Statistics with R

Abdullah Al Mahmud

2020-10-26

Contents

Preface **5**

About Author **7**

1 Getting Started with Statistics and R **9**
1.1 Segment 01 9
1.2 Segemnt 02 14

2 Data Analysis: Base R **19**
2.1 Session 01: Visualization 19
2.2 Session 02: Analysis 21

3 Data Analysis: Introduction to The Tidyverse **23**
3.1 Reading and Manipulating Data 24
3.2 Vizualizing Data 24

4 Advanced Data Warngling with Tidyverse **25**
4.1 Advanced Visualization 26
4.2 Advanced Wrangling 26

5 Modeling **27**
5.1 Regression 27
5.2 Test 27

Preface

This book is a parallel content for the course **Statistics with R**. The book contains codes and output for better understanding how the codes in R work.

If you have any suggestions, feel free to let me know.

About Author

Abdullah Al Mahmud is a lecturer in statistics at PCC.

Chapter 1

Getting Started with Statistics and R

1.1 Segment 01

1.1.1 What is Statistics?

The term has got three different meanings.

- **Plural of the term statistic**, which refers to any function of sample values, for example, $\bar{x} = \frac{\sum_i^n x_i}{n}$
- **Table of values**
- **Technique of dealing with data**
  - Collecting data
  - Organizing
  - Analyzing Interpreting
  - Presenting

Table 1.1: A Subset from Iris Data Set

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

1.1.2 Statistics vs Data Science

Statistics deals mainly with analyzing and interpreting data, while data science deals more with predictive analytics. (more)

1.1.3 Statistics vs Mathematics

Consider the following equations

- $Y = X + 0.2 \times X$

Say, X = Basic Salary of an employee in a company, while Y is computed from X, with adding to X 20% of X. In this scenario, given the values of X, we can always tell what the gross salary would be.

What if we have an equation such as the following:

$Y = X + 0.1 \times R$

Where, R is the revenue the company earns through the employee. In this case, the salary of the employee would vary in each month.

The salaries from month to month would unpredictably vary, which is where statistics comes in. Statistics deals with randomness, situations where we cannot exactly tell which outcome we might get. We may (or may not) know the possible outcomes (like when tossing a coin, we know the possible outcome, but which will happen)

1.1.4 Why R?

R is the most popular programming language for statistical analysis, second most popular for machine learning.

Reasons at a glance

- Free and Open Source Software (FOSS)
- Big Community
- Made by statisticians for statisticians
- Easy to use codes
- Stunning graphics, esp. with *ggplot2*
- Reproducibility

1.1.5 Who Use R?

R is both used in academia and industry.

- Good analyses for theses are now accomplished using R.
- Industries heavily rely on R for statistical analysis, predictive analytics, and machine learning.

Some of the renowned companies using R are:

- Google
- Facebook
- Twitter (Tweet sensitivity analysis)

1.1.6 Who Developed R?

R was developed by Ross Ihaka and Robert Genetleman (MORE)

1.1.7 Other Languages and Packages

Some other languages for data analysis are:

- Python
- Julia
- Java
- Scala

Packages

- SPSS
- STATA
- Eviews

1.1.8 Installing R and Rstudio

☒ Go to Cran [^1]

1.1.9 Start Writing R Code (Windows, Linux, and Command Line)

- Using R Console directly: Not a good idea
- Using Rstudio Console: Equivalent to using R console
- Using R Script from Rstudio: to run, press `Ctrl + Enter`

It is best to use Rstudio.

1.1.10 Effectively Using Rstudio

- **Keep things organized**
- **Make a project** Put all codes, data and output inside that project directory.
- **Use `View` function** to view data tables.

1.1.11 R Script

An R script is a convenient tool to organize a work. A project may consist of several or many such scripts. They can be easily shared with others.

1.1.11.1 Quoting R codes from another R file.

```
source('r_file.R')
```

1.1.12 R Documentation (Help)

To get help, type `?keyword` or `help(keyword)`

For example, `?mean` would show options and examples for the mean function.

1.1.13 Handling Error

- If the code is not run, and shows a `+` sign, it means the code is not complete yet complete it or press `esc` to start over.
- If the erros message shows `could not find function ...`, correct the function name.
- If you do not understand the error message, copy and paste it to your browser search bar, and see what help the community has to offer.

1.1.14 R Packages

R packages are extensions of base R, providing some very useful tasks. Many R packages made R more popular and useful, such as `ggplot2`, `karet`, and `rmarkdown`.

To install a packages, run `install.pckages("package_name")`, for example `install.pckages("tidyverse")` installs the package `tidyverse`. When installing, the package name must be enclosed within quotation marks (" ").

Before being able to make use of a package, one must load the package, by running `library(package_name)`, for example, to load `ggplot2`, run `library(ggplot2)`, this time without quotation marks (" ").

1.1.15 R Mathematical Operations

☐ Make a table: Purpose, code, example, output

1.1.16 Assigning Values

Variables make it easy to assign values and use them later.

- To assign values to variables, you can use either `=` or `<-`, but in R, `<-` is preferred. In Rstudio, pressing `alt + -` is a very good shortcut for correctly typing `<-`.
- Comments start with `hash` (#)

Example

```
x <-  3
y <-  4
x+y
```

```
## [1] 7
```

```
x*y
```

```
## [1] 12
```

```
x+10-y
```

```
## [1] 9
```

```
x^y
```

```
## [1] 81
```

```
x**y
```

```
## [1] 81
```

```
log(x)
```

```
## [1] 1.098612
```

```
round(log(x), 3)
```

```
## [1] 1.099
```

1.1.16.1 Round, Floor, and Ceiling

Suppose, we have a number 3.9856

- `round` rounds the number;

```
x <- 3.9856
round(x,3) # (up to 3 digits)
```

```
## [1] 3.986
```

- `celing` switches the number to the next integer;

```
ceiling(x)
```

```
## [1] 4
```

`floor` gives the previous integer.

```
floor(x)
```

```
## [1] 3
```

☒ `celing` and `floor` always give integer output.

1.1.17 Generating Multiple Numbers

```
x <- 1:10
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
seq(1,20, 2) # Keeping fixed gap between the numbers
```

```
##  [1]  1  3  5  7  9 11 13 15 17 19
```

```
seq(1,50, length.out = 5) # Generating specific amount of numbers.
```

```
## [1]  1.00 13.25 25.50 37.75 50.00
```

1.1.18 Data Types

- Logical
- Numeric
  - integer
  - Double
- Character

1.1.19 Learn More

☒ Stat Mania artciles and link to contents
☒ Books
☒ Coursera, Edx, and other MOOCs.

1.2 Segemnt 02

1.2.1 Vector

A vector is set of similar items. In Linear Algebra, it is defined as a matrix with only one column or one row. It could contain numbers of different types, strings, or logical values.

A vector makes it easy to simultaneously operate on multiple items.

☒ We make a vector when we are dealing with only one variable.
☒ A vector can contain only one type of values, such ac numeric, logical etc.

A vector in `R` is usually made using `c`, which stands for *concatenate*. A vector can also be made using `seq` command shown earlier, or by using a `colon` (`:`) sign, if the values are successive integers.

```
x <- c(4, 5, 7)
a <- 10:12
```

```
y <- c("red", "green", "blue", "black", "orange")
z <- c(TRUE, FALSE, TRUE, TRUE, FALSE)
```

1.2.1.1 Adding Vectors

- If a scalar (a single value) is added to a vector, it would be added to values.
- If two (or more) vectors with equal lengths are added together, corresponding values would be added; the same goes for almost any other mathematical operation (such as subtraction or division).
- If, however, the lengths are unequal, the values of the smaller vector would be repeated from the beginning.

```
x + 3 # Adds 3 all values of x.
```

```
## [1]  7  8 10
```

```
x + a # Corresponding values are added.
```

```
## [1] 14 16 19
```

```
b <- 6:7
x + b # Values of b are repeated.
```

```
## Warning in x + b: longer object length is not a multiple of shorter object
## length
```

```
## [1] 10 12 13
```

1.2.1.2 Indexing Vectors

- Using `[]`:

```
x
```

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

```
x[2] # Extracts the second value.
```

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

```
x[2:3] # Extracts second through third values.
```

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

```
x[c(1,3)] # Extracts the first and third values.
```

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

```
x[-1] # Extracts all except the first value.
```

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

```
x[-c(1,3)] # Extracts all except the first and third values.
```

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

- Using `Logical`

```
x[c(TRUE, TRUE, FALSE)] # Extracts the first and second values.
```

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

```
y
```

```
## [1] "red"    "green"  "blue"   "black"  "orange"
```

```
z
```

```
## [1]  TRUE FALSE  TRUE  TRUE FALSE
```

```
y[z] # Using variables already stored. Does not extract values corresponding to FA
```

```
## [1] "red"   "blue"  "black"
```

1.2.1.3 Changing Value(s) of A Vector

1.2.1.4 Sorting

1.2.2 Matrix

A matrix a rectangular array of similar items. Although it has more than two rows and columns, it can only contain items of a single type.

Contents from Jafar Sir

1.2.3 Data Frame

A Data frame contains many variables; each variable can be different type. Distinct variables are placed in columns and values/observations are in rows.

Example

1.2.3.1 Making A New Data Frame

`data.frame` command is used to produce a data frame.

☒ Length of each variable must be equal.

```
df <- data.frame(x=c(10, 12, 15),
                 y=c("Dhaka", "Cumilla", "Rajshahi"),
                 w=sample(100, 3),
                 v=20:22)
```

Table 1.2: A Subset from mtcars Data Set

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160.0	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875
Datsun 710	22.8	4	108.0	93	3.85	2.320
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440
Valiant	18.1	6	225.0	105	2.76	3.460
Duster 360	14.3	8	360.0	245	3.21	3.570
Merc 240D	24.4	4	146.7	62	3.69	3.190
Merc 230	22.8	4	140.8	95	3.92	3.150
Merc 280	19.2	6	167.6	123	3.92	3.440

Table 1.3: An Example Data Frame

x	y	w	v
10	Dhaka	1	20
12	Cumilla	85	21
15	Rajshahi	32	22

1.2.3.2 Indexing A Data Frame

☒ Methods used for matrices apply.

```
df[2,3] # Extracts value from the third column in the second row.
```

```
## [1] 85
```

1.2.4 List

A list can contain scalars, vectors, matrices, data frames, as well as other lists!

1.2.5 Functions

A function is used to

- avoid repetitive tasks and mistakes therefrom
- find values from a complicated formula

A function to compute Harmonic Mean (HM)

Formula: Reciprocal of Mean of $\frac{1}{x_i}$

Reciprocal of $\frac{\frac{1}{x_1}+\frac{1}{x_2}+\ldots+\frac{1}{x_n}}{n}$

Thus, $HM = \frac{n}{\sum \frac{1}{x_i}} = \frac{1}{\text{Mean of } 1/\text{x}}$

```
hm <- function(x) {
  1/sum(1/x)
}
```

We have, x = 4, 5, 7

Therefore,

```
hm(x)
```

```
## [1] 1.686747
```

Since this function is actually a one-liner, we can write it as

```
hm <- function(x) 1/sum(1/x)
```

1.2.6 Loops (Alternatives and Comparison with Other Languages)

In R, loops are rarely used.

1.2.6.1 For loop example

A for loop to add numbers 1 through 10.

```
sum <- 0
for (i in 1:10){
  sum <- sum + i
}
sum
```

```
## [1] 55
```

Values to loop through can also be called from a variable.

```
x <- c(10, 12, 8, 19, 23, 25)
sum <- 0
for (i in x){
  sum <- sum + i
}
sum
```

```
## [1] 97
```

Contents from Jafar Sir

1.2.7 Apply family (apply, lapply, sapply, etc.).

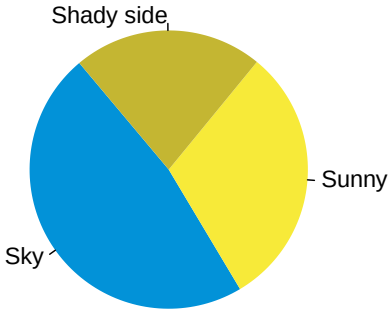
Chapter 2

Data Analysis: Base R

2.1 Session 01: Visualization

2.1.1 Correlation Plot

```
pie(
     c(280, 180, 130),
     c('Sky', 'Sunny', 'Shady side'),
     col = c('#0292D8', '#F7EA39', '#C4B632'),
     init.angle = 130, border = NA )
```


Shady side
Sunny
Sky

2.1.3 Bar Chart

2.1.4 Chart Characteristics (color, title, axes etc.)

2.1.5 How to Use Proper Legends?

2.1.6 Histogram

2.1.7 Ogive (and how to interpret it)

2.1.8 Boxplot

2.1.9 Time Series Plots/Line Chart

2.1.10 Scatter Plot

2.1.11 Equation and Curves

2.1.12 Love Equation and Curve

2.1.13 Different Ways of Coloring Plots

2.1.13.1 RColorBrewer

2.1.14 Wordcloud

2.1.15 Comparison of Suitability of Plots.

2.2 Session 02: Analysis

Measures of Central Tendency and Dispersion (Averages, Quartiles, Variance, etc.); Correlation;

Chapter 3

Data Analysis: Introduction to The Tidyverse

3.1 Reading and Manipulating Data

3.1.1 Reading Data from Different Files

3.1.2 Concept of Tidy Data

3.1.3 How to Make Data Tidy?

3.1.4 Subsetting/Filtering Data

3.1.5 Pipe Operator

3.1.6 Transforming Data

3.1.7 Summarizing Data

3.1.8 Selecting Rows and Columns

3.2 Vizualizing Data

3.2.1 How ggplot2 works

3.2.2 Different geoms and aesthetics

3.2.3 Scatter Plot

3.2.4 Bar Chart

3.2.5 Histogram

3.2.6 Boxplot

3.2.7 Time Series Plots/Line Chart

3.2.8 Pie Chart

3.2.9 Trends within Plots

3.2.10 Piping Output to ggplot2

3.2.11 Piping Output to Plots

Chapter 4

Advanced Data Warngling with Tidyverse

4.1 Advanced Visualization

4.1.1 Correlogram

4.1.2 Themes and Legends

4.1.3 Doughnut Chart

4.1.4 Density Chart

4.1.5 Violin Chart

4.1.6 Bubble Chart

4.1.7 Spider/Radar Chart

4.1.8 Lollipop Chart

4.1.9 Area Chart

4.1.10 Attaching Texts to Plots

4.1.11 Advanced Customization And Attaining What Seems Improbable

4.1.12 Animation

4.2 Advanced Wrangling

Relational Data (e.g, Merging Tables);

Chapter 5

Modeling

5.1 Regression

5.2 Test