**Hierarchical && Non- Hierarchical Clustering**

Instructions:

Please share your answers filled inline in the word document. Submit Python code and R code files wherever applicable.

Please ensure you update all the details:

**Name: Anandakrishnan k v**

**Batch Id: 19042921**

**Topic: Hierarchical Clustering**

**Since both Hierarchical & Non\_hierarchical clustering problem having the same data set. Both are done in single word document, R code, Python code files.**

**1. Business Problem**

* 1. **Objective :**

**Applying different clustering technics and groups the various university according to the available data**

* 1. **Constraints (if any)**

**Maximize: Enhance the performance by minimizinnng the process time**

**Maximize: maximize the accuracy of the analysis results**

**2. Work on each feature of the dataset to create a data dictionary as displayed in the below image:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of feature** | **description** | **Type** | **relavance** |
| **Univ** | **Uviversity name** | **Nominal** | **Irrelevant** |
| **State** | **State name** | **Nominal** | **Irrelevant** |
| **Sat** | **Satscore** | **Quantitative** | **relevant** |
| **Top10** | **Number of times comes under top10** | **Quantitative** | **relevant** |
| **Accept** |  | **Quantitative** | **relevant** |
| **SFratio** | **Students faculty ratio** | **Quantitative** | **relevant** |
| **Expenses** | **Studying expenses** | **Quantitative** | **relevant** |
| **Gradrate** | **Grade rate** | **Quantitative** | **relevant** |

**Using R and Python codes perform**

1. **Data Pre-processing**

**2.1 Data Cleaning, Feature Engineering, etc.**

**Ans:-**

R code begins:-

# Load the dataset

library(readxl)

input <- read\_excel(file.choose())

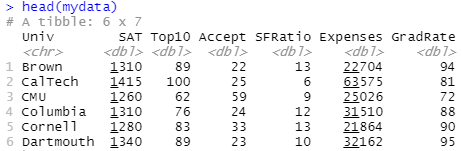
mydata <- input[ , c(1,3:8)]

## DATA CLEANING AND EDA BEGINS

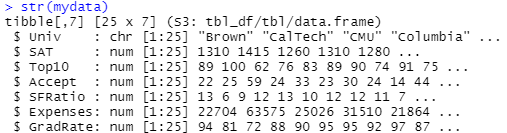
## missing data checking

sum(is.na(mydata)) ## no null values

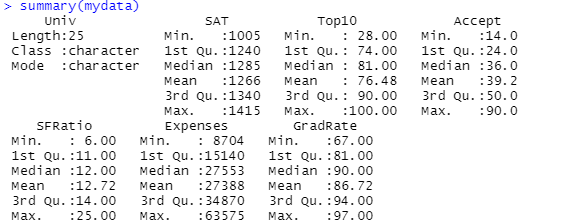
head(mydata)

****

str(mydata)

****

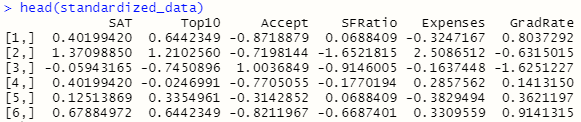
summary(mydata)

****

### Standardize the data ###

standardized\_data <- scale(mydata[, 2:7]) # Excluding the university name

head(standardized\_data)

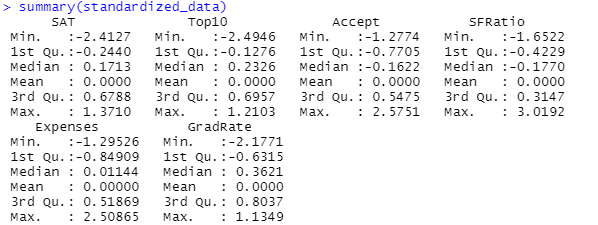
****

**4. Exploratory Data Analysis (EDA):**

**4.1. Summary**

**Ans:-**

summary(standardized\_data)

****

**4.2. Univariate analysis**

**Ans:-**

####### univariate\_analysis ######

univariate\_analysis <- function(variable)

## calling the univariate analysis custom function

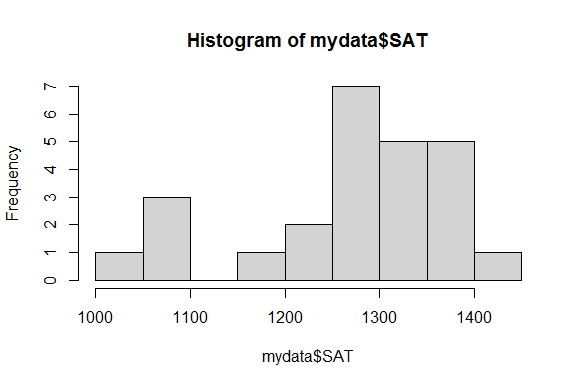
univariate\_analysis(mydata$SAT)

### PLOTS ####

attach(mydata)

## HISTOGRAM

hist(mydata$SAT)

variance= 11741.84 excess kurtosis= -0.182485778776132

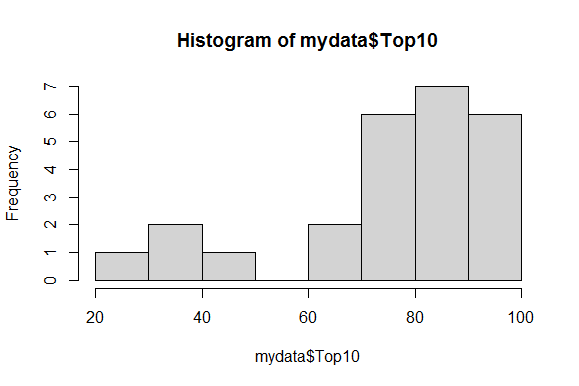
std. deviation= 108.359771133018 outliers= 1081, 1005, 1075, 1085

skewness= -0.83984336376218 number of outliers= 4

## calling the univariate analysis custom function

univariate\_analysis(mydata$Top10)

hist(mydata$Top10)



variance= 377.676666666667 excess kurtosis= 0.0754785695028906

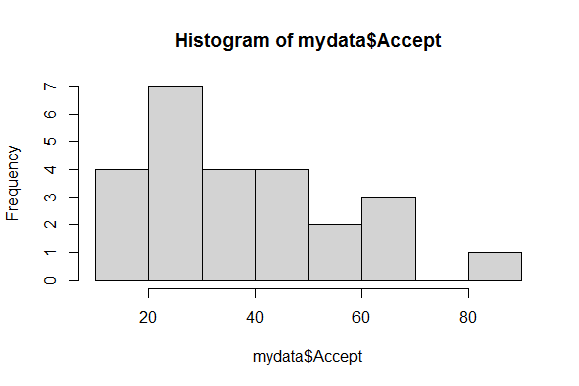
std. deviation= 19.4339050802114 outliers= 38, 28, 49, 40

skewness= -1.07728869215385 number of outliers= 4

## calling the univariate analysis custom function

univariate\_analysis(mydata$Accept)

hist(mydata$Accept)



variance= 389.166666666667 excess kurtosis= -0.258872834338275

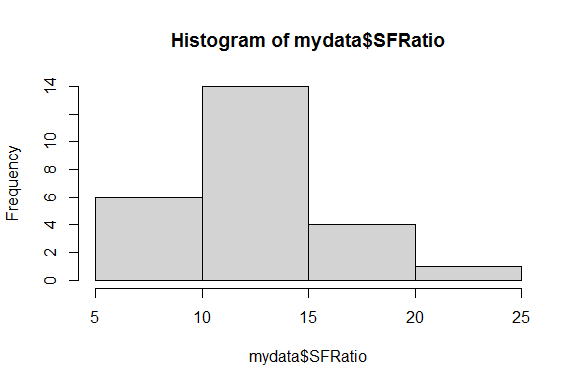
std. deviation= 19.7273076385671 outliers= 90

skewness= 0.76620954408966 number of outliers= 1

## calling the univariate analysis custom function

univariate\_analysis(mydata$SFRatio)

hist(mydata$SFRatio)



variance= 16.5433333333333 excess kurtosis= 1.31112923478999

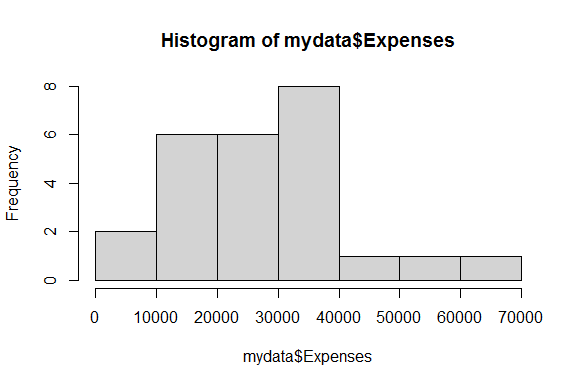
std. deviation= 4.06734966942029 outliers= 6, 19, 25

skewness= 0.989819429359973 number of outliers= 3

## calling the univariate analysis custom function

univariate\_analysis(mydata$Expenses)

hist(mydata$Expenses)



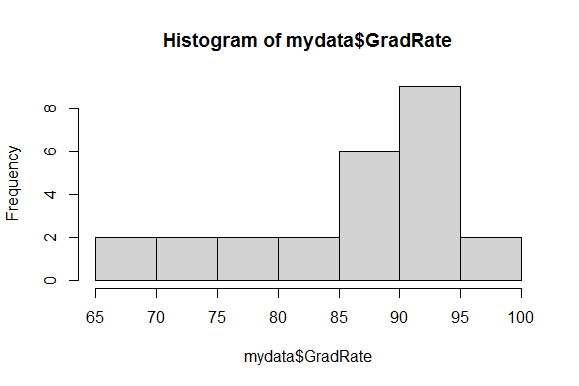
variance= 208077254.333333 excess kurtosis= 0.0290362475731971

std. deviation= 14424.883165327 outliers= no outliers

## calling the univariate analysis custom function

univariate\_analysis(mydata$GradRate)

hist(mydata$GradRate)



variance= 82.0433333333333 excess kurtosis= -0.570990901667379

std. deviation= 9.05777750517937 outliers= no outliers

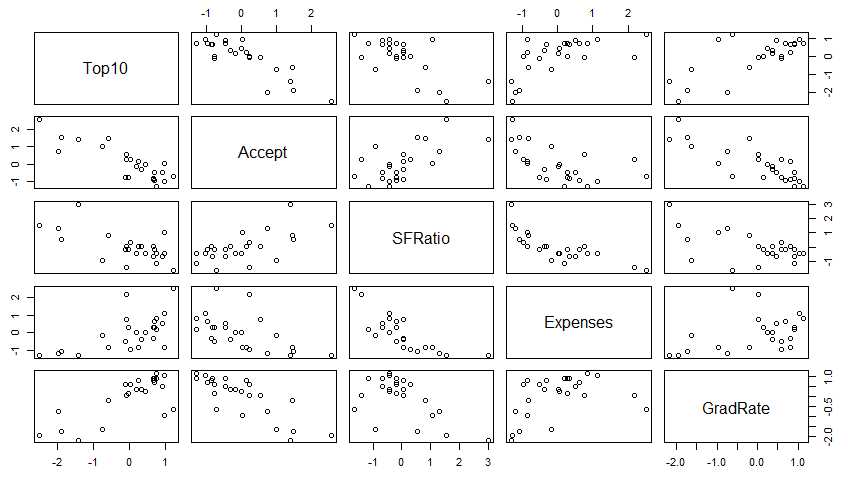
skewness= -0.882777004370466 number of outliers= 0

**4.3. Bivariate analysis**

**Ans:-**

### pair plots of data ###

pairs(standardized\_data[,-c(1)]) ## pair plots of data

****

**5. Model Building**

**5.1 Build the model on the scaled data (try multiple options)**

**Ans:-**

**Not yet covered the portion**

**5.2 Perform the hierarchical & Non\_hierarchical clustering, visualize the clusters using dendrogram & scree-plot**

**i) Perform the hierarchical clustering, visualize the clusters using dendrogram**

**Ans:-**

# Distance matrix of standardized data

d <- dist(standardized\_data, method = "euclidean")

# initiate clustering on distance matrix

fit <- hclust(d, method = "complete")

# Display dendrogram

plot(fit)

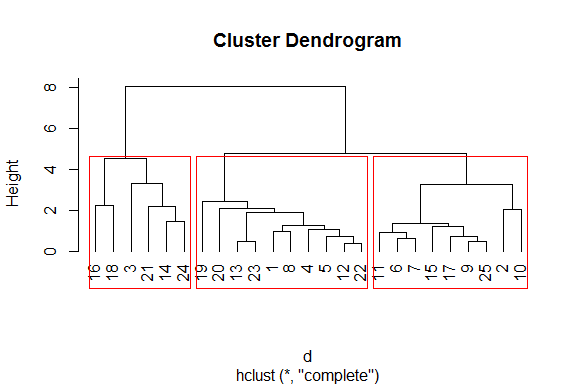
plot(fit, hang = -1)

# cut the total records into 3 clusters according to distance

groups <- cutree(fit, k = 3) # Cut tree into 3 clusters

# draw red outline for each cluster

rect.hclust(fit, k = 3, border = "red")



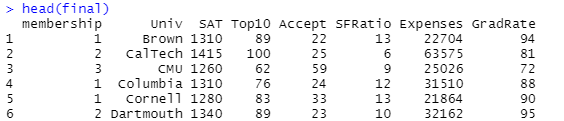
# convert the group data into matrix format

membership <- as.matrix(groups)

# cluster details merge with the original data

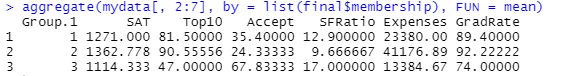
final <- data.frame(membership, mydata)

head(final)

****

# aggregate mean value of each cluster varable

aggregate(mydata[, 2:7], by = list(final$membership), FUN = mean)

****

# generate a .csv file of final clustered data set

library(readr)

write\_csv(final, "hclustoutput.csv")

# save the .csv file

getwd()

**Summary:-**

**1) From the cluster results the universities that’s falls under “Group 2” more better to study. At the same time its seems to be too expensive compare to other 2 groups.**

**2) If we consider the studying expenses the universities that falls under “ Group 1” is more better. For all other feature its stand almost near to “Group 2” , besides that in the case of features like “Accept “ & “ Sfratio” “ Group 1” is superior to that of “Group 2” along with that studying Expenses in “ Group 1” colleges is almost half of that of “Group 2”.**

**3) In every accepts “Group 3” universitie sare falls under the 3rd class except “Accept “ & “ Sfratio” all other features of “Group 3” universities are comes in very low range as that of other 2 groups**

**Python code:-**

## R codes are given all the plots and summery, so simply pasting python code

import pandas as pd

import matplotlib.pylab as plt

Univ1 = pd.read\_excel("G:\\cluster\\hierarchial\_clus\\University\_Clustering.xlsx\\University\_Clustering.xlsx")

Univ1.describe()

Univ1.info()

Univ = Univ1.drop(["State"], axis=1)

### Here the data set is not a mixed data set. Since all the informative datas are numeric applying standardisation scaling here

### standardization scaling ###

#Importing the Libraries

from sklearn.preprocessing import StandardScaler, MinMaxScaler,RobustScaler

# define standard scaler

scaler = StandardScaler() # Standard Scaler or Standardization

# Standardized data frame (considering the numerical part of data)

df\_norm = scaler.fit\_transform(Univ.iloc[:, 1:]) #Fit to data, then transform it.

print("Standardized Scaler :\n",df\_norm)

df\_norm.describe()

# for creating dendrogram

from scipy.cluster.hierarchy import linkage

import scipy.cluster.hierarchy as sch

z = linkage(df\_norm, method = "complete", metric = "euclidean")

# Dendrogram

plt.figure(figsize=(15, 8));plt.title('Hierarchical Clustering Dendrogram');plt.xlabel('Index');plt.ylabel('Distance')

sch.dendrogram(z,

leaf\_rotation = 0, # rotates the x axis labels

leaf\_font\_size = 10 # font size for the x axis labels

)

plt.show()

# Now applying AgglomerativeClustering choosing 3 as clusters from the above dendrogram

from sklearn.cluster import AgglomerativeClustering

h\_complete = AgglomerativeClustering(n\_clusters = 3, linkage = 'complete', affinity = "euclidean").fit(df\_norm)

h\_complete.labels\_

# convert the from array format to dataframe series format

cluster\_labels = pd.Series(h\_complete.labels\_)

Univ['clust'] = cluster\_labels # creating a new column and assigning it to new column

# convert into a dataframe format where first column indicate the cluster details

Univ1 = Univ.iloc[:, [7,0,1,2,3,4,5,6]]

Univ1.head()

# Aggregate mean of each cluster

Univ1.iloc[:, 2:].groupby(Univ1.clust).mean()

# creating a csv file

Univ1.to\_csv("University.csv", encoding = "utf-8")

# save the .csv file in location

import os

os.getcwd()

**ii) Perform the Non\_hierarchical clustering, visualize the clusters using scree-plot**

**R-code:-**

########## Non-hierarchical K-Means clustering ################

install.packages("plyr")

library(plyr)

### Standardize the data ###

standardized\_data

# Elbow curve to decide the k value

twss <- NULL

for (i in 2:8) {

twss <- c(twss, kmeans(standardized\_data, centers = i)$tot.withinss) ## where tot.withinss is a column of kmean function containing TWSS value

}

twss

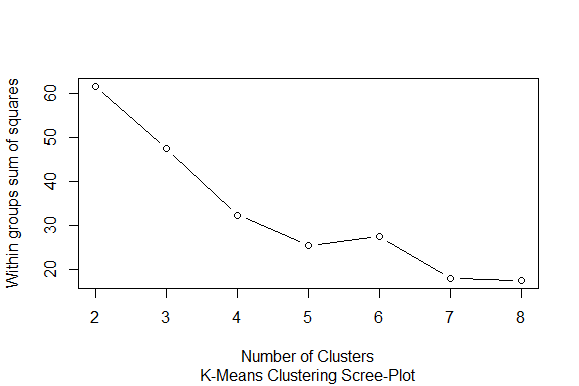
**>twss**

**[1] 61.57947 47.43459 32.30817 25.39704 27.42307 17.87363 17.46716**

# Look for an "elbow" in the scree plot

plot(2:8, twss, type = "b", xlab = "Number of Clusters", ylab = "Within groups sum of squares")

title(sub = "K-Means Clustering Scree-Plot")



**# 3 Cluster Solution**

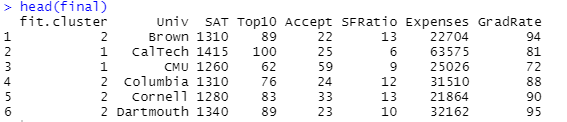
fit <- kmeans(standardized\_data, 3)

str(fit)

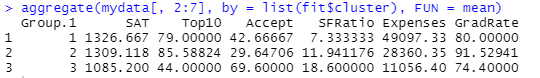
fit$cluster

final <- data.frame(fit$cluster, mydata) # Append cluster membership

head(final)



aggregate(mydata[, 2:7], by = list(fit$cluster), FUN = mean)



**Python-code**:-

##################### non\_hierarchical K- means clustering ###################

import pandas as pd

import numpy as np

import matplotlib.pylab as plt

from sklearn.cluster import KMeans

# from scipy.spatial.distance import cdist

# standardized data frame (considering the numerical part of data)

df\_norm

###### scree plot or elbow curve ############

TWSS = [] # initiate TWSS list

k = list(range(2, 9))

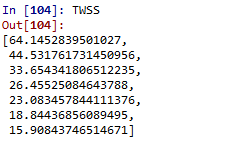
for i in k:

kmeans = KMeans(n\_clusters = i)

kmeans.fit(df\_norm)

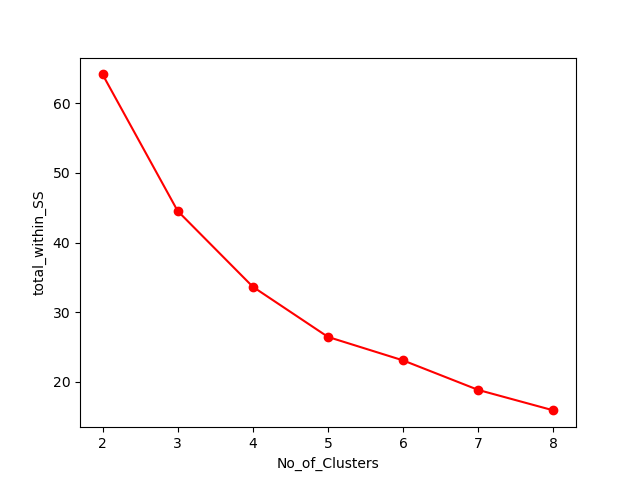
TWSS.append(kmeans.inertia\_) ## appending the value of TWSS for each k\_value

TWSS



# Scree plot

plt.plot(k, TWSS, 'ro-');plt.xlabel("No\_of\_Clusters");plt.ylabel("total\_within\_SS")



# Selecting 3 clusters from the above scree plot which is the optimum number of clusters

model = KMeans(n\_clusters = 3)

model.fit(df\_norm)

model.labels\_ # getting the labels of clusters assigned to each row

mb = pd.Series(model.labels\_) # converting numpy array into pandas series object

Univ['clust'] = mb # creating a new column and assigning it to new column

Univ.head()

df\_norm.head()

Univ = Univ.iloc[:,[7,0,1,2,3,4,5,6]]

Univ.head()

# Aggregate mean of each cluster

Univ.iloc[:, 2:8].groupby(Univ.clust).mean()

Univ.to\_csv("Kmeans\_university.csv", encoding = "utf-8")

import os

os.getcwd()

**5.3 Validate the clusters (try with different no. of clusters) – label the clusters and derive insights (compare the results from multiple approaches)**

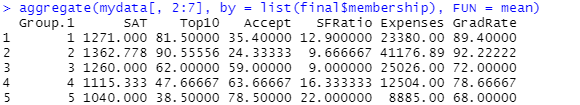
**Ans:-**

**Aggregate mean values of each variables for Cluster size, k= 2**

**G:\cluster\hierarchial_clus\k=2.png**

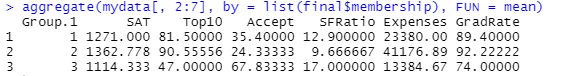
**Except “Accept” & “Sfratio” in every accepts “Group 1” is too much superior while comparing with “Group 2”. But under each group there might have a large number of universities with different feature values. So its difficult to choose the right one while having very less number of clusters.**

**Aggregate mean values of each variables for Cluster size, k= 4**

****

1. **Here “Group 1” & “Group 2” falls under superior class**
2. **All feature values of “Group 1” & “Group 2” are closs to each other.**
3. **Economically forward class students its better to choose “Group 2”**
4. **The studying expenses for “Group 1” universitities are half of that of “Group 2”. And for features like “Accept” & “Sfratio” its comes superior to that of “Group 2”**
5. **All low standared universities are falls under “Group 5”**
6. **As the number of clusters is increases for very small changes in feature values the almost similar records here universities will fal under different clusters. Some times it will be a benefit but some times it affects negatively either.**

**Aggregate mean values of each variables for Cluster size, k= 3**

****

**1) From the cluster results the universities that’s falls under “Group 2” more better to study. At the same time its seems to be too expensive compare to other 2 groups.**

**2) If we consider the studying expenses the universities that falls under “ Group 1” is more better. For all other feature its stand almost near to “Group 2” , besides that in the case of features like “Accept “ & “ Sfratio” “ Group 1” is superior to that of “Group 2” along with that studying Expenses in “ Group 1” colleges is almost half of that of “Group 2”.**

**3) In every accepts “Group 3” universitie sare falls under the 3rd class except “Accept “ & “ Sfratio” all other features of “Group 3” universities are comes in very low range as that of other 2 groups**

**6. Share the benefits/impact of the solution - how or in what way the business (client) gets benefit from the solution provided.**

**It will helps the students to choose the right universities easly from a vast number of universities. Its difficult to go and every universities features and choose the right one by comparing it with others.**

**But if we cluster the universities into different groups according to the feature values it will ease the students choose the right one according their feature preferance. For example if some one want to study the high “Gradrate” universities they caan directly go for (for k=3) universities falls under “Group 2” and from those universities choose the right one having highest “Gradrate”. Its ease the over all searching process.**

**For those students who want to study first class Universities but having less study Expenses. They can search those universities(for k =3) in “Group 1”**

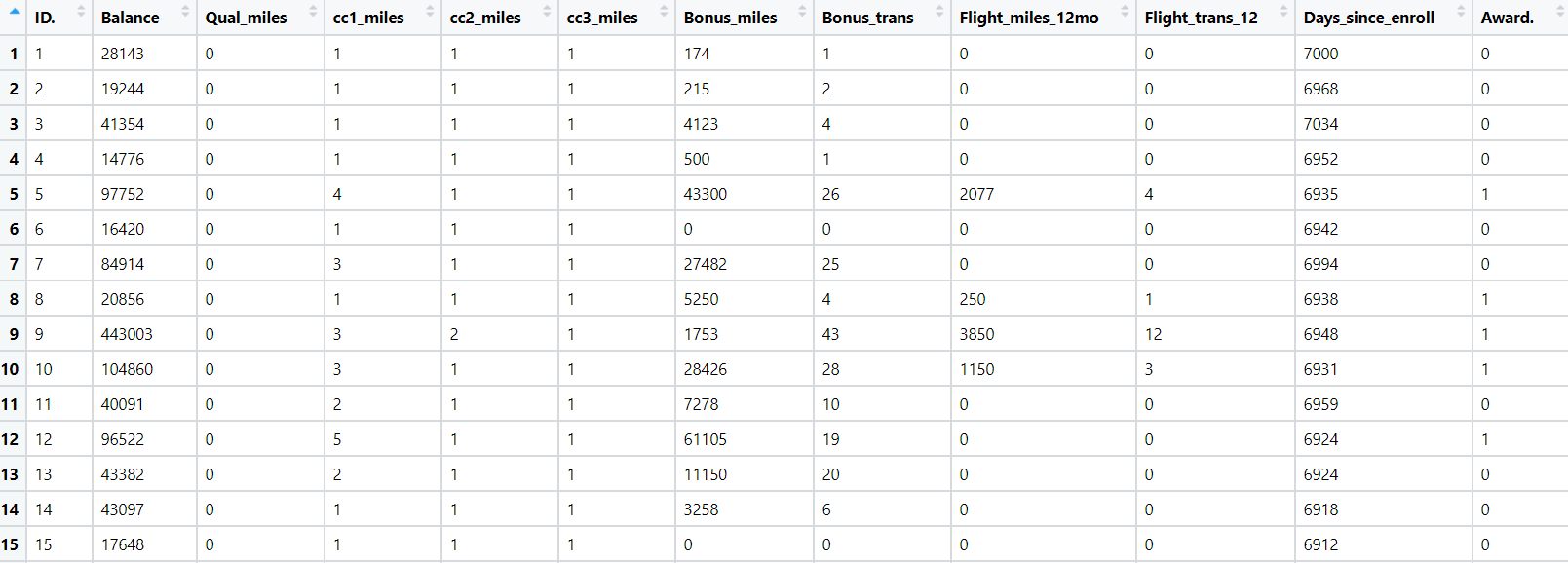
**Note:**

The assignment should be submitted in the following format:

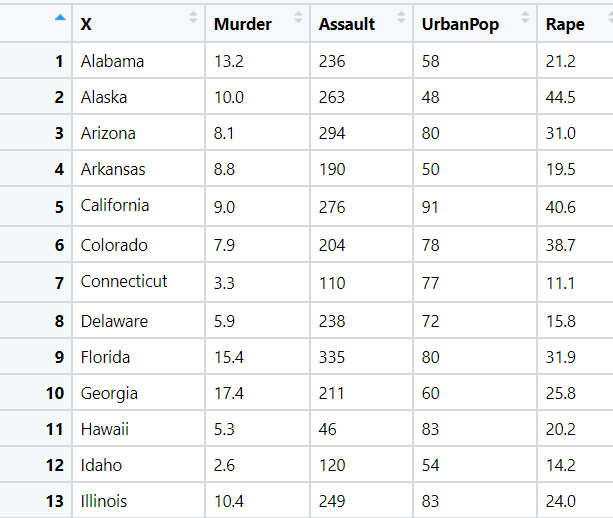
* R code
* Python code
* Code Modularization should be maintained
* Documentation of the modules (elaborating on steps mentioned above)

**Problem Statement:**

1. Perform clustering for the airlines data to obtain optimum number of clusters. Draw the inferences from the clusters obtained. Refer to EastWestAirlines.xlsx dataset.



1. Perform clustering for the crime data and identify the number of clusters formed and draw inferences. Refer to crime\_data.csv dataset.



1. Perform clustering analysis on the telecom data set. The data is a mixture of both categorical and numerical data. It consists the number of customers who churn. Derive insights and get possible information on factors that may affect the churn decision. Refer to Telco\_customer\_churn.xlsx dataset.

Hint:

* Perform EDA and remove unwanted columns.
* Use Gower dissimilarity matrix, In R use daisy() function.



1. Perform clustering on mixed data convert the categorical variables to numeric by using dummies or Label Encoding and perform normalization techniques. The data set consists details of customers related to auto insurance. Refer to Autoinsurance.csv dataset.

