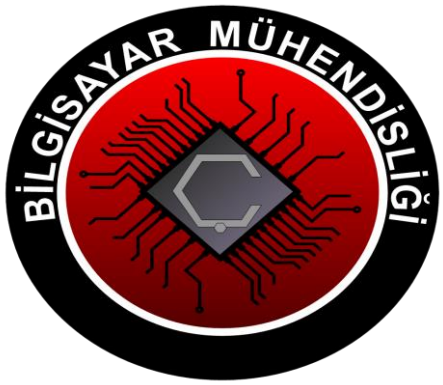


CENG 464

DATA MINING

DATA ANALYSING

PROJECT

Nurseli BAL

201611656

1. INTRODUCTION

In this report, I will be explaning the analysis of the given data. Firstly, I will read the data by using this comment below:

```
setwd("C:/Users/Nurseli/Desktop/dataa")

myData <- read.csv("data2.csv", header = TRUE)

View(myData)
```

The comment View(myData) is used for viewing the data that we have.

After I read the file, I will manipulate the data to make it useful and to get a better results in the analysis.

To use the some strong functions in R, I will need some packages. I download them using pacman package.

```
#necessary packages downloaded using pacman

install.packages("pacman")

require(pacman)

library(pacman)

pacman:: p_load(pacman,dply,GGally,ggplot2,ggthemes, ggvis, httr, lubridate, plotly, rio, rmarkdown, shiny, string, tidyr)
```

2.DATA PREPROCESSING

Since dealing with NA's is easier than NULL's, I will convert the NULL's into NA's using the comment:

myData [myData == "NULL"] <- NA

RELAFFIL	ADM_RATE	ADM_RATE_ALL	SATVR25	SATVR75	SATMT25	SATMT75	SATWR25
NULL	0.9027	0.902671312	365	485	360	495	370
NULL	0.9181	0.918067505	440	630	550	740	NULL
74	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	0.8123	0.812303547	550	660	530	670	NULL
NULL	0.9787	0.97866121	380	485	375	481	NULL
NULL	0.533	0.53295392	530	640	520	640	480
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	0.8254	0.825383994	490	565	475	545	NULL
NULL	0.8393	0.839309429	570	650	560	660	520
71	0.6186	0.618601016	520	630	510	630	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
68	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
74	0.5101	0.51006424	460	590	460	560	430
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SATVR25	SATVR75	SATMT25	SATMT75	SATWR25	SATWR75	SATVRMID	SATMTMID	SATWRMID
365	485	360	495	370	457	425	428	414
440	630	550	740	*NA*	*NA*	535	645	*NA*
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
550	660	530	670	*NA*	*NA*	605	600	*NA*
380	485	375	481	*NA*	*NA*	433	428	*NA*
530	640	520	640	480	600	585	580	540
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
490	565	475	545	*NA*	*NA*	528	510	*NA*
570	650	560	660	520	620	610	610	570
520	630	510	630	*NA*	*NA*	575	570	*NA*
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
460	590	460	560	430	510	525	510	470
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
NA	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*

Since in some attributes we have lots of NA values and we don't need to use some attributes. So I will delete those attributes to have better results using the below code.

myData <- myData %>% select(c(-1,-2,-3, -5, -6, -7,-8, -9, -10, -18, -20, -21,-22, -23, -36,-37,-38, -39, -40, -41, -42,-43,-44,-45,-46,-47,-48,-49,-50,-51,-52,-53,-54,-55,

-56,-57,-58,-59,-60,-61))

```
> summary(myData)
                                        INSTNM     SCH_DEG          HCM2
MAIN          NUMBRANCH         PREDDEG         HIGHDEG          CONTROL
REGION         CCBASIC
 Stevens-Henager College                :   7   1   :3034   Min.   :0.00000
Min.   :0.0000   Min.   : 1.000   Min.   :0.000   Min.   :0.000   Min.   :1
.000   Min.   :0.000   -2     :2343
 Bryan University                       :   5   2   :1277   1st Qu.:0.00000
1st Qu.:1.0000   1st Qu.: 1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1
.000   1st Qu.:3.000   18     : 349
 Columbia College                       :   5   3   :2529   Median :0.00000
Median :1.0000   Median : 1.000   Median :2.000   Median :2.000   Median :2
.000   Median :5.000   22     : 331
 McCann School of Business & Technology:   5   NULL:   0   Mean   :0.01378
Mean   :0.7677   Mean   : 3.883   Mean   :1.835   Mean   :2.234   Mean   :2
.129   Mean   :4.612   24     : 301
 Brittany Beauty Academy                :   4   NA's: 272   3rd Qu.:0.00000
3rd Qu.:1.0000   3rd Qu.: 2.000   3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:3
.000   3rd Qu.:6.000   10     : 265
 Unitek College                         :   4               Max.   :1.00000
Max.   :1.0000   Max.   :73.000   Max.   :4.000   Max.   :4.000   Max.   :3
.000   Max.   :9.000   (Other):3079
 (Other)                                :7082
NA's   : 444
    CCUGPROF        CCSIZSET       HBCU          PBI          ANNHI         TRIBAL
AANAPII      HSI          NANTI       MENONLY      WOMENONLY
 -2     :2343   -2     :2343   0   :6486   0   :6486   0   :6558   0   :655
3   0   :6454   0 :6145   0   :6559   0   :6605   0   :6633
 1      : 543   6      : 649   1   : 101   1   : 101   1   :  29   1   :  3
4   1   : 133   1 : 442   1   :  28   1   :  63   1   :  35
 11     : 460   1      : 500   NULL:   0   NULL:   0   NULL:   0   NULL:
0   NULL:   0   NU: 525   NULL:   0   NULL:   0   NULL:   0
 5      : 438   2      : 438   NA's: 525   NA's: 525   NA's: 525   NA's: 52
5   NA's: 525              NA's: 525   NA's: 444   NA's: 444
 4      : 370   11     : 377
 (Other):2514   (Other):2361
 NA's   : 444   NA's   : 444
```

As we can see that there are some NA values on the dataset. We have to get rid of them.

Since I had some problems on the binary data. I will convert the NA's by seperating the data.

```
for(i in 2:12){
  myData[,i] = as.numeric(myData[,i])
}


for(i in 2:12){
  myData[,i] = ifelse(is.na(myData[,i]),ave(myData[,i], FUN = function(x) mean(x, na.rm = 'TRUE')),myData[,i])
}
```

In the other part of my data, I used this comment:

myData[is.na(myData)] = 0

```
> summary(myData)
                                        INSTNM        SCH_DEG            HCM2
MAIN
 Stevens-Henager College                   :   7   Min.   :1.000   Min.   :0.00
000   Min.   :0.0000
 Bryan University                          :   5   1st Qu.:1.000   1st Qu.:0.00
000   1st Qu.:1.0000
 Columbia College                          :   5   Median :2.000   Median :0.00
000   Median :1.0000
 McCann School of Business & Technology:   5   Mean   :1.926   Mean   :0.01
378   Mean   :0.7677
 Brittany Beauty Academy                   :   4   3rd Qu.:3.000   3rd Qu.:0.00
000   3rd Qu.:1.0000
 Unitek College                            :   4   Max.   :3.000   Max.   :1.00
000   Max.   :1.0000
 (Other)                                   :7082
   NUMBRANCH          PREDDEG          HIGHDEG          CONTROL           REGIO
N          CCBASIC
 Min.   : 1.000   Min.   :0.000   Min.   :0.000   Min.   :1.000   Min.   :0
.000   Min.   : 1.00
 1st Qu.: 1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:3
.000   1st Qu.: 1.00
 Median : 1.000   Median :2.000   Median :2.000   Median :2.000   Median :5
.000   Median :10.97
 Mean   : 3.883   Mean   :1.835   Mean   :2.234   Mean   :2.129   Mean   :4
.612   Mean   :10.97
 3rd Qu.: 2.000   3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:6
.000   3rd Qu.:17.00
 Max.   :73.000   Max.   :4.000   Max.   :4.000   Max.   :3.000   Max.   :9
.000   Max.   :34.00

    CCUGPROF          CCSIZSET        HBCU         PBI         ANNHI        TRI
BAL      AANAPII
 Min.   : 1.000   Min.   : 1.000   0   :7011   0   :7011   0   :7083   0
:7078   0   :6979
 1st Qu.: 1.000   1st Qu.: 1.000   1   : 101   1   : 101   1   :  29   1
:  34   1   : 133
 Median : 5.000   Median : 5.000   NULL:   0   NULL:   0   NULL:   0   NULL
:   0   NULL:   0
 Mean   : 5.631   Mean   : 7.059
 3rd Qu.:10.000   3rd Qu.:12.000
 Max.   :17.000   Max.   :19.000

 HSI         NANTI       MENONLY     WOMENONLY
 0 :6145   0   :7084   0   :7049   0   :7077
 1 : 442   1   :  28   1   :  63   1   :  35
 NU: 525   NULL:   0   NULL:   0   NULL:   0
```

But still, there is a problem. There are unknown NU's in my dataset. Using the code below, I deleted them.

myData[myData == "NU"] <- 0

```
> summary(myData)
                                        INSTNM        SCH_DEG            HCM2
MAIN
 Stevens-Henager College                   :   7   Min.   :1.000   Min.   :0.00
000   Min.   :0.0000
 Bryan University                          :   5   1st Qu.:1.000   1st Qu.:0.00
000   1st Qu.:1.0000
 Columbia College                          :   5   Median :2.000   Median :0.00
000   Median :1.0000
 McCann School of Business & Technology:   5   Mean   :1.926   Mean   :0.01
378   Mean   :0.7677
 Brittany Beauty Academy                   :   4   3rd Qu.:3.000   3rd Qu.:0.00
000   3rd Qu.:1.0000
 Unitek College                            :   4   Max.   :3.000   Max.   :1.00
000   Max.   :1.0000
 (Other)                                   :7082
   NUMBRANCH         PREDDEG         HIGHDEG         CONTROL          REGIO
N         CCBASIC
 Min.   : 1.000   Min.   :0.000   Min.   :0.000   Min.   :1.000   Min.   :0
.000   Min.   : 1.00
 1st Qu.: 1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:3
.000   1st Qu.: 1.00
 Median : 1.000   Median :2.000   Median :2.000   Median :2.000   Median :5
.000   Median :10.97
 Mean   : 3.883   Mean   :1.835   Mean   :2.234   Mean   :2.129   Mean   :4
.612   Mean   :10.97
 3rd Qu.: 2.000   3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:6
.000   3rd Qu.:17.00
 Max.   :73.000   Max.   :4.000   Max.   :4.000   Max.   :3.000   Max.   :9
.000   Max.   :34.00

    CCUGPROF         CCSIZSET        HBCU         PBI         ANNHI        TRI
BAL     AANAPII
 Min.   : 1.000   Min.   : 1.000   0   :7011   0   :7011   0   :7083   0
:7078   0   :6979
 1st Qu.: 1.000   1st Qu.: 1.000   1   : 101   1   : 101   1   :  29   1
:  34   1   : 133
 Median : 5.000   Median : 5.000   NULL:   0   NULL:   0   NULL:   0   NULL
:   0   NULL:   0
 Mean   : 5.631   Mean   : 7.059
 3rd Qu.:10.000   3rd Qu.:12.000
 Max.   :17.000   Max.   :19.000

 HSI         NANTI       MENONLY     WOMENONLY
 0 :6670   0   :7084   0   :7049   0   :7077
 1 : 442   1   :  28   1   :  63   1   :  35
 NU:   0   NULL:   0   NULL:   0   NULL:   0
```

Now, finally we are done with the data preprocessing. We are coming to the clustering algorithms. I used 2 clustering algotirhms which are K means and Hierarchical clustering. These algorithms are really powerfull on data analysis.

3.CLUSTERING

3.1 HIERARCHICAL CLUSTERING

I used below comments to find it

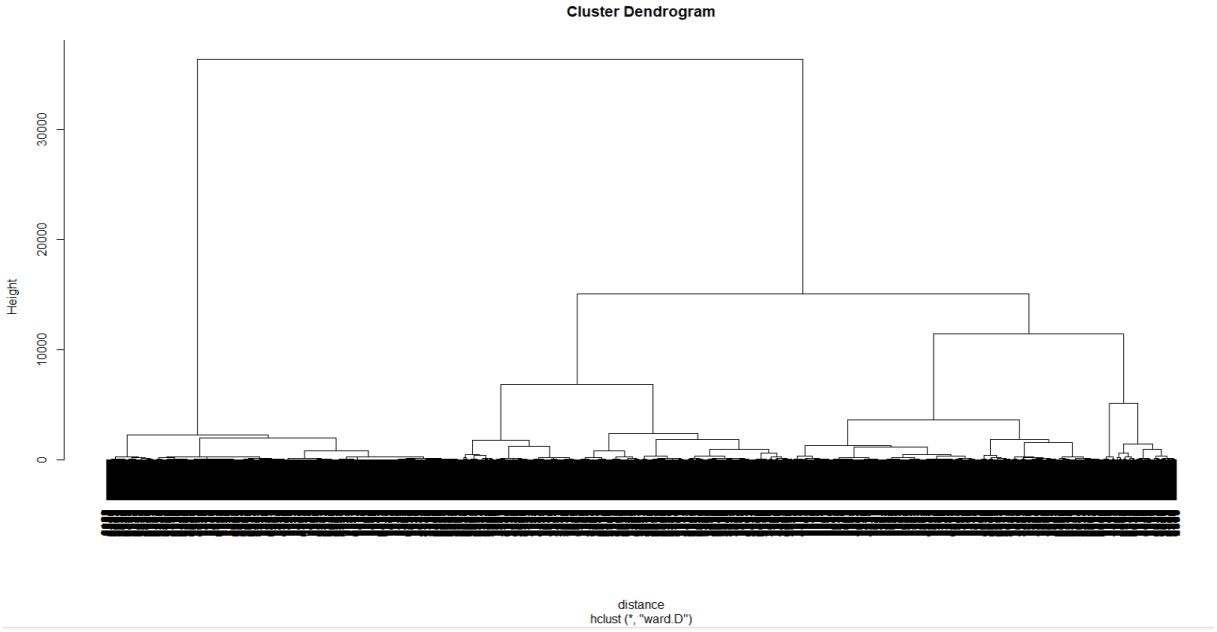
scaled <- myData[-c(1,1)]#omits the first colomn

distance<- dist(scaled, method="euclidean")

hfit <- hclust(distance, method="ward")

plot(hfit)

and the plotting graph is:

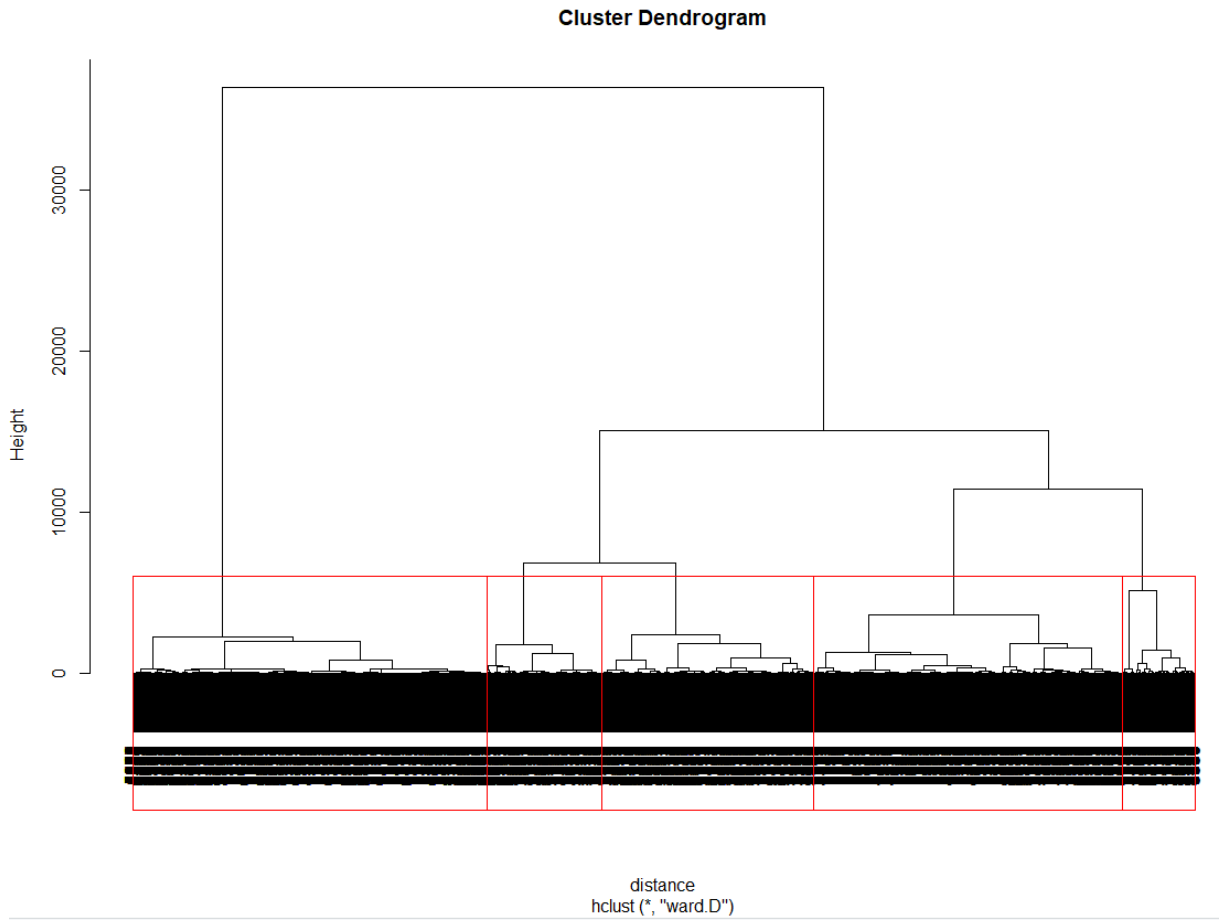


And then, I cut the tree using the below comments:

group = cutree(hfit, k=5)

group

rect.hclust(hfit,k=5,border="red")



The dendogram can be cut where the difference is most significant.

If we make a table with respect to HIGDEG attribute, we will have a result:

```
> table(HclusterCut, myData$HIGHDEG)

HclusterCut    0    1    2    3    4
          1  202    2  464  640 1858
          2   15    1    3  121   98
          3    0    0  685    4   28
          4  167 2275  328   15   36
          5   34    1   21    2    3
          6    0    0   17    6   12
          7   57    0    0    0   17
```

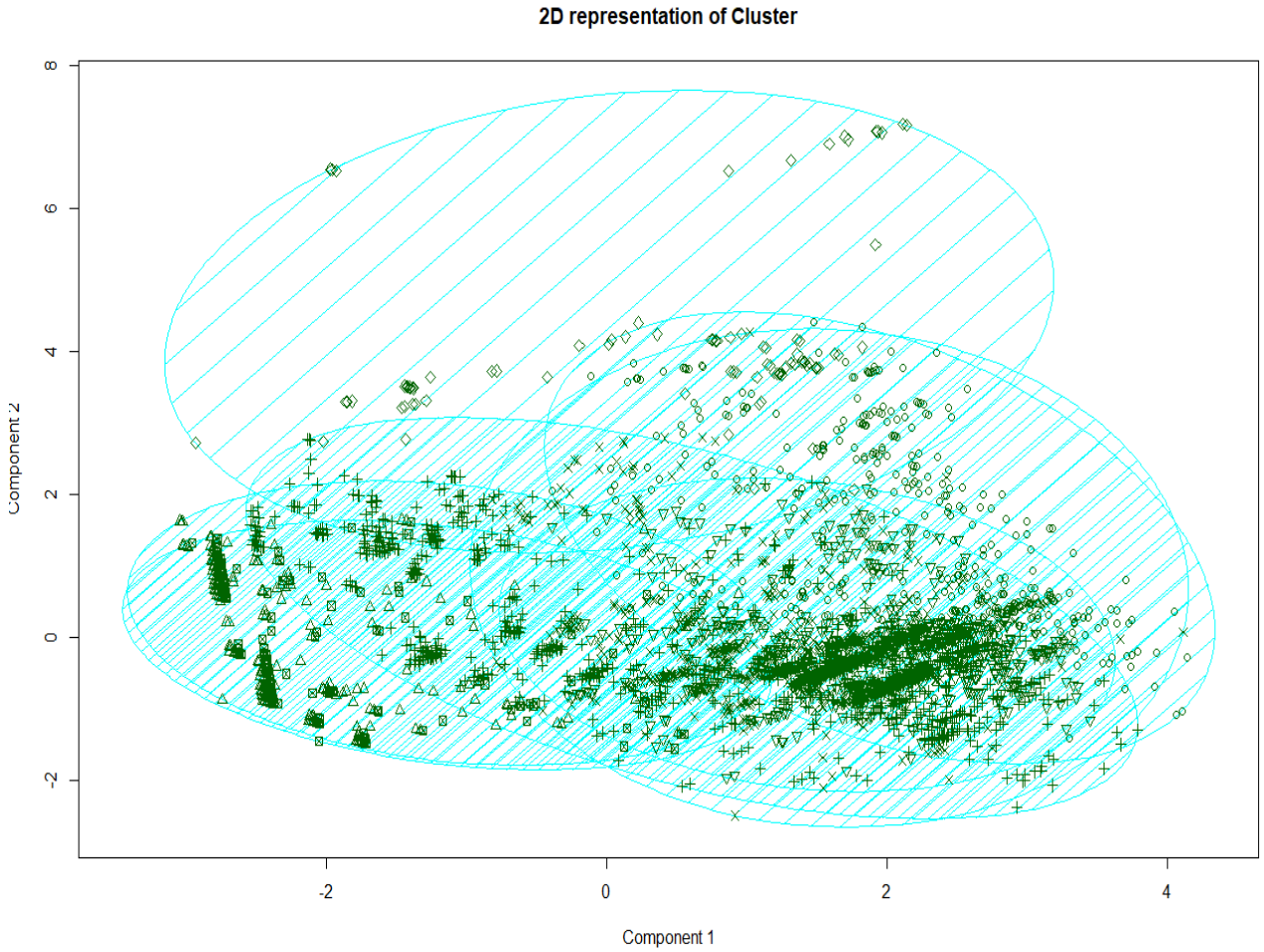
It is certain that it is not that good solution but the best solution.

3.2KMEANS CLUSTERING

I am going to cluster the data using the code below.

```
scaled <- myData[-c(1,1)]#omits the first colomn
kmeansdata <- kmeans(scaled, 7)
attributes(kmeansdata)
kmeansdata$cluster
c1 <- cbind(kmeansdata$cluster)
c1
library(cluster)
clusplot(scaled, kmeansdata$cluster, main="2D representation of Cluster",
shade=TRUE, label=0).
```

This code gives us the below plot.

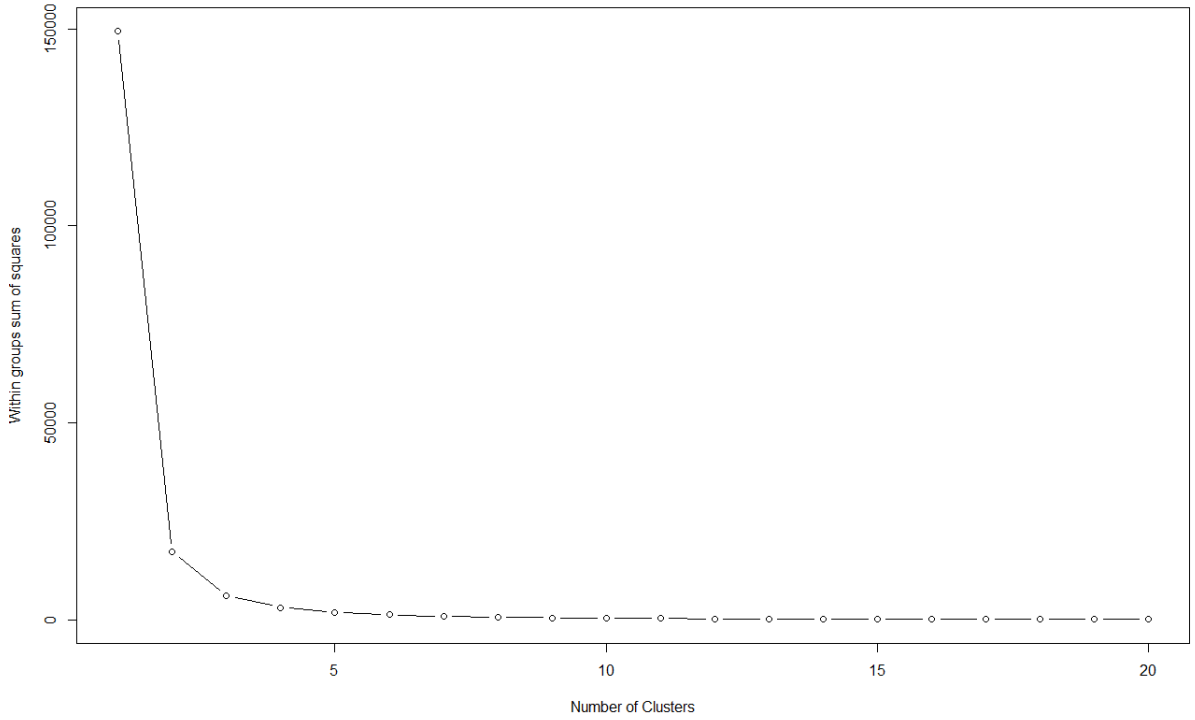


But in the kmeansdata <- kmeans(scaled, 7)

Comment I have chosen the k value as 7. But I did not know if it is the best choice for k value. Therefore we will use the below code to find the best k.

```
data.matrix(myData)
test1 <- scale(na.omit(data.matrix(myData)[-1]))
head(myData)
wssplot <- function(test1, nc=20, seed=123){
  wss <- (nrow(test1)-1)*sum(apply(test1,2,var))
  for(i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(test1, centers=i)$withinss)
  }
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares")
}

wssplot(test1, nc=20).
```

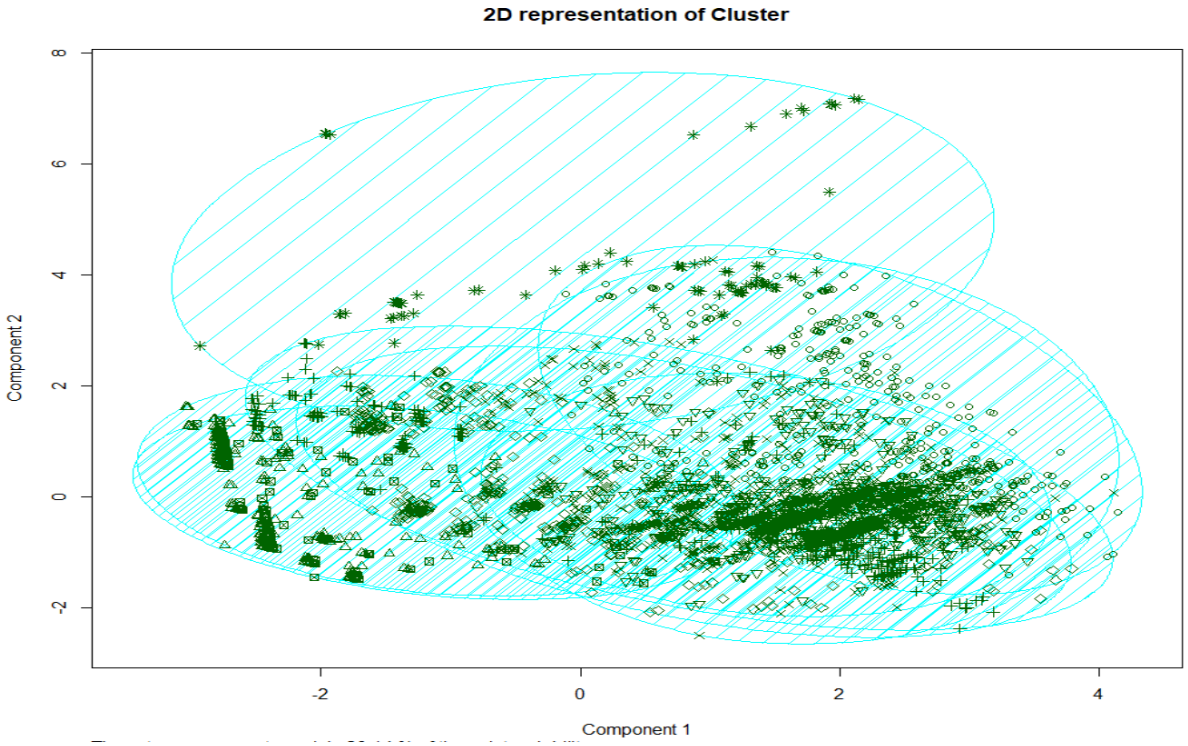
After I ran the comment above, I had a graph below:



In the graph, where the path gets sharp, the best k is there. The best sharp is 8 we'd say. That is why we will choose k as 8 to get better result.

```
scaled <- myData[-c(1,1)]#omits the first colomn

kmeansdata <- kmeans(scaled, 8)

attributes(kmeansdata)


kmeansdata$cluster

c1 <- cbind(kmeansdata$cluster)

c1

library(cluster)


clusplot(scaled, kmeansdata$cluster, main="2D representation of Cluster",
shade=TRUE, label=0).
```

The graph result:



4. CLASSIFICATION

I used two classification algorithms which are Decision Tree Classification algorithm and K Nearest Neighbor (KNN) Classification algorithm. Firstly I installed the necesarry packages to use strong functions of R.

```
library(mlbench)
install.packages("caret",dependencies = TRUE)
install.packages("e1071", dependencies=TRUE)
library(caret)
set.seed(12345)
```

I have chosen the class attribute as MAIN which for flag for main campus.

```
myData$MAIN = as.factor(myData$MAIN)
head(myData)

inTrain = createDataPartition(y = myData$MAIN, p = .75, list = FALSE)
#at start prediction percentage is 75%. If necessary, It will be changed to find an optimal solution.
training = myData[inTrain,]
testing = myData[-inTrain,]
```

4.1 DECISION TREE CLASSIFICATION

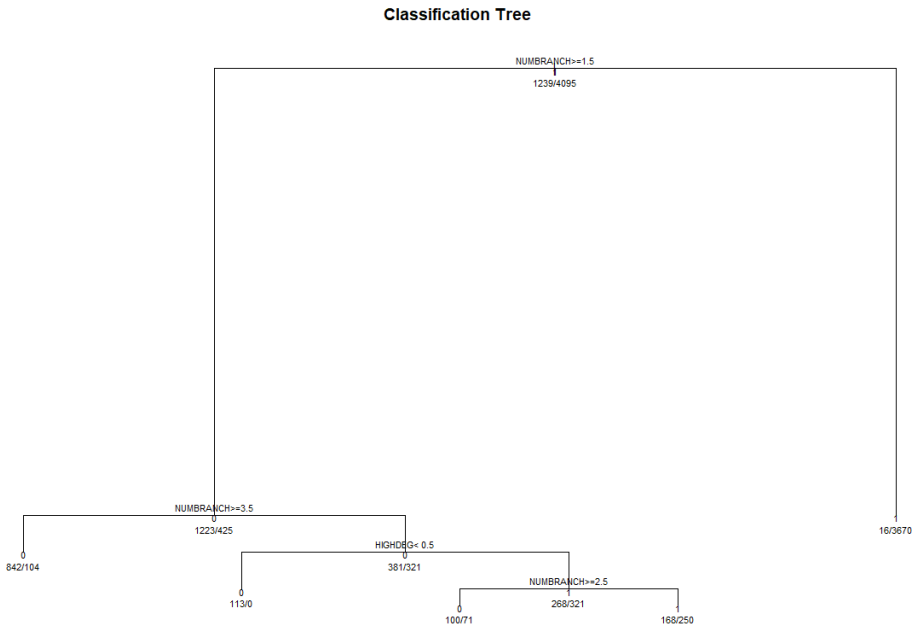
After using the comment below:

```
library(rpart)

myF <- MAIN ~ HCM2 + NUMBRANCH + SCH_DEG + PREDDEG + HIGHDEG +
CCBASIC + CCUGPROF + CCSIZSET

myData_dtree <- rpart(myF, data = training, method="class")

summary( myData_dtree)


# check the prediction

pred<-predict(myData_dtree, training[, c(2,3,5,6,7,10,11,12)], type="class")

confusionMatrix(table (pred, training$MAIN)),
```

We get the result on training data. We have 93% of accuracy on the training data. It is a good ressult. But we will see if the accuracy is high on the test data.

```
Confusion Matrix and Statistics


pred    0    1
   0 1055  175
   1  184 3920

               Accuracy : 0.9327
                 95% CI : (0.9256, 0.9393)
    No Information Rate : 0.7677
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8108

 Mcnemar's Test P-Value : 0.6729

            Sensitivity : 0.8515
            Specificity : 0.9573
         Pos Pred Value : 0.8577
         Neg Pred Value : 0.9552
             Prevalence : 0.2323
         Detection Rate : 0.1978
   Detection Prevalence : 0.2306
      Balanced Accuracy : 0.9044

       'Positive' Class : 0
```

We have the tree is like:



After I apply the below code, I saw that accuracy is still high. That means our model did not overfit. So, we have a good model for now.

predict on test data

testPred <- predict(myData_dtree, testing,type="class")

testPred

```
  testPred
       0    1
0  343   70
1   54 1311

               Accuracy : 0.9303
                 95% CI : (0.9174, 0.9417)
    No Information Rate : 0.7767
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8018

 Mcnemar's Test P-Value : 0.178

            Sensitivity : 0.8640
            Specificity : 0.9493
         Pos Pred Value : 0.8305
         Neg Pred Value : 0.9604
             Prevalence : 0.2233
         Detection Rate : 0.1929
   Detection Prevalence : 0.2323
      Balanced Accuracy : 0.9066

       'Positive' Class : 0
```

4.2 KNN

This is the second classification algorithm that I used.

KNN is a distance based algorithm. If we have different variables with varied scale (one variable which ranges from 1 to 100 and another variable ranges from 1 to 1,00,000), it would be difficult for the model to calculate distance for each and every point. In order to avoid these kind of scenarios, normalization is used. In order to normalize the data we have to convert it to numeric. Below code is doing that.

```
for(i in 1:20){
  if(!is.numeric(myData[,i]))
  {
    myData[,i] <- as.numeric(myData[,i])
  }
 }
```

After that, normalization function is created:

```
normalize <- function(x) {
 return ((x - min(x)) / (max(x) - min(x))) }
```

Then, I will apply the normalization function to our dataset using the below code:

```
myData_n <- as.data.frame(lapply(myData[,c(2,3,5,6,7,10,11,12)], normalize))
```

I will divide the prc_n data frame into prc_train and prc_test data frames for training and testin data.

```
myData_train <- myData_n[1:5000,]

myData_test <- myData_n[5001:7112,]

myData_train_labels <- myData[1:5000, 4]

myData_test_labels <- myData[5001:7112, 4]
```

We again need certain packages to use the necessary functions:

```
install.packages("class") #for knn

install.packages("Rtools")

library(class)

library(caret)
```

Now, we can do prediction by using the code below:

```
prc_test_pred <- knn(train = myData_train, test = myData_test,cl = myData_train_labels, k=20)
```

```
table( myData_test_labels,myData_test_labels)
```

```
> table( myData_test_labels,myData_test_labels)

                  myData_test_labels
myData_test_labels    1    2
                 1 1129    0
                 2    0  983
```

Above function shows that we made a good prediction and a good model.