Chirag Khandhar | CSP 571 Fall 2019 | **A20438926**

Data Preparation and Analysis Assignment 1

Recitation Exercises:

Exercise 1 Answers:

a. The sample size n is extremely large, and the number of predictors p is small.
   - Because of large sample size, inflexible model will try to fit each data point, thus leading to overfitting. However, **Flexible model will** fit the data closer and thus will **perform better** in this case.

b. The number of predictors p is extremely large, and the number of observations n is small.
   - In this case as the sample size is already small, a flexible model will try to fit each data point, thus leading to overfitting. Hence **Inflexible model will perform better** in this case.

c. The relationship between the predictors and response is highly non-linear.
   - In this case due to non-linearity, we have more degrees of freedom, thus **Flexible model will perform better.**

d. The variance of the error terms, i.e. $\sigma^2 = \text{Var}(\epsilon)$, is extremely high.
   - High variance leads to overfitting and using a flexible model will further try to fit the noise in the error terms and would further increase the variance. Thus, **Inflexible model will perform better** here.

Exercise 2 Answers:

a. **Regression**. Because CEO Salary is continuous.
   Inference. Because we are interested in the factors affecting our target variable (CEO Salary)
   N = 500, p = 3.

b. **Classification.** Because the target variable (Result) is discrete i.e. either Success or Failure.
   Prediction. Because we are only interested in the results (Success/Failure) and not concerned on how each feature contributes in getting a particular result
   N = 20, p = 13

c. **Regression**. Because % change in USD/Euro is continuous.
   Prediction. Because we are only interested in the % change (a particular value) and not concerned on how each feature contributes to that % change.
   N = 52 (52 weeks in 2012), p = 3

Exercise 4 Answers:

a. 3 Real-life applications of classification:
   - Consider student's result data having variables like: result in sem1, result in sem2 and so on till 7th sem. The objective is to find out based on the students previous scores, what would be the result of a student in the final exam. This is also an example of prediction as we are only concerned with the result.
   - Whether a person will cheat or not on giving a loan. Predictors: Credit score, marital status, Taxable Income, Employability. Response: Yes / No. Goal: Prediction
   - Which university to pick. Predictors: University Rankings, Program Ranking, Location, retention rate, average salary, cost of attendance, scholarships. Response: List of universities. Goal Prediction.

b. 3 Real-life applications of regression:

- Mileage of a vehicle. Predictors: Weight, Number of cylinders, Engine type, Fuel type, Power, etc. Based on these parameters we get a numeric value for mileage. Response: Mileage. Goal: Inference.
- Per capita income of an economy. Predictors: GDP, Population, Literacy Rate, Tax Revenue, Inflation rate. Response: Per capita income. Goal: Inference
- % of precipitation. Predictors: Temperature, Air Pressure, Moisture, Humidity, etc. Response: Percentage of rainfall (Precipitation). Goal: Inference

c. 3 Real-life applications of cluster-analysis:

- Recognizing communities within large groups of people. Predictors: Country, 1st Language, 2nd Language.
- Cluster analysis can be used to identify areas where there are greater incidences of types of crime. By identifying these distinct areas, hot spots can be created based on where a similar crime has happened in the past
- Cluster analysis can be used to identify customer buying patterns. Frequent items bought together can be identified.

Exercise 6 Answers:

Parametric Learning	**Non-Parametric Learning**
Reduces the problem of estimating *f* down to one of estimating set of parameters because it assumes a form for *f*.	Does not assume a particular form of *f*, so requires a very large sample to accurately estimate *f*.
$y_i=\beta_0+\beta_1 x_i+e_i$	$y_i=f(x_i)+e_i$
In parametric model, the model to fit the data is known in prior.	In non-parametric model, the data tells, what the 'regression' should look like.

Advantage: The main advantage of a parametric model to either a classifier or a regressor is the simplification of model *f* to a few parameters as not many observations are required as compared to the non-parametric model.

Disadvantage: The disadvantages of a parametric model to either a classifier or a regressor are potentially inaccurate estimate *f*, if the form of *f* assumed is wrong or to overfit the observations if more flexible models are used.

Exercise 7 Answers:

a. $Euclidian\ Distance\ = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$

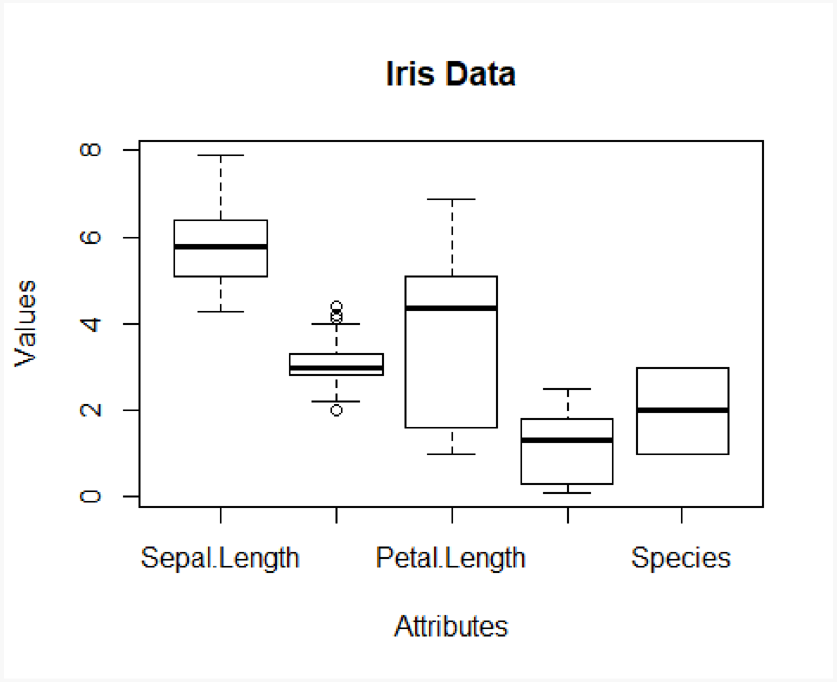
Obs.	X_1	X_2	X_3	***Y***	**Distance**
1	0	3	0	Red	3
2	2	0	0	Red	2
3	0	1	3	Red	3.16
4	0	1	2	Green	2.23
5	-1	0	1	Green	1.41
6	1	1	1	Red	1.73

b. Prediction with *K* = 1:
For K = 1, we choose single nearest point, which in this case is the 5th observation (Distance 1.41). Hence class associated with that observation is Green. Hence for K = 1 our prediction is **Green**.

c. **Prediction with *K* = 3:**
For K = 3, we choose three nearest points, which in this case are Obs. 5 (Distance 1.41), Obs.6 (Distance 1.73) and Obs. 2 (Distance 2). Majority of these observation is class Red. Hence class associated with that observation is Red. Hence for K = 3 our prediction is **Red**.

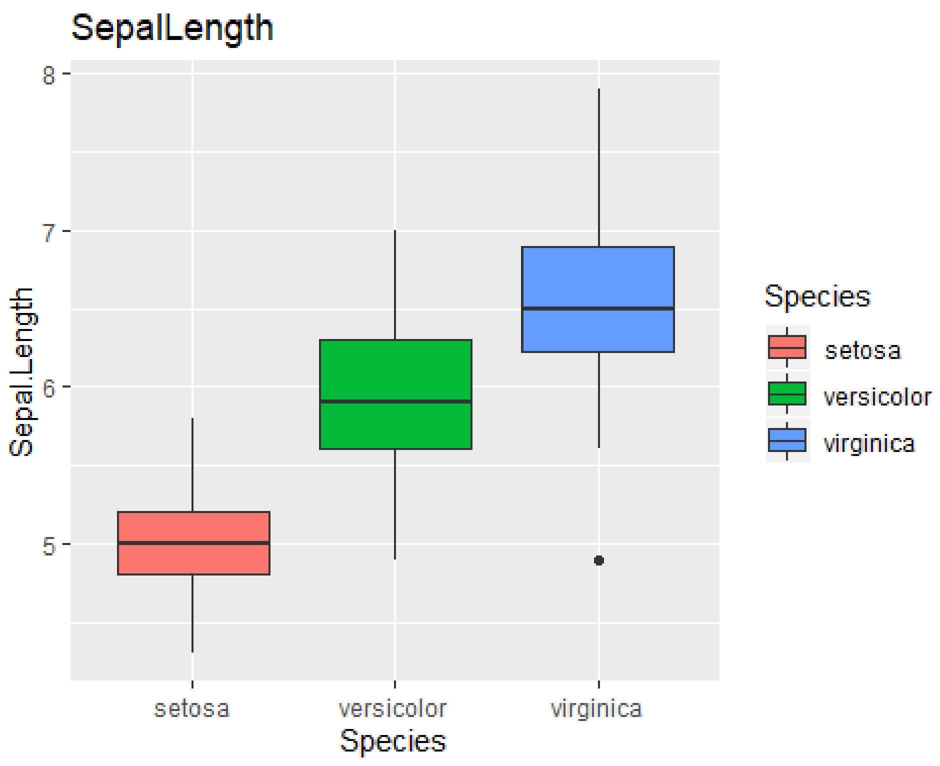
Practicum Problems:

```
#Problem 1

library(datasets)
irisData = data.frame(iris)     #loading iris data set into a dataframe
boxplot(irisData, xlab = "Attributes", ylab = "Values", main = "Iris Data")
```

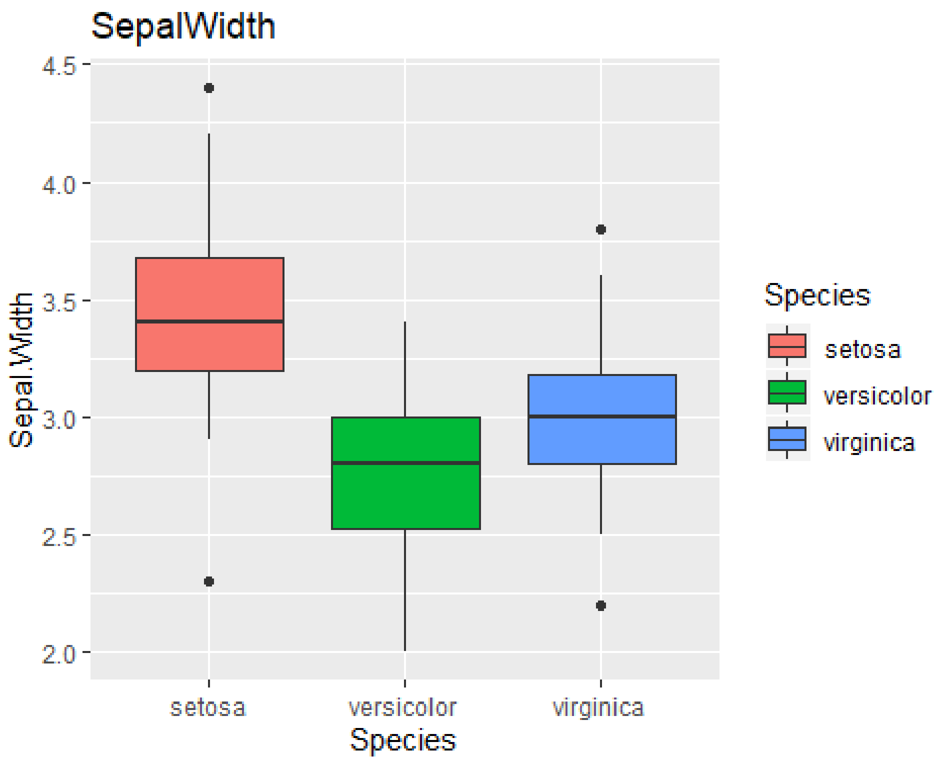


```
cat("\n----------------------------------------------------------------")

##
## ----------------------------------------------------------------

cat("\nIQR of SepalLength: ",IQR(irisData$Sepal.Length))

##
## IQR of SepalLength:  1.3

cat("\nIQR of SepalWidth: ",IQR(irisData$Sepal.Width))

##
## IQR of SepalWidth:  0.5

cat("\nIQR of PetalLength: ",IQR(irisData$Petal.Length))

##
## IQR of PetalLength:  3.5

cat("\nIQR of PetalWidth: ",IQR(irisData$Petal.Width))
```

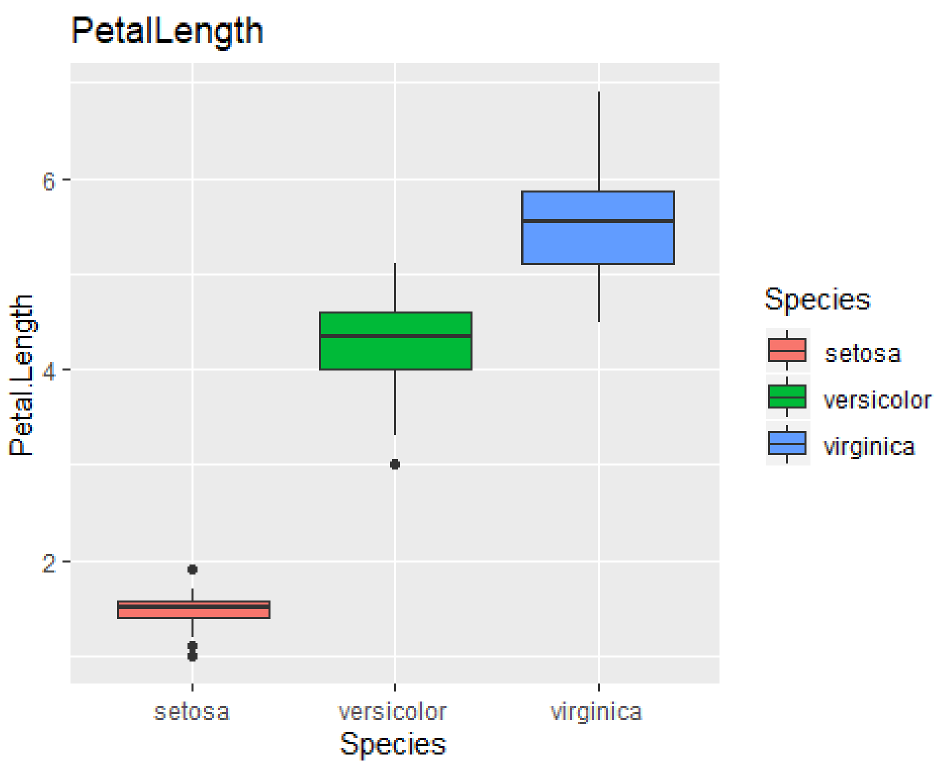
```
##
## IQR of PetalWidth:  1.5

cat("\n-----------------------------------------------------------------")

##
## -----------------------------------------------------------------

cat("\nStandard Deviation of SepalLength: ",sd(irisData$Sepal.Length))

##
## Standard Deviation of SepalLength:  0.8280661

cat("\nStandard Deviation of SepalWidth: ",sd(irisData$Sepal.Width))

##
## Standard Deviation of SepalWidth:  0.4358663

cat("\nStandard Deviation of PetalLength: ",sd(irisData$Petal.Length))

##
## Standard Deviation of PetalLength:  1.765298

cat("\nStandard Deviation of PetalWidth: ",sd(irisData$Petal.Width))

##
## Standard Deviation of PetalWidth:  0.7622377

cat("\n-----------------------------------------------------------------")

##
## -----------------------------------------------------------------

library(ggplot2)

ggplot(data = irisData, aes(x = Species, y = Sepal.Length, fill = Species)) +
geom_boxplot() + ggtitle("SepalLength")
```



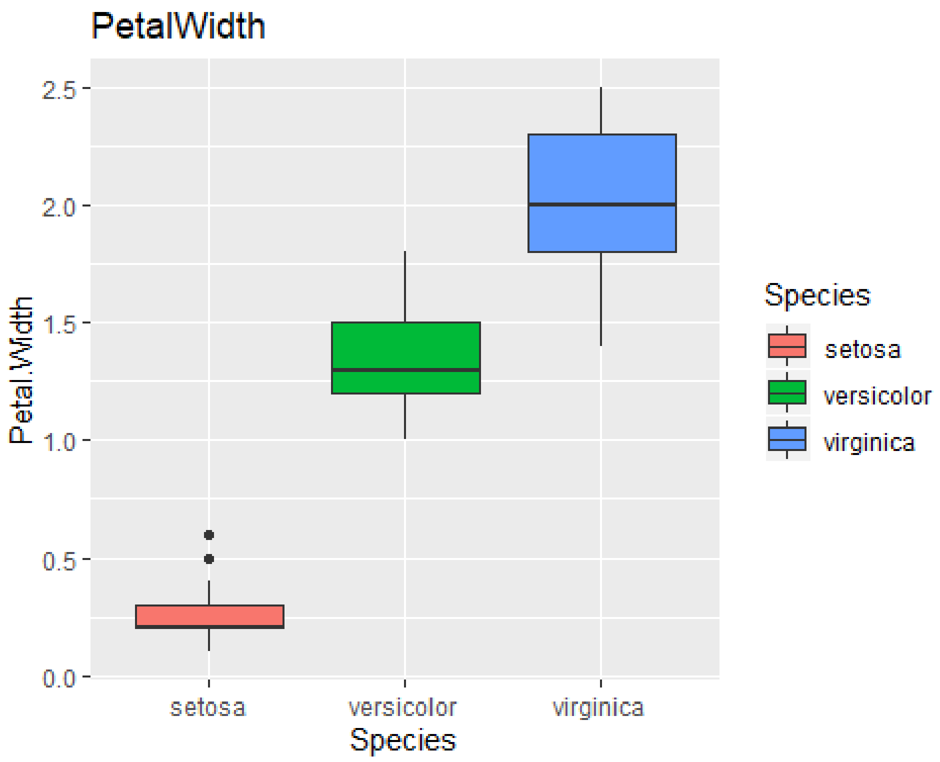
```r
ggplot(data = irisData, aes(x = Species, y = Sepal.Width, fill = Species)) +
geom_boxplot() + ggtitle("SepalWidth")
```



```r
ggplot(data = irisData, aes(x = Species, y = Petal.Length, fill = Species)) +
geom_boxplot() + ggtitle("PetalLength")
```



```
ggplot(data = irisData, aes(x = Species, y = Petal.Width, fill = Species)) +
geom_boxplot() + ggtitle("PetalWidth")
```

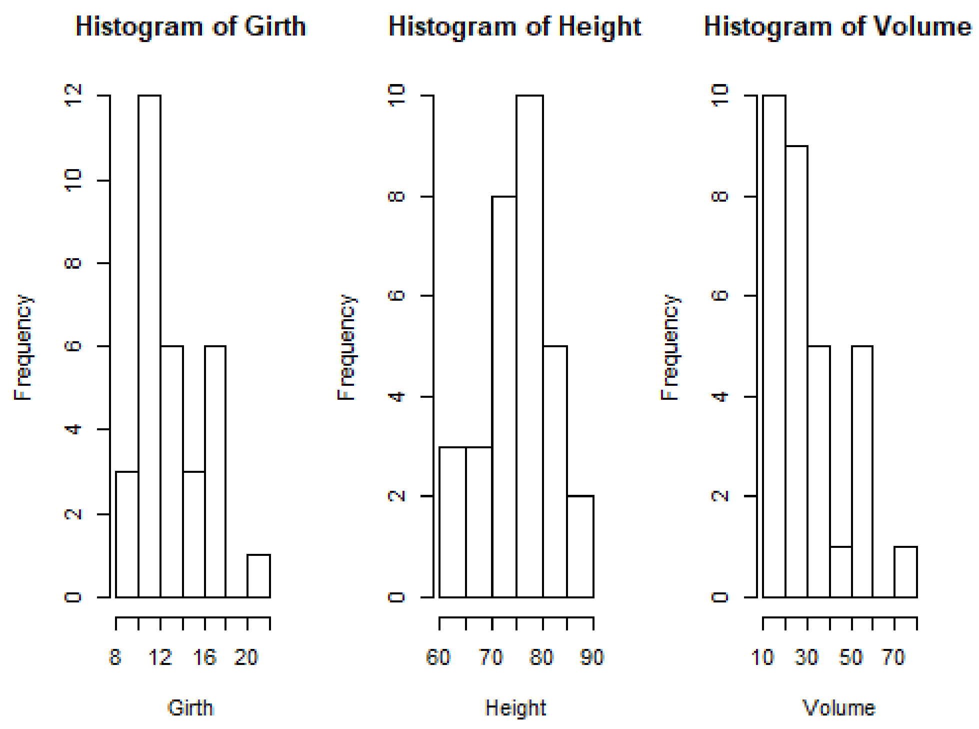


Feature with largest empirical IQR: **Petallength (**`3.5`**)**

Feature with largest Standard Deviation: **Petallength (**`1.765298`**)**

From the above plots of Petallength and Petalwidth, **Setosa Species** comes out to be different from the other classes.

```
#Problem 2

treeData = data.frame(trees)
summary(trees$Girth)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8.30   11.05   12.90   13.25   15.25   20.60

summary(trees$Height)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      63      72      76      76      80      87

summary(trees$Volume)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.20   19.40   24.20   30.17   37.30   77.00

par(mfrow=c(1,3))
hist(trees$Girth,xlab = "Girth", main="Histogram of Girth")
hist(trees$Height,xlab = "Height", main="Histogram of Height")
hist(trees$Volume,xlab = "Volume", main="Histogram of Volume")
```



```
library(moments)
cat("\nSkewness of Girth:",skewness(treeData$Girth))

##
## Skewness of Girth: 0.5263163
```

```
cat("\nSkewness of Height:",skewness(treeData$Height))

##
## Skewness of Height: -0.374869

cat("\nSkewness of Volume:",skewness(treeData$Volume))

##
## Skewness of Volume: 1.064357
```

On visual inspection the **Height** variable seems to look Normally Distributed. Further, after calculating the skewness of each variable below, skewness of Height is closest to 0, which is in line with are earlier inspection for the height variable. Variables **Girth** and **Volume** has **positive skewness** while variable **Height** has **negative skewness**.

```
#Problem 3

mpgData = read.table(url("https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data"),header = F, sep = "", stringsAsFactors = F)
mpgHeader = ("mpg","cylinders","displacement","horsepower","weight","acceleration","model year","origin","car name")
colnames(mpgData) = mpgHeader                          # setting up the headers
mpgData$horsepower = as.numeric(mpgData$horsepower)  # converting factors to numeric type

## Warning: NAs introduced by coercion

cat("\nMean before replacment:",mean(mpgData$horsepower, na.rm = T))

##
## Mean before replacment: 104.4694

mpg_median = median(mpgData$horsepower, na.rm = T)

mpgData$horsepower[is.na(mpgData$horsepower)] = mpg_median # Replacing NAs with median
cat("\nMean after replacment:",mean(mpgData$horsepower, na.rm = T))

##
## Mean after replacment: 104.304
```

After replacing the NA's with median values, we can see that the mean has decreased a little bit from `104.4694` to `104.304`.