Assignment_4

Kevin Gardner

Due 3/20/2022

——————————————————

Following is the link to my GitHub account:

https://github.com/Kgardner22/64060_-kgardner

——————————————————

IMPORT AND PREPARE DATA:

Import the Pharmaceuticals.csv file

```
Pharmaceuticals <- read.table('C:/R/MyData/Pharmaceuticals.csv', header = T,
sep = ',')

summary(Pharmaceuticals)

##     Symbol              Name             Market_Cap         Beta
##  Length:21          Length:21          Min.   :  0.41   Min.   :0.1800
##  Class :character   Class :character   1st Qu.:  6.30   1st Qu.:0.3500
##  Mode  :character   Mode  :character   Median : 48.19   Median :0.4600
##                                        Mean   : 57.65   Mean   :0.5257
##                                        3rd Qu.: 73.84   3rd Qu.:0.6500
##                                        Max.   :199.47   Max.   :1.1100
##     PE_Ratio          ROE             ROA        Asset_Turnover    Leverage
##  Min.   : 3.60   Min.   : 3.9   Min.   : 1.40   Min.   :0.3    Min.
:0.0000
##  1st Qu.:18.90   1st Qu.:14.9   1st Qu.: 5.70   1st Qu.:0.6    1st
Qu.:0.1600
##  Median :21.50   Median :22.6   Median :11.20   Median :0.6    Median
:0.3400
##  Mean   :25.46   Mean   :25.8   Mean   :10.51   Mean   :0.7    Mean
:0.5857
##  3rd Qu.:27.90   3rd Qu.:31.0   3rd Qu.:15.00   3rd Qu.:0.9    3rd
Qu.:0.6000
##  Max.   :82.50   Max.   :62.9   Max.   :20.30   Max.   :1.1    Max.
:3.5100
```

```
##    Rev_Growth     Net_Profit_Margin Median_Recommendation   Location
##  Min.   :-3.17   Min.   : 2.6      Length:21             Length:21
##  1st Qu.: 6.38   1st Qu.:11.2      Class :character      Class :character
##  Median : 9.37   Median :16.1      Mode  :character      Mode  :character
##  Mean   :13.37   Mean   :15.7
##  3rd Qu.:21.87   3rd Qu.:21.1
##  Max.   :34.21   Max.   :25.5
##    Exchange
##  Length:21
##  Class :character
##  Mode  :character
##
##
##
```

Load required libraries

```
library(tidyverse)  #for data manipulation
library(factoextra)  #for clustering and visualization
library(flexclust)
```

Use cluster analysis to explore and analyze the given dataset as follows:

REQUIREMENT A:

Use only the numerical variables (1 to 9) to cluster the 21 firms. Justify the various choices made in conducting the cluster analysis, such as weights for different variables, the specific clustering algorithm(s) used, the number of clusters formed, and so on.

First, we create the data.frame with only the numerical variables 1 to 9

```
set.seed(64060)

df <- Pharmaceuticals[,c(3:11)]

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
```

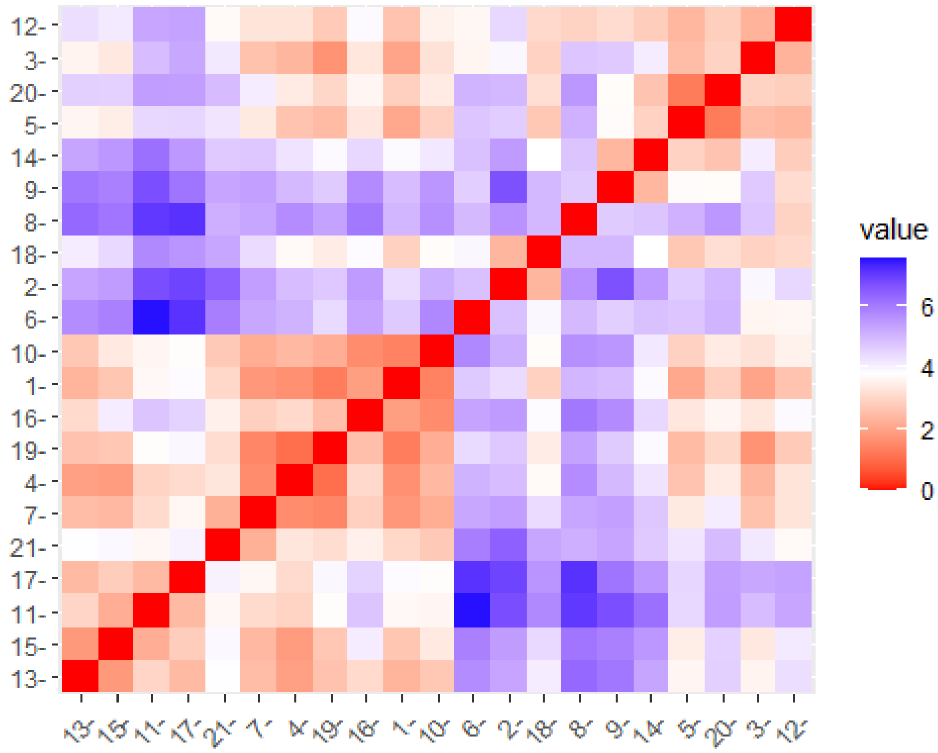
```
##  Net_Profit_Margin
##  Min.   : 2.6
##  1st Qu.:11.2
##  Median :16.1
##  Mean   :15.7
##  3rd Qu.:21.1
##  Max.   :25.5
```

Before we can begin cluster analysis, we must first scale the data.

```
# Scaling the data frame (z-score)
df <- scale(df)

summary(df)

##    Market_Cap           Beta            PE_Ratio            ROE
##  Min.   :-0.9768   Min.   :-1.3466   Min.   :-1.3404   Min.   :-1.4515
##  1st Qu.:-0.8763   1st Qu.:-0.6844   1st Qu.:-0.4023   1st Qu.:-0.7223
##  Median :-0.1614   Median :-0.2560   Median :-0.2429   Median :-0.2118
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2762   3rd Qu.: 0.4841   3rd Qu.: 0.1495   3rd Qu.: 0.3450
##  Max.   : 2.4200   Max.   : 2.2758   Max.   : 3.4971   Max.   : 2.4597
##       ROA          Asset_Turnover       Leverage          Rev_Growth
##  Min.   :-1.7128   Min.   :-1.8451   Min.   :-0.74966   Min.   :-1.4971
##  1st Qu.:-0.9047   1st Qu.:-0.4613   1st Qu.:-0.54487   1st Qu.:-0.6328
##  Median : 0.1289   Median :-0.4613   Median :-0.31449   Median :-0.3621
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.8430   3rd Qu.: 0.9225   3rd Qu.: 0.01828   3rd Qu.: 0.7693
##  Max.   : 1.8389   Max.   : 1.8451   Max.   : 3.74280   Max.   : 1.8862
##  Net_Profit_Margin
##  Min.   :-1.99560
##  1st Qu.:-0.68504
##  Median : 0.06168
##  Mean   : 0.00000
##  3rd Qu.: 0.82364
##  Max.   : 1.49416
```

We'll compute and visualize the distance matrix between rows using get_dist() and fviz_dist()

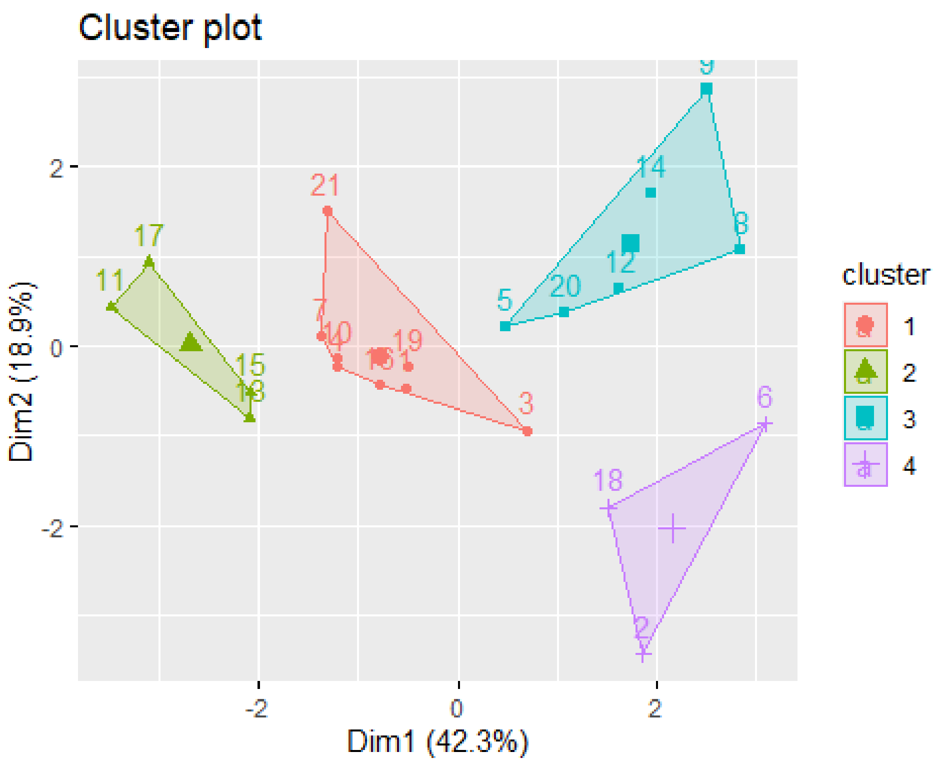
```
distance <- get_dist(df)
fviz_dist(distance)
```



The above graph shows the distance between firms.

I'll run the k-means algorithm to cluster the firms, choosing an initial random value of k = 4.

```
set.seed(64060)
k4 <- kmeans(df, centers = 4, nstart = 25) # k = 4, number of restarts = 25

# the following will help us Visualize the output

k4$centers # output the centers

##    Market_Cap       Beta   PE_Ratio        ROE        ROA Asset_Turnover
## 1 -0.03142211 -0.4360989 -0.3172485  0.1950459  0.4083915   1.729746e-01
## 2  1.69558112 -0.1780563 -0.1984582  1.2349879  1.3503431   1.153164e+00
## 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 -0.2744931 -0.7041516         0.5569544
## 2 -0.4680782  0.4671788         0.5912425
## 3  0.8502201  0.9158889        -0.3319956
## 4 -0.3443544 -0.5769454        -1.6095439

k4$size # Number of firms in each cluster

## [1] 8 4 6 3

fviz_cluster(k4, data = df) # Visualize the output
```



This produces similar size clusters (8,4,6,3)

Other Distances

I'll rerun the example using other distance measures to compare the results

```
set.seed(64060)

k4M = kcca(df, k=4, kccaFamily("kmedians"))  #kmedians uses Manhattan
distance
k4M

## kcca object of family 'kmedians'
##
## call:
## kcca(x = df, k = 4, family = kccaFamily("kmedians"))
##
## cluster sizes:
##
##  1  2  3  4
##  3 12  5  1

k4E = kcca(df, k=4, kccaFamily("kmeans"))  #kmeans uses Euclidean distance
k4E

## kcca object of family 'kmeans'
##
```

```
## call:
## kcca(x = df, k = 4, family = kccaFamily("kmeans"))
##
## cluster sizes:
##
##  1  2  3  4
## 12  4  2  3
```

```
k4A = kcca(df, k=4, kccaFamily("angle"))  #angle uses angle between
observation and centroid
k4A
```

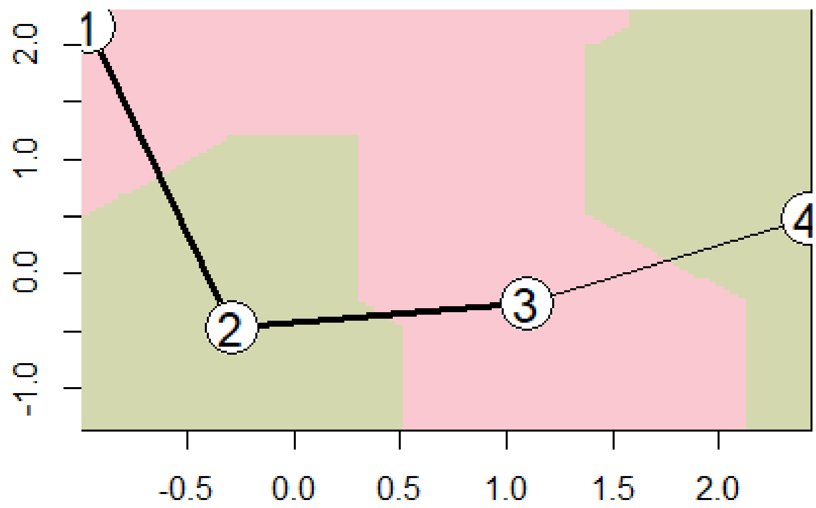
```
## kcca object of family 'angle'
##
## call:
## kcca(x = df, k = 4, family = kccaFamily("angle"))
##
## cluster sizes:
##
##  1  2  3  4
##  4  4 11  2
```

```
# We won't use Jaccard distance as this is primarily used for categorical
data which is not applicable.
```

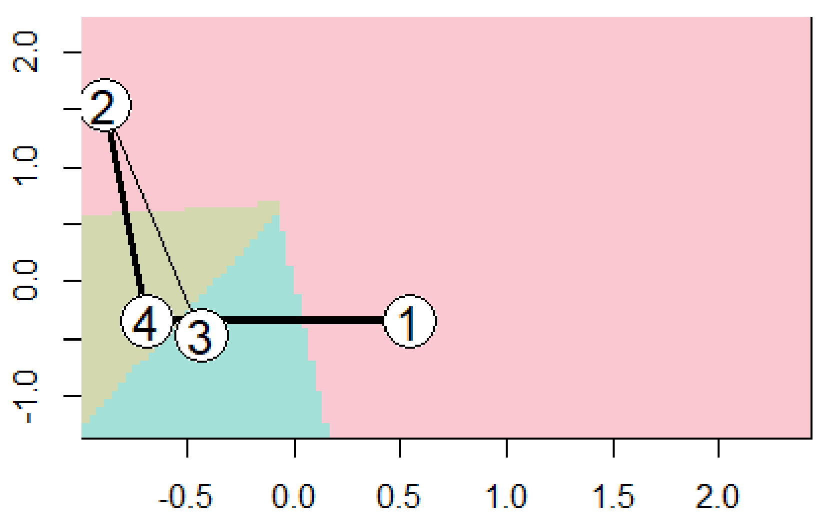
In reviewing these results, we see the cluster sizes using kmedians (Manhattan), kmeans (Euclidean) and angle produce similar results.

Let's take a look at the images of each of these results:

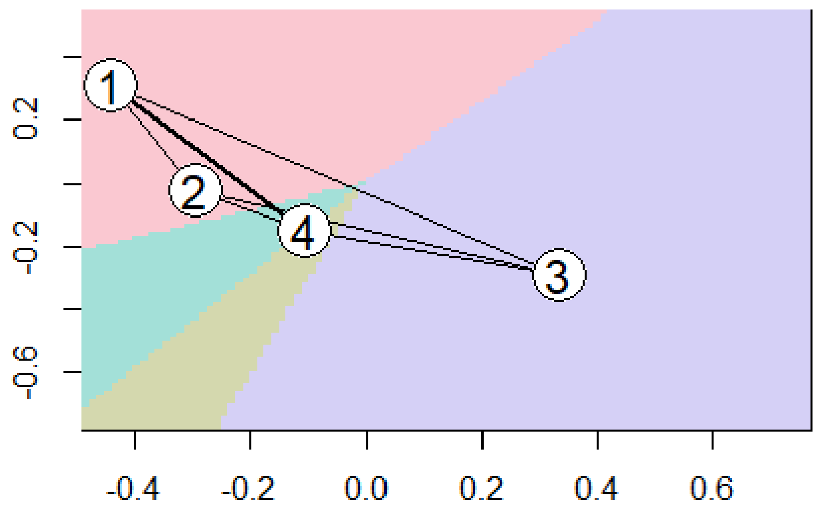
```
image(k4M) # Manhattan distance
```




```
image(k4E) # Euclidean distance
```




```
image(k4A) # angle
```



These images clearly show the various distance measures make a huge difference in the clustering results. It's clear that k4M, which is using Manhattan distance (kmedians), produces better results since the distance between the centroids is maximized in comparison to the other results.

This is also confirmed by looking at the centers:

```
dist(k4M@centers)

##          1        2        3
## 2 3.838654
## 3 5.233264 2.754097
## 4 6.025978 4.677079 2.397111

dist(k4E@centers)

##          1        2        3
## 2 4.169374
## 3 4.306065 4.093207
## 4 3.319190 3.018190 3.847236

dist(k4A@centers)

##          1        2        3
## 2 1.253136
## 3 1.872258 1.823885
## 4 1.158538 1.373808 1.743565
```

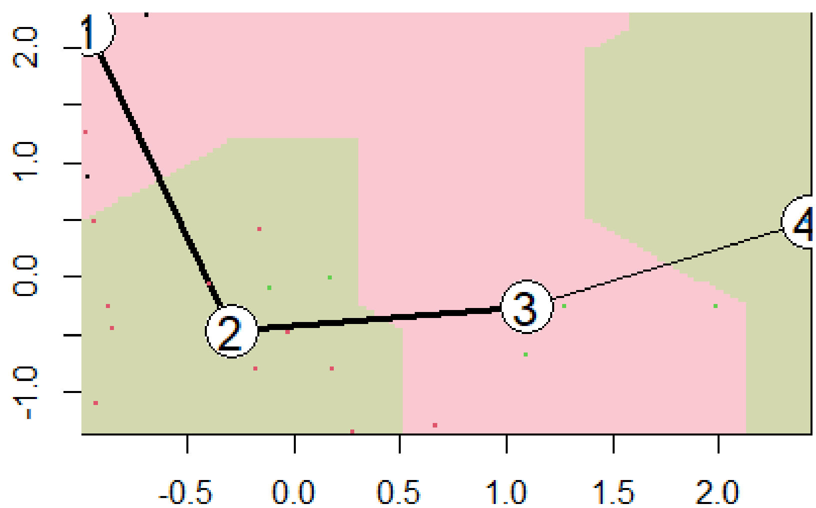
As shown in this data, the distance measure producing the maximum distance between centroids is Manhattan distance (k4M)

I'll now apply the predict function to k4M which uses Manhattan Distance

```
set.seed(64060)
clusters_index4 <- predict(k4M)
dist(k4M@centers)
```

```
##          1        2        3
## 2 3.838654
## 3 5.233264 2.754097
## 4 6.025978 4.677079 2.397111
```

```
image(k4M)
points(df, col=clusters_index4, pch=19, cex=0.3)
```



But is a K of 4 really the best choice? After all, this was just a random choice. Let's use a K of 3 and examine the results.

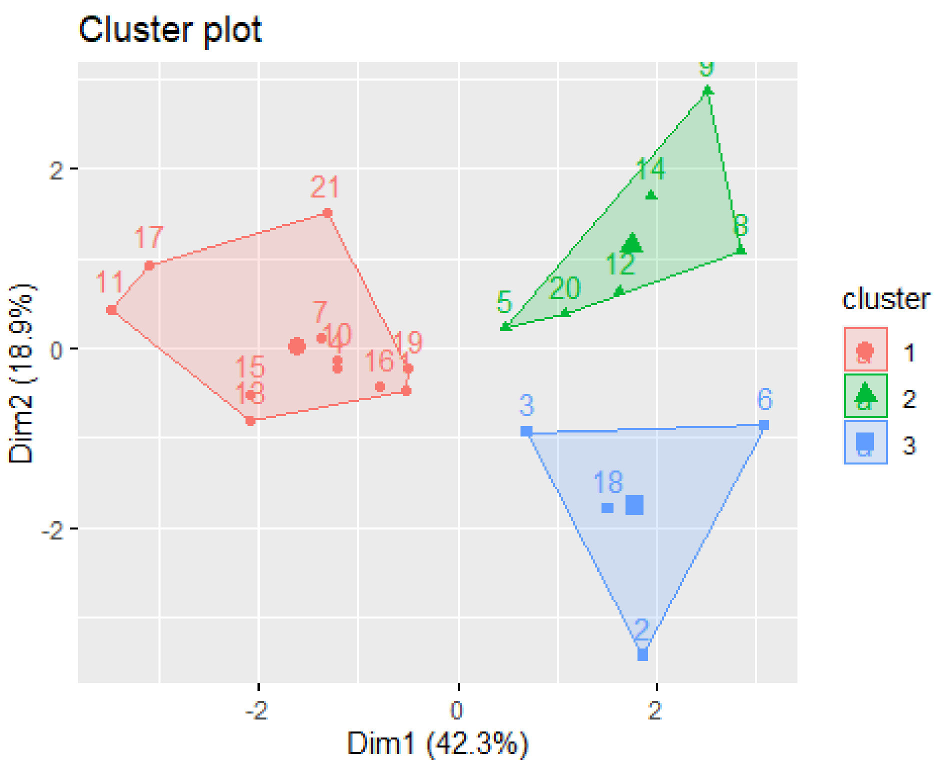
```
set.seed(64060)
k3 <- kmeans(df, centers = 3, nstart = 25) # k = 3, number of restarts = 25

# the following will help us Visualize the output

k3$centers # output the centers
```

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

```
k3$size # Number of firms in each cluster
```

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

```
fviz_cluster(k3, data = df) # Visualize the output
```



Other Distances

I'll rerun the example using other distance measures to compare the results

```
set.seed(64060)

k3M = kcca(df, k=3, kccaFamily("kmedians"))  #kmedians uses Manhattan
distance
k3M

## kcca object of family 'kmedians'
##
## call:
## kcca(x = df, k = 3, family = kccaFamily("kmedians"))
##
```

```
## cluster sizes:
##
##  1  2  3
##  3 12  6
```

```
k3E = kcca(df, k=3, kccaFamily("kmeans"))  #kmeans uses Euclidean distance
k3E
```

```
## kcca object of family 'kmeans'
##
## call:
## kcca(x = df, k = 3, family = kccaFamily("kmeans"))
##
## cluster sizes:
##
##  1  2  3
## 10  7  4
```

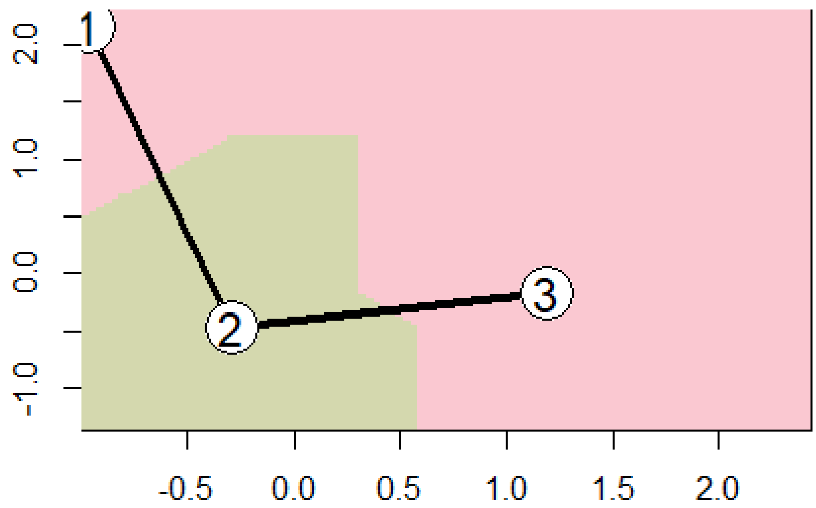
```
k3A = kcca(df, k=3, kccaFamily("angle"))  #angle uses angle between
observation and centroid
k3A
```

```
## kcca object of family 'angle'
##
## call:
## kcca(x = df, k = 3, family = kccaFamily("angle"))
##
## cluster sizes:
##
##  1  2  3
##  6  4 11
```

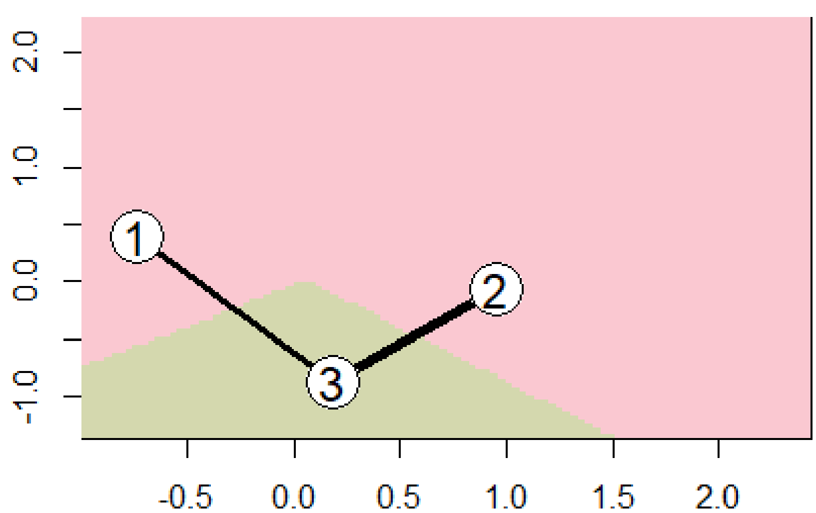
```
# We won't use Jaccard distance as this is primarily used for categorical
data which is not applicable.
```

Let's take a look at the images of each of these results:

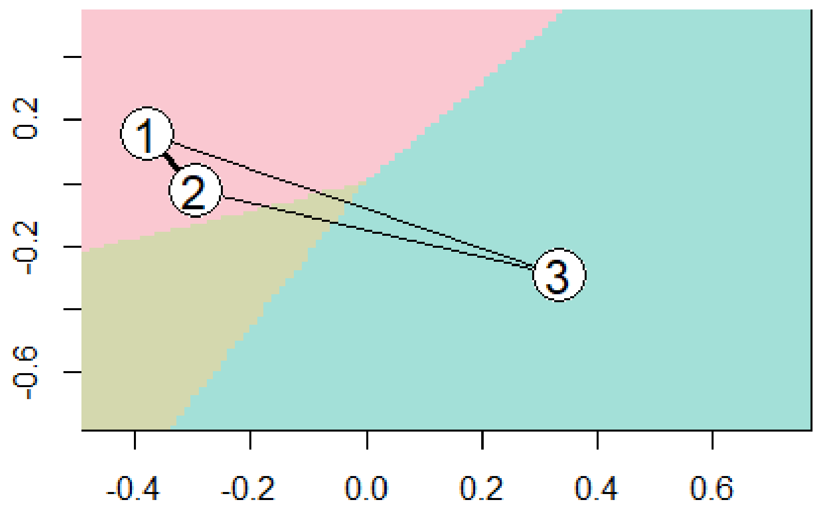
```
image(k3M) # Manhattan distance
```




```r
image(k3E) # Euclidean distance
```




```r
image(k3A) # angle
```



Once again, using Manhattan distance produces better results with the clustering. There is a greater distance between centroids of the clusters than using Euclidean or Angle.

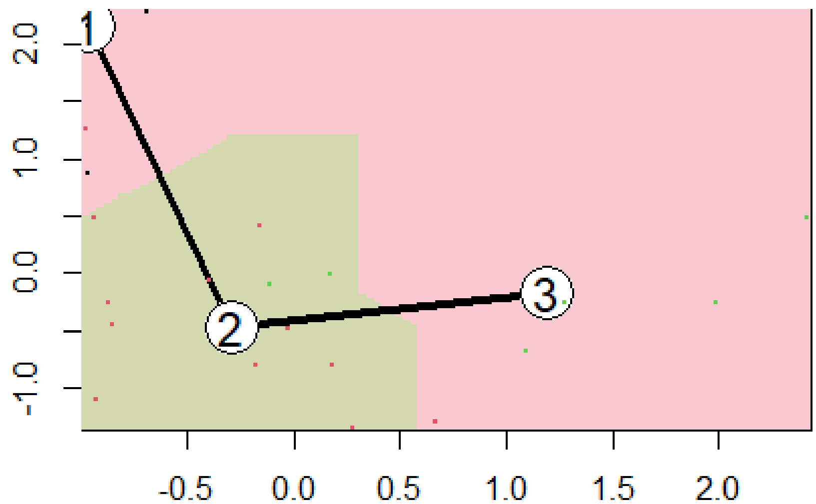
This is also confirmed by looking at the centers:

```
dist(k3M@centers)

##          1        2
## 2 3.838654
## 3 5.334439 2.990655

dist(k3E@centers)

##          1        2
## 2 3.921756
## 3 2.918545 2.236172

dist(k3A@centers)

##          1        2
## 2 1.276399
## 3 1.898562 1.823885
```

Let’s apply the predict function to k3M which uses Manhattan Distance

```
set.seed(64060)
clusters_index3 <- predict(k3M)
dist(k3M@centers)
```

```
##          1        2
## 2 3.838654
## 3 5.334439 2.990655

image(k3M)
points(df, col=clusters_index3, pch=19, cex=0.3)
```



Before we make any conclusions with these results, let's try a K of 5 and analyze the results:

```
set.seed(64060)
k5 <- kmeans(df, centers = 5, nstart = 25) # k = 5, number of restarts = 25

# the following will help us Visualize the output

k5$centers # output the centers

##    Market_Cap       Beta    PE_Ratio        ROE        ROA Asset_Turnover
## 1 -0.87051511  1.3409869 -0.05284434 -0.6184015 -1.1928478     -0.4612656
## 2 -0.43925134 -0.4701800  2.70002464 -0.8349525 -0.9234951      0.2306328
## 3 -0.76022489  0.2796041 -0.47742380 -0.7438022 -0.8107428     -1.2684804
## 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  1.36644699 -0.6912914      -1.320000179
## 2 -0.14170336 -0.1168459      -1.416514761
## 3  0.06308085  1.5180158      -0.006893899
```

```
## 4 -0.27449312 -0.7041516       0.556954446
## 5 -0.46807818  0.4671788       0.591242521

k5$size # Number of firms in each cluster

## [1] 3 2 4 8 4

fviz_cluster(k5, data = df) # Visualize the output
```

Other Distances

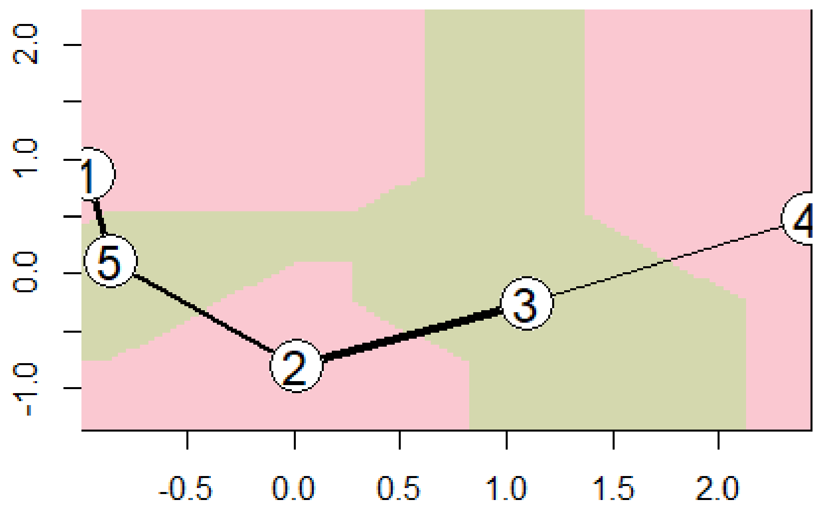
I'll rerun the example using other distance measures to compare the results

```
set.seed(64060)

k5M = kcca(df, k=5, kccaFamily("kmedians"))  #kmedians uses Manhattan
distance
k5M

## kcca object of family 'kmedians'
##
## call:
## kcca(x = df, k = 5, family = kccaFamily("kmedians"))
##
## cluster sizes:
##
```

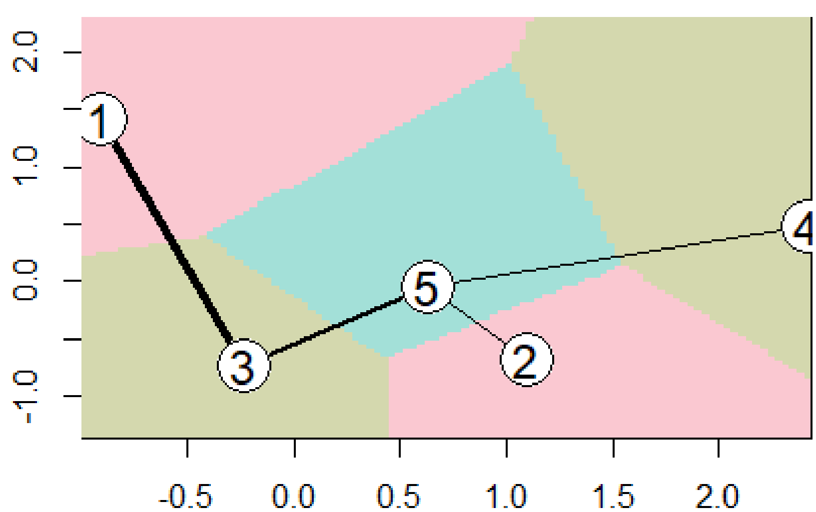
```
## 1 2 3 4 5
## 3 6 5 1 6

k5E = kcca(df, k=5, kccaFamily("kmeans"))  #kmeans uses Euclidean distance
k5E

## kcca object of family 'kmeans'
##
## call:
## kcca(x = df, k = 5, family = kccaFamily("kmeans"))
##
## cluster sizes:
##
## 1 2 3 4 5
## 5 1 9 1 5

k5A = kcca(df, k=5, kccaFamily("angle"))  #angle uses angle between
observation and centroid
k5A

## kcca object of family 'angle'
##
## call:
## kcca(x = df, k = 5, family = kccaFamily("angle"))
##
## cluster sizes:
##
## 1 2 3 4 5
## 8 4 4 3 2

# We won't use Jaccard distance as this is primarily used for categorical
data which is not applicable.
```

Let’s take a look at the images of each of these results:

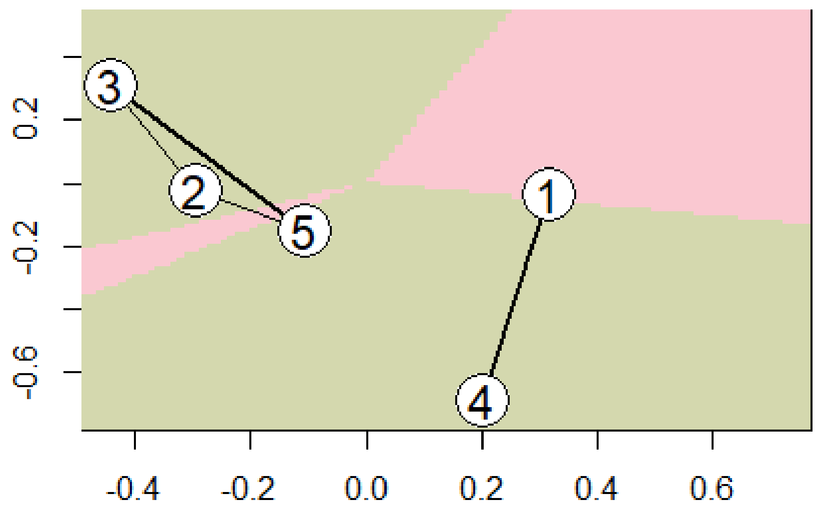
```
image(k5M) # Manhattan distance
```




```
image(k5E) # Euclidean distance
```




```
image(k5A) # angle
```



This is a little more interesting. It seems with the K of 5, the Euclidean distance may be producing a better result

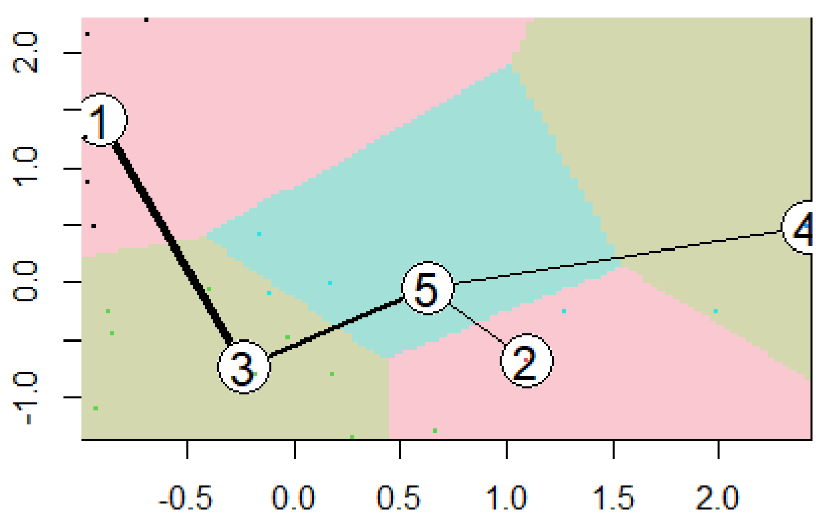
Let's take a closer look at the centers:

```
dist(k5M@centers)

##          1        2        3        4
## 2 3.721628
## 3 4.689876 2.194249
## 4 5.698767 3.925905 2.397111
## 5 2.931609 2.762659 3.804627 5.718298

dist(k5E@centers)

##          1        2        3        4
## 2 6.108448
## 3 3.091631 4.400111
## 4 5.789244 2.447177 4.465604
## 5 4.177287 2.502227 2.448221 2.791316

dist(k5A@centers)

##          1        2        3        4
## 2 1.822650
## 3 1.840575 1.253136
## 4 1.163136 1.635111 1.732785
## 5 1.776303 1.373808 1.158538 1.508880
```

It appears that the Euclidean distance (k5E) is producing better results as can be seen in both the image and the data for the centers. This seems to maximize the distance between the cluster centroids.

Let's apply the predict function to k5E which uses Euclidean Distance

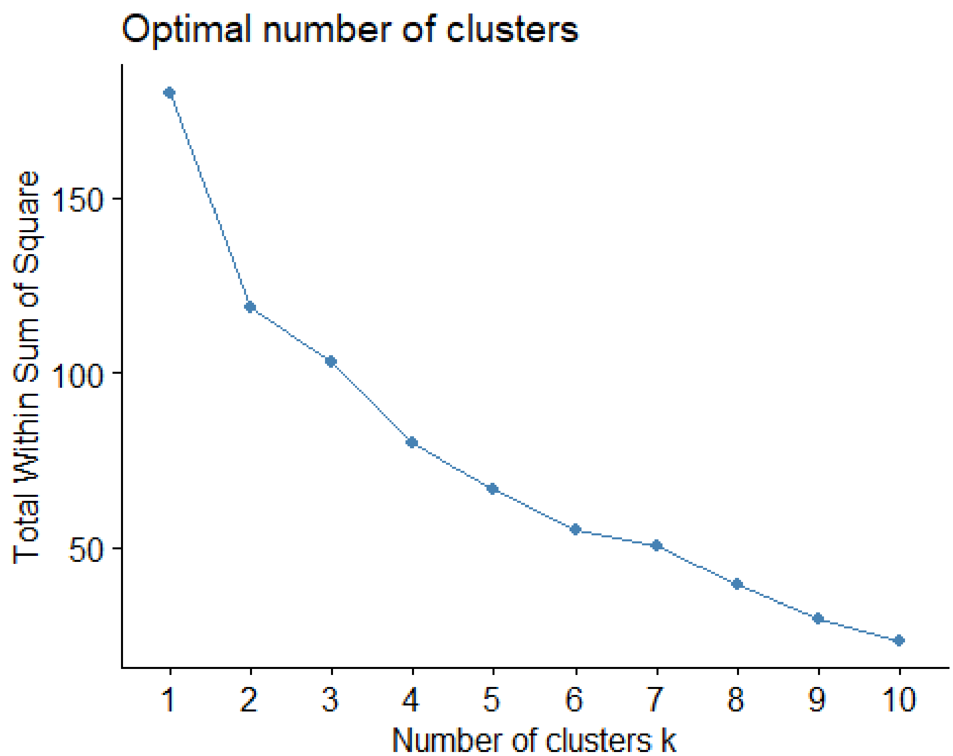
```
set.seed(64060)
clusters_index5 <- predict(k5E)
dist(k5E@centers)

##          1        2        3        4
## 2 6.108448
## 3 3.091631 4.400111
## 4 5.789244 2.447177 4.465604
## 5 4.177287 2.502227 2.448221 2.791316

image(k5E)
points(df, col=clusters_index5, pch=19, cex=0.3)
```



CHOOSING THE BEST K Let's use some tools to help us determine the best K value

We'll first review an "elbow chart" to determine k

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



In reviewing the "elbow chart" (WSS), it is a little unclear as to what the optimal number of clusters (k) should be since there is not a clearly visible "elbow" in the results plot. The total WSS has a substantial drop from 1 to 2, less of a drop from 2 to 3, and then another larger drop from 3 to 4. From 4 to 5 and 5 to 6, the decrease is similar and then from 6 to 7, there is little drop. My first thought is the elbow is either at 4 or 6. Honestly, it is difficult to make a reliable determination of the optimal number of clusters (k) using this method (WSS) for this particular set of data. Therefore, I need to confirm this using a different method (Silhouette Method).

Next, we'll use the Silhouette Method to determine the number of clusters (k)

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

In using the Silhouette Method, it is clear that the optimal number of clusters (k) is 5.

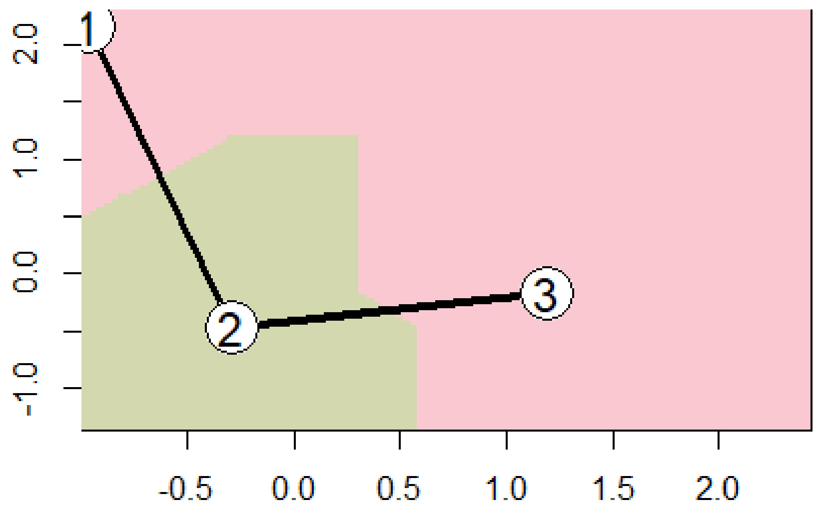
SUMMARY OF REQUIREMENT A:

When clustering, our objective is to minimize the similarity within the cluster and maximize the dissimilarity between the clusters. Meaning, we want the clusters to be as tight as possible with the distance between the clusters to be as great as possible. Also, it is preferable to have the size of the clusters similar.

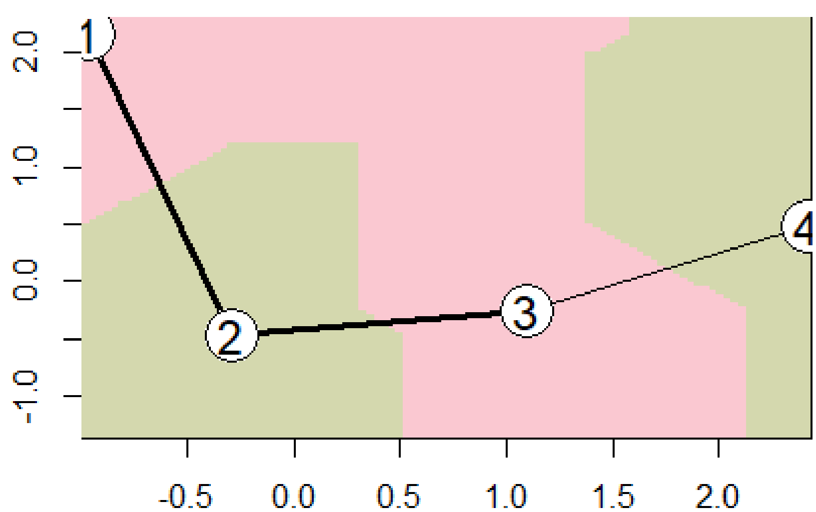
```
k3$size

## [1] 11  6  4

k4$size

## [1] 8 4 6 3

k5$size

## [1] 3 2 4 8 4
```

We have 21 observations. For 3 clusters, the average size would be 7. For 4 clusters, the average size would be 5.25. For 5 clusters, the average size would be 4.2. In reviewing the results of our cluster sizes, the results of our 5 clusters seem to remain closer to the average.

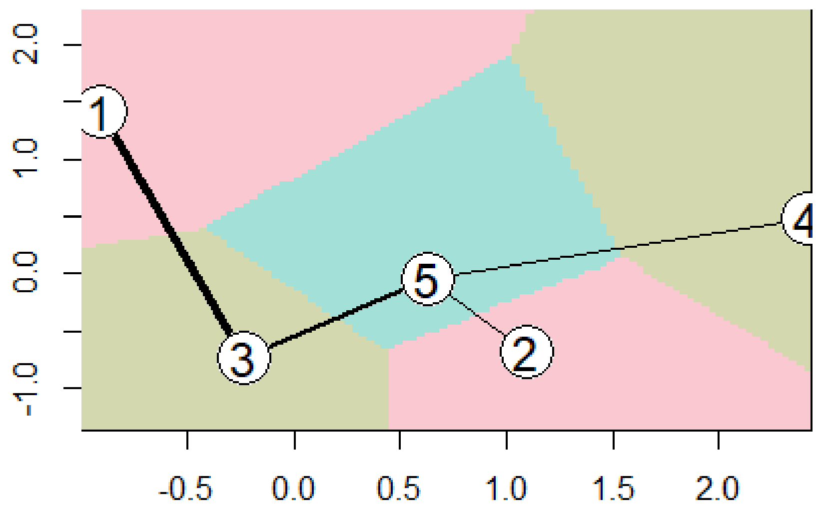
```
image(k3M)
```




```
image(k4M)
```




```
image(k5E)
```



In reviewing these images, the image with the k of 5 seems to be producing more desirable results. The clusters are “tighter” (more tightly grouped) with the distance between clusters maximized.

```
dist(k3M@centers)

##          1        2
## 2 3.838654
## 3 5.334439 2.990655

dist(k4M@centers)

##          1        2        3
## 2 3.838654
## 3 5.233264 2.754097
## 4 6.025978 4.677079 2.397111

dist(k5E@centers)

##          1        2        3        4
## 2 6.108448
## 3 3.091631 4.400111
## 4 5.789244 2.447177 4.465604
## 5 4.177287 2.502227 2.448221 2.791316
```

The above statement is also confirmed by reviewing the centers

Using the silhouette method in determining k, it confirmed the optimal k value is 5.

WEIGHTING THE VARIABLES In reviewing the data, it's logical to think Market_Cap and PE_Ratio would be more substantial in differentiating the various firms. Going with this assumption, let's place more weight on these variables than on the others.

Let's create the data frame we'll use for the weighted results

```
set.seed(64060)

df_weighted <- Pharmaceuticals[,c(3:11)]

summary(df_weighted)
```

```
##    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
```

We need to scale this data frame before we can proceed.

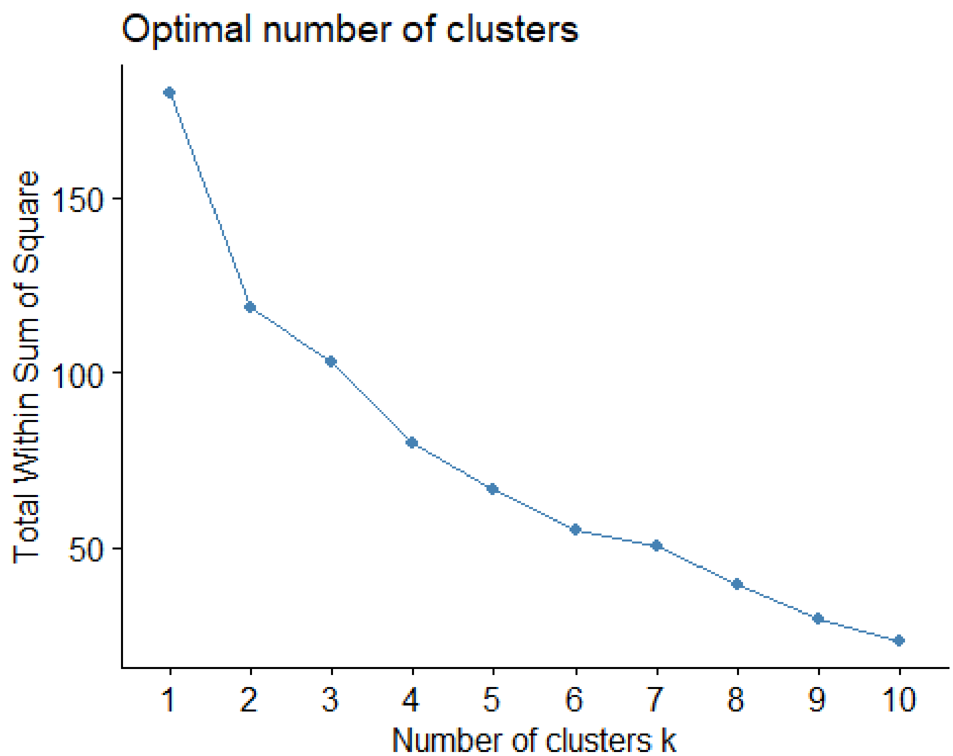
```
# Scaling the data frame (z-score)
df_weighted <- scale(df_weighted)

summary(df_weighted)
```

```
##    Market_Cap          Beta              PE_Ratio             ROE
##  Min.   :-0.9768   Min.   :-1.3466   Min.   :-1.3404   Min.   :-1.4515
##  1st Qu.:-0.8763   1st Qu.:-0.6844   1st Qu.:-0.4023   1st Qu.:-0.7223
##  Median :-0.1614   Median :-0.2560   Median :-0.2429   Median :-0.2118
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2762   3rd Qu.: 0.4841   3rd Qu.: 0.1495   3rd Qu.: 0.3450
##  Max.   : 2.4200   Max.   : 2.2758   Max.   : 3.4971   Max.   : 2.4597
##       ROA          Asset_Turnover       Leverage            Rev_Growth
##  Min.   :-1.7128   Min.   :-1.8451   Min.   :-0.74966   Min.   :-1.4971
##  1st Qu.:-0.9047   1st Qu.:-0.4613   1st Qu.:-0.54487   1st Qu.:-0.6328
```

```
##   Median : 0.1289   Median :-0.4613   Median :-0.31449   Median :-0.3621
##   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##   3rd Qu.: 0.8430   3rd Qu.: 0.9225   3rd Qu.: 0.01828   3rd Qu.: 0.7693
##   Max.   : 1.8389   Max.   : 1.8451   Max.   : 3.74280   Max.   : 1.8862
##   Net_Profit_Margin
##   Min.   :-1.99560
##   1st Qu.:-0.68504
##   Median : 0.06168
##   Mean   : 0.00000
##   3rd Qu.: 0.82364
##   Max.   : 1.49416
```

CHOOSING THE BEST K Let's review an "elbow chart" to determine k

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



Again, these results are a little unclear, so let's defer to the Silhouette method.

Now, let's use the Silhouette Method to determine the number of clusters (k)

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

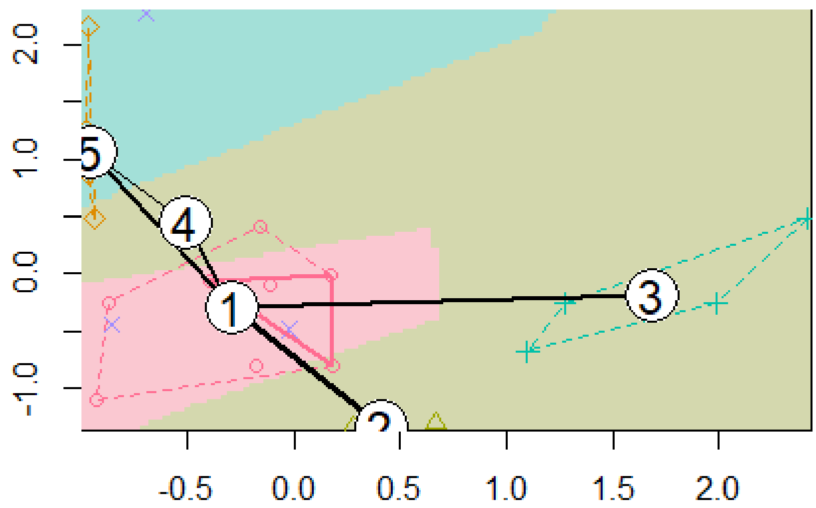
This shows the optimal number of clusters is 5

Now, we'll place more weight on Market_Cap (1st variable) and PE_Ratio (3rd variable)

```
set.seed(64060)

k5_weighted <- cclust(df_weighted, k=5, save.data=TRUE, weights =
c(1,0.5,1,0.5,0.5,0.5,0.5,0.5,0.5), method = "hardcl")

k5_weighted

## kcca object of family 'kmeans'
##
## call:
## cclust(x = df_weighted, k = 5, method = "hardcl", weights = c(1,
##     0.5, 1, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5), save.data = TRUE)
##
## cluster sizes:
##
## 1 2 3 4 5
## 8 2 4 3 4
```

Let's now visualize our results:

```
image(k5_weighted)
```




```
dist(k5_weighted@centers)

##          1        2        3        4
## 2 1.903477
## 3 2.887223 3.278427
## 4 3.498740 4.355677 5.326741
## 5 3.045587 4.188404 5.101772 3.548789
```

CONCLUSIONS: My hypothesis regarding the weight placed on each variable may be incorrect. When reviewing the images of the original k5E and the weighted k5_weighted, k5E produces better cluster results. To get better results with the weighted variables, we would need better estimations of the actual weighted importance of each variable.

REQUIREMENT B:

Interpret the clusters with respect to the numerical variables used in forming the clusters.

REQUIREMENT C:

Is there a pattern in the clusters with respect to the non-numerical variables (10 to 12)?

First, we'll add a column to Pharmaceuticals called "Cluster_No" and set the values equal to the cluster for each observation

```
Pharmaceuticals$Cluster_No = k5E@cluster
```

We'll first compare the Cluster_No to the Median_Recommendation (variable 10)

```
table(Cluster=Pharmaceuticals$Cluster_No,
Median_Recommendation=Pharmaceuticals$Median_Recommendation)

##        Median_Recommendation
## Cluster Hold Moderate Buy Moderate Sell Strong Buy
##       1    2            2             1          0
##       2    1            0             0          0
##       3    4            3             1          1
##       4    0            1             0          0
##       5    2            1             2          0
```

In reviewing the distribution of observations of the Median_Recommendation compared to each Cluster, we'll disregard clusters 2 and 4 since these have only 1 observation.

```
 Cluster 1, 40% are Hold, 40% are Moderate Buy, and 20% are Moderate Sell.
 Cluster 3, 44% are Hold, 33% are Moderate Buy, and 11% are Moderate Sell.
 Cluster 5, 40% are Hold, 20% are Moderate Buy, and 40% are Moderate Sell.
```

Therefore, there does NOT seem to be any strong correlation between this variable (variable 10) and the clusters to which they were assigned.

Next, we'll compare the Cluster_No to the Location (variable 11)

```
table(Cluster=Pharmaceuticals$Cluster_No, Location=Pharmaceuticals$Location)

##        Location
## Cluster CANADA FRANCE GERMANY IRELAND SWITZERLAND UK US
##       1      0      0       1       1           0  0  3
##       2      0      0       0       0           0  1  0
##       3      1      1       0       0           1  1  5
##       4      0      0       0       0           0  0  1
##       5      0      0       0       0           0  1  4
```

In reviewing the distribution of observations of the Location compared to each Cluster, again, we'll disregard Clusters 2 and 4 since these have only 1 observation.

```
Cluster 1, 20% are Germany, 20% are Ireland, and 60% are US.
Cluster 3, 11% are Canada, 11% are France, 11% are Switzerland, 11% are UK,
and 55% are US
Cluster 5, 20% are UK and 80% are US
```

From the raw data, we know that 62% of the firms in the observations are from the US.

While a larger percentage of firms in each Cluster are from the US, the percentage of US firms in each cluster is not much different than the overall average number of firms in the US. Therefore, there does not seem to be a strong correlation between Cluster and Location.

Now, we'll compare the Cluster_No to the Exchange (variable 12)

```
table(Cluster=Pharmaceuticals$Cluster_No, Exchange=Pharmaceuticals$Exchange)
```

```
##        Exchange
## Cluster AMEX NASDAQ NYSE
##       1    1      1    3
##       2    0      0    1
##       3    0      0    9
##       4    0      0    1
##       5    0      0    5
```

In reviewing the distribution of observations of the Exchange compared to each Cluster, again, we'll disregard Clusters 2 and 4 since these have only 1 observation.

```
Cluster 1, 20% are AMEX, 20% are NASDAQ, and 60% are NYSE
Cluster 3, 100% are NYSE
Cluster 5, 100% are NYSE
```

However, we know from the raw data that of the 21 firms listed, only 1 are on the AMEX and only 1 are on the NASDAQ. All remaining 19 firms are on the NYSE.

Since the one firm on the AMEX and the one firm on the NASDAQ are both listed in the same cluster (Cluster 1) together with 3 firms on the NYSE, it seems there is no correlation between Cluster and Exchange.

REQUIREMENT D:

Provide an appropriate name for each cluster using any or all of the variables in the dataset.

For this, I'll use 2 key variables for simplification: Market_Cap and PE_Ratio

```
set.seed(64060)

d_df <- Pharmaceuticals[,c(3,5)]

summary(d_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
```

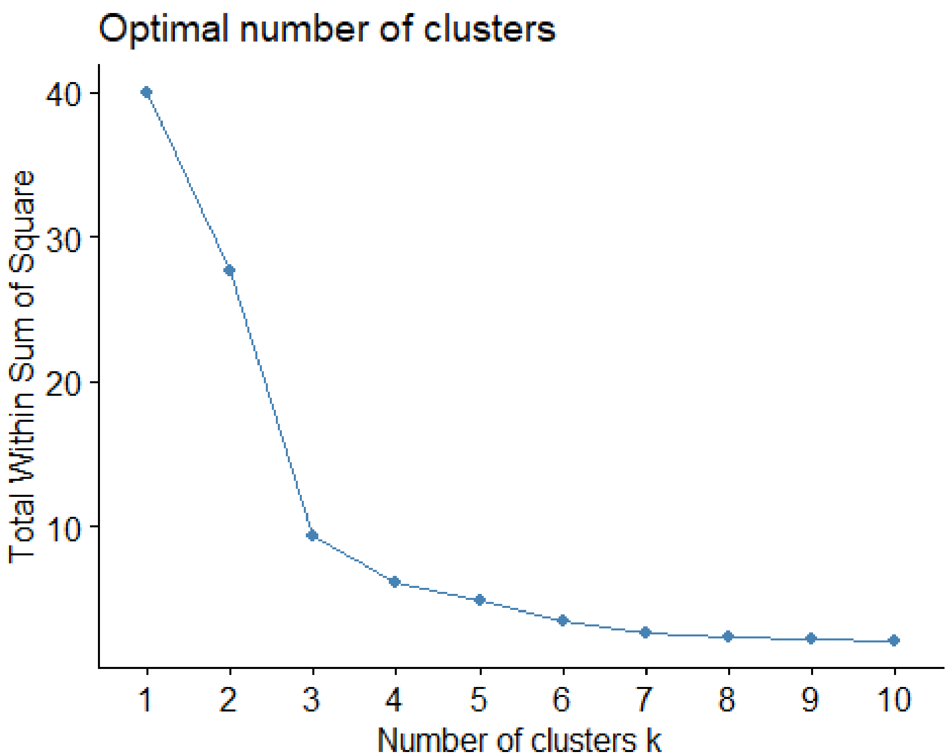
We'll scale the data frame

```
# Scaling the data frame (z-score)
d_df <- scale(d_df)

summary(d_df)

##    Market_Cap         PE_Ratio
##  Min.   :-0.9768   Min.   :-1.3404
```

```
##   1st Qu.:-0.8763   1st Qu.:-0.4023  
##   Median :-0.1614   Median :-0.2429  
##   Mean   : 0.0000   Mean   : 0.0000  
##   3rd Qu.: 0.2762   3rd Qu.: 0.1495  
##   Max.   : 2.4200   Max.   : 3.4971
```

CHOOSING THE BEST K Let's review an "elbow chart" to determine k

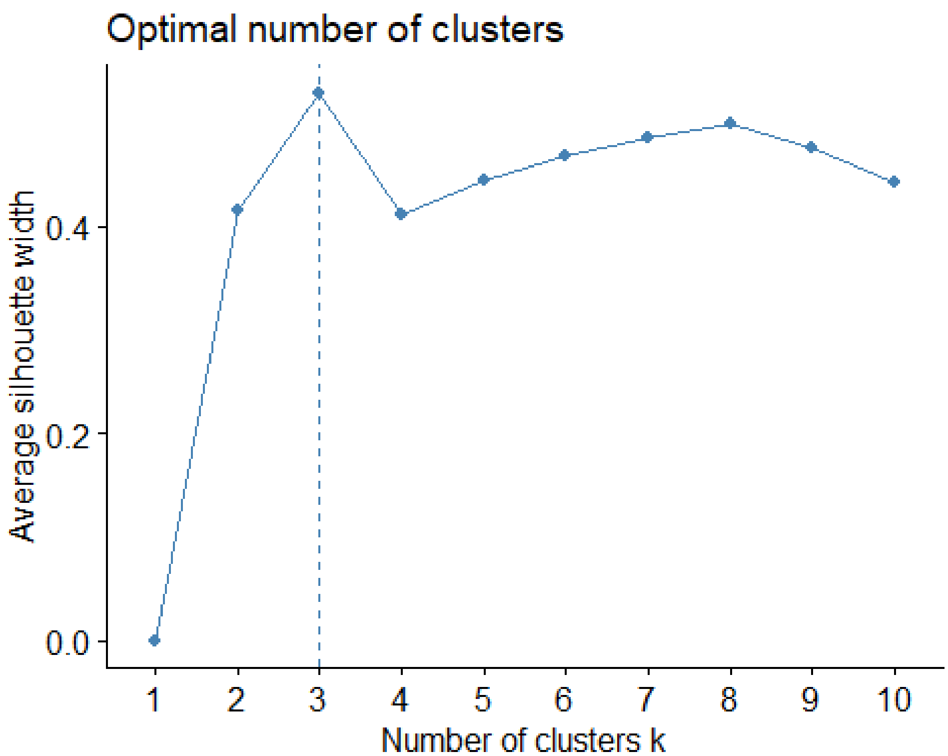
```
fviz_nbclust(d_df, kmeans, method = "wss")
```



This chart clearly shows the knee point at 3, indicating the optimal number of clusters is 3

Now, let's use the Silhouette Method to determine the number of clusters (k)

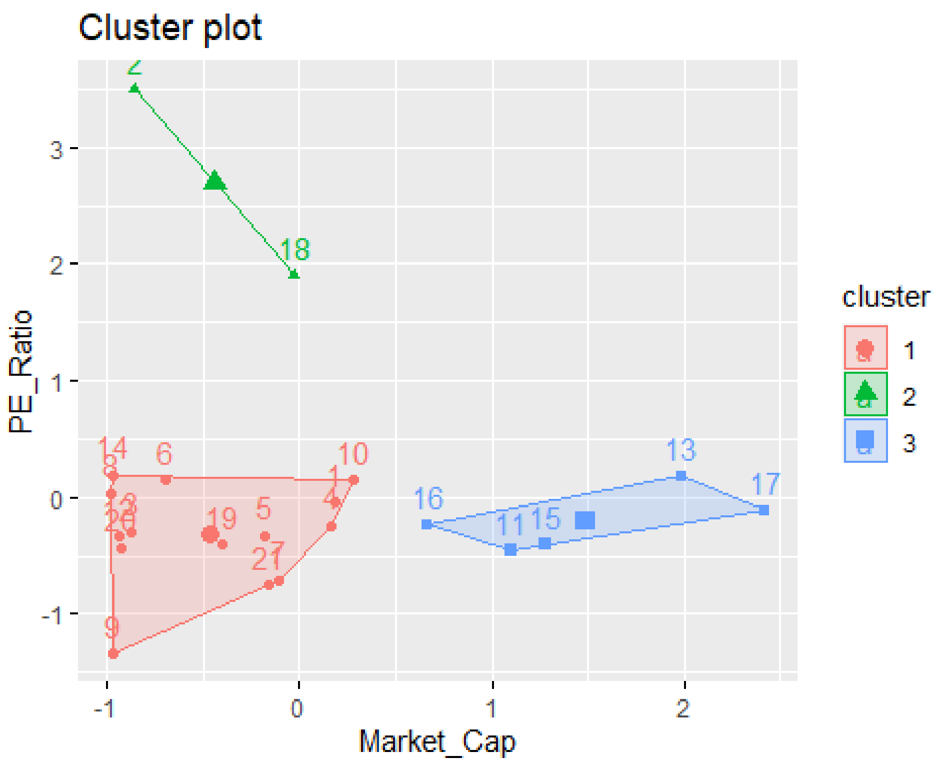
```
fviz_nbclust(d_df, kmeans, method = "silhouette")
```



This is also showing the optimal number of clusters is 3.

I'll run the k-means algorithm to cluster the firms and then visualize the output

```
set.seed(64060)

d_k3 <- kmeans(d_df, centers = 3, nstart = 25) # k = 3, number of restarts =
25

fviz_cluster(d_k3, data = d_df) # Visualize the output
```



Here, we see there are three (3) clusters:

Cluster 1: Has low PE_Ratio (indicating low growth) and small Market_Cap (greater growth potential) Cluster 2: Has high PE_Ratio (indicating high growth) and small Market_Cap (greater growth potential) Cluster 3: Has low PE_Ratio (indicating low growth) and large Market_Cap (lower growth potential)

In general, Market_Cap corresponds to the firm's stage in its business development. Large cap stocks are considered more conservative, less risky and less growth potential.

Also, high PE Ratios suggest investors are willing to pay more because they expecting higher earnings growth in the future. But it could also be an indication that the stock is overvalued. A low PE Ratio is better for investors as it could be an indication that the stock is currently undervalued.

Therefore, I would name each cluster as follows:

Cluster 1: Growth Potential Investments Cluster 2: Riskier Investments Cluster 3: Conservative Investments