Clustering Assignment - 4

Prasanth Yethirajula

Loading libraries and data set

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.3.2
```

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

```
pharmaceutical_data<-read.csv("C:/Users/drpra/Downloads/Pharmaceuticals.csv")
pharmaceutical_data<-na.omit(pharmaceutical_data)
```

Using the numerical variables (1 to 9) to cluster the 21 firms.

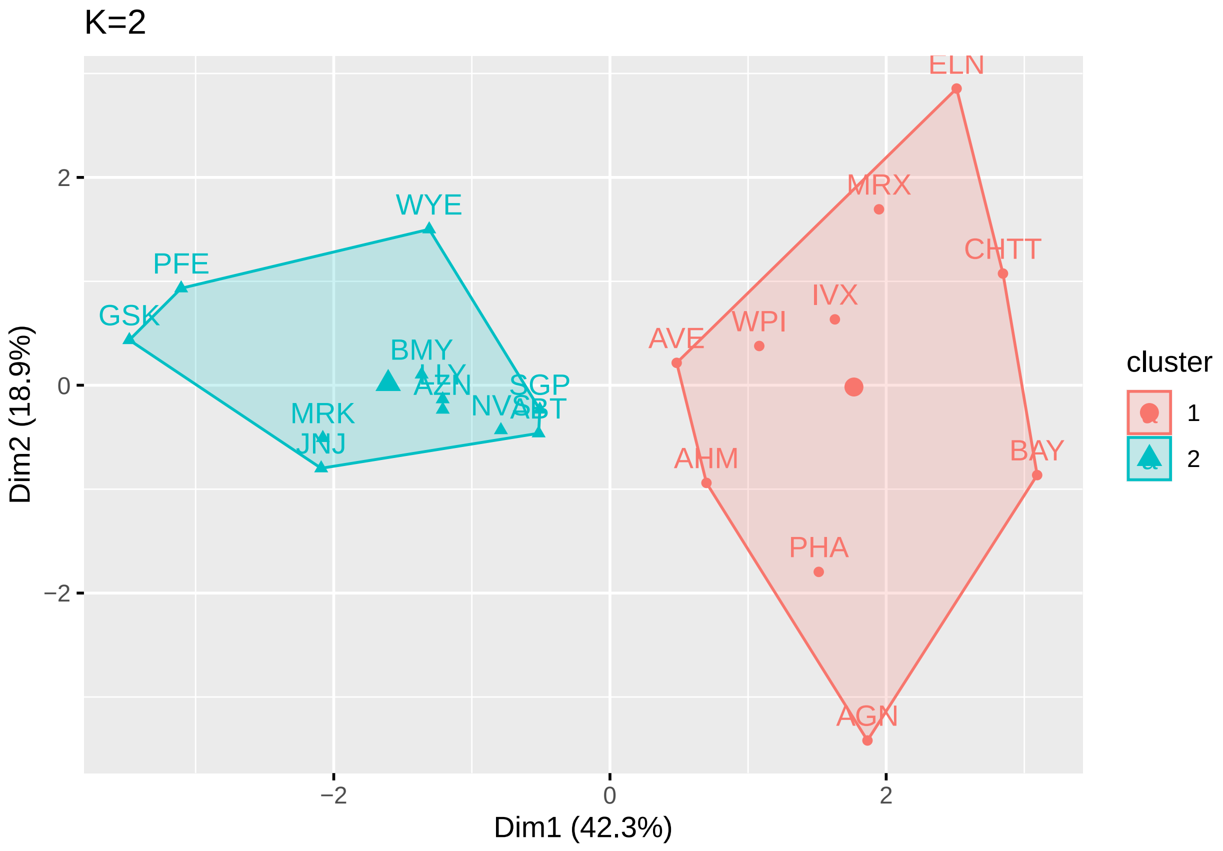
```
row.names(pharmaceutical_data)<-pharmaceutical_data[,1]
Clustering_dataset<-pharmaceutical_data[,3:11]
```

Scaling the data

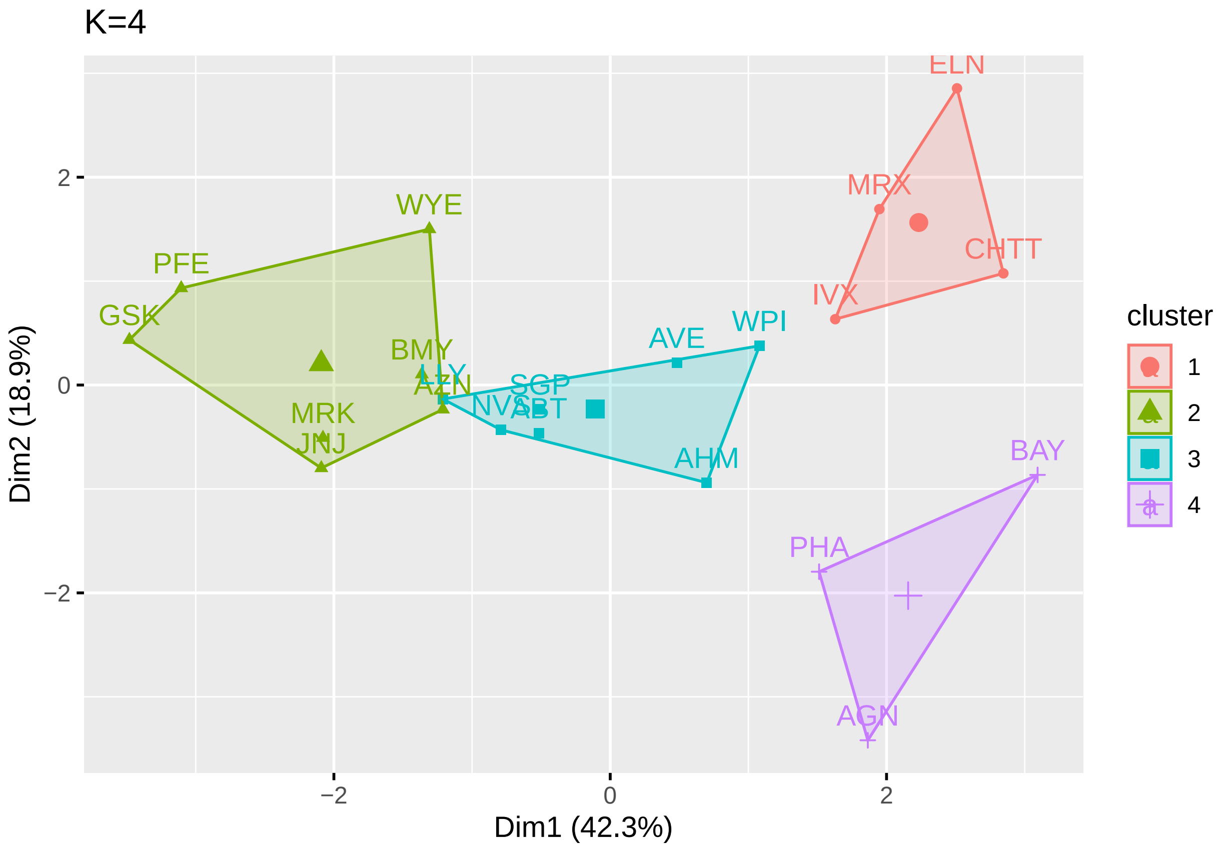
```
set.seed(143)
Scaled_data<-scale(Clustering_dataset)
```

Performing Kmeans for random K values

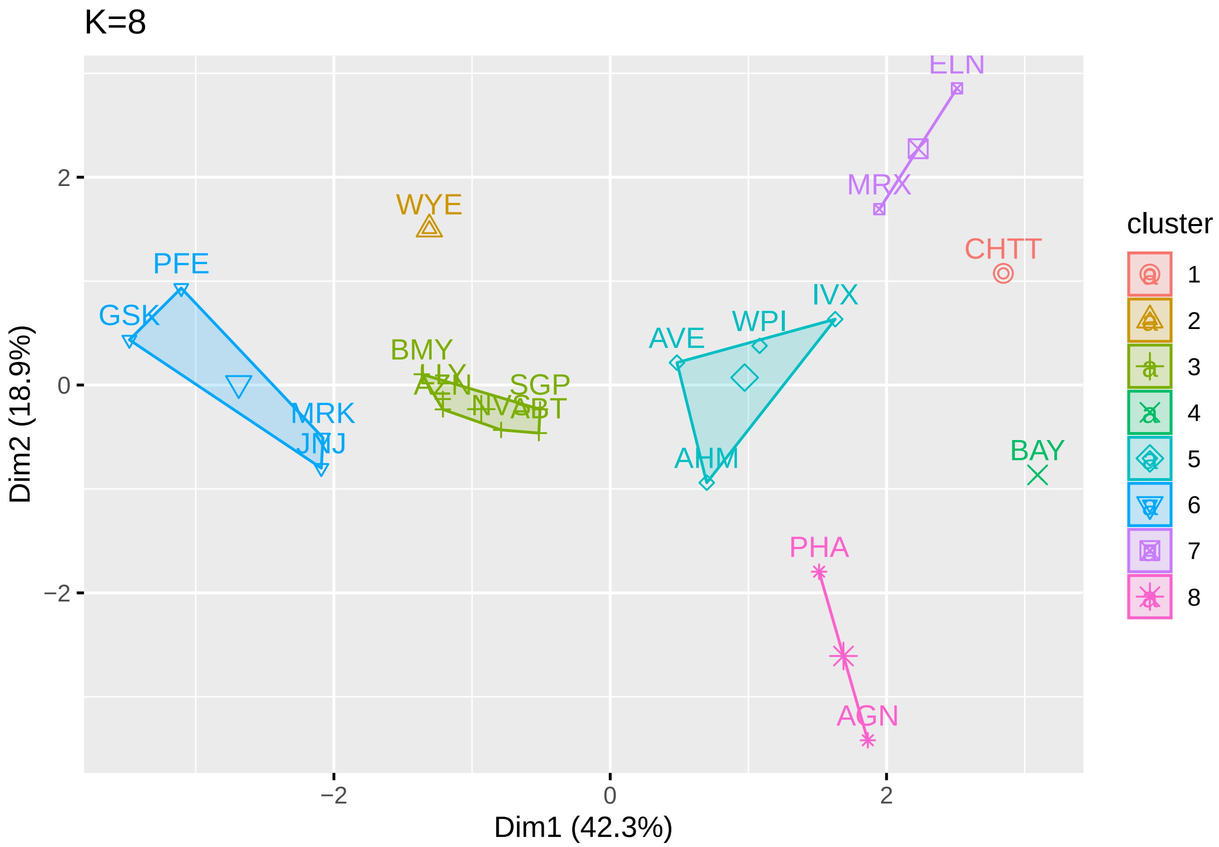
```
set.seed(143)
kmeans_2<-kmeans(Scaled_data,centers = 2, nstart = 15)
kmeans_4<-kmeans(Scaled_data,centers = 4, nstart = 15)
kmeans_8<-kmeans(Scaled_data,centers = 8, nstart = 15)

plot_kmeans_2<-fviz_cluster(kmeans_2,data = Scaled_data) + ggtitle("K=2")
plot_kmeans_4<-fviz_cluster(kmeans_4,data = Scaled_data) + ggtitle("K=4")
plot_kmeans_8<-fviz_cluster(kmeans_8,data = Scaled_data) + ggtitle("K=8")

plot_kmeans_2
```



```
plot_kmeans_4
```

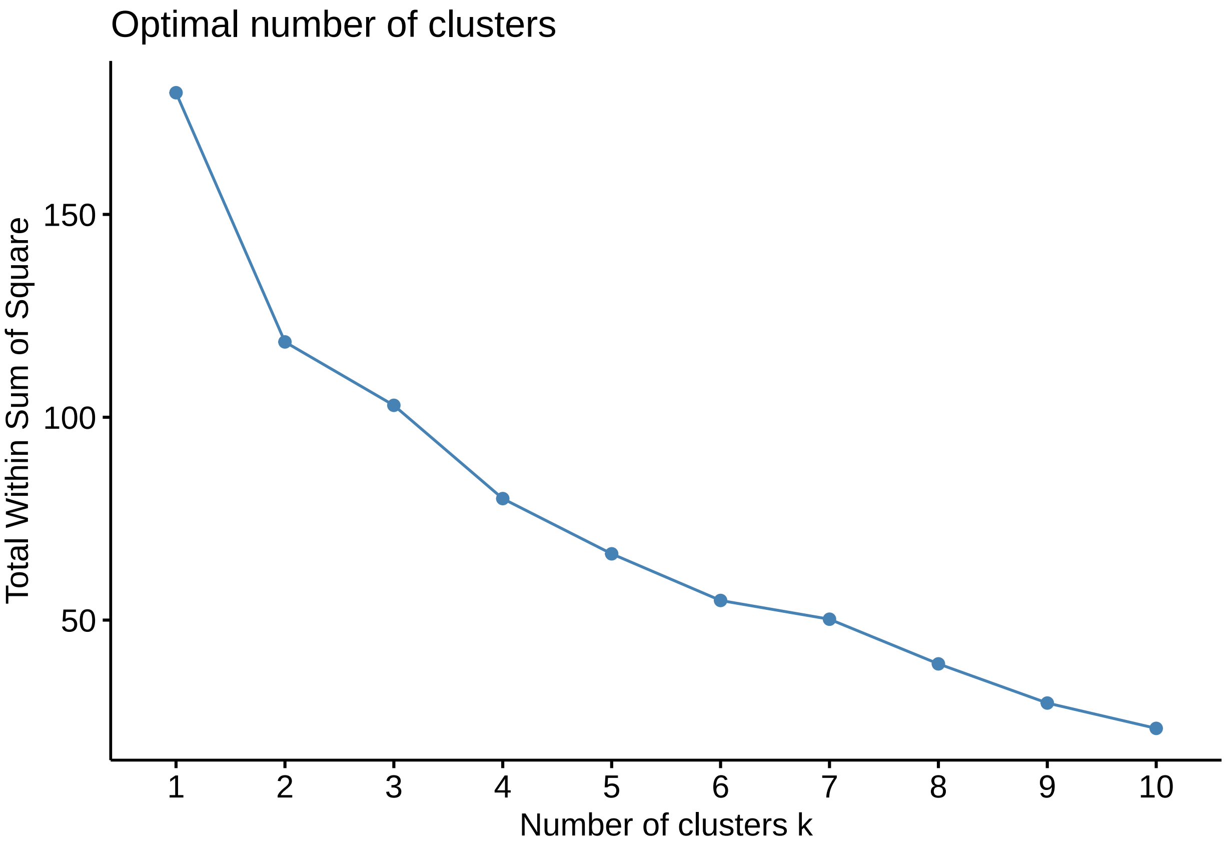


```
plot_kmeans_8
```

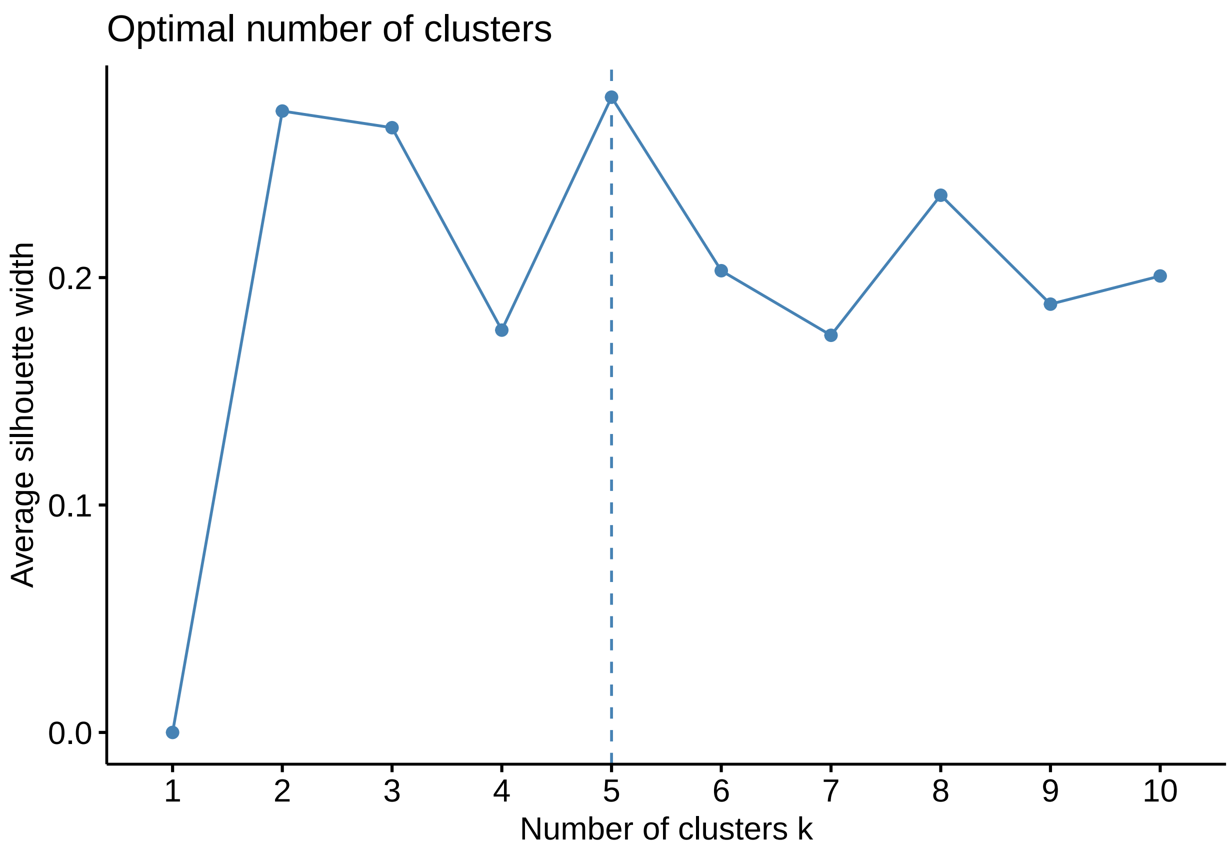


Using WSS and Silhouette to find best K suitable for clustering

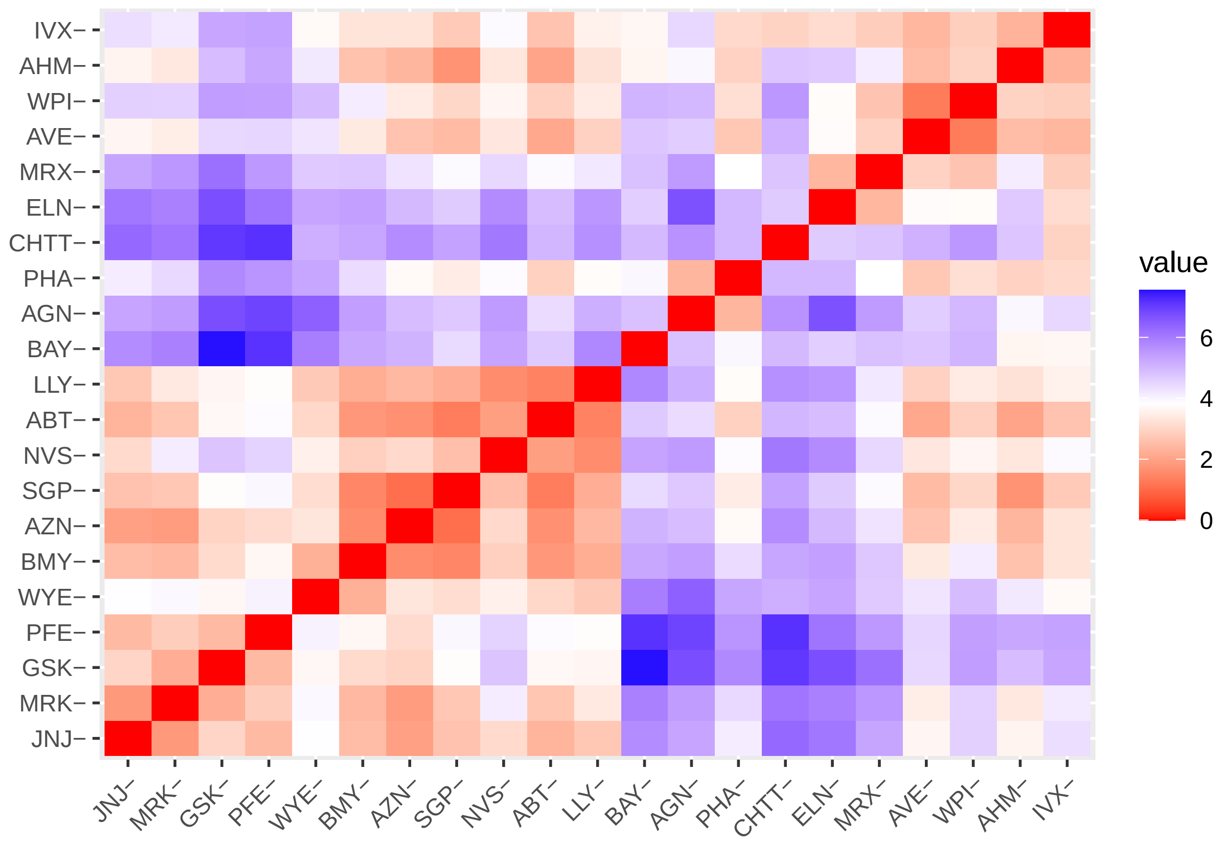
```
k_wss<-fviz_nbclust(Scaled_data,kmeans,method="wss")
k_silhouette<-fviz_nbclust(Scaled_data,kmeans,method="silhouette")
k_wss
```



```
k_silhouette
```



```
distance<-dist(Scaled_data,metho='euclidean')
fviz_dist(distance)
```



The within-sum-of-squares method suggests 2 clusters, while the silhouette method points to 5. We're opting for 5 clusters because this number not only keeps the within-cluster variance low but also maintains clear distinction between clusters.

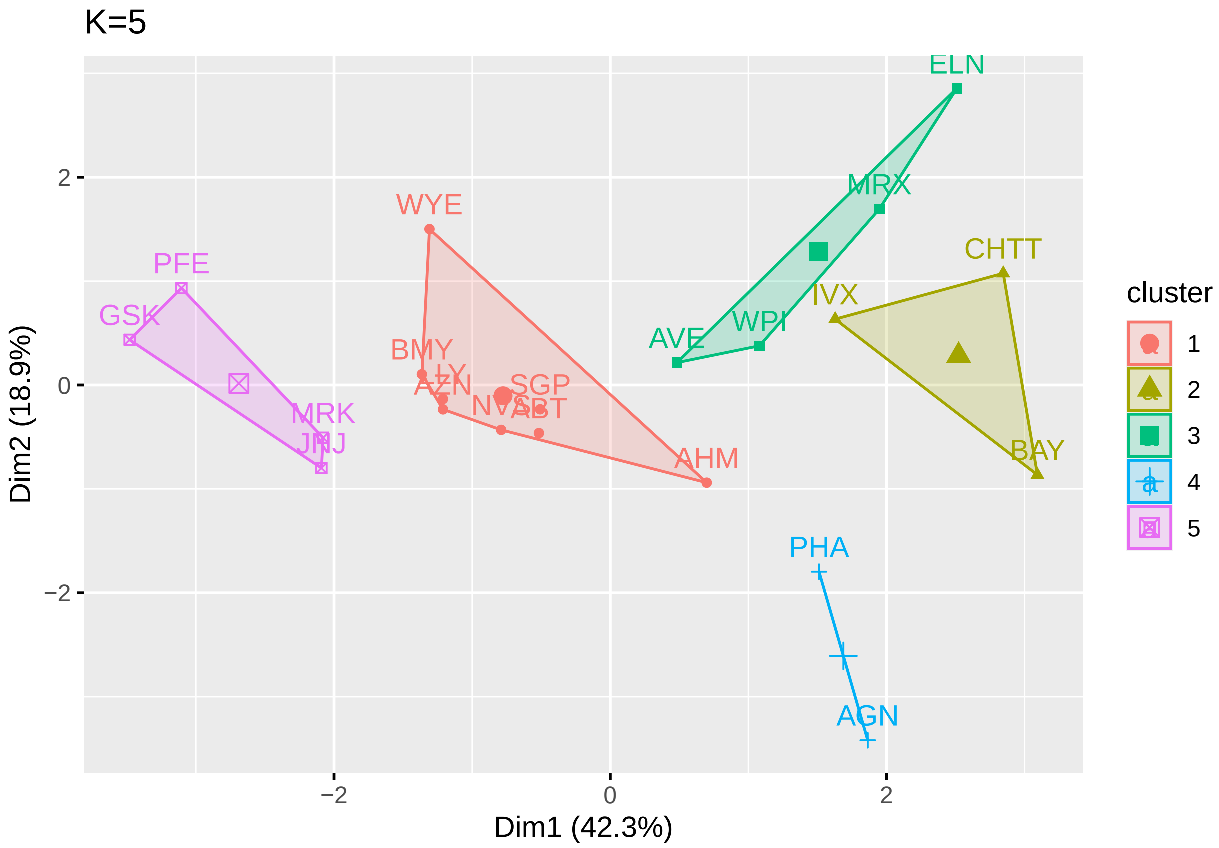
Performing Kmeans for suitable k

```
set.seed(143)
kmeans_5<-kmeans(Scaled_data,centers = 5, nstart = 10)
kmeans_5
```

```
## K-means clustering with 5 clusters of sizes 8, 3, 4, 2, 4
##
## Cluster means:
##    Market_Cap       Beta    PE_Ratio        ROE        ROA Asset_Turnover
## 1 -0.03142211 -0.4360989 -0.31724852  0.1950459  0.4083915      0.1729746
## 2 -0.87051511  1.3409869 -0.05284434 -0.6184015 -1.1928478     -0.4612656
## 3 -0.76022489  0.2796041 -0.47742380 -0.7438022 -0.8107428     -1.2684804
## 4 -0.43925134 -0.4701800  2.70002464 -0.8349525 -0.9234951      0.2306328
## 5  1.69558112 -0.1780563 -0.19845823  1.2349879  1.3503431      1.1531640
##      Leverage Rev_Growth Net_Profit_Margin
## 1 -0.27449312 -0.7041516       0.556954446
## 2  1.36644699 -0.6912914      -1.320000179
## 3  0.06308085  1.5180158      -0.006893899
## 4 -0.14170336 -0.1168459      -1.416514761
## 5 -0.46807818  0.4671788       0.591242521
##
```

```
## Clustering vector:
##   ABT   AGN   AHM   AZN   AVE   BAY   BMY  CHTT   ELN   LLY   GSK   IVX   JNJ   MRX   MRK   NVS
##     1     4     1     1     3     2     1     2     3     1     5     2     5     3     5     1
##   PFE   PHA   SGP   WPI   WYE
##     5     4     1     3     1
##
## Within cluster sum of squares by cluster:
## [1] 21.879320 15.595925 12.791257  2.803505  9.284424
##  (between_SS / total_SS =  65.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
plot_kmeans_5<-fviz_cluster(kmeans_5,data = Scaled_data) + ggtitle("K=5")
plot_kmeans_5
```



```
Clustering_dataset_1<-Clustering_dataset%>%
  mutate(Cluster_no=kmeans_5$cluster)%>%
  group_by(Cluster_no)%>%summarise_all('mean')
Clustering_dataset_1
```

```
## # A tibble: 5 x 10
##   Cluster_no Market_Cap  Beta PE_Ratio   ROE   ROA Asset_Turnover Leverage
##        <int>      <dbl> <dbl>    <dbl> <dbl> <dbl>          <dbl>    <dbl>
## 1          1       55.8  0.414     20.3  28.7 12.7           0.738    0.371
## 2          2        6.64 0.87      24.6  16.5  4.17          0.6      1.65
## 3          3       13.1  0.598     17.7  14.6  6.2           0.425    0.635
## 4          4       31.9  0.405     69.5  13.2  5.6           0.75     0.475
## 5          5      157.   0.48      22.2  44.4 17.7           0.95     0.22
## # i 2 more variables: Rev_Growth <dbl>, Net_Profit_Margin <dbl>
```

Companies are grouped into following clusters:

Cluster_1= ABT,AHM,AZN,BMY,LLY,NVS,SGP,WYE

Cluster_2= BAY,CHTT,IVX

Cluster_3=AVE,ELN,MRX,WPI

Cluster_4=AGN,PHA

Cluster_5=GSK,JNJ,MRK,PFE

From the clusters formed it can be understood that

1. Cluster 1 consists of companies that exhibit a moderate level of return on equity and return on investment.

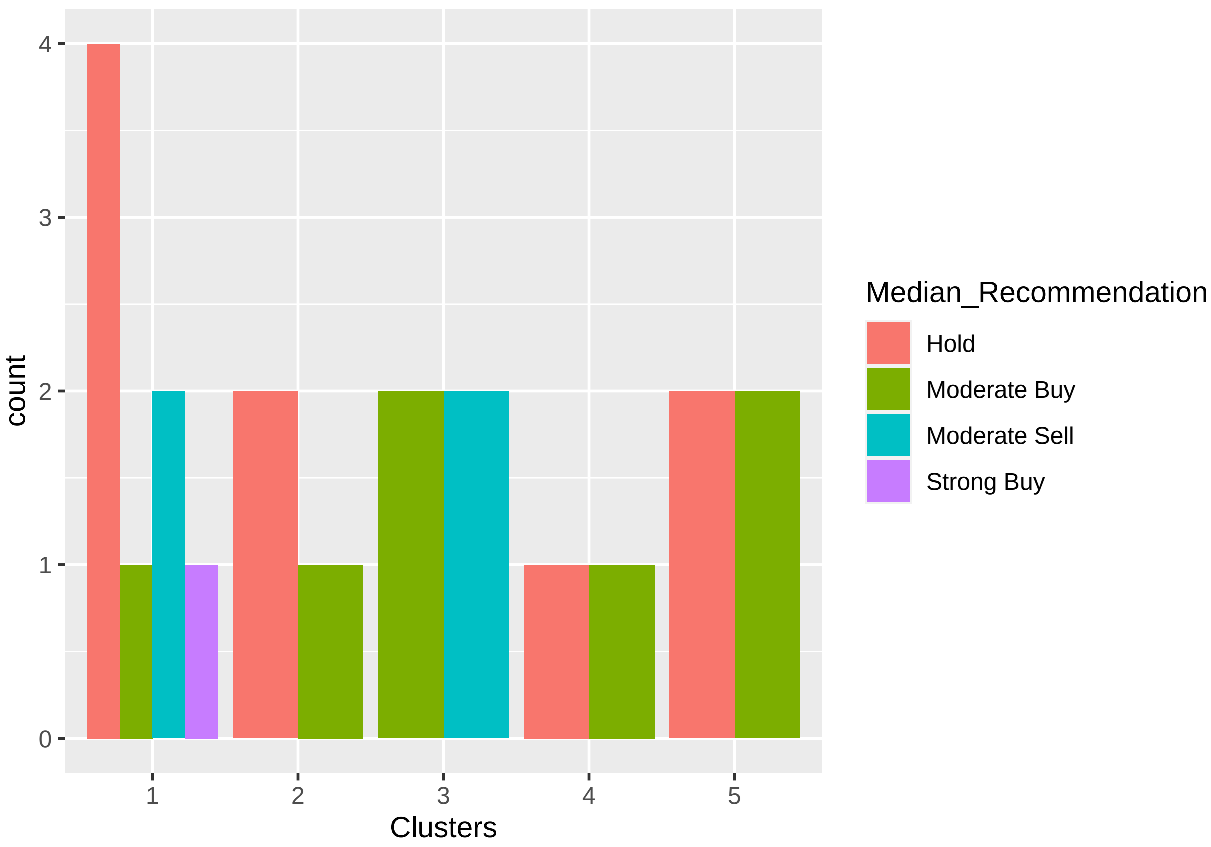
2. Cluster 2 is made up of companies characterized by poor return on assets, return on equity, market capitalization, and asset turnover, indicating these firms carry a higher level of risk.

3. Cluster 3 includes companies that are similar to those in Cluster 2, yet they present slightly lower levels of risk.

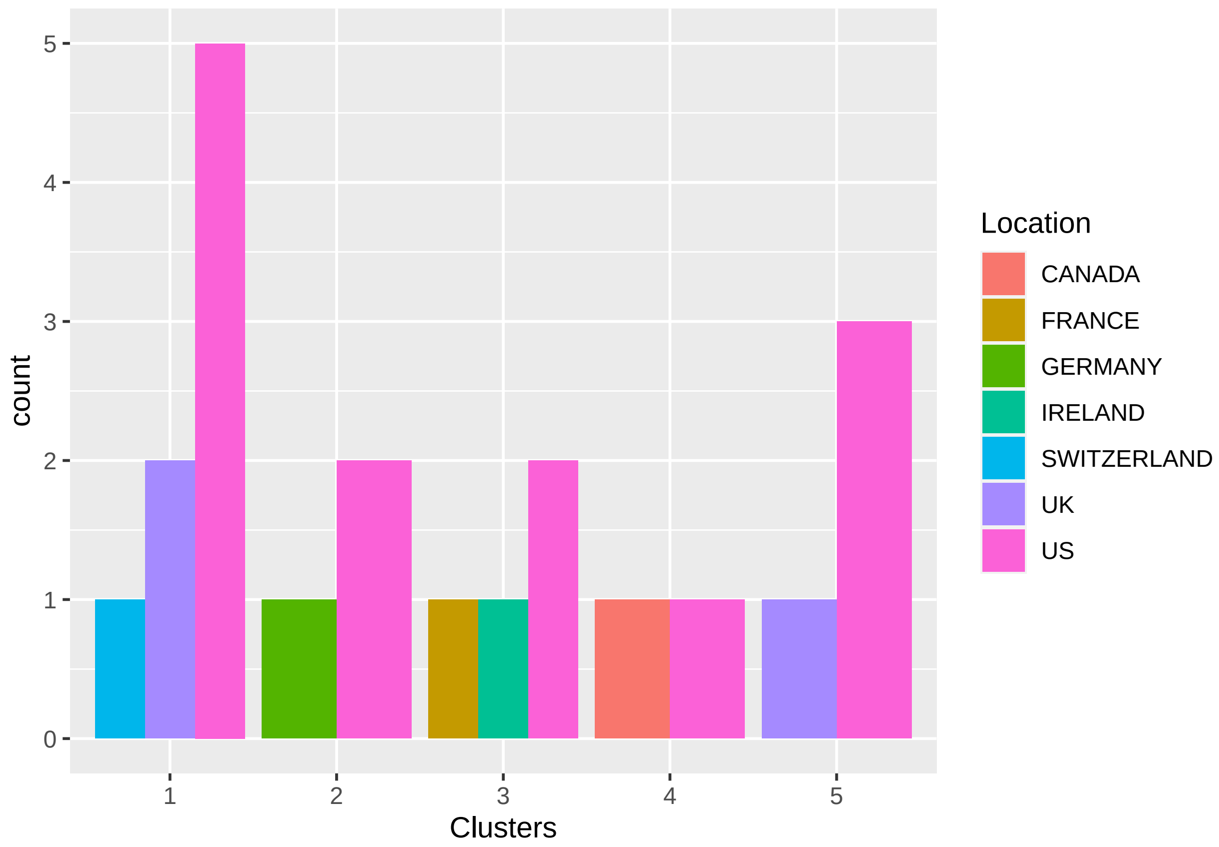
4. Cluster 4 is composed of companies that boast strong price-to-earnings ratios, yet suffer from weak return on assets and equity, making them riskier than the companies in Cluster 2.

5. Cluster 5 encompasses companies with robust market capitalization and high returns on equity and assets.

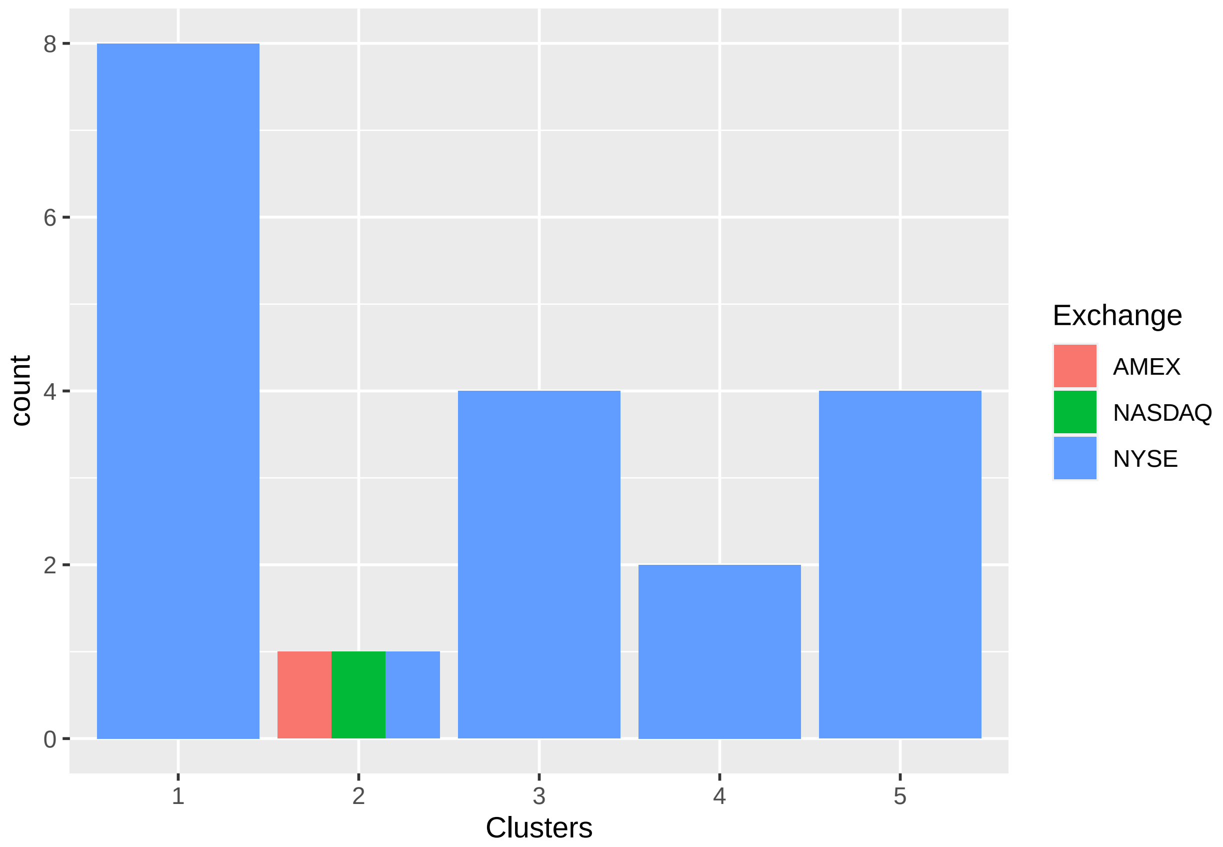
```
Clustering_datase_2<- pharmaceutical_data[,12:14] %>% mutate(Clusters=kmeans_5$cluster)
ggplot(Clustering_datase_2, mapping = aes(factor(Clusters), fill =Median_Recommendation))+geom_bar(posi
```



```
ggplot(Clustering_datase_2, mapping = aes(factor(Clusters),fill = Location))+geom_bar(position = 'dodge
```



```
ggplot(Clustering_datase_2, mapping = aes(factor(Clusters),fill = Exchange))+geom_bar(position = 'dodge
```



The clusters appear to show a trend with the variable 'Median Recommendation.' For instance, the second cluster leans towards recommendations ranging from 'hold' to 'moderate buy,' while the third cluster's recommendations vary from 'moderate buy' to 'moderate sell.' When looking at the companies' locations, it's observed that a large number of them are located in the US, which doesn't suggest any significant clustering pattern based on location. Similarly, in terms of the stock exchange listings, there isn't a distinct pattern correlating with the clusters, although a predominant number of the companies are listed on the NYSE.

Naming clusters:

[It is done based net Market capitalization(size) and Return on Assets(money)]

Cluster 1: Large-Thousands

Cluster 2: Extra Small-Penny

Cluster 3: Small- Dollars

Cluster 4: Medium-Hundreds

Cluster 5: Extra Large-Millions

DBSCAN CLUSTERING

```r
# Load necessary libraries
library(fpc)
```

```
## Warning: package 'fpc' was built under R version 4.3.2
```

```r
library(dbscan)
```

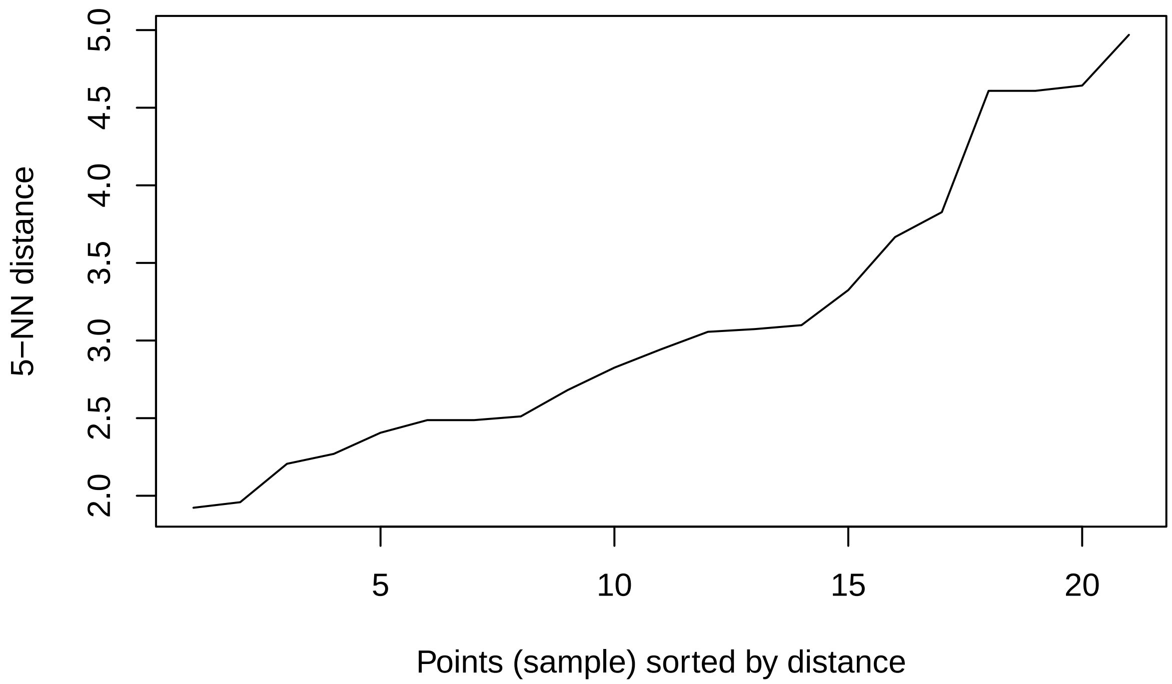
```
## Warning: package 'dbscan' was built under R version 4.3.2
```

```
##
## Attaching package: 'dbscan'
```

```
## The following object is masked from 'package:fpc':
##
##     dbscan
```

```
## The following object is masked from 'package:stats':
##
##     as.dendrogram
```

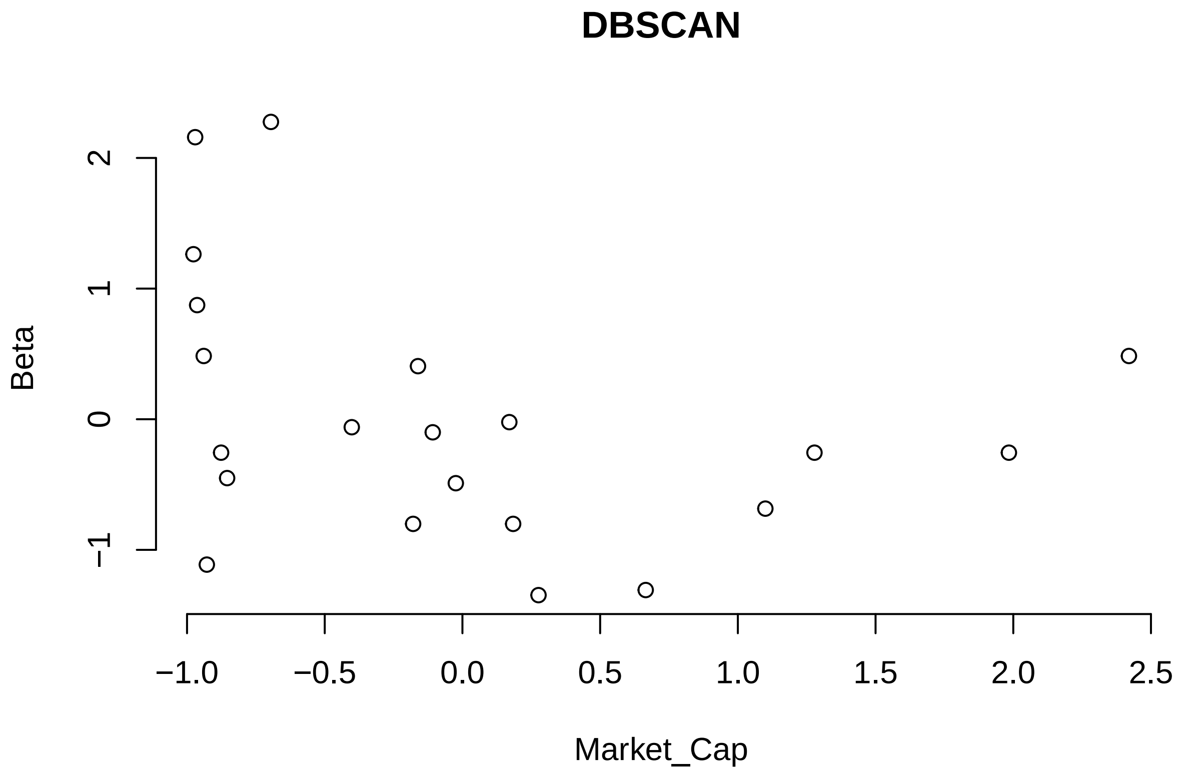
```r
# Use the kNNdistplot to help find a suitable eps value
kNNdistplot(Scaled_data, k = 5)
# Add an abline and try to visually identify the "elbow" point
abline(h = 0.05, col = 'red', lty = 2)  # Start with a small value for eps, adjust based on the plot
```



```
# Using the eps value identified from the kNNdistplot
# Setting minPts to a value that makes sense for your data; minPts = 5 is a common default
dbscan_result_1 <- dbscan(Scaled_data, eps = 0.5, minPts = 5)

# Print the cluster assignments
print(dbscan_result_1$cluster)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
plot(dbscan_result_1, Scaled_data, main= "DBSCAN", frame= FALSE)
```

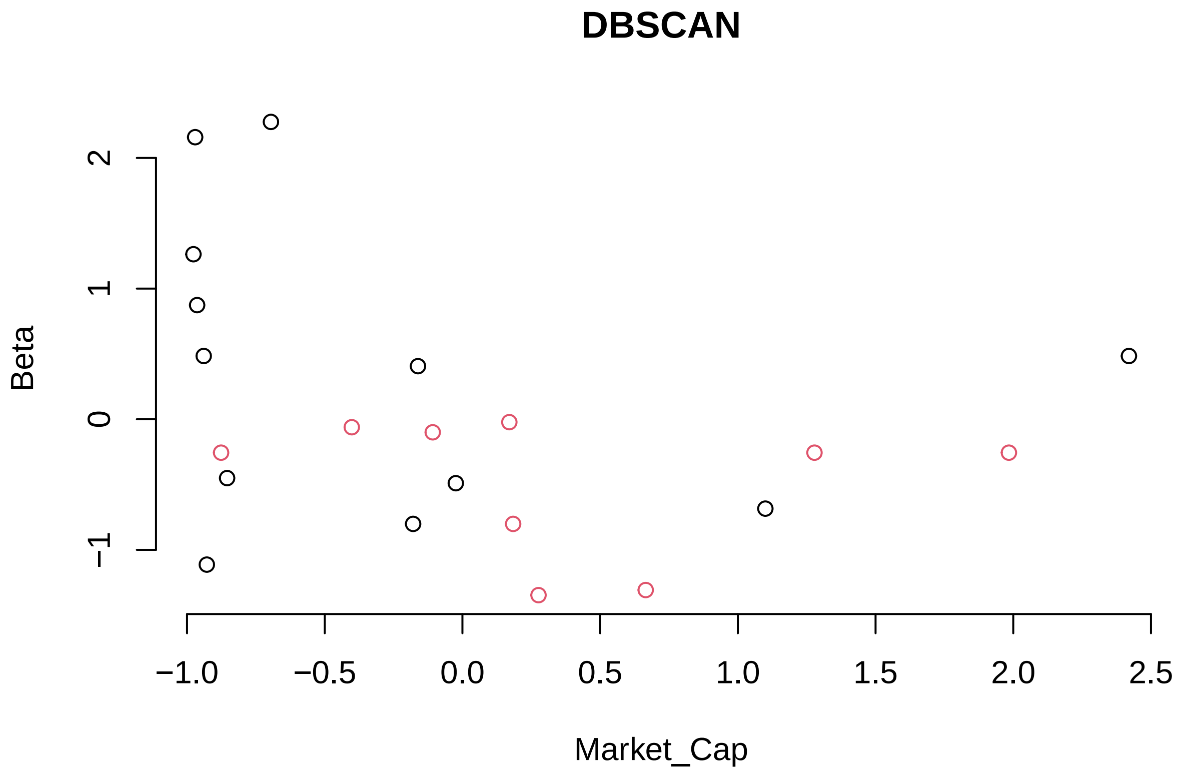


```
# Cluster 0: This is the main cluster identified by DBSCAN, which includes firms that are close togethe
# Cluster -1: This represents outlier points or noise, which are not sufficiently close to enough point
```

```
# USing different eps value for better clustering.
# If the eps value is too low then the output will be zero.
# If the eps value is too high then the output will be 1.
# Giving optimum eps value as 2.
dbscan_result_2 <- dbscan(Scaled_data, eps = 2.0, minPts = 5)

# Print the cluster assignments
print(dbscan_result_2$cluster)
```

```
##  [1] 1 0 1 1 0 0 1 0 0 1 0 0 1 0 1 1 0 0 1 0 0
```

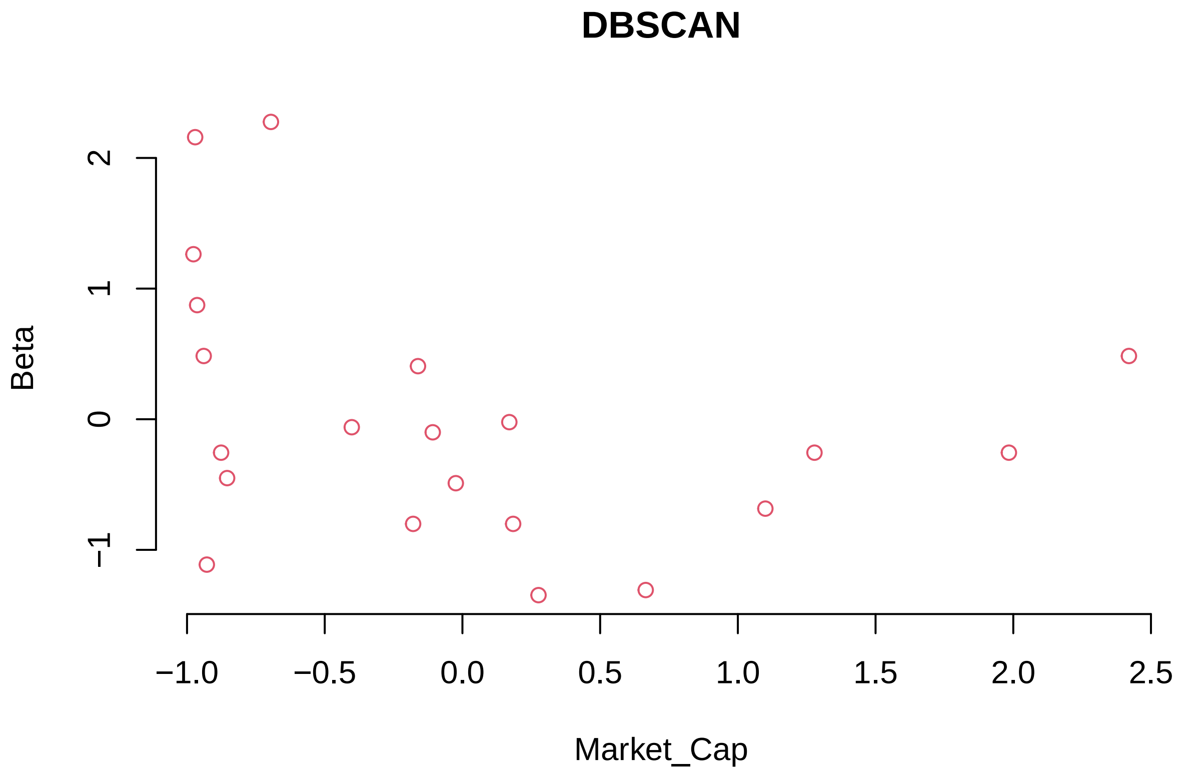
```
plot(dbscan_result_2, Scaled_data, main= "DBSCAN", frame= FALSE)
```



```
#By giving the eps value high the outcome will be 1.
dbscan_result_3 <- dbscan(Scaled_data, eps = 5.0, minPts = 5)

# Print the cluster assignments
print(dbscan_result_3$cluster)
```

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

```
plot(dbscan_result_3, Scaled_data, main= "DBSCAN", frame= FALSE)
```



HIERARCHICAL CLUSTERING

```
# Loading necessary library
library(stats)
# Hierarchical clustering using Ward's method
hc_result <- hclust(dist(Scaled_data), method = "ward.D2")

# Cut the dendrogram to create a specified number of clusters, e.g., 3
clusters <- cutree(hc_result, k = 3)

# Print the clusters
print(clusters)
```

```
##  ABT  AGN  AHM  AZN  AVE  BAY  BMY CHTT  ELN  LLY  GSK  IVX  JNJ  MRX  MRK  NVS
##    1    2    3    1    3    2    1    3    3    1    1    3    1    3    1    1
##  PFE  PHA  SGP  WPI  WYE
##    1    2    1    3    1
```

```
# Load necessary library
library(ggplot2)
library(dendextend)
```

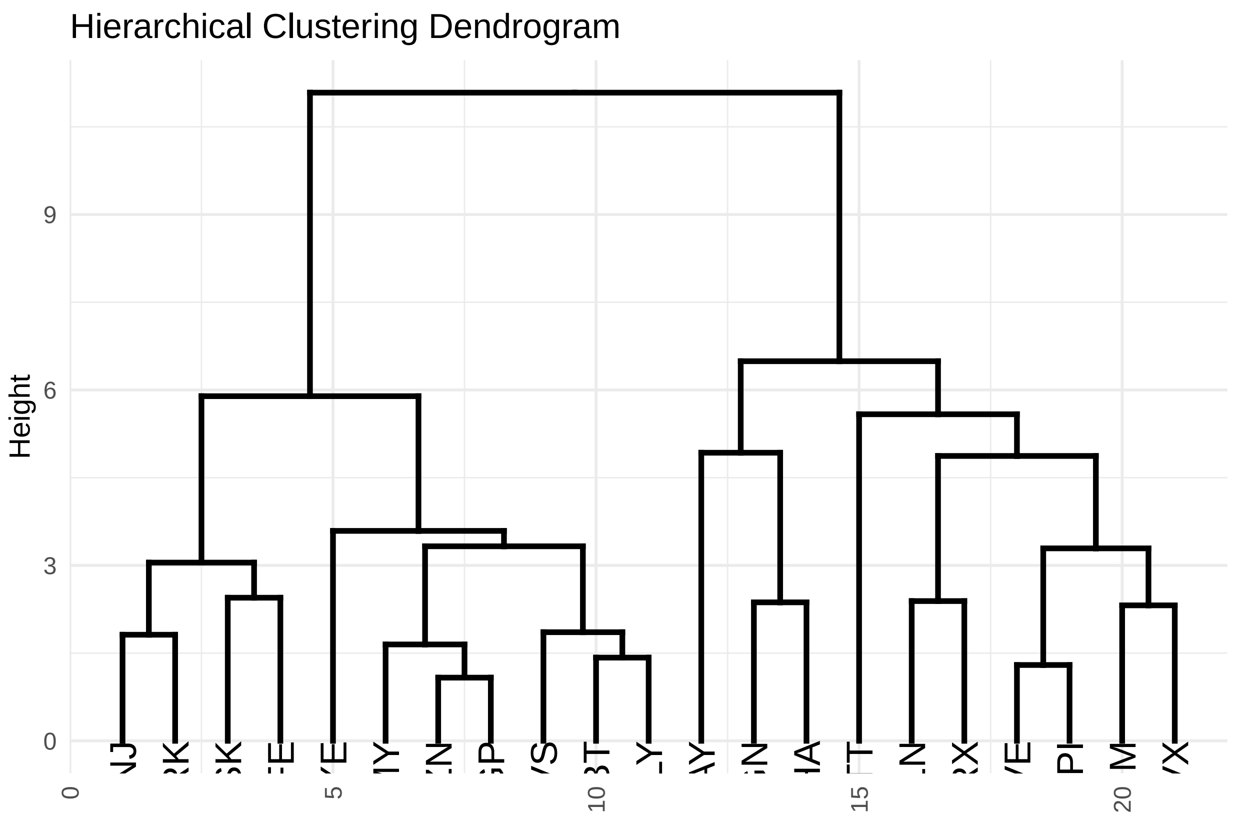
```
## Warning: package ’dendextend’ was built under R version 4.3.2
```

```
##
## ---------------------
## Welcome to dendextend version 1.17.1
## Type citation(’dendextend’) for how to cite the package.
##
## Type browseVignettes(package = ’dendextend’) for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: ’dendextend’
```

```
## The following object is masked from ’package:stats’:
##
##     cutree
```

```r
# Turn the hclust object into a dendrogram
dend <- as.dendrogram(hc_result)

# Create a ggplot object for the dendrogram
ggdend <- as.ggdend(dend)

# Plot the ggplot object
ggplot(ggdend, theme = theme_minimal()) +
  labs(title = "Hierarchical Clustering Dendrogram", x = "", y = "Height") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



***

DBSCAN Clustering: There are two clusters identified by DBSCAN (labeled 0 and 1), with several points labeled as -1, indicating they are considered noise by the algorithm. The silhouette score for DBSCAN is approximately 0.052, which is quite low. This suggests that the clusters defined by DBSCAN are not very dense or well separated.

Hierarchical Clustering: Given that DBSCAN did not yield a useful number of clusters, I opted for an arbitrary choice of 3 clusters for hierarchical clustering. The silhouette score for hierarchical clustering is approximately 0.273, which is better than the DBSCAN result but still indicates moderate cluster overlap or cluster structure.
For hierarchical clustering, I used 2 clusters as the DBSCAN resulted in essentially one cluster when not considering noise. It appears that hierarchical clustering with 2 clusters resulted in a more reasonable silhouette score.

There is no proper answer for this clustering techniques i applied K-Means, DBSCAN and Hierarchial clustering techniques to the dataset and i observed that for each clustering technique has its own importance.

K-Means is a good starting point for partitioning methods, especially if you have a good estimate for the number of clusters.
DBSCAN is ideal for data with noise and when clusters are not necessarily globular.
Hierarchical Clustering is great for exploratory data analysis and when a visual depiction of clusters is beneficial.

To conclude, while each algorithm has its own advantages, the choice of which to use should be guided by the nature of the dataset.

Selection of Clustering:

By observing all the clustering techniques i came to a conclusion that k=5 cluster has better graph and better understanding of clusters so i prefer k-means clustering is much better clustering technique for this dataset.

Let's interpret the values of cluster and k-means: The interpretation of the clusters, considering both the clustering and non-clustering variables, is as follows:

Cluster Characteristics Based on Clustering Variables:
Cluster 0 has a lower average market capitalization and higher average beta (indicating potentially higher volatility) than Cluster 1. The PE Ratio is also higher on average, while the ROE and ROA are lower than those for Cluster 1. This cluster also has a higher average leverage and revenue growth but a lower net profit margin.
Cluster 1 has a significantly higher average market capitalization and lower beta (less volatility). The PE Ratio is lower, suggesting a potentially better price-to-earnings value. It has higher ROE and ROA, indicating generally more profitable and efficient operations. This cluster has lower leverage, lower revenue growth, and a higher net profit margin compared to Cluster 0.
Patterns with respect to Non-Clustering Numerical Variables:
Revenue Growth (Rev_Growth): Cluster 0 has a higher mean revenue growth, but the most common (mode) value for both clusters is negative, which may indicate that the most common trend among companies in both clusters is a decline in revenue growth.
Net Profit Margin: Cluster 1 outperforms Cluster 0 with a significantly higher average net profit margin. The mode of the net profit margin is also higher for Cluster 1.
For the categorical variables, the mode was calculated, but due to the limitations in this environment, the mode for non-numeric data is not displayed here. Typically, you would analyze the most common Median Recommendation, Location, and Exchange for each cluster to discern any patterns or trends.

Based on these observations, clusters could be named to reflect their defining characteristics, such as:

Cluster 0: "High Growth, High Leverage" – characterized by higher revenue growth and leverage, indicating these companies might be in a growth phase but with higher risk.
Cluster 1: "Stable, Profitable Giants" – characterized by large market caps, stable operations with lower beta, and higher profitability.
These names are indicative and would benefit from domain expertise to better reflect the characteristics of the firms within each cluster. The patterns in the clusters concerning the non-clustering variables suggest potential areas for further investigation, such as why certain firms with high leverage and growth have declining revenue growth modes.