STAT 2005 – Programming Languages for Statistics

Tutorial 1 Introduction to R

2020

LIU Ran

Department of Statistics, The Chinese University of Hong Kong

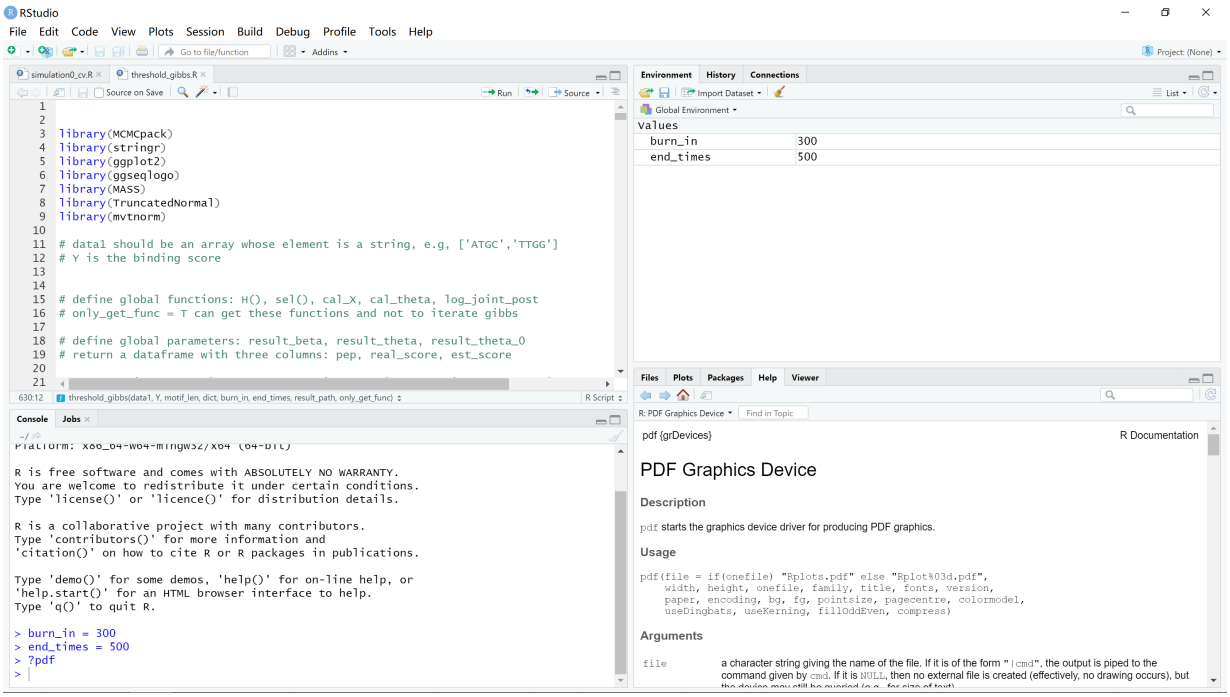
1 RStudio

1.1 RStudio and R

RStudio is an integrated development environment (IDE) for R. It allows users to run R in a more user-friendly environment. You can use R without RStudio, but not vice versa.

To be more specific, it is like the relationship between eyes and eyeglasses. R can be regarded as the eyes that can work without RStudio. And RStudio can be regarded as the eyeglasses that can let us use R more conveniently.

1.2 Interface



1. RScript(s)(Top-Left):
2. Console(Bottom-left):
3. Environment, History, Connections(Top-right):
4. Files, Plots, Packages, Help, Viewer(Bottom-right):

1.2.1 Console

The console is where you can type commands and see the output. It has the same function as the RGUI console.

1.2.2 Environment, History, Connections

After defining variables and assign values to them, you can check them in the Environment tab.

History tab shows a list of commands used so far.

Connections tab is used to connect outside database(not very often).

1.2.3 Files, Plots, Packages, Help, Viewer

For files tab, you can search files that on your computer and open it. Meanwhile, you can also set working directory by clicking on the 'more' icon and click on setting this path as working directory.

You can see the figures you draw in the console from Plots tab.

After using **help(Func)** or **?Func** command, you can see the documentation of the **Func** you want to know.

Viewer is used to display local web files(not very often).

For Packages, you can see the packages we have and the packages we have already loaded in the current environment. The definition of packages will be presented in later classes.

1.2.4 RScript(s)

The R script is where you can keep a record of your work. (Homework Format: ***.R)

```
# Question 1
a = c('1','2','3','4')
a = as.integer(a)
b = rep(a,4)
```

To create a new R script you can go to File → New → R Script, or click on the icon with the '+' sign and select 'R Script'. Make sure to save the script.

To run current line code, click on 'Run' or simply press Ctrl+Enter.

If you source(run) a R script file, it will be carried out all the contents in the script file line by line.

It is equivalent to copy the lines to the console and run them.

Remark 1.1. Although RStudio, Sublime and VS Code provide useful and convenient functions such as auto complete, you should still be able to write code **by hand (in the examination)**.

Remark 1.2. All materials provided in tutorials are based on R in Windows.

2 Tips

- If you have any question using a function or a package: **help** or ? .

```
> help(length)
> ?length
```

- **Change working directory** before writing scripts: **File** → **More**.
- Usually, choose **No** when closing R.
- Recall previous executed commands: ↑ . (not only for R)

- Comment the code: **#**
  (Shortcut in RStudio: Ctrl + Shift + c).

```
> 1+2 #3
[1] 3
> #####1+2 3
```

- Local assignment operator: =,
  Global assignment operator: **<-**,
  Present the object to the upper layer:**<<-**.(optional and not recommended)(superassignment)

```
> sin(x=1)
[1] 0.841471
> x
Error: object 'x' not found
> sin(x<-1)
[1] 0.841471
> x
[1] 1
> assign_value_to_y<-function(){y <<- 2}
> assign_value_to_y()
> y
[1] 2
```

- R is **case sensitive**.

```
> x = c(1,2,3)
> X
Error: object 'X' not found
> x
[1] 1 2 3
> Mean(x)
Error in Mean(x) : could not find function "Mean"
> mean(x)
[1] 2
```

- Remove all objects and run your code once again to make sure you have defined everything in your code.(An example shown in tutorials)

```
> rm(list=ls(all=TRUE))
```

- Clear the console box.

```
> cat('\f')
```

Remark 2.1. It will not affect the history or any objects.

3 Data objects

3.1 Vector

- Combine Command: **c**

```
> (d = c(0,1,2,3,4))
[1] 0 1 2 3 4
> length(d)
[1] 5
> e = c("apple", "orange", "lemon")
> c(d, e) # Numbers are converted into characters automatically.
[1] "0"      "1"      "2"      "3"      "4"      "apple"  "orange" "lemon"
```

- Sequence Command: **seq**

```
> seq(1, 10) # Equivalent to 1:10.
 [1]  1  2  3  4  5  6  7  8  9 10
> seq(10, 1, -3)
[1] 10  7  4  1
> seq(0, 1, length=11)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> seq(0, by=0.03, length=10)
 [1] 0.00 0.03 0.06 0.09 0.12 0.15 0.18 0.21 0.24 0.27
```

- Replicate Command: **rep**

```
> rep(1,5)
[1] 1 1 1 1 1
> rep(1:3, 2)
[1] 1 2 3 1 2 3
> rep(c(0,6), 3)
[1] 0 6 0 6 0 6
> rep(c(0, "x"), 3)
[1] "0" "x" "0" "x" "0" "x"
> rep(1:3, 3:1)
[1] 1 1 1 2 2 3
```

3.2 Matrix

```
> (m = matrix(1:6, ncol=2))
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> dim(m)
[1] 3 2
> nrow(m)
[1] 3

> (m = matrix(1:6, nrow=2))
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

> (m = matrix(1:6, ncol=2, byrow=T))
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6

> x = matrix(1:6, ncol=2)
> y = matrix(7:12, ncol=2)
> rbind(x, y)
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
[4,]    7   10
[5,]    8   11
[6,]    9   12
> (z = cbind(x, y))
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

Remark 3.1. Basic knowledge about matrix: Wikipedia, Linear Algebra by Prof. Gilbert Strang in MIT or MATH1030 (Linear Algebra I) in CUHK.

3.3 (Optional) Array: High-dimensional Matrix

- Array is useful in storing high dimensional data, e.g., simulation results.

```
> (a = array(1:6, dim=c(2,3)))
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> (b = array(1:18, dim=c(2,3,3)))
, , 1

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12

, , 3

     [,1] [,2] [,3]
[1,]   13   15   17
[2,]   14   16   18
> rbind(b[ , 2, ], a)
     [,1] [,2] [,3]
[1,]    3    9   15
[2,]    4   10   16
[3,]    1    3    5
[4,]    2    4    6
```

Remark 3.2. By default, R will use elements to create a matrix/array **columnwisely**.

3.4 List

- List can store different types of data.

```
> x = c(1:3)
> y = c("a", "b")
> z = matrix(1:6, nrow=2)
> (w = list(x, y, z))
[[1]]
[1] 1 2 3

[[2]]
[1] "a" "b"

[[3]]
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> length(w)
[1] 3
> w[[1]]
[1] 1 2 3
> w[1]
[[1]]
[1] 1 2 3
> class(w[1])
[1] "list"
```

3.5 Data frame

- Data frame is a special kind of list.

```
> (L = LETTERS[1:3])
[1] "A" "B" "C"
> (f = sample(L, 10, replace = TRUE))
 [1] "C" "A" "B" "A" "A" "B" "A" "A" "A" "C"
```

```
> (d = data.frame(x = 1, y = 1:10, f = f))
   x  y f
1  1  1 C
2  1  2 A
3  1  3 B
4  1  4 A
5  1  5 A
6  1  6 B
7  1  7 A
8  1  8 A
9  1  9 A
10 1 10 C
```

3.6 Factor

- Factor can store character data.

```
> (s = factor(substring("statistics", 1:10, 1:10)))
 [1] s t a t i s t i c s
Levels: a c i s t
```

Remark 3.3. `Levels` is important for factor.

Feel free to email your questions to me. rliu@linklink.cuhk.edu.hk

By the way, if you want to ask something about homework, you should email to the marker Mr. Zhang 1155149267@link.cuhk.edu.hk

Please submit your homework files with the '*.r' or '*.R' extension.