Project 1

Rong Zhou

2/21/2022

Command rm(list=ls()) removes all objects from the current workspace (R memory), whereas rm() alone does not do anything.

rm(list=ls())

# Set up the work space

First I want to reduce the number of displayed digits.

options(  
 digits = 4  
)

This project analyzes the data set Grad\_Admission. Variables description are as follows:

ID: Unique identification code for each student

GRE: GRE Scores (out of 340)

TOEFL: TOEFL Scores (out of 120)

Urate: University Rating (out of 5)

SOP: Statement of Purpose Strength (out of 5)

LOR: Letter of Recommendation Strength (out of 5)

CGPA: Undergraduate GPA (out of 4)

Chance:Chance of Admission (ranging from 0 to 100)

# 1. Load the data

In R, there is a special functions for reading a data set into R. For .csv files, you will probably use read.csv().

Use file.path(choose.files()) to get a data set’s file path in Windows.

Grad <- read.csv(  
 file = "C:\\zrwork\\UCF\\2022Spring\\4164\\Exams\\Grad\_Admission.csv",  
 header = TRUE  
)

# 2. Print the dataset (Limit your print to include only the first ten observations)

The head() and tail() functions print the first and last rows of a dataset. This is an easy way to quickly get a feel for your data.

It is not possible to visually inspect large datasets. We work with large datasets by taking snapshots and summaries of the data.

Let’s print the dataset and limit our print to include only the first ten observations.

head(Grad[,-1],10)

## GRE TOEFL Urate SOP LOR CGPA Chance  
## 1 337 118 4 4.5 4.5 3.86 92  
## 2 324 107 4 4.0 4.5 3.55 76  
## 3 316 104 3 3.0 3.5 3.20 72  
## 4 322 110 3 3.5 2.5 3.47 80  
## 5 314 103 2 2.0 3.0 3.28 65  
## 6 330 115 5 4.5 3.0 3.74 90  
## 7 321 109 3 3.0 4.0 3.28 75  
## 8 308 101 2 3.0 4.0 3.16 68  
## 9 302 102 1 2.0 1.5 3.20 50  
## 10 323 108 3 3.5 3.0 3.44 45

Let’s print the dataset and limit our print to include only the last ten observations.

tail(Grad,10)

## ID GRE TOEFL Urate SOP LOR CGPA Chance  
## 391 391 314 102 2 2.0 2.5 3.30 64  
## 392 392 318 106 3 2.0 3.0 3.46 71  
## 393 393 326 112 4 4.0 3.5 3.65 84  
## 394 394 317 104 2 3.0 3.0 3.50 77  
## 395 395 329 111 4 4.5 4.0 3.69 89  
## 396 396 324 110 3 3.5 3.5 3.62 82  
## 397 397 325 107 3 3.0 3.5 3.64 84  
## 398 398 330 116 4 5.0 4.5 3.78 91  
## 399 399 312 103 3 3.5 4.0 3.51 67  
## 400 400 333 117 4 5.0 4.0 3.86 95

# 3.Print a table of the n/mean/standard deviation/min/max of three of the variables (GRE, TOEFL, CGPA).

The summary() function prints summary statistics of a dataset.

summary(  
 object = Grad[c(2,3,7)]  
)

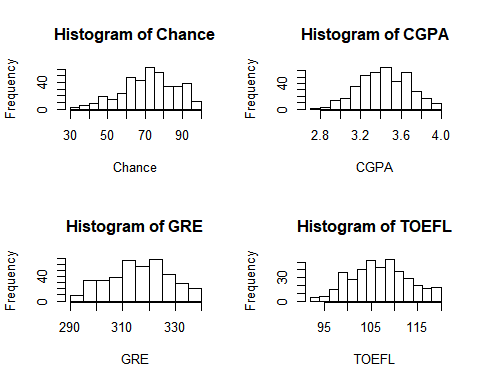
## GRE TOEFL CGPA   
## Min. :290 Min. : 92 Min. :2.72   
## 1st Qu.:308 1st Qu.:103 1st Qu.:3.27   
## Median :317 Median :107 Median :3.44   
## Mean :317 Mean :107 Mean :3.44   
## 3rd Qu.:325 3rd Qu.:112 3rd Qu.:3.62   
## Max. :340 Max. :120 Max. :3.97

# 4.Make histograms for “Chance”, “CGPA”, “GRE” and “TOEFL” variables

When processing data, you need to know how much data you started with and how much you currently have. The dim() function returns the number of rows and columns; nrow() returns the number of rows. ncol() returns the number of columns.

complete.cases() returns a logical vector indicating which rows have zero missing values.

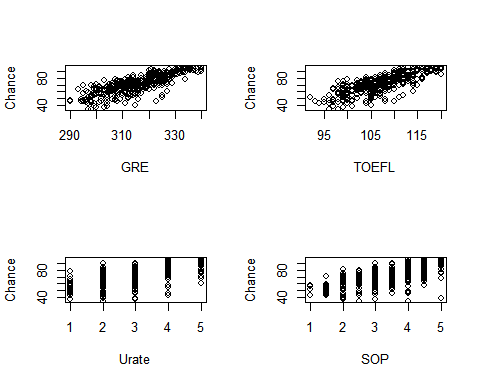
par(mfrow=c(2,2))  
hist(Grad[,8],xlab = "Chance", main = "Histogram of Chance")  
hist(Grad[,7],xlab = "CGPA", main = "Histogram of CGPA")  
hist(Grad[,2],xlab = "GRE", main = "Histogram of GRE")  
hist(Grad[,3],xlab = "TOEFL", main = "Histogram of TOEFL")



It seems that the probability distributions of Chance and CGPA are left skewed and they are not normal. The probability distributions of GRE and TOEFL are approximately normal.

# 5.Consider “Chance” as the response variable and print scatter plots of each of the other six variables against it.

par(  
 mfrow=c(2,2)  
)  
plot(  
 x = Grad$GRE,  
 y = Grad$Chance,  
 xlab="GRE",   
 ylab="Chance"  
)  
plot(  
 x = Grad$TOEFL,  
 y = Grad$Chance,  
 xlab="TOEFL",   
 ylab="Chance"  
)  
plot(  
 x = Grad$Urate,  
 y = Grad$Chance,  
 xlab="Urate",   
 ylab="Chance"  
)  
plot(  
 x = Grad$SOP,  
 y = Grad$Chance,  
 xlab="SOP",   
 ylab="Chance"  
)



plot(  
 x = Grad$LOR,  
 y = Grad$Chance,  
 xlab="LOR",   
 ylab="Chance"  
)  
plot(  
 x = Grad$CGPA,  
 y = Grad$Chance,  
 xlab="CGPA",   
 ylab="Chance"  
)

