Assignment4

saimithra

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 errors
```

```
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
```

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

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

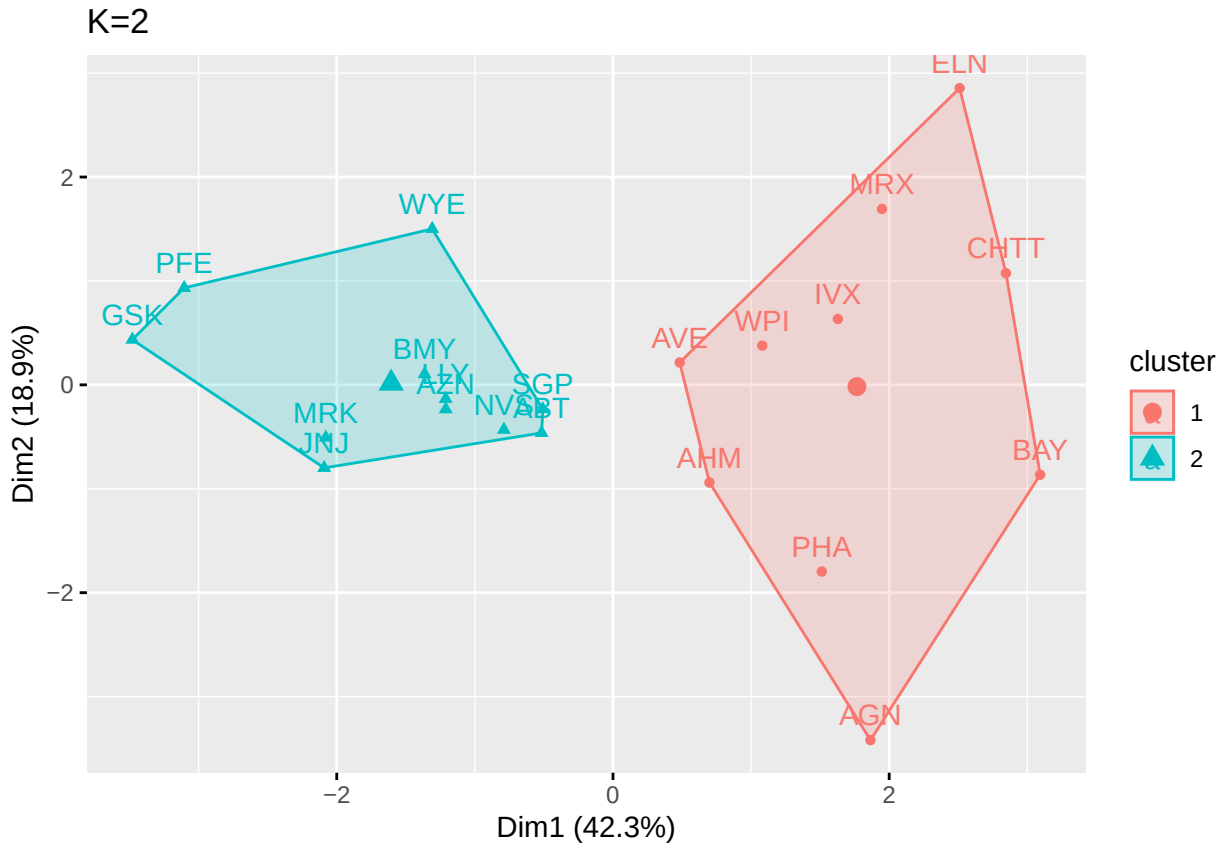
```
row.names(pharma_data)<-pharma_data[,1]
cluster_data<-pharma_data[,3:11]
```

Scaling the data

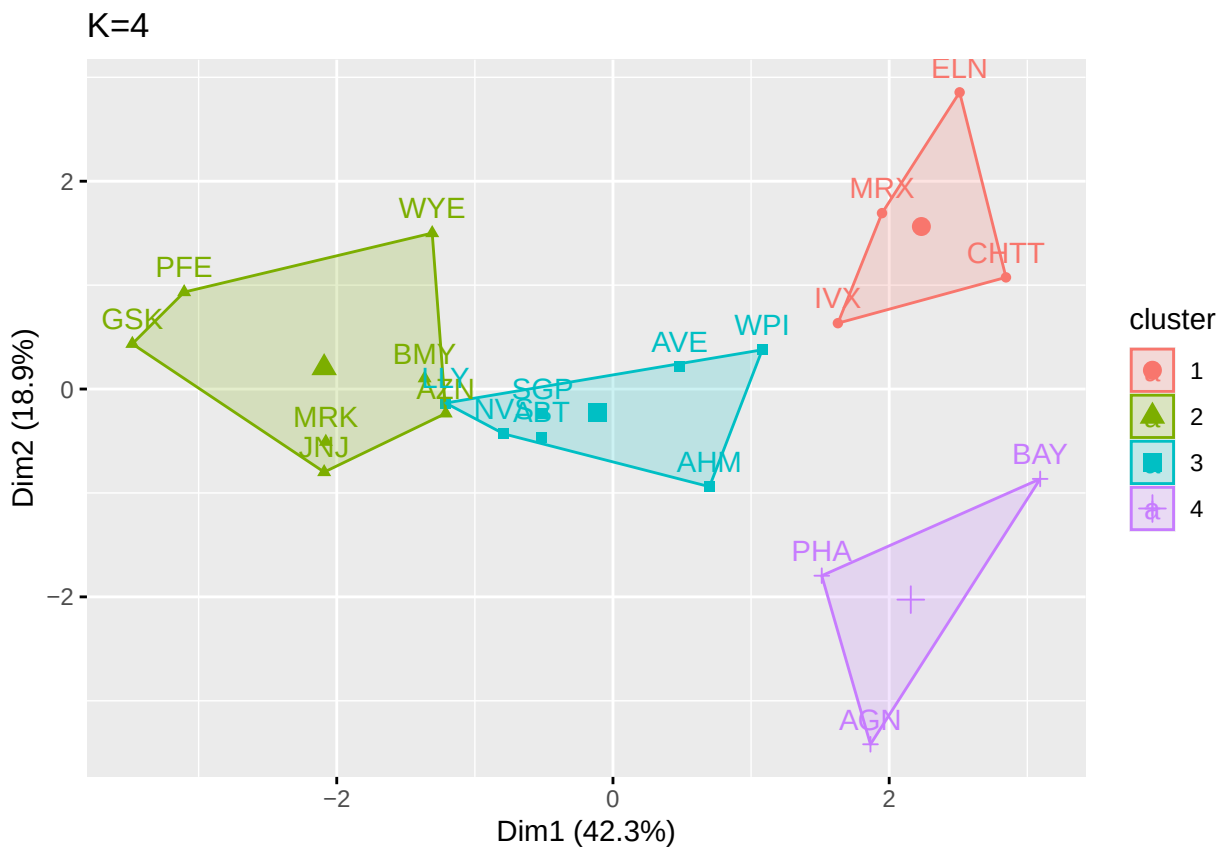
```
set.seed(143)
Scaled_data<-scale(cluster_data)
```

Performing Kmeans for random K values

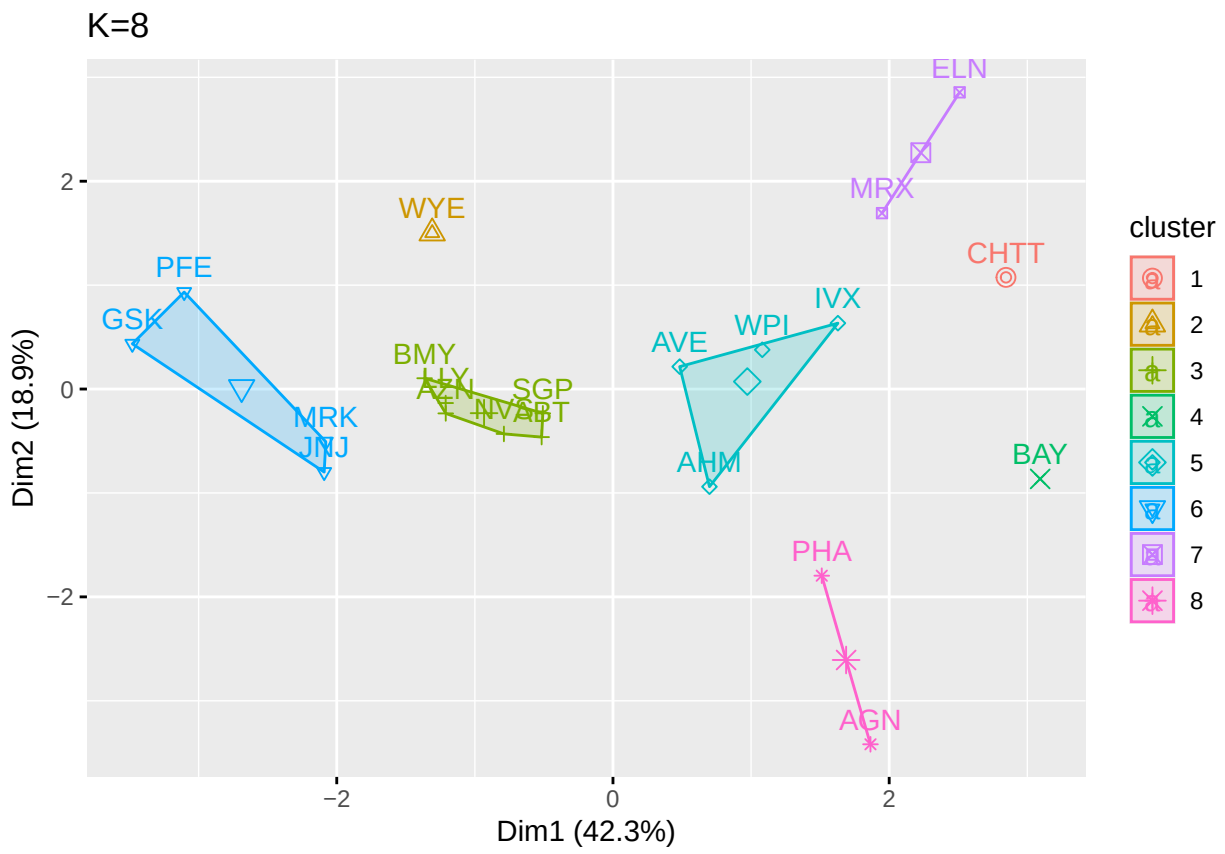
```
set.seed(143)
kmeans_2<-kmeans(Scaled_data,centers = 2, nstart = 15)
kmeans_4<-kmeans(Scaled_data,centers = 4, nstart = 15)
kmeans_8<-kmeans(Scaled_data,centers = 8, nstart = 15)
plot_kmeans_2<-fviz_cluster(kmeans_2,data = Scaled_data) + ggtitle("K=2")
plot_kmeans_4<-fviz_cluster(kmeans_4,data = Scaled_data) + ggtitle("K=4")
plot_kmeans_8<-fviz_cluster(kmeans_8,data = Scaled_data) + ggtitle("K=8")

plot_kmeans_2
```




```
plot_kmeans_4
```

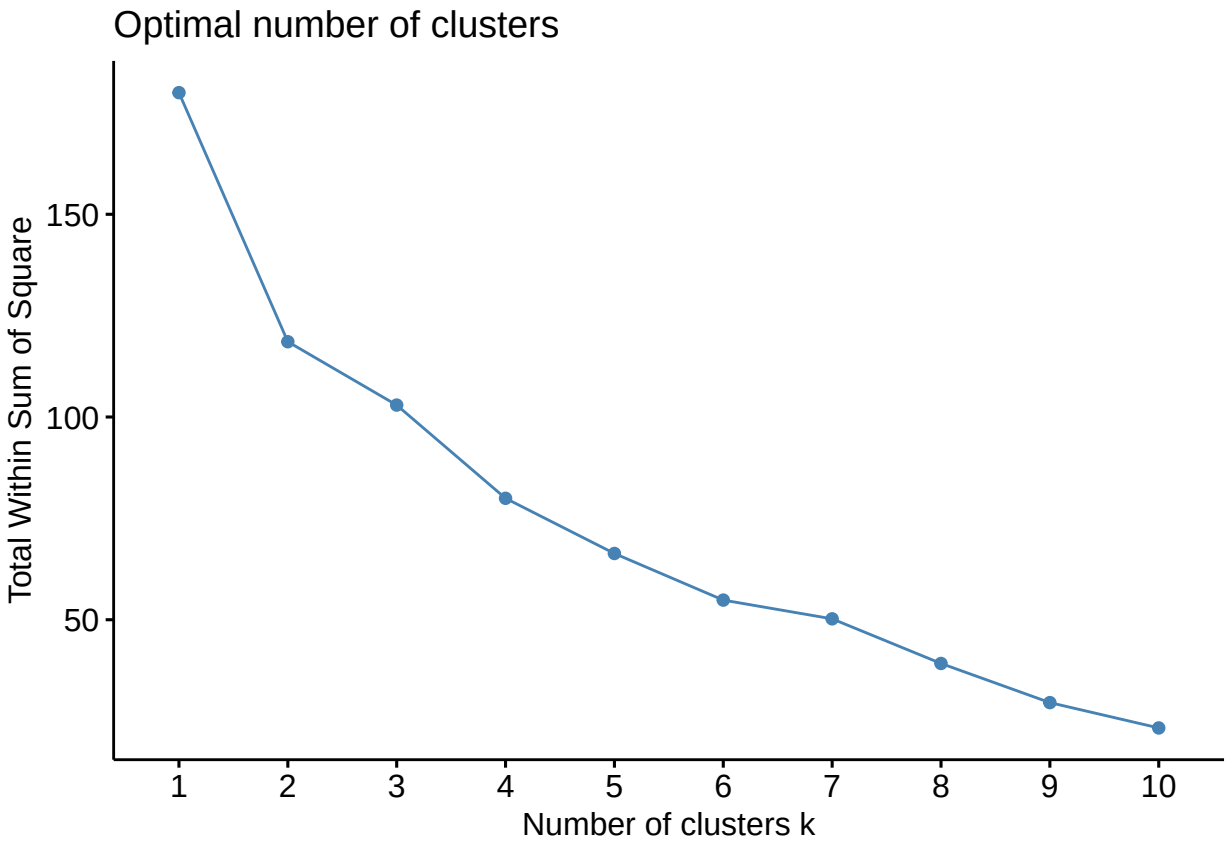



```
plot_kmeans_8
```

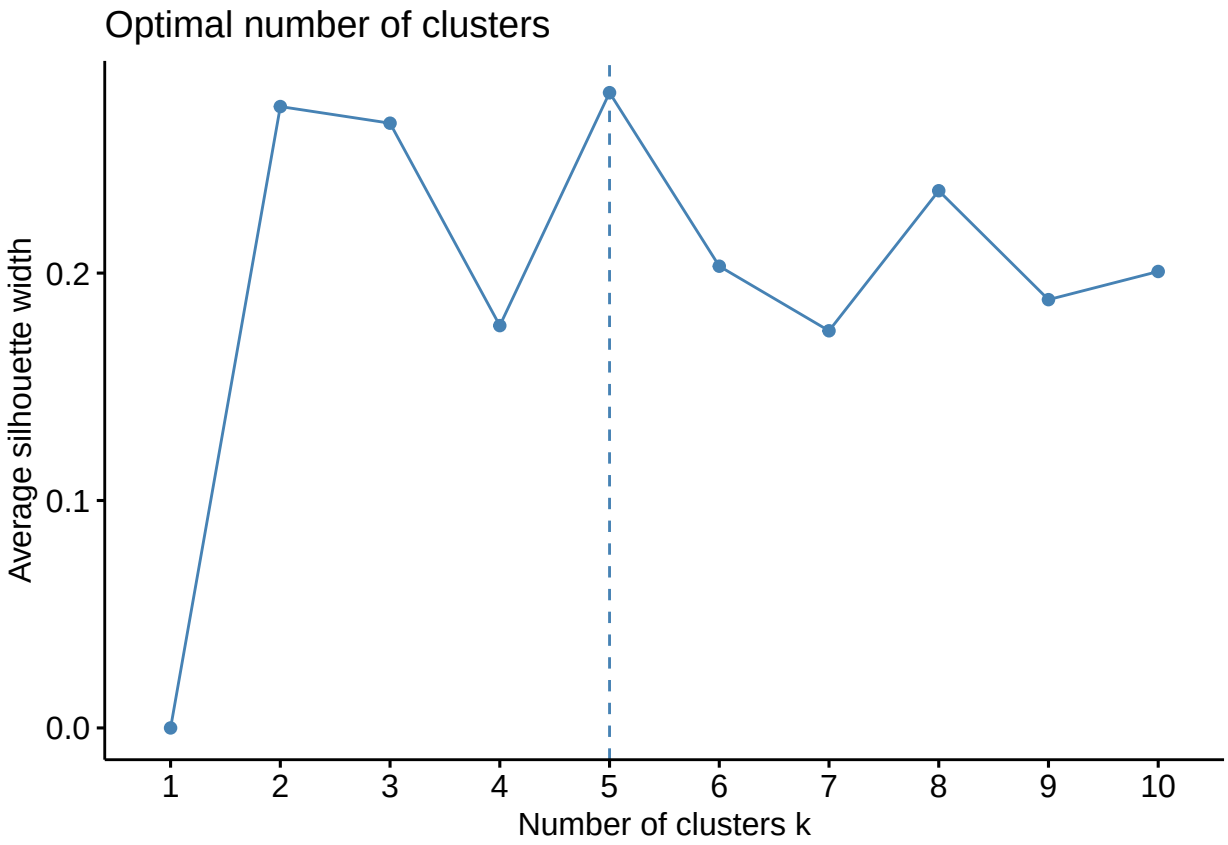


Using WSS and Silhouette to find best K suitable for clustering

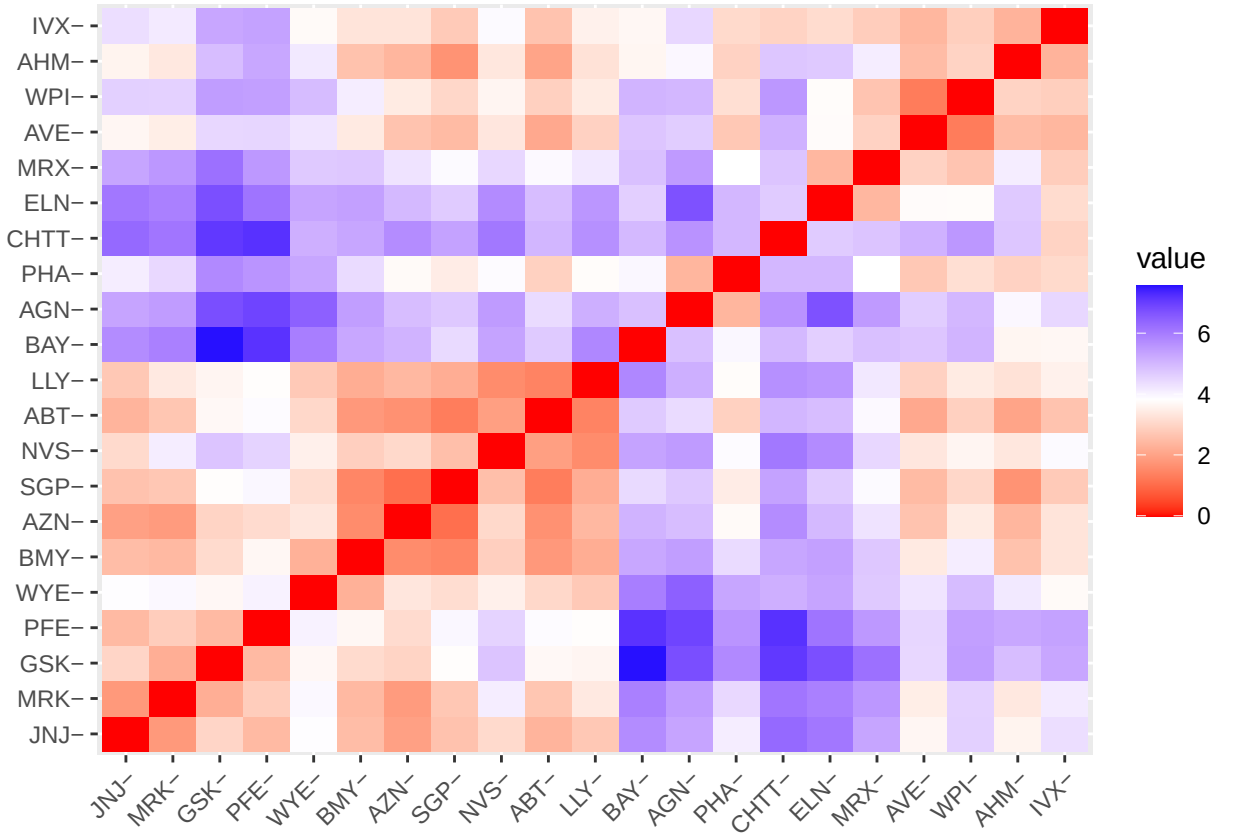
```
k_wss<-fviz_nbclust(Scaled_data,kmeans,method="wss")
k_silhouette<-fviz_nbclust(Scaled_data,kmeans,method="silhouette")
k_wss
```




```
k_silhouette
```




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



The silhouette method indicates five clusters, whereas the within-sum-of-squares method recommends two. Five clusters is the number we've chosen because it retains both a low within-cluster variation and an obvious separation between the clusters.

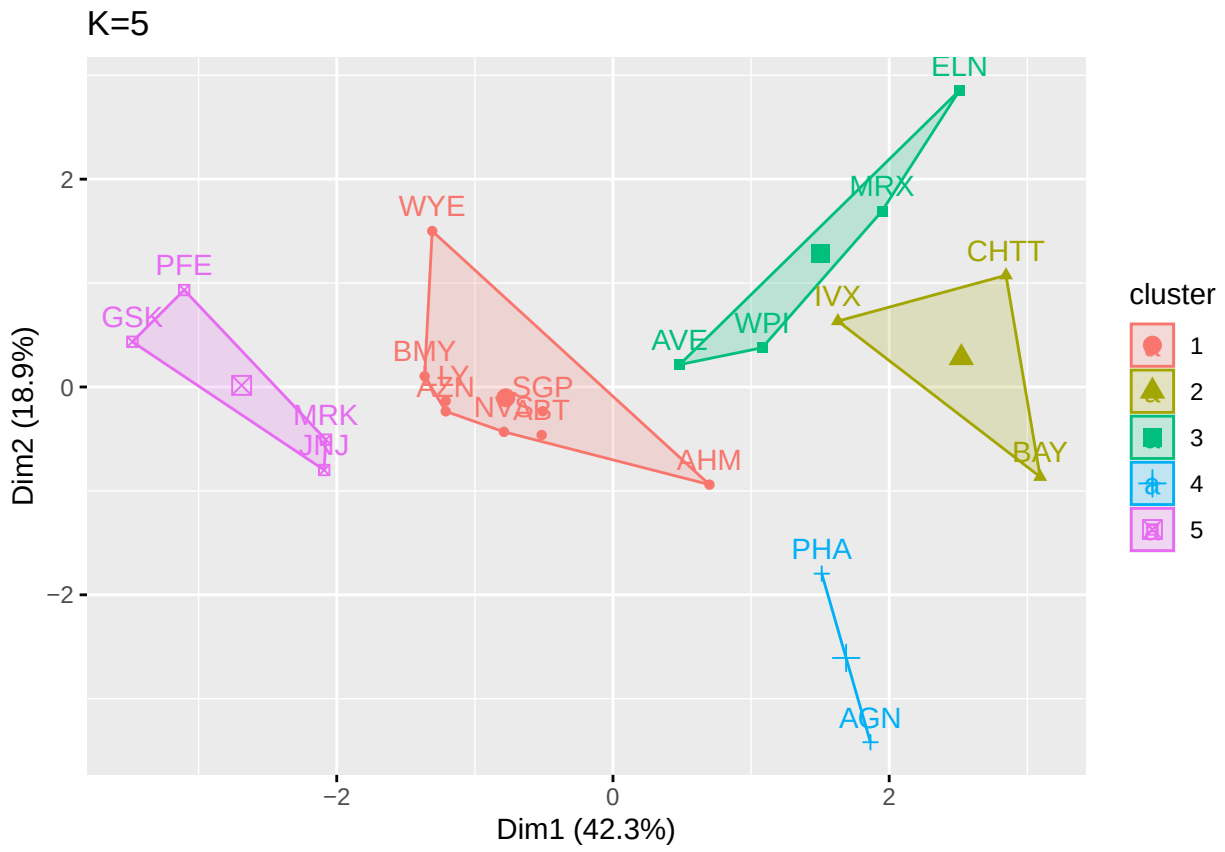
Applying Kmeans for appropriate k

```
set.seed(143)
kmeans5<-kmeans(Scaled_data,centers = 5, nstart = 10)
kmeans5
```

```
## 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"
```

```
plot5kmeans<-fviz_cluster(kmeans5,data = Scaled_data) + ggtitle("K=5")
plot5kmeans
```




```
cluster_data_1<-cluster_data%>%
  mutate(Cluster_no=kmeans5$cluster)%>%
  group_by(Cluster_no)%>%summarise_all('mean')
cluster_data_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>
```

Businesses are divided into the following clusters:

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

Cluster_2= BAY,CHTT,IVX

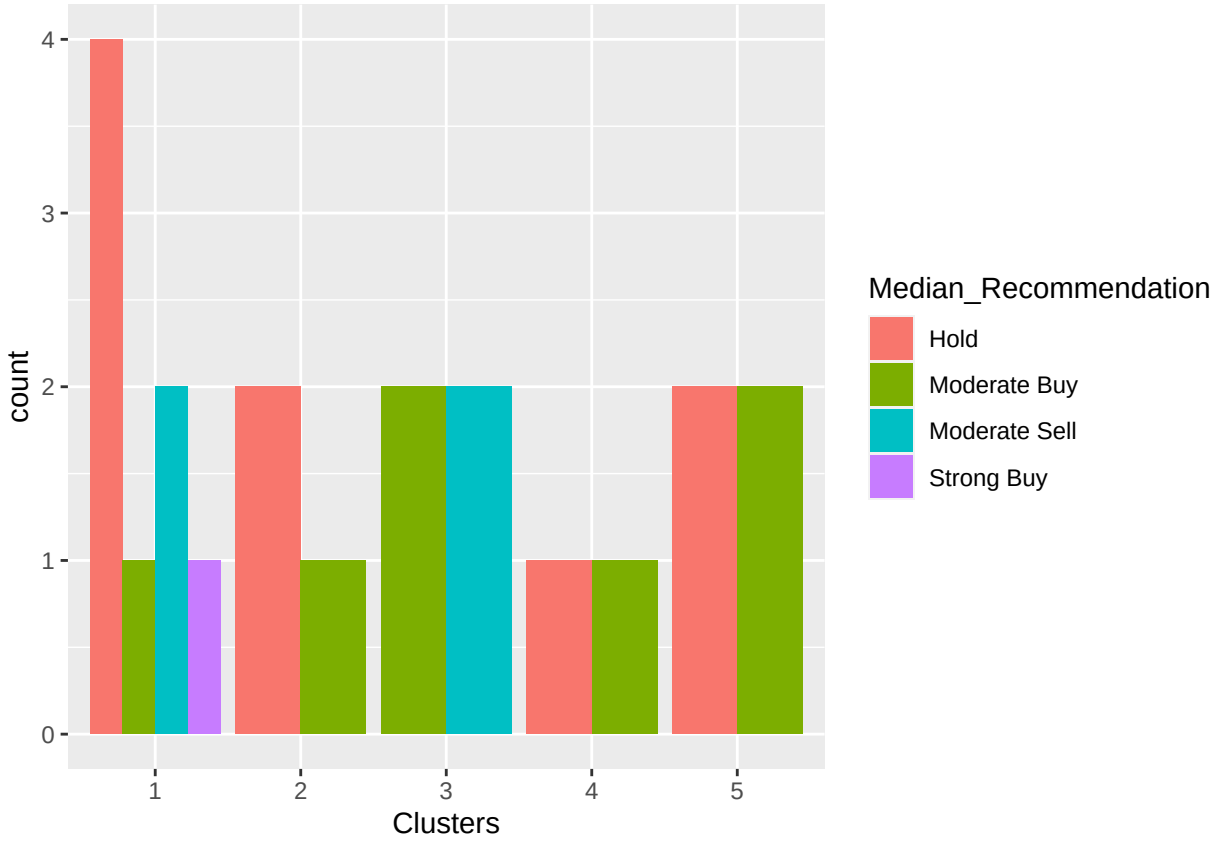
Cluster_3=AVE,ELN,MRX,WPI

Cluster_4=AGN,PHA

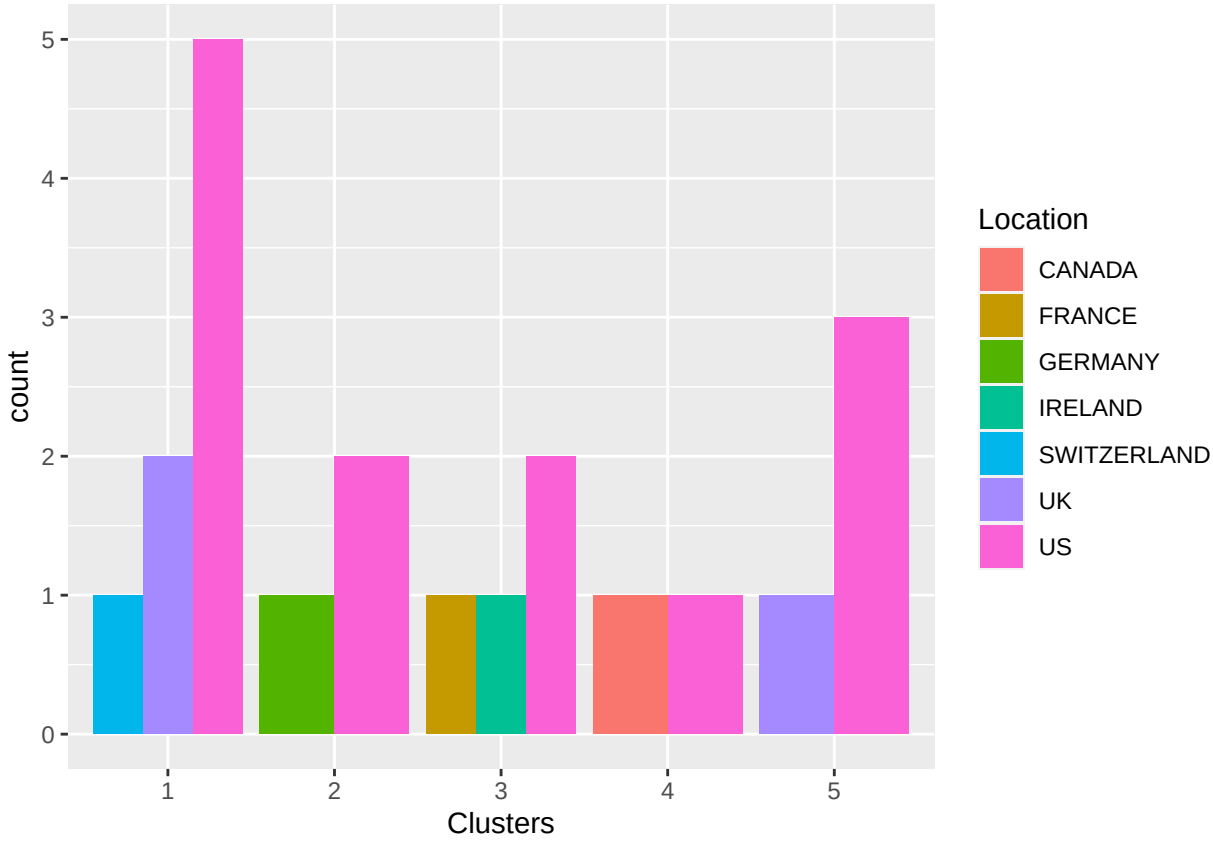
Cluster_5=GSK,JNJ,MRK,PFE

cluster 1: Average returns, moderate risk.
cluster 2: Low returns, high risk.
cluster 3: Slightly lower returns than Group 2, slightly lower risk.
cluster 4: High price-to-earnings ratios, low returns, high risk.
cluster 5: High market value and returns, low risk.

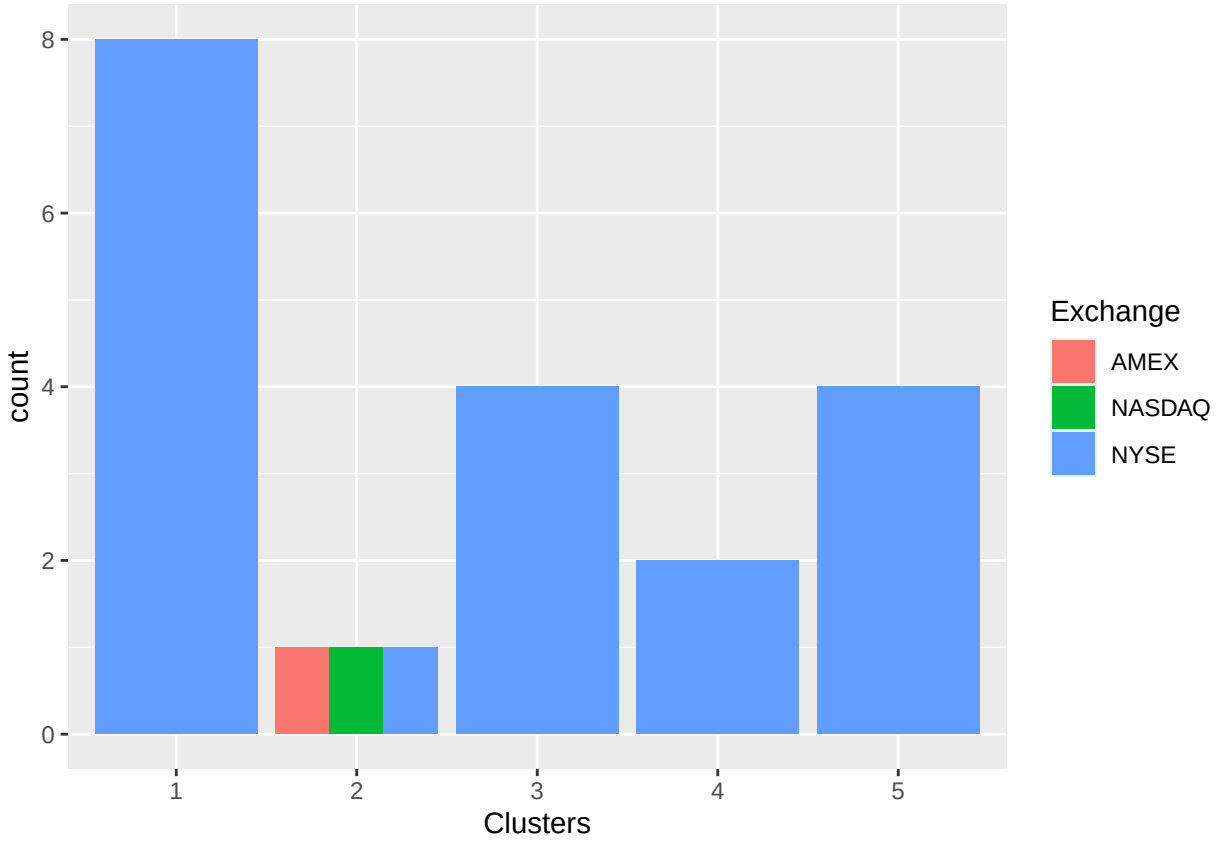
```
Clustering_datase_2<- pharma_data[,12:14] %>% mutate(Clusters=kmeans5$cluster)
ggplot(Clustering_datase_2, mapping = aes(factor(Clusters), fill =Median_Recommendation))+geom_bar(posi
```




```
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
```



With regard to the variable "Median Recommendation," the clusters seem to be trending. For example, suggestions from the second cluster tend to be between "hold" and "moderate buy," but recommendations from the third cluster are between "moderate buy" and "moderate sell." There doesn't appear to be any discernible geographic grouping pattern when considering the locations of the companies, as many of them are found in the US. Similarly, while the majority of the companies are listed on the NYSE, there is no clear pattern tying the stock exchange listings to the clusters.
Naming clusters:

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

Cluster 1: Large companies with thousands of dollars in market capitalization and moderate returns on assets.
Cluster 2: Very small companies with penny stocks and low returns on assets.
Cluster 3: Small companies with dollar stocks and low returns on assets.
Cluster 4: Medium-sized companies with hundreds of dollars in market capitalization and low returns on assets.
Cluster 5: Very large companies with millions of dollars in market capitalization and high returns on assets.

DBSCAN CLUSTERING

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

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

```r
library(dbscan)
```

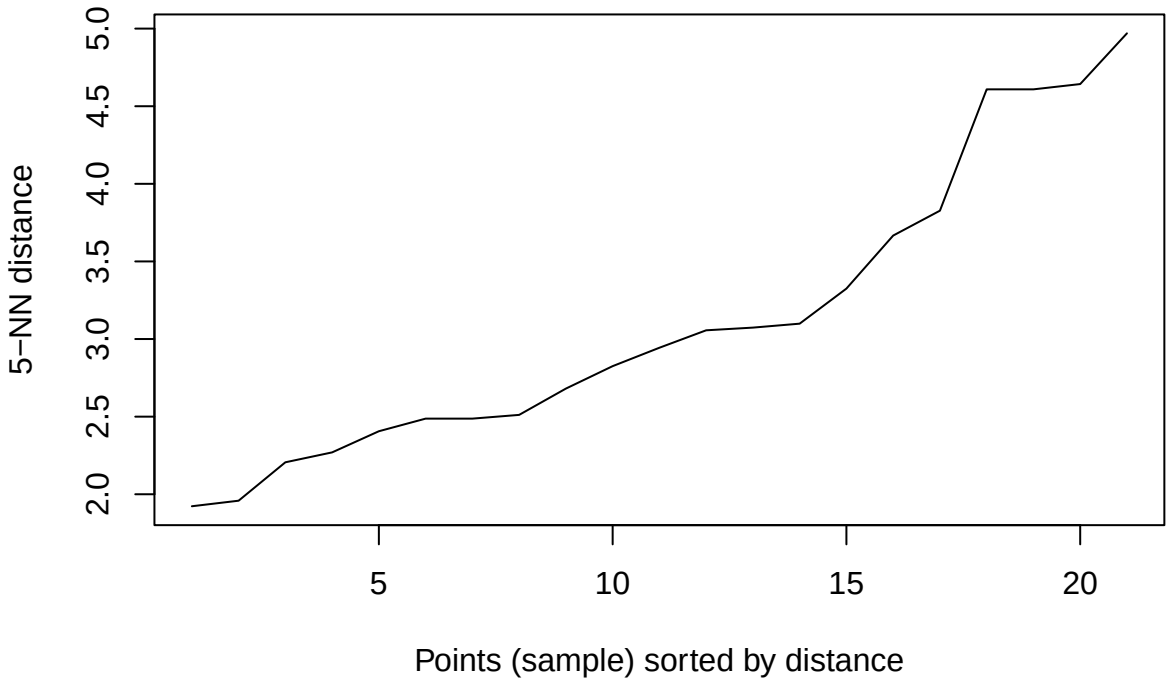
```
## Warning: package 'dbscan' was built under R version 4.3.2
```

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

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

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

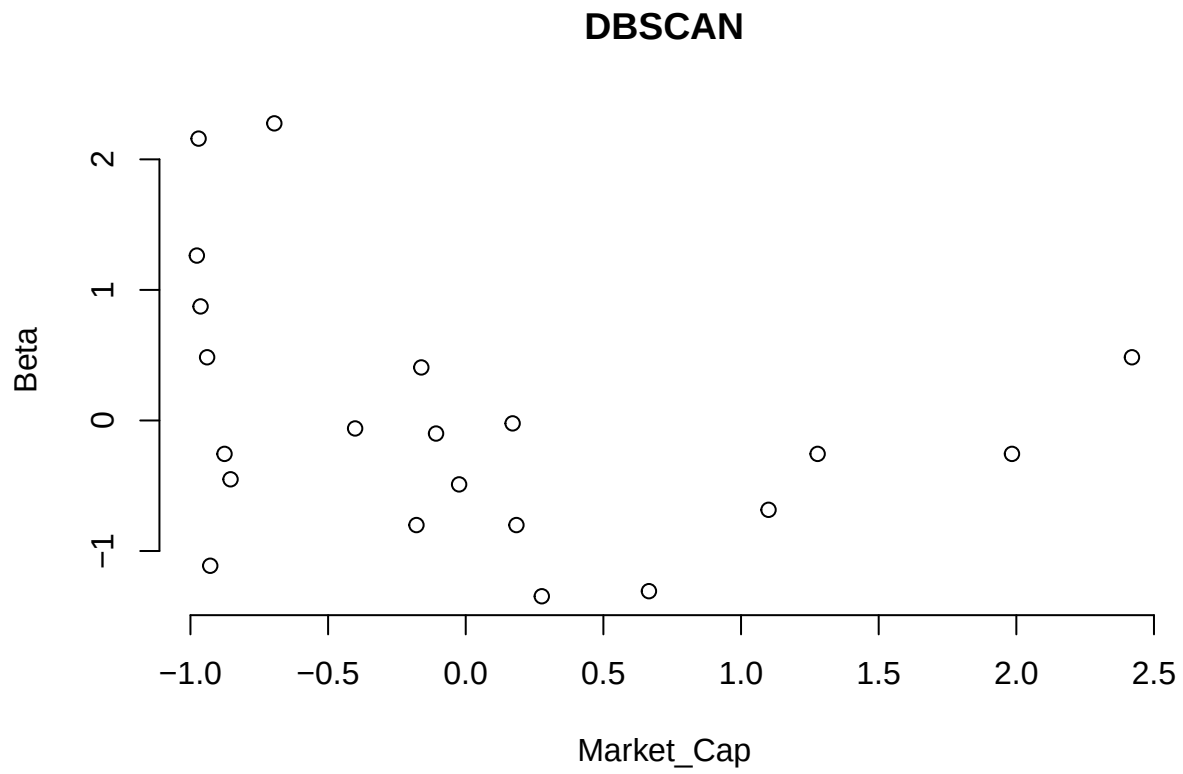
```r
# Use the kNNdistplot to help find a suitable eps value
kNNdistplot(Scaled_data, k = 5)
# Add an abline and try to visually identify the "elbow" point
abline(h = 0.05, col = 'red', lty = 2)  # Start with a small value for eps, adjust based on the plot
```



```r
# 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 0 0 0 0
```

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

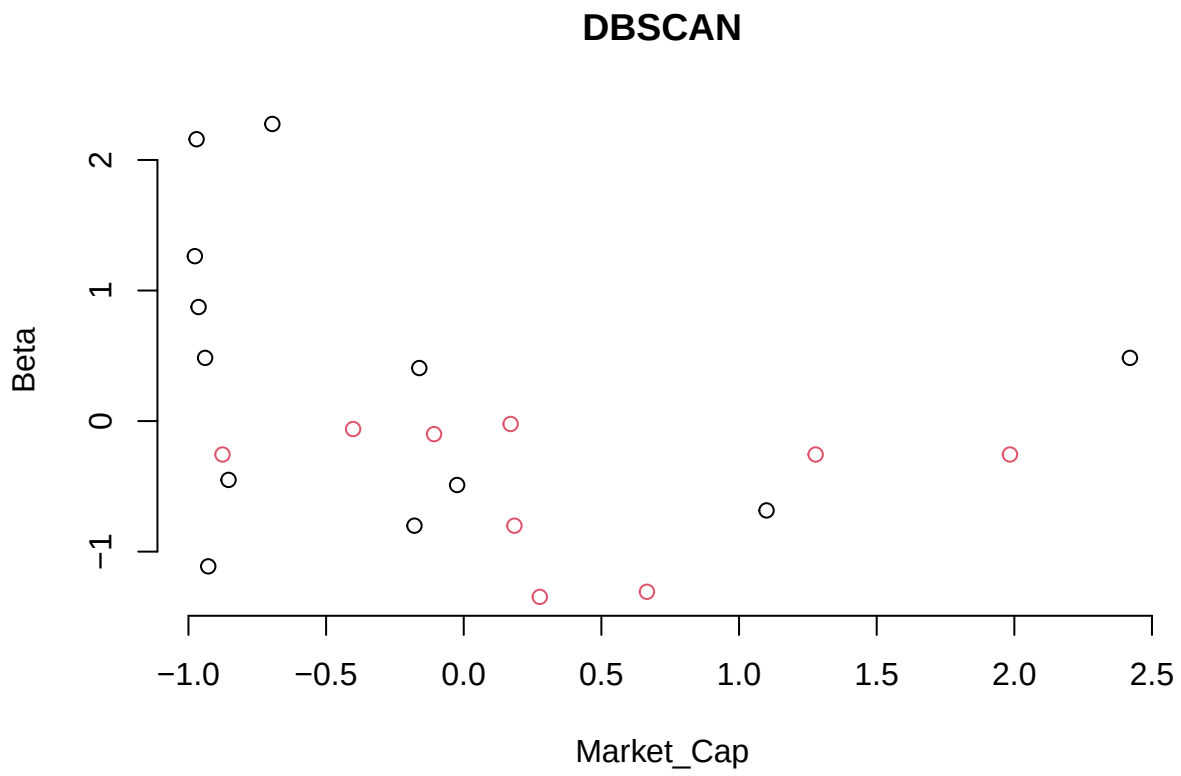



```r
# Cluster 0: This is the main cluster identified by DBSCAN, which includes firms that are close togethe
# Cluster -1: This represents outlier points or noise, which are not sufficiently close to enough point
```

```r
#Using various eps values to improve clustering.
# The output will be 0 if the eps value is too low.
# The result will be 1 if the eps value is excessively high.
# Designating 2 as the ideal eps value.
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
```

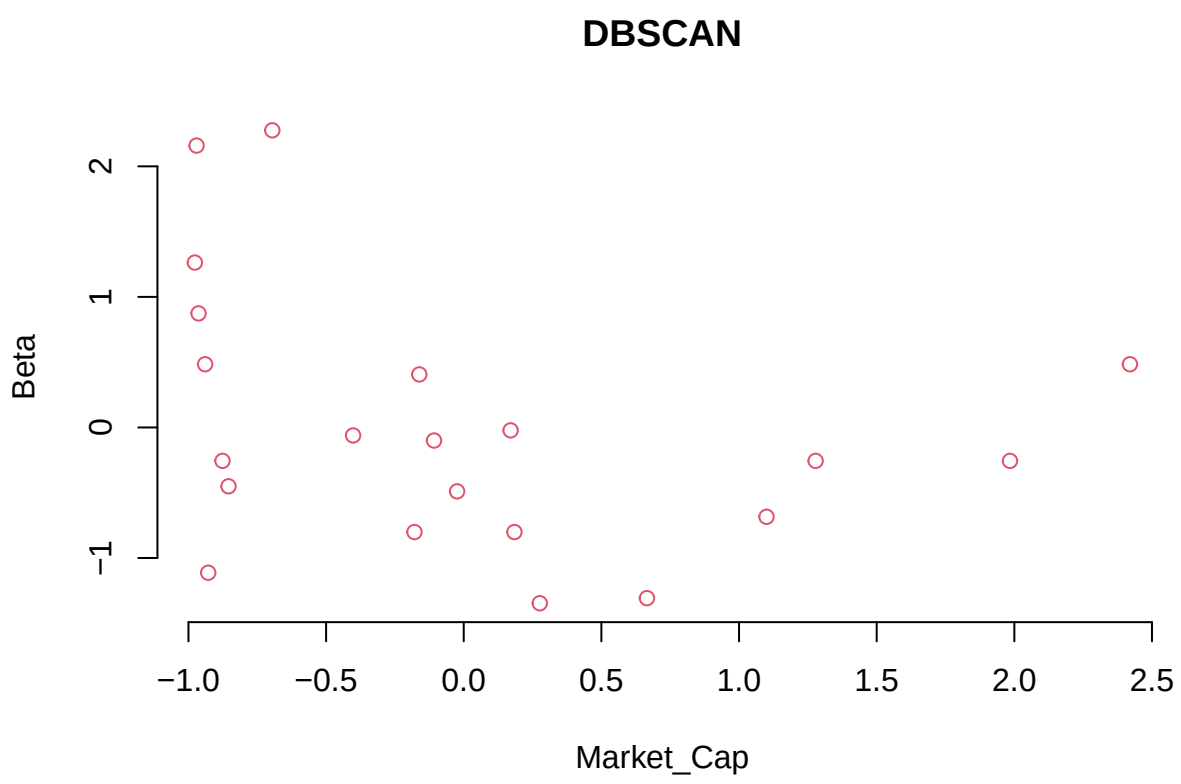
```
plot(dbscan_result_2, Scaled_data, main= "DBSCAN", frame= FALSE)
```




```
#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 1
```

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



HIERARCHICAL CLUSTERING

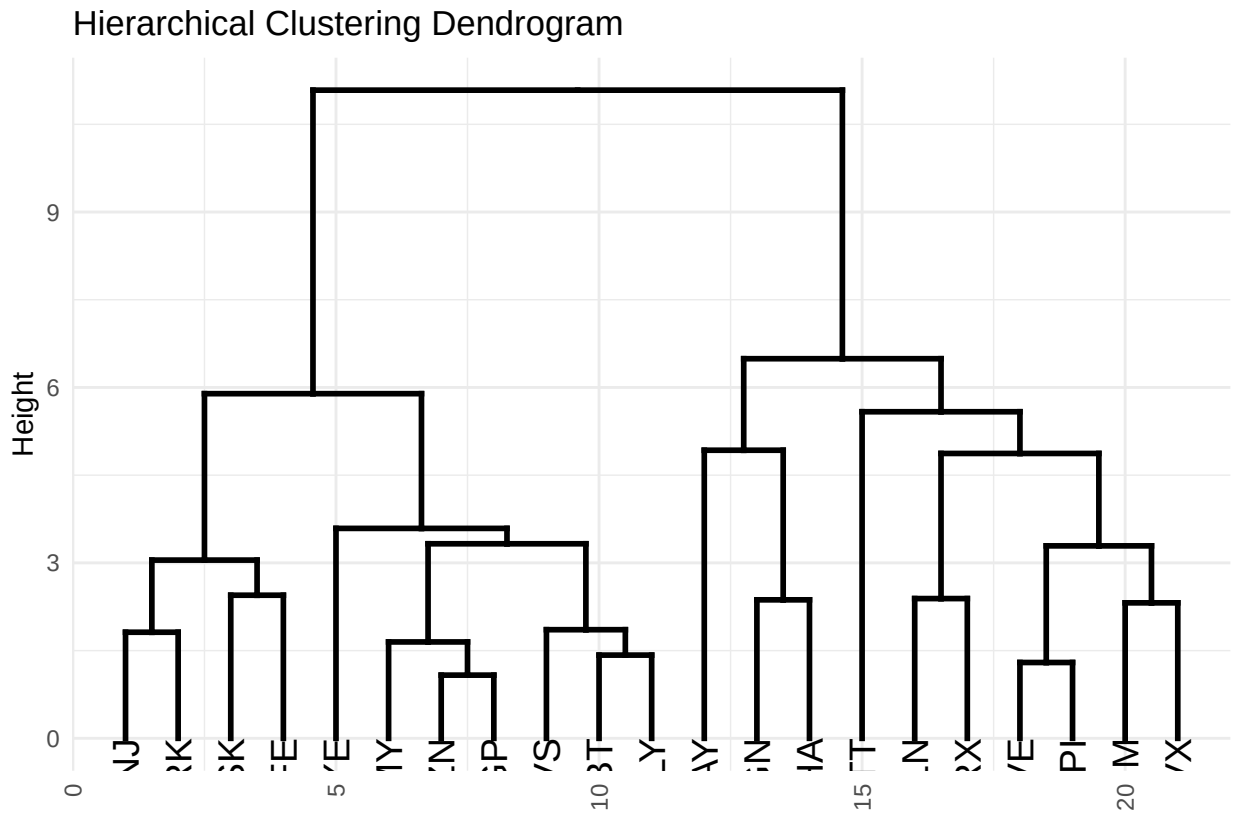
```
# Loading necessary library
library(stats)
# Hierarchical clustering using Ward's method
hc_result <- hclust(dist(Scaled_data), method = "ward.D2")

# 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: The algorithm has found two clusters, denoted as 0 and 1, and has classified some points as -1, meaning they are noise. DBSCAN's silhouette score is roughly 0.052, which is a very low number. This implies that the DBSCAN-defined clusters are neither highly separated or dense.
Hierarchical Clustering: Since DBSCAN was unable to produce a sufficient number of groups, I arbitrarily selected three clusters for hierarchical clustering. Although this is better than the DBSCAN result, the silhouette score for hierarchical clustering is still at 0.273, indicating substantial cluster overlap or cluster structure. Since the DBSCAN produced basically one cluster when noise was ignored, I used two clusters for hierarchical clustering. It seems that a more realistic silhouette score was obtained using hierarchical clustering with two clusters.

For these clustering algorithms, there is no right or wrong response. I used the dataset to apply the K-Means, DBSCAN, and Hierarchical clustering algorithms, and I found that each clustering methodology has a unique significance.

For partitioning techniques, K-Means is a decent place to start, particularly if you have a solid idea of how many clusters there are.
When clusters are not always globular and there is noise in the data, DBSCAN works best.
When a visual representation of the clusters is helpful and exploratory data analysis is required, Hierarchical Clustering excels.

In conclusion, even if every algorithm has benefits of its own, the type of dataset should determine which approach is used.

Selection of Clustering:

I found that the k=5 cluster had a better graph and a better comprehension of clusters after viewing all the clustering strategies, so I believe that k-means clustering is a lot better clustering technique for this dataset.

Let's analyze the cluster and k-means values: Taking into account both the clustering and non-clustering variables, the clusters can be interpreted as follows

Cluster Properties Determining by Clustering Variables
Comparing Cluster 0 to Cluster 1, the latter has a greater average beta (which suggests possibly higher volatility) and a lower average market capitalization. In addition, the average PE Ratio is higher than that of Cluster 1, although the ROE and ROA are lower. Together with increased sales growth and average leverage, this cluster also has a lower net profit margin.
Both Cluster 1's average market capitalization and beta (a measure of volatility) are noticeably greater. Because the PE Ratio is lower, the price-to-earnings ratio may be better. Its operations are generally more profitable and efficient, as seen by its greater ROE and ROA. Compared to Cluster 0, this cluster has a larger net profit margin, less leverage, and slower revenue growth.
Patterns for Non-Clustering Numerical Variables: Revenue Growth (Rev_Growth): Cluster 0 has a higher mean revenue growth, but both clusters have negative most common (mode) values, which could mean that a decline in revenue growth is the most common trend among the companies in both clusters.
Net Profit Margin: With a substantially greater average net profit margin, Cluster 1 performs better than Cluster 0. Additionally, Cluster 1's net profit margin mode is higher. The mode for the categorical variables was determined; however, the mode for non-numeric data is not shown here because of the constraints in this context. To find patterns or trends, you would typically examine the most prevalent Exchange, Location, and Median Recommendation for each cluster.

These findings could lead to the naming of clusters based on the traits that define them, like:

Cluster 0: "High Growth, High Leverage": These businesses may be in a growth period but are also more risky due to their higher revenue growth and leverage.
Cluster 1: "Stable, Profitable Giants": these companies have substantial market capitalizations, steady operations with low beta, and increased profits. These titles are suggestive; domain knowledge would help them more accurately capture the essence of the businesses in each cluster. The non-clustering variables' patterns in the clusters point to possible areas for additional research, such as the reasons for some high-leverage, high-growth enterprises' diminishing revenue growth models.