

PA 5928

Tao Tao

2019-08-13

Contents

1	Prerequisites	5
2	Introduction	7
3	Literature	9
4	Methods	11
5	Applications	13
5.1	Example one	13
5.2	Example two	13
6	Final Words	15

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")  
# or the development version  
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading #.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.name/tinytex/>.

Chapter 2

Introduction to RStudio

2.1 What is R

R is a type of programming language and supports many tasks including statistical computation and graphics.

2.2 What is R Studio

RStudio is a programming software for editing and running R code.

2.3 Install R + RStudio

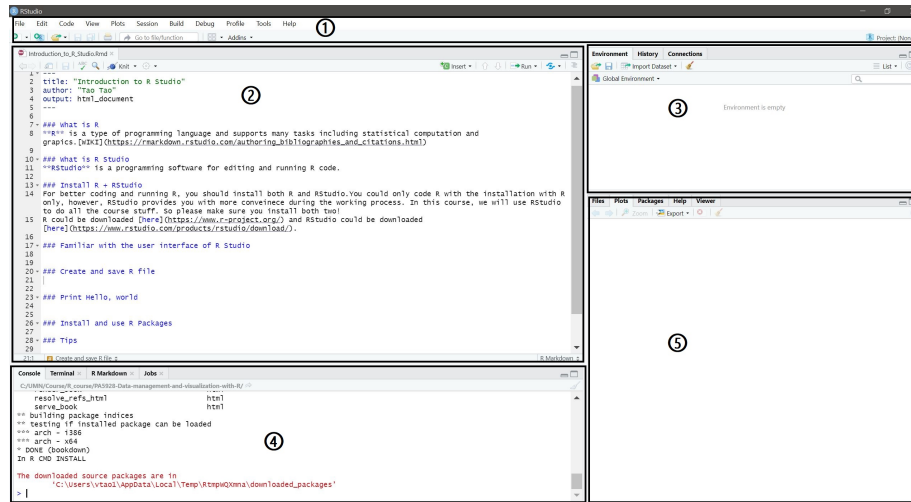
For better coding and running R, you should install both R and RStudio. You could only code R with the installation with R only, however, RStudio provides you with more convenience during the working process. In this course, we will use RStudio to do all the course stuff. So please make sure you install both two! R could be downloaded [here](#) and RStudio could be downloaded [here](#).

2.4 Familiar with the user interface of R Studio

Below is the figure showing a screenshot of the user interface of RStudio. You will find couple of panes/windows with different usages.

1. **Menu/Tool Bar**
2. **Source** The pane where you write and edit your code.
3. **Environment/History** Environment lists all the variables that you are using. History presents the codes you have runned before.
4. **Console** Console is original R interactive window. You could input codes and see the results here.

5. **Plots/Help** Plots window shows the output figures. Help windows presents the information of the function or package you check.



2.5 Create and save R file

Three ways to create a R file in the RStudio:

1. Menu -> File -> New File -> R Script
2. Shortcut: Ctrl + Shift + N
3. Tool Bar -> New file button

Also three ways to save R file

1. Menu -> File -> Save
2. Shortcut: Ctrl + S
3. Tool Bar -> Save file button

2.6 Print Hello, world

Now, let's try to input some codes and run them! Let's let R print the very classic "Hello, world!" with `print()` function. We could run the codes in several ways:

1. Select the codes or put the cursor in the line of your code, and click the Run button located in the right-top position of the console pane.
2. Select the codes or put the cursor in the line of your code, and use shortcut: Ctrl + Enter
3. You could also click the Re-run button near the Run button to re-run the codes you runned last time


```
print('Hello, world!')
```

```
## [1] "Hello, world!"
```

Because what we need to output here is a **string variable**, we have to put them in the quotation mark. Either single quotation or double quotation mark works well. Let's see another example.

```
print(5928)
```

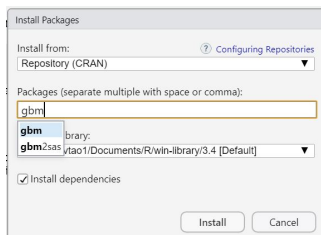
```
## [1] 5928
```

Here, 5928 is an integer and we do not need to put them in the quotation mark.

2.7 Install and use R Packages

R is easy to use because it has many packages with different usages. These packages could help you accomplish some complex tasks with several lines of codes. Some packages have been already been installed and you could use them directly. However, most of the packages have to be installed before you want to use them. There are couple of ways you could install a package. Let's take the **gbm** package for example.

1. Menu -> Tools -> Install Packages... -> Input the package name -> Click Install button



2. Use the code below:

```
install.packages("gbm")
```

After the installation of the package, you have to import it with **library()** function before you use the functions in it.

```
library(gbm)
```

We will spend more time in the future classes to explore the various R packages and their different usages.

2.8 Make notes

It is important to write notes for your codes. It could help others or even yourself understand your codes easily. Use hash tag to indicate the notes. For example,

```
gbm1 <- gbm(AvgMet~PkAreaH+StpNumH+DisToMin, # formula
            data=MetM,                        # dataset
            var.monotone=c(+1, rep(0,10),rep(0,15)),
            distribution="gaussian",           # see the help for other choices
            n.trees=5000,                     # number of trees
            shrinkage=0.001,                  # shrinkage or learning rate, 0.001 to 0
            interaction.depth=6,               # 1: additive model, 2: two-way interact
            bag.fraction = 0.5,                # subsampling fraction, 0.5 is probably i
            n.minobsinnode = 10,               # minimum total weight needed in each no
            cv.folds = 5)
```

R will not run the codes after hash tags in each line.

Please try to write simple but necessary notes for the codes. Keep this as a good habit and you will thank yourself in the future.

2.9 Tips (Updating)

1. You could divide your codes into sections by interting chunks before each sections with the shortcut: Ctrl + Shift + R. This will help you organize your codes.

2.10 Reference

1. Introduction to RStudio