DeSimone_MS64060_Assignment 4

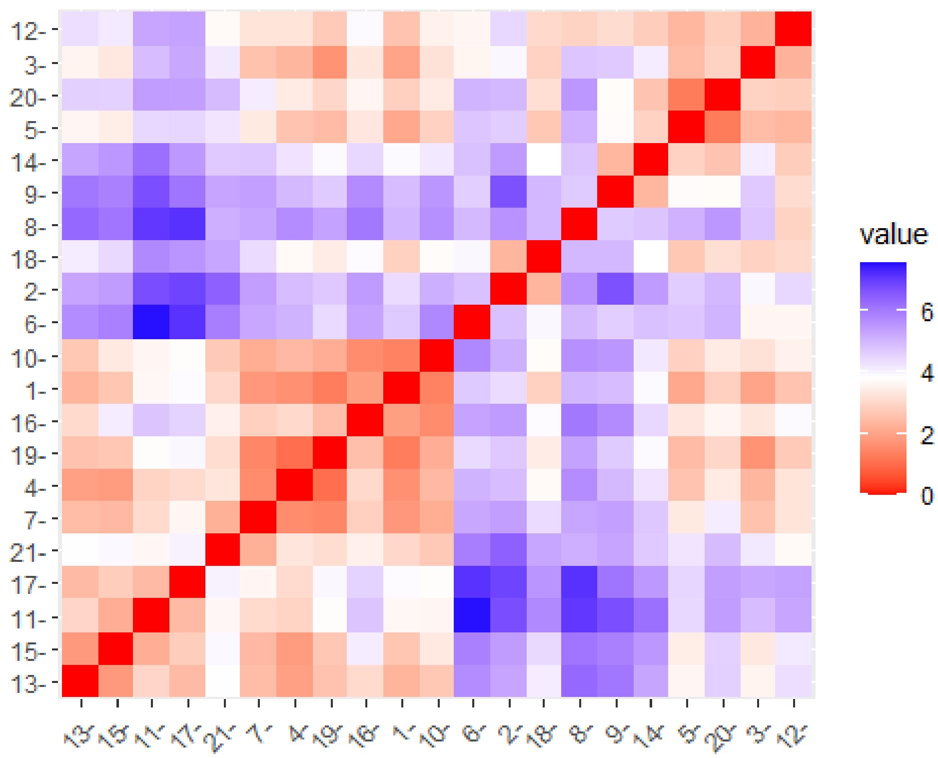
Heather DeSimone

3/19/2022

##First I have loaded in my data frame and called a summary of the information.

```
df.original=read.csv("C:/Users/hdesi/Desktop/MBA/Machine
Learning/Pharmaceuticals.csv")
df.original.names.numerical=read.csv("C:/Users/hdesi/Desktop/MBA/Machine
Learning/Pharmaceuticals.csv")[,c(1:11)]
df=read.csv("C:/Users/hdesi/Desktop/MBA/Machine
Learning/Pharmaceuticals.csv")[,c(3:11)]
library(ISLR)

## Warning: package 'ISLR' was built under R version 4.1.1

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.1.2

## Warning: package 'ggplot2' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'readr' was built under R version 4.1.3

## Warning: package 'stringr' was built under R version 4.1.2
```

```
## Warning: package 'forcats' was built under R version 4.1.3

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(factoextra)

## Warning: package 'factoextra' was built under R version 4.1.3

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

set.seed(123)
summary(df)

##    Market_Cap          Beta           PE_Ratio          ROE
##  Min.   :  0.41   Min.   :0.1800   Min.   : 3.60   Min.   : 3.9
##  1st Qu.:  6.30   1st Qu.:0.3500   1st Qu.:18.90   1st Qu.:14.9
##  Median : 48.19   Median :0.4600   Median :21.50   Median :22.6
##  Mean   : 57.65   Mean   :0.5257   Mean   :25.46   Mean   :25.8
##  3rd Qu.: 73.84   3rd Qu.:0.6500   3rd Qu.:27.90   3rd Qu.:31.0
##  Max.   :199.47   Max.   :1.1100   Max.   :82.50   Max.   :62.9
##       ROA        Asset_Turnover    Leverage        Rev_Growth
##  Min.   : 1.40   Min.   :0.3    Min.   :0.0000   Min.   :-3.17
##  1st Qu.: 5.70   1st Qu.:0.6    1st Qu.:0.1600   1st Qu.: 6.38
##  Median :11.20   Median :0.6    Median :0.3400   Median : 9.37
##  Mean   :10.51   Mean   :0.7    Mean   :0.5857   Mean   :13.37
##  3rd Qu.:15.00   3rd Qu.:0.9    3rd Qu.:0.6000   3rd Qu.:21.87
##  Max.   :20.30   Max.   :1.1    Max.   :3.5100   Max.   :34.21
##  Net_Profit_Margin
##  Min.   : 2.6
##  1st Qu.:11.2
##  Median :16.1
##  Mean   :15.7
##  3rd Qu.:21.1
##  Max.   :25.5

df <- scale(df) #z-score
distance <- get_dist(df)
fviz_dist(distance)
```



##Will try k=4 first since 4 is the median distance shown in the above graph ##Will first use 25 restarts as it seems to be a typical number of random centroids to start with (based on the internet community)

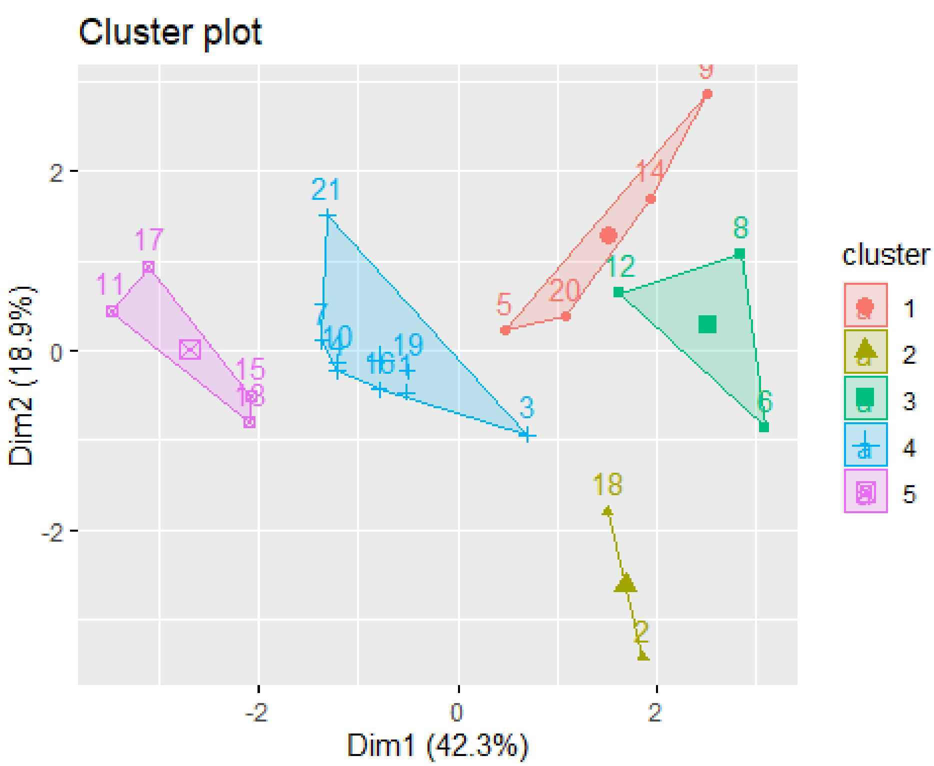
```
k4 <- kmeans(df, centers = 4, nstart = 25)
k4$centers

##    Market_Cap       Beta   PE_Ratio        ROE        ROA Asset_Turnover
## 1  1.69558112 -0.1780563 -0.1984582  1.2349879  1.3503431   1.153164e+00
## 2 -0.03142211 -0.4360989 -0.3172485  0.1950459  0.4083915   1.729746e-01
## 3 -0.82617719  0.4775991 -0.3696184 -0.5631589 -0.8514589  -9.994088e-01
## 4 -0.52462814  0.4451409  1.8498439 -1.0404550 -1.1865838   1.480297e-16
##     Leverage Rev_Growth Net_Profit_Margin
## 1 -0.4680782  0.4671788         0.5912425
## 2 -0.2744931 -0.7041516         0.5569544
## 3  0.8502201  0.9158889        -0.3319956
## 4 -0.3443544 -0.5769454        -1.6095439

k4$size

## [1] 4 8 6 3

## 4, 8, 6, 3

##21 data points total so lets look at where the 1st, last, and middle data
points are
k4$cluster[1] ##cluster 2
```

```
## [1] 2

k4$cluster[10] ##cluster 2

## [1] 2

k4$cluster[21] ##cluster 2

## [1] 2

fviz_cluster(k4, data = df) ##Visual
```

##Lets see what happens with k=5

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

##     Market_Cap       Beta    PE_Ratio        ROE        ROA Asset_Turnover
## 1 -0.76022489  0.2796041 -0.47742380 -0.7438022 -0.8107428     -1.2684804
## 2 -0.43925134 -0.4701800  2.70002464 -0.8349525 -0.9234951      0.2306328
## 3 -0.87051511  1.3409869 -0.05284434 -0.6184015 -1.1928478     -0.4612656
## 4 -0.03142211 -0.4360989 -0.31724852  0.1950459  0.4083915      0.1729746
## 5  1.69558112 -0.1780563 -0.19845823  1.2349879  1.3503431      1.1531640
##       Leverage Rev_Growth Net_Profit_Margin
## 1  0.06308085  1.5180158      -0.006893899
## 2 -0.14170336 -0.1168459      -1.416514761
## 3  1.36644699 -0.6912914      -1.320000179
## 4 -0.27449312 -0.7041516       0.556954446
## 5 -0.46807818  0.4671788       0.591242521
```

```
k5$size

## [1] 4 2 3 8 4

## 4, 2, 3, 8, 4

##21 data points total so lets look at where the 1st, last, and middle data
points are
k5$cluster[1] ##cluster 4

## [1] 4

k5$cluster[10] ##cluster 4

## [1] 4

k5$cluster[21] ##cluster 4

## [1] 4

##These data points are all falling in the same bucket - close together

fviz_cluster(k5, data = df) ##Visual
```



##Lets see what happens with k=3

```
k3 <- kmeans(df, centers = 3, nstart = 25)
k3$centers
```

```
##   Market_Cap       Beta   PE_Ratio        ROE        ROA Asset_Turnover
## 1  0.6733825 -0.3586419 -0.2763512  0.6565978  0.8344159      0.4612656
## 2 -0.6125361  0.2698666  1.3143935 -0.9609057 -1.0174553      0.2306328
## 3 -0.8261772  0.4775991 -0.3696184 -0.5631589 -0.8514589     -0.9994088
##     Leverage Rev_Growth Net_Profit_Margin
## 1 -0.3331068 -0.2902163         0.6823310
## 2 -0.3592866 -0.5757385        -1.3784169
## 3  0.8502201  0.9158889        -0.3319956
```

```
k3$size
```

```
## [1] 11  4  6
```

```
## 11, 4, 6

##21 data points total so lets look at where the 1st, last, and middle data
points are
k3$cluster[1] ##cluster 1
```

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

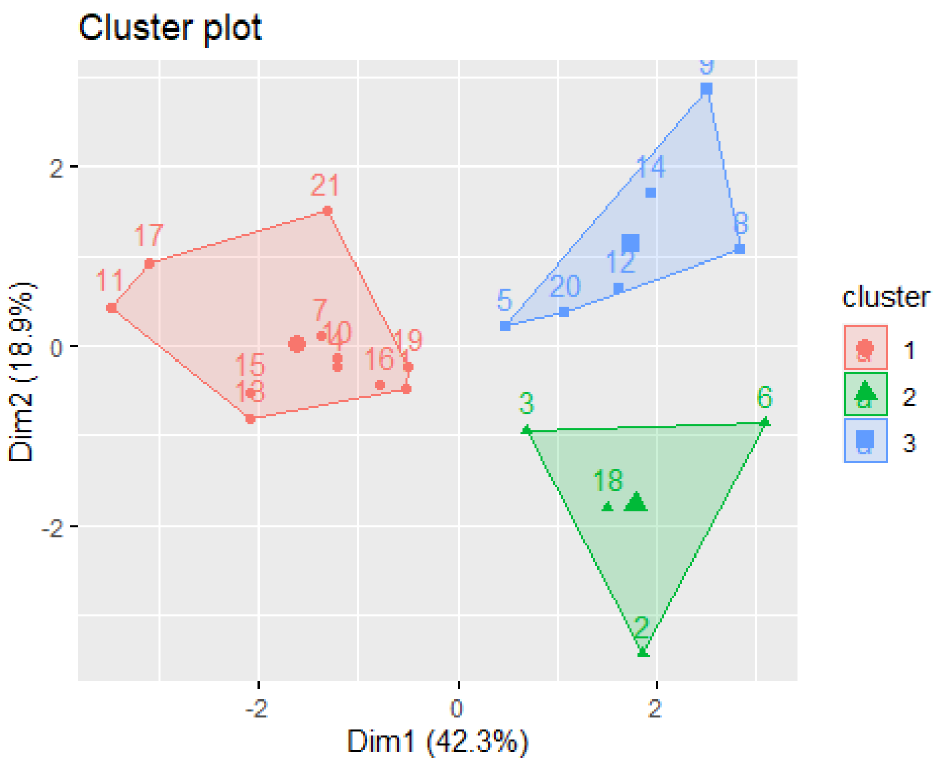
```
k3$cluster[10] ##cluster 1
```

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

```
k3$cluster[21] ##cluster 1
```

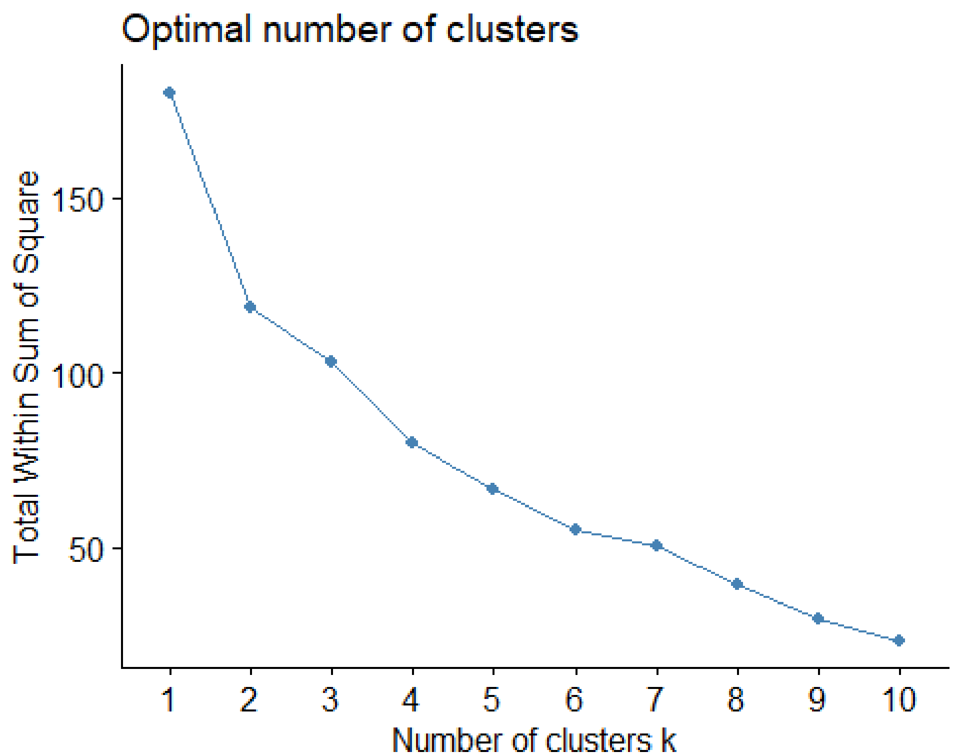
```
## [1] 1
```

```
##These data points are all falling in the same bucket - close together

fviz_cluster(k3, data = df) ##Visual
```



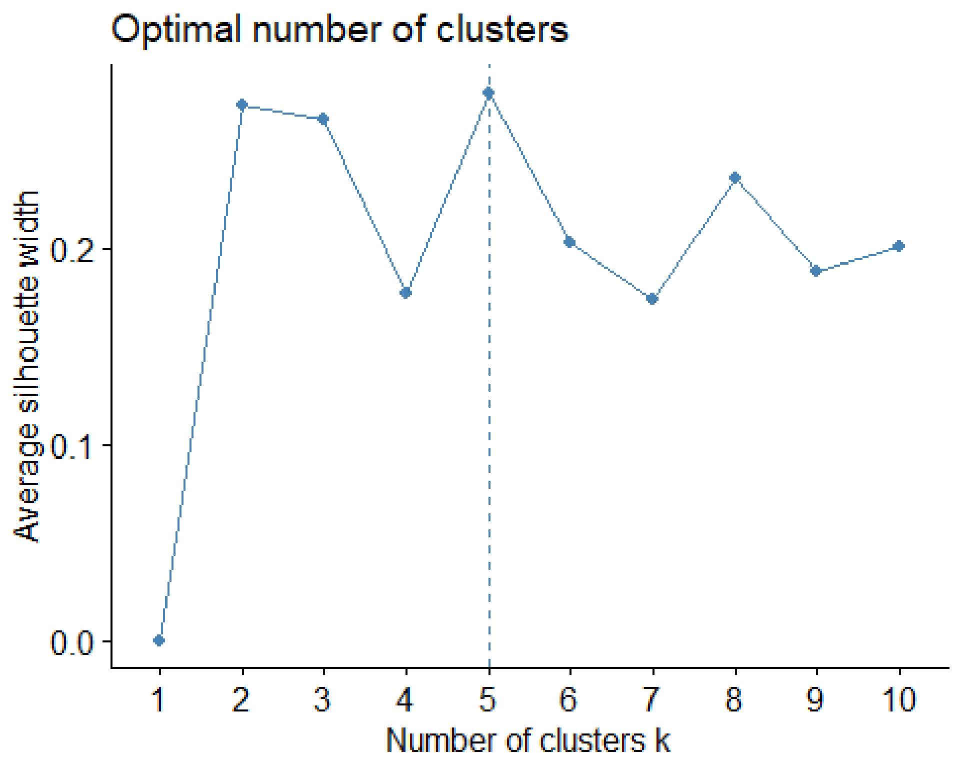
##So far it looks like k=4 is the best option as it disperses the data somewhat evenly without splitting the data too drastically (a cluster of 1 or 2 observations doesn't seem to be helpful)

```
fviz_nbclust(df, kmeans, method = "wss")
```



##Based on the elbow method graph, it looks like K = 4, 5, or 6 is the optimal number of clusters

```
fviz_nbclust(df, kmeans, method = "silhouette")
```



##Based on the silhouette method, K=5 is the optimal number of clusters. We will use k=5 ##I will use k=5 and now use the manhattan distance for clustering the data

```
library(flexclust)

## Warning: package 'flexclust' was built under R version 4.1.3

## Loading required package: grid

## Loading required package: lattice

## Loading required package: modeltools

## Warning: package 'modeltools' was built under R version 4.1.1

## Loading required package: stats4

set.seed(123)
k5.manhattan = kcca(df, k=5, kccaFamily("kmedians"))
k5

## K-means clustering with 5 clusters of sizes 4, 2, 3, 8, 4
##
## Cluster means:
##    Market_Cap       Beta    PE_Ratio        ROE        ROA Asset_Turnover
## 1 -0.76022489  0.2796041 -0.47742380 -0.7438022 -0.8107428     -1.2684804
## 2 -0.43925134 -0.4701800  2.70002464 -0.8349525 -0.9234951      0.2306328
## 3 -0.87051511  1.3409869 -0.05284434 -0.6184015 -1.1928478     -0.4612656
## 4 -0.03142211 -0.4360989 -0.31724852  0.1950459  0.4083915      0.1729746
```

```
## 5  1.69558112 -0.1780563 -0.19845823  1.2349879  1.3503431      1.1531640
##      Leverage Rev_Growth Net_Profit_Margin
## 1  0.06308085  1.5180158      -0.006893899
## 2 -0.14170336 -0.1168459      -1.416514761
## 3  1.36644699 -0.6912914      -1.320000179
## 4 -0.27449312 -0.7041516       0.556954446
## 5 -0.46807818  0.4671788       0.591242521
##
## Clustering vector:
##  [1] 4 2 4 4 1 3 4 3 1 4 5 3 5 1 5 4 5 2 4 1 4
##
## Within cluster sum of squares by cluster:
## [1] 12.791257  2.803505 15.595925 21.879320  9.284424
##  (between_SS / total_SS =  65.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

##Based on our two models, it seems that the firms in cluster 5 are the top performers.

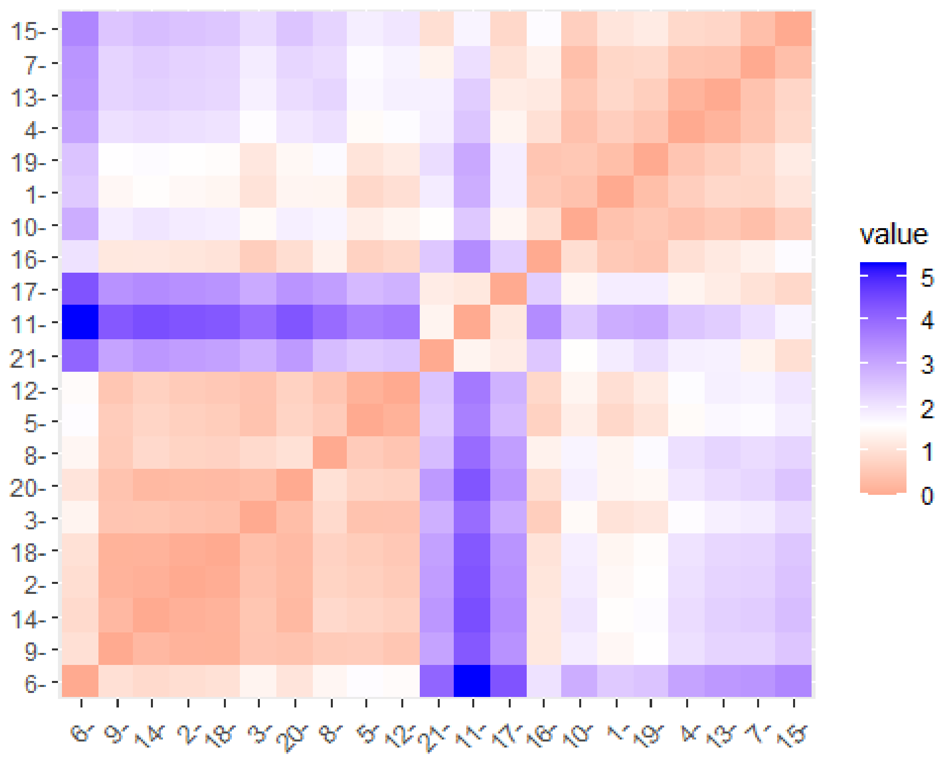
##Now I will examine 2 specific attributes - ROE & ROA - The higher the ROE & ROA are, the better the firm is performing and will probably continue to perform well.

```
ROE.ROA.DF=read.csv("C:/Users/hdesi/Desktop/MBA/Machine
Learning/Pharmaceuticals.csv")[,c(6,7)]
set.seed(123)
summary(ROE.ROA.DF)
```

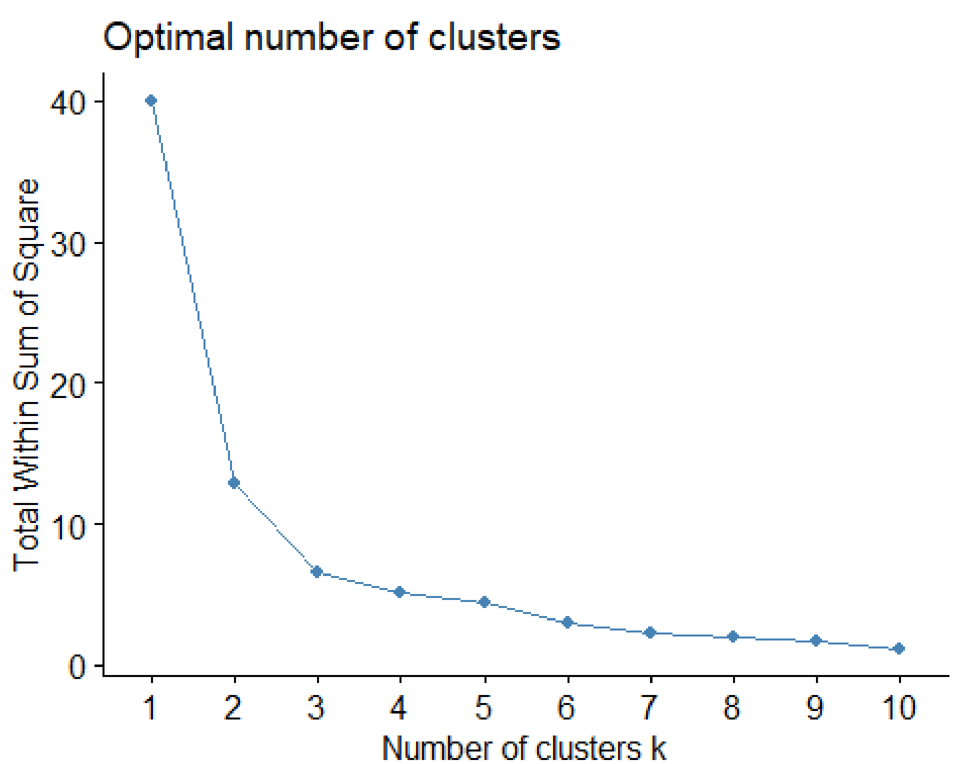
```
##       ROE             ROA
##  Min.   : 3.9   Min.   : 1.40
##  1st Qu.:14.9   1st Qu.: 5.70
##  Median :22.6   Median :11.20
##  Mean   :25.8   Mean   :10.51
##  3rd Qu.:31.0   3rd Qu.:15.00
##  Max.   :62.9   Max.   :20.30
```

##Looking for the best k value

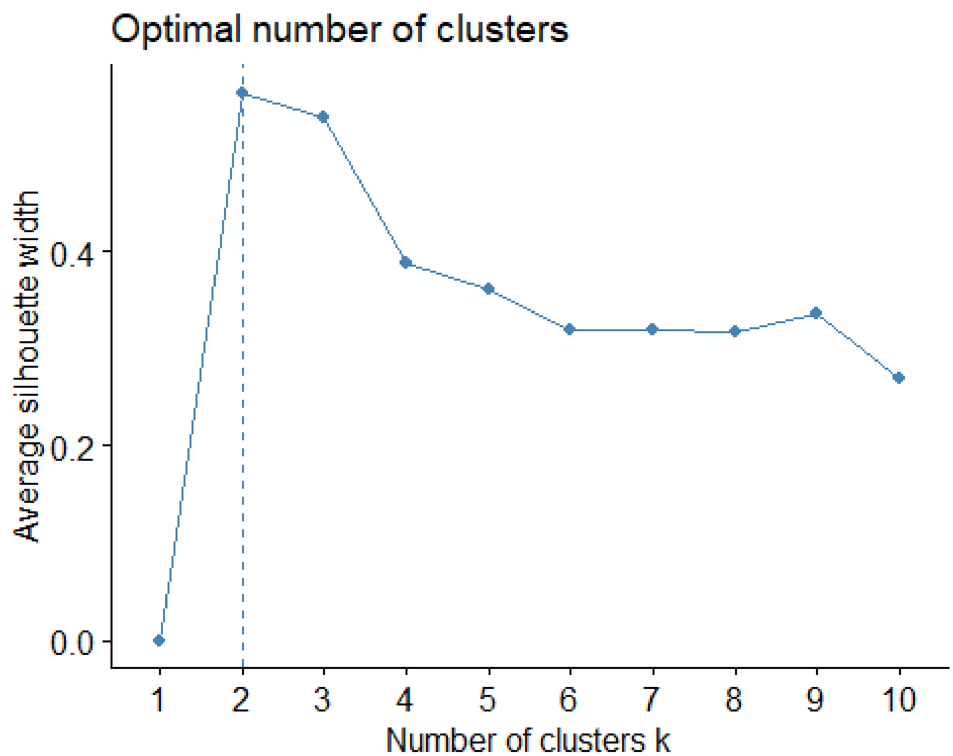
```
ROE.ROA.DF <- scale(ROE.ROA.DF) #z-score
distance <- get_dist(ROE.ROA.DF)
fviz_dist(distance)
```




```
fviz_nbclust(ROE.ROA.DF, kmeans, method = "wss")##elbow
```

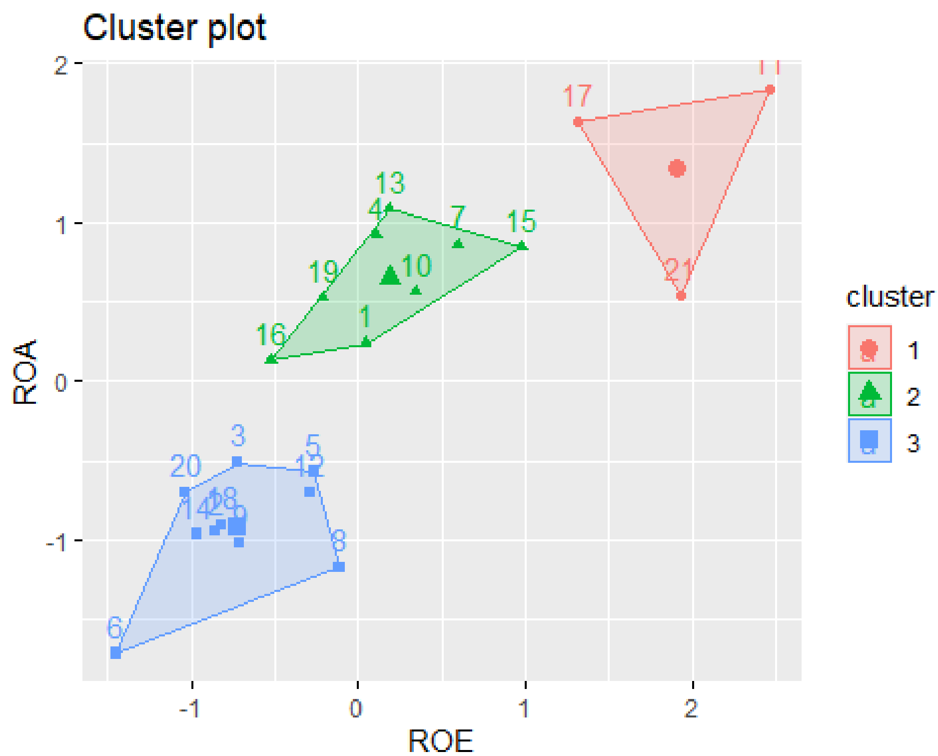



```
fviz_nbclust(ROE.ROA.DF, kmeans, method = "silhouette")
```



##Based on both methods, k=2 or k=3 is optimal - we will use 3 as 2 is too insignificant

```
ROA.ROE.PERFORMANCE <- kmeans(ROE.ROA.DF, centers = 3, nstart = 25)
ROA.ROE.PERFORMANCE$centers

##          ROE        ROA
## 1  1.9006613  1.3378151
## 2  0.1900740  0.6456412
## 3 -0.7222576 -0.9178575

ROA.ROE.PERFORMANCE$size

## [1]  3  8 10

## 3, 8, 10

fviz_cluster(ROA.ROE.PERFORMANCE, data = ROE.ROA.DF) ##Visual
```



##Firms in cluster 1 have the highest ROA and ROE - data points 11,17, & 21 ##Best: 11 - GlaxoSmithKline plc ##Also High Performers: 17 - Pfizer Inc 21 - Wyeth

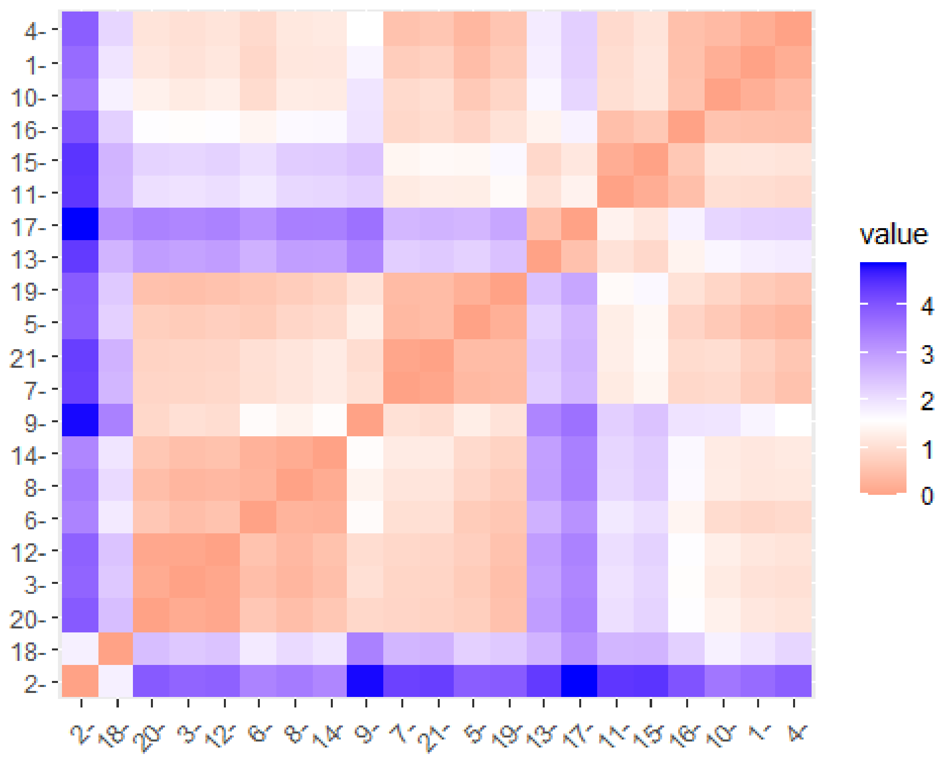
##We will examine the market capitalization and the price to earnings ratio to determine the worth of a firm (in terms of investing).

```
FIRM.WORTH.DF=read.csv("C:/Users/hdesi/Desktop/MBA/Machine
Learning/Pharmaceuticals.csv")[,c(3,5)]
set.seed(123)
summary(FIRM.WORTH.DF)
```

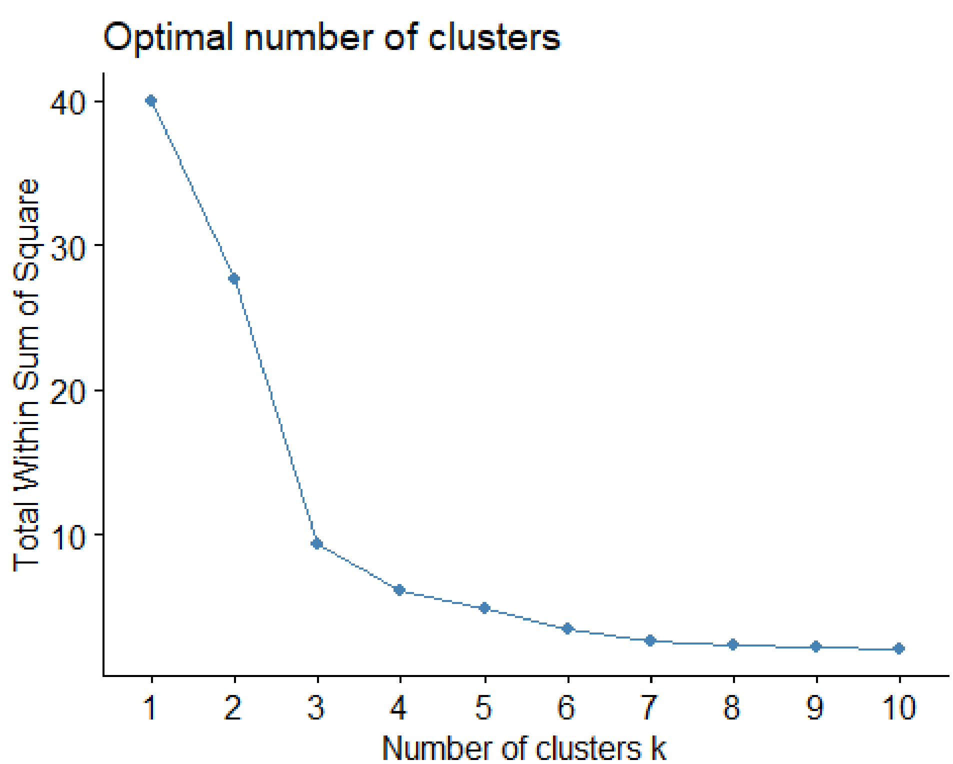
```
##    Market_Cap        PE_Ratio
##  Min.   :  0.41   Min.   : 3.60
##  1st Qu.:  6.30   1st Qu.:18.90
##  Median : 48.19   Median :21.50
##  Mean   : 57.65   Mean   :25.46
##  3rd Qu.: 73.84   3rd Qu.:27.90
##  Max.   :199.47   Max.   :82.50
```

##Looking for the best k value

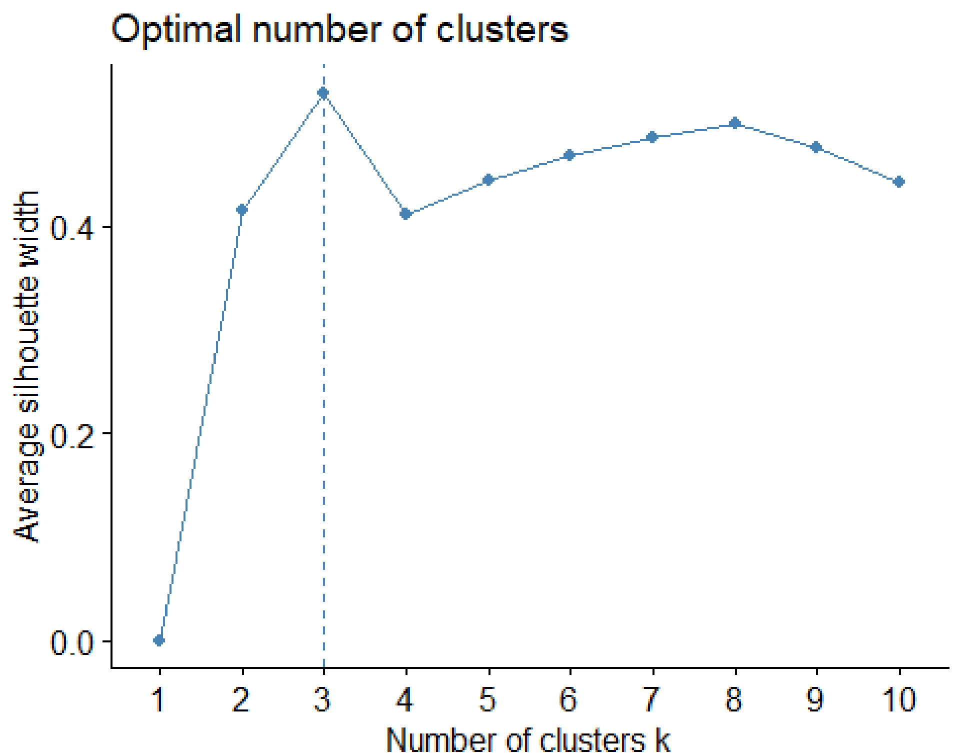
```
FIRM.WORTH.DF <- scale(FIRM.WORTH.DF) #z-score
distance <- get_dist(FIRM.WORTH.DF)
fviz_dist(distance)
```




```
fviz_nbclust(FIRM.WORTH.DF, kmeans, method = "wss")##elbow
```




```
fviz_nbclust(FIRM.WORTH.DF, kmeans, method = "silhouette")
```



##Based on both methods, k=3 is optimal

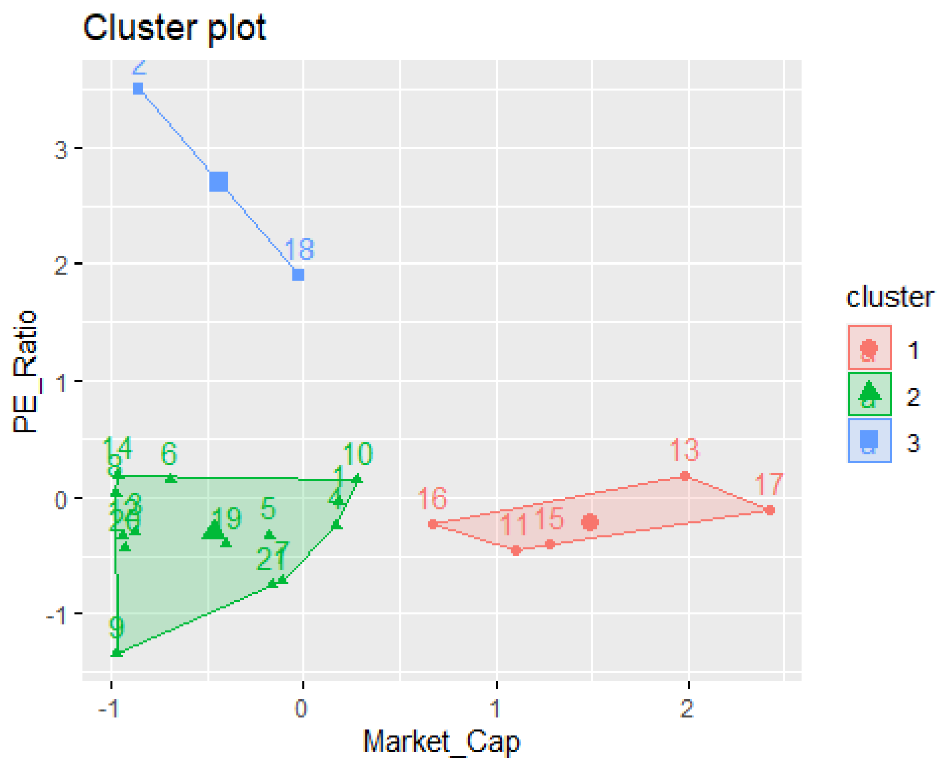
```
FIRM.WORTH <- kmeans(FIRM.WORTH.DF, centers = 3, nstart = 25)
FIRM.WORTH$centers
```

```
##    Market_Cap   PE_Ratio
## 1  1.4895591 -0.2061221
## 2 -0.4692352 -0.3121028
## 3 -0.4392513  2.7000246
```

```
FIRM.WORTH$size
```

```
## [1]  5 14  2
```

```
## 5, 14, 2

fviz_cluster(FIRM.WORTH, data = FIRM.WORTH.DF) ##Visual
```



##A high PE Ratio can be seen as good or bad, depending. Since these are established firms rather than a startup, we want a lower PE Ratio indicating that we are not overpaying for the value of the stock. A higher market capitalization is always better ##The firm with the best stock value (you should invest) is 13 - Johnson & Johnson (in my opinion) ##Options in cluster 1 have the best stock value

##Are other attributes are valuable to look at, but the 4 I have chosen to concentrate on will lead to the best odds of high performance if stock is purchased. Looking into the other attributes may cloud the waters.

##If only one stock can be purchased than the best option would be Pfizer Inc who is located in high performing ares in their clusters of both segmentations.