ML Assignment 4

R Markdown

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

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Loading the Data

```
rm(list = ls())

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## Warning: package 'tibble' was built under R version 4.0.3
## Warning: package 'tidyr' was built under R version 4.0.3
## Warning: package 'readr' was built under R version 4.0.3
## Warning: package 'dplyr' was built under R version 4.0.3
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#install.packages("factoextra")
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.0.3
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ISLR)
set.seed(123)

DFUniver<-read.csv("Universities.csv")
colnames(DFUniver)
```

```
##  [1] "College.Name"              "State"
##  [3] "Public..1...Private..2."   "X..appli..rec.d"
##  [5] "X..appl..accepted"         "X..new.stud..enrolled"
##  [7] "X..new.stud..from.top.10." "X..new.stud..from.top.25."
##  [9] "X..FT.undergrad"           "X..PT.undergrad"
```

```
## [11] "in.state.tuition"          "out.of.state.tuition"
## [13] "room"                      "board"
## [15] "add..fees"                 "estim..book.costs"
## [17] "estim..personal.."         "X..fac..w.PHD"
## [19] "stud..fac..ratio"          "Graduation.rate"
```

```
#summary(DFUniver)

#Changing the column names to suitable ones.
DFUniver<-DFUniver%>%rename(
  Pub.Private=Public..1...Private..2.,
  ApplRec=X..appli..rec.d,
  ApplAccept=X..appl..accepted,
  NewStdEnr=X..new.stud..enrolled,
  Top10=X..new.stud..from.top.10.,
  Top25=X..new.stud..from.top.25.,
  FTUnderG=X..FT.undergrad,
  PTUnderG=X..PT.undergrad,
  InStateFee=in.state.tuition,
  OutStateFee=out.of.state.tuition,
  BookCost=estim..book.costs,
  PerCost=estim..personal..,
  PHD=X..fac..w.PHD,
  StFactRatio=stud..fac..ratio
)

colnames(DFUniver)
```

```
##  [1] "College.Name"    "State"           "Pub.Private"     "ApplRec"
##  [5] "ApplAccept"      "NewStdEnr"       "Top10"           "Top25"
##  [9] "FTUnderG"        "PTUnderG"        "InStateFee"      "OutStateFee"
## [13] "room"            "board"           "add..fees"       "BookCost"
## [17] "PerCost"         "PHD"             "StFactRatio"     "Graduation.rate"
```

Removing missing records from the Dataset (Measurements)

```
#Total NULL fields in the data frame
count(DFUniver[!complete.cases(DFUniver),])
```

```
##     n
## 1 831
```

```
#Ipute the NULL values
DFUniver1<-na.omit(DFUniver)
```

Finding the Data Summary and Measure of Dependence

```
#Summary Data
summary(DFUniver1)
```

```
##  College.Name          State            Pub.Private       ApplRec
##  Length:471         Length:471         Min.   :1.000   Min.   :   77
##  Class :character   Class :character   1st Qu.:1.000   1st Qu.:  802
##  Mode  :character   Mode  :character   Median :2.000   Median : 1646
##                                        Mean   :1.728   Mean   : 3147
##                                        3rd Qu.:2.000   3rd Qu.: 3862
##                                        Max.   :2.000   Max.   :48094
##    ApplAccept        NewStdEnr          Top10           Top25
```

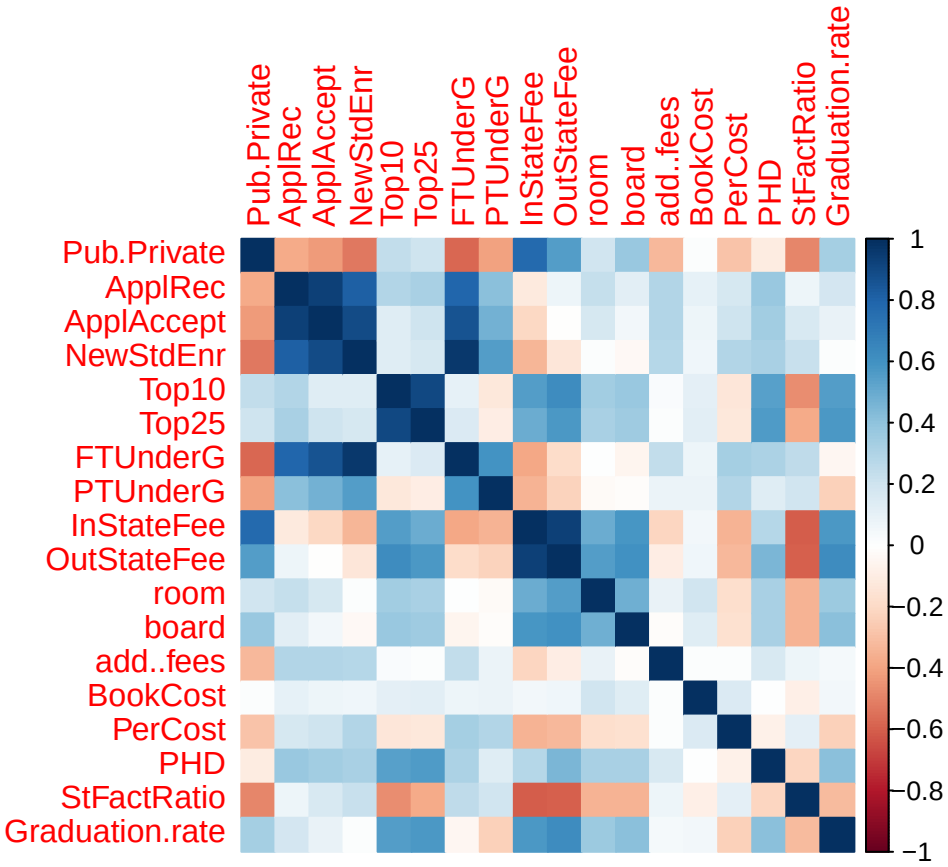
```
##  Min.   :   61.0   Min.   :  27.0   Min.   : 1.00   Min.   :  9.00
##  1st Qu.:  635.5   1st Qu.: 264.0   1st Qu.:15.00   1st Qu.: 40.00
##  Median : 1227.0   Median : 443.0   Median :23.00   Median : 54.00
##  Mean   : 2063.0   Mean   : 780.7   Mean   :28.01   Mean   : 55.65
##  3rd Qu.: 2456.0   3rd Qu.: 896.5   3rd Qu.:36.00   3rd Qu.: 69.00
##  Max.   :26330.0   Max.   :6392.0   Max.   :96.00   Max.   :100.00
##     FTUnderG        PTUnderG         InStateFee     OutStateFee
##  Min.   :  249   Min.   :   1.0   Min.   :  608   Min.   : 1044
##  1st Qu.: 1018   1st Qu.:  81.5   1st Qu.: 3650   1st Qu.: 7290
##  Median : 1715   Median :  299.0   Median : 9858   Median :10100
##  Mean   : 3563   Mean   :  797.5   Mean   : 9407   Mean   :10575
##  3rd Qu.: 4056   3rd Qu.:  869.0   3rd Qu.:13246   3rd Qu.:13286
##  Max.   :31643   Max.   :21836.0   Max.   :20100   Max.   :20100
##       room          board        add..fees         BookCost         PerCost
##  Min.   : 640   Min.   : 531   Min.   :  10.0   Min.   :  90.0   Min.   : 250
##  1st Qu.:1740   1st Qu.:1750   1st Qu.: 137.5   1st Qu.: 500.0   1st Qu.: 850
##  Median :2090   Median :2082   Median : 280.0   Median : 500.0   Median :1200
##  Mean   :2221   Mean   :2122   Mean   : 379.0   Mean   : 548.8   Mean   :1312
##  3rd Qu.:2663   3rd Qu.:2420   3rd Qu.: 486.0   3rd Qu.: 600.0   3rd Qu.:1600
##  Max.   :4816   Max.   :4541   Max.   :3247.0   Max.   :2340.0   Max.   :6800
##       PHD          StFactRatio    Graduation.rate
##  Min.   :  8.00   Min.   : 2.90   Min.   : 15.00
##  1st Qu.: 63.00   1st Qu.:11.30   1st Qu.: 53.00
##  Median : 76.00   Median :13.40   Median : 66.00
##  Mean   : 73.21   Mean   :13.96   Mean   : 65.56
##  3rd Qu.: 87.00   3rd Qu.:16.45   3rd Qu.: 79.00
##  Max.   :103.00   Max.   :28.80   Max.   :118.00
```

```
#Finding the correlation between the data set
#Selecting numerical columns only

DFNumerical<-DFUniver1[,c(-1,-2)]
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.3
```

```
## corrplot 0.84 loaded
```

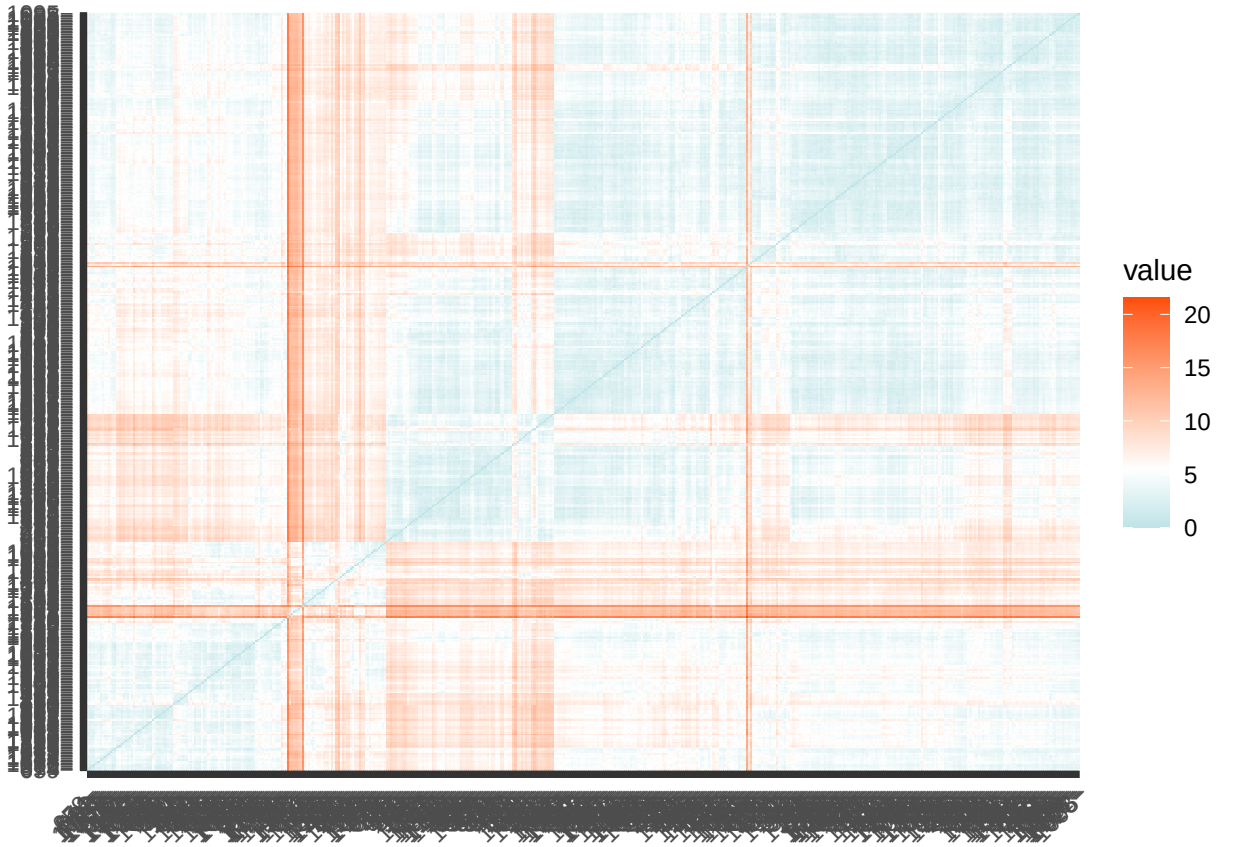
```
corrplot(cor(DFNumerical), method = "color")
```



In the correlation graph, Darker Blue(+1) and Dark Orange(-1) shows the higher correlated data. Using this data to understand any correlation among the column data.

Applying K-means clustering for Numeric Data

```
#Scaling the Data
DFNumerical<-scale(DFNumerical)

#Distance Between Observations
distance <- get_dist(DFNumerical)

fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```




```
#Finding Kmeans using cluster size =4

k4 <- kmeans(DFNumerical, centers = 4, nstart = 25) # k = 4, number of restarts = 25
str(k4)
```

```
## List of 9
##  $ cluster     : Named int [1:471] 2 3 1 2 2 3 2 2 2 2 ...
##   ..- attr(*, "names")= chr [1:471] "1" "3" "10" "12" ...
##  $ centers     : num [1:4, 1:18] 0.575 0.61 -1.516 -1.416 0.117 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:18] "Pub.Private" "ApplRec" "ApplAccept" "NewStdEnr" ...
##  $ totss       : num 8460
##  $ withinss    : num [1:4] 1231 1553 901 980
##  $ tot.withinss: num 4664
##  $ betweenss   : num 3796
##  $ size        : int [1:4] 129 207 94 41
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
# Visualize the output

k4$centers # output the centers
```

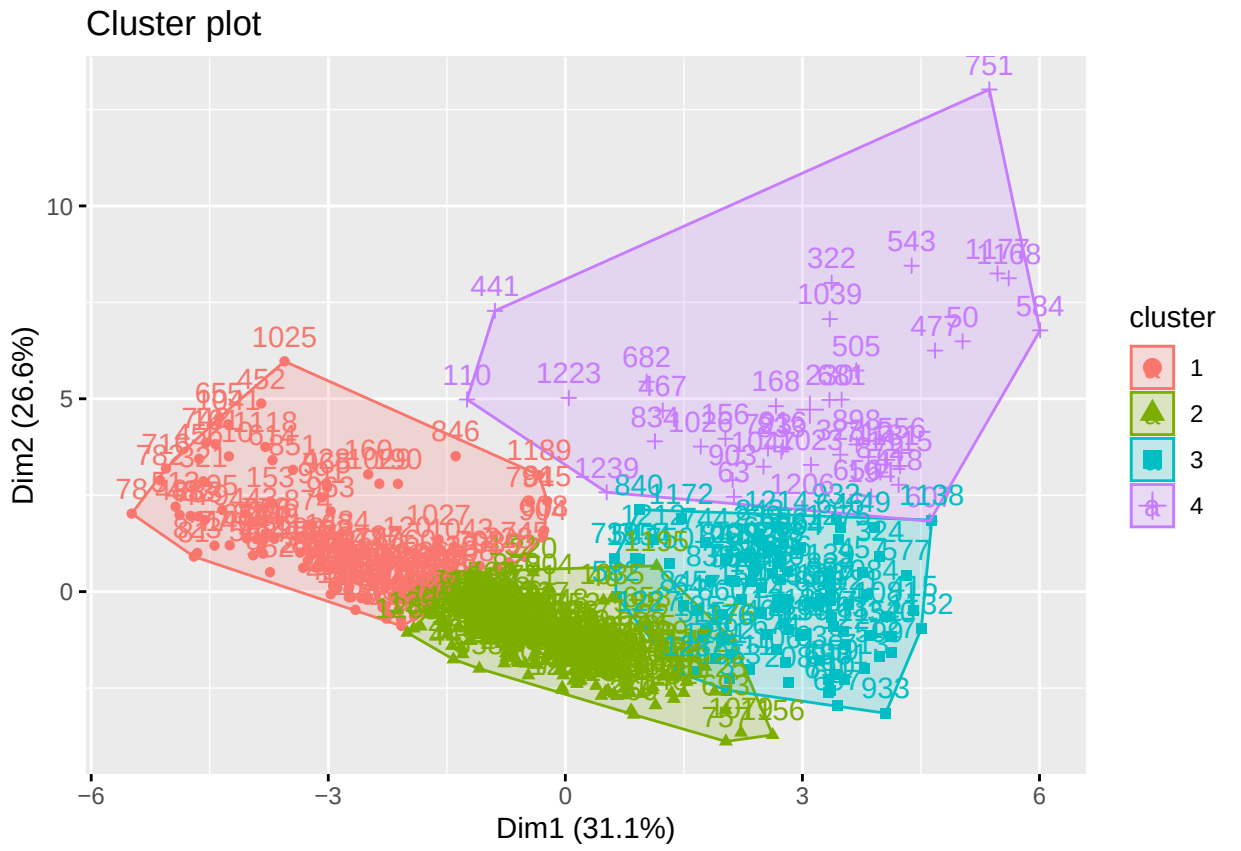
```
##   Pub.Private     ApplRec   ApplAccept  NewStdEnr      Top10      Top25
## 1   0.5754205  0.11722831 -0.005168206 -0.1421639  1.0266549  0.9981228
```

```
## 2   0.6102340 -0.51123189 -0.497187527 -0.5094276 -0.3760952 -0.4016802
## 3  -1.5157965  0.02245762  0.051502226  0.1968843 -0.6529467 -0.6177846
## 4  -1.4161661  2.16076910  2.408373353  2.5678908  0.1656150  0.3039442
##     FTUnderG   PTUnderG  InStateFee OutStateFee       room       board
## 1 -0.2109459 -0.3189375  1.15189926   1.2186862  0.7642290  0.81696559
## 2 -0.5000169 -0.2631323  0.06788283  -0.1844498 -0.2837257 -0.13014031
## 3  0.2524856  0.2243992 -1.28898602  -1.0950488 -0.4342977 -0.80027141
## 4  2.6093137  1.8175077 -1.01175768  -0.3925568  0.0236506 -0.07863426
##     add..fees    BookCost     PerCost         PHD StFactRatio Graduation.rate
## 1 -0.00526603  0.14578619 -0.38510673  0.84849798 -0.74794298       0.9053670
## 2 -0.30637755 -0.07869207 -0.07148593 -0.64092902 -0.03452639      -0.1830111
## 3  0.43150476 -0.11321423  0.28883658 -0.06099844  0.90018939      -0.7708289
## 4  0.57409815  0.19817016  0.91038335  0.70609563  0.46375131      -0.1573444
```

```
#number of Universities in each cluster
k4$size
```

```
## [1] 129 207  94  41
```

```
# Visualize the output - Application Accepted vs Out of State Fee

fviz_cluster(k4, data =DFNumerical)
```



Comparision different cluster values

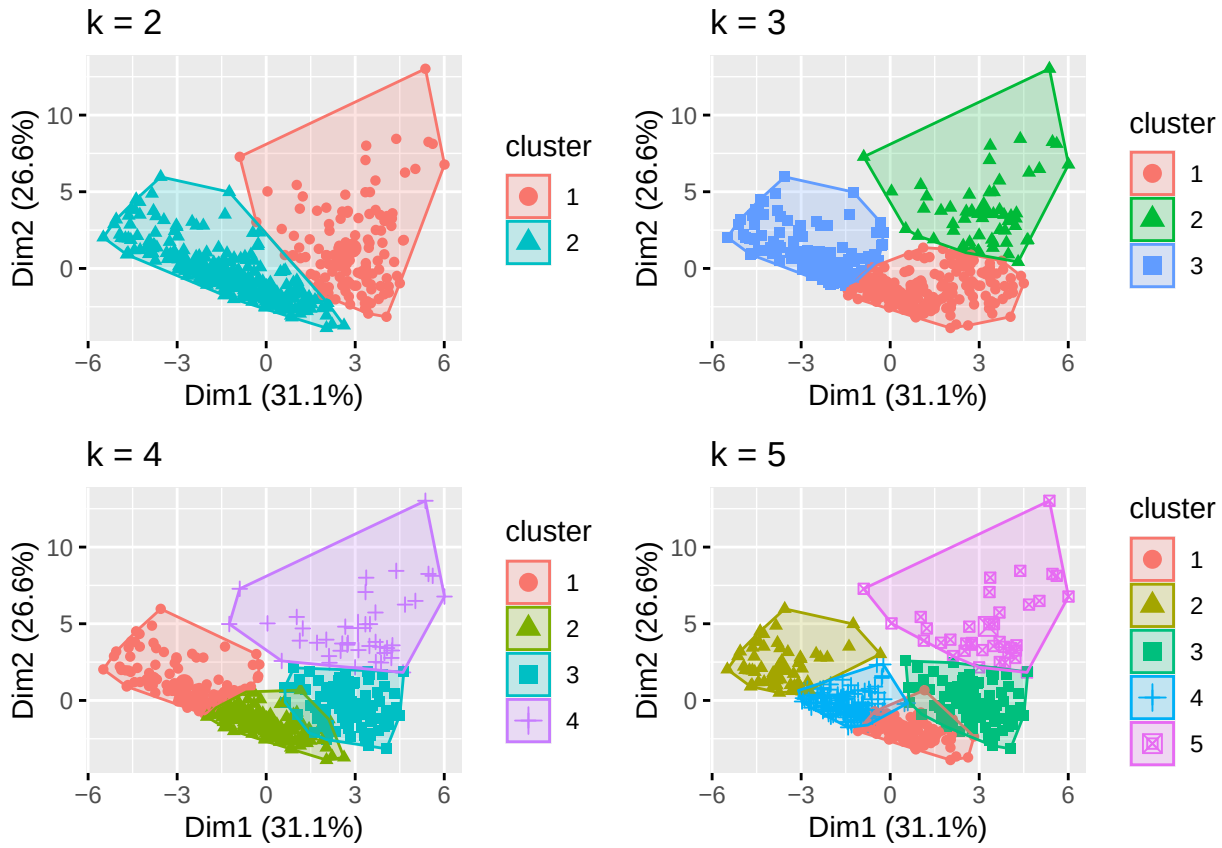
```
k2 <- kmeans(DFNumerical, centers = 2, nstart = 25)
k3 <- kmeans(DFNumerical, centers = 3, nstart = 25)
k4 <- kmeans(DFNumerical, centers = 4, nstart = 25)
```

```
k5 <- kmeans(DFNumerical, centers = 5, nstart = 25)

# plots to compare
p1 <- fviz_cluster(k2, geom = "point", data = DFNumerical) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = DFNumerical) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",  data = DFNumerical) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",  data = DFNumerical) + ggtitle("k = 5")

library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

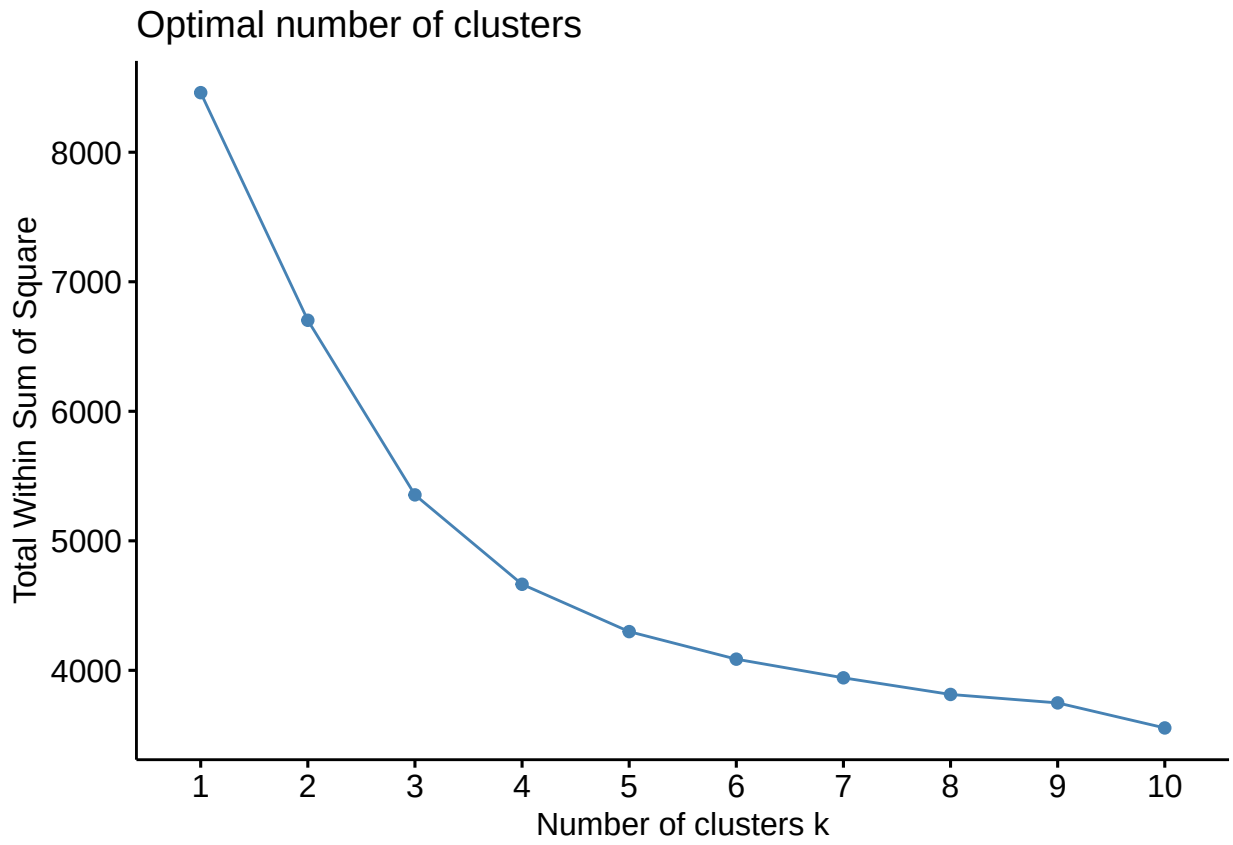
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



From the above comparison it seems that 3 clusters would be good.

```
set.seed(123)
#Finding optimal number of clusters - Elbow Method
fviz_nbclust(DFNumerical, kmeans, method = "wss")
```




```
#From the Elbow method it seems 4 clusters would be optimum.
```

```
DFUniver1 %>%
  mutate(Cluster = k4$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

```
## Warning in mean.default(College.Name): argument is not numeric or logical:
## returning NA

## Warning in mean.default(College.Name): argument is not numeric or logical:
## returning NA

## Warning in mean.default(College.Name): argument is not numeric or logical:
## returning NA

## Warning in mean.default(College.Name): argument is not numeric or logical:
## returning NA

## Warning in mean.default(State): argument is not numeric or logical: returning NA

## Warning in mean.default(State): argument is not numeric or logical: returning NA

## Warning in mean.default(State): argument is not numeric or logical: returning NA

## Warning in mean.default(State): argument is not numeric or logical: returning NA

## # A tibble: 4 x 21
```

```
##   Cluster College.Name State Pub.Private ApplRec ApplAccept NewStdEnr Top10
##     <int>        <dbl> <dbl>       <dbl>   <dbl>      <dbl>     <dbl> <dbl>
## 1       1           NA    NA        1.98   3625.      2050.      651.  47.0
## 2       2           NA    NA        2      1065.       818.      314.  21.1
## 3       3           NA    NA        1.05   3239.      2192.      961.  15.9
## 4       4           NA    NA        1.10  11948.      8093.     3132.  31.1
## # ... with 13 more variables: Top25 <dbl>, FTUnderG <dbl>, PTUnderG <dbl>,
## #   InStateFee <dbl>, OutStateFee <dbl>, room <dbl>, board <dbl>,
## #   add..fees <dbl>, BookCost <dbl>, PerCost <dbl>, PHD <dbl>,
## #   StFactRatio <dbl>, Graduation.rate <dbl>
```

Using the categorical measurements that were not used i n the analysis (State and Private/Public) to characterize the different clusters.

```
#State wise values present in the cluster
table(DFUniver1$State, k4$cluster)
```

```
##
##       1  2  3  4
##   AK  0  1  1  0
##   AL  1  2  1  0
##   AR  0  4  0  0
##   AZ  0  0  1  1
##   CA 10  3  0  2
##   CO  1  0  5  0
##   CT  3  4  2  1
##   DC  4  0  0  0
##   DE  0  1  0  1
##   FL  4  3  0  1
##   GA  2  3  1  1
##   HI  0  0  1  0
##   IA  2 15  1  0
##   ID  0  2  0  0
##   IL  4  7  2  2
##   IN  6  8  1  0
##   KS  0  7  0  0
##   KY  2  3  1  0
##   LA  2  1  1  1
##   MA 11  4  4  3
##   MD  1  0  1  1
##   ME  2  1  3  0
##   MI  2  8  1  2
##   MN  3  5  2  1
##   MO  2 10  2  1
##   MS  0  2  3  0
##   MT  0  1  1  0
##   NC  2 11  7  3
##   ND  0  1  4  0
##   NE  1  3  2  1
##   NH  1  3  1  1
##   NJ  3  3  6  1
##   NM  0  2  0  0
##   NY 18  8 10  2
##   OH  7 13  0  4
##   OK  0  3  3  0
```

```
##   OR  2  3  0  0
##   PA 18 17  4  3
##   RI  2  1  0  1
##   SC  1  5  3  0
##   SD  0  2  2  0
##   TN  3 11  0  1
##   TX  2  9  7  2
##   UT  0  1  0  1
##   VA  3  7  3  2
##   VT  1  2  3  1
##   WA  2  0  0  0
##   WI  1  6  2  0
##   WV  0  1  1  0
##   WY  0  0  1  0
```

Tufts University data imputation

```
# Initial Dataframe DFUniver with no imputation
library(flexclust)
```

```
## Warning: package 'flexclust' was built under R version 4.0.3
```

```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: modeltools
```

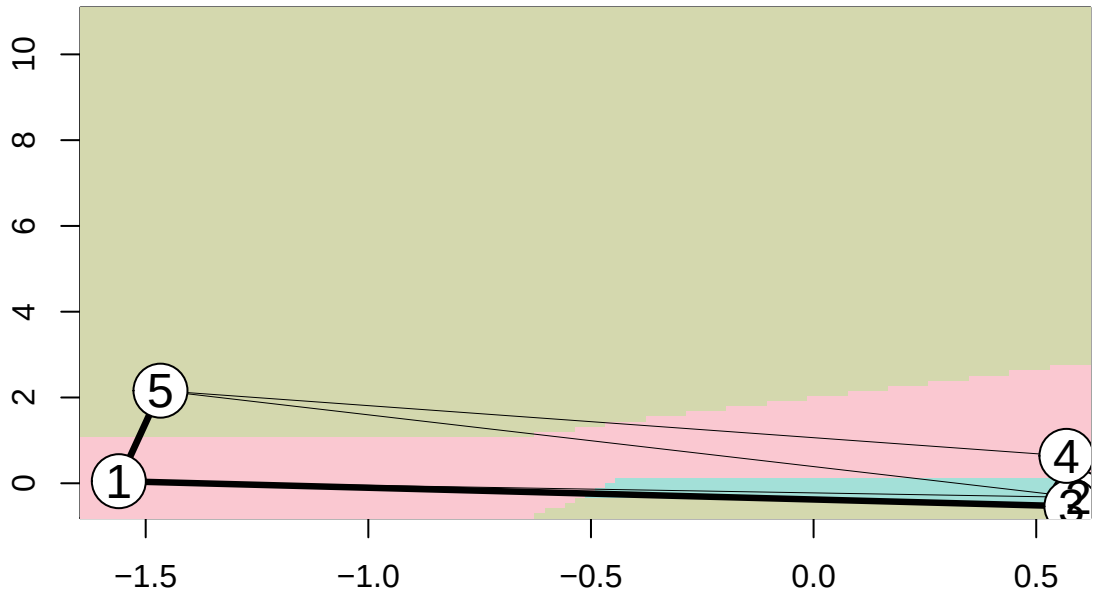
```
## Warning: package 'modeltools' was built under R version 4.0.3
```

```
## Loading required package: stats4
```

```
set.seed(123)
#kmeans clustering, using Euclidean distance
k5 = kcca(DFNumerical, k=5, kccaFamily("kmeans"))
k5
```

```
## kcca object of family 'kmeans'
##
## call:
## kcca(x = DFNumerical, k = 5, family = kccaFamily("kmeans"))
##
## cluster sizes:
##
##   1   2   3   4   5
##  90 144 144  53  40
```

```
#Apply the predict() function
clusters_index <- predict(k5)
image(k5)
```




```
#points(df, col=clusters_index, pch=19, cex=0.3)
```