FML_4

Atshaya Suresh

2023-11-12

```
# Choose a CRAN mirror URL from https://cran.r-project.org/mirrors.html
# Replace 'https://cran.r-project.org/' with the URL of your chosen mirror
cran_mirror <- 'https://cran.r-project.org/'

# Set the CRAN mirror
options(repos = structure(c(CRAN = cran_mirror)))
```

Questions

1. 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.

Ans) The numeric variables used in the algorithm are: "Market_Cap", "Beta", "PE_Ratio", "ROE", "ROA", "Asset_Turnover","Leverage", "Rev_Growth", "Net_Profit_Margin".

Choice of Algorithm: K-means clustering is chosen for its simplicity and ease of implementation. It partitions the data into 'k' clusters based on similarity. The number of clusters (k) is a parameter that should be chosen based on the characteristics of the data or through techniques like the elbow method.Although DBSCAN clustering was conducted, it did not render any meaningful clusters for various "eps" and "minPts".Even though it formed a number of clusters, the reason why DBSCAN could not form meaningful clusters is that, the data is pretty noisy. Hence, K-means Clustering is chosen. Also, one of the main reasons for not choosing Hierarchical clustering is that, if a data point is placed in any cluster, there is no means to change the cluster.

Number of Clusters: The number of clusters chosen is 3. Computationally, the optimal number of Clusters is 9. However, from the elbow method we can observe that, the WSS drops close to 3 clusters. Further, when we choose the number of clusters as 9, there are many clusters with very less elements which makes the clustering noisy. Hence, the number of clusters is chosen to be 3.

Standardization: The numerical variables are standardized using the scale function to ensure that all variables contribute equally to the clustering process. Standardization is crucial when using distance-based algorithms like k-means.

Random Seed: Setting the seed (set.seed(123)) ensures reproducibility. If we run the analysis again with the same seed, we should get the same results.

Weights for different Variables: In k-means clustering, the algorithm does not inherently consider variable weights. All variables are treated equally in terms of their contribution to the computation of distances between data points. However, if we have domain knowledge suggesting that certain variables are more important than others, we might consider adjusting the weights manually before clustering. This is often done by scaling or transforming the variables.We can also use Dimentionality reduction methods like Lasso or Ridge methods to evaluate which variables are important to assign weights.

2. Interpret the clusters with respect to the numerical variables used in forming the clusters. Is there a pattern in the clusters with respect to the numerical variables (10 to 12)? (those not used in forming the clusters)

*Ans)*Cluster Interpretation: With respect to the Numerical Variables

Cluster 1:

Market_Cap: Low

Beta: Moderate

PE_Ratio: High

ROE: Low

ROA: Low

Asset_Turnover: Around the mean

Leverage: Low

Rev_Growth: Positive

Net_Profit_Margin: Low

Cluster 2:

Market_Cap: Moderate to High

Beta: Moderate

PE_Ratio: Low to Moderate

ROE: Moderate to High

ROA: Moderate to High

Asset_Turnover: Positive (around the mean or higher)

Leverage: Low to Moderate

Rev_Growth: Negative to Positive

Net_Profit_Margin: Moderate

Cluster 3:

Market_Cap: Low to Moderate

Beta: High

PE_Ratio: Low to Moderate

ROE: Low to Moderate

ROA: Low to Moderate

Asset_Turnover: Negative (below the mean)

Leverage: Moderate to High

Rev_Growth: Positive

Net_Profit_Margin: Low to Moderate

Analyzing the additional features (Median_Recommendation, Location, Exchange) for each cluster and see if there are any patterns or commonalities within each cluster:

Cluster 1: Median_Recommendation: Mix of recommendations (Moderate Buy, Hold, Moderate Sell) Location: Canada, UK, France, US Exchange: NYSE Interpretation: This cluster seems to have a diverse set of recommendations, with a presence in different countries. All firms are listed on the NYSE.

Cluster 2: Median_Recommendation: Mix of recommendations (Moderate Buy, Moderate Sell, Hold) Location: US, UK, Switzerland Exchange: NYSE Interpretation: Similar to Cluster 1, this cluster has a mix of recommendations and a global presence. All firms are listed on the NYSE.

Cluster 3: Median_Recommendation: Mix of recommendations (Hold, Moderate Buy, Moderate Sell) Location: Germany, US, Ireland Exchange: NYSE, NASDAQ, AMEX Interpretation: This cluster also has a mix of recommendations and a presence in different countries. Firms are listed on various exchanges, including NYSE, NASDAQ, and AMEX. Observations:

Clusters 1 and 2 share similarities in having a mix of recommendations and a global presence, primarily on the NYSE. Cluster 3, while also having a mix of recommendations, shows a wider variety of exchange listings, including NYSE, NASDAQ, and AMEX. It appears that there is some consistency in the types of recommendations across clusters, but the diversity of locations and exchanges indicates that the clustering is not solely based on these features. The financial metrics used for clustering captures underlying patterns that are not directly reflected in the location, exchange, or median recommendations.

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

Ans) **Cluster 1: "Stable Performers"** This cluster seems to consist of firms with lower market capitalization, moderate risk (Beta), high Price/Earnings ratio, lower return metrics (ROE and ROA), and a mix of positive revenue growth and lower net profit margins.

Cluster 2: "Balanced Performers" This cluster includes firms with moderate to high market capitalization, moderate risk, a balanced combination of financial metrics (moderate to high ROE and ROA), positive asset turnover, and a mix of negative to positive revenue growth and moderate net profit margins.

Cluster 3: "High-Risk, High-Reward" Firms in this cluster exhibit a combination of lower to moderate market capitalization, higher risk (high Beta), low to moderate Price/Earnings ratio, lower return metrics, negative asset turnover, and higher leverage. The firms may have positive revenue growth but lower net profit margins.

```r
p.df <- read.csv("Pharmaceuticals.csv")
str(p.df)
```

```
## 'data.frame':    21 obs. of  14 variables:
##  $ Symbol               : chr  "ABT" "AGN" "AHM" "AZN" ...
##  $ Name                 : chr  "Abbott Laboratories" "Allergan, Inc." "Amersham plc" "AstraZeneca PL
##  $ Market_Cap           : num  68.44 7.58 6.3 67.63 47.16 ...
##  $ Beta                 : num  0.32 0.41 0.46 0.52 0.32 1.11 0.5 0.85 1.08 0.18 ...
##  $ PE_Ratio             : num  24.7 82.5 20.7 21.5 20.1 27.9 13.9 26 3.6 27.9 ...
##  $ ROE                  : num  26.4 12.9 14.9 27.4 21.8 3.9 34.8 24.1 15.1 31 ...
##  $ ROA                  : num  11.8 5.5 7.8 15.4 7.5 1.4 15.1 4.3 5.1 13.5 ...
##  $ Asset_Turnover       : num  0.7 0.9 0.9 0.9 0.6 0.6 0.9 0.6 0.3 0.6 ...
##  $ Leverage             : num  0.42 0.6 0.27 0 0.34 0 0.57 3.51 1.07 0.53 ...
##  $ Rev_Growth           : num  7.54 9.16 7.05 15 26.81 ...
##  $ Net_Profit_Margin    : num  16.1 5.5 11.2 18 12.9 2.6 20.6 7.5 13.3 23.4 ...
##  $ Median_Recommendation: chr  "Moderate Buy" "Moderate Buy" "Strong Buy" "Moderate Sell" ...
##  $ Location             : chr  "US" "CANADA" "UK" "UK" ...
##  $ Exchange             : chr  "NYSE" "NYSE" "NYSE" "NYSE" ...
```

```r
numeric.df <- p.df[, c("Market_Cap", "Beta", "PE_Ratio", "ROE", "ROA", "Asset_Turnover",
                       "Leverage", "Rev_Growth", "Net_Profit_Margin")]
numeric.df
```

```
##    Market_Cap Beta PE_Ratio  ROE  ROA Asset_Turnover Leverage Rev_Growth
## 1       68.44 0.32     24.7 26.4 11.8            0.7     0.42       7.54
## 2        7.58 0.41     82.5 12.9  5.5            0.9     0.60       9.16
## 3        6.30 0.46     20.7 14.9  7.8            0.9     0.27       7.05
## 4       67.63 0.52     21.5 27.4 15.4            0.9     0.00      15.00
## 5       47.16 0.32     20.1 21.8  7.5            0.6     0.34      26.81
## 6       16.90 1.11     27.9  3.9  1.4            0.6     0.00      -3.17
## 7       51.33 0.50     13.9 34.8 15.1            0.9     0.57       2.70
## 8        0.41 0.85     26.0 24.1  4.3            0.6     3.51       6.38
## 9        0.78 1.08      3.6 15.1  5.1            0.3     1.07      34.21
## 10      73.84 0.18     27.9 31.0 13.5            0.6     0.53       6.21
## 11     122.11 0.35     18.0 62.9 20.3            1.0     0.34      21.87
## 12       2.60 0.65     19.9 21.4  6.8            0.6     1.45      13.99
## 13     173.93 0.46     28.4 28.6 16.3            0.9     0.10       9.37
## 14       1.20 0.75     28.6 11.2  5.4            0.3     0.93      30.37
## 15     132.56 0.46     18.9 40.6 15.0            1.1     0.28      17.35
## 16      96.65 0.19     21.6 17.9 11.2            0.5     0.06      -2.69
## 17     199.47 0.65     23.6 45.6 19.2            0.8     0.16      25.54
## 18      56.24 0.40     56.5 13.5  5.7            0.6     0.35      15.00
## 19      34.10 0.51     18.9 22.6 13.3            0.8     0.00       8.56
## 20       3.26 0.24     18.4 10.2  6.8            0.5     0.20      29.18
## 21      48.19 0.63     13.1 54.9 13.4            0.6     1.12       0.36
##    Net_Profit_Margin
## 1               16.1
## 2                5.5
## 3               11.2
## 4               18.0
## 5               12.9
## 6                2.6
## 7               20.6
## 8                7.5
## 9               13.3
## 10              23.4
## 11              21.1
## 12              11.0
## 13              17.9
## 14              21.3
## 15              14.1
## 16              22.4
## 17              25.2
## 18               7.3
## 19              17.6
## 20              15.1
## 21              25.5
```

```r
numeric_std <- scale(numeric.df)
```

```r
library(cluster)

# K-means clustering
set.seed(123)  # Set seed for reproducibility
num_clusters <- 3  # You can adjust this based on your analysis

# Perform k-means clustering
kmeans_model <- kmeans(numeric_std, centers = num_clusters, nstart = 10)

# Add cluster assignments to the original dataframe
p.df$cluster <- as.factor(kmeans_model$cluster)

# Print the cluster assignments
print(p.df$cluster)
```

```
##  [1] 2 1 1 2 1 3 2 3 3 2 2 3 2 3 2 2 2 1 2 1 2
## Levels: 1 2 3
```

```r
# Summary of the cluster centers
print(kmeans_model$centers)
```

```
##   Market_Cap       Beta   PE_Ratio        ROE        ROA Asset_Turnover
## 1 -0.5723845 -0.6220844  0.8692748 -0.7381675 -0.7242993   1.554312e-16
## 2  0.6733825 -0.3586419 -0.2763512  0.6565978  0.8344159   4.612656e-01
## 3 -0.9090570  1.4110965 -0.2613021 -0.7063477 -1.1114156  -1.014784e+00
##     Leverage Rev_Growth Net_Profit_Margin
## 1 -0.2991312  0.3682951        -0.8069490
## 2 -0.3331068 -0.2902163         0.6823310
## 3  1.0319661  0.2701808        -0.6941793
```

```r
install.packages("factoextra")
```

```
## Installing package into 'C:/Users/Atshaya Suresh/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'factoextra' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Atshaya Suresh\AppData\Local\Temp\RtmpIluTVG\downloaded_packages
```

```r
# Load necessary libraries for visualization
library(ggplot2)
library(factoextra)
```

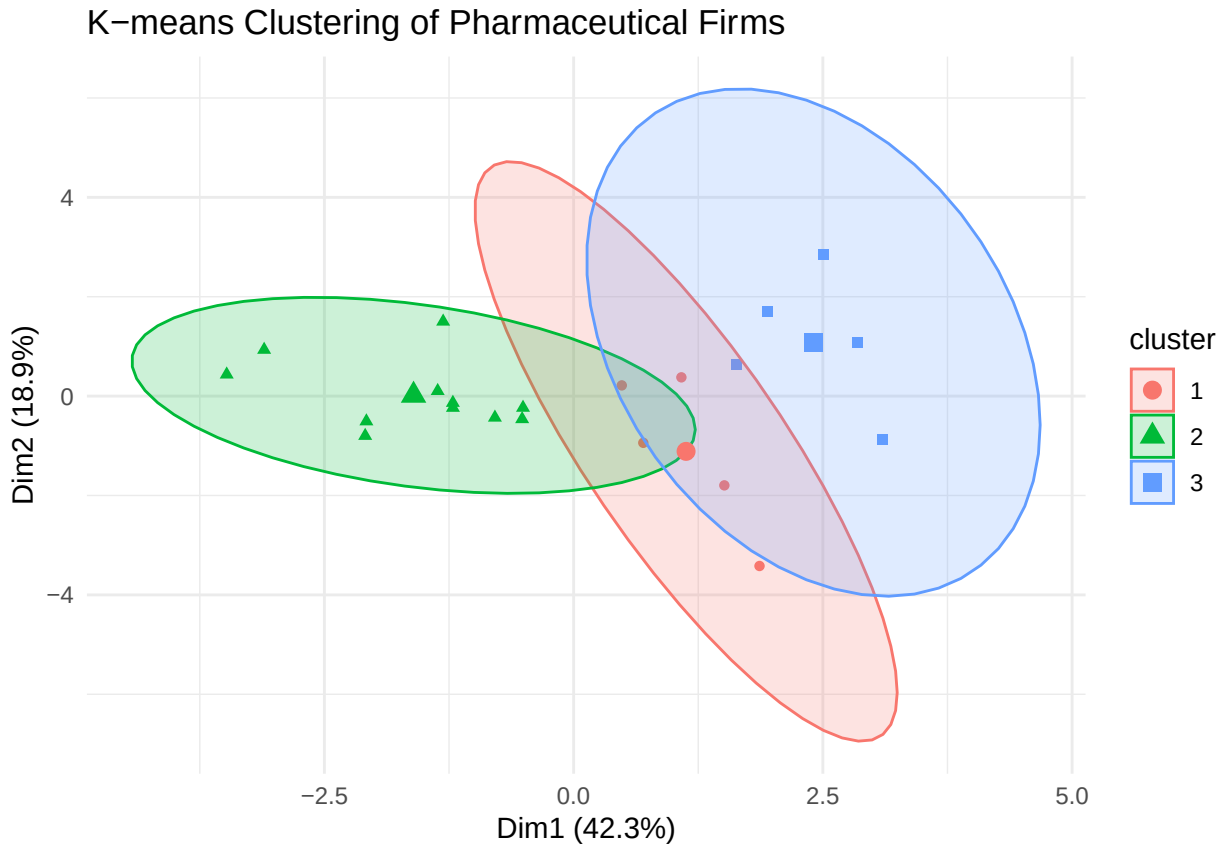
```
## Warning: package 'factoextra' was built under R version 4.3.2

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

```
# Cluster plot
fviz_cluster(kmeans_model, data = numeric_std,
             geom = "point",
             stand = FALSE,
             frame.type = "norm") +
  ggtitle("K-means Clustering of Pharmaceutical Firms") +
  theme_minimal()
```

```
## Warning: argument frame is deprecated; please use ellipse instead.
```

```
## Warning: argument frame.type is deprecated; please use ellipse.type instead.
```



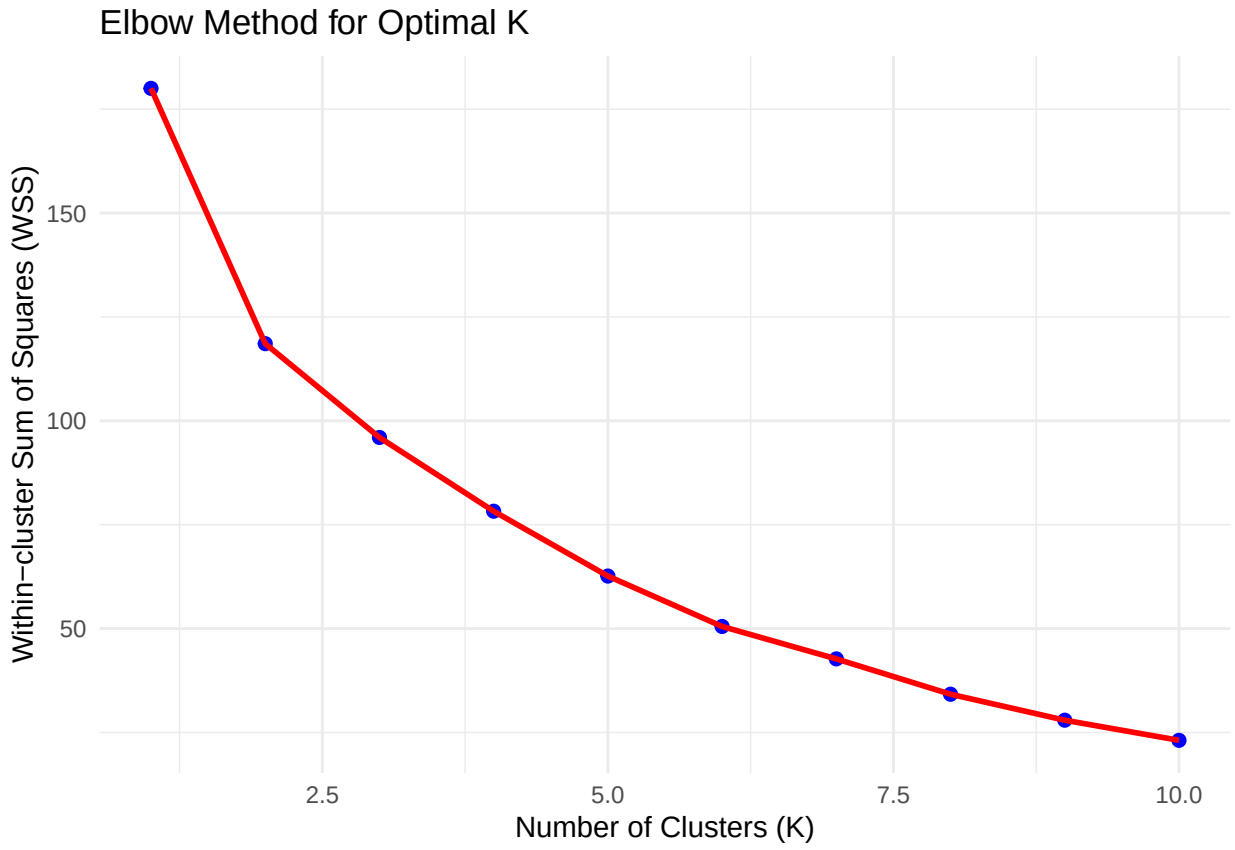

```
# Elbow method plot
wss <- numeric(10)
for (i in 1:10) {
  kmeans_model <- kmeans(numeric_std, centers = i, nstart = 10)
  wss[i] <- sum(kmeans_model$withinss)
}

elbow_plot <- ggplot() +
  geom_point(aes(x = 1:10, y = wss), color = "blue", size = 2) +
  geom_line(aes(x = 1:10, y = wss), color = "red", size = 1) +
  labs(title = "Elbow Method for Optimal K",
       x = "Number of Clusters (K)",
```

```
       y = "Within-cluster Sum of Squares (WSS)") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
# Print the plots
print(elbow_plot)
```



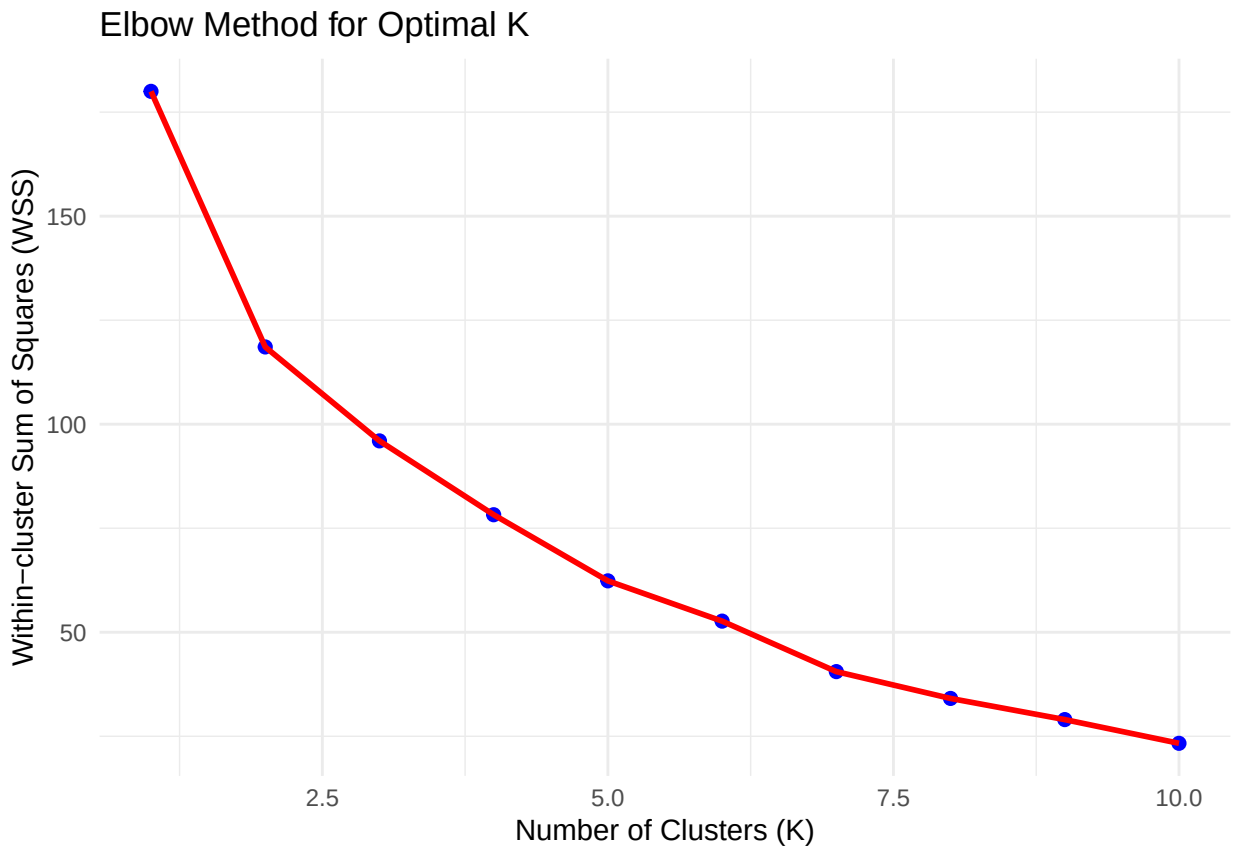

```
# Load necessary libraries
library(cluster)
library(ggplot2)

#numeric_std is your standardized data
wss <- numeric(10)  # Initialize a vector to store within-cluster sum of squares

# Iterate through different values of k (number of clusters)
for (i in 1:10) {
  kmeans_model <- kmeans(numeric_std, centers = i, nstart = 10)
  wss[i] <- sum(kmeans_model$withinss)
}
```

```r
# Create an elbow plot
elbow_plot <- ggplot() +
  geom_point(aes(x = 1:10, y = wss), color = "blue", size = 2) +
  geom_line(aes(x = 1:10, y = wss), color = "red", size = 1) +
  labs(title = "Elbow Method for Optimal K",
       x = "Number of Clusters (K)",
       y = "Within-cluster Sum of Squares (WSS)") +
  theme_minimal()

# Print the elbow plot
print(elbow_plot)
```




```r
# Find the optimal K (number of clusters) programmatically
optimal_k <- which(diff(wss) == max(diff(wss))) + 1
cat("Optimal K (number of clusters):", optimal_k, "\n")
```

```
## Optimal K (number of clusters): 9
```

```r
# Install the dbscan package
install.packages("dbscan")
```

```
## Installing package into 'C:/Users/Atshaya Suresh/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'dbscan' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'dbscan'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\Atshaya
## Suresh\AppData\Local\R\win-library\4.3\00LOCK\dbscan\libs\x64\dbscan.dll to
## C:\Users\Atshaya
## Suresh\AppData\Local\R\win-library\4.3\dbscan\libs\x64\dbscan.dll: Permission
## denied

## Warning: restored 'dbscan'

##
## The downloaded binary packages are in
##  C:\Users\Atshaya Suresh\AppData\Local\Temp\RtmpIluTVG\downloaded_packages
```

```
# Load the dbscan package
library(dbscan)
```

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

##
## Attaching package: 'dbscan'

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

```
# Assuming we have already loaded the necessary libraries
library(cluster)

# DBSCAN clustering
dbscan_model <- dbscan(numeric_std, eps = 2.4, minPts = 2)

# Add cluster assignments to the original dataframe
p.df$cluster_dbscan <- as.factor(dbscan_model$cluster)

# Print the DBSCAN cluster assignments
print(p.df$cluster_dbscan)
```

```
##  [1] 1 2 1 1 1 0 1 0 3 1 1 1 1 3 1 1 0 2 1 1 1
## Levels: 0 1 2 3
```

```
# Summary of the DBSCAN clusters
print(table(p.df$cluster_dbscan))
```

```
##
##  0  1  2  3
##  3 14  2  2
```

```r
# Load necessary libraries for visualization
library(factoextra)

# Visualize DBSCAN clusters
fviz_cluster(dbscan_model, data = numeric_std,
             geom = "point",
             stand = FALSE,
             frame.type = "norm") +
  ggtitle("DBSCAN Clustering of Pharmaceutical Firms") +
  theme_minimal()
```

```
## Warning: argument frame is deprecated; please use ellipse instead.

## Warning: argument frame.type is deprecated; please use ellipse.type instead.

## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
```