Clustering Stock by Sectors

Final project

By

Joey CODY & Morgan GEFFROY

In this project, we will use Data Mining techniques to prove that sector provide good clustering of the NASDAQ stocks and that it can give us information on the structure of this economic sectors. Moreover, we will demonstrate that the Beta from the CAPM theory is a good indicator of a stock behaviour.

PSTAT 131

Professor R. Feldman

Teaching Assistant: Sergio Rodriguez

I. Introduction

a. Problem

We want to use Data Mining techniques implemented PSTAT 131 and apply them to financial management. We are primarily interested in financial issues and financial analysis.

The most important problem in finance, especially for beginners, is to be able to find information from the tons of data available, and to be able to make a decisions and predictions from this information. Data mining techniques are highly coveted for this kind of financial analysis.

One problem that faces financial managers when managing a portfolio is the constitution of this portfolio and which assets to buy or sell. According to economic theory, the more different and heterogeneous the stocks contained in a portfolio are the less risky this portfolio is. Thus, being able to recognize the difference between the assets and to be able to pick a very heterogeneous group of stocks is a motivation of every financial manager.

- **How Data Mining Techniques could help to differentiate several clusters of assets?**
- **How relevant the classification by sector or/and by industry is it?**

b. Techniques uses

Since our goal was to produce subsets of similar stocks within our dataset of the NASDAQ stocks, we used mostly classification techniques. We try different approaches: hierarchical clustering, partitional clustering and finally density-based clustering.

c. Results

We find that different clustering Data Mining techniques show that stocks of companies within the same sector do behave similarly and that sectors represents a good way to clustering stocks. Moreover, we find that the more likely stocks in of

company in the sectors are to form a cluster, the more homogeneous this sector. Then, Data Mining helps us to understand the structure of the different sector of the economy. Finally, we find that the Beta from the economics theory is a good prediction of the behavior of a stock.

d. Conclusions

The Data Mining techniques could, even if they assume any distribution of those variables help to extract information from continuous variables as the historical prices of stocks.

There exist some patterns in the price of the stock that depends on the sector of the company.

II. Sections

a. Getting the Dataset

Our principal concern before starting the project was to find data an available and ready data set. Since having information implies a strategic and pecuniary benefit, we were afraid not to have quality data readily available.

i. Choosing the data:

After deciding to orientate the final project towards classifying different assets, we began the process of choosing and obtaining data.

1. Problem

Our goals are to produce a subset of a set of assets in order to differentiate them. It quickly became aware that it would be easier to focus on a specific type of data. We quickly chose the companies' stocks as a good field to test.

2. Economics theory

Fortunately, we both come from an economic background could readily apply economic theory and best practices.

There is much theory built around the financial market and many papers written about financial management and the indicators to look at in order to differentiate them from one another.

Since, a financial manager is first in assessing an asset and a company's stock as an economic good, the most important thing will be for him his price. This is why we chose to look after the price of several companies' stocks.

The financial manager also tries to take decision for the future and to plan result in the future. Thus, he looks for patterns in data that are relevant over time. This is why we choose to look after historical prices and not only current prices to discriminate stocks.

Finally, we also analyse other factors about the companies that could be relevant for our project to better differentiate stocks.

3. Looking for the data

The importance of the financial market in our economy and the popularity of financial management make financial data about assets and companies' stocks easily accessible. These are only the first and only numerical data that we are able to reach just by opening a newspaper or browsing the web.

ii. Internet

The Internet was our first source in order to find any R packages relevant for our project. Secondly, the Internet allowed us to gather about the companies and about the variables of the market.

1. Companies' names

a. Nasdaq

We decided to focus on the NASDAQ, which is " The NASDAQ Stock Market, also known simply as the NASDAQ, is an American stock exchange. "NASDAQ" originally stood for National Association of Securities Dealers Automated Quotations. It is the second-largest stock market comparing to official stock exchanges by market capitalization in the world, after the New York Stock Exchange."[1]

The names of the companies and their representative symbol were required to be able to find later with the quantmod R package their results. We also needed some information about the companies and the belonging to already existing and objective index as sector or/and industry.

We find on www.nasdaq.com a csv file that was very convenient to deal with in R[2].

[1] Wikipedia.org, "NASDAQ"

[2] www.nasdaq.com/screening/companies-by-name.aspx?letter=0&exchange=nasdaq&render=download

b. Outliers and missing values

We ended up with 2698 different companies and 10 variables for each of the stock. We decided first to drop all the stocks with a missing value for the sector.

We also arbitrarily retain only the company with a market capitalisation above 1 billion. Even if, the one billion rule is roughly arbitrary it makes sense in our approach to only focus on big companies. Indeed, we can assume that the bigger the market Capitalisation is, the more likely a given stock would be to influence the market. Then, by focusing only on the Biggest companies, we should catch most of the patterns existing. Finally, for Nasdaq standards, a market capitalisation above 1 billion is not that rare.

We also get rid of the unnecessary or redundant variables. The URL of the company, it's IPO year (since the data were not reliable and the last sale as would have it through R and the quantmod package.

We ended with 587 companies with 6 variables each.

2. Risk-free

To compute the beta, an economic coefficient that we will introduce later, we needed the return of a risk-free asset. Usually, we considered the Treasury Bonds delivered by the Federal Reserve, especially long-term bond, since we considered the risk of the Federal Bank to default and the rate is of those Bonds is fixed as risk-free assets. Moreover, the historical of Treasury Bond are publicly available[3] on the website of the Federal Bank of St. Louis.

We use the monthly rate.

3. Market return

For the computation of the Beta, we needed as well the return rate of the market.

[3] http://research.stlouisfed.org/fred2/series/GS20/

a. S&P

We use the S&P 500 as the evaluation of the market return. "*The S&P 500, or the Standard & Poor's 500, is a stock market index based on the market capitalizations of 500 leading companies publicly traded in the U.S. stock market, as determined by Standard & Poor's. It differs from other U.S. stock market indices such as the Dow Jones Industrial Average and the NASDAQ due to its diverse constituency and weighting methodology. It is one of the most commonly followed equity indices and many consider it the best representation of the market as well as a bellwether for the U.S. economy."*[4]

We took the value of the S&P 500 every month.

b. Getting Returns

Then, we had to do some variable transformation by getting from the monthly levels of the S&P500 the monthly returns of this index. I use then a percentage calculus.

iii. R

Then, we had to do some variable transformation by getting from the monthly levels of the S&P500 the monthly returns of this index. I use then a percentage calculus.

1. Quantmod

Although the R language already contains many helpful functions, some of the best statistical assets come from third-party R libraries.

Gathering stock indices and calculating return on investments over various periods of time is a tedious yet essential task for our analysis. Luckily, there is an R library called quantmod available from CRAN, the R package repository. According to the project's website the aim of quantmod is to "is designed to assist the

[4] http://en.wikipedia.org/wiki/S%26P_500

quantitative trader in the development, testing, and deployment of statistically based trading models." [5] [http://www.quantmod.com]

We primarily use quantmod to grab time series of historical stock prices and returns over given periods of time.

As seen below, quantmod allows for the getSymbol() function which retrieves a specific stock's time series market capitalizations. We also use quantmod's periodReturn() function which encapsulates yearlyReturn(), monthlyReturn(), etc. As seen below our function period return handles this very nicely for a specified period.

2. Functions to get a large dataset

Now that we have quantmod doing the grunt work, we need to write a function that scrapes randomly 400 stocks from the NASDAQ index of 1000 stocks.

Below is our code to select the desired number of stocks, retrieve the historical prices, and calculate yearly/monthly/quarterly/weekly returns.

```
## Grab the timeseries data of j specified stocks
for (j in 1:length(SymbolList)) {
  getSymbols(SymbolList[j])
}

# Return yearly return of specific input stocks, adjust to remain `xtc`
# class for handling of data with quantmod
periodreturn <- function(i, time) {

  # paste time period with return ex. 'yearly' + 'Return'
  duration <- paste(time, "Return", sep="")
  # get specified time series using SymbolList
  timeseries <- adjustOHLC(get(SymbolList[i]), adjust = "split", use.Adjusted=TRUE)

  # call duration of return requested with computed time series
  do.call(duration, list(timeseries))
}

# call yearly function and merge results
mydata1 <- do.call(merge, lapply(1:length(SymbolList), periodreturn, list("yearly")))
```

b. Pre-processing:

i. Dimension Reduction

The reason we chose not to scrape the full NASDAQ index was due in part to two reasons. First, many of the companies have not been on the market for the full time period from '07 - present that we desire to analyze. Many, in the case of Facebook, only IPO'd in 2012 and others simply weren't contained in the index during our desired time period. Secondly, quantmod and thus our function seems to run into an error when scraping ~400 stocks. Both the matrix and the data frame that we hope to manipulate crash before we can even grab our full index. Thus, we are left with a testing set of ~400 stocks.

ii. Variable transformation

We have transformed some variables from S&P500 level into percentage returns.

iii. Computing the Beta: Feature creation /Aggregation

1. Economics theory

The Beta is an indicator that has been introduced by William Sharpe in his work on the Capital Asset Pricing Model (C.A.P.M), which earned him the Nobel Prize in 1990. This is a model expressed by an equation that tries to gives a way to calculate the return of a given asset (stock).

It says that the return of a given asset (stock), Ks is the sum of the return of the risk-free asset, Krf, and the risk premium, RP, difference between the market return and the risk-free asset return, weighted by the propensity of the stock to move with the market. This propensity of a stock to move with the market is its Beta, ß.

$$Ks = Krf + ß * RP$$

This is then a good indicator of the specificities of a given stock.

2. Computation

a. Preprocessing the data

To compute the beta, we use several steps. First, we collect the data as said in the first section of this report. We decided to aggregate the data on a monthly basis. This seemed for us as a good trade-off between accuracy and modelling. A month is small to reveal little difference between stocks but it is long enough to smooth the little variations of a stock's price.

To compute the beta, we needed the return of the market (Km) and used the S&P500 index to stand for the market return. We had, as explained before, to transform historical levels of the S&P500 into monthly returns by using a percentage formula.

We needed also the return of the risk-free asset. U.S Treasury Bonds stand for the risk-free asset. I transform the return given into percentage monthly return. Unfortunately, we only had available on the Internet the monthly return of the U.S Treasury Bonds up to April 2013. Thus we had to delete from our other data the value of the last two months.

```
Ret.month <- last(monthlyReturn(symbol), '-2 weeks')
```

We needed as well the return of a stock and this is where came to limitation forcing us to pre-process the data. We were able to upload the monthly of any given stock just by using its symbol, thanks to the R package quantmod. Since Quantmod was created in 2007 it only provides the historical prices of a stock since 2007. This is why we had to restrain our computation of the Beta to the period from 2007 to nowadays.

```
date.filtered <- date[date > "2006-12-31"]
```

Finally, we had to control the good matching between the different indicators once aggregate in single object. We had to reorder order the monthly returns of the U.S. Treasury Bonds, since they were order from the most recent one to the oldest one while the others were organized the other way around.

b. R function

In term of coding, we use two steps:

- **First,** we create a function to compute the beta of a given stock as output with its symbols as output. After having, create and pre-process two vectors SPretm.filtered and Rfree, respectably equals to 76 monthly returns of the S&P500 index and of the U.S. Treasury Bonds. We created a vector riskprem, which is the difference between the market return and the risk-free asset.

```
riskprem<- vector("numeric", 76)
for (i in 1:76){
        riskprem[i] <- SPretm.filtered$x[i] - Rfree[i]
}
```

We take the monthly returns of a stock thanks to quantmod and getSymbol. Then, make a linear regression and give as output the coefficient of the linear regression.

```
############# BETA function one stock.
BETAs <- function (symbol) {
               a <- length(symbol)
                      #number of month needed and the dimension of the vector created.
               riskprem <- riskprem

               Ret.month <- last(monthlyReturn(symbol), '-2 months')
#Monthly return (Not the last two week to have the same date as the period available
for the risk-free asset)
               Ret.monthF <- as.vector(Ret.month$monthly.returns)
#As a vector.

               a <- lm((Ret.monthF) ~ riskprem)

#linear regression
               return(coef(a)["riskprem"])
               #Return the coefficient of this linear regression.
}
```

- **Second** step, we created a function to compute the beta of several stocks by using a vector of stock indices. We used get() to get rid of the quotes. We return a vector with all the betas as output.

```
##Beta function
BETAm <- function (symbol) {
                              a <- length(symbol)
                              b <- vector("numeric", 76)
                              riskprem <- riskprem

                              for (i in 1:a) {
```

```
                c <- BETAs(get(symbol[i]))
                b[i] <- c
        }
        return(b)
}
```

iv. Sampling

To minimize the time of the calculus, we change a bit the R function and we added:

```
symbols <- function(i) {
evctor <- as.vector(sample(CompanyNasd.filtered$Symbol, i, replace = FALSE, prob =
NULL))
  return(getSymbols(vector, src="yahoo"))
}
```

to have a random sample of a given size.

c. Hierarchical agglomerative clustering

i. Preparation

To prepare our data for hierarchical agglomerative clustering model we need to scale our data set. To do this we subtract the median and divide by the median average deviation. Secondly, we need to compute a proximity matrix and in our case we use Euclidean distances.

```
medians1 <- apply(mydata1, 2, median)
medians2 <- apply(mydata2, 2, median)
mads1 <- apply(mydata1, 2, mad)
mads2 <- apply(mydata2, 2, mad)
s.mydata1 <- scale(mydata1, center=medians1, scale=mads1)
s.mydata2 <- scale(mydata2, center=medians2, scale=mads2)
d.mat1 <- dist(s.mydata1)
d.mat2 <- dist(s.mydata2)
h.ward1 <- hclust(d.mat1, method = "ward")
h.ward2 <- hclust(d.mat2, method = "ward")
h.single1 <- hclust(d.mat1, method= "single")
h.single2 <- hclust(d.mat2, method= "single")
h.complete1 <- hclust(d.mat1, method= "complete")
h.complete2 <- hclust(d.mat2, method= "complete")
```

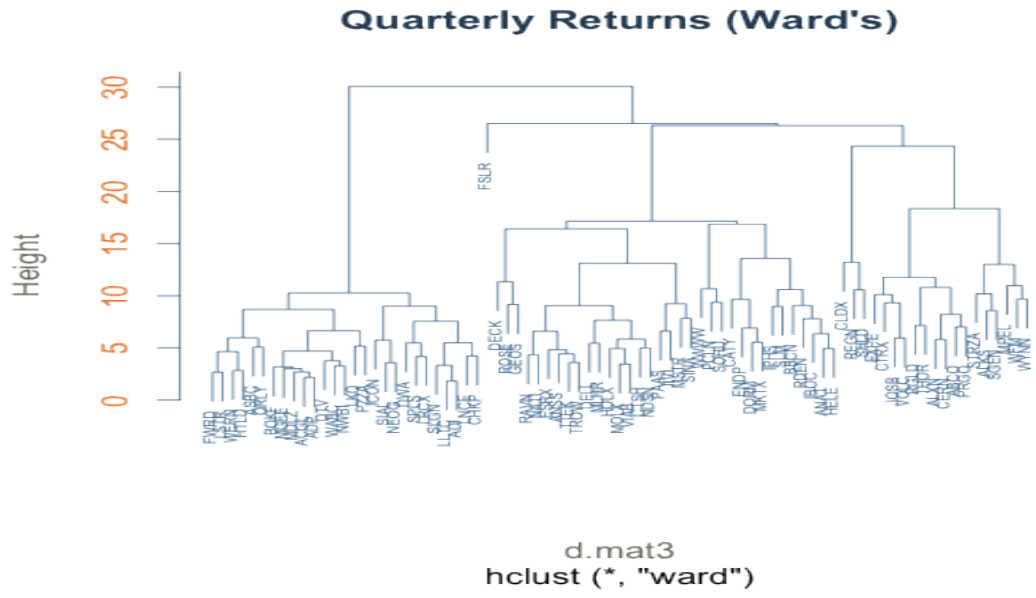
ii. Dendrograms

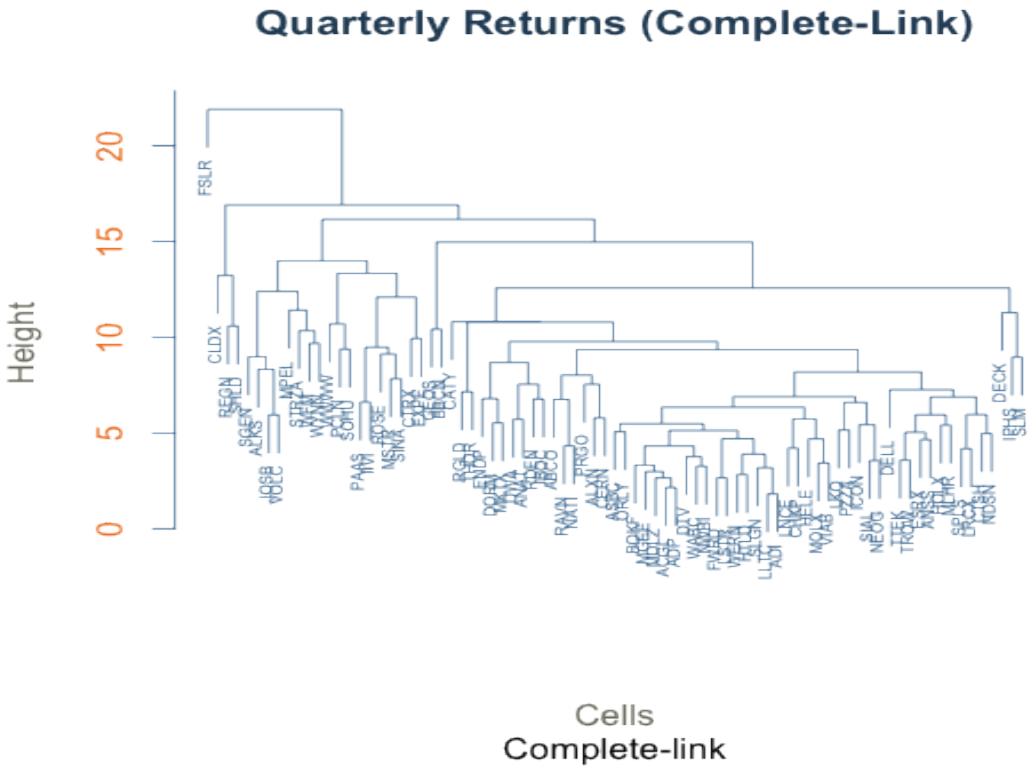
For our data the best graph-based approach to our data is to use Ward's method with quarterly returns. This leads to a very balanced dendrogram with three main clusters clearly visible [figure x1].

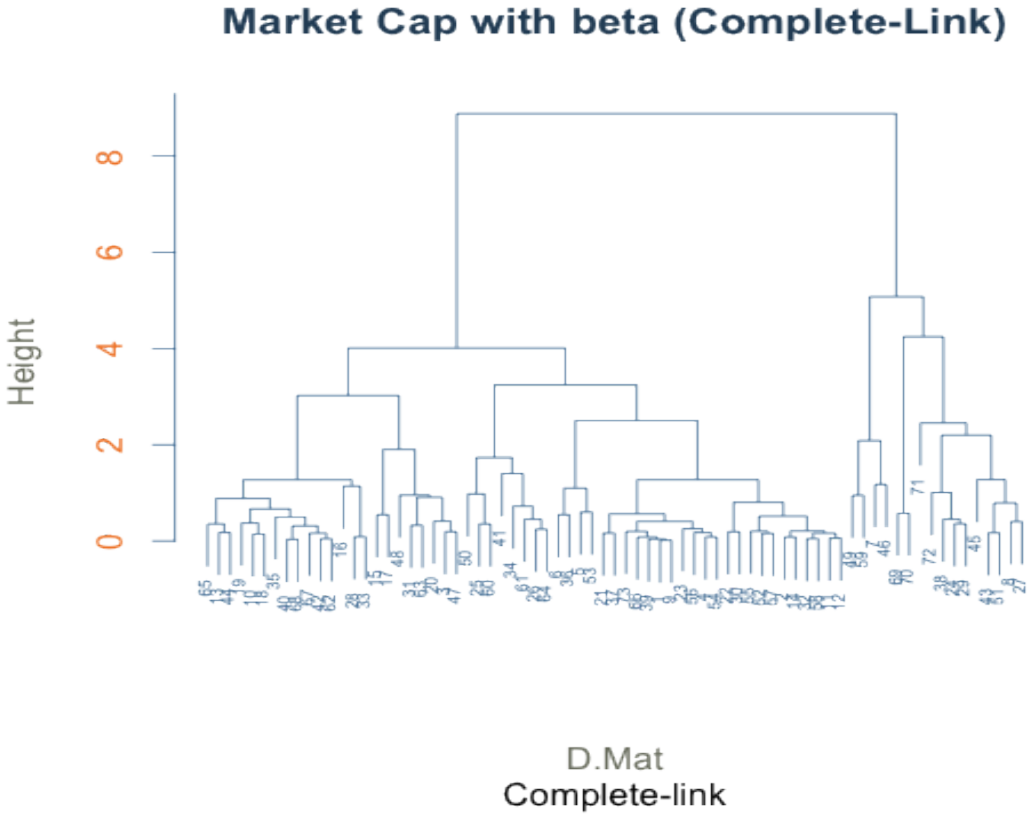
We have also computed and plotted a complete-link clustering dendrogram for our data set. It is not as clearly defined nor as optimal as our clustering using Ward's method. And although complete-link clustering is less susceptible to noise and outliers, our dendrogram visibly shows a subset of outliers as evidenced by the far right three stocks below [figure x2]. This may be useful for eliminating outliers in our data set.

In addition, we are interested to see the agglomerative clustering model as applied to our beta indicator data frame [figure x3] .

```
plot(h.ward1, cex = 0.5, main="yearly"    ,sub = "Ward's Method", xlab = "Cells", col
= "red")
plot(h.ward2, cex = 0.5, main="monthly"   ,sub = "Ward's Method", xlab = "Cells", col
= "red")
plot(h.single1, cex = 0.5, main="yearly"   ,sub = "Single-link", xlab = "Cells", col =
"blue")
plot(h.single2, cex = 0.5, main="monthly"  ,sub = "Single-link", xlab = "Cells", col =
"blue")
plot(h.complete1, cex = 0.5, main="yearly"   ,sub = "Complete-link", xlab = "Cells",
col = "green")
plot(h.complete2, cex = 0.5, main="monthly"  ,sub = "Complete-link", xlab = "Cells",
col = "green")
```




Quarterly Returns (Complete-Link)
Height
0
5
10
15
20
FSLR
CLDX
DECK
Cells
Complete-link


Market Cap with beta (Complete-Link)
Height
0
2
4
6
8
D.Mat
Complete-link

iii. Evaluation

Ward's method is optimal in our case because it plots the increase in SSE (sum squared error) that results from merging two clusters, which fit very well with our quarterly data set.

On the other hand, if we were to use complete link clustering we see a dendrogram with clearly visible outliers. Complete link clustering takes the maximum distance between two clusters, thus leading to many different branches as illustrated in our plot.

Finally, computing complete link on our beta indicator leads to a graph very similar to what we found with Ward's method on quarterly returns. This is likely because it is a scaled subset of our data and the distances between the nodes are not as extreme as they were in quarterly complete-linking, thus we have a very balanced dendrogram. Also of note is the height of this dendrogram. This reaffirms that our clustering is highly correlated.

Thus, the hierarchical clustering approach tells us that there is loose correlation between some subsets of stocks, which might indicated the possible existence of clusters. But the hierarchical clustering method is not precise enough to give information on the specificities of those clusters. This is why we try another approach.

d. Partitional clustering

i. K-means

After using hierarchical clustering, it ended up that a prototype-Based approach would more relevant to visualize the cluster, and be able to evaluate their relative position. We start with the K-means approach.

1. With Historical Prices

We start we the same dataset of a sample of 100 stocks and there prices from January 2007 for different periodicity: daily, weekly, monthly, quarterly and yearly.

a. Determination of the optimal number of clusters

The first step of the computation of the K-means approach is the determination of the optimal number of clusters.

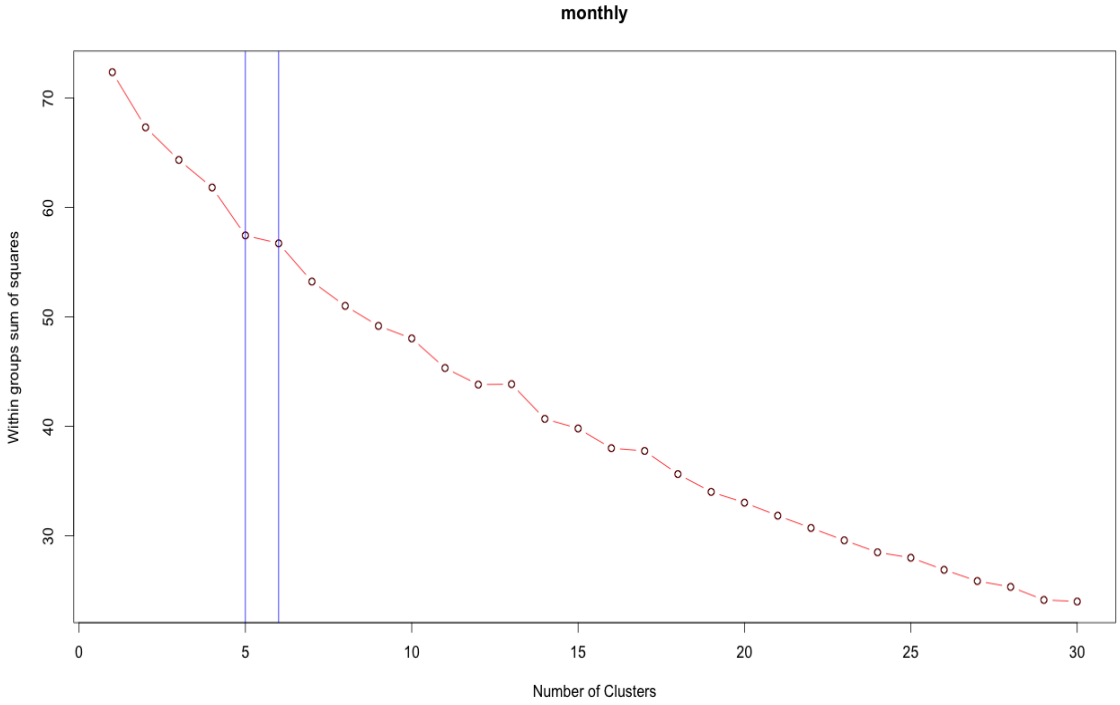
So for each periodicity, we try to make a K-means clustering with different number of centroids. For each, different number of centroids, we compute the Sum of Squared Errors.

```
wss1 <- (nrow(mydata1)-1)*sum(apply(mydata1,2,var))
wss2 <- (nrow(mydata2)-1)*sum(apply(mydata2,2,var))
wss3 <- (nrow(mydata3)-1)*sum(apply(mydata3,2,var))
wss4 <- (nrow(mydata4)-1)*sum(apply(mydata4,2,var))
wss5 <- (nrow(mydata5)-1)*sum(apply(mydata5,2,var))

# sum of squares
for (i in 1:30) {
  wss1[i] <- sum(kmeans(mydata1, centers=i)$withinss)
  wss2[i] <- sum(kmeans(mydata2, centers=i)$withinss)
  wss3[i] <- sum(kmeans(mydata3, centers=i)$withinss)
  wss4[i] <- sum(kmeans(mydata4, centers=i)$withinss)
  wss5[i] <- sum(kmeans(mydata5, centers=i)$withinss)
}
```

Then, we plot the results over the number of cluster to reveal the knee in the curve and then find the optimal number of cluster for this particular periodicity.

For instance monthly returns:



Thus in this example 5 is clearly a knee o the curve, and thus the optimal number of cluster for this dataset is 5.

We compute then the k-means of this dataset, say 100 stocks, with the supposed optimal number of clusters. We have to note that the number of different sector in this sample is 9.

b. Computation of errors

After having create the K.means for each periodicity, we needed to obtain all k means to evaluate which aggregate period was the best for a clustering-driven analysis of the dataset.

We then use several different indicators to evaluate the performance and the rightness of our different K-means clustering regarding the periodicity. We compare the sector of the company and the clustering given by the K-means to evaluate the entropy and the error of the K-means.

To do this, we create a dataframe with two columns, one with the sector of the company and one for the cluster given by the K-means algorithm.

(SymbolList.kmeans2 <- data.frame(type = mydata2.sector, # True value of classes

clusterA = mydata.kclust2)) #the cluster given by the K-means algorithm

Then, with the functions given in the solutions of the Homework 5, we compute the lumping, splitting error and the error rate, and the entropy.

We do it for every periodicity here is what we get :

	Periodicity	lumping.error	splitting.error	total.errors	total.error.rate
1	Yearly	702	472	1174	0.29
2	monthly	948	416	1364	0.36
3	quarterly	511	537	1048	0.27
4	daily	901	366	1267	0.34
5	weekly	1490	322	1812	0.48

Our first observation is that the total error rate is pretty low which is a relatively good indicator of the correlation between a stock prices and behavior and the sector of this stock.

We can see that the total error is bigger with short periodicity (daily or weekly) and quite lower for long periodicity (yearly and quarterly). There are several explanation for this.

- First, since the period of analysis is fixed, the shorter the aggregate periodicity is the more data we have and then the more chance to make a lumping or splitting error we have.

- Second, we have to think of what it means in term of the financial market behavior. By looking for clusters that will be correlated with sector, we assume that the market, which fixes the price of a stock, make links between stocks from the same sectors. This relation could is not obvious. The relatively lower total error rate of long period periodicity indicates the market should make those links in the relative long run.

- Third, the link between stocks within the sector good be explained by the fact that those company are in the same market. The evolution of stocks of the same industry is then representative of the evolution of the industry itself. So this influence is linked with the activity of the company, which is revealed to the

market every quarter usually by the turnover. But those announcements are not simultaneous and they could be lagged by month regarding the company. This is maybe why the stock prices need at least a quarter to really adjust. This explains that the quarterly return has the lowest total error rate.

c. Computation of errors with 9 clusters

Knowing that the number of different sectors in this set of stocks were 9, we compute for all the periodicity, a K-means with 9 centers instead of the optimal number of clusters for each periodicity.

	Periodicity	lumping.error	splitting.error	total.errors	total.error.rate
1	Yearly	118	116	234	0.24
2	monthly	184	74	258	0.30
3	quarterly	105	96	201	0.22
4	daily	195	73	268	0.31
5	weekly	230	62	292	0.34

We can notice that we have a significant decrease of the total error rate to a very low level, almost 20% for the quarterly returns.

d. Entropy of each Sector

We use the entropy to evaluate which of the sector were easy to evaluate and then were homogeneous and which were heterogeneous.

```
round(apply(table(SymbolList.kmeans1$type,SymbolList.kmeans1$clusterA),1,entropy.from.counts),2)
   Basic Industries        Capital Goods     Consumer Durables Consumer Non-Durables
                1.00                 0.92                  0.00                  0.00
  Consumer Services               Energy              Finance          Health Care
                1.21                 0.92                  1.42                  2.41
      Miscellaneous                  n/a      Public Utilities            Technology
                1.00                   NA                  1.25                  2.01
     Transportation
                0.00
```

We know that the higher the entropy is the less efficient the K-means is to predict this class. We can see that the Healthcare sector and the Technology sector

have very high entropy. It means that the K-means algorithm struggles to cluster them properly. We can deduct that the company within this sector are quite different, at least that they have stock with different behavior.

Comparatively, the Capital Goods sector and the Energy sector has a lower entropy which means that the company within those sector are more similar. This makes sense when we consider that companies in those sector are hugely big companies and very few in their market.

2. With the Beta

One of our goals was to find a good aggregate indicator of the behavior a stock. Moreover, using fewer variables will allow us to plot more easily our K-means.

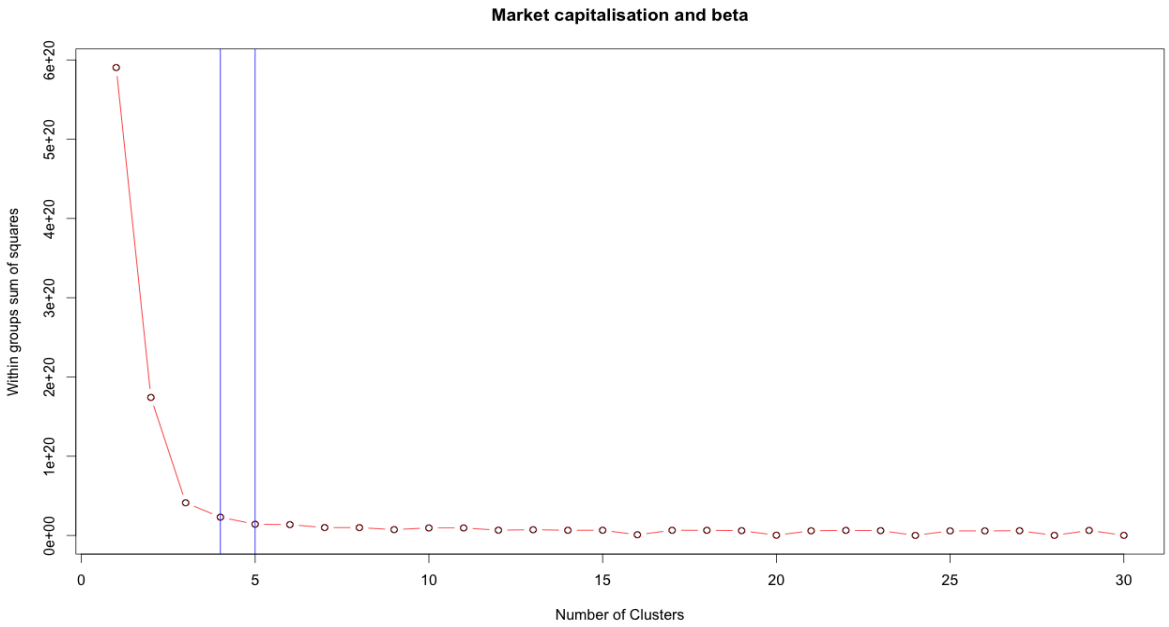
So we decided to build a matrix with one column for the market capitalisation and one column for the beta of the stock.

```
MC.B <- function (symbols) {
        a <- length(symbols)
        A <- matrix (0,a,2)
        A[,1] <- Market.Cap(symbols)
        for (i in 1:a) {
                A[i,2] <- 12*BETAs(get(symbols[i]))
                }
        return(A)
}
```

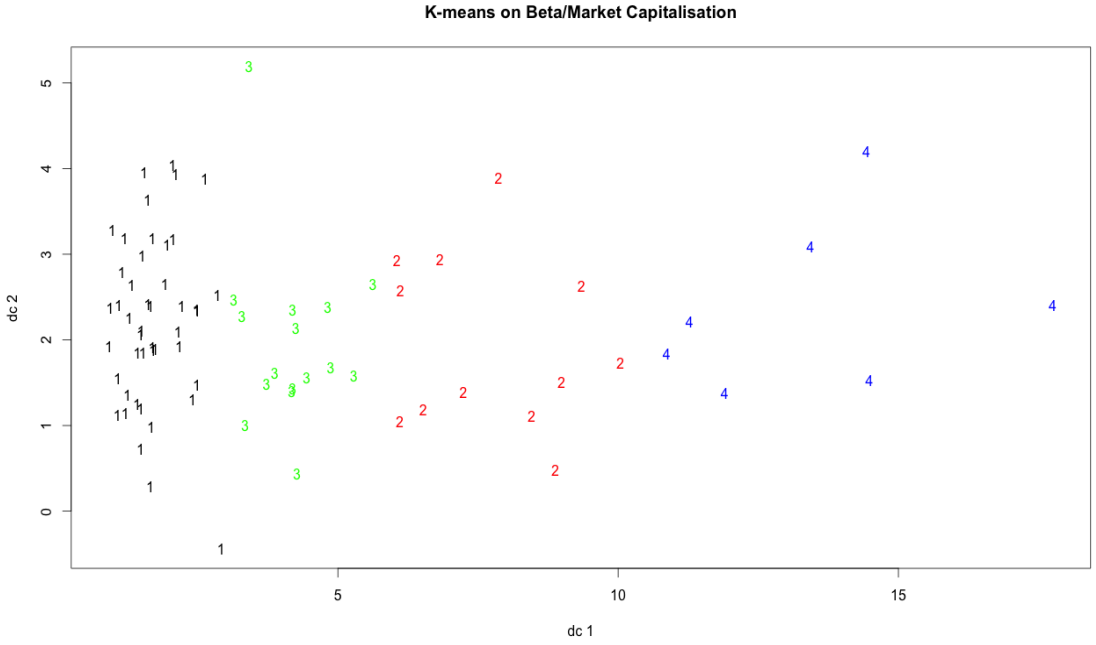
Because some companies as Google had a market capitalisation 20 times bigger than the others, we decided to take only companies with a market capitalisation between 1 and 20 billions of dollars.

```
Mydata.Beta.filtered <- Mydata.Beta[(Mydata.Beta[,1] < 20000000000),]
```

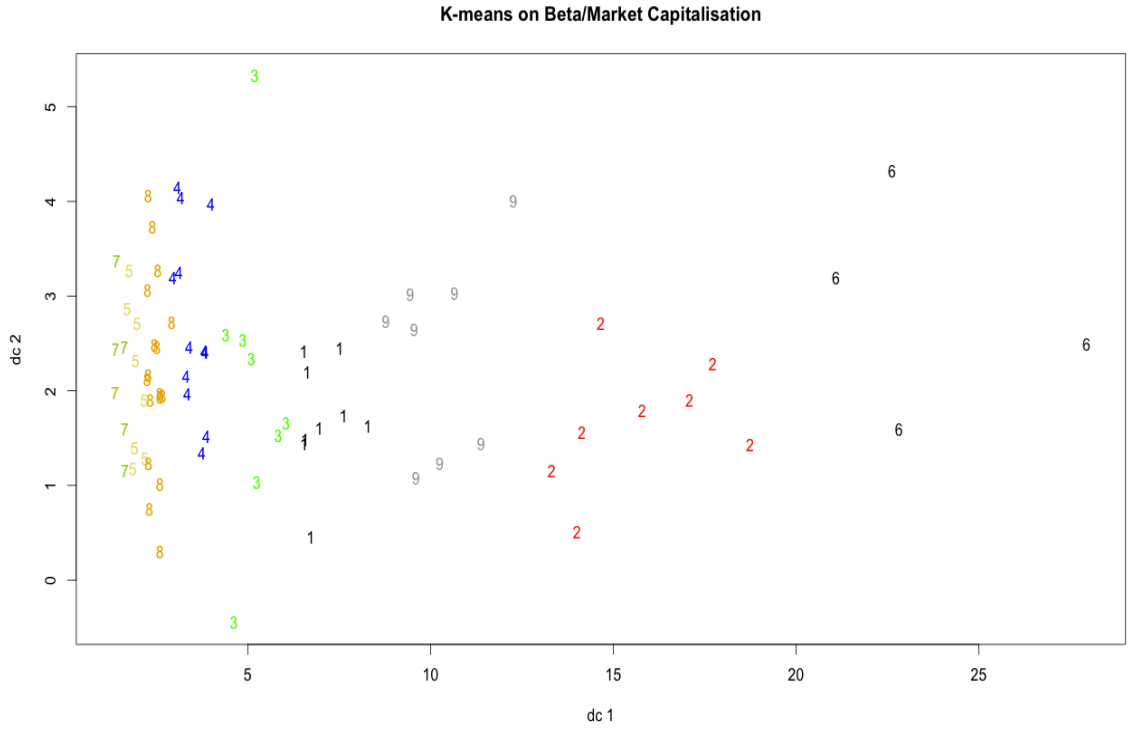
Then, we determine the optimal number of clusters.



Here is what we get with a 4 cluster K-means on the new matrix with the beta.



Here are a K-means with 9, as the number of different sectors, clusters.



ii. K-medoids

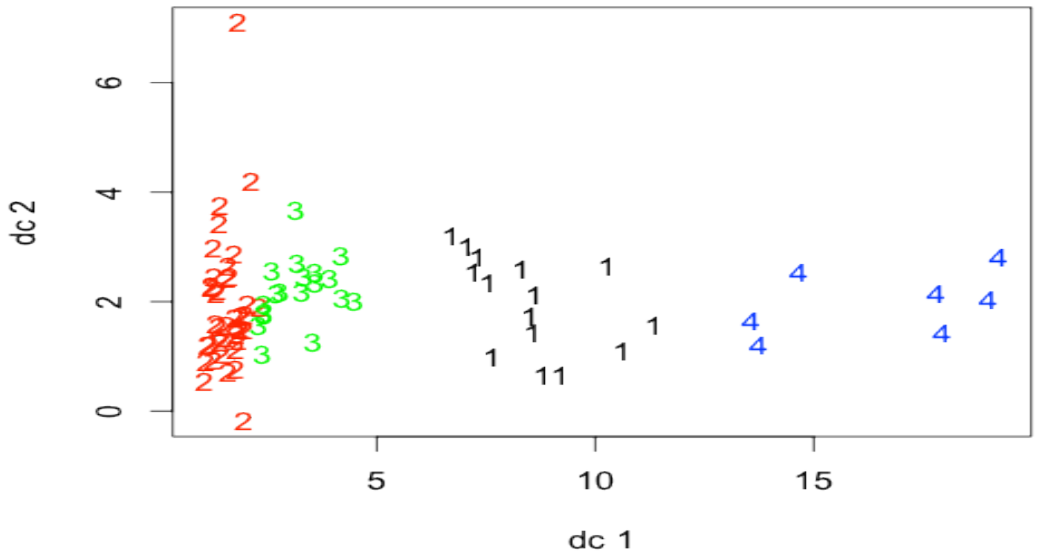
An extension of K-means is K-medoids, which unlike K-means requires that centroids be cluster centers. Also, K-medoids is more robust to outliers which leads to more quality clustering than K-means. For our clustering, we specified four clusters as it seems to be most significant given our small data set.

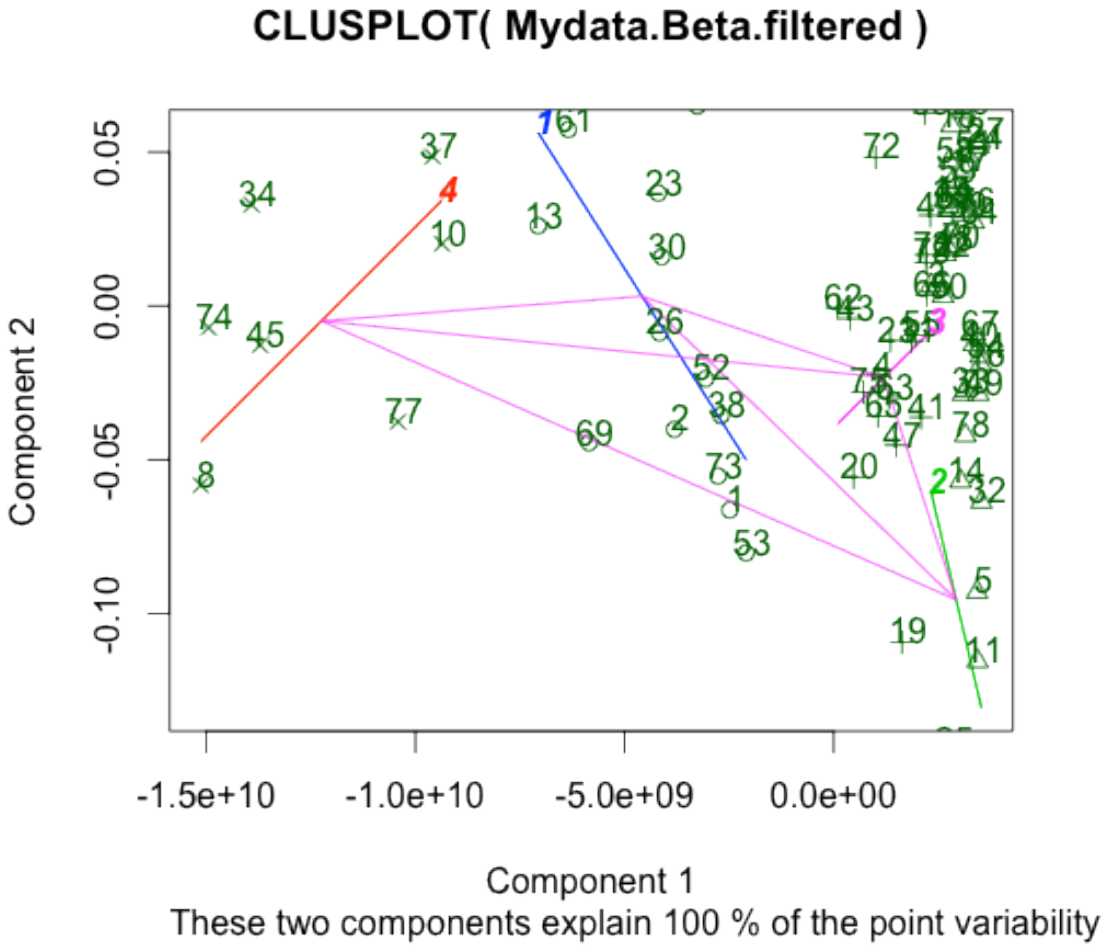
```
Kmedoids1 <- pam(Mydata.Beta.filtered,4)
plotcluster(Mydata.Beta.filtered, Kmedoids1$cluster)
clusplot(Mydata.Beta.filtered, Kmedoids1$cluster, color=TRUE, shade=TRUE, labels=2, lines=1)
```

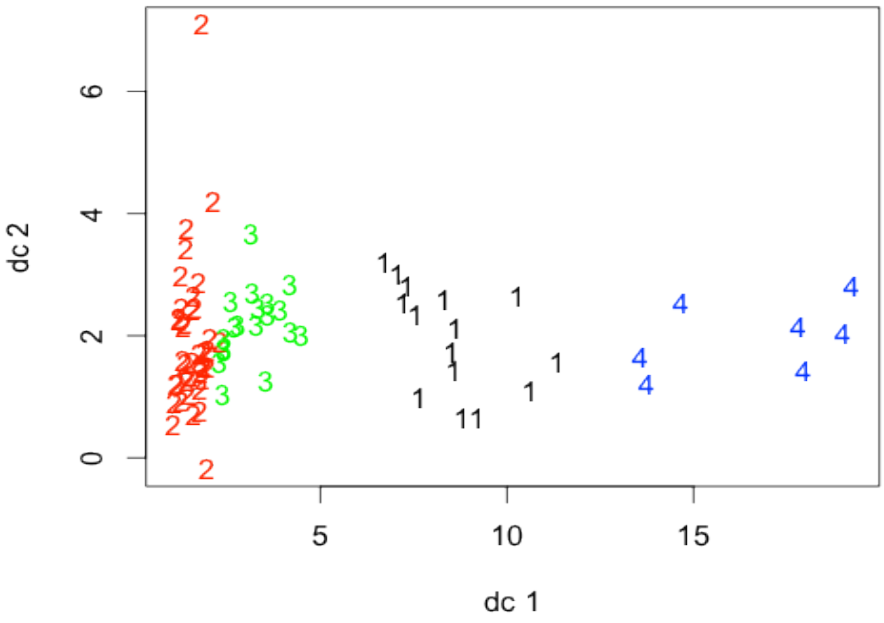
An extension of K-means is K-medoids, which unlike K-means requires that centroids be cluster centers. Also, K-medoids is more robust to outliers which leads to more quality clustering than K-means. For our clustering, we specified four clusters as it seems to be most significant given our small data set.

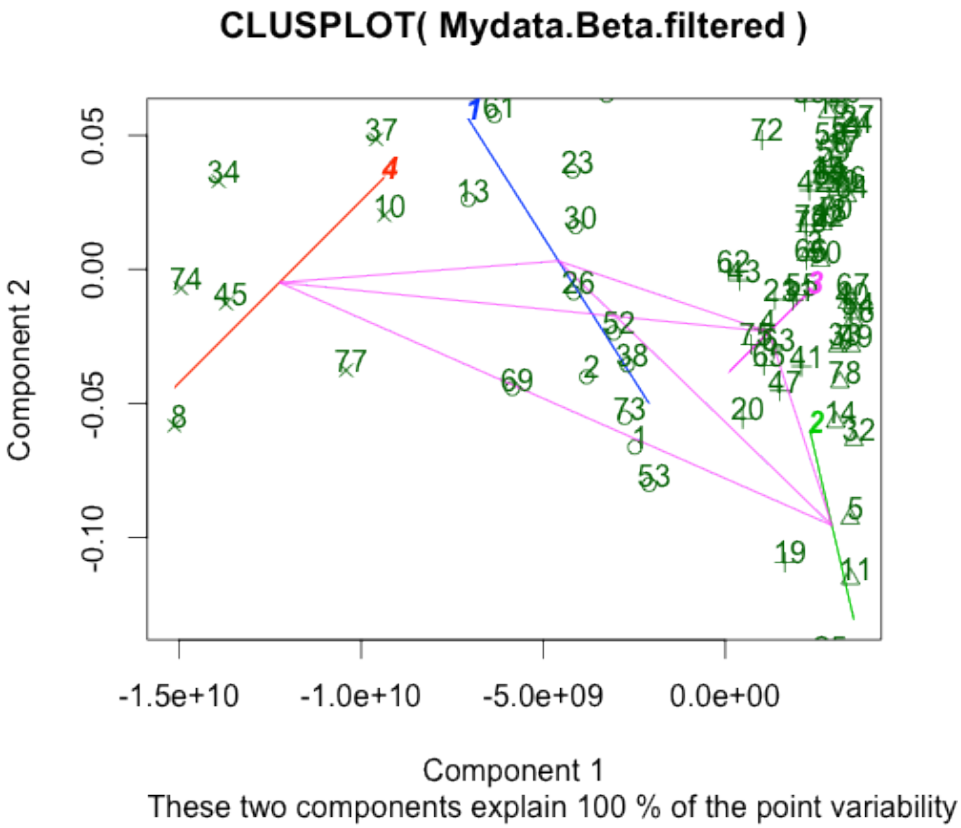
```
Kmedoids1 <- pam(Mydata.Beta.filtered,4)
plotcluster(Mydata.Beta.filtered, Kmedoids1$cluster)
```

clusplot(Mydata.Beta.filtered, Kmedoids1$cluster, color=TRUE, shade=TRUE, labels=2, lines=1)






dc 2
dc 1
0
2
4
6
5
10
15

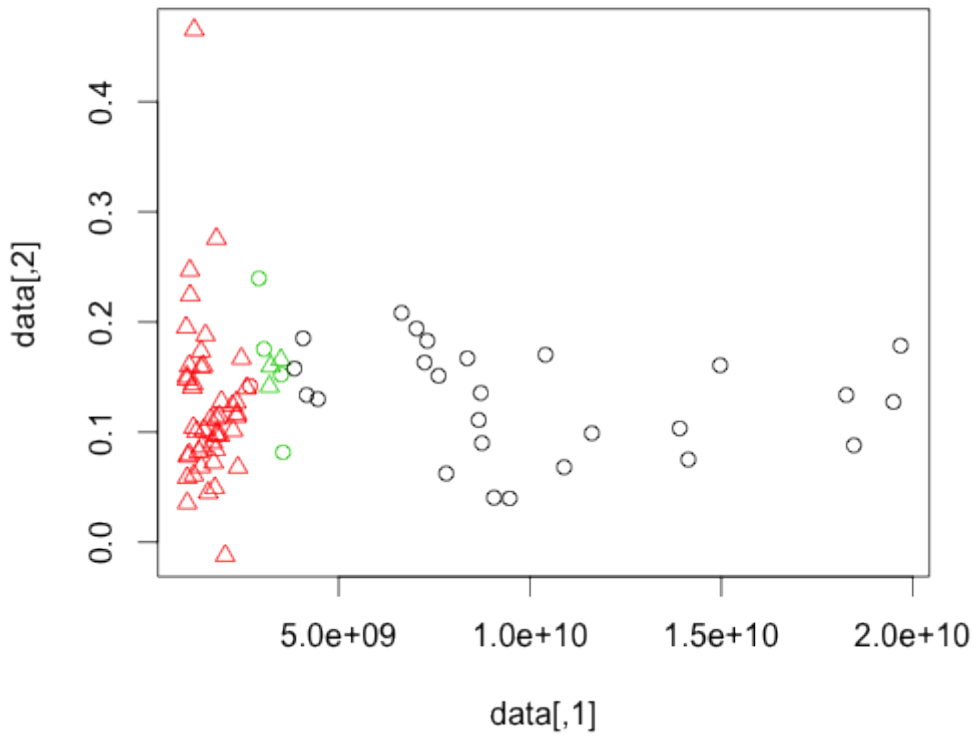

CLUSPLOT(Mydata.Beta.filtered)
Component 2
Component 1
These two components explain 100 % of the point variability
0.05
0.00
-0.05
-0.10
-1.5e+10
-1.0e+10
-5.0e+09
0.0e+00

e. Density-Based clustering

i. DBSCAN

First we remove outliers to prevent companies like Google and Exon, which have market caps exponentially higher than the rest, to prevent skewing of our dbscan cluster. We chose an Eps value of 2 to find the suitable amount of clusters, while marking the points surrounding them as noise. This Eps value also corresponds to sharpest increase of the fourth nearest neighbor of each point. We also chose the minimum points parameter to be 5.

```
Mydata.Beta <- MC.B(symbolfinal2)
Mydata.Beta.filtered <- Mydata.Beta[(Mydata.Beta[,1] < 20000000000),]
BETA.symbol <- rownames(Mydata.Beta.filtered)
Mydata.Beta.sector <- CompanyNasd.filtered[match(Mydata.Beta.filtered,
CompanyNasd.filtered$Symbol),]$Sector
dbscan(Mydata.Beta.filtered, 2, showplot=TRUE)
DBSCAN1 <- dbscan(Mydata.Beta.filtered, 300000000, showplot=TRU
```



ii. C-means

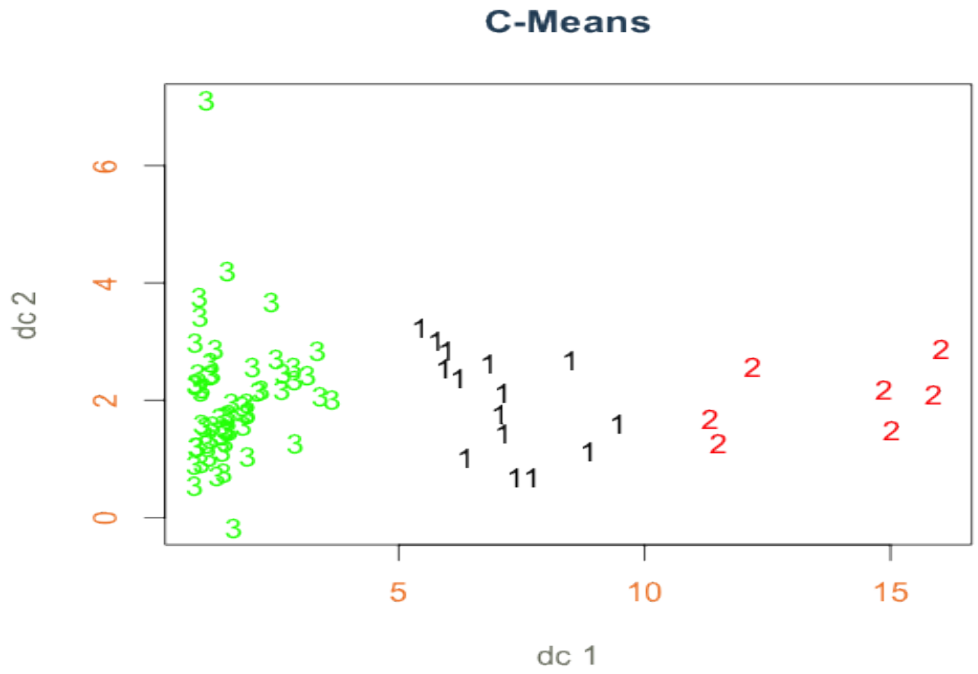
1. Setting

Lastly, we considered clustering our data set using a fuzzy version of K-means, called c-means. Like K-means it attempts to cluster by minimizing the SSE. We assign 3 centers and a weight of 1 to every data point. The fuzzy algorithm in turn calculates a weighted average for each point and this average assigns the point to a specific cluster.

```
cl <- cmeans(Mydata.Beta.filtered, 3, iter.max = 100, verbose = FALSE,
                              dist = "euclidean", method = "cmeans", m = 2,
                      rate.par = NULL, weights = 1, control = list())

plotcluster(Mydata.Beta.filtered, cl$cluster,
                      main="C-Means", col="#487AA1", col.main="#2B4C66",
                      col.lab="#7C8071", col.axis="#F38630")
```

2. Evaluation



III. Conclusions

a. Goals

We set out to discover a relevant clustering pattern in our dataset of stocks and test the hypothesis that sector is a good indicator to classify stocks.

We want to know what was the best aggregate level to analyse a particular asset in data mining.

We want to find a good aggregate indicator to sum up the historical behavior of an asset.

b. Results

i. Monthly and quarterly is a good periodicity

We find that by comparing a k-means clustering with the sector of the stocks, we ended with a clustering producing a quite low error rate, especially for quarterly periodicity.

We then use the Beta and the monthly return of a stock and of the market to produce several partitional and fuzzy clustering.

ii. Sectors are predictable even though some are more heterogeneous than others.

We computed the entropy of the different sector in comparison with a Kmeans on a dataset of 100 stocks and we ended with different level of entropy regarding the sectors.

This showed us that data mining could give us information on the structure of economic sectors.

iii. The beta is a good aggregate measure of stock's behaviour

The beta gives us a good indication on stocks behaviour.

c. Further Direction

i. Signal Analysis

We use data mining techniques and then we use discrete data, even for the historical prices of a stock, even if stock's prices change instantaneously and could be seen as continuous variables rather than discrete one. This is why signal analysis, or financial theory or mathematics is a more appropriate fields to analysis financial stock more precisely.

ii. More Data

a. Longer Period

Our analysis has been limited by the data available with the package (Quantmod) in R, thus we only analyze historical prices from 2007. This means that the period of study was very short, only 6 years, and then disadvantaged aggregate level of analysis as yearly prices, since there were only 6 prices available for each stock.

Moreover, the period 2007-2013 has particularly been marked by the financial crisis, which started in July 2007, then our analysis period was essentially a troubled period and might then not usual behavior of stocks by sector through time.

b. Larger number of stocks

On the financial markets, everyday millions of asset are exchanged and we only analyse less than 500 hundreds of them within the same stock exchange. Thus, we should probably extended our analysis to different stock exchange and to a large set of stocks and even assets.

Sources :

- *Fundamentals of Corporate Finance*, Ross, Westerfield, Jaffe, Jordan, 2011
- http://www.stat.pitt.edu/stoffer/tsa3/R_toot.htm
- *Introduction to DATA MINING,* P. Tan, M. Steinbach, V. Kumar, 2002