FML Assignment_4 CLUSTERING

ALLEN RICHARDS

2023-11-10

Statement : Directions - An equities analyst is studying the pharmaceutical industry and would like your help in exploring and understanding the financial data collected by her firm. Her main objective is to understand the structure of the pharmaceutical industry using some basic financial measures. Financial data gathered on 21 firms in the pharmaceutical industry are available in the file Pharmaceuticals.csv Download Pharmaceuticals.csv. For each firm, the following variables are recorded:

Market capitalization (in billions of dollars) Beta Price/earnings ratio Return on equity Return on assets Asset turnover Leverage Estimated revenue growth Net profit margin Median recommendation (across major brokerages) Location of firm’s headquarters Stock exchange on which the firm is listed Use cluster analysis to explore and analyze the given dataset as follows:

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

Summary : As per my observation The provided statement describes a situation in which a financial analyst for stocks is examining information from 21 pharmaceutical companies. The goal is to use numerical variable cluster analysis to comprehend the structure of the pharmaceutical industry. Market capitalization, beta, price/earnings ratio, return on equity, return on assets, asset turnover, leverage, projected revenue growth, and net profit margin are just a few of the financial metrics included in the data. So in this R I mentioned various library tools and formulated to get plot diagrams. Since each

explanation mentioned on the codes respectively

```
# Let's add the required library

library(class)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
## Loading required package: lattice
```

```
library(e1071)
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
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ lubridate 1.9.2     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ purrr::lift()   masks caret::lift()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(ISLR)
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
```

```
library(fpc)
```

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

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

```
# Each library serves a specific purpose and provides functions and tools that can be used fo
r various tasks in data analysis, machine learning, and statistics.
```

Let import the CSV file from the directory

```
# Import CSV file
pharma.data <- read.csv("C:/Users/Vishal/Downloads/Pharmaceuticals.csv")
dim(pharma.data)
```

```
## [1] 21 14
```

```
t(t(names(pharma.data)))# The 't' function creates a transpose of the dataframe
```

```
##       [,1]
##  [1,] "Symbol"
##  [2,] "Name"
##  [3,] "Market_Cap"
##  [4,] "Beta"
##  [5,] "PE_Ratio"
##  [6,] "ROE"
##  [7,] "ROA"
##  [8,] "Asset_Turnover"
##  [9,] "Leverage"
## [10,] "Rev_Growth"
## [11,] "Net_Profit_Margin"
## [12,] "Median_Recommendation"
## [13,] "Location"
## [14,] "Exchange"
```

```
# Interpretation : Checking the dimensions of the data frame gives an idea of the number of o
bservations (rows) and variables (columns). Transposing the column names might be done to dis
play them in a different format or orientation
```

Dropping the columns which not required for clustering

```
pharma.data <- pharma.data[ ,-c(1,2,12,13,14)]# 1 and 5 are the indexes for columns ID and ZI
P
dim(pharma.data)
```

```
## [1] 21  9
```

```
summary(pharma.data)
```

```
##    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     
```

```
t(t(names(pharma.data)))
```

```
##       [,1]               
##  [1,] "Market_Cap"       
##  [2,] "Beta"             
##  [3,] "PE_Ratio"         
##  [4,] "ROE"              
##  [5,] "ROA"              
##  [6,] "Asset_Turnover"   
##  [7,] "Leverage"         
##  [8,] "Rev_Growth"       
##  [9,] "Net_Profit_Margin"
```

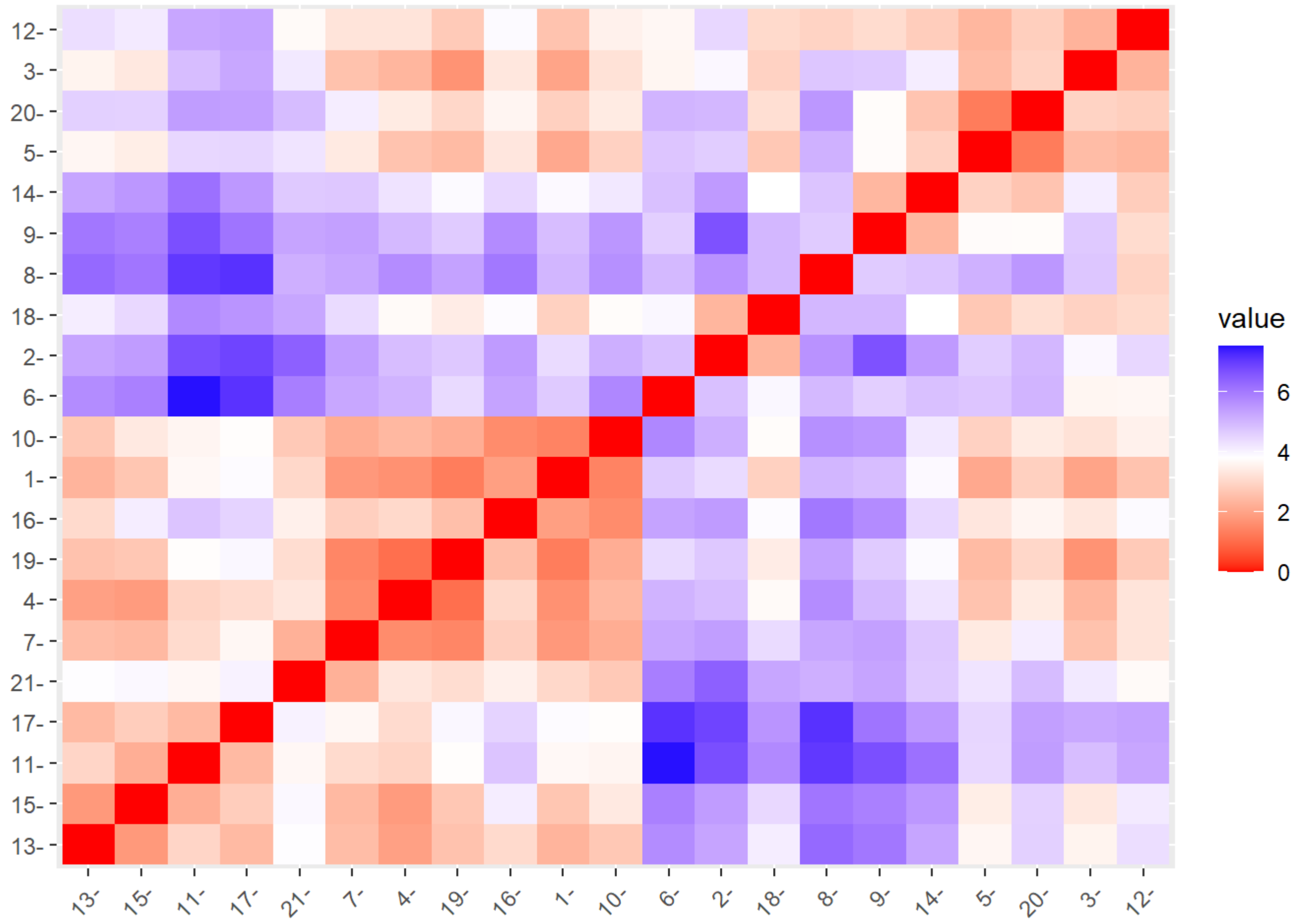
```
# Interpretation : The removal of columns with indexes 1, 2, 12, 13, and 14 may be due to the
fact that the information in those columns is duplicated, unnecessary for the analysis, or de
termined to be unrelated to the work at hand. A brief description of the distribution and cen
tral tendency of the remaining variables in the altered data frame can be obtained by computi
ng summary statistics. This can be useful in comprehending the data's properties.
```

#kmeans

```
pharma.data1 <- scale(pharma.data)
head(pharma.data1)
```

```
##       Market_Cap        Beta    PE_Ratio         ROE        ROA Asset_Turnover
## [1,]  0.1840960 -0.80125356 -0.04671323  0.04009035  0.2416121      0.0000000
## [2,] -0.8544181 -0.45070513  3.49706911 -0.85483986 -0.9422871      0.9225312
## [3,] -0.8762600 -0.25595600 -0.29195768 -0.72225761 -0.5100700      0.9225312
## [4,]  0.1702742 -0.02225704 -0.24290879  0.10638147  0.9181259      0.9225312
## [5,] -0.1790256 -0.80125356 -0.32874435 -0.26484883 -0.5664461     -0.4612656
## [6,] -0.6953818  2.27578267  0.14948233 -1.45146000 -1.7127612     -0.4612656
##        Leverage Rev_Growth Net_Profit_Margin
## [1,] -0.2120979 -0.5277675        0.06168225
## [2,]  0.0182843 -0.3811391       -1.55366706
## [3,] -0.4040831 -0.5721181       -0.68503583
## [4,] -0.7496565  0.1474473        0.35122600
## [5,] -0.3144900  1.2163867       -0.42597037
## [6,] -0.7496565 -1.4971443       -1.99560225
```

```
distance <- get_dist(pharma.data1)
fviz_dist(distance)
```



```
# Interpretation : # Interpretation : The pharmacy's columns are standardized by applying the
scale function.data frame of data. Standardization entails scaling by the standard deviation
and centering the variables by deducting the mean. The code probably belongs to an explorator
y study or is used as a warm-up for a clustering task, where knowing the separations between
data points are crucial. The particular needs of the study or algorithm being utilized to det
ermine which standardization and distance calculation needs to apply.
```

Consider k=3

```
set.seed(159)
k <- 3
k3 <- kmeans(pharma.data1, centers = k, nstart=21)
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 -0.8261772  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
```

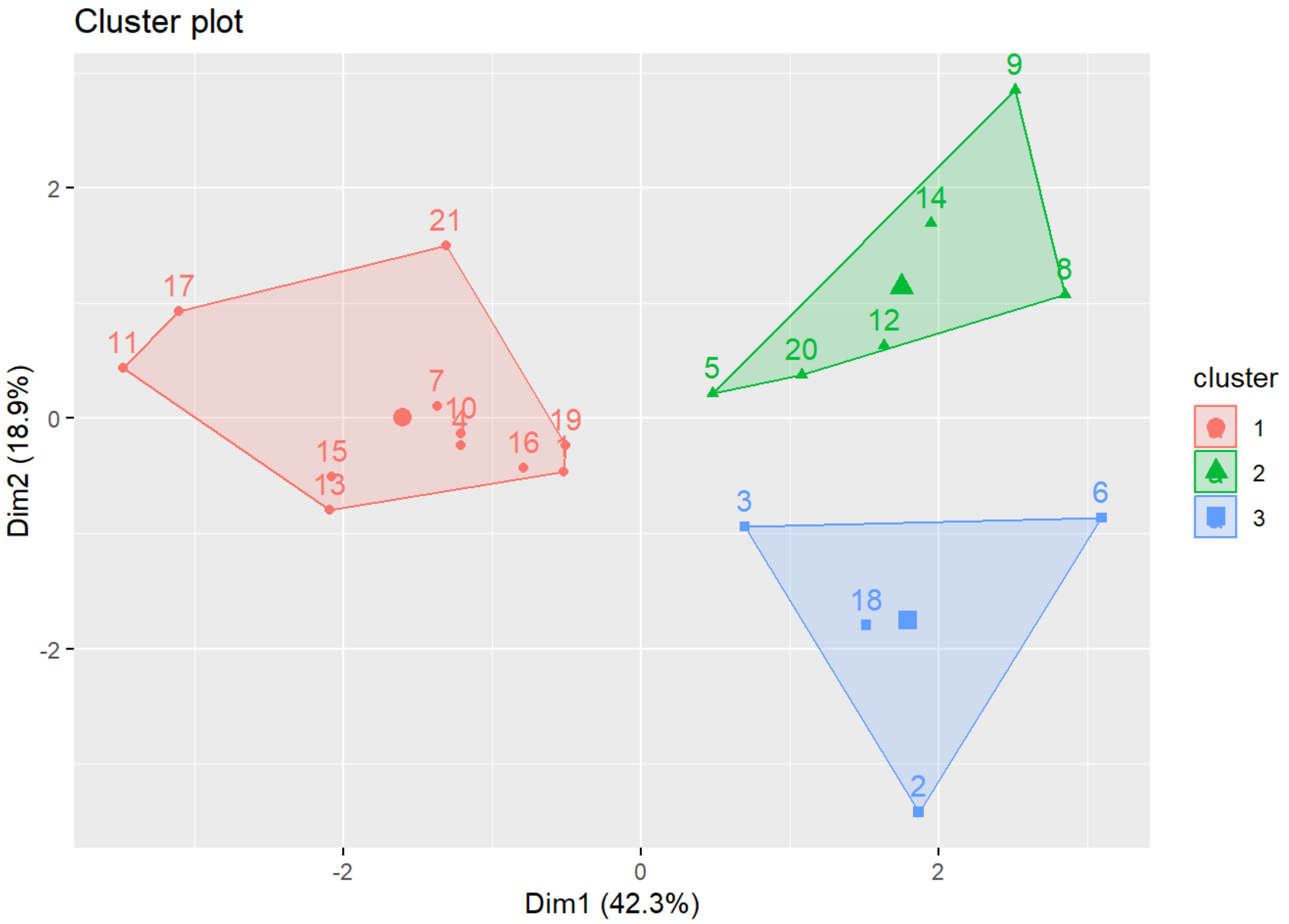
```
k3$size
```

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

```
k3$cluster
```

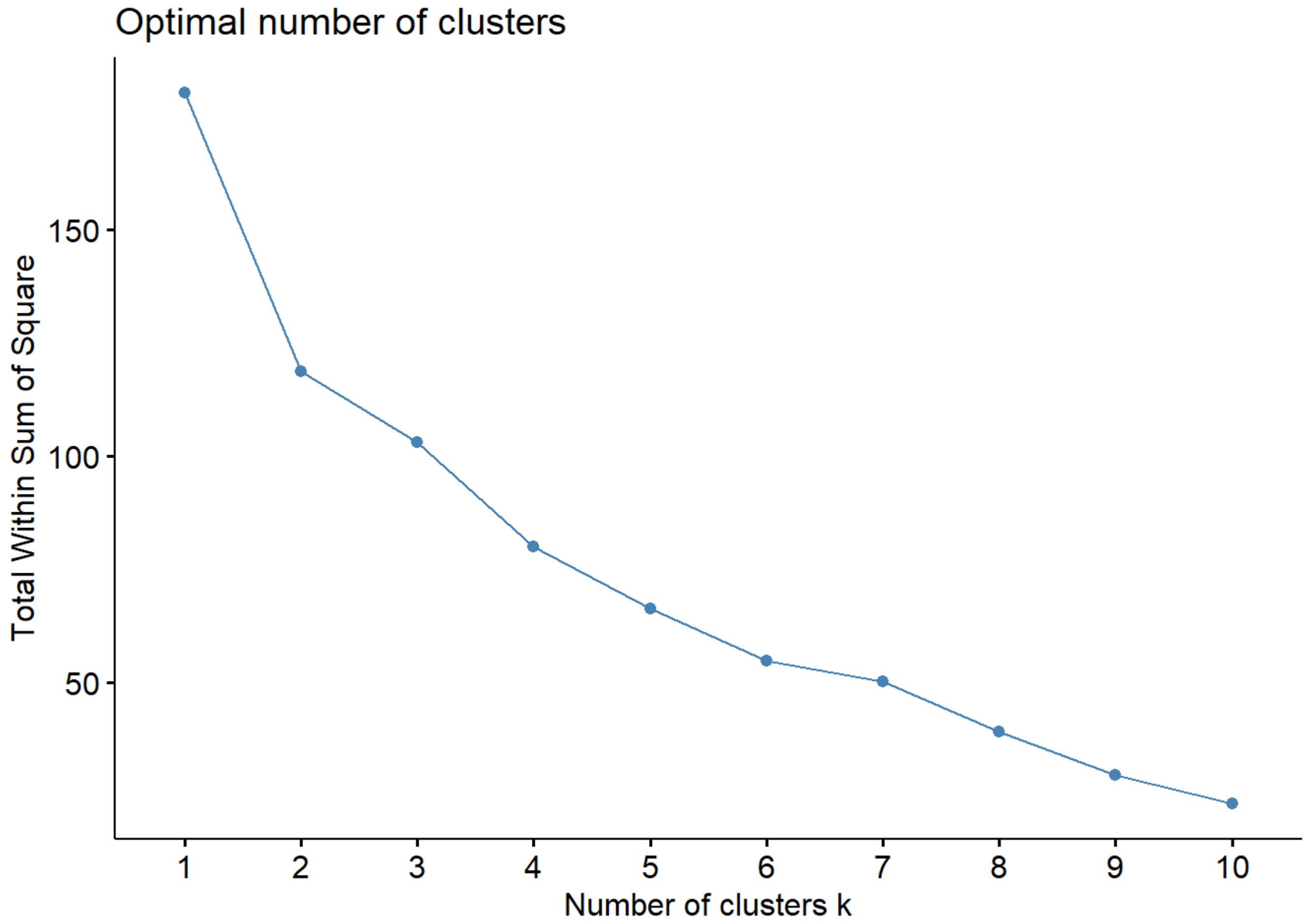
```
##  [1] 1 3 3 1 2 3 1 2 2 1 1 2 1 2 1 1 1 3 1 2 1
```

```
fviz_cluster(k3, pharma.data1)
```



```
# Interpretation : Using three clusters, the code applies k-means clustering to the standardi
zed pharmaceutical data. The information about the cluster centers, sizes, and assignments is
extracted and shown in the following lines. In order to examine the detected clusters in the
data graphically, a visualization is created at the end. The assumption or previous knowledge
that the data may be efficiently categorized into three different clusters is the basis for t
he selection of three clusters (k = 3).
```

```
fviz_nbclust(pharma.data1, kmeans, method = "wss")
```



```
# Interpretation : The plot usually looks like an elbow, and the "elbow" represents a good ca
ndidate for the optimal number of clusters. This is the point where the rate of decline in th
e within-cluster sum of squares slows down, indicating that performance decreases as the with
in-cluster variance decreases as you increase the number of clusters. Looking at this plot, y
ou can visually identify the point where adding more clusters improves the reduction of  with
in-cluster variability. Andquot;elbowquot; is often considered a reasonable choice for the op
timal number of clusters.
```

DBSCAN

```
library(dbscan)
d <- read.csv("C:/Users/Vishal/Downloads/Pharmaceuticals.csv")

# Interpretation : It may include performing density-based cluster analysis of pharmaceutical
data. DBSCAN is a clustering algorithm that groups  data points that are close to each other
in terms of density and is effective in identifying clusters of arbitrary shape.
```

```
data1 <- d[ ,-c(1,2,12,13,14)] # subsetting the data depend on the goals of the analysis
data1
```

```
##    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
```

```
# These all the characteristics of the dataset
```

```
set.seed(12)
db <- dbscan::dbscan(data1, eps = 25, MinPts = 2) #perform clustering
```

```
## Warning in dbscan::dbscan(data1, eps = 25, MinPts = 2): converting argument
## MinPts (fpc) to minPts (dbscan)!
```

```
print(db) #print cluster details
```

```
## DBSCAN clustering for 21 objects.
## Parameters: eps = 25, minPts = 2
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 2 cluster(s) and 7 noise points.
##
## 0 1 2
## 7 7 7
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

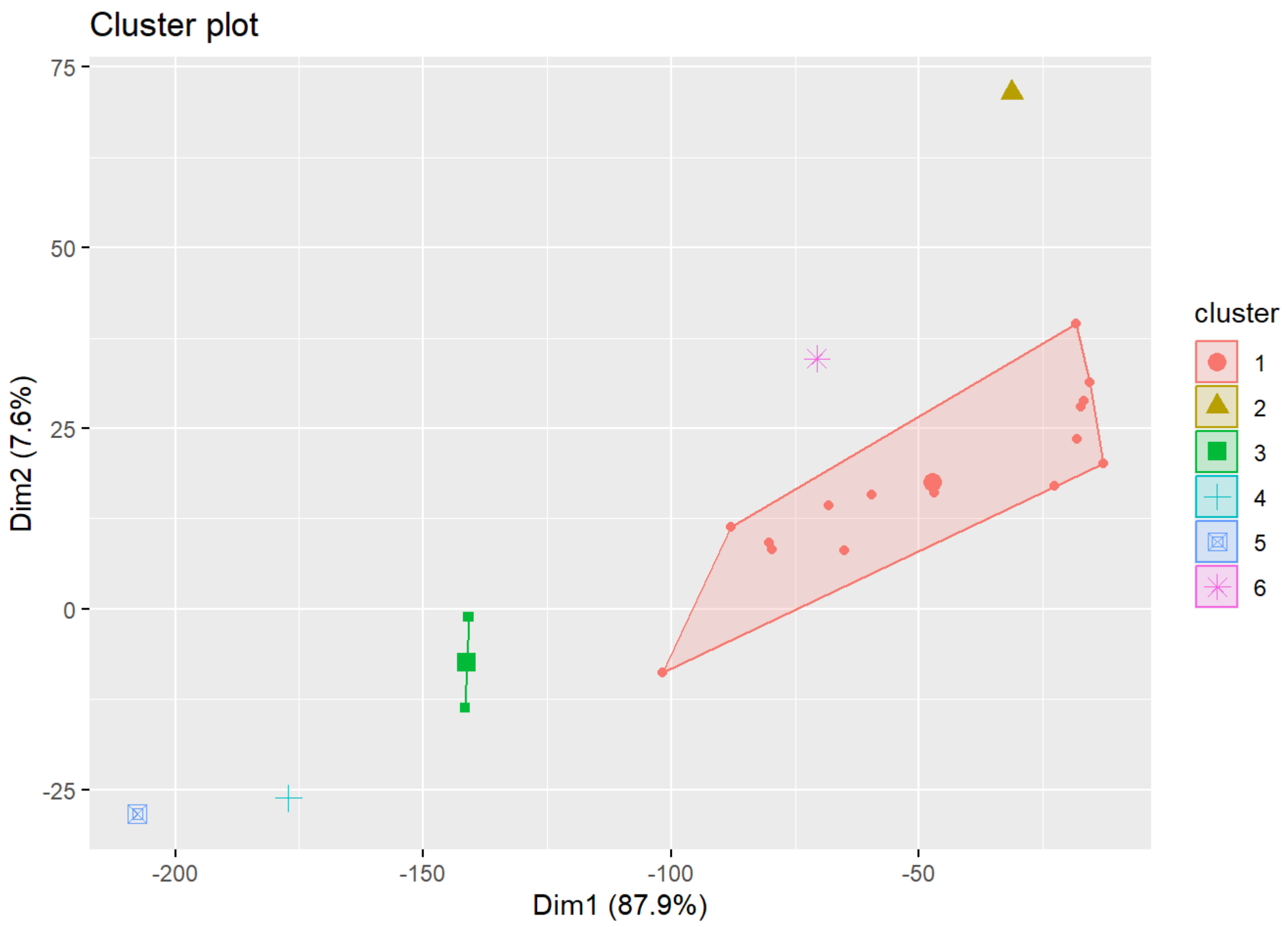
```
#install.packages("fpc")
library('factoextra')
library('fpc')
df <- data1[, 1:9]

set.seed(123)
db <- fpc::dbscan(data1, eps = 35, MinPts = 1) # DBSCAN using fpc package

print(db) # show clusters' details
```

```
## dbscan Pts=21 MinPts=1 eps=35
##        1 2 3 4 5 6
## seed  15 1 2 1 1 1
## total 15 1 2 1 1 1
```

```
# To identify dense regions or clusters within the data, and the specific parameters (eps and
MinPts) are set based on the characteristics of the data or the analysis goals. The factoextr
a library can later be used to visualize clustering results. The print(db) statement is used
to view details of the clusters detected by the algorithm, such as the number of clusters and
the points assigned to each cluster. This step helps to understand the results of cluster ana
lysis.
```

```
fviz_cluster(db, data1,   stand = FALSE, frame = FALSE, geom = "point") # Alternative way to
depict plot
```

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



```
# Visualizing clusters can be beneficial for interpreting the results of a clustering algorit
hm like DBSCAN. It helps to understand the spatial distribution of data points within cluster
s and assess the effectiveness of the clustering algorithm in grouping similar data points to
gether.
```

#Let do Hierarchical Clustering

```
A <- read.csv("C:/Users/Vishal/Downloads/Pharmaceuticals.csv")
Sorted.data <- A[ ,-c(1,2,12,13,14)]
```

Let Compute Euclidean distance

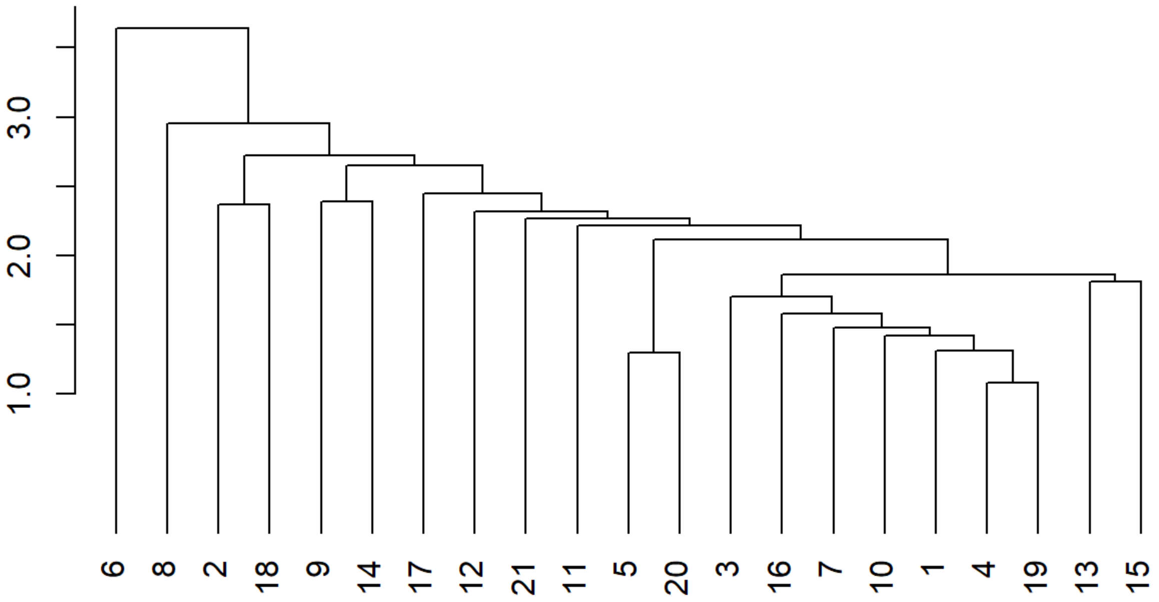
```
# (to compute other distance measures, change the value in method = )
d <- dist(Sorted.data, method = "euclidean")
d.norm <- dist(Sorted.data[,c(4,8)], method = "euclidean")

#  Euclidean distance is often used in clustering algorithms to measure the dissimilarity bet
ween data points. It forms the basis for many clustering methods, including hierarchical clus
tering and k-means clustering.
```

Table 15.4

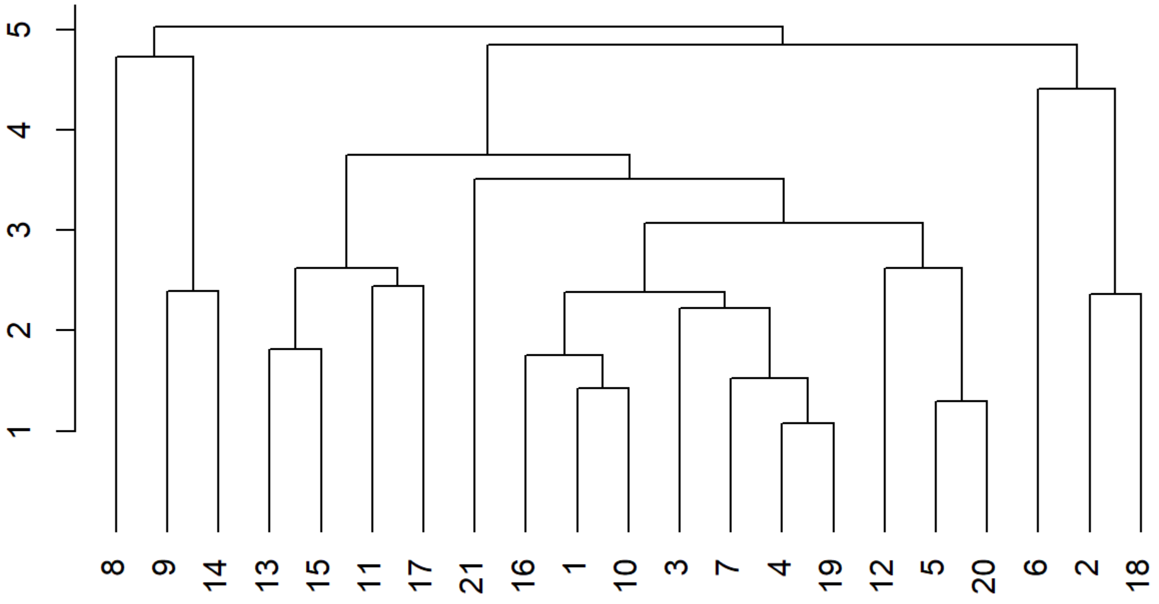
```
# normalize input variables
filt.data <- sapply(Sorted.data, scale)

# add row names: utilities
row.names(filt.data) <- row.names(Sorted.data)

# compute normalized distance based on variables ROE and Rev_Growth
d.norm <- dist(filt.data[,c(4,8)], method = "euclidean")

# The plot usually has an elbow-like shape, and the "elbow point" is a good place to look for
the ideal number of clusters. This is the point at which the within-cluster sum of squares de
clines more slowly, indicating that the number of clusters increases with diminishing returns
in terms of reducing the variance within clusters.This plot makes it easy to see when the imp
rovement in reducing within-cluster variability decreases with the number of clusters added.
Many people believe that the "elbow" number of clusters corresponds to a reasonable number of
clusters, which is the ideal number.
```

Figure 15.3

```
# compute normalized distance based on all 8 variables
d.norm <- dist(filt.data, method = "euclidean")

# in hclust() set argument method =
# to "ward.D", "single", "complete", "average", "median", or "centroid"
hc1 <- hclust(d.norm, method = "single")
plot(hc1, hang = -1, ann = FALSE)
```



```
hc2 <- hclust(d.norm, method = "average")
plot(hc2, hang = -1, ann = FALSE)
```



```
# Interpretation : Data is arranged using hierarchical clustering into a hierarchy of nested
clusters, and the formation of clusters is influenced by the linkage method selected. Various
linkage techniques capture different facets of the data structure. The dendrogram that result
s from employing the plot function to visualize the hierarchical clustering structures illust
rates how the observations are arranged into clusters according to varying degrees of similar
ity. It is common practice to investigate various linkage methods in order to obtain insights
into the underlying clustering patterns of the data, as the choice of linkage method can impa
ct the shape and structure of the resulting dendrogram.
```

Table 15.6

```
memb <- cutree(hc1, k = 3)
memb
```

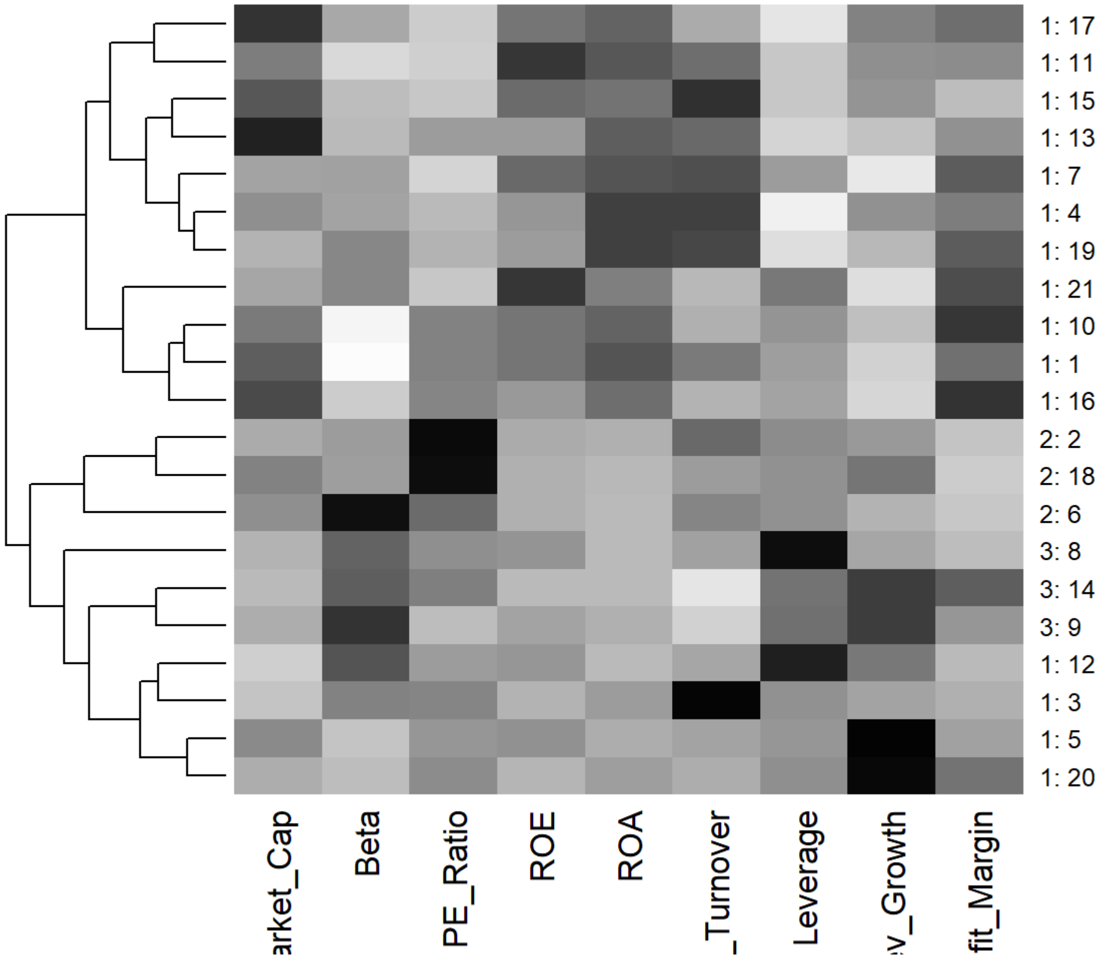
```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##  1  1  1  1  1  2  1  3  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
memb <- cutree(hc2, k = 3)
memb
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##  1  2  1  1  1  2  1  3  3  1  1  1  1  3  1  1  1  2  1  1  1
```

Figure 15.4

```
# set labels as cluster membership and utility name
row.names(filt.data) <- paste(memb, ": ", row.names(Sorted.data), sep = "")

# plot heatmap
# rev() reverses the color mapping to large = dark
heatmap(as.matrix(filt.data), Colv = NA, hclustfun = hclust,
        col=rev(paste("grey",1:99,sep="")))
```



```
# Interpretation : In order to highlight the patterns and connections both within and between
clusters, the code attempts to graphically depict the data's structure. This can be especiall
y helpful for preliminary data analysis and learning more about the traits of various cluster
s.
```