CHAPTER 17

R Basics

Starting from this chapter, the rest of the book will focus on R. To facilitate a quick learning, every chapter is kept very short, usually less than ten pages and covers a narrow topic. Even for those simple and narrow topics, we will discuss only essential features. The objective is to make learning R less time-consuming and more enjoyable. Since we have discussed the basic concepts related to R in chapters 1 and 2, there will be many overlapping topics between chapters 1 and 2 and chapters 17 and 18. Thus, if you have gone through the first two chapters, just skip chapters 17 and 18.

17.1. Installation of R

Based on the following steps, we can download and install R.

1. Go to http://www.r-project.org.
2. Click **Download** on the left-hand side.
3. Choose a mirror address.
4. Download the appropriate PC software (Windows, Mac).

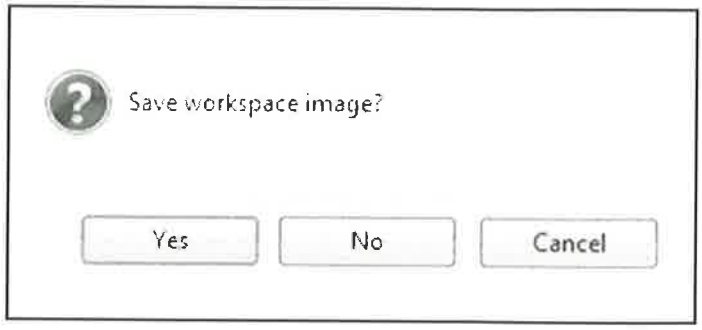
17.2. Starting and Quitting R

To launch, we double-click the R icon on our desktops.

To quit R, type q() from the R prompt > (see below).

```
> q() # the first way to quit
```

Note that > is the R prompt. When quitting, you will be asked whether to save the work-space image. If you want to save your variables or programs for future use, then answer yes. At this stage, just choose no.



A leading # before a line or a phrase indicates that this is a comment.

```
> # this is a comment line
```

There are several other ways to quit R. The second way to quit is given below.

```
> quit() # the 2nd way to quit
```

Here is the third way to quit:

```
> # click "file" – >"exit" # the 3rd way to quit
```

```
> # click right-top "x" # the 4th way to quit
```

17.3. R Basics

The simplest way to assign a value to a variable is to use <-.

```
> x<-10
```

To show its value, simply type the variable name.

```
> x
[1] 10
```

You don't have to define a variable before assigning a value to it.

```
> # a variable is not formally defined before its assignment
> fv<-100 # you don't define the variable called fv
```

R is case sensitive. This means that X and x are two different variables.

```
> x<-1.234
> X
Error: object 'X' not found
```

A comma is not necessary after each command line when the command stands alone. However, if we put several commands in one line, we need to use a semicolon (;) to separate them.

```
> fv<-10; pv<-100; n<-10; rate<-0.05
```

We have the following easy way to assign a vector. In the following example, **c()** means column.

```
> x<-c(1,2.5,4,6) # assign a vector
> y<-1:50 # from small to big with increment of 1
> z<-10:0 # from big to small
```

It is always a good idea to list all variables (objects). For this purpose, we use **ls()**.

```
> ls()
```

When a variable is no longer useful, we can remove it from the memory.

```
> rm(x) # remove variable called x
> rm(x,y) # remove both x and y
> rm(list=ls()) # remove all objects (variables)
# the second way to remove all objects
# go to "Misc" on the menu bar -> "Remove all objects"
```

To make our typing easier, we use the up and down arrow keys to recall our previous commands.

```
# use upper (down) arrow keys to recall previous commands
```

To print one line on the screen, we can use **cat("our sentence here")**.

```
> cat("hello, world!\n") # \n is for new line
```

17.4. Finding Help

There are many ways to find information related to specific functions. Assume that we are interested in mean. We then issue **help(mean)**.

```
> help(mean) # find information related to mean
> ?mean # find information related to mean
> example(median) # show examples related to median
```

If you're not sure about the exact spelling of a function, we use the **apropos()** function instead.

```
> apropos("me") # find information related to "me"
```

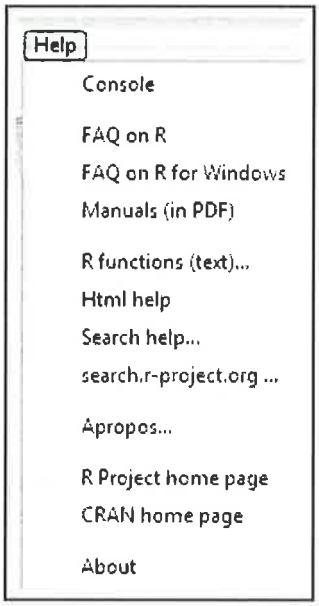
```
# second way to use apropos
# click "Help" on the menu bar
# click "apropos ..."
# enter your phrase
```

```
> # file - -> about [see the version of the current R]
```

```
> help.start() # start help
```

```
> help.search("topic") # start help
```

An alternative way is to click **Help** on the menu bar (see the following graph).



17.5. Using R as an Ordinary Calculator

We can use R to make calculations just like an ordinary calculator.

```
> x<-1:50
> mean(x) # you can try max(), min(),median(),sd(),var(),range()
```

We can use the **ceiling()** and **floor()** functions to process our values.

```
> x<-9.5
> ceiling(x)
[1] 10
```

```
> x<-9.5
> floor(x)
[1] 9
```

We can use the as.integer() function to convert a real number into an integer.

```
> x<-9.5
> y<-as.integer(x)
> y
[1] 9
```

The following table summarizes a set of most widely used functions.

Table 17.1. A list of some basic functions

function	Meaning	Examples
mean(x)	Mean	x<-1:10 mean(x) # [1] 5.5
median(x)	Median	median(x) # [1] 5.5
min(x)	Minimum	min(x) # [1] 1
max(x)	Maximum	max(x) # [1] 10
var(x)	Variance	>var(x) # [1] 9.166667
sd(x)	Standard deviation	sd(x) # [1] 3.027650
exp(x)	Exponential function	exp(2.3) # [1] 9.974182
log(x)	Natural log function	log(4.5) # [1] 1.504077
log10(x)	Log function based on 10	log10(4.3) # [1] 0.6334685
sum(x)	Take the summation	sum(x) # [1] 55
sort(x)	Sort in ascending order	
range(x)	Range of a variable	x<-1.5:10 range(x) # [1] 1.5 9.5
diff(x)	Calculate the difference for a vector	x<-c(1,2.3,4.5) diff(x) # [1] 1.3 2.2
ceiling(x)	Get the smallest integer larger than x	x<-9.5 ceiling(x) # [1] 10
floor(x)	Get the largest integer smaller than x	x<-9.5 floor(x) # [1] 9
as.integer(x)	Take the integer value	x<-9.5 as.integer(x) # [1] 9
prod()	Get product of a vector >	x<-1:3 > prod(x) # [1] 6
quantile(x)	> quantile(x) 0% 25% 50% 75% 100% -2.3210170 -0.6862249 0.1460511 0.7151348 2.0682096	
	> quantile(x,probs=c(0.01,0.05,0.95,0.99)) 1% 5% 95% 99% -2.000058 -1.609246 1.256627 1.476727	

Sometimes we need to change the directory for convenience. The related procedure is given below.

```
# change the directory
# [click] File – > "Change dir…" [choose your working directory]
```

We can change our starting directory by modifying the properties of our R icon.

1. Right-click the R icon on your desktop.
2. Click **Properties**.
3. Choose your directory in **Start in** (e.g., **C:\test**).