Clustering

Pardeepan

2023-11-07

Summary:

In my observation, the provided statement describes a situation in which a financial analyst for stocks is reviewing data from 21 pharmaceutical companies. The goal is to use numerical variable cluster analysis to understand 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. Because each explanation is mentioned on the codes.

Problem:

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:

1. Market capitalization (in billions of dollars)
2. Beta
3. Price/earnings ratio
4. Return on equity
5. Return on assets
6. Asset turnover
7. Leverage
8. Estimated revenue growth
9. Net profit margin
10. Median recommendation (across major brokerages)
11. Location of firm's headquarters
12. Stock exchange on which the firm is listed

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

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.

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)

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

Statement:

The equities analyst tasked with researching the pharmaceutical industry seeks assistance in exploring and comprehending the financial data collected on 21 pharmaceutical firms. The goal is to gain insights into the industry's structure by using fundamental financial measures. The dataset, Pharmaceuticals.csv, includes variables such as market capitalization, beta, price/earnings ratio, return on equity, return on assets, asset turnover, leverage, estimated revenue growth, net profit margin, median recommendation, location of the firm's headquarters, and stock exchange on which the firm is listed. To classify the 21 pharmaceutical firms, the analyst intends to use cluster analysis, focusing on numerical variables (1–9). This necessitates justifying decisions made during the cluster analysis, such as the weighting of various variables, the selection of clustering algorithm(s), the number of clusters, and other pertinent considerations. The analysis should not only form clusters but also interpret them in terms of the numerical variables used to create them. Furthermore, the equities analyst wants to know if there are any discernible patterns in the clusters regarding the variables that were not used in the clustering process (variables 10 to 12). As part of the analysis, the equities analyst must assign appropriate names to each cluster using any or all of the variables in the dataset. This will contribute to a comprehensive understanding of the pharmaceutical industry's financial landscape and assist in making informed investment decisions.

```
# Calling Required Libraries

library(class)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(e1071)
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 become 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(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(fpc)
```

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

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

```
# Calling the csv file

pharma.data <-
read.csv("C:\\Users\\spard\\OneDrive\\Desktop\\FML\\Assignment_4\\Pharmaceuti
cals.csv")
dim(pharma.data)
```

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

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

```
##       [,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 : One can determine the number of observations (rows) and variables (columns) by looking at the data frame's dimensions. To display the column names in a different format or orientation, you can flip them.

 # Dropping thge columns that are not required for clustering

```
pharma.data <- pharma.data[ ,-c(1,2,12,13,14)]
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"
```

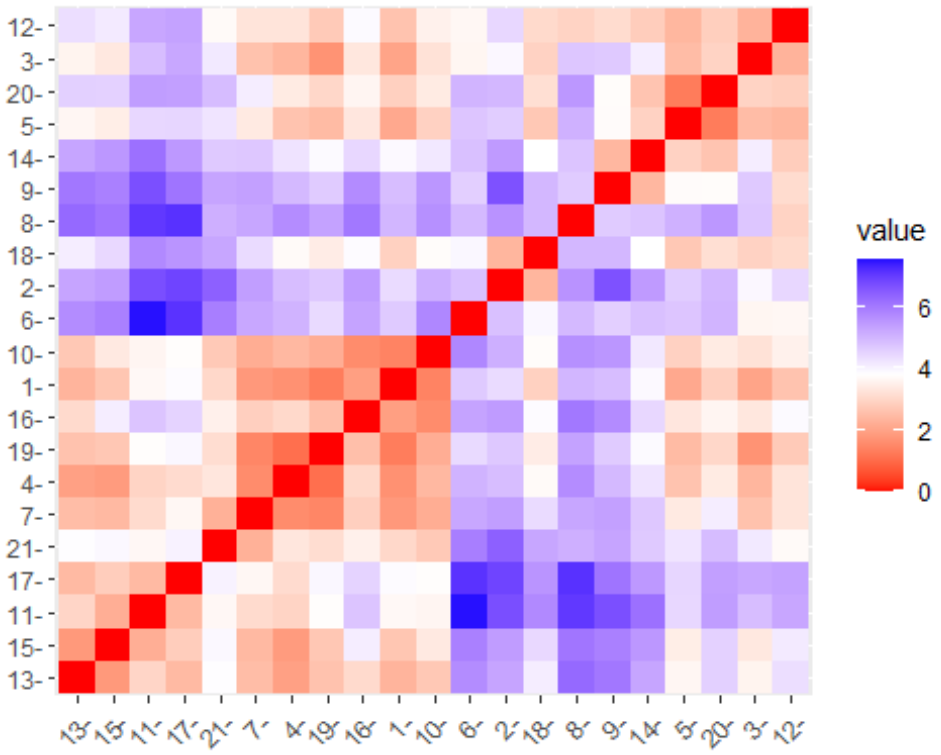
 # Interpretation : Columns with indexes 1, 2, 12, 13, and 14 may have been removed because the data in those columns is redundant, doesn't need to be analyzed, or isn't seem relevant to the task at hand. Calculating summary statistics yields a concise summary of the distribution and central tendency of the remaining variables in the modified data frame. Understanding the properties of the data can benefit from this.

 # Initiating with K means

```
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
```

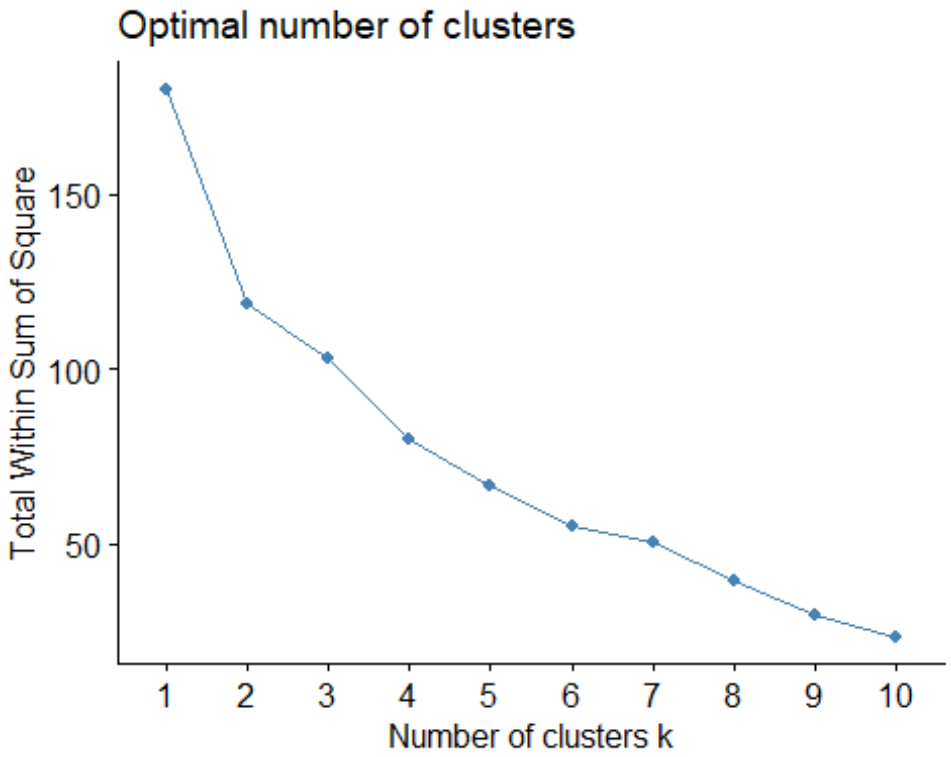
```
distance <- get_dist(pharma.data1)
fviz_dist(distance)
```




```
# Interpretation : The scale function is used to standardize the columns in
the pharmacy.an array of data frames. Scaling by the standard deviation and
centering the variables by subtracting the mean are the two steps involved in
standardization. It is likely that the code is part of an exploratory study
or is a warm-up for a clustering task where the separations between data
points are important to know. The specific requirements of the research or
the algorithm being used to decide which distance and standardization
calculations are necessary.

# Defining 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)
```


```
# The code uses three clusters to apply k-means clustering to the
standardized pharmaceutical data. The information about cluster centers,
sizes, and assignments is extracted and shown in the lines that follow. At
the end, a visualization is created to allow for a visual inspection of the
found clusters in the data. Based on the assumption or prior knowledge that
the data can be effectively divided into three distinct clusters, three
clusters (k = 3) have been selected.

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




```
# Plots typically have an elbow-like shape, and the "elbow" stands for a
viable option for the ideal number of clusters. This is the point at which
the within-cluster sum of squares declines at a slower rate, meaning that as
the number of clusters increases, performance gets worse as the within-
cluster variance gets smaller. With the help of this plot, you can see where
increasing the number of clusters reduces within-cluster variability more
effectively. "Elbow" is frequently regarded as a suitable option when
determining the ideal number of clusters.

# Using dbscan library

library(dbscan)
d <-
read.csv("C:\\Users\\spard\\OneDrive\\Desktop\\FML\\Assignment_4\\Pharmaceuti
cals.csv")

data1 <- d[ ,-c(1,2,12,13,14)]
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
```

```
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)
```

```
## 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
```

```
library('factoextra')
library('fpc')
df <- data1[, 1:9]

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

print(db)
```

```
## 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
```

```
# In order to detect dense areas or groups within the data, the precise
parameters (eps and MinPts) are established according to the attributes of
the data or the objectives of the analysis. The clustering results can then
be visualized using the factoextra library. To view information about the
clusters identified by the algorithm, including the number of clusters and
the points allocated to each cluster, use the print(db) statement.
Understanding the cluster analysis results is aided by this step.

fviz_cluster(db, data1,   stand = FALSE, frame = FALSE, geom = "point")
```

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


```
# It can be useful to visualize clusters when interpreting the output of a clustering algorithm such as DBSCAN. Analyzing the spatial distribution of data points within clusters and evaluating how well the clustering algorithm groups together similar data points are helpful.
```

#Hierarchical

```
A <-
read.csv("C:\\Users\\spard\\OneDrive\\Desktop\\FML\\Assignment_4\\Pharmaceuticals.csv")
Sorted.data <- A[ ,-c(1,2,12,13,14)]
```

Compute Euclidean distance

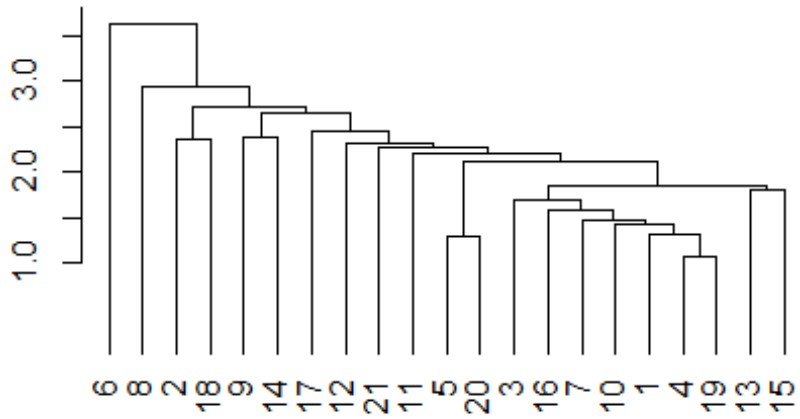
```
d <- dist(Sorted.data, method = "euclidean")
d.norm <- dist(Sorted.data[,c(4,8)], method = "euclidean")
```

Table 15.4

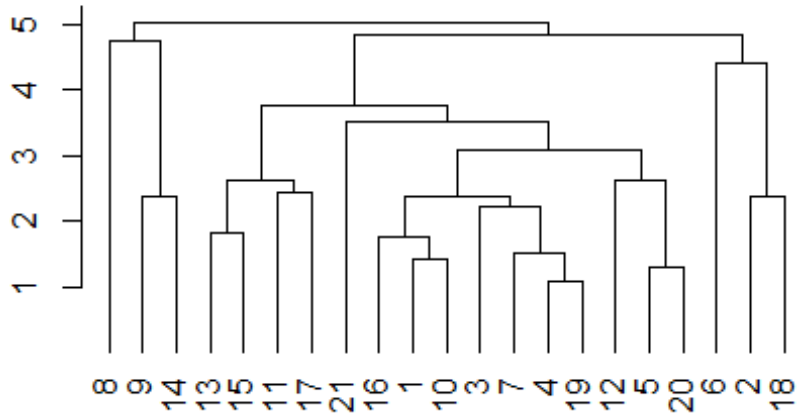
```
filt.data <- sapply(Sorted.data, scale)

row.names(filt.data) <- row.names(Sorted.data)


d.norm <- dist(filt.data[,c(4,8)], method = "euclidean")
```

Figure 15.3

```
d.norm <- dist(filt.data, method = "euclidean")

hc1 <- hclust(d.norm, method = "single")
plot(hc1, hang = -1, ann = FALSE)
```




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




```
# Hierarchical clustering is used to organize data into a hierarchy of nested
clusters; the choice of linkage method affects how the clusters form. The
data structure is captured in different ways by different linkage techniques.
When the hierarchical clustering structures are visualized using the plot
function, the dendrogram that emerges shows how the observations are
organized into clusters based on different levels of similarity. Since the
choice of linkage method can affect the shape and structure of the resulting
dendrogram, it is common practice to investigate various linkage methods to
gain insights into the underlying clustering patterns of the data.
```

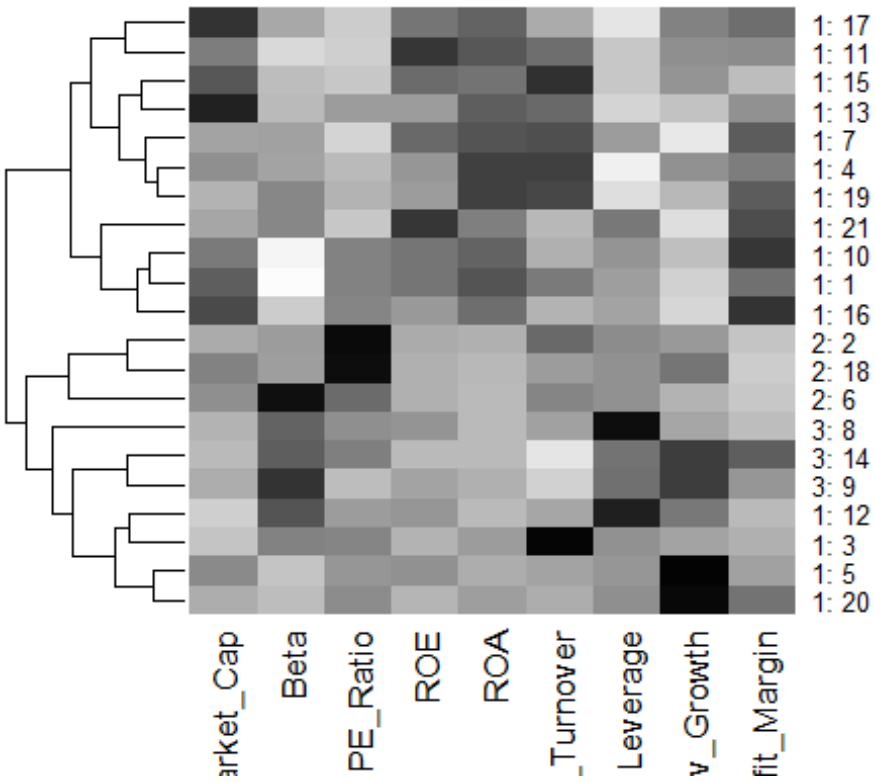
Table 15.6

```
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

```
# setting labels as cluster membership and utility name

row.names(filt.data) <- paste(memb, ": ", row.names(Sorted.data), sep = "")
```

```r
# plot heatmap
# rev() reverses the color mapping to large = dark

heatmap(as.matrix(filt.data), Colv = NA, hclustfun = hclust,
        col=rev(paste("gray",1:99,sep="")))
```




```r
# The code tries to graphically represent the structure of the data in order
to highlight the connections and patterns both within and between clusters.
This can be particularly useful for learning more about the characteristics
of different clusters and for preliminary data analysis.
```