Studying the Stock Market via Unsupervised Machine Learning

How Well Do Sectors Align With Clusters?

Mehmet Aydın

December 29, 2024

This paper utilizes k-means clustering to cluster stocks in the Istanbul Stock Exchange based on certain financial indicators, principal component analysis to graph the data points and clusters with as much variance as possible in two dimensions. In the end, a confusion matrix is created to calculate the accuracy of the cluster prediction given sectors.

Table of contents

Introduction and Questions

Stock markets are comprised of multiple companies that operate in many different sectors. These different sectors may have financial dynamics unique to themselves that separate them from other sectors, or they may not, in which case the dynamics may be characteristics of different levels (e.g., geographic, country-wide, company-specific etc.). This report aims to measure how much of a similarity there is between the financial dynamics of companies grouped by their sectors and the clusters created through machine learning algorithms based on these financial data.

Data

Source

All the data subject to this study has been sourced from BIST DataStore, the official data portal of the Istanbul Stock Exchange. The data used in this study can be reached by the tab "Equity Market Data". The data frames registered as starting with "*price_*" can be reached by navigating to "Equity Based Data" and then to "Prices (Close, Minimum, and Maximum) and Traded Volume, Traded Value (monthly)". Those registered as starting with "*data_*", on the other hand, can be reached by navigating to "Company Data" and then to "Basic Ratios (monthly)". The portal requires an online membership free of charge to access the data. Due to the very complex nature of the spreadsheet provided by the Istanbul Stock Exchange, the file containing sector information of companies (*sectors202409.xlsx*) had to be created manually from the file containing data for September 2024 (*oran202409.xlsx*). For the sectors the sector supra-title, one written in bold in the original file, was used instead of the sector-proper, as the latter resulted in the need to conduct cluster analysis with 38 clusters, which hurt the interpretability of the data. The rest of the files have been edited solely using R.

Tidying the Data

```
# Loading the required libraries
library(tidyverse)
library(readxl)
library(broom)
library(gridExtra)
library(GGally)
library(ggthemes)
library(caret)
library(factoextra)
```

```r
# Loading the data sets
price_june <- read_excel("PP_AYLIKOZET.M.202406.xlsx", skip = 2)
price_july <- read_excel("PP_AYLIKOZET.M.202407.xlsx", skip = 2)
price_august <- read_excel("PP_AYLIKOZET.M.202408.xlsx", skip = 2)
price_september <- read_excel("PP_AYLIKOZET.M.202409.xlsx", skip = 2)
data_june <- read_excel("oran202406.xlsx", skip = 5)
data_september <- read_excel("oran202409.xlsx", skip = 5)
sectors <- read_excel("sectors202409.xlsx")

# Tidying up the data frames and extracting variables of interest
price_june <- price_june |>
  select("İşlem Kodu / Instrument Code",
         "Kapanış Fiyatı / Closing Price") |>
  separate_wider_delim("İşlem Kodu / Instrument Code",
                       delim = ".",
                       names = c("company", "type")) |>
  filter(type == "E") |>
  select(!type)
names(price_june) <- c("company", "price")

price_july <- price_july |>
  select("İşlem Kodu / Instrument Code",
         "Toplam İşlem Hacmi (TL) / Total Traded Value (TL)") |>
  separate_wider_delim("İşlem Kodu / Instrument Code",
                       delim = ".",
                       names = c("company", "type")) |>
  filter(type == "E") |>
  select(!type)
names(price_july) <- c("company", "total_traded_value")

price_august <- price_august |>
  select("İşlem Kodu / Instrument Code",
         "Toplam İşlem Hacmi (TL) / Total Traded Value (TL)") |>
  separate_wider_delim("İşlem Kodu / Instrument Code",
                       delim = ".",
                       names = c("company", "type")) |>
  filter(type == "E") |>
  select(!type)
names(price_august) <- c("company", "total_traded_value")

price_september <- price_september |>
  select("İşlem Kodu / Instrument Code",
```

```
         "Toplam İşlem Hacmi (TL) / Total Traded Value (TL)",
         "Kapanış Fiyatı / Closing Price") |>
  separate_wider_delim("İşlem Kodu / Instrument Code",
                       delim = ".",
                       names = c("company", "type")) |>
  filter(type == "E") |>
  select(!type)
names(price_september) <- c("company", "total_traded_value", "price")

data_june <- data_june |>
  select("...1",  "NET KAR\r\n(SON 4 ÇEYREK)", "F/K", "PD/DD") |>
  na.omit()
names(data_june) <- c("company",
                      "net_profit",
                      "price_to_earnings",
                      "market_to_book_value")

data_september <- data_september |>
  select("...1",  "NET KAR\r\n(SON 4 ÇEYREK)", "F/K", "PD/DD") |>
  na.omit()
names(data_september) <- c("company",
                           "net_profit",
                           "price_to_earnings",
                           "market_to_book_value")

# Creating a data frame with all the relevant data
data_main <- inner_join(price_june,
                        price_september,
                        by = "company",
                        suffix = c("_start", "_end")) |>
  inner_join(price_july, by = "company") |>
  inner_join(price_august, by = "company") |>
  inner_join(data_june, by = "company") |>
  inner_join(data_september, by = "company", suffix = c("_start", "_end")) |>
  inner_join(sectors, by = "company")

# Transforming non-numeric variables to numeric and omitting missing values
data_main <- data_main |>
  mutate_at(c("price_start",
              "total_traded_value.x",
              "price_end",
              "total_traded_value.y",
```

```
              "total_traded_value",
              "net_profit_start",
              "price_to_earnings_start",
              "market_to_book_value_start",
              "net_profit_end",
              "price_to_earnings_end",
              "market_to_book_value_end"),
            as.numeric) |>
  na.omit()

# Summing up total traded values and removing the monthly ones
data_main <- data_main |>
  mutate(total_traded_value = total_traded_value +
           total_traded_value.x +
           total_traded_value.y) |>
  select(!c(total_traded_value.x, total_traded_value.y))
```

The data found inside the main data frame was then used to construct a new data frame of which included the variables of interest to the study. The variables except *company*, *sector* and *total_traded_value* are all recorded as the relative changes in the respective financial factors between the ends of June and September 2024 (2024 Q3).

```
# Creating the data frame of relative changes and variables of interest
data_interest <- data_main |>
  mutate(price =
           (price_end - price_start) /
           price_start) |>
  mutate(net_profit =
           (net_profit_end - net_profit_start) /
           net_profit_start) |>
  mutate(market_to_book_value =
           (market_to_book_value_end - market_to_book_value_start) /
           market_to_book_value_start) |>
  mutate(price_to_earnings =
           (price_to_earnings_end - price_to_earnings_start) /
           price_to_earnings_start) |>
  mutate(sector = as.factor(sector_name)) |>
  mutate(sector = unclass(sector)) |>
  select(company,
         sector_name,
         sector,
         price,
```

```
         total_traded_value,
         net_profit,
         market_to_book_value,
         price_to_earnings)

# Creating a data frame of sector names by number
sector_names <- data_interest |>
  select(sector_name, sector) |>
  unique()

# Removing sector names from the data frame of variables of interest
data_interest <- data_interest |>
  select(!sector_name)
```

Below, one may find a table of the variables of interest to the study that *data_interest* contains.

Variable	Type	Meaning
company	chr	Company code in the Istanbul Stock Exchange.
sector	int	The number assigned to the company's sector.
price	dbl	Change in price of the stock between the start and end of the quarter.
total_traded_value	dbl	Total value that the stock has been traded for during the quarter.
net_profit	dbl	Change in net profit of the company between the start and end of the quarter.
market_to_book_value	dbl	Change in market to book value ratio of the company between the start and end of the quarter.
price_to_earnings	dbl	Change in price to earnings ratio of the company between the start and end of the quarter.

Below, one may find a table of the sector numbers and the sectors themselves.

Sector Number (*sector*)	Sector Name (*sector_name*) in English
1	Information and Communication
2	Education, Health, Sports and Entertainment Services
3	Electricity, Gas and Water
4	Real Estate Activities
5	Administrative and Support Service Activities
6	Manufacturing Industry
7	Construction and Public Works
8	Mining and Quarrying
9	Financial Institutions
10	Professional, Scientific and Technical Activities
11	Hotels and Restaurants
12	Pre-Market Trading Platform
13	Agriculture, Forestry and Fishing
14	Technology
15	Wholesale and Retail Trade
16	Transportation and Storage

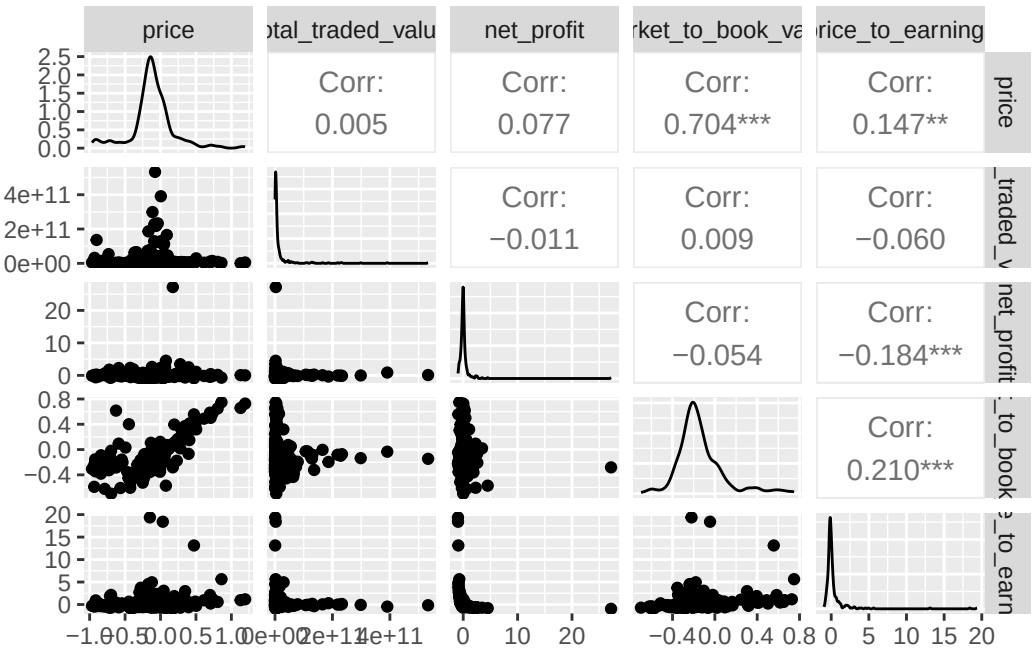
Analysis

In this report, the tools used are unsupervised machine learning tools, namely principal component analysis and k-means clustering. They will be used to cluster the data into as many cluster as there are sectors and to graph them efficiently. Then, a confusion matrix will be calculated for the analysis which will show the accuracy of clustering vis-à-vis the sectors.

Pairwise Correlation Plot

```
# Creating a data frame with quantitative variables
data_features <- data_interest |>
  select(!c(company, sector))

# Plotting the pairwise correlation plot
data_features |>
  ggpairs()
```



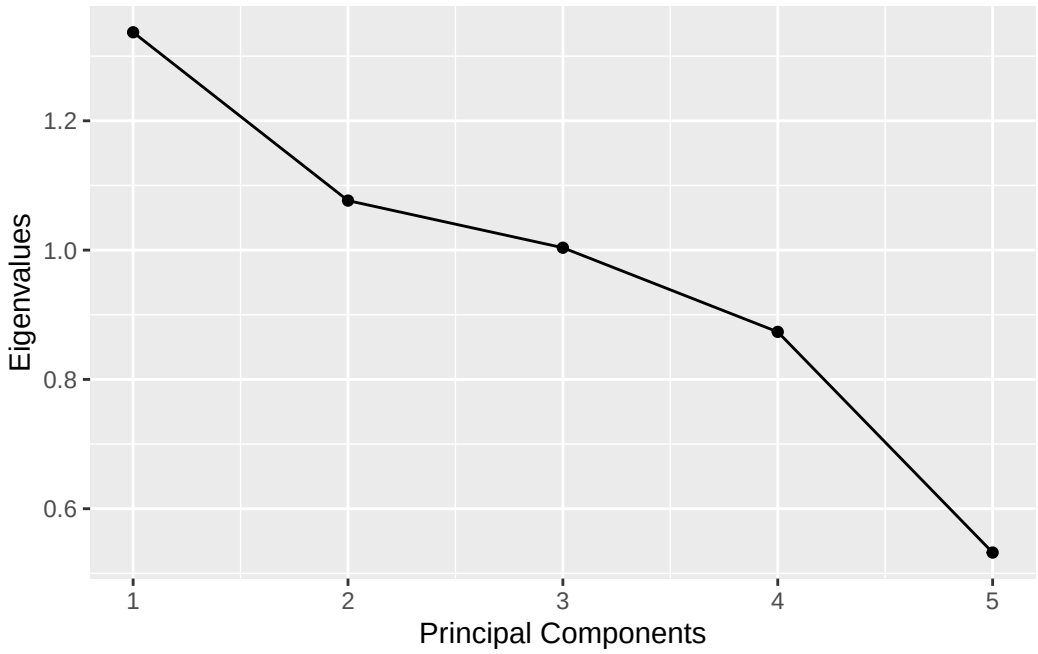
As one can see, the variables are not significantly correlated, with the exception of those that are imperfect linear transformations of one another.

Principal Component Analysis

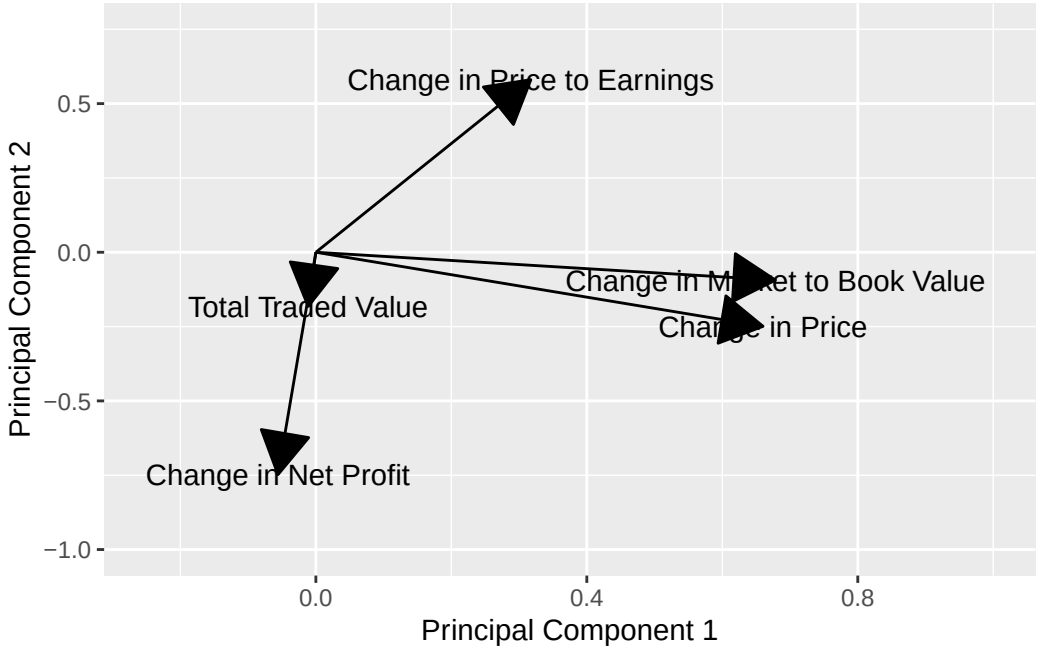
To be able to show as much variance as possible while plotting the data on two dimensions, the data is subjected to principal component analysis.

```
# Calculating the principal components
pca <- data_features |>
  prcomp(center = T, scale = T)

# Scree plot to evaluate how much each PC contributes
pca |>
  tidy(matrix = "eigenvalues") |>
  ggplot(aes(x = PC, y = std.dev)) +
  geom_point() +
  geom_line() +
  labs(x = "Principal Components", y = "Eigenvalues")
```



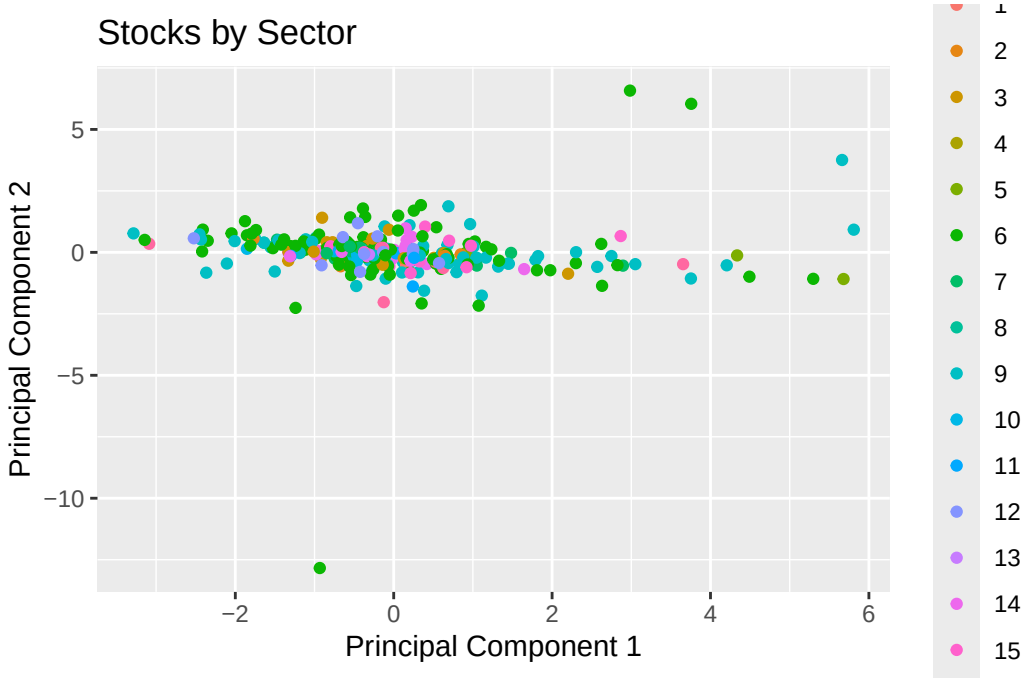

```
# Seeing how original variables contribute to the PCs
pca |>
  tidy(matrix = "rotation") |>
  pivot_wider(names_from = "PC",
              names_prefix = "PC",
              values_from = "value") |>
  ggplot(aes(PC1, PC2)) +
  geom_segment(xend = 0,
               yend = 0,
               arrow = arrow(ends = "first", type = "closed")) +
  geom_text(aes(label = case_when(
    column == "price_to_earnings" ~ "Change in Price to Earnings",
    column == "market_to_book_value" ~ "Change in Market to Book Value",
    column == "price" ~ "Change in Price",
    column == "net_profit" ~ "Change in Net Profit",
    column == "total_traded_value" ~ "Total Traded Value"
  ))) +
  labs(x = "Principal Component 1", y = "Principal Component 2") +
  lims(x = c(-0.25, 1), y = c(-1, 0.75))
```



One may see that the 1st principal component is very close to the change in market to book value and that in price. As book value is quite stable, one can take this axis to be very close to the change in market value. The 2nd principal component, however, is quite more complex.

```
# Augmenting and displaying data
pca_augmented <- pca |>
  augment(data_interest)

pca_augmented |>
  ggplot(aes(x = .fittedPC1, y = .fittedPC2, color = as.factor(sector))) +
  geom_point() +
  labs(title = "Stocks by Sector",
       x = "Principal Component 1",
       y = "Principal Component 2",
       color = "Sector")
```

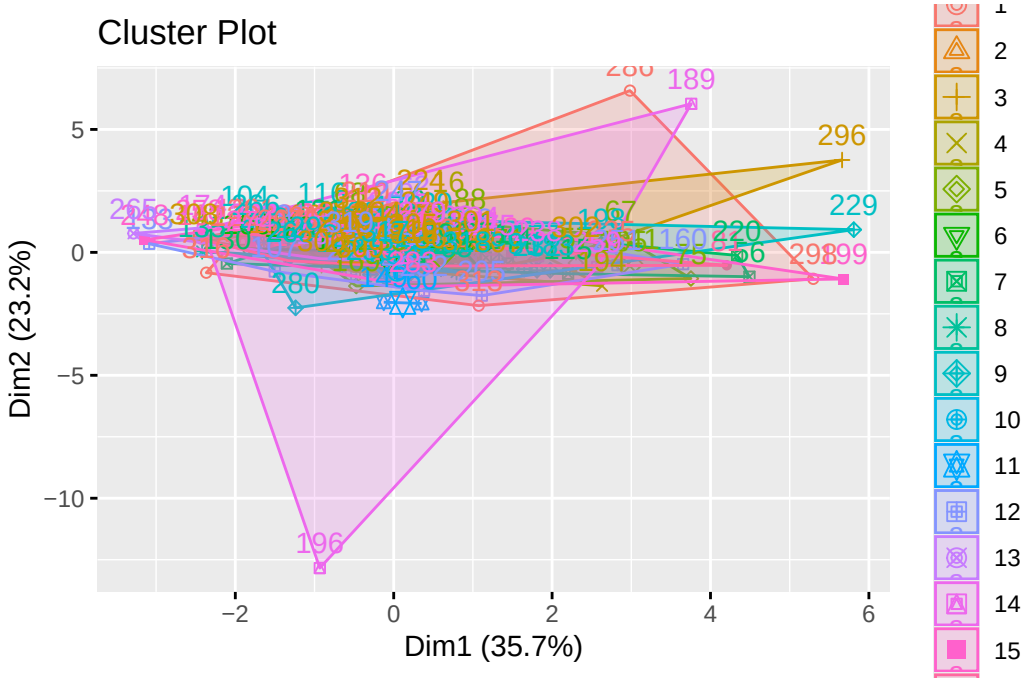


One may observe with the help of the plot above that stocks of companies that belong to different sectors have different characteristics, such as the existence of outliers on the top-right corner of sectors 6-8 and the concentration around the origin of sector 12.

K-means clustering

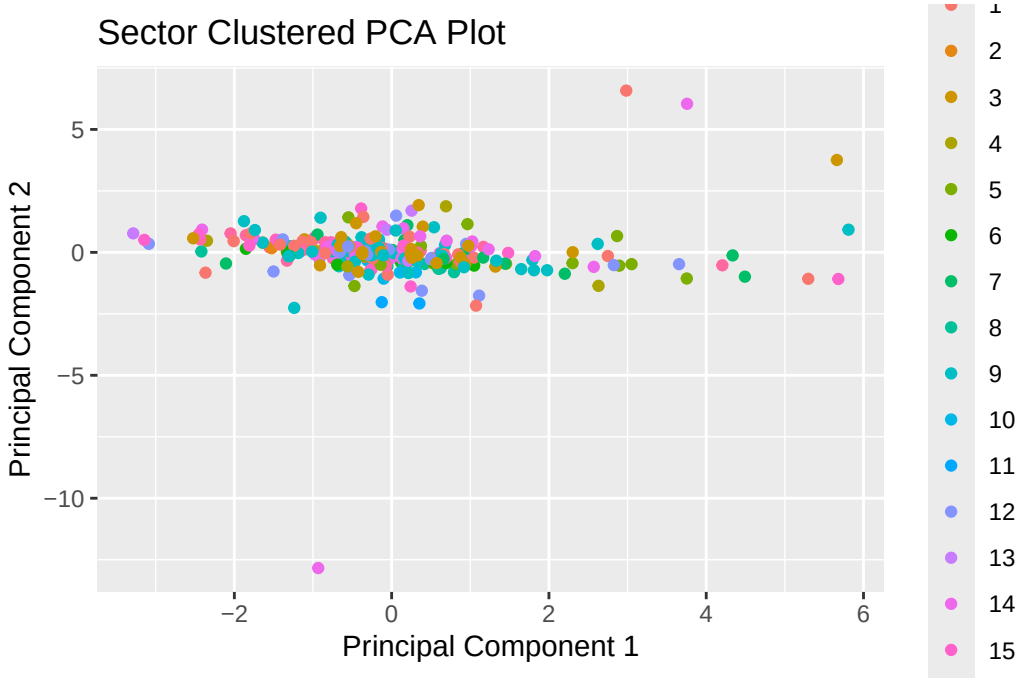
```
# Clustering
set.seed(31)

kmeans_base <- kmeans(data_features, centers = 16)
fviz_cluster(kmeans_base, data = data_features) + labs(title = "Cluster Plot")
```




```
pca_augmented <- pca_augmented |>
  mutate(sector = as.factor(sector)) |>
  mutate(cluster = as.factor(kmeans_base$cluster))

# Plotting clusters on PCs
pca_augmented |>
  ggplot(aes(x = .fittedPC1, y = .fittedPC2, color = cluster)) +
  geom_point() +
  labs(title = "Sector Clustered PCA Plot",
       x = "Principal Component 1",
       y = "Principal Component 2",
       color = "Cluster")
```



Unfortunately, the graph is not very informative regarding the clusters. However, one can see a few patterns, such as how cluster 3 is focused around the origin and how clusters 8 and 9 are scattered closer to the x-axis than other clusters.

Confusion Matrix and Accuracy

```
# Evaluating the accuracy of clustering
confusionMatrix(pca_augmented$sector, pca_augmented$cluster)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
        1   0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0
        2   1  0  0  0  0  0  0  0  2  0  0  0  1  1  0  1
        3   2  0  0  0  2  1  3  0  4  0  0  2  3  1  1  3
        4   0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
        5   0  0  0  0  0  0  2  0  0  0  0  1  0  0  1  0
        6  15  5  5  7  8  2  6  2 22  1  1 14  3  9  8 11
        7   1  0  0  0  1  1  0  0  1  0  0  1  0  0  2  2
        8   0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
```

```
         9   9  1 10  2 11  3  7  1 12  4  0  6  4  8  8  5
         10  0  0  0  1  0  1  0  0  1  0  0  0  0  1  0  0
         11  0  0  1  0  0  0  1  0  2  0  0  0  0  1  1  0
         12  0  0 11  0  0  0  0  0  0  0  0  0  0  0  0  0
         13  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
         14  1  1  0  0  1  0  2  1  2  0  0  1  1  5  3  1
         15  1  0  3  2  1  1  0  1  4  0  0  0  1  2  0  0
         16  0  0  0  0  0  0  1  1  1  0  1  2  0  0  0  1

Overall Statistics

               Accuracy : 0.0627
                 95% CI : (0.0387, 0.0952)
    No Information Rate : 0.1599
    P-Value [Acc > NIR] : 1

                  Kappa : -0.0211

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
Sensitivity           0.00000  0.00000  0.00000 0.000000  0.00000  0.20000
Specificity           0.99308  0.98077  0.92388 0.996732  0.98639  0.62136
Pos Pred Value        0.00000  0.00000  0.00000 0.000000  0.00000  0.01681
Neg Pred Value        0.90536  0.97764  0.89899 0.959119  0.92063  0.96000
Prevalence            0.09404  0.02194  0.09404 0.040752  0.07837  0.03135
Detection Rate        0.00000  0.00000  0.00000 0.000000  0.00000  0.00627
Detection Prevalence  0.00627  0.01881  0.06897 0.003135  0.01254  0.37304
Balanced Accuracy     0.49654  0.49038  0.46194 0.498366  0.49320  0.41068
                     Class: 7 Class: 8 Class: 9 Class: 10 Class: 11 Class: 12
Sensitivity           0.00000 0.000000  0.23529   0.00000   0.00000   0.00000
Specificity           0.96970 0.996795  0.70522   0.98726   0.98107   0.96220
Pos Pred Value        0.00000 0.000000  0.13187   0.00000   0.00000   0.00000
Neg Pred Value        0.92903 0.977987  0.82895   0.98413   0.99361   0.90909
Prevalence            0.06897 0.021944  0.15987   0.01567   0.00627   0.08777
Detection Rate        0.00000 0.000000  0.03762   0.00000   0.00000   0.00000
Detection Prevalence  0.02821 0.003135  0.28527   0.01254   0.01881   0.03448
Balanced Accuracy     0.48485 0.498397  0.47026   0.49363   0.49054   0.48110
                     Class: 13 Class: 14 Class: 15 Class: 16
Sensitivity           0.000000   0.17857   0.00000  0.041667
Specificity           0.996732   0.95189   0.94576  0.979661
```

Pos Pred Value	0.000000	0.26316	0.00000	0.142857
Neg Pred Value	0.959119	0.92333	0.92079	0.926282
Prevalence	0.040752	0.08777	0.07524	0.075235
Detection Rate	0.000000	0.01567	0.00000	0.003135
Detection Prevalence	0.003135	0.05956	0.05016	0.021944
Balanced Accuracy	0.498366	0.56523	0.47288	0.510664

As one may see, the accuracy achieved through plotting is only 6.27%.

Conclusion

Given the low accuracy rate, this report shows that a cluster analysis made using the quarterly change in financial indicators and total traded value found in the Istanbul Stock Exchange records. This in turn indicates that the quarterly changes in these financial indicators and total traded value are not sector-specific. This report gives researchers interested in finding variables with which they may cluster sectors accurately a starting point regarding which variables they should de-prioritize. However, one must keep in mind that this data concerns only the 3rd quarter of 2024, and the same study structure conducted using data from a different period is likely to return a different result.