CIND 123 - Data Analytics: Basic Methods

Fatemeh Kamyabkalantari

Assignment 1 (10%)

[Fatemeh Kamyabkalantari]

[DM0,501087713]

Instructions

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. Review this website for more details on using R Markdown http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

Use RStudio for this assignment. Complete the assignment by inserting your R code wherever you see the string “#INSERT YOUR ANSWER HERE”.

When you click the **Knit** button, a document (PDF, Word, or HTML format) will be generated that includes both the assignment content as well as the output of any embedded R code chunks.

Submit **both** the rmd and generated output files. Failing to submit both files will be subject to mark deduction.

Sample Question and Solution

Use `seq()` to create the vector $(1, 2, 3, \ldots, 10)$.

```
seq(1,10)
```

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

Question 1

a. Create and print a vector `x` with all integers 1-100, and a vector `y` with every fifth integer in the same range. Hint: use `seq()` function. What is the difference in lengths of the vectors `x` and `y`?.

```
#INSERT YOUR ANSWER HERE
x <- seq (1,100)
x1 <-length(x)
x1
```

```
## [1] 100
```

```
y <- seq(1,100, by=5)
y1 <- length(y)
y1
```

```
## [1] 20
```

```
# The  x length is five times of y length because we use "by" as a step in seq() for  y.
```

b. Create a new vector, “x_square”, with the square of elements at indices 3, 6, 7, 10, 15, 22, 23, 24, and 30 from the variable “x”. Hint: Use indexing rather than a for loop. Calculate the mean and median of the last five values from x_square.

```
#INSERT YOUR ANSWER HERE

x <- seq (1,100)

x_square <- x [c(3,6,7,10,15,22,23,24,30)]^2
x_square
```

```
## [1]   9  36  49 100 225 484 529 576 900
```

```
x_square_mean<- mean(x_square[5:9])
x_square_mean
```

```
## [1] 542.8
```

```
x_square_median<- median(x_square[5:9])
x_square_median
```

```
## [1] 529
```

c. To convert factor to number, would it be correct to use the following commands? Explain your answer.

```
factorVar <- factor(c(1, 6, 5.4, 3.2));as.numeric(factorVar)
```

```
#INSERT YOUR ANSWER HERE

factorVar <- factor(c(1, 6, 5.4, 3.2));as.numeric(factorVar)
```

```
## [1] 1 4 3 2
```

```
# No, factorVar class remains still as "factor" until the new identity is assigned to the old variable.
class(factorVar)
```

```
## [1] "factor"
```

```
#To change its class to numeric, we should add another line to rewrite it.
factorVar <- as.numeric(factorVar)

class(factorVar)
```

```
## [1] "numeric"
```

d. A comma-separated values file `dataset.csv` consists of missing values represented by question marks (`?`) and exclamation mark (`!`). How can you read this type of files in R?

```
#We can use gsub in which we are able to  remove, replace or substitute question marks (`?`) and exclamation mark (`!`) by
 NA or NULL.

#gsub("?", "NULL", dataset)
#gsub("?", "NA", dataset)
#gsub("?". "", dataset)


#gsub("!", "NULL", dataset)
#ogsub("!", "NA", dataset)
#gsub("!". "", dataset)
```

Question 2

a. Compute:

$$\sum_{n=1}^{100} \frac{2^n}{(n-1)!}$$

```
#INSERT YOUR ANSWER HERE
n <- (1:100)
Resultf=sum((2^n)/factorial(n-1))
Resultf
```

```
## [1] 14.77811
```

b. Compute:

$$\sum_{n=1}^{10} \left(\frac{2^n}{n^2} + \frac{n^4}{4^n} \right)$$

```
#INSERT YOUR ANSWER HERE
n <- (1:10)
Results = sum(((2^n)/(n^2))+((n^4)/(4^n)))
Results
```

```
## [1] 35.80589
```

c. Compute:

$$\sum_{n=0}^{10} \frac{1}{(n+1)!}$$

```
#INSERT YOUR ANSWER HERE
n <- (0:10)
Resultfs <- sum(1/factorial(n+1))
Resultfs
```

```
## [1] 1.718282
```

d. Compute:

$$\prod_{n=3}^{33}\left(3n+\frac{3}{\sqrt[3]{n}}\right)$$

```
#INSERT YOUR ANSWER HERE
n <- (3:33)
Resultp <- prod((3*n)+(3/n^1/3))
Resultp
```

```
## [1] 3.025499e+51
```

e. Explain the output of this R-command: `c(0:5)[NA]`

```
#INSERT YOUR ANSWER HERE

R_command <- c(0:5)[NA]
R_command
```

```
## [1] NA NA NA NA NA NA
```

```
# R_command repeats "NA" for 6 times as the intervals of zero to five equals six and  it creates a vector consisted of six N
As.
```

f. What is the difference between is.vector() and is.numeric() functions?

```
#INSERT YOUR ANSWER HERE
#Vector can consist of numeric values or a combination of strings/characters. In case a vector consists of numeric values, both is.vector and is.numeric trigger TRUE. However, if the vector consists of strings, is.vector triggers TRURE and is.numeric triggers FALSE and also, is.numeric() returns True if its argument is a type of double or integer.
```

g. List at least three advantages and three disadvantages of using `RShiny` package?

```
#INSERT YOUR ANSWER HERE
#RShiny is a powerful and open source interactive data visualization Web Framework based on R Language.
#Advantages:1- Rshiny is user-friendly ,2- it easily eliminates long, messy and complicated codes, 3- iRShiny is flexible and compatible with various platforms such as R, CSS, HTML, and Java.
#Disadvantages : 1- poor support,2- not being flexible with distributed computing, and 3- expensive price on enterprise editions.
```

Question 3

`iris` dataset gives the measurements in centimeters of the variables sepal length, sepal width, petal length and petal width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

Install the `iris` dataset on your computer using the command `install.packages("datasets")`. Then, load the `datasets` package into your session using the following command.

```
library(datasets)
```

a. Display the first six rows of the `iris` data set.

```
#INSERT YOUR ANSWER HERE
iris[1:6,]
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

b. Compute the average of the first four variables (Sepal.Length, Sepal.Width, Petal.Length and Petal.Width) using `sapply()` function.

Hint: You might need to consider removing the `NA` values, otherwise the average will not be computed.

```
#INSERT YOUR ANSWER HERE
Average <- sapply(iris[,-5], mean, rm=NA)
Average
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

c. Show how to use R to replace the missing values in this dataset with plausible ones.

```
#INSERT YOUR ANSWER HERE

iris[is.na(iris)] <- 0
```

d. Compute the standard deviation for only the first and the third variables (Sepal.Length and Petal.Length)

```
#INSERT YOUR ANSWER HERE
SepalLengthStd <- sd(iris[,1])
SepalLengthStd
```

```
## [1] 0.8280661
```

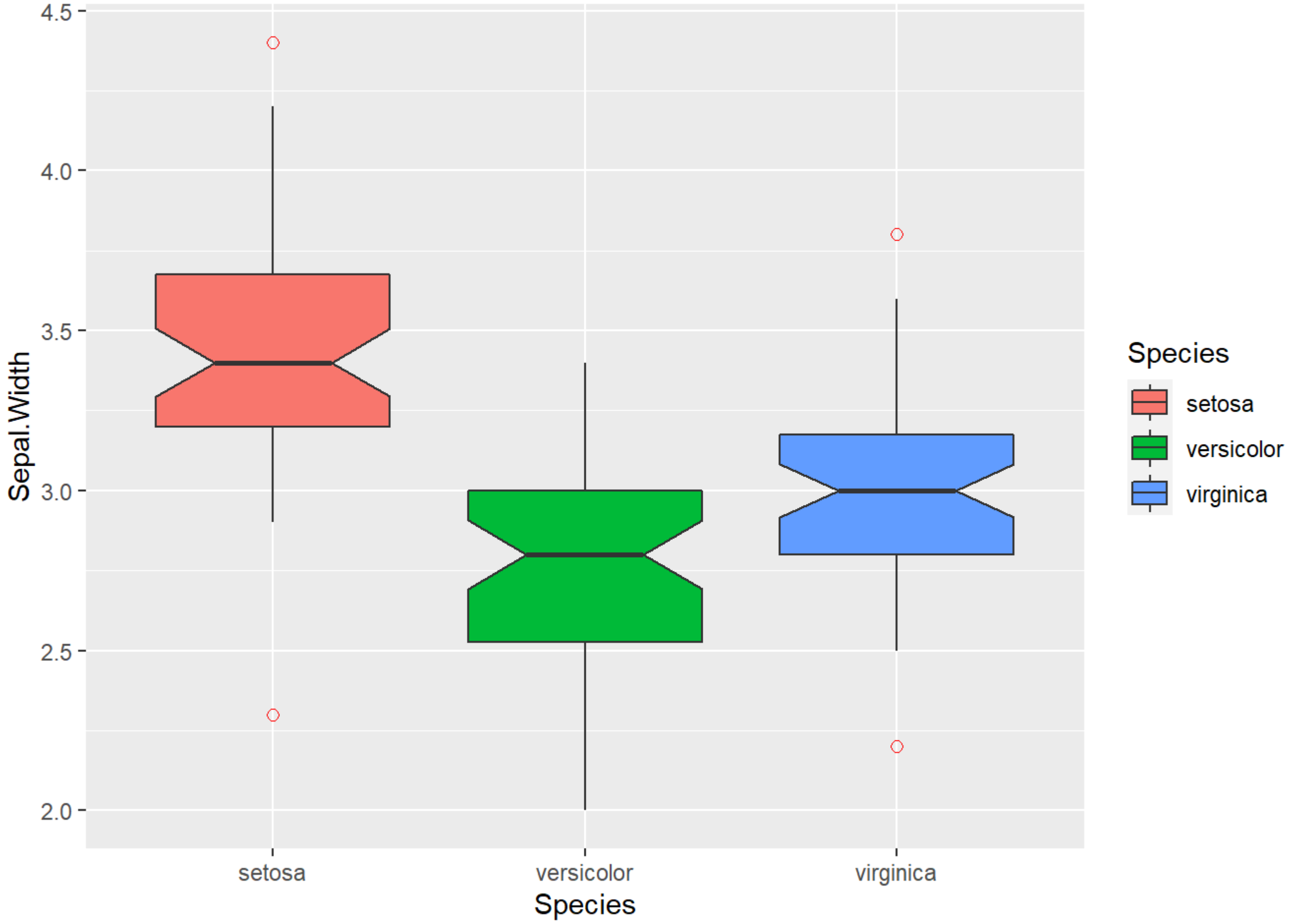
```
PetalLenqthStd <- sd(iris[,3])
PetalLenqthStd
```

```
## [1] 1.765298
```

e. Construct a boxplot for `Sepal.Width` variable, then display the values of all the outliers. Explain how these outliers have been calculated.

```
#INSERT YOUR ANSWER HERE
library(ggplot2)

ggplot(iris, aes(Species, Sepal.Width, fill=Species))+
  geom_boxplot(outlier.color = "red",
               outlier.size = 2,
               outlier.shape =1 , notch = TRUE)
```



```
# A boxplot helps to visualize a quantitative variable by displaying five common location summary (minimum, median, first an
d third quartiles and maximum) and any observation that was classified as a suspected outlier using the interquartile range
 (IQR) criterion. The IQR criterion means that all observations above q0.75 + 1.5 · IQR or below q0.25 − 1.5 · IQR (where q0.
25 and q0.75 correspond to first and third quartile respectively) are considered as potential outliers by R. In other words,
all observations outside of the following interval will be considered as potential outliers:
#I = [q0.25 − 1.5 · IQR; q0.75 + 1.5 · IQR]
```

f. Compute the upper quartile of the `Sepal.Width` variable with two different methods.

```
#INSERT YOUR ANSWER HERE

#1
UpperQuartile <- quantile(iris$Sepal.Width,.75)
UpperQuartile
```

```
## 75%
## 3.3
```

```
#2
summary(iris$Sepal.Width)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.800   3.000   3.057   3.300   4.400
```

g. Construct a pie chart to describe the species with ‘Sepal.Length’ less than 7 centimeters.

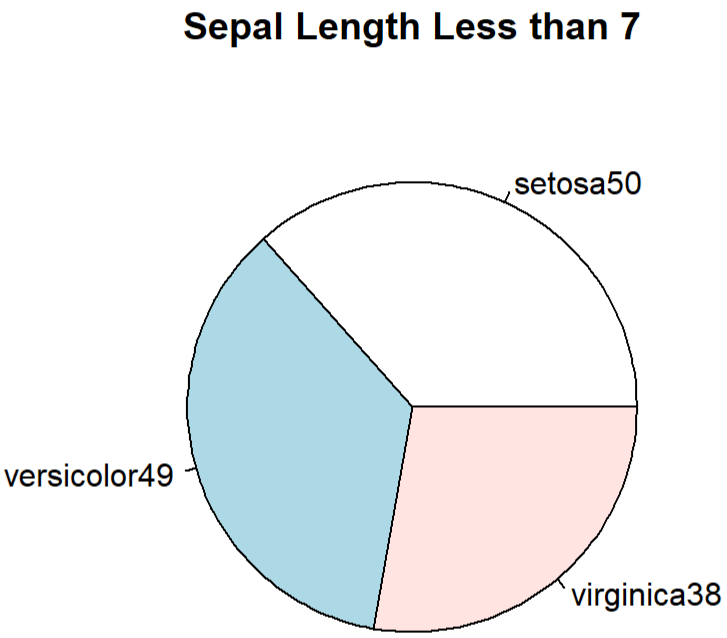
```
#INSERT YOUR ANSWER HERE

SevenLengthSepal <- subset(iris,iris$Sepal.Length < 7)

SevenLengthSepalTable <- table(SevenLengthSepal$Species)

SevenLengthSepalLable <- paste(names(SevenLengthSepalTable),SevenLengthSepalTable, sep ="")

pie(SevenLengthSepalTable,SevenLengthSepalLable,
    main="Sepal Length Less than 7")
```



END of Assignment #1.