Chapter 1: Introduction

Elements of Finite Data Analysis

January 2021

We provide a brief motivation to the analysis of functional data, presenting the stochastic process and random element points of view for modelling functional data and survey some of the topics that will be covered in the coming chapters. The monograph of Ramsay and Silverman [2007] elaborates on many topics touched upon in this chapter; further detailed examples are presented in Kokoszka and Reimherr [2017] and Ramsay and Silverman [1997].

1 What is Functional Data?

A *stochastic process* is an indexed collection of random variables, all of which are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Given an index set I, a stochastic process is characterised by the collection

$$\{X(t,\omega) \,:\, t \in I, \omega \in \Omega\},$$

where $X(t,\cdot)$ is an $\mathcal{F}$-measurable function on the sample space Ω. The ω argument is often suppressed for simplicity, so that $X(t,\omega)$ is shortened to $X(t)$ when the context is clear.

If we were to observe $X(t)$ for every $t \in I$, then the process has been realised, with the resulting collection of real numbers called a *sample path* for the process. *Functional Data Analysis* is concerned with methods for analyzing data which represented sample paths of processes where the index set is some interval of the real line (or more generally of $\mathbb{R}^d$). In this context, our data consists of collections of such random curves.

In practice, one does not actually observe a functional data set in its entirety. Some form of discretisation take places. A typical data set might consist of samples of the form

$$\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \text{ where } \mathbf{x}_i = (X_i(t_j))_{j=1}^M, \text{ for } t_1, \ldots, t_M \in I.$$

This data set involves n (possibly independent) sample paths $X_1(\cdot), \ldots, X_n(\cdot)$ of some stochastic process with each sample path only being evaluated at the points $t_1, \ldots, t_M$. Given that the data is inherently finite dimensional, i.e. $\mathbf{x}_i \in \mathbb{R}^M$, one might ask why one might just resort to standard multivariate data analysis techniques. A naive treatment of this setting may have undesired consequences.

Example 1.1. To make things concrete, assume we have a sample $X_1, \ldots, X_N$ which we model as i.i.d realisations of a stochastic process $X(t)$ such that $\mathbb{E}[X(t)^2] < \infty$. We are interested in making inferences about the mean function $m(t) = \mathbb{E}[X(t)]$, $t \in I$, specifically we wish to test the null hypothesis $H_0 : \mu \equiv 0$. Suppose we have access to measurements $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and let $\overline{\mathbf{x}} \in \mathbb{R}^M$ be the associated sample mean vector. The standard test in multivariate analysis would be Hotelling's T-squared test, based on the test statistic $T^2 = \overline{\mathbf{x}}^\top \widehat{\Sigma}^{-1} \overline{\mathbf{x}}$, where $\widehat{\Sigma}$ is the sample covariance matrix. In the context of functional data, the dimension M is somewhat arbitrary. Indeed, it can often occur that the number of measurements per sample path may be larger than the sample size N, i.e. $M \gg N$. Clearly, $\widehat{\Sigma}$ will be singular in this case. However, even if one keeps $M < N$, as M is arbitrary, we expect that $\widehat{\Sigma}$ will typically be poorly conditioned for many values of $M < N$. Thus, the results may depend critically on the choice of M.

The above example about one-sample testing clearly indicates that we must employ methods to be tailored to functional data. Indeed, knowing that the data arises from the discretisation of a set of curves carries with it important information: smoothness will imply that multiple nearby measurements of the same curve will have similar value. In this setting, smooth curves and surfaces can be readily approximated by a finite (often small) number of smooth functions. A central task in the functional data analysis pipeline is to approximate a functional object by a smaller number p of standard building blocks, with p being much smaller than M. Such approximations give rise to many interesting and challenging questions not encountered in statistical inference for scalars or vectors.

Generally speaking, methods for functional data analysis should satisfy the Grid Refinement Invariance Principle (GRIP) Lee et al. [2015]: any procedure for functional data should be insensitive to the dimension of the representation, as long as the dimension is sufficiently large to give an accurate representation. Generally speaking, there are two strategies which can be employed to ensure a method satisfies GRIP:

1. Devise a method for true functional data (i.e. directly acting on curves and surfaces), then "project" the method into finite dimensions so that the resulting finite-dimensional method remains robust with respect to discretisation.
2. Devise a method for the finite dimensional data, then see if it has a well-defined limit as $M \to \infty$.

At the other end of the spectrum, many important applications in FDA deal with situations where we observe each curve at a very small number of sample points, which differ for every curve. That is, we observe

$$\{\mathbf{x}_1, \dots, \mathbf{x}_n\}, \text{ where } \mathbf{x}_i = (X_i(t_{ij}))_{j=1}^{M_i} \in \mathbb{R}^{M_i}, \text{ for } t_{i1}, \dots, t_{iM_i} \in I \text{ and } i = 1, \dots, N.$$

In this setting, modelling these observations as (possibly noisy) evaluations of curves at different points is a power assumption. Such cases require custom methodology, which involve carefully pooling of measurements to extract large-scale features from the data. This scenario will be revisited throughout the next chapters. Functional data for which only a few observations are available for every curve are referred to as *sparsely* observed as opposed to the case of *densely* observed curves. There is no obvious cut-off between sparsely and densely observed, so this is a subjective qualification, and there will be many situations which are difficult to classify in this fashion.

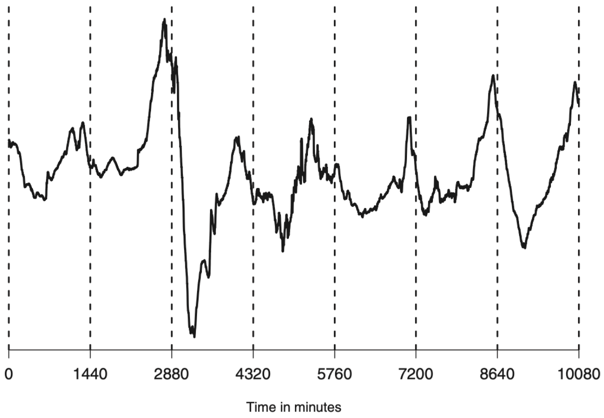

Figure 1: The horizontal component of the magnetic field measured at Honolulu magnetic observatory from 1/1/2001 00:00 UT to 1/7/2001 24:00 UT. The vertical dashed lines separate 24h days. Each daily curve is viewed as a single functional observation. Example from Kokoszka and Reimherr [2017].

2 Examples of Functional Data

A classical first example provided in many FDA textbooks (e.g. Kokoszka and Reimherr [2017]) is that of magnetometer records. In Figure 1, we plot the strength $X_n(t)$ of the magnetic field measured at the Honololu magnetic observatory at time t on day n. In this case, a digital magnetometer records the values every five seconds, resulting in $17,280$ observations per day, however the magnetic field exists at any moment of time, so it is natural to think of a magnetogram as an approximation to a continuous record. For magnetometer data, space physics researchers are not interested in the value of the field every five seconds, but in the shape of the curve over several days. The shapes of such curves at many locations allow researchers to make inferences about physical processes occurring in near Earth space where no instruments are placed.

A classical second example is that of high frequency financial data. Shares of large companies are traded up to a thousand times every second – so frequently that one can practically think of a price curve that is defined at any moment of time. Indeed, stock prices are frequently modelled by continuous time Markov processes such as Stochastic Differential Equations (SDEs). Figure 2 shows the values of Microsoft stock minute by minute. Choosing one day as the underlying time interval It is natural to choose one trading day as the underlying time interval, Figure 2 shows 10 consecutive functional observations. From these functional observations, various statistics can be computed, for example the mean function, two-point correlation function, functional principal components, etc.

By explicitly assuming that our data set of vectors arises as discrete observations of curves or surfaces, we immediately gain access to function-specific operations which may shed new insight on our dataset. This is perfectly illustrated in the first dataset presented in Ramsay and Silverman's 2002 seminal text on FDA. Figure 3(a), shows the heights of girls measured at a set of 31 ages in the Berkeley Growth Study (Tuddenham and Snyder, 1954). Note that the ages are not equally spaced and there is an uncertainty in the measurement of height values with a standard deviation of $3mm$. Clearly, these are discrete observations of a continual growth process, for which it is

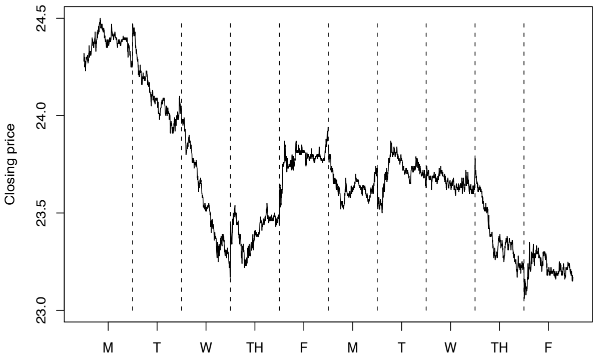

Figure 2: Microsoft stock prices in one-minute resolution, May 1-5, 8-12, 2006. The displayed data can be viewed as a sample of 10 functional observations. Example from Kokoszka and Reimherr [2017].

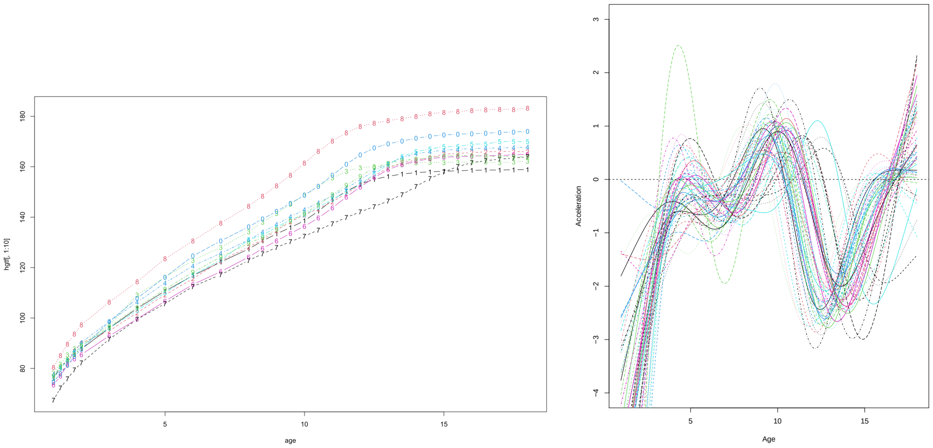

Figure 3: **Left:** The heights of girls measured at 31 ages. The markers indicate the unequally spaced ages of measurement. **Right:** The estimated accelerations of height for the girls. Example from Ramsay and Silverman [2007]

natural to assume that the is an underlying height function $H_i(t)$ for the i^{th} girl. Figure 3(a) is not immediately yielding any obvious insight. On the other hand, in Figure 3(b) we plot an approximation to the second deriative $\frac{d^2H_i(t)}{dt^2}$ of the height function, using an approximation scheme which will be discussed in Lecture 2. We immediately see new structure in the data: pubertal growth spurt shows up as a pulse of strong positive acceleration followed by sharp negative deceleration. Most records also show a bump at around six years. We therefore conclude that some of the variation from curve to curve can be explained at the level of the second derivative. Some type of principal components analysis would certainly shed more light on the variability between the curves, but to do so would require a procedure able to take account of the unequal age spacing and the smoothness of the underlying height functions.

3 Elements of the FDA toolkit

While it is clear that a naive application of Multivariate Analysis methodology will not be a fruitful avenue to follow for FDA, one the other hand, we would expect analogous of these techniques to be meaningful. Data analysis tools such as canonical correlation analysis, discriminant analysis, factor analysis, multivariate analysis of variance (MANOVA), and principal components analysis provide powerful means of summarizing complex datasets and carry out inference about the underlying parent population. Thus we would expect them to remain conceptually valid in the FDA setting, even if these method require adjustment to deal with the infinite dimensional nature of the underlying functional objects, as well as notions of smoothness of data, etc.

3.1 Summary Statistics

Having transformed raw discretised functional data to curves (or other functional objects) it is common to compute various summary statistics of the data. Assuming that the curves $x_1, \ldots, x_N$ are realisations of an underlying stochastic process $X(t)$ the most common summary statistics are the *functional / pointwise mean* and the *covariance function.* The pointwise mean $\mathbb{E}[X(t)]$ will itself be function on the same interval as the function samples. It is typically estimated via the sample mean, given by

$$\overline{X}_N(t) = \frac{1}{N}\sum_{n=1}^{N} X_n(t), \quad t \in [0,1].$$

The covariance function summarizes the dependence of functions across different argument values t_1 and t_2, i.e. $\text{Cov}[X(t), X(t)]$. This is typically estimated using the following sample covariance function:

$$\widehat{c}(t,s) = \frac{1}{N-1}\sum_{n=1}^{N}(X_n(t) - \overline{X}_N(t))(X_n(s) - \overline{X}_N(s)).$$

The sample covariance function is typically noisy and difficult to interpret. Therefore, bivariate smoothing is usually employed. Local linear smoothers Fan and Gijbels [1996], and thin plate regression splines Wood [2003] are among the popular methods for smoothing the sample covariance function. We shall discuss properties of these estimators and associated tests iin Chapter 2.

3.2 Functional Principal Component Analysis (FPCA)

One of the most useful and often used tools in functional data analysis is the principal component analysis. As we shall show in Chapters 2 and 3, the *functional principal components* can be related analytically to the covariance function. Speaking heuristically, the objective the idea of FPCA is to find a sequence of functions $v_1, v_2, \ldots$ which are *orthonormal* in an appropriate sense (indeed, they form an orthonormal basis in an appropriate Hilbert space) such that

$$X_n(t) - \overline{X}_N(t) \approx \sum_{j=1}^{p} \xi_{n,j} v_j,$$

where p is much smaller than N and $\xi_{n,j}$ are uncorrelated random variables. The coefficient $\xi_{n,j}$ is called the *score* of X_n with respct to v_j and quantifies the contribution of v_j to the variation in X_n. Indeed, as we shall see in Chapter 3, the total variability of a sample of curves about the mean function, can be decomposed into the sum of variabili- ties explained by each functional principal component.

3.3 Functional Canonical Correlation Analysis

Functional Canonical Correlation Analysis (FCA) is a tool to quantify correlations between pairs of observed random curves for which a sample is available. Canonical correlation analysis is used to identify and measure the associations among two sets of variables. Canonical correlation analysis determines a set of canonical variates, orthogonal linear combinations of the variables within each set that best explain the variability both within and between sets. More specifically, suppose we observe a sample of pairs of functions

$$(x_1, y_1), \ldots, (x_N, y_N),$$

we wish to find two *linear functionals* (i.e. linear functions which act on functions), a and b such that we maximise the sample correlation between the $N \times 1$ vectors

$$A = [a(x_1), \ldots, a(x_N)] \quad \text{ and } B = [b(y_1), \ldots, b(y_N)].$$

FCA provides a very powerful technique to identify the "propagation of variability" between two sets of dependent functions. A crucial aspect of the functional context is that standard finite dimensional CCA will not work reliably for functional data. Crucially, it does not pass the GRIP criterion outlined in the introduction: it is possible to find linear functionals a and b such that the correlations between A and B are arbitrarily close to on. As the resolution of the functional data increases, we expect the weights a and b to converge to these irregular solutions. To mitigate this abundance of flexibility, regularisation must be introduced to penalise overly flexible functionals. We will discuss functional canonical correlation analysis and its applications in Chapter 4.

3.4 Outlier Detection for Functional Data

Given a set of curves, it is not entirely clear how one would go to classify a particular curve to be an outlier. To answer this questions we require appropriate notions of a centre and distance over samples of curves. The value of a function at every point t may not be an outlier, but the curve itself may be a functional outlier. The concept of *depth* of functional data offers a possible framework for identifying central and outlying observations; those with maximal depth are central, and those with minimal depth are potential outliers. For scalar data, the depth of a sample point can be defined in terms of the empirical cumulative distribution function, via the halfspace depth.

$$HD(X_i) = \min\left(F_N(X_i), 1 - F_N(X_i)\right),$$

where $F_N(\cdot)$ is the empirical CDF. This assigns depth 1/2 to the median point, which has the largest possible depth. How could one generalise this to functional data, particularly given the infinite dimensional nature of the data, and that we might only have partial visibility of each sample curve? Assuming we have complete access to the curves, Fraiman and Muniz [2001] provide the following direct generalisation of functional depth, generalising the above:

$$FD(X_i) = \int_0^1 \left[1 - |\frac{1}{2} - F_{N,t}(X_i(t))\right] dt,$$

where $F_{N,t}$ is the empirical CDF of the functional data at time t. There are also other approaches to defining functional depth. We will discuss thes generalisations in Chapter 5. Very often the interest is focused on studying how extreme a sample curve x is with respect to a given the distribution X of functions. In multivariate analysis, one often considers the *Mahalanobis distance*:

$$M(x) = (x - \mathbb{E}[X])^\top \Sigma^{-1} (x - \mathbb{E}[X]), \tag{1}$$

where Σ is the (non-singular) covariance matrix of X. Mahalanobis distances are nowadays used in supervised classification, outlier detection as well as hypothesis testing (through Hotelling's statistic).

So how do we extend the notion of the multivariate (finite-dimensional) Mahalanobis distance (1) to the functional case? The natural analogue of the covariance matrix Σ is the *covariance operator* $\mathcal{K}$ defined by,

$$\mathcal{K}f(t) = \int_0^1 c(t,s)f(s)\, ds,$$

where $c(t,s)$ is the two-point covariance function. We will define this rigorously in Chapter 2. While $\mathcal{K}$ shares many common properties with that of a multivariate covariance matrix, there are some very important differences. While $\mathcal{K}$ is positive definite, its eigenvalues can get arbitrarily close to zero, unlike the finite dimensional analogue. This results in the non-invertibility of the covariance operator in infinite-dimensional cases. To circumvent this issue a new construction of Mahalanobis distance in infinite dimensions is required. This will be revisited in Chapter 5, where we discuss some hypothesis tests for functional data.

4 Reconstructing functions from discrete functional data

Often, the first step when analysing discretised observations of functional data is to try to estimate the underlying functional curves. Generally speaking there are two approaches for approximating the underlying curves from functional data: using *basis expansions* or *kernel based smoothing*. We will discuss both very briefly, without covering much of the underlying theory. Much of the content of this section follows closely Ramsay and Silverman [1997] and Ramsay and Silverman [2007].

4.1 Representing functions by basis functions

In this approach, we postulate that we make (noisy) observations $y_1, \dots, y_J$ of the underlying curve $x(\cdot)$ at points $t_1, \dots, t_J$.

$$y_j = x(t_j) + \varepsilon_j, \qquad \text{for } j = 1, \dots, J. \tag{2}$$

We also assume that the curve $x(\cdot)$ is expressible in terms of M basis functions:

$$x(t) = \sum_{m=1}^{M} c_m \Phi_m(t). \tag{3}$$

The functions $\Phi_m(\cdot)$, $m = 1, 2, \ldots$ are typically chosen to be some (infinite) set of *basis functions.* We will discuss bases on function spaces in far more detail in the next chapter. For now we will just suspend belief and pretend that everything is finite dimensional. Intuitively, it makes sense to choose a basis such that the unknown "true" curve (i) can be expressed as a linear combination of these basis functions, at least in the limit as $M \to \infty$; (ii) but it can be well approximated (this will be made precise later) using only a small number M of the basis functions. In practice, we would want that $M < J$.

If we assume the *standard model for error*, i.e. that the residuals ε_j are independent and identically distributed random variables with mean zero and constant variance σ^2, a simple linear smoother is obtained by minimising the following least squares loss:

$$L(\mathbf{y} \,|\, \mathbf{c}) = \sum_{j=1}^{n} \left| y_j - \sum_{m=1}^{M} c_m \Phi_m(t_j) \right|^2 .$$

Introducing the notation $\mathbf{\Phi}_m = (\Phi_m(t_1), \ldots, \Phi_m(t_J))^\top \in \mathbb{R}^J$ and the $J \times M$ matrix $\mathbf{\Phi} = [\Phi_1| \ldots |\Phi_M]$ and $\mathbf{c} = (c_1, \ldots, c_M)^\top$ we can express the loss as

$$\|\mathbf{y} - \mathbf{\Phi}\mathbf{c}\|^2 = (\mathbf{y} - \mathbf{\Phi}\mathbf{c})^\top (\mathbf{y} - \mathbf{\Phi}\mathbf{c}) .$$

Taking the derivative of the loss with respect to $\mathbf{c}$ yields the least-squares solution:

$$\widehat{\mathbf{c}} = (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top \mathbf{y}.$$

Suppose we wish to evaluate the predicted curves at points $\mathbf{t}^* = t_1^*, \ldots, t_K^*$. Defining $\mathbf{\Phi}_m^* = (\Phi_m(t_1^*), \ldots, \Phi_m(t_K^*))^\top$ and $\mathbf{\Phi}^* = [\Phi_1^*| \ldots |\Phi_M^*]$ the vector $\widehat{\mathbf{y}}^*$ of predicted values is given by

$$\widehat{\mathbf{y}}^* = \mathbf{\Phi}^* (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top \mathbf{y}.$$

What do we mean by the smoother being *linear*? A linear smoother will estimate the function value $\widehat{x}(t_j)$ by a linear combination of the discrete observations

$$\widehat{x}(t) = \sum_{l=1}^{n} S_l(t) y_l,$$

where $S_l(t)$ weights the l^{th} discrete data value in order to generate the fit to y_j. Intuitively, we expect $S_l(t)$ to be large when t is close to the associated evaluation point t_l and small otherwise. In matrix notation $\widehat{\mathbf{x}}(t) = \mathbf{S}(t)\mathbf{y}$, where $\widehat{\mathbf{x}}(t)$ is a column vector containing the values of the estimate of the function x at each sampling point t_j. In the standard least squares case we clearly have that $\mathbf{S}(\mathbf{t}^*) = \mathbf{\Phi}^* (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top$.

Remark 4.1. We can clearly view $\mathbf{S}$ in two ways. By fixing $\mathbf{t}$, $\mathbf{S}(\mathbf{t})$ is a finite dimensional matrix, i.e a linear operator between $\mathbb{R}^J$ and $\mathbb{R}^K$ where $K = |\mathbf{t}|$. This viewpoint allows us to do smoothing without introducing machinery from infinite dimensional analysis, however it does become cumbersome. As we shall discuss in the next chapter, a more rewarding viewpoint is to view $\mathbf{S}$ as a linear map between $\mathbb{R}^J$ and a function space $\mathcal{H}$.

It turns out that $\mathbf{S}(t)$ is a projection matrix, i.e. it has the idempotency property that $\mathbf{S}(t)\mathbf{S}(t) = \mathbf{S}(t)$. Intuitively, this map returns the vector $\widehat{\mathbf{y}}$ lying within the vector space spanned by the columns of $\mathbf{\Phi}$ which is closest to the provided data vector $\mathbf{y}$. More specifically, the residual vector $\mathbf{e} = \mathbf{y} - \widehat{\mathbf{y}}$ is orthogonal to the fit vector $\widehat{\mathbf{y}}$.

Often, the standard model for error is not realistic, for example due to non-stationarity and/or auutocorrelated errors. In this situation, it might be realistic to introduce a weighting of residuals, leading to the following least squares loss:

$$(\mathbf{y} - \mathbf{\Phi}\mathbf{c})^\top \mathbf{W} (\mathbf{y} - \mathbf{\Phi}\mathbf{c}) .$$

If the covariance matrix $\mathbf{\Sigma}$ for the residuals ε_j is known, then we can choose $\mathbf{W} = \mathbf{\Sigma}^{-1}$. If not, there are various ways in which one can estimate $\mathbf{\Sigma}$. The weighted least squares estimate is then given by

$$\widehat{\mathbf{c}} = (\mathbf{\Phi}^\top \mathbf{W}\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top \mathbf{W}\mathbf{y}.$$

We can define an associated linear smoothing operator $\mathbf{S}_W$ in an analogous fashion.

4.2 Choosing the number of basis elements

Having chose a specific functional, the next question is how do we choose the number of basis elements M in the expansion (2). If M is small, then we run the risk of missing some important details of the smooth function x we are trying to estimate. Taking M larger, we fit the data better, but we then risk also fitting noise or variation that we wish to ignore.

To make this tradeoff precise, consider the *mean squared error*

$$\mathrm{MSE}[\widehat{x}(t)] = \mathbb{E}\left[\widehat{x}(t) - x(t)\right]^2 ,$$

which can be decomposed into

$$\mathrm{MSE}[\widehat{x}(t)] = \mathrm{Bias}[\widehat{x}(t)]^2 + \mathrm{Var}[\widehat{x}(t)].$$

Here the bias is given by $\mathrm{Bias}[\widehat{x}(t)] = x(t) - \mathbb{E}[\widehat{x}(t)]$. Clearly, the bias will converge to zero as the number of basis elements involved goes to infinity, however, this will also give rise to increase variance of the estimator $\widehat{x}(t)$. Indeed, smoothing is achieved by keeping the number of basis elements in the expansion low. The relation between mean-square error, bias and variance suggest that it would be worthwhile to tolerate a little bias if the result is a big reduction in sampling variance. Due to this trade-off and the discreteness of M, tuning the number of basis elements automatically is not straightforward, and poses some challenges. This is why regularised smoothing methods are often preferred.

4.3 Computing Confidence Intervals: Simultaneous and Pointwise Confidence Intervals

If the observational noise in (2) given by $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_J)$ is assumed to be $\mathcal{N}(0, \mathbf{\Sigma})$ distributed then the estimated coefficients $\widehat{\mathbf{c}}$ using weighted least squares (with weight matrix $\mathbf{W}$) will have covariance

$$\mathrm{Cov}[\widehat{\mathbf{c}}] = (\mathbf{\Phi}^\top \mathbf{W}\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top \mathbf{W}\mathbf{\Sigma}\mathbf{W}\mathbf{\Phi}(\mathbf{\Phi}^\top \mathbf{W}\mathbf{\Phi})^{-1}.$$

In the observational noise covariance is a priori known, then it is natural to choose $\mathbf{W} = \mathbf{\Sigma}^{-1}$, in which case we obtain the familiar expression:

$$\mathrm{Cov}[\widehat{\mathbf{c}}] = (\mathbf{\Phi}^\top \mathbf{W}\mathbf{\Phi})^{-1}.$$

Since the noise is Gaussian, we can compute the sampling variance of the the fit to the data defined. Indeed one can readily check that

$$\mathrm{Var}[\widehat{\mathbf{y}}] = \mathbf{\Phi}\mathrm{Var}[\mathbf{c}]\mathbf{\Phi}^\top.$$

Confidence limits are typically computed by adding and subtracting a multiple of the standard errors $\sqrt{\mathrm{Var}[\widehat{x}(t)]}$, that is, the square root of the sampling variances, to the actual fit. Note that we must make a crucial distinction here: confidence limits computed in this fashion are called *point-wise* because they reflect confidence regions for fixed values of $t$rather than regions for the curve as a whole. More specifically, the pointwise $1-\alpha$-confidence interval $(\widehat{x}(t) - c(t), \widehat{x}(t) + c(t))$ satisfies the condition that

$$\mathbb{P}\left[\widehat{x}(t) - c(t) \leq x(t) \leq \widehat{x}(t) + c(t)\right] = 1 - \alpha,$$

where $\widehat{x}(\cdot)$ is the estimate of the unknown function $x(\cdot)$. The simultaneous coverage probability of a collection of confidence intervals is the probability that all of them cover their corresponding true values simultaneously. A *simultaneous* $1-\alpha$ *confidence interval* $(\widehat{x}(t) - c(t), \widehat{x}(t) + c(t))$ satisfies the condition that:

$$\mathbb{P}[]\widehat{x}(t) - c(t) \leq x(t) \leq \widehat{x}(t) + c(t) \text{ for all } t \in [0,1]] = 1 - \alpha.$$

In nearly all cases, a simultaneous confidence band will be wider than a pointwise confidence band with the same coverage probability. Simultaneous confidence interval are often calculated using simulation methods.

4.4 Kernel-based smoothing

Intuitively, we would expect a prediction of $x(t)$ at t to depend most strongly on the observations near t. This feature is an implicit property of the estimators we have considered so far. We can also consider linear smoothers where the local dependence is made more explicit by means of local weight functions. More specifically, a prediction at a given point is assumed to be linear combination of the form

$$\widehat{x}(t) = \sum_{j}^{M} S_j(t) y_j,$$

for some suitably defined weight functions S_j. Probably the most popular kernel estimator the Nadaraya-Watson estimator

$$S_j(t) = \frac{k((t_j - t)/h)}{\sum_{r=1} k((t_r - t)/h},$$

where k is a kernel function. This kernel function is designed to have most of its mass concentrated close to 0, and to either decay rapidly or disappear entirely for $|u| \geq 1$. Commonly used kernels include the Gaussian radial basis function kernel

$$k(u) = \frac{1}{\sqrt{2h^2\pi}} e^{-u^2/(2h^2)}.$$

The primary determinant of the degree of smoothing is the bandwidth h, rather than the number of basis functions used. The bandwidth controls the balance between two considerations: bias and variance in the estimate. Small values of h imply that the expected value of the estimate $\widehat{x}(t)$ must be close to the true value $x(t)$, but the price we pay is in terms of the high variability of the estimate, since it is be based on comparatively few observations. On the other hand, variability can always be decreased by increasing h, although this is inevitably at the expense of higher bias, since the values used cover a region in which the function's shape varies substantially. There is a variety of data-driven automatic techniques for choosing an appropriate value of h, usually motivated by the need to minimize mean squared error across the estimated function. Unfortunately, none of these can always be trusted, and the problem of designing a reliable data-driven bandwidth selection algorithm continues to be a subject of active research and considerable controversy. Our own view is that trying out a variety of values of h and inspecting the consequences graphically remains a suit- able means of resolving the bandwidth selection problem for most practical problems.

Related to the notion of kernel estimators is that of localized basis function estimators

The basic idea is to relax criterion (2) to give a localized measure of error as follows:

$$L_w(\mathbf{y} \mid \mathbf{c}) = \sum_{j=1}^{n} w_j(t) \left[y_j - \sum_{k=1}^{K} c_k \phi_k(t_j) \right]^2 .$$

Typically the weight functions $w_j(t)$ are typically constructed from the kernel function as $w_j(t) = k((t_j - t)/h)$. The loss can be expressed in matrix form as

$$L_w(\mathbf{y} \mid \mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})^\top \mathbf{W}(t) (\mathbf{y} - \mathbf{\Phi c}) ,$$

with $\mathbf{W}(t) = \mathrm{diag}[w_1(t), \ldots, w_n(t)]$.

4.5 Regularisation and Roughness Penalties

As we have seen in this section, we can adjust the tradeoff between bias and variance by increasing the number of basis elements. The discrete nature of the the number of elements is inconvenient from an optimisation and tuning point of view. Kernel smoothing approaches circumvent this issue, but they are seldom optimal solutions to an explicit statistical problem, such as minimizing a measure of total squared error. *Regularization* introduces an alternative approach to enforcing smoothness of the inferred functions by introducing a roughness penalty. Regularization methods are based on optimizing a fitting criterion that defines what a smooth of the data is trying to achieve. But here the precise meaning of "smooth" is expressed explicitly at the level of the criterion being optimized, rather than implicitly in terms of the number of basis functions being used.

One such approach to quantifying roughness of a function is in terms of its "curvature", i.e. it's second derivative of the function x, $Dx(t)$ In particular we can define the penalty

$$P_2(x) = \int |D^2 x(s)|^2 \, ds.$$

Highly variable functions can be expected to yield high values of $P(x)$ because their second derivatives are large over at least some of the range of interest. We can then define the regularised sum of squared errors loss as:

$$L_\lambda(\mathbf{y}\,|\,x) = (\mathbf{y} - x(\mathbf{t}))^\top \mathbf{W}\,(\mathbf{y} - x(\mathbf{t})) + \lambda P_2(x), \tag{4}$$

where $x(\mathbf{t}) = (x(t_1), \ldots, x(t_M))^\top$ and $\mathbf{W} \in \mathbb{R}^{M\times M}$ is a positive definite weighting matrix. Our estimate of the function is obtained by finding the function x that minimizes $x \to L_\lambda(\mathbf{y}\,|\,x)$ over the space of functions x for which $P_2(x)$ is defined. More generally, we can define

$$P_k(x) = \int |D^k x(s)|^2\, ds,$$

using a derivative of arbitrary order k.

Remark 4.2. Solutions to problems that involve optimizing with respect to functions rather than with respect to parameters are called variational problems.

The parameter λ is a smoothing parameter that measures the rate of exchange between fit to the data, as measured by the residual sum of squares in the first term, and variability of the function x, as quantified by $P_2(x)$ in the second term. As λ becomes larger and larger, functions which are not linear must incur a more and more substantial roughness penalty through the term $P_2(x)$. On the other hand, for small λ the curve tends to become more and more variable since there is less and less penalty placed on its roughness.

Recall that, without a roughness penalty, the coefficient vector $\mathbf{c}$ in the expansion

$$\widehat{x}(t) = \sum_{m=1}^{M} c_m \Phi_m(t) = \mathbf{\Phi}(t)\mathbf{c}$$

where $\mathbf{c}$ is the K-vector of coefficients and $\Phi(t)$ is the K-vector of basis functions. This has solution

$$\widehat{\mathbf{c}} = (\mathbf{\Phi}^\top \mathbf{W}\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top \mathbf{y}.$$

Assuming that $x(t) = \mathbf{\Phi}(t)\mathbf{c}$ we can express the roughness penalty in terms of $\mathbf{c}$ as follows

$$\begin{aligned} P_m(x) &= \int |D^m x(s)|^2\, ds \\ &= \int |\mathbf{c}^\top D^m \mathbf{\Phi}(s)|^2 \\ &= \mathbf{c}^\top \left[\int (D^m\mathbf{\Phi}(s))(D^m\mathbf{\Phi}(s))^\top\, ds\right]\mathbf{c} \\ &= \mathbf{c}^\top \mathbf{R}\mathbf{c}, \end{aligned}$$

where

$$\mathbf{R} = \int D^m\mathbf{\Phi}(s) D^m\mathbf{\Phi}(s)^\top\, ds.$$

Note that $\mathbf{R}$ involves integrals of the derivatives of the basis elements over the domain. Generally, these will only be available via numerical quadrature (unless it is a particularly nice basis like spline or Fourier). The penalised sum of squared loss is given by

$$L_\lambda(\mathbf{y}\,|\,\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi}\mathbf{c})^\top \mathbf{W}\,(\mathbf{y} - \mathbf{\Phi}\mathbf{c})) + \lambda \mathbf{c}^\top \mathbf{R}\mathbf{c}. \tag{5}$$

Finding the coefficients $\mathbf{c}$ which minimise this loss is straightforward, and is given by

$$\widehat{\mathbf{c}} = (\mathbf{\Phi}^\top \mathbf{W} \mathbf{\Phi} + \lambda \mathbf{R})^{-1} \mathbf{\Phi}^\top \mathbf{y}.$$

We can then use the penalised smooth to predict values of the curve at new points:

$$\widehat{x}(t) = \mathbf{\Phi}(t)(\mathbf{\Phi}^\top \mathbf{W} \mathbf{\Phi} + \lambda \mathbf{R})^{-1} \mathbf{\Phi}^\top \mathbf{y} = \mathbf{S}_\lambda(t)\mathbf{y}.$$

Comparing $\mathbf{S}_\lambda$ expression with $\mathbf{S}$ in the un-penalised case shows us that the only change is the addition of $\lambda\mathbf{R}$. The more general operator $\mathbf{S}_\lambda$ can be called a sub-projection operator because, unlike the projection operator, the sub-projection does not satisfy the idempotency relation, since $\mathbf{S}_\lambda \mathbf{S}_\lambda \neq \mathbf{S}_\lambda$.

Expressions of the form (5) occur often in statistics where linear models are used. In Bayesian models for regression the prior covariance matrix would appear in the place of $\mathbf{R}$. Indeed, we can think of the roughness penalty as analogous to the logarithm of a prior density, just as the error sum of squares term is, except for a scale factor, the logarithm of a likelihood. This is effectively a functional form of *ridge regression.*

When we fit data using a roughness penalty instead of least squares, we switch from defining the smooth in terms of degrees of freedom K to defining the smooth in terms of the smoothing parameter λ. One widely adopted strategy to choosing λ is using *cross-validation.* The idea of cross-validation is based on the following process: we choose a data point y_k. We train the model on the remaining data $\mathbf{y}_{-k}$, and attempt to predict y_k. This is done repeatedly for different data points and for different values of λ. We then choose the value λ^* that attains the minimum. Cross-validation can be used in a wide range of situations, and in effect rests only on the assumption that observations are relatively independent of one another. However, the method has two problems. First, it is usually computationally intensive, and not the sort of thing that would be feasible for sample sizes in the thousands. However, there are specific situations in which some computational tricks can be used to reduce the computational burden. The second problem is that minimizing CV can lead to under- smoothing the data because the method tends too often to favor fitting noisy or high-frequency types of variation that we would prefer to ignore. The *generalized cross-validation* approach of Craven and Wahba [1978] is a frequently adopted alternative which avoids the need to re-fit the model n times. It involves minimising a GCV score $GCV(\lambda)$ over the space of admissible regularisation parameter λ. Using the notation of the previous section the score is given by

$$GCV(\lambda) = \frac{J^{-1} \sum_{j=1}^{J} (y_j - \mathbf{S}_\lambda(t_j))^2}{(1 - J^{-1}\mathrm{trace}(\mathbf{S}_\lambda(\mathbf{t})))^2},$$

where $\mathbf{t} = (t_1, \ldots, t_J)$. The minimization of GCV with respect to λ will inevitably involve trying a large number of values of λ, whether grid-search or a numerical optimization algorithm is used. However, various extensions and variants of this approach have been proposed which greatly speed up the optimisation process, see Gu and Kim [2002].

4.6 Choice of Basis

4.6.1 Polynomial and Spline Bases

The most familiar basis function system is the collection of monomials from which power serieis can be constructed

$$\phi_0 = 1, \phi_1(t) = t, \phi_2(t) = t^2, \ldots, \phi_k(t) = t^k, \ldots.$$

Linear combinations of $\phi_0, \ldots, \phi_K$ yields the set of all polynomials up to degree K. Note that there are many possible functional bases which span the space of polynomials, each with different properties. In the next chapter we shall discuss the *Hermite polynomials* which possess very useful orthogonality properties.

For open-ended data, *spline functions* have replaced the use of polynomials which are substantially more flexible, and offer more stable function representations. Given an interval $I = [0, 1]$ we subdivide it into L subintervals separated by values $\tau_l,$, $l = 1, \ldots, L-1$ known as *knots*. Over each interval, a spline is a polynomial of specified order m. Adjacent polynomials join up smoothly at the breakpoint which separates them for splines of order greater than one, so that the function values are constrained to be equal at their junction. Moreover, derivatives up to order $m2$ must also match up at these junctions. For a cubic spline (which has order four), the second derivative is a polygonal line and the third derivative is a step function. If there are no interior knots, the spline reduces to being a simple polynomial.

Splines arise very naturally in functional approximation. Indeed suppose we are using the penalisied sum of squared errors criterion (4) for smoothing. Suppose we make no assumptions about function x except that it has a second derivative. It is well known (see de Boor [2001]) that the curve that minimises (4) must be a cubic spline at the data points $\mathbf{t} = (t_1, \ldots, t_M)^\top$.

So how does one express a spline function as a linear combination of a set of function basis? The B-spline basis system developed by de Boor (2001). Given knots $\tau_1, \ldots, \tau_{L-1}$ the B-spline basis is constructed recursively as follows:

$$B_{i,0}(x) := \begin{cases} 1 & \text{if } \tau_i \leq x \leq \tau_{i+1} \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$B_{i,k}(x) = \frac{x - \tau_i}{\tau_{i+k} - \tau_i} B_{i,k-1}(x) + \frac{\tau_{i+k+1} - x}{\tau_{i+k+1} - \tau_{i+1}} B_{i+1,k-1}(x).$$

Application of the recursion formula with the knots at $(0, 1, 2, 3)$ gives the basis for the uniform B-spline of order 3:

$$\begin{aligned} B_1 &= x^2/2, \quad 0 \leq x < 1 \\ B_2 &= (-2x^2 + 6x - 3)/2, \quad 1 \leq x < 2 \\ B_3 &= (3 - x)^2/2, \quad 2 \leq x < 3 \end{aligned}$$

One can show that any spline function of order m on a set of knots can be expressed as a linear combination of B-splines $\sum_i c_i B_{i,n}(x)$. In `R` there are various packages which implement spline smoothing and interpolation. In Figure 4 we plot an example of a cublic B-spline basis having 6 degrees of freedom (equivalent to 3 uniformly distributed knots).

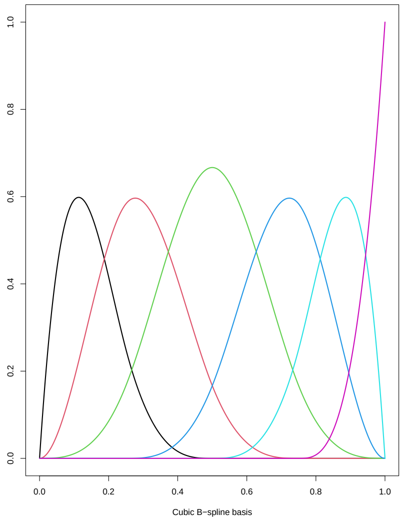

Figure 4: Plot of B-Spline cubic spline basis with 6 degrees of freedom.

```
spline.basis=create.bspline.basis(rangeval=c(0,10), nbasis=5)
plot(spline.basis, lty=1, lwd=2)
```

4.6.2 Fourier and Sinusoidal Bases

Another well known basis expansion is provided by the Fourier series:

$$\widehat{x}(t) = c_0 + c_1 \sin(\omega t) + c_2 \cos(\omega t) + c_3 \sin(2\omega t) + c_4 \cos(2\omega t) + \ldots,$$

defined by the basis $\phi_0(t) = 1$, $\phi_{2r-1}(t) = \sin(r\omega t)$ and $\phi_{2r}(t) = \cos(r\omega t)$. This basis is periodic and the parameter ω determines the period $2\pi/\omega$. If the values of t_j are equally spaced on I and the period is equal to the length of the interval, then the basis is *orthogonal* in the sense that $\mathbf{\Phi}^\top\mathbf{\Phi}$ is diagonal, and can be made equal to the identity by dividing the basis functions by $\sqrt{N}$ for $j = 0$ and $\sqrt{N/2}$ for all other j. When N is a power of 2 and the arguments are equally spaced, the *Fast Fourier Transform* makes it possible to find all the coefficients extremely efficiently. In thisi case we can evaluate the coefficients c_k and all n smooth values $x(t_j)$ in $O(N \log N)$ operations. While Fourier series are widely used by statisticians, it is worth acknowledging their limitations: Fourier series generally yield expansions which are uniformly smooth, making them inappropriat for data known to arise from functions with suspected discontinuities. In addition, all linear combinations of fourier basis are periodic too. Ideally, the periodicity of the Fourier series should be reflected to some degree in the data, as is certainly the case for many important situations.

A drawback to the Fourier representation is that it the frequency components apply to the function as a whole. One approach to identify how the frequency components of a function are varying is to compute the fourier coefficients over short windows or sub-intervals, for example using Short Time Fourier Transforms. However, by doing so, we pay a price: we lose the orthogonality and economy of representation of the Fourier method. This motivates approaches based on *wavelets*.

The wavelet approach is based on representing the fuunction at different scales and resolutions. At a particular resolution, we approximate the function by a sum of scaling functions. The difference between the resolutions can be expressed by a sum of wavelet functions. For certain scaling and wavelet functions, this hierarchical or multiresolution representation can be constructed using the scaled versions of the same functions at each resolution. A detailed exposition of function smoothing using wavelets is beyond the scope of these notes, and we refer the interested reader to (Nason, 2008).

4.7 Some Practicalities: Curve Alignment and Registration

In many situations when analysing functional data, the observed curves have similar shapes, but they are in some way shifted. Without preprocessing, any statistical statements made may be misleading. Consider Figure 5 where we compute mean arising from noisy observations of 5 cosine curves with slightly differing phase shifts. We immediately see that amplitude of the mean is significantly unrepresentative of any of the cosine functions. In this situations, it makes sense to align the curves together before calculating any such statistics. Given that the curves are varying in phase, this realignment must be achieved by stretching or compressing the x-axis.

4.7.1 Landmark Warping

Let us revisit the example discussed in Figure 3 in Section 2. Consider the time the curves cross the zero line for the last time. This is the age when the acceleration drops to zero, so this is the age of the fastest growth before reaching the final height. After that age, the acceleration becomes negative, and so the speed of growth decreases. This crossing time corresponds to the ages between 10 and 15 years for individuals. It is clear that the mean curve here isn't particularly a good representation of the data. To align the data we need some *a priori* knowledge of which points along each curve should be identified. We call such points *landmark* and the procedure is known as *landmark registration.* This is commonly achieved using *time-warping.* For each curve, x_n we assume that there is a *time warping function* $h_n(t)$ which can stretch or shrink time, i.e. it is a map between 'reference' time t and the individual specific time $h_n(t)$. In this particular example, the curves must be chosen such that, if each girl hits maximal puberty growth at time t_n, then $h_n(t_n)$ is the same for each n. Mathematically, we assume that the observed curves $x_n(t) = x^*_n(h_n(t))$, where x^*_n are properly aligned curves. To respect the landmarks, we introduce the constraints that $h_n(0) = 0$, $h_n(1) = 1$ and $h_n(t_a) = t_n$ for some $0 < t_a < 1$. These three points define a unique quadratic function $t \to h_n(t)$ passing through these three points. If $x_1, \ldots, x_n$ are original curves, then $x_1 \circ h_1^{-1}, \ldots x_n \circ h_n^{-1}$ are the resulting *registered curves.*

4.7.2 Dynamic Time Warping

Sometimes it's not obvious how to choose the landmarks, this has motivated various registration methods which do not explicitly make use of landmarks. Dynamic Time Warping (DWT) is an early registration method applied to discrete sequences of phonemes. Sakoe and Chiba [1978] devised an insertion/deletion algorithm based on dynamic programming for finding the optimal warp between two sequences. DTW can be effective as a feature alignment method, as it provides a globally optimal solution, albeit on the restricted search space (piecewise-linear h on a fixed grid).

4.7.3 Continuous Registration Methods

The desire for finding smooth warping functions between family of curves has motivated *continuous registration methods*. A number of such methods have been developed, and the problem continues to be actively researched. The idea is to search through a family of sufficiently smooth monotonic functions and find one which minimises the phase discrepancy between all the curves. We shall assume the following paramtrisation of h:

$$h(t) = C_0 + C_1 \int_0^t e^{W(u)}\, du.$$

The constants C_0 and C_1 are fixed by the requirement $h(t) = t$ at the upper and lower extremities of the registration interval. Without loss of generality, suppose we wish to align the phase between a target curve x_0 and a sample registered curve $x(h(t))$. The idea is to choose h such that the functions x_0 and $x \circ h$ differ only in amplitude, i.e. they are parallel to each other. One approach is to minimise the maximum eigenvalue of the matrix

$$C(h) = \begin{pmatrix} \int x_0(t)^2\, dt & \int x_0(t) x \circ h(t)\, dt \\ \int x_0(t) x \circ h(t)\, dt & \int x \circ h(t)^2\, dt \end{pmatrix}$$

The optimisation problem then becomes to find W which minimises

$$\max \sigma\left(C(h)\right) + \lambda \int |D^m W(t)|^2\, dt,$$

where an additional regularisation term is imposed on W to enforce smoothness. The generalisation to registration of multiple curves can be achieved using the above loss function in a manner akin to taking Frechet or Karcher means. Continuous time warping is the subject of much evolving work. See Gervini and Gasser [2004] and Liu and Müller [2004] for thorough reviews of the literature including some theoretical issues. In Figure 6 we display the height growth acceleration curves originally shown in Fig 3 after continuous registration using the `register.fd` function in the `fda` package.

References

Peter Craven and Grace Wahba. Smoothing noisy data with spline functions. *Numerische mathematik*, 31(4):377–403, 1978.

C de Boor. A practical guide to splines, revised edition, 2001.

Jianqing Fan and Irene Gijbels. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, volume 66. CRC Press, 1996.

Ricardo Fraiman and Graciela Muniz. Trimmed means for functional data. *Test*, 10(2):419–440, 2001.

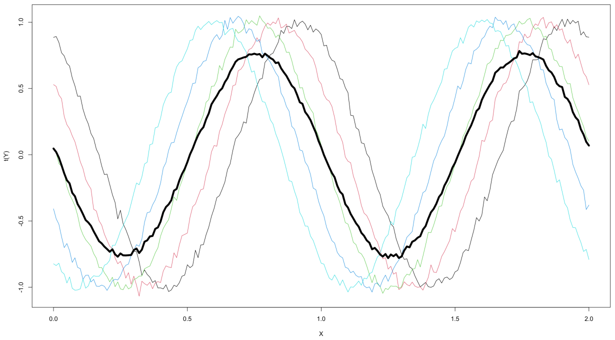

Figure 5: Noisy observations of multiple sinusoidal curves each having slightly shifted phases. The bold line represents the mean curve

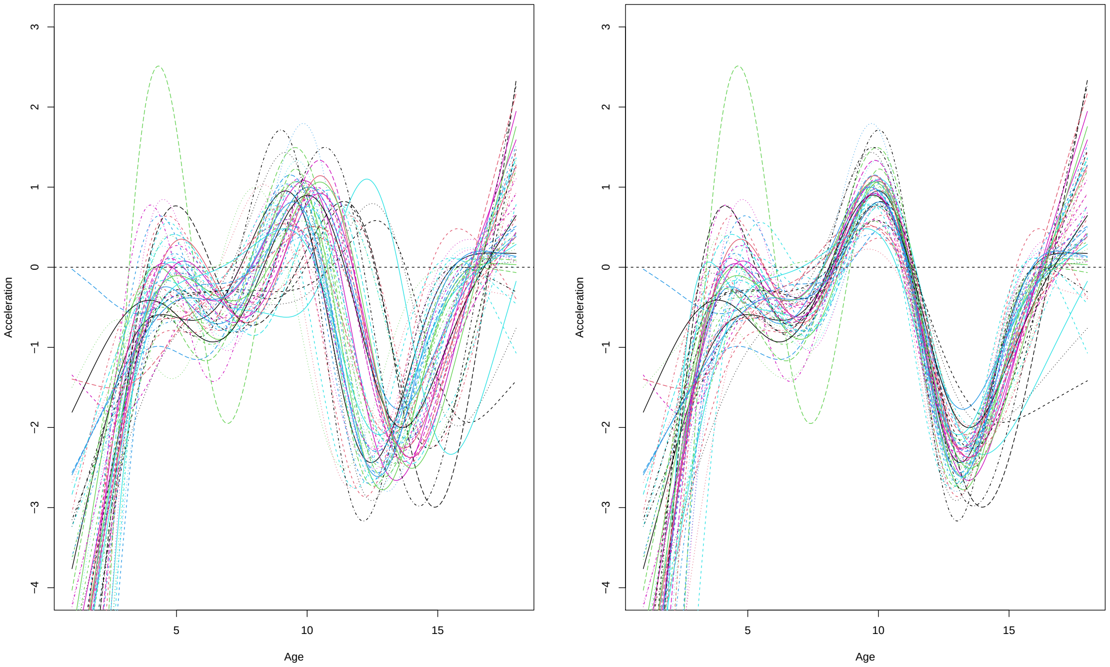

Figure 6: **Left:** Acceleration curves of the girls height dataset before alignment. **Right**: Acceleration curves of the girls height dataset after continuously registration using the `fda` default options.

Daniel Gervini and Theo Gasser. Self-modelling warping functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(4):959–971, 2004.

Chong Gu and Young-Ju Kim. Penalized likelihood regression: general formulation and efficient approximation. *Canadian Journal of Statistics*, 30(4):619–628, 2002.

Piotr Kokoszka and Matthew Reimherr. *Introduction to functional data analysis*. CRC Press, 2017.

Jong Soo Lee, Dennis D Cox, and Michele Follen. A two sample test for functional data. *Communications for Statistical Applications and Methods*, 22(2):121–135, 2015.

Xueli Liu and Hans-Georg Müller. Functional convex averaging and synchronization for time-warped random curves. *Journal of the American Statistical Association*, 99(467):687–699, 2004.

James Ramsay and Bernard W Silverman. Functional data analysis (springer series in statistics). 1997.

James O Ramsay and Bernard W Silverman. *Applied functional data analysis: methods and case studies*. Springer, 2007.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

Simon N Wood. Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114, 2003.