Statistics with R

Krisna Gupta[1] Donny Pasaribu

2020-08-10

[1] krisna.gupta@anu.edu.au

Contents

Chapter 1

Some Introduction

1.1 About this book

1.1.1 What is this book and why

Learning a statistical software can be a daunting experience for everyone. However, as the importance of data analysis is increasing, we think more talent in this sector will only benefit everyone.

This book is created using the Bookdown (Xie, 2020) package and is written in Rmarkdown (Allaire et al., 2020) language. All content in this book is free to be used and distributed by anyone for learning purpose. Please credit us if you find this book to be useful.

This book handles a mainstream curriculum of undergrad statistics, therefore it is best to use this book while also doing an undergrad statistics course. We use social sciences data set to illustrate some commands in R, but it can be easily changed to data set relevant to yourself.

1.1.2 Pre-requisite

This book is intended to anyone who are interested in learning statistics using R language. It is assumed that people has zero experience with using any programing language at all. Experience with spreadsheet program such as Microsoft Excel or Google Sheet is certainly helpful but not needed.

This book will not cover a lot of the theoretical part of statistics. Readers are assumed to understand that already. A little bit of intro will be given but that is pretty much it. Readers are expected to understand already the theory behind these techniques, or are expected to learn it from elsewhere.

1.1.3 Learning outcome

Information in this book would help you to:

- install and update R and Rstudio with ease
- input, edit, and manage dataset
- visualise dataset as needed
- conducting hypothesis testing
- running a regression
- write a report using the results from above

1.2 About R

1.2.1 What is R

R is an open source statistical package useful for doing some statistical stuff, while RStudio is a software which improve greatly our user interface when using R compared to using R GUI. RStudio also can be used to do other stuff but let's discuss that for later.

R can be used to manipulate a huge amount of data, as well as running a handful of statistical analysis such as regression which is essentially just a matrix algebra in the background. Use of command instead of point-and-click is great because it means it can be replicable with just a script (and a bit of other stuff). Great for collaboration and peer-reviewing.

1.2.2 Good reasons to start with R

R is a bit behind in popularity compared to a more general language such as python or javascript. However, R is generally used among researchers, so there are many people in academics setting use R. R is a bit more specialised on statistical purpose. If you are learning statistics, it is easier to learn using r compared to more general programming language. If you want to learn other language later, it is easier to start with R.

Here's a more comprehensive reasons on why you should learn R from University of Chicago.

1.2.3 Computer requirements

R and RStudio is really light you can install them in your microwave. R and RStudio supports Windows, Mac and Linux. Also, good internet connection never hurts.

1.2.4 Installing R and RStudio

You can easily google to get your R and RStudio installer. But we will give you links anyway. Get your R installation here if you are windows user, here for Mac users, and here for ubuntu users. Get your RStudio from this links.

We are using recommended setting when we install R to our machine, and we suggest you do too.

1.3 About the authors

We are nobody.

Chapter 2

Starting with R

Under Construction

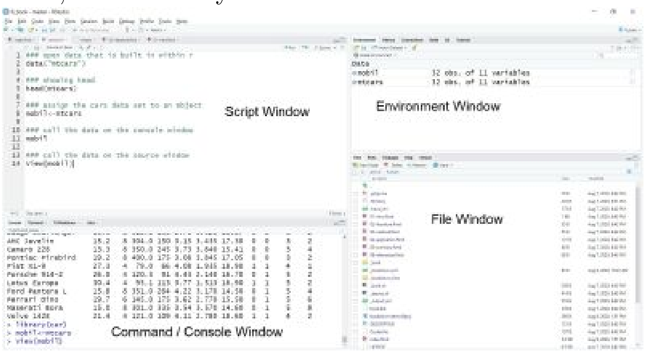
Now that you have R and RStudio installed, let's start playing around with it. It is easy to be scared with command-type type of language, but do not worry. The first step is always the hardest. You will get there in no time at all.

One thing to note is that you will never have to use R GUI. Just keep RStudio's shortcut in your desktop and everything will be allright.

Now let's get to RStudio

2.1 RStudio's interface

First, familiarise yourself with RStudio interface. RStudio looks like this:



```
### open data that is built in within r
data("mtcars")

### showing head
head(mtcars)

### assign the cars data set to an object
mobil<-mtcars

### call the data on the console window
mobil

### call the data on the source window
View(mobil)
```

```
Data
mobil          32 obs. of 11 variables
mtcars         32 obs. of 11 variables
```

```
AMC Javelin        15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28         13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird   19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Fiat X1-9          27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2      26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa       30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Ferrari Dino       19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
Maserati Bora      15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Volvo 142E         21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
> library(car)
> mobil<-mtcars
> View(mobil)
```

As you can see from the above figure, RStudio have four main windows.

2.1.1 console window

Arguably the most important window in RStudio. We write R command in this window.

2.1.2 script window

Forget what we said about the previous window. This is the most important window in RStudio. We write our sets of command in this window

2.1.3 environment window

You can observe your environment here, such as the data and variables / series you created. This window also have other useful tab such as "history" which can show you the history of what happens in your environment, but we will skip them for now.

2.1.4 files window

Although we call it 'files window', this window consists of 5 tabs. The tab file shows your working directory. When you run a code that generates file such as graphics, it will be stored in this window. Plots tab shows you your latest plot.

Packages tab show you all the package that you have. You can also add new package from this tab instead of using the `install.packages()` command. In this tab, you can also activate package that you have instead of using `library()` command.

"help" tab will show you some information about things that you ask help for when you use `?'command'` command. Will show you how it works in a minute.

Lastly, "viewer" tab. I actually forgot why it exists so whatever.

2.2 using your console window

Generally, you type commands and operators such as `print()` and `1+1` and the likes on the console window. Let's try it a bit.

If you type 1+1 in the console and hit enter, you will get:

```
1+1
```

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

you can print your name in r

```
print('Hello world! My name is Jane Doe')
```

```
## [1] "Hello world! My name is Jane Doe"
```

In short, chunks of code we use in this book is meant to be typed on your console window.

2.3 using R script

However, to be a good coder, you will, most likely, store your chunks of code on the R script.

2.4 package

2.5 Update

You need to keep your R, RStudio and your installed packages updated as much as you can. Here's how to do it using the console window.

Firstly, install a package called `installr` by typing `install.packages('installr')` in your console window. Don't forget to hit enter. You only need to do this once. Next, call it using `library(installr)`. Calling the package will be needed to be done everytime you start a new R session (i.e., everytime you open R).

After you called it with `library()`, type `updateR()` on your console, and let it do its magic.

To keep your packages updated, you only need to click "tools" on the toolbar up there, then "Check for Packages Updates…". Depending on how much packages you have installed on your machine, can take a while.

Keep in mind that you need to stay connected to the internet when you update your R, RStudio and packages.

Chapter 3

Beginner stuff

Under Construction

This chapter contains instruction the most basic and useful commands in R. We also covers data types and why it matters.

3.1 Setting your preamble

Firstly, it is a good idea to start your script with a clearing environment sets of code.

```
# Close all graphics, clear memory and screen
graphics.off(); remove(list=ls());cat("\14");
```

There are three different command in this script. `graphics.off()` clears out the environment from any graphs from previous R session. the command `remove(list=ls())` is used to remove your environment from any data and variables/series that you may have. Lastly, `cat("\14")` clears your console window.

You need to set your working directory

```
setwd('your directory')
```

Chapter 4

Data Management

Under Construction

Chapter 5

Inferences

Under Construction

```
mean(price)
```

```
summary(price)
```

```
var(price)
```

```
sd(price)
```

```
cov(price, bedrooms)
```

```
cor(price, bedrooms)
```

Chapter 6

Visualisation

Under Construction

6.1 Plot command

6.1.1 quick plot

6.2 histogram

6.2.1 quick histogram

6.3 pie chart

6.3.1 quick pie chart

6.4 box plot

6.4.1 quick box plot

6.5 using ggplot2

Chapter 7

Hypothesis Testing

Under Construction

Chapter 8

ANOVA

Under Construction

Chapter 9

Simple Regression

Under Construction

We assume you are familiar with regression or Ordinary Least Square (OLS), but let's make a quick recap on what regression is all about.

9.1 Univariate regression

is a regression between one dependent variable and one independent variable. Most textbooks use a notation Y for dependent variable, and use X as the independent variable. A univariate regression assume this form:

$$Y_i = \alpha + \beta X_i + e_i$$

where Y_i is your observed dependent variable and X_i is your observed independent variable.

There are many versions of the Gauss-Markov assumption, but really there are two main thing we assume for OLS to be unbiased:

- error term e_i have a conditional zero mean, i.e., $E(e_i|X_i) = 0$
- error term e_i is independent and identically distributed

We call α as intercept. We generally don't interpret α although it is very important to include α to avoid bias. We are interested in β, as it shows the general strength of relationship between our X and our Y is. That is, we often say:

> an increase of X by 1 unit, is associated with an increase of Y by β unit, assuming everything else constant.

Mathly, $\beta = \frac{dy}{dx}$

9.1.1 running your univariate regression

suppose you have

9.1.2 plotting your regression

In the previous chapter, we have learned how to plot our data using `plot()`. You can add to your script a line showing your regression result

9.2 Multivariate regression

Multivariate regression is just like your univariate regression, but we have more independent variable than just one. Dependent variable still only one. That is:

$$Y_i = \alpha + \beta X_i + \gamma Z_i + e_i$$

The interpretation is similar.

9.2.1 running multivariate regression

Now, instead of

unfortunately, you can't plot multivariate regression as this regression has more than two dimension.

Bibliography

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., and Iannone, R. (2020). *rmarkdown: Dynamic Documents for R.* R package version 2.3.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown.* R package version 0.20.