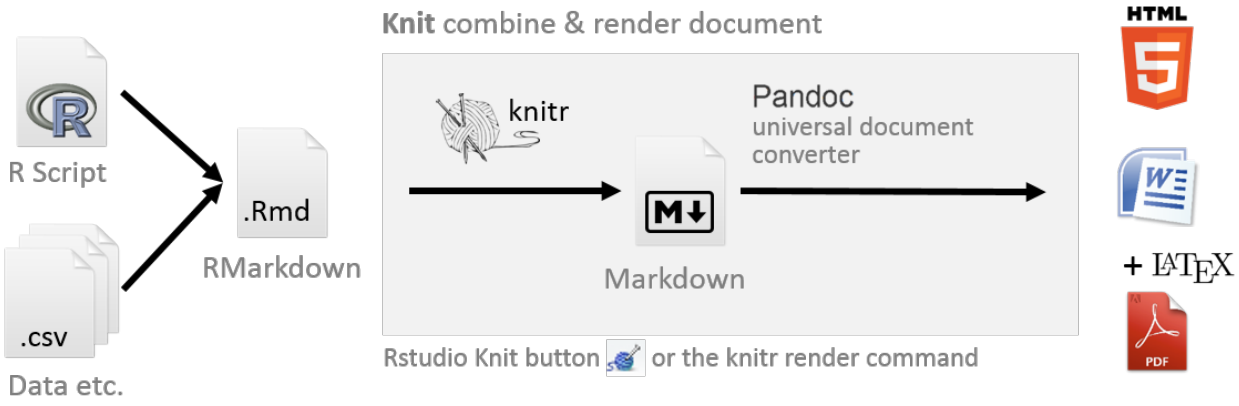

Figure 1:

`R Markdown` is a file with the file extension **.Rmd**, the `knitr` package will then transform the file into a **Markdown** file with the extension **.md.** Then Rstudio can (Xie, 2015):

- Use `LaTeX` to transform the file into a **.pdf**
- Load another package called `markdown` to transform the file into **.html**
- Use Pandoc to even convert to file to a **Word** document (ugly)

Is this a popular** method for creating reports?**

Check out Rpubs. This website shares lots of documents written in the way we will introduce below.

> `R Markdown` documents are fully reproducible. Use a productive **notebook** interface to weave together narrative text and code to produce elegantly formatted output. Use multiple languages including R, Python, and SQL (Allaire et al., 2018).
>
> `knitr` is an engine for dynamic report generation with R. It is a package in the statistical programming language R that enables integration of **R code** into LaTeX, LyX, HTML, Markdown, AsciiDoc, and **text** documents (Xie, 2018).

Figure 2:

Figure 3:

Chapter 1

Install R

Here is where we talk about installing R.

1.1 First Time Installation

Go to: www.r-project.org

Get the latest released version of FREE **Base** *R* from *CRAN*

- Choose a mirror close to your geographical location
- Select **base** *R* for your computer *(Windows, Mac, ect.)*
- The defaults are good...don't change them...just keep clicking *'Next'*

1.2 Update Regularly

Figure 1.1:

Chapter 2

Install R Studio

Here is where we talk about installing R Studio.

2.1 First Time Installation

Go to: www.rstudio.com

Get the latest version of the FREE Open Source **Desktop** Edition of R Studio

- The defaults are good...don't change them...just keep clicking *'Next'*

2.2 Update Regularly

2.3 Panel Layout

Figure 2.1:

Chapter 3

Install TeX

Here is where we talk about installing Tex.

3.1 Use `tinytex` package

3.2 Mac - use `MacTeX`

Go to: http://tug.org/mactex/

- Download (5+ min) to a folder and them double click on the **PKG file**
- Follow the installation instructions.
- You don't need to open anything after MacTeX is finished installing.

Figure 3.1:

3.3 Windows - use MikTeX

Go to: http://miktex.org/download

- Pick the latest version of the **Net Installer**, not the Basic!
- You need the full version 64-bit is better, if you have a 64-bit machine
- When your download is complete, run the downloaded installer.
- Windows may ask you if you want to *"allow this app from an unknown publisher to make changes to your PC"*. If it does, make sure to click **Yes!**
- This is the slowest part…

Chapter 4

Install Packages

We describe packages and their management

4.1 What are packages

> **R packages** are collections of functions and data sets developed by the community. They increase the power of **R** by improving existing base **R** functionalities, or by adding new ones.

More information may be found here: https://www.datacamp.com/community/tutorials/r-packages-guide

---

4.2 INSTALL packages (via the user interface)

> You only need to INSTALL packages ONCE per computer.

In **R Stuido**:

1. Click on the **Packages** tab the panel with the most tabs
2. Click on the word **Instsall** just under and to the left of the tab
3. In the **Packages** box, type in the name of the packages you would like to download. You can do several at once, just seperate them with multiple spaces or a comma.

Note: Leave the installation library path as the default. Also, make sure the box for 'Installing dependencies' is checked.

4.3 LOAD packages (via R code)

> You will need to LOAD packages in EVERY SESSION you want to use them in.

```
library(tidyverse)
```

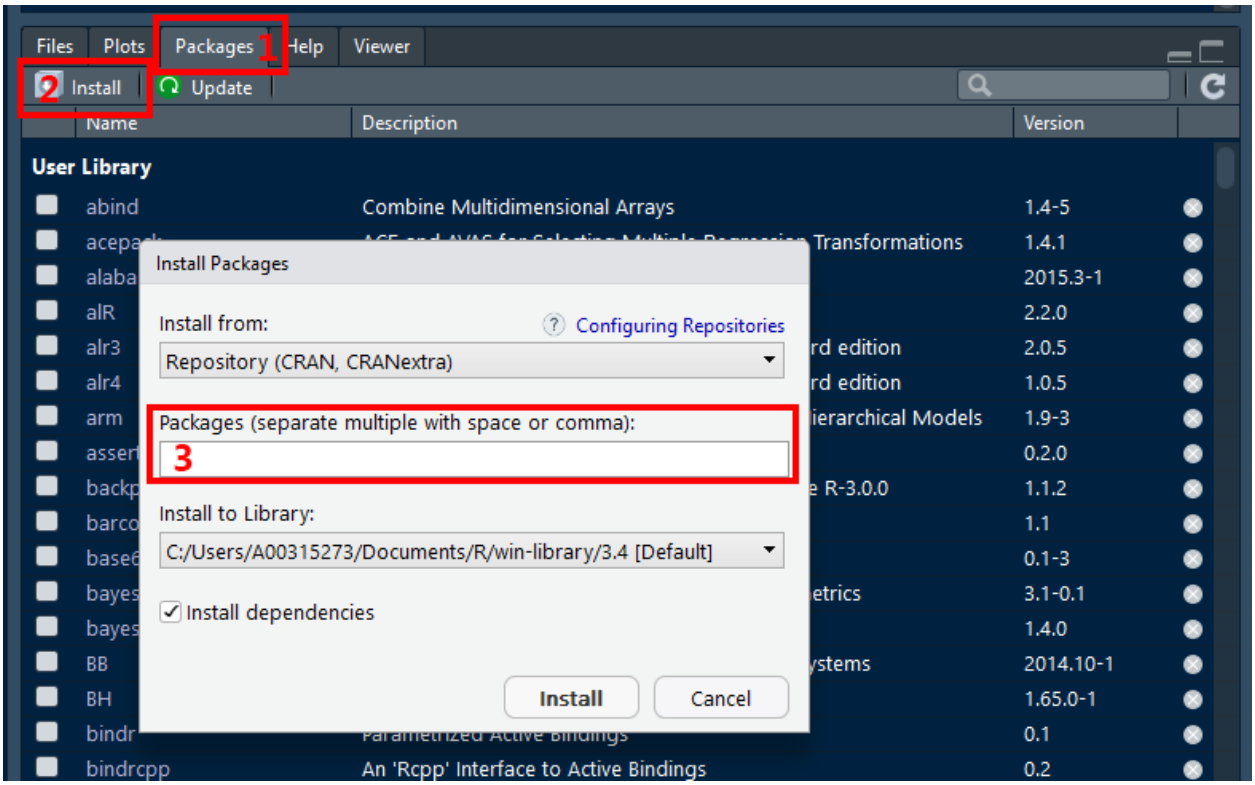

Figure 4.1:

Please don't get confused: `library()` is the command used to **load a package**, and it refers to the **place** where the package is contained, usually a folder on your computer, while a package is the collection of functions bundled conveniently.

Maybe it can help a quote from **Hadley Wickham**, Chief data scientist at RStudio, and instructor of the "*Writing functions in R*" DataCamp course (December 8, 2014):

> "a package is a like a book, a library is like a library; you use `library()` to check a package out of the library"

Here is link to an AWSOME 'cheat sheet' for begginers working with the `tidyverse` package. I highly suggest checking it out.

More 'cheat sheets' are available under the "Help" menu option in **R Studio**

4.4 Updating packages

Chapter 5

Suggested Packages

```
# set global chunk options...
#  this changes the defaults so you don't have to repeat yourself
knitr::opts_chunk$set(comment     = NA,
                      cache       = TRUE,
                      echo        = TRUE,
                      warning     = FALSE,
                      message     = FALSE,
                      fig.align   = "center", # center all figures
                      fig.width   = 6,        # set default figure width to 4 inches
                      fig.height  = 4)        # set default figure height to 3 inches
```

Here is where we talk about usefull packages...

A curated list of awesome *R* packages and tools: https://awesome-r.com/

5.1 The Tidy-Universe from *RStudio*

```
install.packages("tidyverse")
```

> The `tidyverse` (www.tidyverse.org) is an opinionated **collection of *R* packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

5.1.1 Core

The core tidyverse includes the packages that you are likely to use in everyday data analyses. As of `tidyverse 1.2.0`, the following packages are included in the core tidyverse:

```
library(tidyverse)
```

website	description
`dplyr`	A Grammar of Data Manipulation
`forcats`	Tools for Working with Categorical Variables *(Factors)*
`ggplot2`	Create Elegant Data Visualisations Using the Grammar of Graphics
`purrr`	Functional Programming Tools
`readr`	Read Rectangular Text Data

website	description
`stringr`	Simple, Consistent Wrappers for Common String Operations *(Text)*
`tibble`	Simple Data Frames
`tidyr`	Easily Tidy Data with `spread()` and `gather()` Functions

5.1.2 Supplemental

The tidyverse also includes many other packages with more specialised usage. They are not loaded automatically with `library(tidyverse)`, so you'll need to load each one with its own call to `library()`.

```
library(haven) # example...may replace with any individual package name
```

website	description
`broom`	Convert Statistical Analysis Objects into Tidy Tibbles
`haven`	Import and Export **SPSS**, **Stata** and **SAS** Files
`hms`	Pretty Time of Day
`lubridate`	Make Dealing with Dates a Little Easier
`magrittr`	A Forward-Pipe Operator for *R*
`glue`	Interpreted String Literals
`readxl`	Read **Excel** Files
`tibble`	Simple Data Frames

5.2 Groups of Individual Packages on *CRAN*

5.2.1 Creating Nice Tables

```
package_list_tables <-  c("furniture",
                          "pander",
                          "stargazer",
                          "texreg",
                          "xtable")
```

website	description
`furniture`	Tables for Quantitative Scientists
`pander`	An R 'Pandoc' Writer (makes tables look nice)
`stargazer`	Well-Formatted Regression and Summary Statistics Tables
`texreg`	Conversion of R Regression Output to LaTeX or HTML Tables
`xtable`	Export Tables to LaTeX or HTML

5.2.2 Visualization

```
package_list_visual <- c("RColorBrewer",
                         "gghighlight",
                         "ggthemes",
                         "ggfortify",
                         "ggalt",
                         "ggExtra",
                         "GGally",
                         "ggeffects",
                         "corrplot",
                         "gpairs",
                         "gridextra",
                         "likert",
                         "vcd",
                         "scales",
                         "cowplot",
                         "yarrr")
```

website	description
RColorBrewer	Color Palettes
gghighlight	Highlight Lines and Points in ggplot2
ggthemes	Extra Themes, Scales, and Geoms for ggplot2
ggExtra	Add Marginal Histograms to ggplot2, and More ggplot2 Enhancements
ggfortify	Data Visualization Tools for Statistical Analysis Results
ggalt	Lots of extras for ggplot2
GGally	Extension to ggplot2
corrplot	Visualization of a Correlation Matrix
gpairs	The Generalized Pairs Plot
gridextra	Miscellaneous Functions for "Grid" Graphics
likert	Analysis and Visualization Likert Items
vcd	Visualizing Categorical Data
scales	Scale Functions for Visualization
cowplot	Streamlined Plot Theme & Annotations for ggplot2
yarrr	The Pirate's Guide to *R*

5.2.3 Generally Handy

```
package_list_general <- c("psych",
                          "polycor",
                          "corpcor",
                          "sjlabelled",
                          "sjPlot",
                          "sjmisc",
                          "sjstats",
                          "Hmisc",
                          "labelled")
```

website	description
polycor	Polychoric and Polyserial Correlations
psych	Psychological or Psychometric Procedures
corpcor	Covariance and (Partial) Correlation
sjlabelled	Labelled Data Utility Functions
sjPlot	Data Visualization for Statistics in Social Science
sjmisc	Data and Variable Transformation Functions
sjstats	Convenient Functions for Common Statistical Computations
Hmisc	Harrell Miscellaneous
labelled	Manipulating Labelled Data

5.2.4 t-Tests, ANOVA, and RM ANOVA

```
package_list_anova <- c("afex",
                        "emmeans",
                        "corpcor",
                        "multicomp",
                        "multcompView")
```

website	description
afex	Analysis of Factorial Experiments
emmeans	Estimated Marginal Means, aka Least-Squares Means
multicomp	Simultaneous Inference in General Parametric Models
multcompView	Visualizations of Paired Comparisons

5.2.5 Regression (ML, GLM)

```
package_list_regression <- c("car",
                             "effects",
                             "predictmeans")
```

website	description
car	Companion to Applied Regression
effects	Effect Displays for Linear, Generalized Linear, and Other Models
predictmeans	Calculate Predicted Means for Linear Models

5.2.6 Multilevel Models (MLM, HLM, GEE)

```
package_list_multilevel <- c("nlme",
                             "lme4",
```

```
                          "lmerTest",
                          "HLMdiag",
                          "geepack",
                          "gee",
                          "gee4",
                          "optimx",
                          "MuMIn")
```

website	description
`nlme`	Linear and Nonlinear Mixed Effects Models
`lme4`	Linear Mixed-Effects Models
`lmerTest`	Tests in Linear Mixed Effects Models
`HLMdiag`	Diagnostic Tools for Hierarchical (Multilevel) Linear Models
`geepack`	Generalized Estimating Equation Package
`gee`	Generalized Estimation Equation Solver
`gee4`	Generalised Estimating Equations (GEE/WGEE)
`optimx`	A Replacement and Extension of the `optim()` Function
`MuMIn`	Multi-Model Inference

5.2.7 Structural Equation Modeling (SEM)

```
package_list_sem <- c("lavaan",
                      "OpenMx",
                      "sem",
                      "semPlot")
```

website	description
`lavaan`	Latent Variable Analysis
`OpenMx`	Extended Structural Equation Modelling
`sem`	Structural Equation Modelling
`semPlot`	Path Diagrams and Visual Analysis of Various SEM Packages' Output

---

5.2.8 Random Forests

```
package_list_forest <- c("randomForest",
                         "randomForestSRC",
                         "ggRandomForests",
                         "party",
                         "partykit")
```

website	description
`randomForest`	Random Forests for Classification and Regression

website	description
randomForestSRC	Random Forests for Survival, Regression, and Classification
ggRandomForests	Visually Exploring Random Forests
party	A Laboratory for Recursive Partytioning
partykit	A Toolkit for Recursive Partytioning

5.2.9 Other Models

```
package_list_models <- c("mgcv",
                         "glmnet",
                         "survival",
                         "caret")
```

website	description
mgcv	Mixed GAM Computation Vehicle with Automatic Smoothness Estimation
glmnet	Lasso and Elastic-Net Regularized Generalized Linear Models
survival	Survival Analysis
caret	Classification and Regression Training

5.2.10 Reproducibility and Reporting

```
package_list_report <- c("bookdown",
                         "blogdown",
                         "tidytex",
                         "xaringan",
                         "slidify",
                         "ReportRs")
```

website	description
bookdown	
blogdown	
tidytex	
xaringan	
slidify	
ReportRs	

5.2.11 REDCap Interface

```
package_list_redcap <- c("REDCapR",
                         "redcapAPI")
```

website	description
redcapAPI	Interface to 'REDCap'
REDCapR	Interaction Between R and REDCap

5.2.12 Creating Your Own Packages

```
package_list_package <- c("devtools",
                          "testthat",
                          "roxygen2")
```

website	description
devtools	
testthat	
roxygen2	

5.3 Installing All the *CRAN* packages at Once

Review the list of all the packages above:

```
package_list_tables
```

```
[1] "furniture" "pander"    "stargazer" "texreg"    "xtable"
```

```
package_list_visual
```

```
 [1] "RColorBrewer" "gghighlight"  "ggthemes"     "ggfortify"
 [5] "ggalt"        "ggExtra"      "GGally"       "ggeffects"
 [9] "corrplot"     "gpairs"       "gridextra"    "likert"
[13] "vcd"          "scales"       "cowplot"      "yarrr"
```

```
package_list_general
```

```
[1] "psych"      "polycor"    "corpcor"    "sjlabelled" "sjPlot"
[6] "sjmisc"     "sjstats"    "Hmisc"      "labelled"
```

```
package_list_anova
```

```
[1] "afex"         "emmeans"      "corpcor"      "multicomp"
[5] "multcompView"
```

```
package_list_regression
```

```
[1] "car"          "effects"      "predictmeans"
```

```
package_list_multilevel
```

```
[1] "nlme"     "lme4"     "lmerTest" "HLMdiag"  "geepack"  "gee"
[7] "gee4"     "optimx"   "MuMIn"
```

```
package_list_sem
```

```
[1] "lavaan"  "OpenMx"  "sem"     "semPlot"
```

```
package_list_forest
```

```
[1] "randomForest"    "randomForestSRC" "ggRandomForests" "party"
[5] "partykit"
```

```
package_list_models
```

```
[1] "mgcv"     "glmnet"   "survival" "caret"
```

```
package_list_report
```

```
[1] "bookdown" "blogdown" "tidytex"  "xaringan" "slidify"  "ReportRs"
```

```
package_list_redcap
```

```
[1] "REDCapR"   "redcapAPI"
```

```
package_list_package
```

```
[1] "devtools" "testthat" "roxygen2"
```

Get all the packages from *CRAN (updates if new version available)*:

```
install.packages(c("tidyverse",
                   package_list_tables,
                   package_list_visual,
                   package_list_general,
                   package_list_anova,
                   package_list_regression,
                   package_list_multilevel,
                   package_list_sem,
                   package_list_forest,
                   package_list_models,
                   package_list_report,
                   package_list_redcap,
                   package_list_package))
```

5.4 Other Developmental non-*CRAN* Packages on *GitHub*

First, make sure you have the `devtools` package installed on your computer *(hint: its in the `package_list_package` list above)*.

```
install.packages("devtools")
```

5.4.1 Templates for writing tutorials, practicals or examination papers with *RMarkdown*

`unilur` is a *R* package to help writing tutorials, practicals or examination papers with *RMarkdown*.

With `unilur` you can render the following outputs from a single rmarkdown file:

- the exam or tutorial questions (answers remaining hidden) as a PDF or HTML file.
- the exam or tutorial questions with sample answers as a PDF or HTML file.

In addition, you will be able to:

- Create coloured boxes to highlight some markdown or R content.
- Create examination papers with
  - multiple choice questions
  - a candidate identification form
  - dotted lines placeholders to fill in answers
- Create a new *RMarkdown* file with solution chunks replaced by empty ones.

  Website: GitHub - `unilur`

  Tutorial: blog post - `unilur`

```
devtools::install_github("koncina/unilur")
```

5.4.2 Prepare APA journal articles with *RMarkdown*

`papaja` is a *R* package in the making including a *RMarkdown* template that can be used with (or without) *R* **Studio** to produce documents, which conform to the **American Psychological Association (APA)** manuscript guidelines (6th Edition). The package uses the *LaTeX* document class `apa6` and a .docx-reference file, so you can create PDF documents, or Word documents if you have to. Moreover, papaja supplies *R* functions that facilitate reporting results of your analyses in accordance with APA guidelines.

`papaja` has not yet been submitted to *CRAN* because it is under active development. Currently, there are still a couple of loose ends they would like to tie up before we release the package to a larger audience. There are two versions you can install from the *GitHub* website.

Website: GitHub - `papaja`

Tutorial: eBook - `papaja`

```
# Install the stable development verions from GitHub
devtools::install_github("crsh/papaja")

# Install the latest development snapshot from GitHub
devtools::install_github("crsh/papaja@devel")
```

Chapter 6

Kniting Notebooks

6.1 Storing all associated files

If you are using any files, such as *datasets* or *images*, they need to be stored in the same folder location as the R Notebook (`.Rmd` file).

This folder location must be the **Working Directory** for the R Studio session. If you opened your `.Rmd` notebook file by double-clicking on its name, then this should be the case.

6.2 Setting the working directory

To ensure that R Studio knows where to find the files, you can manually set the **Working Directory** through the menu:

- Click `Session`
- Select `Set Working Directory` by hovering your mouse over it
- Click on `To Source File Location`

You can double check that you were successful by

- Click on the `Files` tab in the many-tab panel
- Click on the button with the gear that says `More`
- Click `Go To Working Directory`

At this point you should see all the files that reside in the folder location where the open `.Rmd` files is also saved.

6.3 Press the *Knit* button

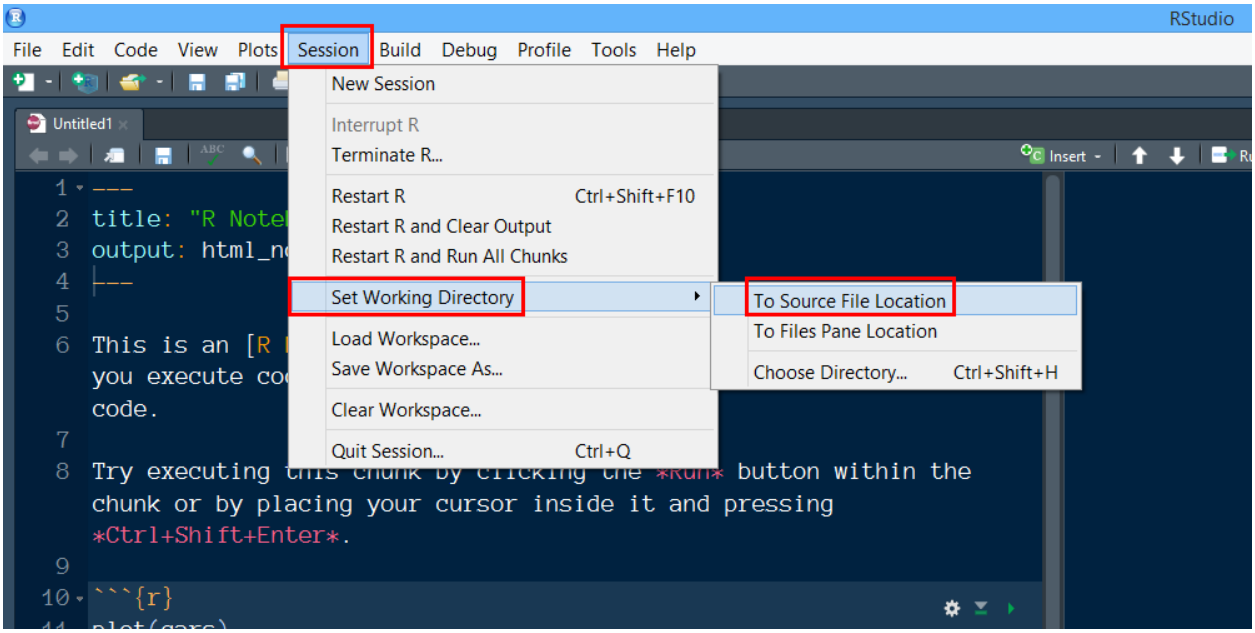

Figure 6.1:

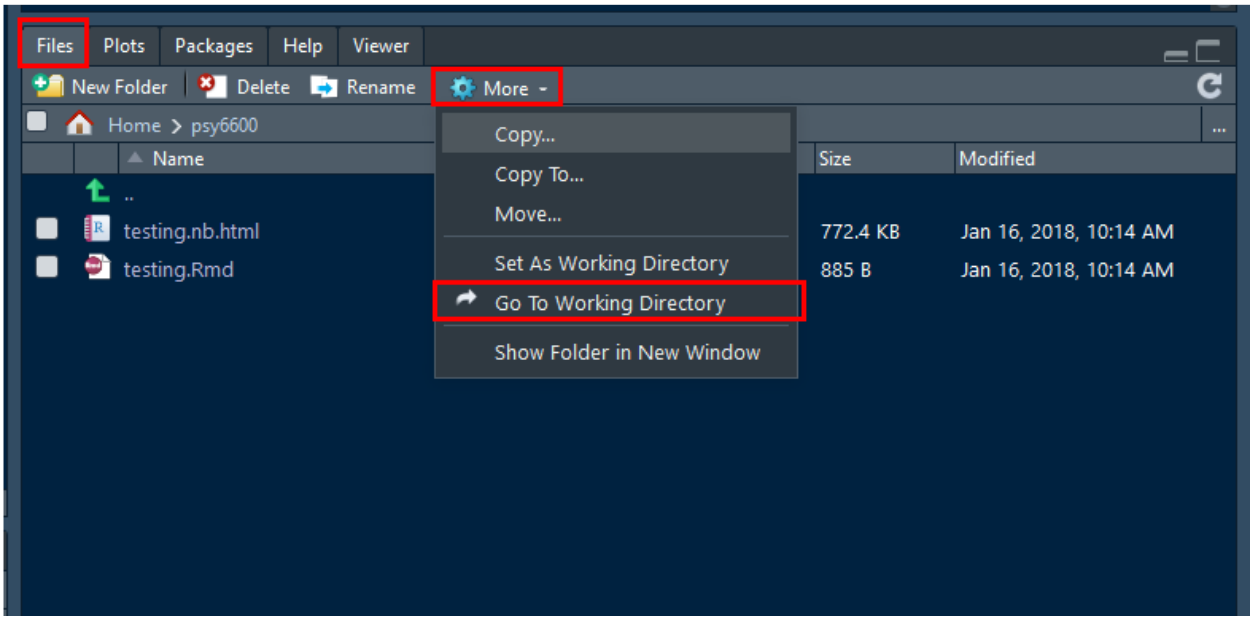

Figure 6.2:

Bibliography

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., and Chang, W. (2018). *rmarkdown: Dynamic Documents for R*. R package version 1.10.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2018). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.20.