

Introduction to R & RStudio (R Instructional Worksheet)

November 12, 2017

Contents

1.1 Installing R	1
1.2 Getting Started in R	1
1.3 Help in R	2
1.4 Setting the Working Directory	3
1.5 Saving your Work	3
1.6 Comments in R	4
1.7 Simple Data Entry	4
1.8 Vectors	5
PROBLEMS	7

[1.1 Installing R](#)

Download R using one of the following links – depending on whether you have a PC or a Mac. After you have downloaded R, you can download R Studio. R Studio is an open source integrated development environment (IDE) for R. Both R and R studio are free.

Windows Users:

<http://cran.at.r-project.org/bin/windows/base/>

Mac Users:

<http://cran.r-project.org/bin/macosx/>

R-Studio:

<http://www.rstudio.com/products/rstudio/download/>

[1.2 Getting Started in R](#)

In R Studio, there are 4 main windows. 1. Console Window 2. Environment/History Window 3. Files/Plots/Packages/Help Window 4. Script Window (does not open automatically)

The Console Window is where the code is run. You can type R commands directly into this window, or you can run R code or script (which is just a collection of valid R commands) from a script file. All of your commands and outputs are also displayed in the console.

The Environment/History Window is in the top right of the screen. The Environment tab lists all of the data and values that have

been created and saved. This allows you to see what each variable you have created represents. You can also import data into R directly using this window. The History tab keeps track of all of the code that you have run. You can save this into a file.

The Files/Plots/Packages/Help Window is in the bottom right of the screen. The Files tab shows all of the files on your computer; you can also set working directory (see below for details), create new folders, and copy or move files from the Files tab. You can open data files directly from this window to look at, but they must be imported into R Studio for you to be able to use or analyze the data. The Plot tab is where all of the plots you create are displayed, and where they can be saved into files. The Packages tab is a list of all currently installed packages, the place to update packages, and the place to install new packages. The Help tab displays the help files for all of the commands and packages.

The Script Window is created by clicking File > New File > New Script. An R script can be saved, so you can reopen it to continue working on a project, and it can be sent to another person for them to open and edit.

It is best to organize all code within an R Script. Press <CTRL Enter> to run code from script in the Console Window or use the 'Run' button at the top of the Script Window.

1.3 Help in R

Introduction to R manual

<http://cran.r-project.org/doc/manuals/R-intro.pdf>

R help mailing list

<http://r.789695.n4.nabble.com/R-help-f789696.html>

R built in help

The help files can be searched in the Help tab in the bottom right corner. In addition, the following commands can be entered into the Console Window to open the specified files (in this case help on the *plot* function)

```
help(plot)
'?'(plot)
'?'('?'(plot))
```

The single question mark ? is used to search for a particular function, the double question mark ?? is used if you do not know the function you want to use, but want to search for potential options.

R built in examples

The example files can be accessed by entering the following command into the Console Window. This will bring up examples of how the function can be used.

```
example(plot)
```

You can also find R help by searching Google!

1.4 Setting the Working Directory

Before beginning work in R/R studio it is important to set the working directory. This is the folder where all files that you are going to access are located, as well as where all files that you create will be saved.

You can set working directory in three ways: using Files tab and using R commands. In the Files tab in the bottom left corner, locate the folder that you want as your working directory. Once you have clicked on the folder and it is open, under 'More' (menu at the top of the Files tab) click 'Set as Working Directory'. An alternative way of achieving the same goal is to use R Studio main menu -> Session -> Set working directory.

You can also set working directory using the setwd (stands for set working directory) command as follows (you need know the full name of the directory or folder):

```
setwd("C:/Statistics/R/")
setwd("C:\\Statistics\\R\\")
```

All slashes needs to be forwards slashes (/) or double backward slashes (\\). Also, R is case sensitive, so make sure capital letters are used only when needed.

1.5 Saving your Work

You can save everything in all of the windows in R Studio (data, R objects etc.) using save.image command.

```
save.image("File.R")
```

This will create a file called File.R in your working directory.

```
load("File.R")
```

This will load your data back so you can keep working where you left off.

This is helpful where you are working with code that takes a long time to run, because you do not have to rerun it every time you want to view your results.

You can also save just the R script or single plots that you have created.

1.6 Comments in R

Comments are notes that you include in your code, but that are not run. R does not run any commands included in your comments.

It is good practice to always put comments into your code, so you or anyone else who access your code later knows what you were doing. This might seem unnecessary for simple code, but will be very useful with more complicated code.

In R, you can comment a line by putting the # sign in front of the text. If you want to comment a large chunk of code all at once, highlight the code and then use the dropdown menu (Code -> Comment/Uncomment Lines).

```
# This is a comment

a = c(1, 2, 3)
# creates a vector, a, containing the numbers
# 1, 2, and 3
sum(a)
# calculates the sum of the numbers in vector
# a
```

1.7 Simple Data Entry

Data can be entered into R at the command line or typed directly into an R script.

As a calculator

```
10 + 5
```

```
## [1] 15
```

Compute a sum

```
sum(1, 5, 10)
```

```
## [1] 16
```

Create a vector of numbers called Numbers

```
Numbers <- c(5, 4, 7)
```

```
Numbers
```

```
## [1] 5 4 7
```

Numbers is the name of the vector containing the numbers 5, 4, 7. To display the contents of the vector type 'Numbers' into the console.

The c used in the vector left stands for 'concatenate' or 'combine'. It is used whenever a list of data points is entered into R.

1.8 Vectors

Vectors are one-dimensional arrays containing numbers, or characters.

```
a <- c(1, 2, 5.3, 6, -2, 4) # numeric vector
```

```
b <- c("one", "two", "three") # character vector
```

```
c <- c(1, "one") #mixed vector
```

Vectors can also be created using sequences of numbers, or repetitions of the same number.

```
5:9
```

```
## [1] 5 6 7 8 9
```

```
seq(5, 9)
```

```
## [1] 5 6 7 8 9
```

```
seq(5, 9, 0.5)
```

```
## [1] 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0
```

```
rep(5, 3)
```

```
## [1] 5 5 5
```

Note: Characters must be entered inside double quotes.

Access the elements of the vectors using their position in the vector in []

```
a[3] #3rd element of vector a
```

```
## [1] 5.3
```

```
a[c(2, 4)] #2nd and 4th elements of vector a [1] 2 6
```

```
## [1] 2 6
```

```
a[3:5] #return the 3rd, 4th, and 5th element of vector a
```

```
## [1] 5.3 6.0 -2.0
```

Add elements to the vector

add a 7th element to the vector

```
a[7] <- 0 #add a 7th element to the vector
```

```
a
```

```
## [1] 1.0 2.0 5.3 6.0 -2.0 4.0 0.0
```

Change elements in the vector

```
a + 1
```

```
## [1] 2.0 3.0 6.3 7.0 -1.0 5.0 1.0
```

Multiply a value to every element of the vector (you can also subtract or divide, etc.)

```
a * 2
```

```
## [1] 2.0 4.0 10.6 12.0 -4.0 8.0 0.0
```

Add two vectors together (of subtract, multiple, divide)

```
b <- 1:7
```

```
a + b
```

```
## [1] 2.0 4.0 8.3 10.0 3.0 10.0 7.0
```

The above commands simply output the desired results – if you want to use the new vector it must be saved as a new variable.

the sum of vectors a and b is now saved as variable c

```
c <- a + b
```

PROBLEMS

1. Open up R Studio and set your working directory to the folder you will be working from (the one that contains the data files you want to use).
2. Save your R script into this folder. Include comments throughout your script to explain what each command is doing.
3. What is one way that you can find help if you forget how to use the *subset* function?
4. Using R as a calculator, what is the sum of 15 and 32?
5. Using R as a calculator, what is the product of 34 and 543?
6. Using R as a calculator, what is the value of 2^4 ?
7. Create a vector of the numbers 3, 8, 4, 2 and give it the name *A*.
8. Create a vector of the characters "monkey", "rat", "snake", and give it the name *B*.
9. Create a vector of the sequence of numbers between 89 and 143, and give it the name *C*.
10. Create a vector containing 15 of the number 3 – use the *rep* command, and give it the name *D*.
11. What is the 3rd element in the vector *A*?
12. What are the last two elements of the vector *B*?
13. What is the 4th and 30th element in the vector *C*?
14. What is the sum of all of the elements in the vector *D*?
15. Add the character "bird" to the end of vector *B*.
16. Change the first element in vector *D* to the number 5.
17. Multiple vector *A* by 3 and save the results as vector *A2*.
18. Add 5 to every element of vector *C* and save the results as *C2*.