École polytechnique fédérale de Lausanne

Semester project spring 2023

Bachelor in Mathematics

Explore representation and analysis of functional data

Author:
Salya Diallo

Supervisor:
Maria Laura Battagliola

Contents

1 Introduction

In our world, data is everywhere, whether it be from bio-medical research, environmental monitoring or finance, among other examples. Nowadays, it is important to know how to interpret these data in order to use them as best as possible.

One can see the data as a scalar-valued vector not depending on time, where each coordinate represent a data point. Functional data analysis offers a different perspective by considering data as functions (in general of the time) rather than as a finite set of observations. This can be useful when one considers a data set that evolves with time or when measurement are taken over a grid of time.

Essentially, functional data analysis focuses on capturing the essence of data through functions, as it is explained in [5]. By modeling the data as functions, one can study the smoothness, periodicity and fluctuations, ultimately gaining understanding of the underlying process generating our observations.

The following report is structured as follows. In the first section, mainly based on [5] and [9], we will study how to represent our data through a chosen basis function system, focusing on Fourier and B-spline basis. From this, we will be able to smooth our data through splines. In the second section, based on [9], we will look into functional principal component analysis, starting from principal component analysis for the multivariate case (that is, the data is a scalar-valued vector) and then extend it to functional data. Then we will study an example to illustrate the theory. In the third section, based on [5] and [6], we will explain how to do regression on multivariate data and look into different possibilities of functional regression, namely scalar-on-function (scalar response, functional observation) and function-on-function (functional response and observation) linear models. Then we will look into an example of functional regression on an actual data set. Finally in the last section, we will apply the theory summarized before to an actual data set. In particular, we will base our work on [4], in order to study the financial volatility of 30 different stocks during a certain time-frame. In order to do so, we will introduce a few financial terms and then summarize the used model to study the volatility, which we will then apply to our data set and study the results.

2 Background and methodology

2.1 What does functional data analysis deal with?

Functional data analysis deals with data that are functions, which is very different from multivariate data. In this project, we will consider what we call FIRST GENERATION FUNCTIONAL DATA. That is, we look at a sample of independent random functions $X_1(t), \ldots, X_n(t)$ for $t \in I$ where $I = [0, T] \subseteq \mathbb{R}$ is a compact interval. In general, these functions will be supposed to be smooth, i.e., continuous with a compact support and square integrability.

The two main features of functional data are the following:

1) the functions are assumed to be smooth, that is, they have one or more derivatives;

2) they come in pairs $(t, X(t))$, so you cannot mix the values of t and obtain the same measurements.

Thus, functional data differ from multivariate data because there is no exchangeability along the functional coordinates.

For the theoretical part of the report, we will use the data of the Canadian Weather present in the *fda* package in R in order to illustrate what we study.

When we analyze data, it may be useful to look at the derivatives. In Figure 1, we see an example of data where t represents time and $X(t)$ is the weather at a certain station in Canada at time t. One may be interested in other quantities, like for example velocity (here it would give us how fast the temperature changes over a period of time). This is one of the reason why it is useful to look at the data as functional and not only scalar or vectorial.

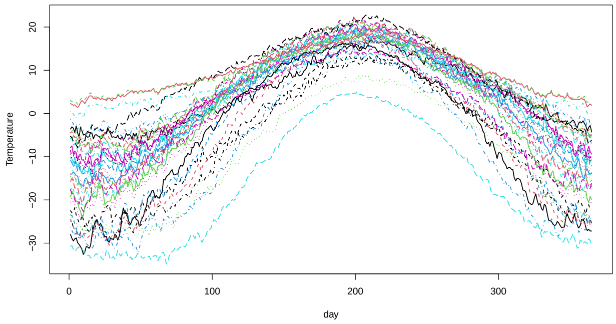


Figure 1: Average daily temperature in Canada at different stations (for each curve) for each day of the year

Moreover, functional data are infinitely dimensional and this is why we will see that dimension reduction is a great tool of functional data analysis, mainly through basis functions, developed in the next sections.

In most cases, we can only collect data discretely and not continuously over time, hence we cannot observe exactly what is happening. Measurement errors can be seen as noise and will disrupt the smoothness of the observation, as we can see in Figure 1, which we will study in more details in the following section.

2.2 Representation of functions with basis functions

Definition 2.2.1 (BASIS FUNCTION SYSTEM). A basis function system is a sequence of known functions $\{\Phi_k\}_k$ that are mutually linearly independent and that spans the set of functions, i.e., for any function f we can approximate it by $f(t) = \sum_{k=1}^{K} A_k \Phi_k$ for some scalars $A_1, \cdots, A_K$ and a sufficiently large number K of basis functions.

The main reason for using basis functions is for dimensional reduction since we consider a sufficiently large and finite number of basis functions to approximate the data. Moreover, if we know the derivatives of the functions of the basis, derivating any function will be easier.

There are two main basis functions that we will study here and are the most commonly used: Fourier and B-spline. There is plenty other basis systems, as wavelets for example.

2.2.1 Fourier basis

Let us define the orthogonal and periodic Fourier basis as follows:

$$x(t) = a_0 \cdot \Phi_0(t) + a_1 \cdot \Phi_1(t) + a_2 \cdot \Phi_2(t) + a_3 \cdot \Phi_3(t) + \cdots \tag{1}$$

where $\Phi_0(t) = 1$, and for any $r \in \mathbb{N}$: $\Phi_{2r+1}(t) = \sin(r\omega t)$, $\Phi_{2r+2}(t) = \cos(r\omega t)$ and ω will determine the period T of the basis by $T = \frac{2\pi}{\omega}$. Here, $\{a_r\}_{r=0}^{\infty}$ are real numbers.

This basis is mostly used for uniformly smooth functions and its periodicity should appear in a way in the data we want to study and approximate.

In Figure 2, we show an example of Fourier basis composed of six functions on the interval $[0, 1]$. We can see that all the functions are periodic but with a different period, that is the value of ω changed.

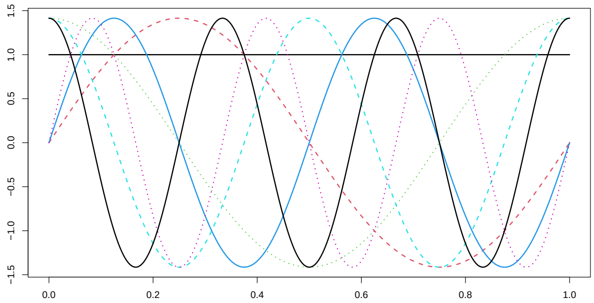


Figure 2: Fourier basis with K = 6 on [0,1]

2.2.2 Splines

Definition 2.2.2. (SPLINES AND KNOTS) Suppose that we are working on an interval $I = [\tau_0, \tau_L]$ and that we divide it into L sub-intervals $\{I_k = [\tau_k, \tau_{k+1}]\}_{k=0}^{L-1}$. Over each sub-intervals I_k, a spline S_k is a polynomial of specified order $m \in \mathbb{N}\backslash\{0\}$, where $m = \textit{degree of the polynomial} + 1$. The points $\{\tau_k\}_{k=0}^{L}$ are called knots. At the interior knot τ_k, for all $k = 1, \ldots, L-1$, the polynomials S_{k-1} and S_k have the same derivatives of order 0 up to order $m-2$.

Proposition 2.1. *The space of spline functions $\mathcal{S}_m^{\tau}$ of order m and knots sequence τ is a linear space.*

The B-spline basis $\mathcal{B} = \{B_k\}_k$ is the most generic basis and is mostly used for dimension reduction. This basis has to respect the following properties:

- each B_k has to be a spline defined by an order m and a sequence τ of knots;
- any linear combinations of the elements of $\mathcal{B}$ is a spline;
- any spline function of order m and knots sequence τ can be defined by a linear combination of the B_k.

In Figure 3, we show an example of B-spline basis composed of six functions with equidistant knots $\{\tau_k\}_{k=1}^{6}$ and of degree $m = 3$ on the interval $[0, 1]$, with $\tau_1 = 0$ and $\tau_6 = 1$.

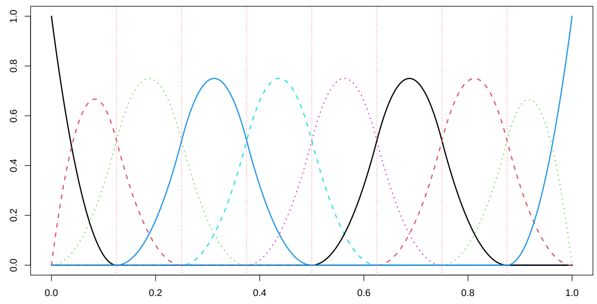


Figure 3: B-spline basis of order 3 on [0,1] with K = 6 and 3 interior knots equally spaced

One question remains: How to decide the placements of the knots on our interval I. This will depend on the structure of the data. If it is equally spaced, we can choose to take equally spaced knots, that is for all $k = 1, \dots, L$ we have $\tau_k - \tau_{k-1} = h_L$ with $h_L = \frac{\tau_L - \tau_0}{L+1}$. Otherwise, one possibility is to put a knot at every data point y_j.

There exists algorithms to find the best possible placements for the knots depending on the data and what we want to analyze. For example, we can start with a dense set of knots, and then we remove unnecessary ones by using an algorithm similar to variable selection used in linear regression on multivariate data, such as backward elimination. The goal is to have a high number of knots / number of splines enough to have smooth approximations, but without exceeding in order to avoid overfitting and hence wiggliness. We will not develop these algorithms here.

2.3 Smoothing procedures

Let $X(\cdot)$ be the true function with $X \in L^2(I)$ for I a closed interval of $\mathbb{R}$. Suppose that we observe $y_j = X(t_j) + \epsilon_j$ for a dense discretization of I with equidistant points $\mathcal{G} = \{t_1, \dots, t_J\} \subseteq I$ and $\epsilon_j \sim \mathcal{N}(0, \sigma^2)$ white noises, independent of the true function $X(\cdot)$. The goal is to smooth y_j in order to obtain a new function $\hat{X}(\cdot)$ that is a good fit to the data $X(\cdot)$.

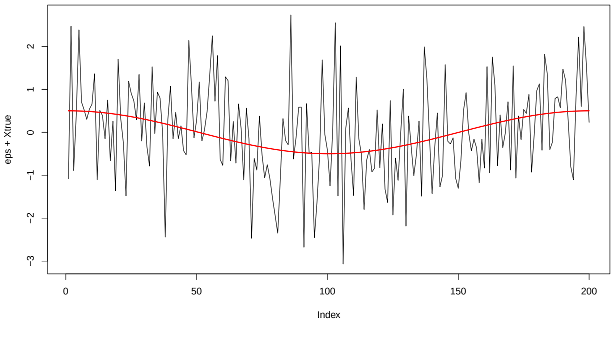


Figure 4: The bold red line represents the true value of $X(t) = \frac{1}{2}\cos(2\pi t)$ for $t \in [0, 1]$ and the black one represents y_j

For example, in Figure 4 we see the case of $X(t) = \dfrac{1}{2}\cos(2\pi t)$ for $t \in [0, 1]$, the time grid $\mathcal{G}$ with $J = 200$ equidistant points and $\epsilon_j \sim \mathcal{N}(0, 1)$. We see that $\{y_j\}_j$ in black is hard to analyse and violates the assumption of smoothness functional data analysis is funded on, whereas $X(\cdot)$, a smooth function, is much simpler.

There is different methods to smooth a function depending on its nature (linearity example). Here, we will develop the spline smoothing (also called roughness penalty method with splines).

2.3.1 Spline smoothing

Suppose that we observe $\{y_j\}_{j=1}^n$ but that the true data is given by the smooth function $X(\cdot) \in L^2(I)$. We can write $y_j = \beta_j X(t_j) + \epsilon_j$ for independent and identically distributed (iid) white noises $\epsilon_j \sim \mathcal{N}(0, \sigma^2)$, independent of $X(\cdot)$, with the time grid $G_I = \{t_1, \ldots, t_n\}$.

Let us define a useful type of spline for smoothing.

Definition 2.3.1. (natural cubic spline) A natural cubic spline with knots $\{t_j\}_j$ satisfy the following properties:

- it is a piece-wise polynomials of degree three,
- with pieces defined at the knots,
- with two continuous derivatives at the knots,
- and linear outside the data boundary.

The main goal is to find $\hat{X}(\cdot) \in L^2(I)$ that minimizes the following:

$$\underbrace{\sum_{j=1}^{n}(y_j - \hat{X}(t_j))^2}_{\text{Fit Penalty}} + \lambda \underbrace{\int_I (g''(t))^2 dt}_{\text{Roughness Penalty}} \tag{2}$$

The parameter λ is here to balance fidelity to the data (fit penalty) and smoothness (roughness penalty) of $\hat{X}(\cdot)$. It is called a smoothing parameter. Hence, equation (2) is a loss function since it is measuring the discrepancy between observed and estimated data, which is something we wish to minimize.

Proposition 2.2. *The minimization of equation* (2) *has a unique explicit solution: natural cubic spline with knots* $\{t_j\}_{j=1}^n$.

One of our issue about splines is here resolved directly: the placement of the knots. Indeed, in this case they correspond directly to the data points. Nevertheless, having a knot in every point of the grid give us something that is excessively wiggly and over-fitted.

We can represent splines by using natural cubic spline basis functions $\{B_j(\cdot)\}_j$ as $s(t) = \sum_{j=1}^n \gamma_j B_j(t)$. Define the matrix $B = (B_{ij})_{i,j=1}^n$ with $B_{ij} = B_j(t_i)$ and the vector $\gamma = (\gamma_1, \ldots, \gamma_n)^\top \in \mathbb{R}^n$. Let $\Omega_{ij} = \int_I B_i''(t) B_j''(t) dt$.

Equation (2) can be re-written as:

$$(y - B\gamma)^\top (y - B\gamma) + \lambda \gamma^\top \Omega \gamma \tag{3}$$

By differentiating and equating to zero (3), we get that the vector $\hat{\gamma}$ that will minimize the penalized likelihood is:

$$(B^\top B + \lambda\Omega)\hat{\gamma} = B^\top y \implies \hat{\gamma} = (B^\top B + \lambda\Omega)^{-1} B^\top y \tag{4}$$

We say that $\mathrm{S}(\lambda) = B(B^\top B + \lambda\Omega)^{-1}B^\top$ is a smoothing matrix with $\mathrm{trace}(\mathrm{S}(\lambda)) = \sum_{j=1}^n \dfrac{1}{1 + \lambda \times \eta_j}$, where $\{\eta_j\}_{j=1}^n$ are the eigenvalues of $K = (B^\top B)^{-1/2}\Omega(B^\top B)^{-1/2}$.

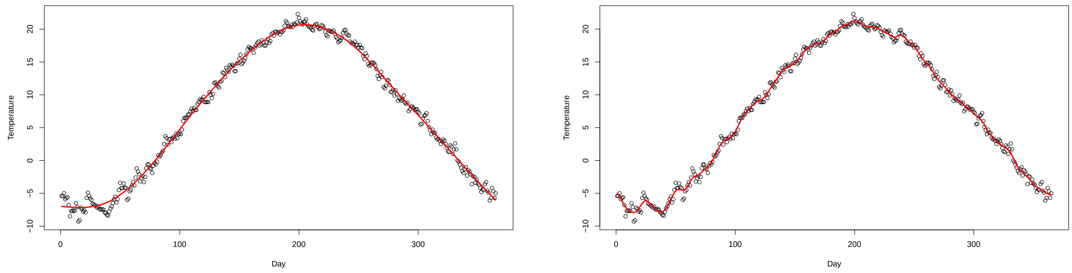
Definition 2.3.2. (degrees of freedom) The degrees of freedom of smoother is given by $\mathrm{df}(\lambda) := \mathrm{trace}(\mathrm{S}(\lambda))$.

For example, in Figure 5, we see the plots of the weather in Toronto (in black) that has been splines smoothed with $\mathrm{df}(\lambda) = 10$ in Figure 5a and $\mathrm{df}(\lambda) = 50$ in Figure 5b. We can remark that the estimated smoothed line in the second plot is more adherent to the characteristics of the data, especially at the border of the time interval, because the degrees of freedom are higher. In addition, Figure 5a is smoother than Figure 5b.

This comes from the following relation between degrees of freedom, number of internal knots k and degree of the splines m:

$$\mathrm{df}(\lambda) = k + m.$$

Thus, if $\mathrm{df}(\lambda) = 50$ and $m = 3$ (cubic), we have $k = 50 - 3 = 47$ interior knots, while if $\mathrm{df}(\lambda) = 10$ we have only 7 interior knots.



(a) Spline smoothing with 10 degrees of freedom

(b) Spline smoothing with 50 degrees of freedom

Figure 5: Spline smoothing of the weather data taken in Toronto with different degrees of freedom

Consider the fitted value $\hat{y} = \mathrm{S}(\lambda)y$ and define $\mathrm{SSE} = \sum_{j=1}^{n}(y_j - \hat{y}_j)^2$. To choose the right value of λ, we can use one of the following criteria:

- Cross-validation sum of squares: $\mathrm{CV} = \sum_{j=1}^{n}\left(\dfrac{y_j - \hat{y}_j}{1 - \mathrm{S}_{jj}(\lambda)}\right)^2$;
- Generalized cross-validation sum of squares: $\mathrm{GCV} = \sum_{j=1}^{n}\left(\dfrac{y_j - \hat{y}_j}{1 - \mathrm{df}(\lambda)/n}\right)^2 = \left(\dfrac{n}{n - \mathrm{df}(\lambda)}\right)\left(\dfrac{\mathrm{SSE}}{n - \mathrm{df}(\lambda)}\right)$.

To find the best value of λ, the idea is then to minimize one of the criterion stated above.

On one hand, by using CV, we need to perform n regressions, and so n smoothing (by estimating the vector γ_{-j} for $j = 1, \ldots, n$, where γ_{-j} is obtained by removing the j-th value of the initial vector γ). This method is thus computationally intense and hard to do for large sample size.

On the other hand, GCV do not require to re-smooth our model n times but only once and we avoid under-smoothing, to the opposite of CV. Hence, using GCV is more stable and reliable in practice.

We see in Figure 6 that the smoothing is even more fitted to the data when using GCV than with 50 degrees of freedom. The value of λ and of degrees of freedom computed in R are, respectively, $\lambda = 2.67e - 20$ and $\mathrm{df}(\lambda) = 111$. This illustrates the fact that this criterion help us choose the right parameters for smoothing.

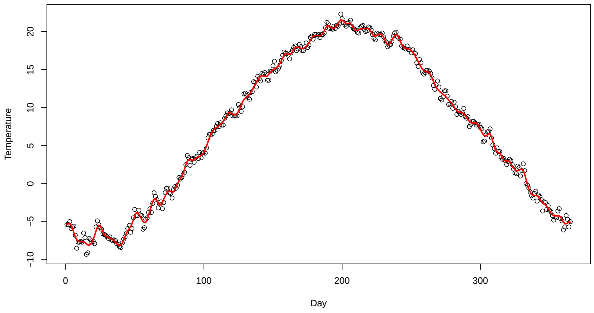


Figure 6: Splines smoothing of the weather taken in Toronto using GCV

3 Functional Principal Component Analysis

Principal components analysis (PCA) is a useful dimensional reduction tool for multivariate statistics and it has been, in time, extended to functional data.

3.1 Framework

Let us work with first generation functional data: consider a sample of smooth and independent random functions $X_1(t), \dots, X_n(t)$ for $t \in I = [0, T] \subset \mathbb{R}$. We then look at the $L^2(I)$ stochastic process $X(\cdot)$ following from the functional data.

Definition 3.1.1. (MEAN AND COVARIANCE FUNCTIONS) We define the mean function μ by $\mu(t) = \mathbb{E}[X(t)]$ and the covariance function Σ by $\Sigma(s,t) = \mathrm{Cov}(X(s), X(t)) = \mathbb{E}[(X(s) - \mu(s))(X(t) - \mu(t))]$.

Let us consider time grids $\mathcal{T}_i = \{t_{i1}, \dots, t_{in_i}\}$, for $i = 1, \dots, n$ and some $n_i \in \mathbb{N}\backslash\{0\}$, for $X_i = X_i(\cdot)$ and the corresponding observations denoted by $\mathbf{X}_i = (X_{i1}, \dots, X_{in_i})$ with $X_{ij} = X_i(t_{ij})$. We can now consider the following model: $Y_{ij} = X_{ij} + \epsilon_{ij}$, for any i, j, where ϵ_{ij} is a random white noise.

For simplicity, we assume for the next parts that for any i, $n_i := p \in \mathbb{N}\backslash\{0\}$.

3.2 Principal components analysis for multivariate data

Let us consider $Y_j = \beta^\top \mathbf{X}_j$ where $\beta = (\beta_1, \dots, \beta_p)^\top$ is the weight vector and $\mathbf{X}_j = (X_{j1}, \dots, X_{jp})^\top$ is the observed vector.

A principal components analysis consists of solving the following problem:

(a) Find a weight vector $\beta_1 = (\beta_{11}, \dots, \beta_{1p})^\top$ that has a unit norm, that is $\sum_{k=1}^p \beta_{1k}^2 = 1$, and that will maximizes the mean square $\dfrac{1}{n} \sum_{k=1}^p Y_{1k}^2$.

(b) For $j = 2, \dots, n$, resolve (a) (replace 1 by j to find weight vector β_j) and add the condition that all the weight vectors are orthogonal, that is $\sum_{k=1}^p \beta_{ik}\beta_{jk} = 0$ for any $j \neq i$.

Remark that the weight vectors are not unique because changing the signs, i.e., considering $-\beta_j$ instead of β_j, does not change the value of the mean squares.

Definition 3.2.1. (PRINCIPAL COMPONENTS) We call principal components all the values Y_{jk} from the PCA resolution.

Before doing PCA, we usually center at the mean each variable, that is, we subtract the mean. By doing this, maximizing the mean square is equivalent to maximizing the sample variance. In this case, the principal components corresponds to the eigenvectors of the covariance matrix of the sample.

Most of the data that we want to analyze are functional rather than multivariate in order to reflect better the smooth nature of the data. Thus we have interest in generalizing PCA to functional data.

3.3 Principal components analysis for functional data

In this section, we suppose that we centered our stochastic process at the mean function, that is we consider $X_i(\cdot) - \mu(\cdot)$ instead of $X_i(\cdot)$.

The idea is now to extend PCA to functional data. This means that now, instead of looking at sums for norms and scalar products, we will be looking at integrals. For example, the norm of the weight vectors $\{\beta_j\}_j$ being equal to one is instead written as $\int_I \beta_j(t)^2 dt = 1$ with $\{\beta_j(\cdot)\}_j$ the weight functions.

In order to do so, let us first give a definition and state an important result.

Definition 3.3.1. (Kernel) A kernel $K : I \times I \to \mathbb{R}$ is a symmetric and positive semi-definite function, that is, $\forall (s,t) \in I \times I$, $K(s,t) = K(t,s)$ and, $\forall s_1, \ldots, s_n \in I, \forall a_1, \ldots, a_n \in \mathbb{R}$, we have $\sum_{i=1}^{n} \sum_{j=1}^{n} K(s_i, s_j) a_i a_j \geq 0$. We can associate to K a linear operator L_K on functions defined by $[L_K g](x) = \int_I K(x,s) g(s) ds$.

Remark that the covariance function $\Sigma(\cdot, \cdot)$ defined in our framework is a kernel. Indeed, since it is a covariance it is positive semi-definite and obviously $\text{Cov}(X(s), X(t)) = \text{Cov}(X(t), X(s))$ thus it is also symmetric.

We define the linear operator of the covariance function $\Sigma(\cdot, \cdot)$ by $\Sigma(g)(t) = \int_I \Sigma(s,t) g(s) ds$ for any smooth function $g \in L^2(I)$.

We recall that a basis is orthonormal if each element is of norm equal to one and they are all mutually orthogonal.

Proposition 3.1. *Suppose that K is a continuous kernel. Then there is an orthonormal basis $\{\varphi_j\}_j \subseteq L^2(I)$ consisting of eigenfunctions of L_K, the linear operator corresponding to K, for which the corresponding eigenvalues $\{\lambda_j\}_j$ are non-negative, that is, they are greater or equal than zero. Thus, K has the representation $K(s,t) = \sum_{j=1}^{\infty} \lambda_j \varphi_j(s) \varphi_j(t)$, where the convergence holds uniformly for $s, t \in I$.*

This proposition implies that we can write the following for any $s, t \in I$:

$$\Sigma(s,t) = \sum_{k=1}^{\infty} \lambda_k \varphi_k(s) \varphi_k(t) \tag{5}$$

where $\{\varphi_k(\cdot)\}_k$ are the eigenfunctions of $\Sigma(\cdot)(\cdot)$, the linear operator of the covariance function, corresponding to non-negative eigenvalues $\{\lambda_k\}_k$ where the λ_k are in descending order, i.e., $\{\varphi_k(\cdot)\}_{k \geq 1}$ so that $\int_I \Sigma(s,t) \varphi_k(s) ds = \lambda_k \varphi_k(t)$ for any $t \in I$, $k \geq 1$ with $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ the corresponding eigenvalues.

We can write the Karhunen-Loève FPCA expansion:

$$X_i(t) = \sum_{k=1}^{\infty} A_{ik} \varphi_k(t), \text{ for any } t \in I \tag{6}$$

where $A_{ik} = \int_I X_i(t) \varphi_k(t) dt$ are called the functional principal components, or scores, of X_i.

The purpose of FPCA is to reduce the dimension, this is why we do the following. Since the space defined by eigenfunctions of a kernel generates the space of function and by the orthogonality of eigenfunctions (that is, $\int_I \varphi_k(t) \varphi_s(t) dt = 0$ for any $k \neq s$), $\{\varphi_k\}_k$ is an orthogonal basis functions system. Thus we can find K big enough such that we have the following approximation:

$$X_i(t) = \sum_{k=1}^{K} A_{ik} \varphi_k(t) \tag{7}$$

This means that the information contained in X_i is mainly contained in the vector $A_K = (A_{i1}, \ldots, A_{iK})$. For a fixed large enough K, functional principal components (FPC) expansion will explains most of the variation in X in the L^2 sense.

Remark that when choosing the value of K, there is a bias-variance trade-off (for bigger K the bias gets smaller and the variance gets bigger). Indeed, when K increases there is a smaller approximation error (thus bias decreases) but we have to approximate more components so we add random error (thus variance increases). Hence, the idea is to choose K so as to optimally increase variance / decrease bias.

Definition 3.3.2. (mode of variation) The k-th mode of variation is the set of functions $\mu(t) \pm \alpha \sqrt{\lambda_k} \varphi_k(t)$, with $\alpha \in [-A, A]$ for some A (in general, $A = 2$ or 3), that are viewed at the same time over the range of α.

Modes of variation permit to visualize and describe the variation pattern, in our functional data, that is contributed by each of the eigenfunction $\varphi_k(\cdot)$, $k \in \mathbb{N}$.

Definition 3.3.3. (percentage of variability) The percentage of variability of the i-th principal component is given by $\dfrac{\lambda_i}{\sum_j \lambda_j} \times 100$.

This value tells us how much of the total variance is contributed by the i-th component.

3.4 Example: Canadian weather

Suppose that the weather is taken in different stations of Canada throughout the day and that, each day, the average temperature is computed.

In Figure 7, the solid curve represents the overall mean temperature and the dashed curves represent the two modes of variation for the corresponding eigenfunction. We see that the first two components have the more effects on the data.

The first component corresponds to a shift of the temperature which is not constant through the year. We see that it is hotter in summer than in winter. The first mode of variation shows that this is a trend shared by all stations, but some of them are overall warmer or colder than the mean throughout all year.

The second principal component tells us that some Canadian stations are colder in winter and hotter in summer than the mean, and vice-versa.

The third principal component corresponds to a time shift (the shape of the solid curve globally stays the same but is translated horizontally) which is not constant through the year. We remark that the impact of the shift on the variation is way less important for this component than the first.

The fourth component does not change much of our original curve. Hence the fourth eigenfunction does not contribute considerably to the variation.

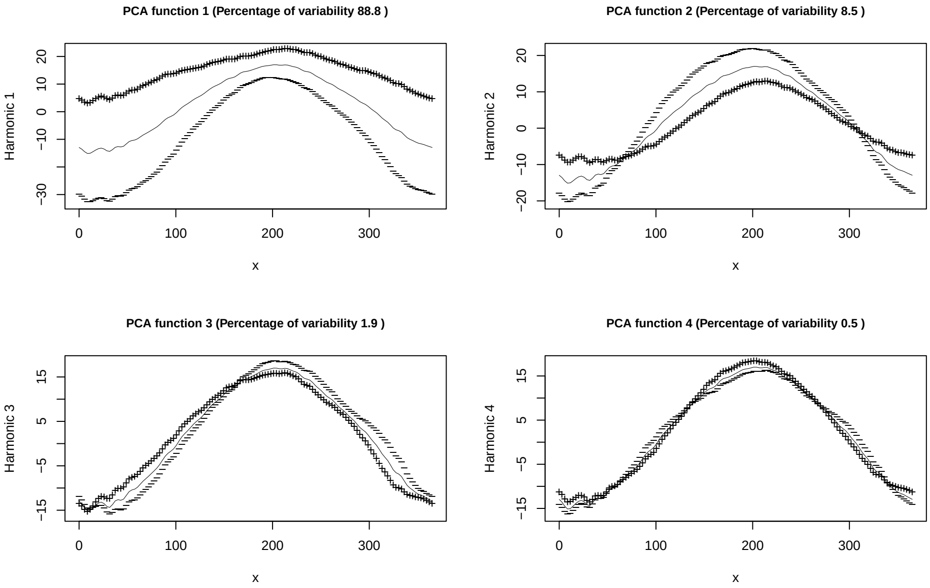


Figure 7: The overall mean temperature curve and the effect of adding (+) and subtracting (-) a multiple of each principal components curve

4 Functional regression

4.1 Multivariate linear model

In classical statistics, the basic linear model is given by:

$$Y = \beta \mathbf{X} + \epsilon \tag{8}$$

where Y is the response vector of size $n \in \mathbb{N}$, $\mathbf{X} \in \mathbb{R}^{n \times p}$ $(p \leq n)$ is the design matrix of our model with columns $\{X_j\}_{j=1}^{n}$, it defines a linear transformation. We say that there is n observations and p explanatories. The vector ϵ of length n is what we call the measurement errors or the random noise. We suppose that, $\forall i \neq j$, ϵ_j and ϵ_i are independent and identically distributed. In general, ϵ follows the distribution $\mathcal{N}_n(0, \sigma^2 I)$, in other words it is supposed to be white noise. The vector β of length p is what we want to estimate.

When the design matrix $\mathbf{X}$ is of full column-rank p, we can use least squares estimation on β. That is, compute $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top Y$ the maximum likelihood estimator (MLE) of β. We call $\hat{\beta}$ the LEAST SQUARE ESTIMATOR. From this, we can compute the FITTED VALUE: $\hat{y} = \mathbf{X}\hat{\beta}$.

To estimate σ^2, we have two possibilities: $\hat{\sigma}^2 = \frac{1}{n}(Y - \mathbf{X}\hat{\beta})^\top (Y - \mathbf{X}\hat{\beta})$ or $S^2 = \frac{1}{n-p}(Y - \mathbf{X}\hat{\beta})^\top (Y - \mathbf{X}\hat{\beta})$. The advantage of the latter is that it is unbiased, that is $\mathbb{E}[S^2] = \sigma^2$.

The goal is to generalize this model to functional data to be able to analyse more information and situations. There exists different functional linear models:

(i) a functional response and non-functional independent variable,

(ii) a scalar response and a functional independent variable, or

(iii) a functional response and a functional independent variable.

We will focus here on the last two cases.

4.2 Scalar-on-function linear model

Let us first consider scalar-on-function model, i.e., case (ii). We can write this model as:

$$\mathbb{E}[Y] = \beta_0 + \int_I \beta(s) X(s) ds \tag{9}$$

where the data is given by $\{Y, X(s)\}_{s \in I}$, with $\{X(\cdot)\}$ the functional independent variable, and Y as in the classical model (8).

One can express the functions $X(\cdot)$ and $\beta(\cdot)$ in an orthonormal basis functions $\{\varphi_k(\cdot)\}_k$. The most common choice, usable for any data, is the eigenfunction basis, as in the FPCA. Writing the corresponding Karhunen and Loève expansions as in (6), for $K \geq 1$ large enough, we have $X(s) = \sum_{k=1}^{K} A_k \varphi_k(s)$ and $\beta(s) = \sum_{k=1}^{K} B_k \varphi_k(s)$, using the same basis.

With these equalities, one can re-write (9) as:

$$\mathbb{E}[Y] = \beta_0 + \sum_{k=1}^{K} B_k A_k \tag{10}$$

by orthonormality of the basis functions. This is the same as resolving a basic linear model, as in (8).

The goal is to choose K large enough so that it will not imply any important loss of information and also control the shape and smoothness of $\beta(\cdot)$. One can choose K, for example, with the CV criterion.

One may choose another basis functions $\{\psi_k(\cdot)\}_k$ that is not necessarily orthonormal, such as B-splines or Fourier, and write $\beta(s) = \sum_{k=1}^{K} B_k \psi_k(s) = \psi(s)^\top B$ with $B = (B_1, \ldots, B_K)^\top, \psi(s) = (\psi_1(s), \ldots, \psi_K(s))^\top \in \mathbb{R}^K$. Then, one can solve the following penalized equation:

$$(\hat{\beta_0}, \hat{B}) = \arg\min \left(\sum_{k=1}^{K} (Y_j - \beta_0 - \int_I X(s)(\psi(s)^\top B) ds)^2 + \lambda P(B) \right) \tag{11}$$

for some value $\lambda > 0$ and some penalty function $P(\cdot)$. Depending on the basis function that has been chosen, the penalty function will be different in order to be optimal.

For example, for spline approach we may take $P(B) = B^\top MB$ where M is the $K \times K$ matrix given by $M_{ij} = \int_I \psi_i(s)\psi_j(s)ds$ for any $i, j \in \{1, \dots, K\}$.

4.3 Function-on-function linear model

For this section, we assume without loss of generality that $\mathbb{E}[X(t)] = 0 \forall t$ (otherwise we consider $X(\cdot) - \mu_X(\cdot)$ instead of $X(\cdot)$).

The function-on-function model, i.e., case (iii), can be written as:

$$\mathbb{E}[Y(t)] = \beta_0(t) + \int_{I_X} \beta(t, s)X(s)ds \text{ , for any } t \in I_Y \tag{12}$$

Here, I_X and I_Y are the domains of definition of the functions $X(\cdot)$ and $Y(\cdot)$ respectively. In addition, $\beta(\cdot, \cdot)$, which has $I_Y \times I_X$ as domain of definition, is a non-parametric smooth function.

When the domains of definitions are the same, that is $I_X = I_Y$, it is often assumed that $\beta(t, s) = 0$ whenever $t < s$. In other words, $Y(\cdot)$ is only affected by the past of $X(\cdot)$.

The methodology to solve this model is similar to the previous case, i.e., scalar-on-function model (9).

Definition 4.3.1. (AUTO-COVARIANCE AND CROSS-COVARIANCE FUNCTIONS) We define the auto-covariance functions of $X(\cdot)$ and $Y(\cdot)$ by $\Sigma_{XX}(s_1, s_2) = \text{Cov}(X(s_1), X(s_2))$ for any $s_1, s_2 \in I_X$, and $\Sigma_{YY}(t_1, t_2) = \text{Cov}(Y(t_1), Y(t_2))$ for any $t_1, t_2 \in I_Y$, respectively.

Similarly, we define the cross-covariance function of $X(\cdot)$ with $Y(\cdot)$ by $\Sigma_{XY}(s, t) = \text{Cov}(X(s), Y(t)) = \Sigma_{YX}(t, s)$ for any $s \in I_X$ and $t \in I_Y$.

One can write, for $K_X, K_Y \geq 1$ large enough and any $t \in I_Y$, $s \in I_X$:

$$X(s) = \sum_{m=1}^{K_X} A_k \psi_m(s) \tag{13}$$

and

$$\beta(t, s) = \sum_{k=1}^{K_Y} \sum_{m=1}^{K_X} B_{km} \varphi_k(t) \psi_m(s) \tag{14}$$

where $\{\varphi_k(\cdot)\}_k$ and $\{\psi_m(\cdot)\}_m$ are the eigenfunctions of $\Sigma_Y(f)(s) = \int_{I_Y} \Sigma_{YY}(s, t)f(t)dt$ and $\Sigma_X(g)(s) = \int_{I_X} \Sigma_{XX}(s, t)g(t)dt$, respectively.

Similarly, we can expand the intercept function $\beta_0(\cdot)$ as $\beta_0(t) = \sum_{k=1}^{K_Y} C_k \varphi_k(t)$, for any $t \in I_Y$.

To regularize the fit, one can restrict the basis. Indeed, one can truncate the basis $\{\psi_k\}_k$ for the covariates in order to avoid over-fitting, whereas truncating the other basis $\{\varphi_k\}_k$ ensure smoothness of the prediction. Thus, the goal is to choose K_X and K_Y wisely in order to ensure smoothness and avoid over-fitting as best as possible, using CV criterion for example.

Remark that here our basis are known, only our coefficients $\{A_m\}_m$, $\{B_{km}\}_{k,m}$ and $\{C_k\}_k$ are unknown and thus we need to estimate them using usual regression seen with the model (8).

In the end, using the orthonormality of our basis $\{\psi_m(\cdot)\}_m$, we get the following equality:

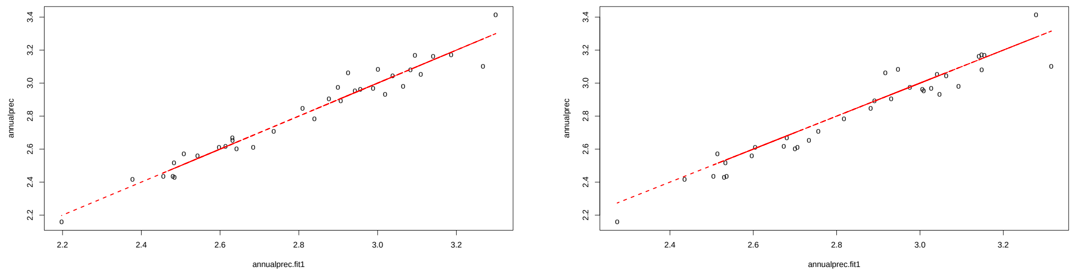
$$\begin{aligned}\mathbb{E}[Y(t)] &= \sum_{k=1}^{K_Y} C_k \varphi_k(t) + \int_{I_X} \left(\sum_{k=1}^{K_Y} \sum_{m=1}^{K_X} B_{km} \varphi_k(t) \psi_m(s) \right) \left(\sum_{l=1}^{K_X} A_l \psi_l(s) \right) ds & (15) \\ &= \sum_{k=1}^{K_Y} \sum_{m=1}^{K_X} (C_k + B_{km} A_k) \varphi_k(t) & (16)\end{aligned}$$

which we know how to estimate.

4.4 Example of regression

In practice, to see if a fit is better than another one, one may compute the Root Mean Squared Error (RMSE) given by RMSE= $\sqrt{\mathbb{E}[(Y - \hat{Y})^2]}$, where Y represent the response and $\hat{Y}$ the fit, that is, $\hat{Y}$ is the predictor. The smaller the RMSE, the better the fit.

In Figure 8, we see an example of functional regression where Y is the observed precipitation and X is the smoothed temperature, both data taken in Canada. We plotted the data values against the fit. On both plots, we have that the black dots represent logarithm in base 10 of the annual precipitation in all the different stations of Canada and the red line represents the identity. We can remark that the RMSE for the Fourier basis is smaller than for the B-spline basis, thus the Fourier basis is better than the B-spline basis for this data. This may come from the fact that the precipitation data present a sort of periodicity during some period of time (cf Figure 9).



(a) RMSE = 0.0599

(b) RMSE = 0.0797

Figure 8: Regression of the precipitation data in Canada with Fourier basis (8a) and B-spline basis (8b) of degree $m = 2$, both with $n = 21$ functions

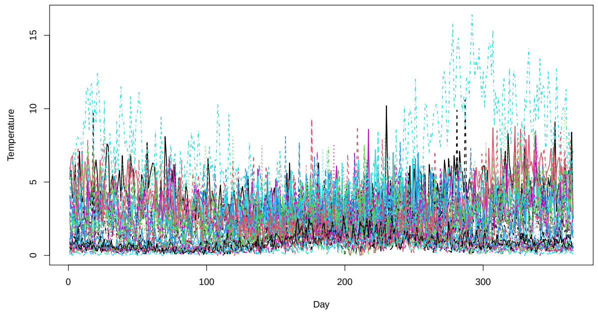
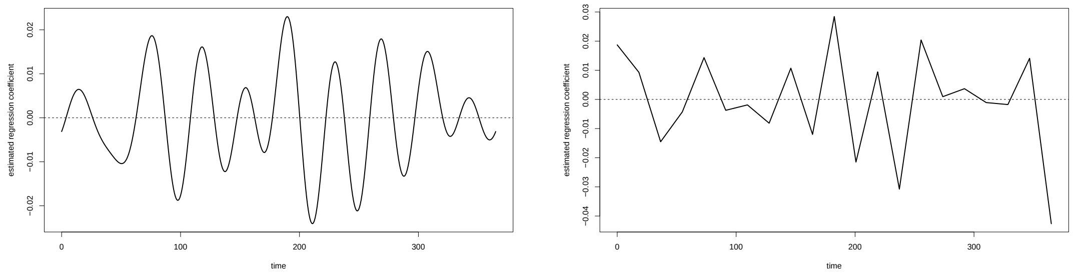


Figure 9: Precipitation in the 35 stations in Canada

In Figure 10, we see that with both basis, Fourier and B-spline, our $\beta(\cdot)$ function is close to zero at any time. This means that a small change in X will result in a negligible change in our response Y. In other words, there is almost no correlation between Y and X.



(a) Fourier basis with 21 functions

(b) B-spline basis with 21 functions of degree 2

Figure 10: Estimated coefficient $\hat{\beta}(\cdot)$ in function of the time for the two different basis

5 Application: Functional data analysis for volatility in finance

Let us now take a look at an example of data modeled as functions. We will study here the modeling of patterns of volatility for financial data, based on the paper [4]. To do so, let us first define the necessary background and financial modeling and then analyze a financial data set.

5.1 Background and financial modeling

5.1.1 Principal financial concepts for our analysis

For this section, we will often talk about financial asset. A FINANCIAL ASSET [7] represents a legal claim to future cash flows or economic resources. Financial assets are usually bought and sold in financial markets, and they can be owned by individuals, businesses or institutions. The most commons assets are stocks, bonds, currencies and options, among others.

Moreover, EQUITY PRICES [8], also known as stock prices, represent the current market value of publicly traded companies' stocks. These prices are determined by the supply and demand functions in the stock market. If a company is performing well and is expected to continue to do so, its stock price may rise as the demand for the stock increases. Conversely, if a company is struggling, its stock price may fall as investors sell their shares.

Equity prices are commonly tracked through STOCK MARKET INDICES (i.e., measures of the performance of a group of stocks in a certain market), such as the S&P 500 (market capitalization-weighted index of 500 large publicly traded companies in the USA) or the Dow Jones Industrial Index (price-weighted index of 30 large publicly traded companies in the USA), which provide a brief overview of how the stock market as a whole is performing.

In finance, trading patterns and volatility are two key concepts that are closely related. Understanding these concepts is essential to anyone who wants to be successful in trading financial instruments / assets.

TRADING PATTERNS [2] refer to recurrent patterns or trends in the financial markets that traders and investors use to identify potential buying or selling opportunities. These patterns can be identified by analyzing historical market data and can be used to make more informed trading decisions. There exists several kinds of trading patterns that are commonly used by traders. Let us state a few:

- TREND: directional movement in the price of a financial instrument. In an uptrend, prices are generally rising over time, while in a downtrend they are generally falling. Traders will look for stocks or other assets that are in a strong trend and then enter a trade in the direction of the trend, that is, buy when it is an uptrend and sell when it is a downtrend.

  In particular, there is SEASONAL TRENDS which refers to a trend that occurs consistently during a particular time of the year. These trends can be seen in many financial markets. They are often driven by seasonal factors, such as weather, holidays or change in customer habits, among others. These trends can help identify potential opportunities or risks in a particular market or industry. Nevertheless, they are not always predictable and can be influenced by a wide range of factors, which is why traders have to be careful when using them.

- RANGE: trading pattern in which prices move between a clear resistance level and support level (can be seen as lower and higher bounds). Traders will look for stocks that are bouncing between these two levels and enter a trade when the stocks hits either of these levels.

Overall, trading patterns can provide valuable insights into market trends and can be used to identify potential buying or selling opportunities, while being cautious about the risks.

On the other hand, VOLATILITY [3] represents how large an asset's prices swing around the mean price over time. In other words, volatility is a measure of the degree of uncertainty or risk associated with an asset or a financial market. In most cases, the higher the volatility, the riskier the security. That is, if the volatility is high, the price of the asset fluctuates rapidly and to a large extent, whereas a small volatility means that the price is more stable and experience less fluctuation.

Moreover, volatility can make patterns hard to predict. High level of volatility can lead to sudden and unexpected price movements, which can cause trading patterns to break down or produce false signals.

Many factors can contribute to volatility. For examples: economic and geopolitical events, changes in interest rate, or inflation may impact it. This implies that volatility prediction is not a hundred percent reliable since many factors come into consideration.

Those two concepts are closely related because traders can use certain patterns to predict future volatility. For example, if the stock is trading in a narrow range for an extended period of time, this could mean that the stock is about to experience a significant price movement, either up or down. Thus, traders can use this to make better investment decision, while being careful.

For example, in Figure 11 we plotted the prices at the end of business day of the ESTX 50 PR stock in euro, downloaded from the website *Yahoo! Finance*. We can see the upward (increasing parts) and downward (decreasing parts) trends in this graph. Moreover, the resistance level over all the period is 4408.49 and the support level is 3279.04.

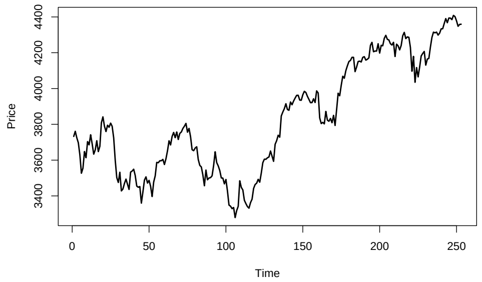


Figure 11: Closing price of the ESTX 50 PR stocks (in euros) from May 22nd 2022 to April 28th 2023

5.2 Functional data modeling and interpretation

In [4], they introduce a functional modeling of the volatility, based on the Black and Scholes model [1]. The main purpose of modeling volatility is to forecast volatility. Let us first define a particular type of stochastic process.

Definition 5.2.1. (Wiener process) A stochastic process $W(\cdot)$ is said to be a Wiener process if it satisfies the following properties:

- $W(0) = 0$;
- $\forall t > 0$, $W(t+u) - W(t)$ is independent of $W(s)$, $\forall u \geq 0$, $s < t$ *[independent increment property]*;
- $W(t+u) - W(t)$ follows the distribution $\mathcal{N}(0, u)$, $\forall t > 0$, $u \geq 0$;
- $W(\cdot)$ is continuous.

Let us consider the stochastic process of equity prices $X(\cdot)$. Then the (simplified) classical continuous time model for returns in the diffusion equation is given by:

$$\frac{dX(t)}{dt} = \mu X(t) + \sigma X(t)\frac{dW(t)}{dt}, \text{ for } t \geq 0 \tag{17}$$

where μ represent the drift term, $\sigma > 0$ the volatility and $W(\cdot)$ is a Wiener process. We remark that in this model, the drift and volatility are independent of the time, which is actually not the case.

In reality, these processes are observed on a discrete time grid, that we assume here to have equidistant time points, $\mathcal{G} = \{t_j = j\Delta\}_{j=1}^{[T/\Delta]}$ of the time interval $[0, T]$, with $\Delta > 0$ representing a period of time (in the paper, they took $\Delta = 5$ minutes). Here, for $r \geq 0$, $[r]$ denote the biggest integer before r. When the value of Δ is small, we refer to the data as high-frequency financial data.

For the study of high-frequency financial data, a class of diffusion model has been developed and they consist of two equations, one for the volatility $\sigma > 0$ and one directing the log returns according to

$$\frac{d(log(X(t)))}{dt} = \mu + \beta\sigma^2(t) + \sigma(t)\frac{dW(t)}{dt} \tag{18}$$

where, as before, $W(\cdot)$ is a Wiener process. Here, β is called the RISK PREMIUM and it represents the additional return that investors require to compensate for the additional risk they are taking on by investing in a riskier asset.

In order to infer the volatility, one has to consider the following. For $i = 1, \dots, n$ and a dense grid $\mathcal{G} = \{t_j = j\Delta\}_{j=1}^{[T/\Delta]}$:

$$Z_\Delta(t) = \frac{1}{\sqrt{\Delta}} \log\left(\frac{X(t+\Delta)}{X(t)}\right) \tag{19}$$

where $X(t)$ is a vector that represent the close (or open) price of the studied stocks at time t, where the CLOSE (OPEN) PRICE is the price of the stock from the last (first) transaction of a business day.

From (19), one can find the approximation $Z_\Delta(t) \approx \sigma(t)W_\Delta(t)$ where $W_\Delta(t) = \frac{1}{\sqrt{\Delta}}(W(t+\Delta) - W(t))$ and $\sigma(\cdot)$ is assumed to be smooth but not necessarily stationary.

From this, our target for inference is the process $V(t) = \log(\sigma(t)^2)$, which is called the FUNCTIONAL VOLATILITY PROCESS and does not depend on Δ.

Our goal is now to be able to apply this model to an actual data set, which we summarize below.

In reality, the data points are not necessarily equidistant, that is, the time grid can be written as $\mathcal{G} = \{t_j\}_j$ where $\Delta_j = t_{j+1} - t_j$ does not have to be a constant with respect to j.

In order to infer the volatility from a data set $\{X_i(\cdot)\}_{i=1}^n$ of prices of n different stocks, one has to consider the following. For $i = 1, \dots, n$ and a dense grid $\mathcal{G} = \{t_j\}_{j=1}^m$:

$$Z_{ij} = \frac{1}{t_{j+1} - t_j} \log\left(\frac{X_i(t_{j+1})}{X_i(t_j)}\right) \tag{20}$$

for all $j = 1, \dots, m-1$. Here $X_i(t)$ represents the close (or open) price of the i-th stock at time t. From this, we can construct our transformed and adjusted data

$$Y_{ij} = \log(Z_{ij}^2) - q_0 \tag{21}$$

where $q_0 = -1.27$ is a constant.

From this, we have that $\mu_V(t) \approx \mathbb{E}[\log(Z(t)^2)] - q_0 = \mathbb{E}[Y(t)]$ and $\Sigma_V(t,s) = \mathrm{Cov}(Y(t), Y(s))$ where $\mu_V(t) = \mathbb{E}[V(t)]$ and $\Sigma_V(t,s) = \mathrm{Cov}(V(t), V(s))$.

5.3 Data analysis

In order to do this data analysis, we downloaded data from the *Yahoo! Finance* website of $n = 30$ different stocks, with the euro currency, from May 2nd 2022 to April 28th 2023, being careful that the data points are taken on the same dates (removing the ones that are not). In our case, we have $m = 253$ data points. First, we studied the unmodified close price of the data and then we applied the model from the paper [4], summarized above in (20) and (21).

Using the unmodified data by putting each close price's columns for each stock in a matrix (a column represent a certain stock), one can smooth each data from every stock. For example, in Figure 12 we can see the smoothing (red line) of the close price of the *ESTX 50 PR* stock using splines.

We see that for this particular stock, the raw data was already smooth, except at the change of trends directions.

After this, one can do FPCA, using the same methodology than in section 3.3. That is, using Fourier basis with 52 functions (one for each week of the year) and with period the number of data points (here it is $m = 253$). We obtain Figure 13.

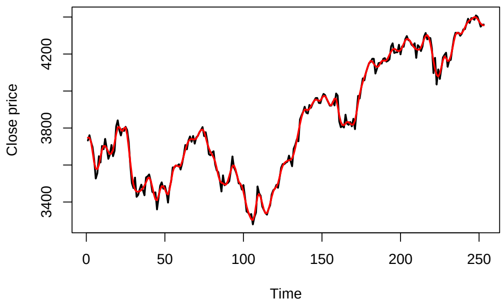


Figure 12: Smoothing of the ESTX 50 PR stock data using splines

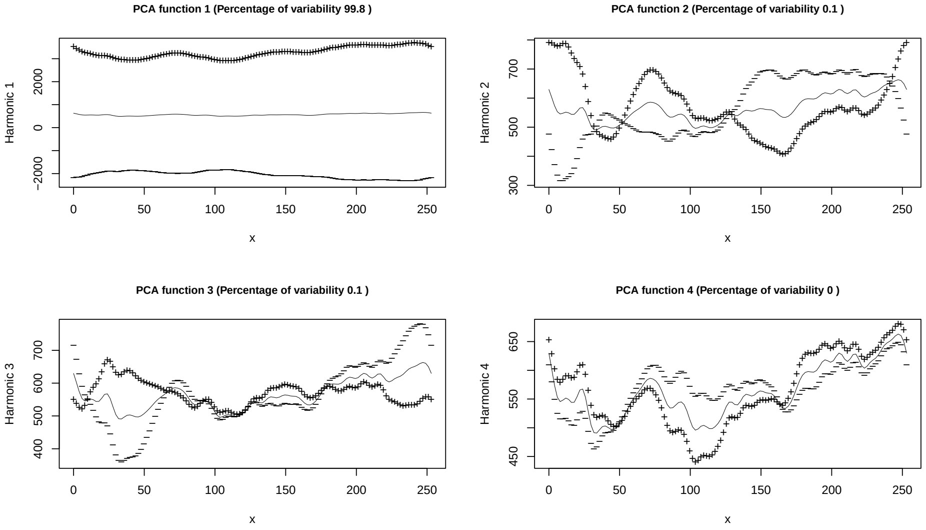


Figure 13: The overall mean closing price and the effect of adding (+) and subtracting (-) a multiple of each principal components curve

We see that the first components will mainly describe the variation pattern (a shift of the price). This can be explained from the fact that the stocks are really different and do not depend from one another.

Thus, we cannot say much about volatility by studying the unmodified data, so this is why we want to look next at the transformed and adjusted data (20).

5.3.1 FPCA (mean and major mode of variation)

Using FPCA, we get that the transformed data (with close price) has three principal components, represented in Figure 14 (fourth plot), with corresponding decreasing eigenvalues $\lambda_1 = 121.95$, $\lambda_2 = 8.76$, $\lambda_3 = 3.23$ and percentage of variability $v_1 = 91.05$, $v_2 = 6.54$, $v_3 = 2.41$.

The first eigenfunction mainly reflects the overall level of volatility. It represents a shift of volatility throughout the year.

The second eigenfunction corresponds to a shift in volatility that is not constant throughout the year. We remark that the impact of this shift on the variation is way less important for this component than the first.

The third eigenfunction tells us that some stocks have higher volatility during May to July and December to April than others, and vice-versa. It means that this eigenfunction differentiate between these periods of time.

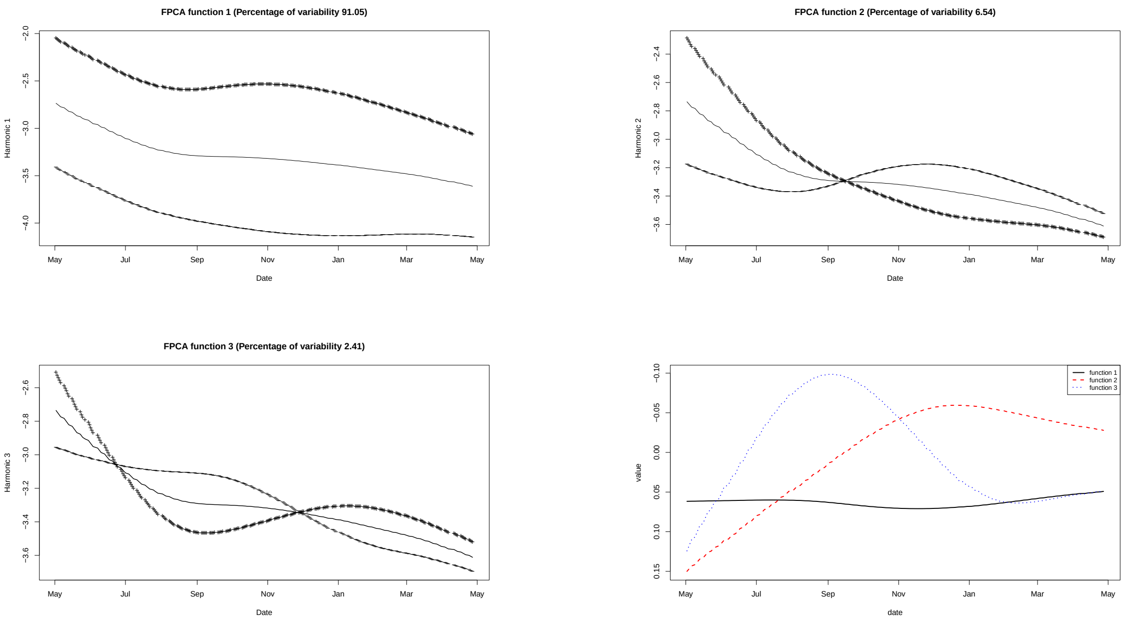


Figure 14: The overall mean closing price and the effect of adding (+) and subtracting (-) a multiple of each principal components curve (first three figures) and the first three estimated eigenfunctions of the functional volatility

One can try to compare the result obtained from the close price data to the open price data. Using FPCA as before, we also get three principal component, represented in Figure 15, with corresponding decreasing eigenvalues $\beta_1 = 122.08, \beta_2 = 6.58, \beta_3 = 2.69$ and percentage of variability $v'_1 = 92.95$, $v'_2 = 5.01$, $v'_3 = 2.04$.

We remark that the three principal components acts in the exact same way than for the opening price.

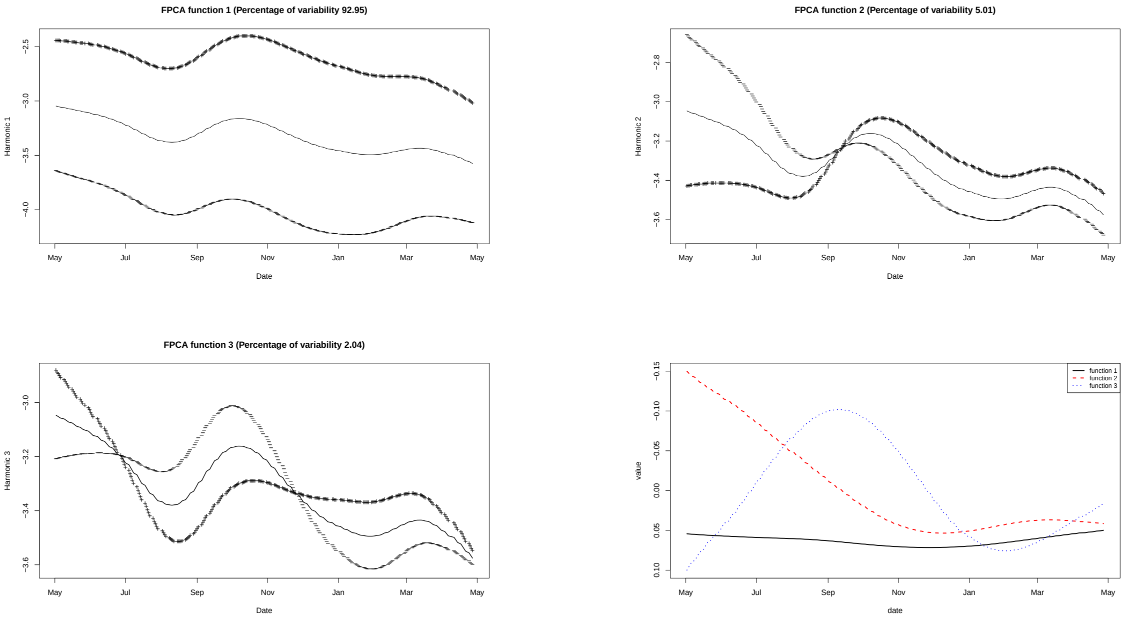


Figure 15: The overall mean opening price and the effect of adding (+) and subtracting (-) a multiple of each principal components curve (first three figures) and the first three estimated eigenfunctions of the functional volatility

The main aspect that changes is the form of our estimated μ. A few factors can explain this change, for example when studying volatility, the choice of time frame is crucial. Volatility measures based on the open price focus on the initial moments of the trading session, while those based on the close price cover the entire trading day. Depending on the trading style and objectives, we may choose to analyze intra-day volatility (open price) or longer-term volatility (close price).

Thus, depending on what the trader is looking for and its trading strategies, it may be more useful to look at the volatility with the open or close price. Indeed, on one hand volatility measures based on the open price may be useful for traders who prefer to capture early morning price movements and take advantage of market openings. On the other hand, volatility measures based on the close price can be valuable for traders interested in overall daily price movements or for those who use end-of-day trading strategies.

6 Discussion

Throughout this report, we summarized a part of the theory behind functional data analysis and applied it to a real data set.

We started by the basic results needed along the report: basis functions (looking at Fourier and B-splines basis) used to reduce dimension, and smoothing. If time had permitted, we could have talked about other basis functions, such as wavelets or exponentials, and smoothing methods. Indeed, many smoothing methods exists, we can cite Least squares smoothing or Kernel smoothing for example. Here, we decided to only summarize the one we used in our examples: splines smoothing, which we compared to the smoothing with Fourier basis functions. Then we summarized is functional principal component analysis: it provides insights into the underlying structure of functional data, helps visualize and interpret the major sources of variation, and allows for efficient data reconstruction. Thus it offers a powerful framework for analyzing functional data and extracting important features. However, functional principal component analysis has its limitations. For example, it is very sensitive to outliers since an extreme value can heavily influence the estimation of the principal components, which can lead to less accurate representation of the underlying variation in the data. Next, we looked at functional regression (scalar-on-function and function-on-function). It helps us find a fit that corresponds the most to our data, depending mainly on the used basis. To compare different fits, one can use the root mean squared error (the smaller it is, the better the fit). Nevertheless, it has its limitations. For example, it assumes a linear relationship between the predictor and the response. However, in many real-world cases, functional relationships can be nonlinear or show complex interactions.

Finally, we applied the theory to a concrete example: financial volatility of 30 stocks with 253 data points for each stock from May 2nd 2022 to April 28th 2023, studied thanks to a R program. In order to do this, we introduced basic definitions of financial terms, such as financial asset or stock market index. Then we summarized the model for the volatility that we applied to our data set. Afterward, we used functional principal component analysis. Nevertheless, the way of doing functional regression on our data set was not direct, mainly due to a lack of smoothness. Thus we used the transformation introduced in [4] to work with smooth representatives of volatility. Having had access to more data, we could have done regression using those smooth representatives of volatility, which would be our response.

Overall, this report outlined useful definitions and methods in order to do functional data analysis. It may come handy in real-world situations since data are everywhere. Naturally, one has to use a coding language (such as R or python) to do so, since in general one has to deal with high-dimensional data.

Appendix A Programming of the examples

In order to plot the figures present in the report, we used R as coding language. We mainly used the following functions and library:

- In the *fda* package, we used the functions: matplot to plot the columns of a matrix, create.fourier.basis and create.bspline.basis in order to create the basis we want to use, smooth.basis, fRegress, fdPar, pca.fd, plot.pca.fd, etc.

- In the *refund* package, we used the function fpca.face in order to do the application in the modified data case.

A.1 Background and methodology

Figure 1:

```r
matplot(CanadianWeather$dailyAv[,,"Temperature.C"], type = "l", pch=16:19,
    lwd=1.5, xlab = "day", ylab = "Temperature")
```

Figure 2:

```r
library('pracma')
Fn <- create.fourier.basis(rangeval = c(0,1), nbasis = 6)
plot(Fn, lwd = 2)
```

Figure 3:

```r
library('pracma')
Bn <- create.bspline.basis(rangeval = c(0,1), norder = 3, breaks = linspace(0,1,9))
plot(Bn, lwd = 2)
```

Figure 4:

```r
X <- (function(i) {cos(2*pi*i) / 2})
H = 200
G = seq(0,1,length.out = H)
eps = rnorm(H,mean=0,sd=1)
Xtrue = c()

for (h in G) {
X(h)
Xtrue = c(Xtrue, X(h))
}

plot(eps+Xtrue, type = "l")
points(Xtrue, type = "l", lwd=2, col="red")
```

Figure 5 and 6:

```r
data("CanadianWeather")

x <-CanadianWeather$dailyAv[,"Toronto","Temperature.C"]
plot(x, xlab="Day", ylab="Temperature")

#Figure 5,(a)
ss <- smooth.spline(x, y=NULL, w=NULL, df=10, df.offset = 0, penalty = 1)
#Figure 5,(b)
ss <- smooth.spline(x, y=NULL, w=NULL, df=50, df.offset = 0, penalty = 1)
#Figure 6
ss <- smooth.spline(x) #GCV

#Add ss (choose one of the above and put the other in commentary) to the plot of x
lines(ss, lwd=2.5, col="red")

fRegress(x)
```

A.2 Functional Principal Component Analysis

Figure 7: (we used the example already given in R)

```
library('fda')

daybasis65 <- create.fourier.basis(c(0, 365), nbasis=65, period=365)

harmaccelLfd <- vec2Lfd(c(0,(2*pi/365)ˆ2,0), c(0, 365))
harmfdPar    <- fdPar(daybasis65, harmaccelLfd, lambda=1e5)
daytempfd <- smooth.basis(day.5, CanadianWeather$dailyAv[,,"Temperature.C"],
                      daybasis65, fdnames=list("Day", "Station", "Deg C"))$fd

daytemppcaobj <- pca.fd(daytempfd, nharm=4, harmfdPar)

#plot harmonics in a pdf to control the size of the 4 graphs
plot
getwd()
setwd("/Users/salyadiallo/bureau")
pdf("pcafctns.pdf", width = 11, height = 6.5)
op <-par(mfrow=c(2,2))
plot.pca.fd(daytemppcaobj, lwd = 0.5,cex.main = 0.9)
par(op)
dev.off()
```

A.3 Functional regression

Figure 8 and 10: (put in commentary parts for figure (a) or (b))

```
data("CanadianWeather")
library('fda')

annualprec <- log10(apply(CanadianWeather$dailyAv[,,"Precipitation.mm"], 2, sum))

#Figure 8.(a)
smallbasis <- create.fourier.basis(c(0,365), 21)
#Figure 8.(b)
smallbasis <- create.bspline.basis(rangeval = c(0,365), nbasis = 21, norder = 2)

tempfd <- smooth.basis(day.5, CanadianWeather$dailyAv[,,"Temperature.C"], smallbasis)$
fd
precip.Temp1 <- fRegress(annualprec ˜ tempfd)

#plot of the data and the fitted value
annualprec.fit1 <- precip.Temp1$yhatfdobj

#Figure 8.(a)
plot(annualprec.fit1, annualprec, type="p", pch="o")
lines(annualprec.fit1, annualprec.fit1, lty=2, col="red", lwd=2)
#Figure 10.(a)
plot(precip.Temp1$betaestlist$tempfd, lwd=2, ylab = "estimated regression coefficient")

#Figure 8.(b)
plot(annualprec.fit1, annualprec, type="p", pch="o")
lines(annualprec.fit1, annualprec.fit1, lty=2, col="red", lwd=2)
#Figure 10.(b)
plot(precip.Temp1$betaestlist$tempfd, lwd=2, ylab = "estimated regression coefficient")

RMSE <- sqrt(mean((annualprec-annualprec.fit1)ˆ2))
print(paste("RMSE =",RMSE))
```

Figure 9:

```
matplot(CanadianWeather$dailyAv[,,"Precipitation.mm"], type = "l", pch=16:19, lwd=1.5,
xlab = "day", ylab = "Temperature")
```

A.4 Application: Functional data analysis for volatility in finance

Study of the non modified data: (Figure 11, 12 and 13)

```r
library('fda')

# Data taken from the website: Yahoo! finance
STOXX50E <- read.table("/Users/salyadiallo/bureau/STOXX50E_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
FCHI <- read.table("/Users/salyadiallo/bureau/FCHI_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
ADS.DE <- read.table("/Users/salyadiallo/bureau/ADS.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
ETH.EUR <- read.table("/Users/salyadiallo/bureau/ETH-EUR_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
SY1.DE <- read.table("/Users/salyadiallo/bureau/SY1.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
TL0.MU <- read.table("/Users/salyadiallo/bureau/TL0.MU_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
APC.MU <- read.table("/Users/salyadiallo/bureau/APC.MU_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
PAH3.DE <- read.table("/Users/salyadiallo/bureau/PAH3.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
MBG.DE <- read.table("/Users/salyadiallo/bureau/MBG.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
MOH.FR <- read.table("/Users/salyadiallo/bureau/MOH.FR_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
RMS.PA <- read.table("/Users/salyadiallo/bureau/RMS.PA.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
AIR.DE <- read.table("/Users/salyadiallo/bureau/AIR.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
ALV.DE <- read.table("/Users/salyadiallo/bureau/ALV.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
BAS.DE <- read.table("/Users/salyadiallo/bureau/BAS.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
BNR.DE <- read.table("/Users/salyadiallo/bureau/BNR.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
CON.DE <- read.table("/Users/salyadiallo/bureau/CON.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
MRK.DE <- read.table("/Users/salyadiallo/bureau/MRK.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
ZAL.DE <- read.table("/Users/salyadiallo/bureau/ZAL.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
HEIA.AMS <- read.table("/Users/salyadiallo/bureau/HEIA.AMS_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
VOW3.DE <- read.table("/Users/salyadiallo/bureau/VOW3.DE_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
OR.PA <- read.table("/Users/salyadiallo/bureau/OR.PA_bis.tsv",  header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
BN.PA <- read.table("/Users/salyadiallo/bureau/BN.PA_bis.tsv", header=FALSE, sep ="\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose", "Volume"))
```

```
KER.PA <- read.table("/Users/salyadiallo/bureau/KER.PA_bis.tsv", header=FALSE, sep ="\t
", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "
AdjClose", "Volume"))
RNO.PA <- read.table("/Users/salyadiallo/bureau/RNO.PA_bis.tsv", header=FALSE, sep ="\t
", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "
AdjClose", "Volume"))
HO.PA <- read.table("/Users/salyadiallo/bureau/HO.PA_bis.tsv", header=FALSE, sep ="\t",
 stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose
", "Volume"))
RI.PA <- read.table("/Users/salyadiallo/bureau/RI.PA_bis.tsv", header=FALSE, sep ="\t",
 stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose
", "Volume"))
DG.PA <- read.table("/Users/salyadiallo/bureau/DG.PA_bis.tsv", header=FALSE, sep ="\t",
 stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose
", "Volume"))
SU.PA <- read.table("/Users/salyadiallo/bureau/SU.PA_bis.tsv", header=FALSE, sep ="\t",
 stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "AdjClose
", "Volume"))
CAP.PA <- read.table("/Users/salyadiallo/bureau/CAP.PA_bis.tsv", header=FALSE, sep ="\t
", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "
AdjClose", "Volume"))
ANA.MC <- read.table("/Users/salyadiallo/bureau/ANA.MC_bis.tsv",    header=FALSE, sep =
"\t", stringsAsFactors=TRUE, col.names = c("Date", "Open", "High", "Low", "Close", "
AdjClose", "Volume"))

# Creation of the data frame by taking the same column for each stocks
# Here we chose to study the close price
n <- length(STOXX50E[,1]) #date
v <- c(STOXX50E[,5], FCHI[,5],   ADS.DE[,5],   ETH.EUR[,5],
       SY1.DE[,5],   TL0.MU[,5], APC.MU[,5],   PAH3.DE[,5],
       MBG.DE[,5],   MOH.FR[,5], RMS.PA[,5],   AIR.DE[,5],
       ALV.DE[,5],   BAS.DE[,5], BNR.DE[,5],   CON.DE[,5],
       MRK.DE[,5],   ZAL.DE[,5], HEIA.AMS[,5], VOW3.DE[,5],
       OR.PA[,5],    BN.PA[,5],  KER.PA[,5],   RNO.PA[,5],
       HO.PA[,5],    RI.PA[,5],  DG.PA[,5],    SU.PA[,5],
       CAP.PA[,5],   ANA.MC[,5])
data <- matrix(v, nrow=n)


# Smoothing of each stocks' data
for (k in 1:30) {
 x <- data[,k]
 ss <- smooth.spline(x)
 plot(x, lwd = 2, type = "l", xlab = "Time", ylab = "Close price")
 lines(ss, lwd = 2, col = "red")
}

# FPCA
daybasis <- create.fourier.basis(c(0,n), nbasis = 52, period = n)
harmaccelLfd <- vec2Lfd(c(0,(2*pi/n)^2,0), c(0, n))
harmfdPar    <- fdPar(daybasis, harmaccelLfd, lambda=1e5)
pricefd <- smooth.basis(FCHI[,1], data, daybasis, fdnames = list("Day", "Close", "Price
"))$fd
pricefdobj <- pca.fd(pricefd, nharm = 4, harmfdPar)

op <-par(mfrow=c(2,2))
plot.pca.fd(pricefdobj, lwd = 0.5,cex.main = 0.9)
par(op)
```

Study of the modified data to obtain the volatility: (Figure 14 and 15 –> in the vector *v*, take the second columns of each stock's data and modify the *ylim* for each function in the FPCA section)

```
library('fda')
library('base')
library('refund')

# Same data as before taken from the website: Yahoo! finance

# X = data from the previous code
n <- length(STOXX50E[,1]) #date
v <- c(STOXX50E[,5], FCHI[,5],   ADS.DE[,5],   ETH.EUR[,5],
       SY1.DE[,5],   TL0.MU[,5], APC.MU[,5],   PAH3.DE[,5],
       MBG.DE[,5],   MOH.FR[,5], RMS.PA[,5],   AIR.DE[,5],
       ALV.DE[,5],   BAS.DE[,5], BNR.DE[,5],   CON.DE[,5],
       MRK.DE[,5],   ZAL.DE[,5], HEIA.AMS[,5], VOW3.DE[,5],
       OR.PA[,5],    BN.PA[,5],  KER.PA[,5],   RNO.PA[,5],
       HO.PA[,5],    RI.PA[,5],  DG.PA[,5],    SU.PA[,5],
       CAP.PA[,5],   ANA.MC[,5])
X <- matrix(v, nrow=n)

# Vector to determine the time between each data points (each number represents a
certain day)
# Here: 1 = May 2nd 2022, 361 = April 28th 2023
base <- c(1,2,3,4,5,8,9,10,11,12,15,16,17,18,19,22,23,24,26,29,30,
31,32,33,37,38,39,40,43,44,45,46,47,50,51,52,53,54,57,58,59,60,
61,64,65,66,67,68,71,72,73,74,75,78,79,80,81,82,85,86,87,88,89,
93,94,95,96,99,100,101,102,103,106,107,108,109,110,113,114,115,116,117,120,121,122,
123,124,127,128,129,130,131,134,135,136,137,138,141,142,143,144,145,148,149,150,151,
152,155,156,157,158,159,162,163,164,165,166,169,170,171,172,173,176,177,178,179,180,183,
 184,185,186,187,190,191,192,193,194,197,198,199,200,201,204,205,206,207,208,211,211,
213,214,215,218,219,220,221,222,225,226,227,228,229,232,233,234,235,236,240,241,242,243,
 247,248,249,250,253,254,255,256,257,260,261,262,263,264,267,268,269,270,271,274,275,
276,277,278,281,282,283,284,285,288,289,290,291,292,295,296,297,298,299,302,303,
304,305,306,309,310,311,312,313,316,317,318,319,320,323,324,325,326,327,330,331,332,
333,334,337,338,339,340,344,345,346,347,350,351,352,353,354,357,358,359,360,361)

# Creation of the adjusted data
m <- dim(X)[2] #number of stocks studied --> here: 30
Z <- matrix(nrow = n-1, ncol = m)
for (i in 1:n-1) {
  for (j in 1:m) {
    Z[i,j] <- (log10(X[i+1,j]) - log10(X[i,j]))/sqrt(base[i+1] - base[i])
  }
}
# Remove infinite and NaN values
Z[is.infinite(Z)] <- 0
Z[is.nan(Z)] <- 0


qo <- -1.27 # given value
Y <- log10(Z^2) - qo
# Remove infinite and NaN values
Y[is.infinite(Y)] <- 0
Y[is.nan(Y)] <- 0


#FPCA
pca      <- fpca.face(t(Y), pve = 0.99, var = FALSE, center = TRUE, knots = 25,
                   p = 3, m = 2, alpha = 1, method = "L-BFGS-B")
Y_smooth <- pca$Yhat
mu       <- pca$mu   #represent the mean of the volatility
eigfun   <- pca$efunctions
eigval   <- pca$evalues

lambda <- sum(eigval)

# Plots of the different modes of variation
#Function 1
vari1 <- eigval[1]*100/lambda # 91.0508 (variability of the 1st principal component)
mode1_plus  <- mu + sqrt(eigval[1])*eigfun[,1]/6
mode1_minus <- mu - sqrt(eigval[1])*eigfun[,1]/6
plot(mu, type = "l", ylim = c(-3.704,-2.623), xlab = "x", ylab = "Harmonic 1",
     main = "FPCA function 1 (Percentage of variability 91.05)")
points(mode1_plus, pch = "+")
points(mode1_minus, pch = "-")

#Function 2
vari2 <- eigval[2]*100/lambda # 6.536557 (variability of the 2nd principal component)
mode2_plus  <- mu + sqrt(eigval[2])*eigfun[,2]/6
mode2_minus <- mu - sqrt(eigval[2])*eigfun[,2]/6
plot(mu, type = "l", ylim = c(-3.63,-2.662), xlab = "x", ylab = "Harmonic 2",
     main = "FPCA function 2 (Percentage of variability 6.54)")
points(mode2_plus, pch = "+")
points(mode2_minus, pch = "-")

#Function 3
vari3 <- eigval[3]*100/lambda # 2.412648 (variability of the 3rd principal component)
mode3_plus  <- mu + sqrt(eigval[3])*eigfun[,3]/6
```

```r
    mode3_minus <- mu - sqrt(eigval[3])*eigfun[,3]/6
    plot(mu, type = "l", ylim = c(-3.63,-2.699), lwd=1.5, xlab = "x", ylab = "Harmonic 3",
    main = "FPCA function 3 (Percentage of variability 2.41)")
    points(mode3_plus, pch = "+")
    points(mode3_minus, pch = "-")

    # Plots of the three main eigenfunctions
    plot(date, eigfun[,1], ylim = c(0.151,-0.1), ylab = "value", type = "l", col = "black",
         lwd = 2)
    points(date, eigfun[,2], type = "l", lty = "dashed", col = "red", lwd = 2)
    points(date, eigfun[,3], type = "l", lty = "dotted", col = "blue", lwd = 2)
    legend("topright", legend = c("function 1", "function 2", "function 3"),
            col = c("black","red","blue"), lty = 1:3, cex = 0.8, lwd = 2)
```

References

[1] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, Vol. 81(3):637–654, 1973.

[2] Adam Hayes. Introduction to stock chart patterns. *Investopedia*, 2023.

[3] Adam Hayes. Volatility: Meaning in finance and how it works with stocks. *Investopedia*, 2023.

[4] Hans-Georg Müller, Rituparna Sen, and Ulrich Stadtmüller. Functional data analysis for volatility. *Journal of Econometrics*, Vol. 165(2):233–245, 2011.

[5] J.O. Ramsay and B.W. Silverman. *Functional Data Analysis, Second edition*. Springer Series in Statistics, 2005.

[6] Philip T. Reiss, Jeff Goldsmith, Han Lin Shang, and R. Todd Ogden. Methods for scalar-on-function regression. *International Statistical Review*, Vol. 85(2):228–249, 2017.

[7] CFI Team. Financial assets - definition and classification. *CFI*, 2023.

[8] CFI Team. Stock price. *CFI*, 2023.

[9] Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional data analysis. *Annual review statistics and its application*, Vol. 3:257–295, 2016.