

Chapter 1: The First Class for R Programming

Kevin Kam Fung Yuen

PhD, Senior Lecturer, School of Business, Singapore University of Social Sciences

kfyuen@suss.edu.sg, kevinkf.yuen@gmail.com

May 26, 2019

Contents

1 Overview of R	2
2 Setting up R Integrated Development Environment	2
3 Cheatsheets in RStudio	2
4 Understanding RStudio UI	2
5 Using Help in RStudio	4
5.1 Start Help	4
5.2 Search in Help	6
6 External Learning resources	6
7 Basic Arithmetic Operations	7
8 Working Directory and Creating Project	8
8.1 Working Directory	8
8.2 Creating Project	8
9 Assignment and Data Object	9
10 Constants	11
11 Exercises	12
11.1 Exercise 1	12
11.2 Exercise 2	12
11.3 Exercise 3	12
12 References	13

1 Overview of R

In general, R provides an interactive environment for data analytics. The user may not require significant programming knowledge at the beginning stage and will gradually enrich more programming knowledge in latter stage.

According to R core team, R is an integrated suite of software facilities for data manipulation, calculation and graphical display. R has

1. an effective data handling and storage facility,
2. a suite of operators for calculations on arrays, in particular matrices,
3. a large, coherent, integrated collection of intermediate tools for data analysis,
4. graphical facilities for data analysis and display either directly at the computer or on hard copy, and
5. a well-developed, simple and effective programming language (called ‘S’) which includes conditionals, loops, user-defined recursive functions and input and output facilities. (Indeed most of the system supplied functions are themselves written in the S language.)

2 Setting up R Integrated Development Environment

To set up the R integrated development environment, we need to install R and then install RStudio.

1. Installing R According to the user’s operation system (OS), the user should download and install the latest version of R installation files with one of the links below (It may change in the future).
 - Microsoft Windows: <http://cran.r-project.org/bin/windows/base/>
 - MacOS: <http://cran.r-project.org/bin/macosx/>
 - Linux: <http://cran.r-project.org/bin/linux/>
2. Installing RStudio Browse the website with the following link. <https://www.rstudio.com/products/rstudio/download>

There are several versions of R studio. For this foundation course, we use “RStudio Desktop Open Source License”, which is FREE.

3 Cheatsheets in RStudio

Whilst there are many functions and commands in R, RStudio includes the Cheatsheets for quick references in various kinds of projects (Fig. 1).

4 Understanding RStudio UI

To have a quick look for the RStudio UI, we just open *RStudio IDE Cheat Sheet*. Part of essential information is shown in Fig. 2)

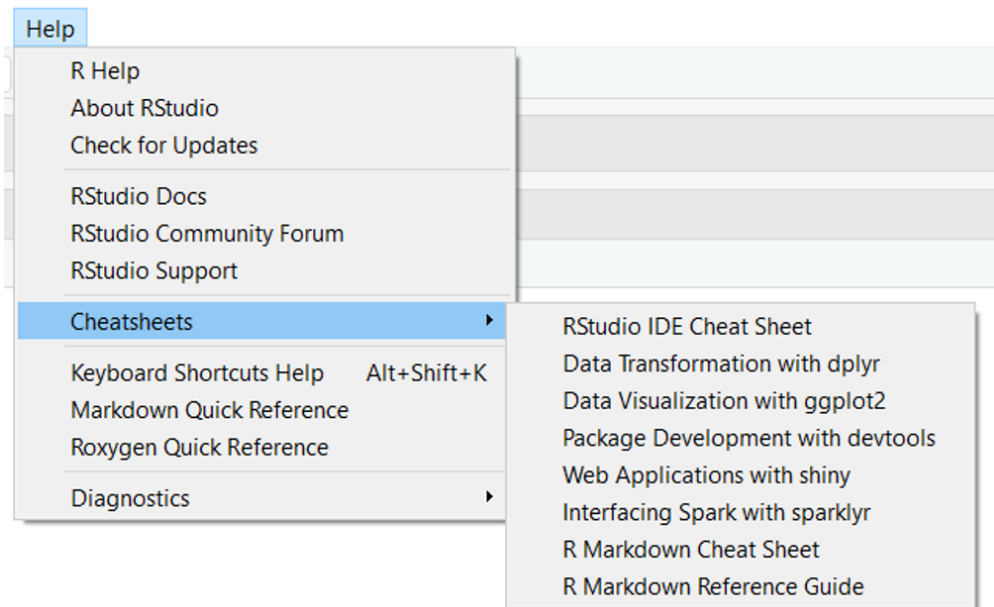


Figure 1: A list of Cheatsheets

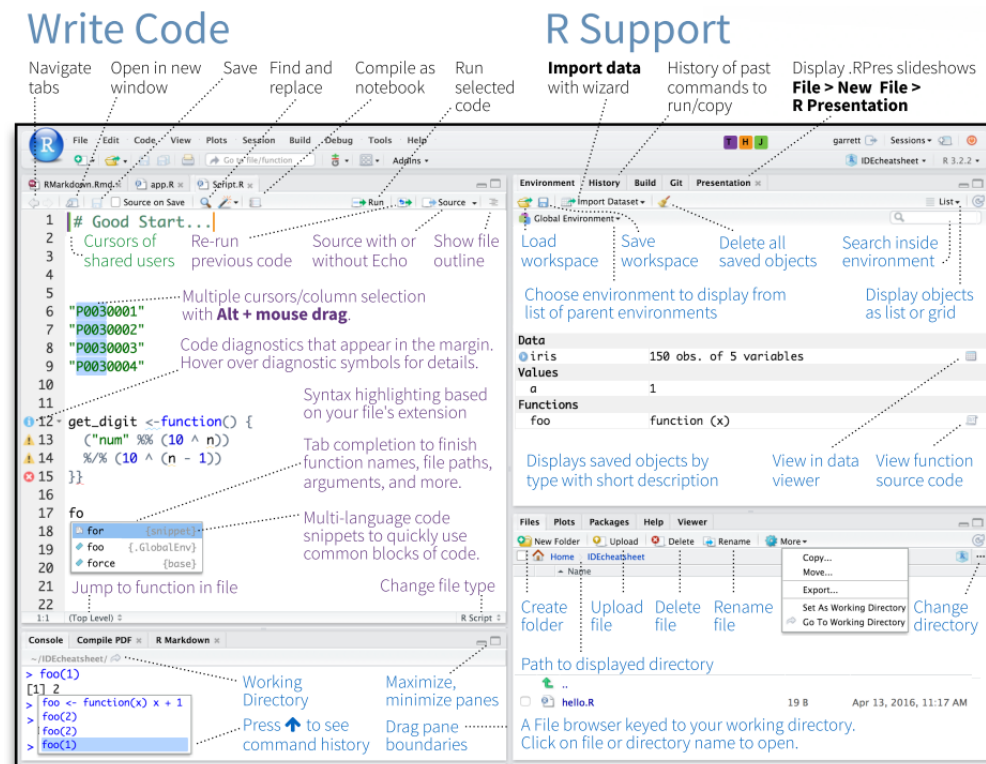


Figure 2: RStudio IDE in Cheat Sheet

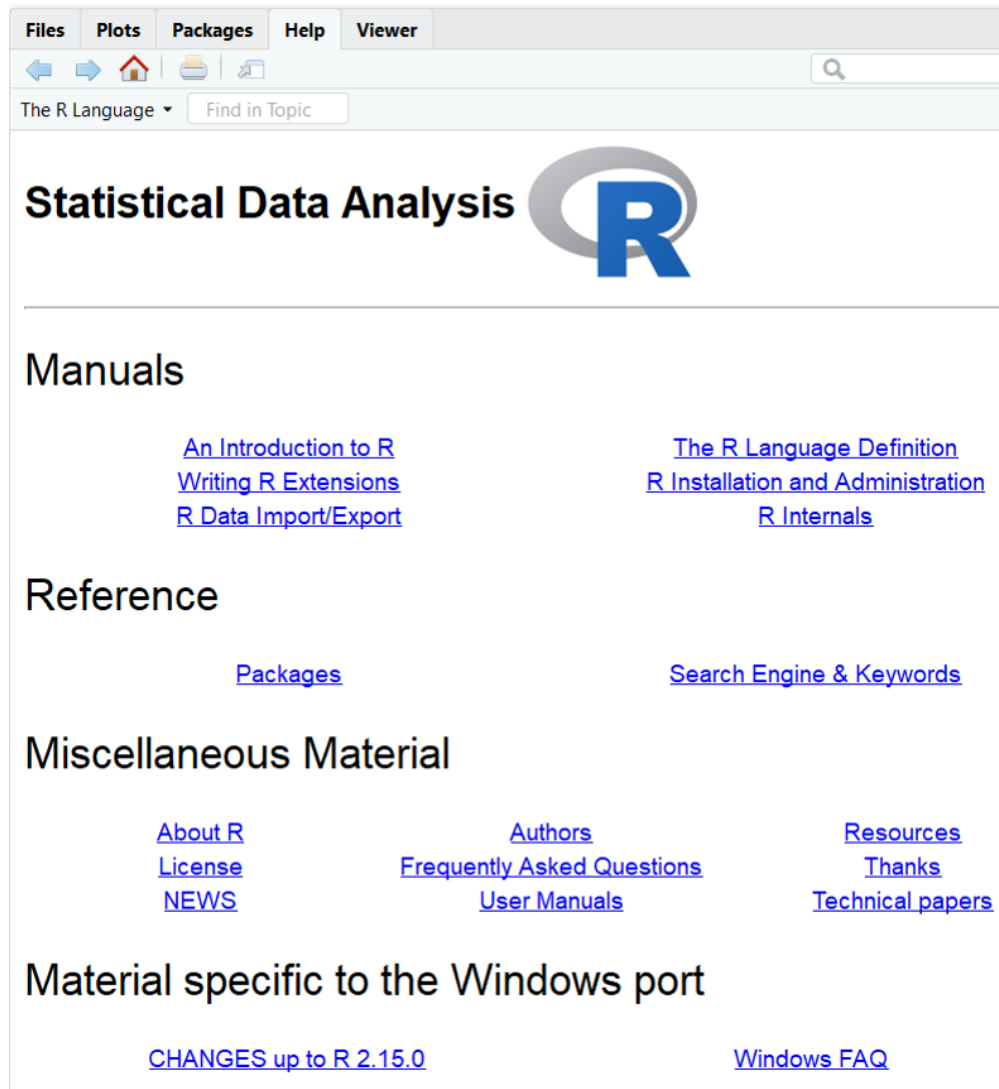


Figure 3: Using help in RStudio

5 Using Help in RStudio

5.1 Start Help

RStudio provides a lot of free resources as we can find in *help* tab in RStudio (see Figure 3). An alternative way to have the help screen is to use the code below in Console window.

```
help.start()
```

An introduction to R under Manuals is a useful guide manual for the beginner to start using R. Figure 4 shows the Table of Contents of the manual. This course will look at the essential operations and concepts of R, which are covered in this manual.

You are recommended to read this manual to understand more functions of R. This course will not cover all functions in R but most conventional ones, as there are vast amounts of R functions. After this course guiding your entry to R, you should be the independent learner to further explore R by yourself.

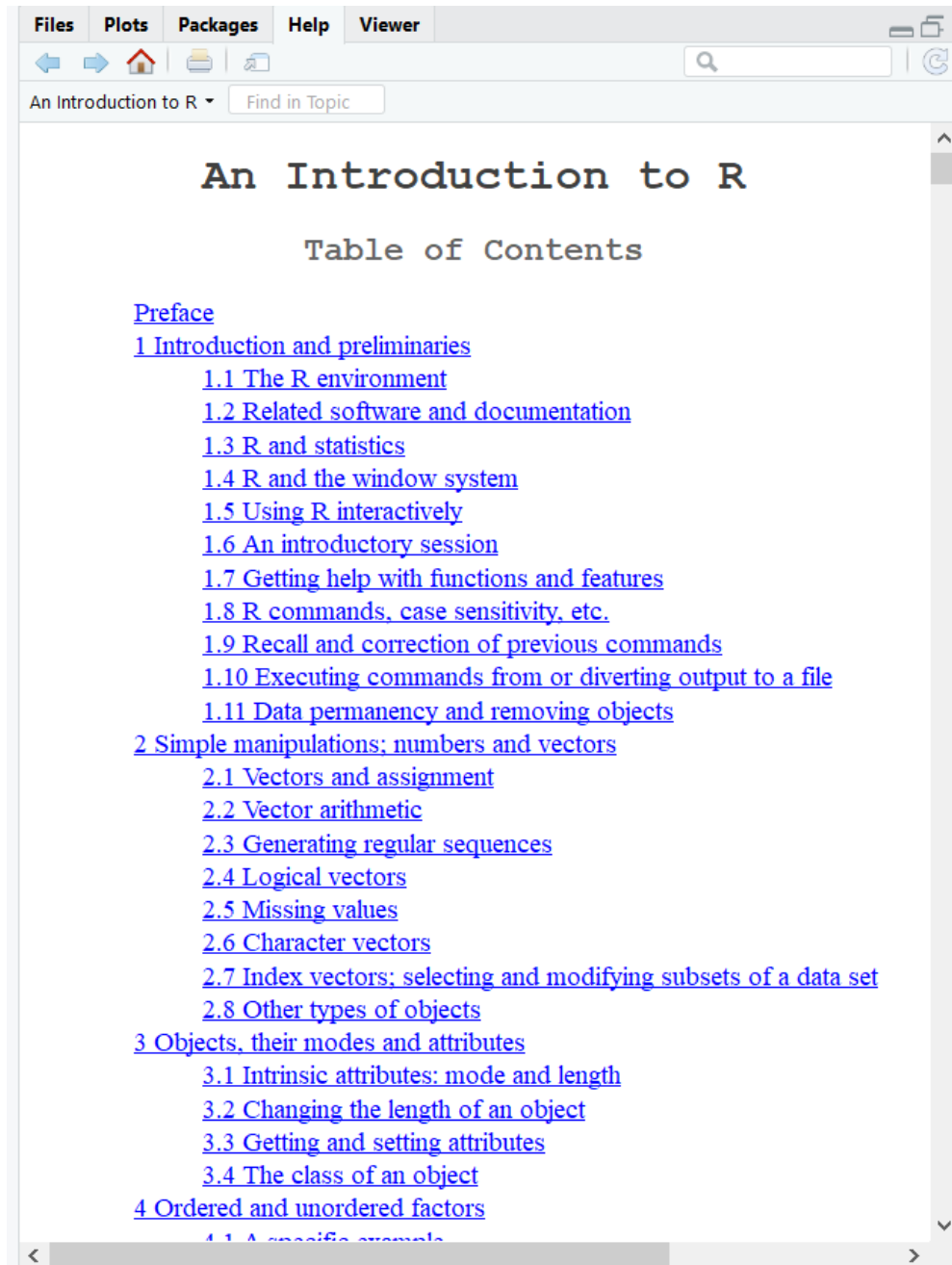


Figure 4: An Introduction to R

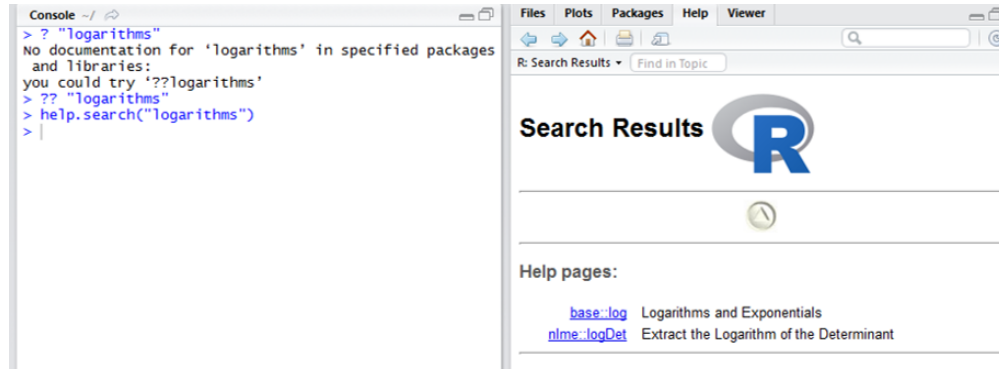


Figure 5: An Introduction to R

5.2 Search in Help

The `help(function.name)` or `?function.name` is used to find the document on how to use the function. The `help.search("topic")` or `?? "topic"` is used to find the documents including the search keywords.

For example, if you would like to understand how to use “log” function in R, there are several ways which you can find the information. The simplest way is to search “log” in the search box in “help”. The system will search the related document to show the details for you (Figure 5). Alternatively, we can use one of the commands below to the same result.

```
# 3 alternative ways to call help
help(log)
help("log")    # the double quotes " " means the content is string
?log
```

The `log()` function is the build-in function in R. The methods above are used to search the known function name. However, the statement below:

```
? "logarithms"
```

The system recommends to try `?? "logarithms"`, which will return to any functions including the query string. Alternatively, `help.search("logarithms")` produces the same result as `??` (Figure 5).

```
?? "logarithms"
```

6 External Learning resources

There are a lot of good resources to learn R. Some free learning resources are recommended as below.

1. <https://rpubs.com/>
2. <https://rstudio-education.github.io/hopr/index.html>
3. <https://r4ds.had.co.nz/index.html>
4. <https://bookdown.org/>

7 Basic Arithmetic Operations

Now we start to perform the basic operations in R. RStudio is used as the interactive environment. R Console is used as a calculator to perform the basic calculation shown as below.

```
# "#" means comments. Anything after # is not processed by R.
#Addition
5+6

#multiplication
8*9

#division
7/8

# subtraction
3-10    #3 minus 10 leaves -7

# 9 to the 11th power
9^11

#Obeying order of operations
#Using parentheses to group expressions
5*(10-3) / (71*65)
```

Statements are separated by either a semi-colon (;), or a newline. Comments can be put almost anywhere, and everything from after a hash mark (#) to the end of the line is a comment.

In summary, R contains a number of operators which are listed as below (R Core Team, 2019).

Notation	Explanation
-	Minus, can be unary or binary
+	Plus, can be unary or binary
!	Unary not
~	Tilde, used for model formulae, can be either unary or binary
?	Help
:	Sequence, binary (in model formulae: interaction)
*	Multiplication, binary
/	Division, binary
^	Exponentiation, binary
%%	Modulus, binary
%/%	Integer divide, binary
%*%	Matrix product, binary
%o%	Outer product, binary
%in%	Matching operator, binary (in model formulae: nesting)
<	Less than, binary
>	Greater than, binary
==	Equal to, binary
>=	Greater than or equal to, binary
<=	Less than or equal to, binary
&	And, binary, vectorized
&&	And, binary, not vectorized
	Or, binary, vectorized

Notation	Explanation
	Or, binary, not vectorized
<-	Left assignment, binary
->	Right assignment, binary
\$	List subset, binary

8 Working Directory and Creating Project

8.1 Working Directory

It is not convenient to type and execute commands one by one, as a solution quite often involves a lot of commands. In this case, we need to work in an R file. We can create an R file in RStudio (Figure 1.18).

1. In RStudio, click File -> New File -> R Script.
2. After a blank file is opened, click File -> Save.
3. Choose the folder for the files to saved.

When we would like to save files in the same folder, the system will have the default directory (or folder). We can use *getwd()* to display the current working folder

```
getwd()
```

The Console will return the display of our Working Directory. Sometimes, if we would like to change the working directory for the current working folder, we can use the *setwd()* function as below.

```
setwd("D:\\yourfolder1\\yourfolder2")
getwd()
```

Please make sure that your folder exists before you use *setwd()*. Otherwise, you will get the error.

In R, "\\" is an escape character. There will be error if we code as below.

```
setwd("D:\yourfolder1\yourfolder2")
getwd()
```

Alternatively, we can use "/" for the path in R.

```
setwd("D:\\yourfolder1\\yourfolder2")
getwd()
```

8.2 Creating Project

If we close and reopen RStudio, we run *getwd()* and find that R will use default folder again. To deal with this, one of the methods is to create an R project.

1. File -> New Project, and then Fig. 6 is shown.
2. Follow the instruction to set up.

After creating a project, all files within the project has the same working directory.

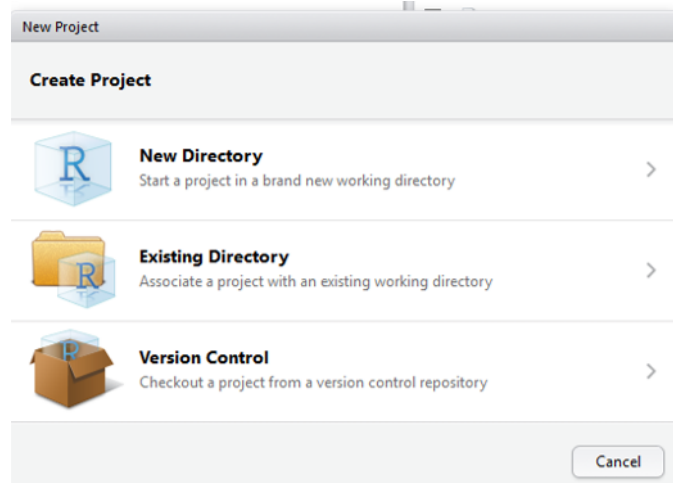


Figure 6: Creating R Project wizard

9 Assignment and Data Object

Assignment means that a variable is assigned to store value(s). Three ways of assignments, `=`, `->`, or `<-`, can be used to assign values to variables. For example,

```
a = 256
256 -> a
a <- 256
```

The assignments of the three ways above are the same subject to developer's preference. We may observe that the R console does not display the results for the assignments above. Again we have three coding ways to display the results as below.

```
# use print() to print the variable
print(a)

#use parentheses for the whole form
(256 -> a)

#use ;
256 -> a;a
```

By writing a programme, we need to create different names for the variables subject to the naming rules as below:

- letters, numbers, a dot (.) , and a underline(_) can be used.
- The first letter of a variable name must start with a dot . and a letter.
- Letters for the variable names are case sensitive. For example, the R programme will consider speed and sPeed as two different variables.

For example,

```
walk.speed = 6.5 # km/hour
(walk_time = 2) #hours

walk_distance = walk.speed * walk_time
walk_distance
```

The value in the variable is quite often needed to be updated. For example,

```
# set and print value for x
(x=1000)

# calculate and print new value of x = current value of x (old) +3
(x=x+3)

# If we continue to do this
(x=x-33)

(x=x*33)
```

(...) print the result of variable assigned.

Apart from numbers, values can be other type of data. We can use `typeof` to check the data type of a variable, such as “logical”, “integer”, “double”, “complex”, “character”, and “NULL”. Let’s evaluate the statement below.

```
(a1= "Hello World")
typeof(a1)

(a2= "Z")
typeof(a2)

(a3= TRUE)
typeof(a3)

(a4= "139")
typeof(a4)

(a5= 139)
typeof(a5)

(t6 =4.5)
typeof(t6)

(t7 = 5i+6)
typeof(t7)

(t8=NULL)
typeof(NULL)
```

We will learn more data type in the future. In the R example above, if we try `t4 +3`, there will be an error message due to the inconsistency of data type.

```
#recall a4
a4="123"

#what will happen?
a4+30
```

As `typeof(t4)` shows that `t4` is a character type, instead of double, we need to convert the data type to a numerical value. To convert a data type, we normally use the function of the format like `as.targetdatatype()`. We can use the help function to search the right information. In this case, we can use `as.double()` as shown as below.

```
# assign value
a4 = "139"

# check class and type
typeof(a4)
class(a4)

#convert to double type
da4 =as.double(da4)

# check new class and type
typeof(da4)
class(da4)

#print value
da4

# check calculation
da4 + 30
```

When a variable object is created, it is stored in the computer memory. We can use `ls()` to list the variables in the memory of R environment. Alternatively, we can use Environment GUI tab window in RStudio to browse the objects.

```
ls()
```

10 Constants

R has a small number of built-in constants: `LETTERS`, `letters`, `month.abb`, `month.name`, and `pi`.

LETTERS: the 26 upper-case letters of the Roman alphabet.

```
LETTERS
```

letters: the 26 lower-case letters of the Roman alphabet.

```
letters
```

month.abb: the three-letter abbreviations for the English month names.

```
month.abb
```

month.name: the English names for the months of the year.

```
month.name
```

pi: the ratio of the circumference of a circle to its diameter.

```
pi
```

We need to pay attention that definition of constant is very different from the conventional programming language like java, C++, and C#. In R, a built-in constant value can be changed in run time. We may think a constant in R is an initial value for a pre-defined variable name.

```
pi  
(pi=pi+2)
```

11 Exercises

11.1 Exercise 1

In Singapore, Amy pays \$1.65 for one liter of diesel. on a visit to Hawaii, she paid 4.48 US dollars for one gallon of diesel.

USD 1 US = SGD 1.242 1 gallon = 3.785 liters

Is diesel cheaper in Singapore or Hawaii? Solve this by R code.

(The problem is revised from question 12 of O level exam paper: 4048/01/O/N/16)

11.2 Exercise 2

A circle area A is calculated by

$$A = \pi r^2$$

A Perimeter of circle is calculated by

$$P = 2\pi r$$

if the radius r is 7 cm, what are the values of A and P? show your results by R code.

11.3 Exercise 3

1. Which platform(s) below can we use R for development?

- a. Microsoft Windows
- b. Android
- c. MacOS
- d. Linux

2. R contains a series of open-sources data in a package named “datasets”. What is the R command to find the link to view the summary of the datasets?

- a. ?? “dataset”
- b. ? “datasets”
- c. library(help = “datasets”)
- d. ? “dataset”

3. Using R to calculate, what is the result for $\sqrt{\frac{5 \times (10-3)^3}{71 \times 65}}$?

- a. 0.008973469
- b. 39.62411
- c. 0.1858072
- d. 0.6096018

4. Supposed there are some R files in a subfolder called *code* under folder called *kevin-yuen*. We would like to set the current working environment to refer to these files together. Which comment(s) is/are correct?

- a. `getwd(“D:/kevin-yuen/Code”)`
- b. `setwd(“D:/kevin-yuen/Code”)`
- c. `setwd(“D:\kevin-yuen\Code”)`
- d. `getwd(“D:\kevin-yuen\Code”)`

5. Which way(s) is(are) the correct assignment(s)?

- a. `a = 256`
- b. `256 -> a`
- c. `a <- 256`
- d. `256 =a`

6. Which assignment statement(s) is(are) valid in R?

- a. `i586 = 4 *6`
- b. `2xx2 =3-2`
- c. `a.b= 3*5`
- d. `.a= 9`

7. Which one(s) is(are) the character object(s) for the variable a?

- a. `a=“a3”`
- b. `b3=5; a= b3`
- c. `a=TRUE`
- d. `b=2;a=typeof(typeof(b))`

12 References

R Core Team. 2019. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.

R Core Team (2017b), An Introduction to R, free download at <https://cran.r-project.org/doc/manuals/r-release/R-intro.html>