MSc Data Science & AI First Year
Raphaëlle ROTY

Internship Report

Stock Market Data Clustering: Functional Data Analysis Approach

May-July 2020

Research Director: Valentin PATILEA
Pedagogical Tutor: Michel RIVEILL

Contents

1 Acknowlegements

The internship I had with CREST Ensai was a great chance for learning new subjects and for professional development. Therefore, I consider myself very lucky as I could enrich my knowledge in data science modeling and use it to investigate real data problems.

I express my deepest thanks to Valentin Patilea who in spite of being busy with his duties, took time out to listen, guide and keep me on the correct path to carry out my project. I was highly pleased for having the opportunity to work for CREST Lab, a worldwide known research team.

2 Summary

The final goal of this study is to classify 120 companies listed on the Paris Stock Exchange into clusters by unsupervised methods. To observ patterns in reaction to the Covid-19 crisis we select a period from January, 1^{st} 2019 to July, 14^{th} 2020.

The expertise of clustering on a stock market dataset implies doing clusters on a high dimensionality matrix, which is of size n, the number of companies studied, times J, the number of days. To avoid this problem we use functional data analysis.

Functional data analysis (FDA) is an increasingly common class of statistical analysis. It consists of the study of data that can be considered a set of observed continuous functions like curves. In this report, we experiment clustering on functional data to reduce the size of the n vectors.

We will need a simulation study to compare the performance of the clustering methods on functional data: K-Means, Hierarchichal methods and Model-based clustering based on Gaussian mixture model.

The principle is to create artificial datasets by simulating trajectories according to a data generating process that mimicks the realizations from the real dataset. To compare the performance of each clustering method, we store the accuracy and the Adjusted Rand Index for each experience. The real stock market dataset is clustered with the best methods and parameters obtained. We finally try to explain the similarities between companies in the same cluster and reveal information about their behaviour facing the Covid-19 crisis.

3 Introduction

Stock market data are generally used to make investment choices, but they are also useful for understanding financial, economic and social phenomena. We observe the stock returns of 120 french companies as curves over a given period. The objective of this subject is to perform a classification of these curves to see the clusters according to the impact of Covid-19 and the consequences of this pandemic on French companies.

However doing clustering on really large dataset is not recommended, here the size is $n \times J$ with n, the 120 companies and J, the 390 days we observe. In order to rectify this, we use **functional data analysis** to extract from the curves smaller vectors than those of size J. To use it properly we need to know the mechanics specific to this analysis.

Functional data analysis is about the analysis of information contained in curves or functions. It deals with the statistical description and modeling of samples of random curves or functions. The fields of applications for FDA can be Medicine (EEG, ECG, Monitoring of Blood Pressure for patients), Weather and Pollution Forecast (Daily Profiles of Temperature, Solar Radiation), Econometrics (Stock market), Production Management (Quality of a product), Sports (performance analysis using sensor data) and many others. FDA is largely present in the current developement of AI.

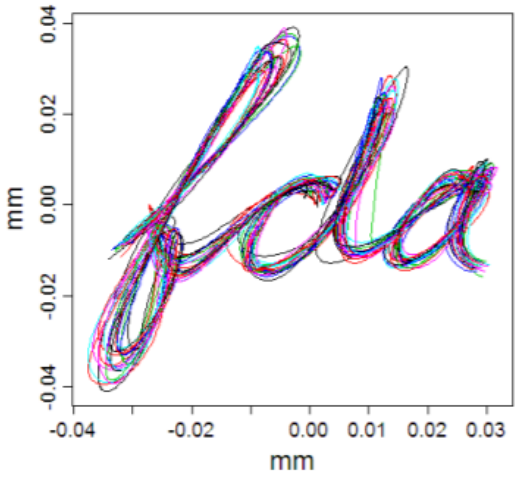


Figure 1: For example, these twenty replicas of the writing of "fda" are curves in two ways: first, as static traces on the page that you see after the writing is finished, and second, as two sets functions of time, one for the horizontal "X" coordinate, and the other for the vertical "Y" coordinate.

In this report, we focus on the econometrics side of the functional data by analysing data from the Stock Exchange of Paris. A curve represents the stock returns for a company quoted on the stock exchange over a fixed period. Using functional data analysis will allow us to obtain best performances on the clusters. Indeed instead of implying clustering methods on the whole dataset we do it on the extracted features of the curves, of reasonable dimensions.

The main challenge of this study is to carry out a model with satisfying and revealing results. To be certain of the confidence in our model, we generate functional simulated data closed to the stock market data. It will allow us to evaluate the quality and the revelance of the clustering methods and be sure we are not losing informations with the functionnal data analysis. We finally test the clustering methods on the features extracted from the simulated dataset. The clustering models compared are K Means, Hierarchical Methods and Gaussian Mixture Modelling for Model-Based Clustering.

We perform at the end the best algorithm, including methods and parameters on our real stock returns dataset.

4 Presentation of the research Laboratory

My research director is Valentin Patilea, a professor of statistics and the head of the master for smart data program in ENSAI[1]. He is a research director in CREST[2] where his research interests are: Semi and nonparametric statistics, survival analysis,time series, econometrics.

He has a PhD both in Statistics (Université catholique de Louvain, Louvain-la-Neuve) and Economics (University of Bucharest).

4.1 CREST Lab

CREST *(Center for Research in Economics and Statistics)* is a joint research center gathering the faculty members of ENSAE[3], ENSAI and Ecole Polytechnique Economics Department. It is located in both Saclay (91) for ENSAE Paris and Polytechnique and Ker Lann (35) for ENSAI. An interdisciplinary Center focused on quantitative methods applied to the social sciences, CREST comprises four sub-areas: Economics, Statistics, Finance-Insurance and Sociology.

The common culture of CREST is characterized by a strong attachment to quantitative methods, data, mathematical modeling, and the continuous back-and-forth movement between theoretical models and empirical evidence to analyze concrete economic and societal problems.

4.2 ENSAI

ENSAI is a prestigious higher education establishment for statistical engineering and information analysis that prepares its students to become Data Scientists.

ENSAI is home to part of the CREST research laboratory. ENSAI's professors are nearly all members of the CREST research laboratory, a joint research center which includes researchers from ENSAE and the Department of Economics of Polytechnique Paris, both members of the prestigious Institut Polytechnique de Paris.

The majority of the CREST lab at ENSAI is made up of Statisticians who are supported by Economics researchers and Computer Science researchers specialized in Machine Learning.

Research in Statistics and Computer Science at CREST – ENSAI, both applied and theoretical, covers a wide spectrum of themes related to Statistical Modeling and Economics.

[1] *ENSAI: National School of Statistics and Information Analysis*
[2] *CREST: Center for Research in Economics and Statistics*
[3] *ENSAE: National School of Statistics and Economic Administration Paris*

5 The stake behind the functional data analysis

To understand why it is revelant to use functional data analysis for clustering on stock market data we must know how is constructed a functional dataset.

5.1 What is a functional variable?

[Ferraty and Vieu, 2006] A random variable X is said functional if its values $X(t)$ are in a dimensional space infinite.

$$X = \{X_t : t \subseteq T\}$$

with $T \subseteq \mathbb{R}$ for a curve and $T \subseteq \mathbb{R}^2$ for an image. In our study we will consider curves and, without loss of generality, we take $T = [0, 1]$.

Let $X_i(t)$, $1 \leq i \leq n$, be an independent sample of size $n \geq 1$ of a functional random variable X. In practice, the curves are observed at discrete points in T (an alternative terminology is the "curves are measured or sampled"). For simplicity we suppose that the observation points are the same for all the curves in the sample. Let $t_1, \ldots, t_J$ be the observation points. Thus, the data can be organized as a table with n lines (the number of observation units) and J columns (corresponding to the J measurements of each observation units). There is no general relationship between n and J, but for the applications we have in mind $n \leq J$.

	t_1	t_2	$\cdots$	t_j	$\cdots$	t_J
X_1	$X_1(t_1)$	$X_1(t_2)$	$\cdots$	$X_1(t_j)$	$\cdots$	$X_1(t_J)$
X_2	$X_2(t_1)$	$X_2(t_2)$	$\cdots$	$X_2(t_j)$	$\cdots$	$X_2(t_J)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
X_n	$X_n(t_1)$	$X_n(t_2)$	$\cdots$	$X_n(t_j)$	$\cdots$	$X_n(t_J)$

Each observation $X_i(t_j)$ can be observed with or without error. Here we will consider the case where the curves are observed with unknown random error. That means, the available data are:

$$Y_i(t_j) = X_i(t_j) + \varepsilon_i(t_j), \quad 1 \leq i \leq n, \quad 1 \leq j \leq J.$$

The interest lies in X but only the values $Y_i(t_j)$ are available. That is, when the curves are observed with error, the data are available under the form:

	t_1	t_2	$\cdots$	t_j	$\cdots$	t_J
Y_1	$Y_1(t_1)$	$Y_1(t_2)$	$\cdots$	$Y_1(t_j)$	$\cdots$	$Y_1(t_J)$
Y_2	$Y_2(t_1)$	$Y_2(t_2)$	$\cdots$	$Y_2(t_j)$	$\cdots$	$Y_2(t_J)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
Y_n	$Y_n(t_1)$	$Y_n(t_2)$	$\cdots$	$Y_n(t_j)$	$\cdots$	$Y_n(t_J)$

5.2 Spaces of functions & Representation in a Basis

For modeling purposes, we consider that the curves, which are real-valued signals, live in a space of functions $\mathcal{X}$. The usual space considered in functional data analysis is the space of squared

integrable functions, that is $f \in \mathcal{X}$ if and only if $\int_T f^2(t)dt < \infty$, usually denoted $L_2(T)$ or $L_2[0,1]$ when $T = [0,1]$.

We suppose that $\mathcal{X}$ admits a countable basis $\{\psi_1, \psi_2, \ldots\}$ of curves defined on T (sometimes also called "dictionary"). In this case, any curve $X \in \mathcal{X}$ could be decomposed

$$X(t) = \sum_{k=1}^{\infty} c_k \psi_k(t), \quad t \in T.$$

Since the basis is fixed in the analysis, any curve in $\mathcal{X}$ is determined by the coefficients c_k, $k = 1, 2, \ldots$ When the curve X is supposed to be random, the coefficients c_k are random.

To make this formalism useful in practice, one should reduce the infinite representation of the curve, that is choose a K and consider the truncated approximate representation

$$X(t) \approx \sum_{k=1}^{K} c_k \psi_k(t), \quad t \in T.$$

Larger the K, more accurate this approximation is. Then we could say that the random X is (approximately) determined by a finite dimension random vector $(c_1, \ldots, c_K)$, instead of an infinite one. This vector is often called the vector of *scores* (in the basis), *loadings* or *features* of the curve. The choice of the basis has influence of the parsimony of the approximated representation. The interest, and the challenge, is to choose a basis such that the approximation of the curve is quite accurate with low values of K.

With a sample of curves $X_i(t)$, $1 \leq i \leq n$, we could use the same approximate basis decomposition

$$X_i(t) \approx \sum_{k=1}^{K} c_{i,k} \psi_k(t), \quad t \in T.$$

Then, to each curve observation X_i corresponds a $K-$dimensional vector $(c_{i,1}, \ldots, c_{i,K})$.

The tasks for the statistician given a sample $Y_i(t_j)$, $1 \leq i \leq n$, $1 \leq j \leq J$, of curve measurements with random error :

1. Determine a basis; it could be a given one (Fourier, polynomials,*etc.*), or a data-driven one (such as FPCA which stands for functional principal component);

2. Compute the vectors $(c_{i,1}, \ldots, c_{i,K})$, $1 \leq i \leq n$.

When J is large, and it is possible to obtain a good an accurate approximation with small K, the matrix $(c_{i,k})_{1 \leq i \leq n, 1 \leq k \leq K}$ has much less columns that the matrix $(Y_{i,j})_{1 \leq i \leq n, 1 \leq j \leq J}$. This means that the data can be summarized (compressed) by a lower dimension matrix.

6 Presentation of the dataset

A functional dataset ${X_i}^n{}_{i=1}$ is the observation of n functional variates identically distributed as X. We work on stock market dataset, however to prove the revelance of our method we will need to simulate a similar dataset where we know the label of each curve.

6.1 Stock market dataset

Stock market data are generally used to make investment choices, yet they are also useful for understanding financial, economic and social phenomena. For our case study, we want to examine whether a company - a sector or a group of companies - has experienced significant growth or decrement since the Coronavirus crisis. We consider only the french market with companies listed on the Paris stock exchange.

Our problem is as the following: we want to know if an event that took place on a specific date had an impact on a company. Intuitively, however, we suspect that the impact of the event can be anticipated by the markets; it can also have effects in the days following the event, beyond the day itself.

In our special case of the COVID-19 crisis we assist to different events. We got the first illness cases in China on November 17, 2019 and in France which is the same than in Europe on January 24, 2020. The World Health Organization (WHO) declares a state of public health emergency of international concern on January 30, 2020.

We must therefore consider the different political measures in France that happened between February and June 2020 to have a general understanding of the climate for investors and companies.

Insofar as revelant, we will consider a period of the event which goes from January 2020 to mid of July. To know if during this period, the action performed differently than usual, we must however have a comparison, an earlier period which is used to judge the difference. It is called the estimation period, it goes from January 2019 to January 2020.

For the estimation period, the literature recommends the use of a period of 180 to 200 days in order to obtain a period long enough to know the behavior of an action. We will recover the data of 390 days from the 1^{st} January 2019 to the 14^{th} of July 2020.

From where the data came from? We use the *quantmod* package on R to extract the stock data of the companies from Yahoo website. At the start, we just needed to know the symbol called ticker of the company. The 120 companies studied compose the stock index SBF120. The description of the index and all the companies selected can be found in appendix. [4]

[4] cf. Description of the companies studied

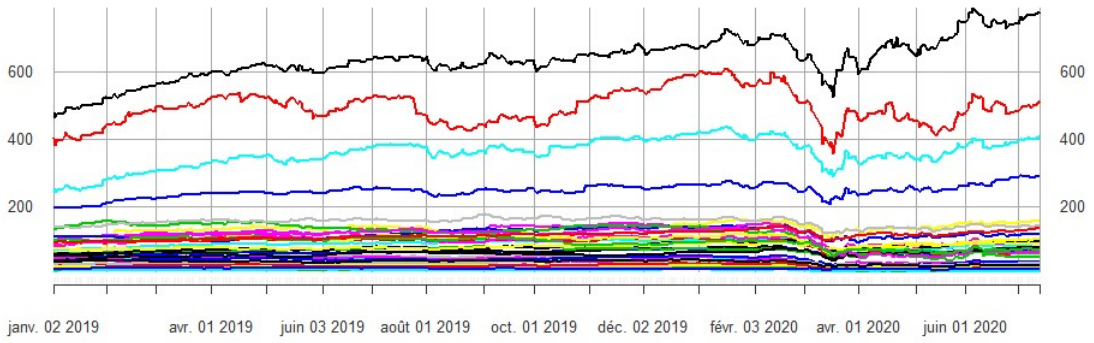


Figure 2: Original stock market curves from the quantmod package for the Cac40 companies.

After downloading the data from yahoo, we transform them into stock return [Fama et al., 1969] from the closed prices.

$$R_t = \frac{(P_t - P_{t-1})}{P_{t-1}}$$

where P_t is the closed stock price at the time t and R_t the stock return at the time t.

Thereby we obtain a percentage which represents the gain which would have been obtained by an investor who bought a share the day before and would have sold it the next day. By considering the 120 most actively traded stocks listed in Paris,we have a dataset composed by companies from various fields, which may have behaved very differently in consequence of the crisis.

We faced two extreme values in the stock market dataset. A value bigger than 1000 for Vallourec, a steel company and one smaller than -100 for Solvay, a chemicals company. They were extreme in the way that they unbalance the curves for only one day because of missing values. They have been replaced by the value of the previous day.

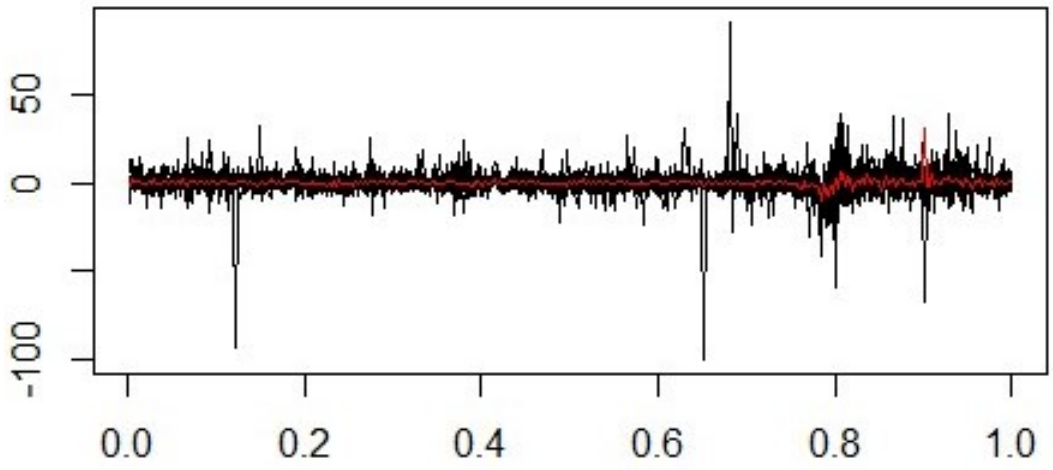


Figure 3: The overview of all the stock market curves $Y(t)$ once the data management is done. The axis x represents t the sequence of size 390, corresponding to the dates. The mean of the dataset is drawn in red.

6.2 Simulated data

Simulating data will allow us to test the efficiency of our method and the clustering algorithms. We create on R 1000 artificial samples to scientifically prove that the conserved methodology works.

To generate the simulated curve we disturb the average signal of the real data. Then from a financial point of view we add noise. Indeed, what we observe from stock market data does not correspond to the intrinsic value of the company. We must add a noise corresponding to market friction, delay, imperfect information. If the observed noise is not the true one, the same will be true for the performance.

6.2.1 The pure signal X_t

To create these data, we built three types of mean curve by randomly modifying the mean curve on the real data in three different ways. More precisely first we consider:

$$Mcluster_1 = 0.4 \times (mean_{data} + cos(n \times \pi \times t))$$

Next we generate:

$$Mcluster_2 = mean_{data} + sin(\frac{n}{10} \times \pi \times t)$$

Finally we generate for the last cluster:

$$Mcluster_3 = 1.7 \times mean_{data}$$

Where $mean_{data}$ is the vector of size J composed of the mean of the data at every time t_j. Those three mean clusters allow the simulation of curves that are at the same time similar but sufficiently different to have a significant distinction during the clustering. The clusters are created around these mean curves $Mcluster_1$, $Mcluster_2$ and $Mcluster_3$ by pertubating randomly all J values.

$$X_t = Meancluster_a \times random$$

With X_t the pure signal, $a \in \{1, 2, 3\}$ and $random$ a multivariate normal random vector which gives J normally distributed random numbers of $mean = 1$ and $sd = \frac{sigma_{pure}}{i_{pure}}$.

i_{pure} is a inflating or deflating factor to create more difference between classes, it is set to $i_{pure} = 0.25$ for the cluster 1, to $i_{pure} = 4$ for the cluster 2 and to $i_{pure} = 1$ for the cluster 3. $sigma_{pure}$ corresponds of the variability of the random parameter, it is set to $sigma_{pure} = 0.25$.

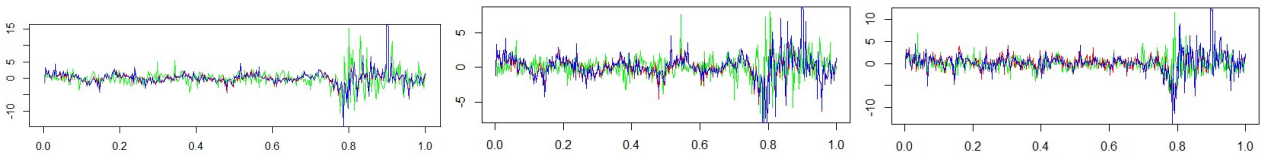
The data are randomly declared in the cluster 1, 2 or 3 thanks to the parameter p, an uniform random variable between 0 and 1.

For $p < b$, X_t is in the cluster 1.

For $b < p < c$, X_t is in the cluster 2.

For $p > c$, X_t is in the cluster 3.

The bounds b and c are fixed to 0.33 and 0.66 to encourage a balanced dataset. After the results on the real dataset, as far as it is unbalanced we change c to 0.91 so the proportions match.



(a) Cluster 1 (b) Cluster 2 (c) Cluster 3

Figure 4: For each cluster, the green curve corresponds to the original stock market curve for one company, the red curve is the signal with noise Y_t of a simulated curve in the cluster and the blue one is the pure signal X_t of this same curve.The curves displayed are choosen randomly.

6.2.2 The noise ϵ_t

To have noisy functional data closed to the real dataset, we need to build a suitable error variable ϵ_t.

To do so, we perform a linear regression to extract the c_k features of the noise which corresponds to a polynomial. The linear regression is executed on a new data $\bar{Z}$. Instead of using the average of the stock market data, we create a new average $\bar{Z}$.

$$\bar{Z} = \sum_{i=1}^{n} \sum_{t=0}^{1} [\, data[i,j] - mean(data)[i] \,]$$

.

The degree of the functions and the constant are defined in the way that the ajusted R^2 score for the linear regression is optimal.The degree of the polynomial regression for the noise is fixed from 5 to 5 until 25 and the simplified coefficients kept are: $c_{noise} = [-9, -203, 2526, -7528, 8577, -3380]$

The noise of the simulated data corresponds to the following expression.

$$\epsilon_t = \sqrt{\| \sum_{k=1}^{K} (c_{noise}[k] \times t^{5\times k}) \|} \times eps$$

K is the size of the vector c_{noise} and eps is drawn randomly by a uniform law times σ_{eps}, the variance of epsilon specific to each cluster. We fit the polynomial in t at the end to determine the variance of the error terms as a function of time. this allows to build heteroscedastic noise, a feature that we observe in the real data, the noise has not the same amplitude given circumstances.

7 Features extraction from functional data

Our model is as the following, each observation $Y_i(t_j)$ can be represented into a basis as:

$$Y_i(t_j) = X_i(t_j) + \varepsilon_i(t_j), \quad 1 \leq i \leq n, \quad 1 \leq j \leq J.$$

where $X_i(t_j)$ represents the real value not the observation and $\varepsilon_i(t_j)$ the error of measurement.

$$X_i(t) \approx \sum_{k=1}^{K} c_{i,k}\psi_k(t), \quad t \in T.$$

The goal of this part is to transform the observed curves into functional data, it means that we will extract the vector of the features c_i for each curve $Y_i(t)$ according to a basis $\psi(t)$ of K functions. In that way we will obtain smaller vectors to perform the clustering than taking into account the ones of size J.

Least squares smoothing

Recall that $\{\Psi_1, \Psi_2, ...\}$ denotes a basis of functions. The coefficients $c_{i,k}$ can be obtained by the regression of the criterion of least squares by minimizing the sum of squared estimate of errors [5].

$$SSE(c) = (Y - \Psi c)' (Y - \Psi c)$$

with the vector of the features c_i of length K and Ψ a matrix of size $n \times K$ that contains the values of $\psi_k(t)$ for each curve. In our work the basis $\{\Psi_1, \Psi_2, ...\}$ is a data-driven basis. You will find the detailed method in appendix.

At this stage we know how to obtain the features of the curves $c_{i,k}$ however we still have to deal with the size K of those objects.

A large value of K leads to an almost perfect fit of the curve to observations, which however leads to the risk of adjusting also the measurement error which should be ignored. At the same time, a small value of K can lead to ignore some important aspects of the smoothed function. In other words, **the larger K is, the greater the variance, the smaller the K, the greater the bias**.

To do a compromise between bias and variance, we will keep the K that minimize the Mean Squared Error:

$$MSE(\ \hat{y}(t)\ ) = \frac{1}{J}\sum_{t=1}^{J}[y(t) - \hat{y}(t)].$$

At the end, with the least squares smoothing we obtain the following $Y_{smooth}(t)$ curve for every company i with $i \in [1 : n]$ from the original market dataset. $Y_{smooth}(t)$ is created as functional data object from knowing t and the $c_{i,k}$ extracted, hence its shift with the original data.

[5] also known as the residual sum of squares (RSS) or the sum of squared residuals (SSR)

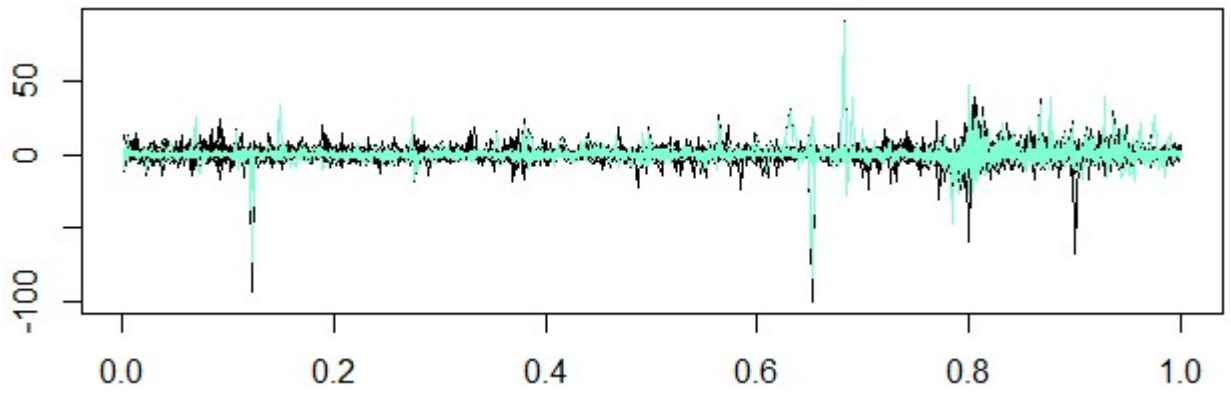


Figure 5: Smooth Dataset is drawn in green and the original stock market dataset in black according to t the sequence from 0 to 1 of size 390.

8 Clustering algorithms and measures of revelance

The aim of the cluster analysis is to build homogeneous groups, called clusters, of observations representing realisations of some random variable X.

The particular choice of a cluster algorithm depends on the type of data and on the type of goal.

Partition Method: A partition method construct k groups where the number k is provided by the user. The goal of the method is to provide the best partition into k clusters.

Hierarchical Method: An hierarchical method works with all possible values of k from 1 to n using certain rules to separate or aggregate data into a group.

First of all we will describe how to measure the revelance of a method and which indicators we kept to do model selection. Secondly we will detail the methods used.

8.1 Measures of revelance

As far as our method will be trained on simulated data we can measure the revelance of the algorithms. There are two purposes to compute such indicators. First it will be easier to have better performances by knowing with which parameters to play. Secondly we are going to be able to compare differents methods together thanks to those indicators.

The confusion matrix will allows us to visualize the performance of an algorithm.

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

The matrix of confusion, also known as an error matrix, is a special kind of contingency table, with two dimensions "actual" and "predicted", and identical sets of "classes" in both dimensions.

From the confusion matrix, we decide to compute two main indicators: the accuracy and the F1 score.

8.1.1 Accuracy

The accuracy is the proportion of correct predictions (both true positives and true negatives) among the total number of cases examined.

$$ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

where TP = True positive, FP = False positive, TN = True negative, FN = False negative.

8.1.2 F1 score

The F1 score (also F-score or F-measure) considers both the precision p and the recall r of the test to compute the score.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The precision is the number of correct positive results divided by the number of all positive results returned by the classifier.

$$Precision = \frac{TP}{TP + FP}$$

The recall is the number of correct positive results divided by the number of all relevant samples.

$$Recall = \frac{TP}{TP + FN}$$

8.1.3 ARI

The Adjusted Rand Index (ARI) is frequently used in cluster validation since it is a measure of agreement between two partitions: one given by the clustering process and the other defined by external criteria.
Given the following contingency table $[n_{i,j}]$ where $n_{i,j} = |X_i \cap Y_i|$.

X/Y	Y_1	Y_2	$\cdots$	Y_s	$sums$
X_1	$n_{1,1}$	$n_{1,2}$	$\cdots$	$n_{1,s}$	a_1
X_2	$n_{2,1}$	$n_{2,2}$	$\cdots$	$n_{2,s}$	a_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
X_r	$n_{r,1}$	$n_{r,2}$	$\cdots$	$n_{r,s}$	a_r
$sums$	b_1	b_2	$\cdots$	b_s	

The original Adjusted Rand Index using the Permutation Model is computed with the $n_{i,j}, a_i, b_j$ values from the contingency table.

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}$$

8.2 The algorithm of the K-Means

K-means clustering tends to find k clusters around some initial centers: $\{m_1^{(1)}, ...m_k^{(1)}\}$

The original algorithm proceeds by alternating the next two steps:

- **Assignment step:** Assign each observation to the cluster with the "nearest mean".

$$S_i^{(t)} = \{X_p \ : \ \|X_p - m_i^{(t)}\|^2 \leq \ \|X_p - m_j^{(t)}\|^2, \forall\, 1 \leq \ j \leq \ k\}$$

  where X_p is assigned to exactly one cluster.

- **Update means step:** Compute the new means to be the centroids of the observations in the new clusters. $m_i^{(t+1)} = \frac{1}{\|S_i^{(t)}\|}\sum_{X_j \in S_i^{(t)}} X_j$

This algorithm could be modified to avoid the problems with the mean and the outliers. Whether by using the K-means algorithm with the $(1-\alpha)\%$ deepest data. Or instead of using the means, use another centroid (deepest point in each cluster).

8.3 Hierarchical methods

Basically, the hierarchical cluster uses the distance matrix to produce the dendrogram. The key is to select an agglomerative or divisive method and to select a method to compute distances between a group and a individual datum (Linkage criteria). The number of clusters is define by computing the percentage of inertia explained by the clustering tree.

8.4 Model-based clustering based on parameterized finite Gaussian mixture models

We assume that distribution of the features of every observation is specified by a probability density function through a finite mixture model of G components, which takes the following form:

$$f(c_{i,k}; \psi) = \sum_{k=1}^{G} \pi_k f_k(c_{i,k}, \omega_k)$$

where $\psi = \pi_1, ..., \pi_{G-1}, \omega_1, ..., \omega_G$ are the parameters of the mixture model, $f_k(c_{i,k}, \omega_k)$ is the k^{th} component density for the feature $c_{i,k}$ with parameter vector ω_k, $(\pi_1, ...\pi_{G-1})$ are the mixing weights or probabilities (such that $\pi_k > 0, \sum_{k=1}^{G} \pi_k = 1$), and G is the number of mixture components.

In the case of Gaussian mixture model, each cluster corresponds to a gaussian distribution. Models are estimated by EM algorithm [6] initialized by hierarchical model-based agglomerative clustering.

The EM algorithm [Dempster et al., 1977] starts with two randomly placed Gaussians (μ_a, σ_a) and (μ_b, σ_b) somehere in the space, similarly to the K-means algorithm. Then it decides for each point X_i if it came from the Gaussian a or b and gives the corresponding probability:

$$P_a(X_i) \; and \; P_b(X_i)$$

In the case of K-means, it would have assign it. That is why the EM algorithm is considered as soft clustering. Using the computed probabilities, the mean and variance of the Gaussian are adjusted to fit points assigned to them. The optimal model is then selected according to BIC (Bayesian information criterion).

9 Simulated Data Experiences

Sampling simulated data avoid us to jump into conclusions by revealing robust and credible results. Indeed as far as we do not have a test dataset, simulated data will allow us to try parameters and to trust the results by evaluating the methods.

For each method, we test the different parameters of the algorithms before running them between 250 to 1000 replications on the simulated dataset. We store the measures of revelance to compare them and use only the best one on the stock market data. Finally we mainly use the accuracy and the ARI to confront the results as far as the F1 score is composed of a score for each dimension.

At the end, we run 1000 times each clustering algorithm on the different data parameters:

- The dataset is large enough and balanced ($n = 100$,$J = 400$, $b = 0.33$, $c = 0.66$, so with each cluster corresponds to 33% of the dataset)
- The dataset is large enough and unbalanced ($n = 100$,$J = 400$, $b = 0.33$, $c = 0.91$, so each cluster coresponds to respectively 33%, 58% and 9% of the dataset)
- The dataset is small and balanced ($n = 50$,$J = 400$, $b = 0.33$, $c = 0.66$)

We test small dataset to see if we can perform the clustering only on a group of companies, for example only on the Cac40 companies, hence 50 choosen as the size of the sample.

9.1 K-means and dendogram results

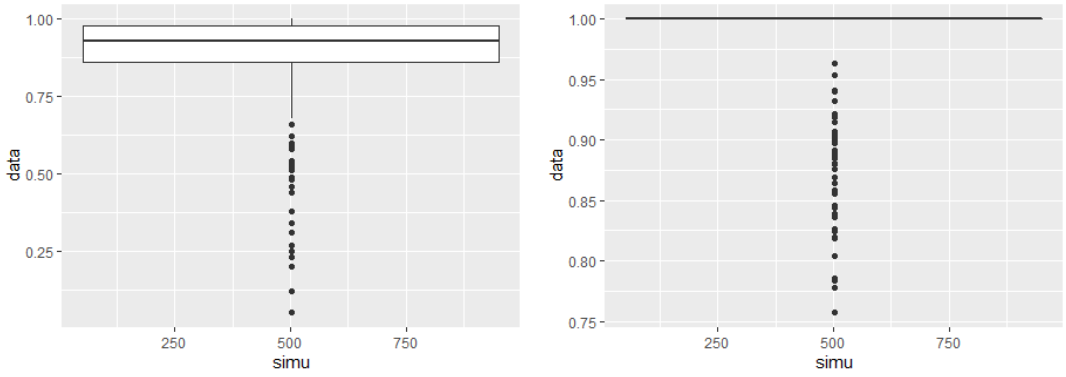
As far as the results of the K-means and the dendogram were not revelant for our analysis you may find the application, the results and the reasons of not keeping those methods in appendix (*K-Means application and results* and *Dendogram application and results* from page 28).

[6] Expectation and Maximization algorithm

9.2 Mclust application and results

In R, the function used to perform the model-based clustering is mclust. With 1000 repetitions on the simulated dataset, the mean of the accuracy obtained is 0.90 and the ARI is 0.99 on the features extracted for the balanced simulated dataset. Given these excellent results, we keep this method for the application on the stock market dataset.

Nevertheless to be sure of the results given by this method, we examine a little bit deeper the clusters obtained. Firstly for balanced dataset, secondly for unbalanced dataset to confirm that it is working for our real data.



(a) Accuracy (b) ARI

Figure 6: Boxplots of Accuracy and ARI for the large balanced dataset by mclust

As we can see on the two boxplots above the results are really satisfying for our study. For the balanced dataset, the percentage that mclust find the right number of cluster[7] is 95.8%. We can affirm that with this method and a large dataset the risk of error is below 5%, we observ this rate on the following graph.

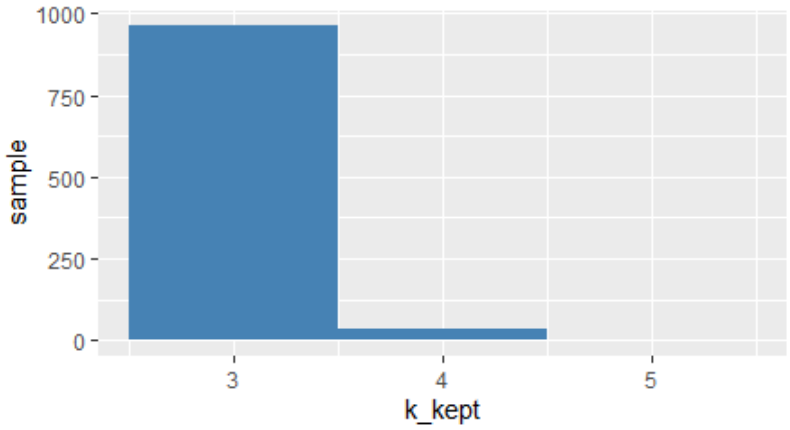
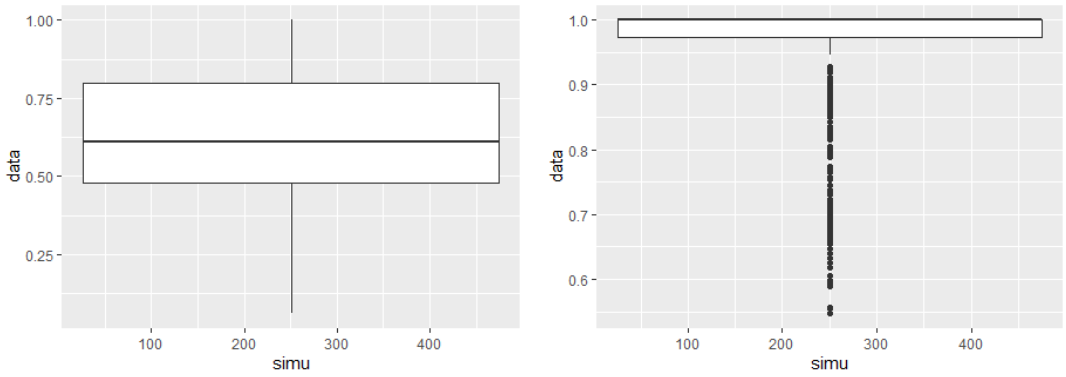


Figure 7: K kept by the mclust method for balanced dataset of size 100

[7]k kept is the optimal number of clusters found by the algorithm

For the small sample even if the indexes are closed ($accuracy = 0.84$ *and* $ARI = 0.98$), k is significantly different as far as for 100% of the simulations the algorithm found $k = 4$. We conclude that we can not use the model-based clustering based on parameterized finite Gaussian mixture on small dataset such as the Cac40 companies.

Concerning the unbalanced large dataset, all the simulations concluded $k = 3$. We decide to check the indexes to be sure of our conclusions.



(a) Accuracy (b) ARI

Figure 8: Boxplots of Accuracy and ARI for the large unbalanced dataset by mclust

The accuracy for the unbalaced dataset is more scattered than for the balanced dataset. This is due to the 27.6% of the accuracy results below 0.5. Although the ARI results maintain high, we need to take into account those results: the mclust method is less efficient on unbalanced dataset. Since for all the simulation it it still finds the correct number of clusters, we will trust this algorithm for our study.

10 Results on the Stock Market Dataset

We apply the Model-Based clustering on Gaussian mixture on the stock market dataset composed of 120 companies listed on the Paris Stock Exchange. By curiosity we also run the two other clustering algorithms. We may use them as comparators.

10.1 Clusters obtained

We obtained 3 clusters with both methods: the mclust function and the k-means. The results are summarize below for the whole dataset.

	Cluster 1	Cluster 2	Cluster 3
Mclust	**40**	**74**	**6**
K-means	40	79	1
In common	40	74	1

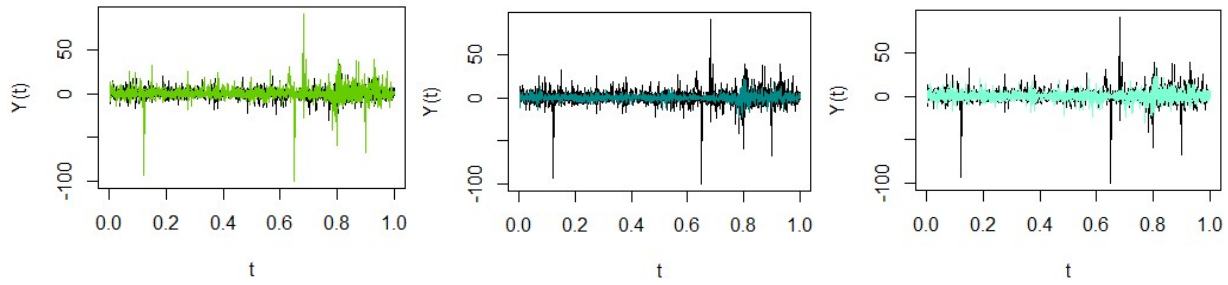
With the dendogram algorithm we obtain 7 clusters. The companies are distributed as the following: Cluster 1: 23, Cluster 2: 17, Cluster 3: 58, Cluster 4: 11, Cluster 5: 8, Cluster 6: 2, Cluster 7: 1. According to the part *Dendogram application and results*, we decide to not take into account those results and to focus on Mclust method and K-means as a comparator.

We apply the clustering algorithms a second time only on the hundred last dates this time. We want to see if the classification significantly change if we condense on the events related to the Covid-19 in France.

For the clusters by the k-means we obtain the exact same ones than previously: 40 for the Cac40 companies, Imerys (NK) a minearal company alone and the other companies.
By the model-based clustering we have also the 40 companies from the Cac40 and 2 companies have been changed into the third cluster.

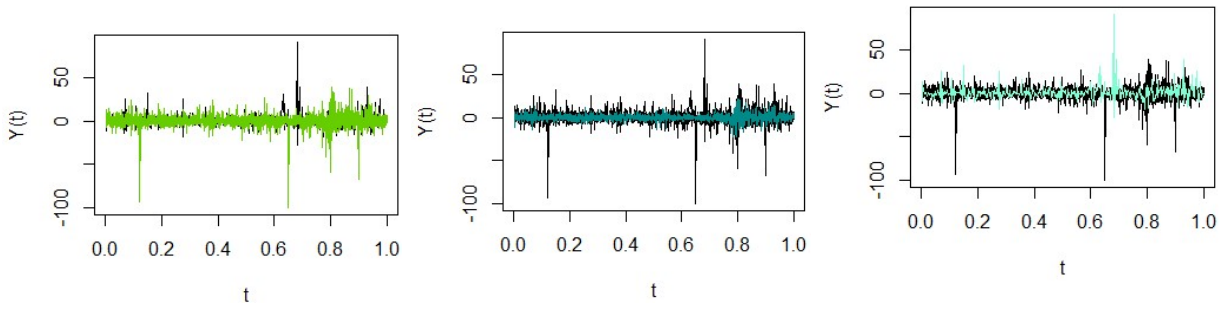
	Cluster 1	Cluster 2	Cluster 3
Mclust	**40**	**72**	**8**
K-means	40	79	1
In common	40	72	1

At the end, we obtain 115 companies with the same classification for both algorithm, 2 between the clusters 2 and 3 for mclust (*VK.PA* corresponding to Vallourec, a steel company and *NK* corresponding to Imerys, a company specialised in minerals). We can observe the clusters obtained by the two methods in the following graphs.



(a) Cluster 1 (b) Cluster 2 (c) Cluster 3

Figure 9: Stock market Dataset according to the clusters defined by mclust, in black the whole dataset, in colour the cluster highlighted.



(a) Cluster 1 (b) Cluster 2 (c) Cluster 3

Figure 10: Stock market Dataset according to the clusters defined by K-means, in black the whole dataset, in colour the cluster highlighted.

10.2 The interpretation of the results according to the Covid-19 events

In the cluster 1 we find all the french companies from the Cac40 without exeption for both methods, for the whole dataset and the 100 last days.

It seems that whatever the field of the company, the companies of the Cac40 behaved similarly face to the crisis. They had huge stock exchanges: positive or negative.

- On the one hand, these are the companies which have seen their activity drop sharply due to containment measures, the closure of points of sale, lower consumption and which will benefit from economic plans (Air France, Renault, Fnac Darty. ..).
- But also those who have benefited from the collateral effects of the crisis on consumption by having supported the population during the crisis (Carrefour, Danone ...).

In the next cluster we found Imerys (NK) a company specialised in minerals, Scor ($SCR.V$) a reinsurance company, CGG ($CGG.PA$) specialised in energy,DBV technologies ($DBV.PA$) a biopharmaceutical company, Erofins scientif(ERF) a testing labatories, Noeoen ($NEOEN.PA$) working in energy, and Vallourec ($VK.PA$) a steel company.

With this cluster we indentify two groups of companies.

- "Winning" companies face to the Covid-19 and the quarantine because of their health-related activities like Scor providing life reinsurance solutions to its clients, DBV technologies a biopharmaceutical firm or Erofins scientif providing testing and support services to pharmaceutical industries.
- Companies supporting in energy the population during this crisis and providing minerals.

The last cluster included all the other companies. The variations of stock returns are more reasonable than for companies in the other two clusters.

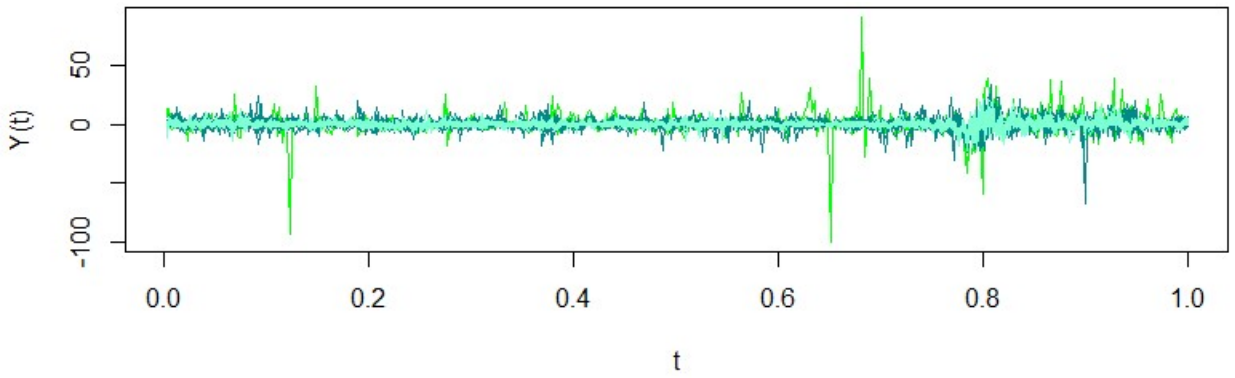


Figure 11: This graph represents the three clusters found by mclust. In turquoise we have the cluster 3 with most of the companies. In dark cyan we have the cluster 2 and in green the cluster 1 composed by the companies of the Cac40.

11 Conclusion

This project of clustering on stock market data faces two main challenges. The first one was to understand what is functional data and how works its analysis. By employing an extracting method on the features of the curves, we obtain smaller vectors. As a consequence, functional data analysis allows us to perform better results for the classification on stock market data.

Secondly we needed to find the proper way to simulate a comparable dataset to the stock market data. The simulated dataset helps to justify the selected parameters and clustering algorithm we keep for the real dataset. In this study, we apply on the stock market data the Model-based clustering based on parameterized finite Gaussian mixture models.

Thanks to this last challenge we are more confident in the clusters we obtained on the stock market data. The companies are split into three groups. The companies quoted in the Cac40 made more important positive or negative stock returns. Some companies stand out from the crowd by their activities for health, energy or minerals. The rest of the SBF120 companies similarly behave by medium exchanges.

To extend this study, it could be interesting in the future to directly use the stock market curves from every hours and not the stock returns by days to support or refute the clusters of companies obtained.

References

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

[Fama et al., 1969] Fama, E. F., Fisher, L., Jensen, M. C., and Roll, R. (1969). The adjustment of stock prices to new information. *International economic review*, 10(1):1–21.

[Ferraty and Vieu, 2006] Ferraty, F. and Vieu, P. (2006). *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media.

[Kassambara, 2017] Kassambara, A. (2017). *Practical guide to cluster analysis in R: Unsupervised machine learning*, volume 1. Sthda.

[McQuitty, 1960] McQuitty, L. L. (1960). Hierarchical linkage analysis for the isolation of types. *Educational and Psychological Measurement*, 20(1):55–67.

[Ramsay et al., 2009] Ramsay, J., Hooker, G., and Graves, S. (2009). *Functional data analysis with R and MATLAB*. Springer.

[Rousseeuw and Kaufman, 1990] Rousseeuw, P. J. and Kaufman, L. (1990). Finding groups in data. *Hoboken: Wiley Online Library*, 1.

List of Figures

12 Appendix

12.1 Description of the companies studied

12.1.1 CAC 40

The **CAC 40** is the share index that shows the levels of shares in 40 large companies among the 100 largest market caps on the Euronext Paris, called the Paris Bourse. CAC stands for "Compagnie des Agents de Change".

The following companies composed the CAC 40 in May 2020, whether the beggining of the internship.

Company	Sector	Ticker
Accor	hotels	AC.PA
Air Liquide	commodity chemicals	AI.PA
Airbus	aerospace	AIR.PA
ArcelorMittal	steel	MT.AS
Atos	IT services	ATO.PA
AXA	full line insurance	CS.PA
BNP Paribas	banks	BNP.PA
Bouygues	heavy construction	EN.PA
Capgemini	IT services	CAP.PA
Carrefour	food retailers and wholesalers	CA.PA
Crédit Agricole	banks	ACA.PA
Danone	food products	BN.PA
Dassault Systèmes	software	DSY.PA
Engie	gas and electric utility	ENGI.PA
Essilor	medical supplies	EL.PA
Hermès	clothing and accessories	RMS.PA
Kering	retail business	KER.PA
L'Oréal	personal products	OR.PA
Legrand	electrical components and equipment	LR.PA
LVMH	clothing and accessories	MC.PA
Michelin	tires	ML.PA
Orange	telecommunications	ORA.PA
Pernod Ricard	distillers and vintners	RI.PA
PSA Peugeot	automobiles	UG.PA
Publicis	media agencies	PUB.PA
Renault	automobiles	RNO.PA
Safran	aerospace and defence	SAF.PA
Saint-Gobain	building materials and fixtures	SGO.PA
Sanofi	pharmaceuticals	SAN.PA
Schneider Electric	electrical components and equipment	SU.PA
Société Générale	banks	GLE.PA

Sodexo	food services and facilities management	SW.PA
STMicroelectronics	Semiconductors	STM.PA
Thales	defense	HO.PA
Total	integrated oil and gas	FP.PA
Unibail-Rodamco-Westfield	real estate investment trusts	URW.AS
Veolia	water, waste, transport, energy	VIE.PA
Vinci	heavy construction	DG.PA
Vivendi	broadcasting and entertainment	VIV.PA
Worldline (fr)	IT services	WLN.PA

Sources: Euronext website at

12.1.2 SBF 120

The **SBF 120 (Société des Bourses Françaises 120 Index)** is a French stock market index. The index is based on the 120 most actively traded stocks listed in Paris. It includes all 60 stocks in the CAC 40 and CAC Next 20 indexes and 60 additional stocks listed on the Premier Marché and Second Marché under Euronext Paris. Infact this is the same than the values of the Cac40 and the 80 values of the SBF 80.

The CAC Next 20 is composed of the following companies.

Company	Sector	Ticker
Air France–KLM	airline	AF
Alstom	rail transportation	ALO
Arkema	specialty chemicals	AKE
Bureau Veritas	business support services	BVI
Edenred	financial administration	EDEN
EDF	electricity	EDF
Eiffage	construction	FGR
Faurecia	automotive parts	EO
Gecina	real estate	GFC
Getlink	rail transport	GET
Ingenico	electronic transactions	ING
Klépierre	real estate	LI
Natixis	banks	KN
Scor	reinsurance	SCR
SES	telecommunications	SESG.PA
Solvay	chemicals	SOLB
Suez	Environnement water	SEV
Teleperformance	business support services	TEP.PA
Ubisoft	video games	UBI
Valeo	automotive parts	FR

Sources: Euronext website at

Apart from the companies composing the CAC 40 and CAC Next 20, to obtain the SBF 120, we need to add the sixty following companies:

Company	Sector	Ticker
AKKA Technologies	digital technologies and industries support	AKA.PA
Albioma	energy	ABIO.PA
ALD	automotive location	ALD.PA
Alten	technology consulting and engineering	ATE.PA
Amundi	shareholding	AMUN
Aperam	steels	APAM
ArcelorMittal	steels	MT
Bic	pens,razors	BB.PA
bioMerieux	science	BIM
Bolloré	transport	BOL
Casino Guichard	supermarkets	CO.PA
CGG	energy	CGG.PA
CNP Assurances	insurance	CNP.PA
Coface	insurance	COFA.PA
Covivio	real estate	COV
Dassault aviation	aviation	AM.PA
DBV Technologies	biopharmaceutical	DBV.PA
Elior Group	commercial catering and foodservice	ELIOR
Elis	cleaning	ELIS.PA
Eramet	minerals	ERA
Eurazeo	investment	RF
Eurofins scientif	Testing laboratories	ERF
Euronext	European stock exchange	ENX.PA
Eutelsat Communica	satellite operator	ETL
Fnac Darty	distribution	FNAC.PA
Gaz Transport Techn	gas	GTT.PA
Genfit	biopharmaceutical	GNFT
Icade	real estate	ICAD
Iliad	Telecommunication	ILD.PA
Imerys	minerals	NK
IPSEN	biopharmaceutical	IPN
IPSOS	survey and opinion marketing	IPS
JC Decaux	advertissement	DEC
Korian	Homes for the elderly	KORI
FDJ	french money game	FDJ
Lagardere	production and distribution	MMB
M6	Television channel	MMT.PA

Maisons du monde	distribution of furnitures	MDM
Mercialys	shopping center management	MERCY
Neoen	energy	NEOEN.PA
Nexans	buldings, telecommunication, industry	NEX
Nexity	real estate	NXI
Orpea	eldery health and houses	ORP
Plastic omnium	industry	POM
Remy cointreau	alcohol	RCO
Rexel	electricity and renewable energy	RXL
Robertet	distribution aromatic products	RBT
Rubis	oils	RUI
Sartorius stedim	Bioprocess and labs	MSDHF
SEB	domestic equipment	SK
Soitec	industry	SOI
Sopra Steria Group	digital	SOP
SPIE	engineering	SPIE
Tarkett	floor coverings and sports surfaces	TKTT
Techni pfmc	energy	FTI
TF1	Television channel	TFI.PA
Trigano	manufacturing and distribution of vehicles and leisure equipment	TRI
Vallourec	steel	VK
Virbac SA	veterinary pharmaceutical	VIRP
Wendel	investment	MF

12.2 Functional data object in R

Our stocks return data is transform into functional data objects. This makes it easier to display curves and retrieve parameters. We fix $J = 390$ the number of days, $n = 120$ the number of companies. For the simulated dataset, J is fixed to 400 and n to 100.

Functional data objects are composed of three parameters:

rangeval: the limits of the intervall T, for our case it is $[0, 1]$.

argvals: the sequence of $t \in T$, for exemple if there is J=10 measures between 0 and 1 hour we will have $(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$.

data: $Y_i(t_j)$ the observation measured at every t. For a sample, data will be a matrix of size $n \times J$, n the number of curves.

12.3 Least Square smoothing in R

We recall that c_i is the vector of the features of length K and Ψ a matrix of size $n \times K$ that contains the values of $\psi_k(t)$ for each curve.

In R, the function *create.*nameofthebasis.*basis* from the package fda [Ramsay et al., 2009] allows us the create this matrix Ψ. Each of the parameter depends on the basis selected. Here we use a principal component basis. The basis of principal components is the most effective way of summarizing the information of X. The goal of this method is to optimize the amount of variance in the data explained. We need to determine the number K of basis functions we keep and *period* the periodicity or the length of J.

Then we have: $\hat{Y} = \Psi \times \hat{c}$ with

$$\hat{c} = (\Psi' \Psi)^{-1} \Psi' (Y - \mathbb{E}(X_t))$$

the solution of the minimization problem of the SSE. The expectation allows us to work on centered data with $\mathbb{E}(X_t) \approx \frac{1}{J} \sum_{t=1}^{J} X_t$.In R, we compute $\hat{c}$ with the commands:

crossprod: With one argument Ψ, this function returns $A = {}^t\Psi\Psi$. With two arguments Ψ and y, it returns $b = {}^t\Psi Y$.

solve: Knowing the two objects A and b obtained with the *crossprod*, the vector $\hat{c}$ is obtained by resolution of the system $A \times \hat{c} = b$, which can be done by calling the function solve with arguments A and b.

12.4 K-Means application and results

A common problem to clustering studies is the choice of the number of clusters. For our study, we tested three methods to determine how many clusters to keep: The Elbow method, The Average silhouette method and the Gap statistic method. [Kassambara, 2017]

Recall that, the basic idea behind partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation or total within-cluster sum of square (WSS) is minimized. The total WSS measures the compactness of the clustering and we want it to be as small as possible.

We kept the average silouhette method, it provides better results than the Gap statistic method and was easier to implement for 1000 repetitions than the Elbow method.

The **average silhouette** approach measures the quality of a clustering. That is, it determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering.

Average silhouette method computes the average silhouette of observations for different values of k. The optimal number of clusters k [8] is the one that maximize the average silhouette over a range of possible values for k [Rousseeuw and Kaufman, 1990].

With 1000 repetitions on the first simulated dataset, the mean of the accuracy obtained is 0.96 on the features extracted while on the simulated data, the matrix of size $n \times J$, the mean of the

[8] In this part, when we refer to k we talk about the optimal number of clusters choosen by the algorithm

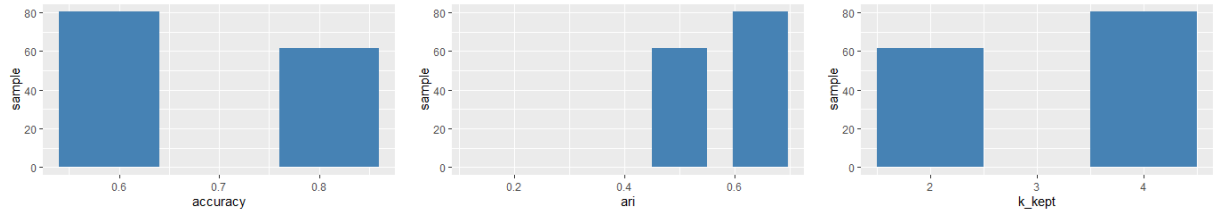
accuracy drops to 0.60. The mean of the ARI is 0.89 on the $c_{i,k}$ and drops to 0.59 for the all of the simulated data.

Therefore there is a real gain in performing clustering on the features of the curve, by reducing the dimensions we obtain better separation.

However after analysing more deeper the results obtained, we notice that the k-means method was not working for our dataset. The results follow a kind of pattern by switching to the same results endlessly. To be sure it was not an unfortunate coincidence, we re-run the repetition of simulations again 500 times. By lack of time we decrease the precision of the method, that is why the means of the indexes decrease. We obtain the following results. We remind that k kept is the optimal number of clusters kept by the algorithm.

Sample	Accuracy mean	ARI mean	Percentage of correct k kept
n=100 and balanced	0.56	0.69	0.19
n=100 and unbalanced	0.69	0.57	0
n=50 and balanced	0.46	0.63	25

The percentage of k kept is really small despite the correct accuracy because most of the time $k_{kept} = 4$ or $k_{kept} = 5$ and the last clusters are composed only by two or a single company. Let me introduce you some graphs to overview the situation with the kmeans algorithm. Here are the results for the unbalanced dataset of size 100 as far as it corresponds the closest to the stock market data.



(a) Accuracy results (b) ARI results (c) K Kept results

Figure 12: Those barplots point the repartition of the results for 250 repetitions of the sample of size 100 as a unbalanced dataset. Half the results are the same and switch every turn, that is why we didn't perform the loop until 500 and stop it to 250 repetitions.

For those reasons we finally take the decision to not perform the k-means on the stock market dataset.

12.5 Dendogram application and results

In R, the command to perform hierarchical clustering is *hclust*, it needs to have the distance defined before. For example the L_p distance with *metric.lp*. We compare different distances with the accuracy and keep the one minimizing the error.

For the simulated dataset, the distance kept is the Canberra distance and the method used is mcquitty [McQuitty, 1960].The Canberra distance d between vectors p and q in an n-dimensional real vector space is given as follows:

$$d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|}$$

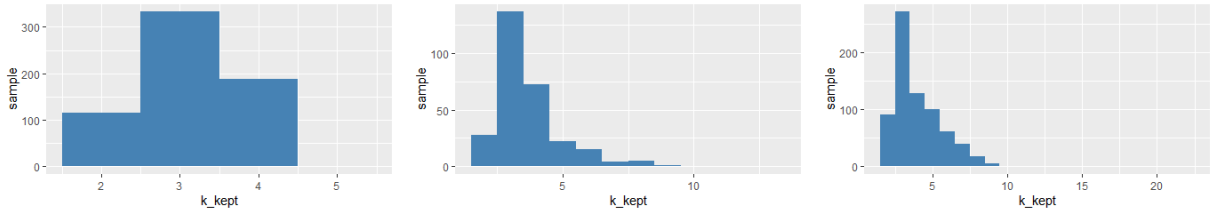
where
$\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$.

Complete linkage [McQuitty, 1960] merges groups based on the maximum distance between two objects in two groups. In other words, the distance between clusters A and B is defined as

$$d_C(A, B) = \max_{a \in A, b \in B} d(a, b)$$

.

With 1000 repetitions on the simulated dataset, the mean of the accuracy obtained is 0.61 and the ARI is 0.57 on the features extracted. These measures are low, we compare the different k, number of clusters kept by the method to decide if we keep it for our real data or not.



(a) Balanced sample size 100 (b) Unbalanced sample size 100 (c) Balanced sample size 50

Figure 13: Those barplots point the proportion of the k kept for 500 repetitions of the sample of size 100 as balanced and unbalanced dataset and of size 50 as balanced dataset. We observ that for the first case the results are not too bad. The number of clusters is closed to the real one (which is 3). For the unbalanced dataset or the small dataset even if most of the results are correct, giving 3 clusters, there is a risk to be completly mistaken with for instance more than 10 clusters for some simulations.

Given this error hazard and the indexes results far from perfect we won't risk this method for the stock market dataset, or only as a results comparator with another method.