uc3m | Universidad Carlos III de Madrid

Master Degree in Statistics for Data Science

2023-2024

Master Thesis

"Semi-Functional Partial Linear Single-Index Model with Missing at Random responses"

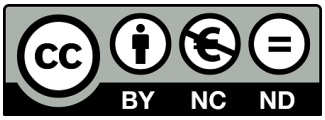
Marcos Crespo Díaz

Silvia Novo Díaz

Madrid, September 2024

AVOID PLAGIARISM

The University uses the **Turnitin Feedback Studio** for the delivery of student work. This program compares the originality of the work delivered by each student with millions of electronic resources and detects those parts of the text that are copied and pasted. Plagiarising in a TFM is considered a **Serious Misconduct**, and may result in permanent expulsion from the University.



SUMMARY

This thesis presents a novel approach to handling missing data in Functional Data Analysis (FDA) through the development of the SFPLSIM-MAR methodology and algorithm. This method combines the flexibility of Semi-Functional Partial Linear Single-Index Model (SFPLSIM) with existent work in handling Missing At Random (MAR) responses in order to address an existent gap in the scientific literature. The research introduces both existent strategies for imputation and estimation, Kernel and KNN nonparametric estimation techniques, rigorously tested in both simulation studies and real-world applications.

Keywords: Functional Data Analysis, Functional Regression, Semiparametric functional estimation, Scalar on Functional Regression, Missing at Random Responses, Semi-Functional Partial Linear Single-Index Model, Tecator, FDA, SFPLSIM, MAR, SFPLSIM-MAR.

DEDICATION

Thanks to my tutor, Silvia Novo Díaz, for her respect, patience and expertise throughout this challenging academic assignment.

To all my loved ones, for their constant support and for making all my successes their own. It has been an exciting journey.

"You said science was about admitting what we don't know."
Murphy Cooper. Interstellar.

CONTENTS

1. INTRODUCTION 1
1.1. Functional Regression 2
2. THE MODEL 5
2.1. Model estimation procedures 5
2.1.1. Nonparametric functional estimation 6
2.1.2. Model parameter estimation 9
2.2. Missing at random responses 10
2.3. SFPLSIM-MAR responses estimation 11
3. SIMULATION STUDY 13
3.1. Design 13
3.2. Results 15
4. REAL CASE STUDY 19
4.1. The Tecator Dataset 19
4.2. Methodology 20
4.3. Results 22
5. CONCLUSIONS AND FUTURE WORK 24
BIBLIOGRAPHY 26
APPENDIX . . . . .

LIST OF FIGURES

3.1 Example of $X(t)$ generation 13
3.2 Metrics comparison by size and method 15
3.3 Metrics comparison by linear predictors covariance 17
3.4 Metrics comparison by signal to noise ratio 18

4.1 Tecator absorbance curves 20
4.2 Outlier detection in Tecator dataset with 22

LIST OF TABLES

3.1 MSEP comparision between Kernel and KNN estimation 16

3.2 MAR ratios comparison in imputed model MSEP 17

3.3 MSEP ratios comparison 18

4.1 Comparison of Kernel and KNN estimation for different cases in Tecator without Outliers 22

1 Aggregated Metrics - Kernel Estimation . . . . . .

2 Aggregated Metrics - KNN estimation . . . . . .

3 Comparison of Kernel and KNN estimation for different cases in Tecator with Outliers . . . . . .

1. INTRODUCTION

During the age of big data and advanced computation, the nature of the collected data has evolved significantly. Traditional data analysis methods often assume scalar data points or vector observations. However, many modern applications involve data that are inherently continuous, such as time series, growth curves, or spatial data. These types of data are more naturally and accurately represented as functions over a continuous domain, leading to the growing field of Functional Data Analysis (FDA). Instead of treating each point as an isolated observation, FDA considers the entire trajectory of the variable as the observation. This holistic perspective allows for more nuanced analysis and interpretation than traditional methods.

The term 'Functional Data Analysis' was coined in 2005 by J. O. Ramsay and B. W. Silverman in an homonymous foundational work (Ramsay and Silverman, 2005) and since then, FDA has found applications in a wide range of fields, such as medicine, finance, environmental science and engineering (Ferraty and Vieu, 2006). Key components of FDA include the representation of functional data using basis functions, smoothing techniques to handle noisy observations, and statistical methods (either derived from multivariate analysis or originally proposed ones) like Functional Principal Component Analysis (FPCA) or Functional Regression. This methods were introduced gradually since the decades of 1940s and were not taken under the wings of FDA until mid 2000s. At present, we can identify a trend in the FDA scientific literature that focuses on finding more and more real world applications, developing increasingly complex models. To this use, Functional Regression provides very powerful tools.

As will be revealed later on, the main goal of a regression model is explaining the relationship between some random variables. The main problem with functional regression, and FDA in general, has to do with dimensionality. The idiosyncrasy of a function makes it infinite dimensional, and this results in complex handling, computation and estimation. It is for this reason that many of the efforts of the scientific community have been to improve in the way infinite dimensional objects are introduced in the typical regression frameworks. Single-Index Models (SIM), which aim to reduce dimensionality by projecting high-dimensional data onto a single-index, were proposed at the end of last century (foundational work was proposed by Hardle et al., 1993). Building up on this, partial linear models (PLM) (Speckman, 1988) were introduced to accommodate both linear and non-linear relationships within the data. Note how this methods were introduced to cope with high dimensional data, not necessarily functional data. Combining this two approaches to a generalised response, then the Generalised Partially Linear Single-Index Model (GPLSIM) was proposed (Carroll et al., 1997). Naturally, the above mentioned models were adapted to functional data, known as Functional Single-Index Model (FSIM) (Ferraty et al., 2003) and Semi-Functional Partial Linear Model (SFPLM) (Aneiros-Pérez

and Vieu, 2006). Combining this two approaches is that the Semi-Functional Partial Linear Single-Index Model (SFPLSIM) (Wang et al., 2016), the object of study of this document, has its genesis.

In statistics, missing data, or missing values, arise when a data point is absent for a particular variable in an observation. The Missing at Random (MAR) responses scenario is a specific type of missing data that has been widely addressed by other statistical models. Since SFPLSIM hasn't been applied yet in this particular case, in this thesis, we will try to apply the main Missing at Random responses techniques to the SFPLSIM in order to ensure its viability and unlocking its usage in real life applications, creating the SFPLSIM-MAR. This will need developing new methodologies by the generalisation of existent work, as well as a great deal of simulations and tests.

The primary objective of this thesis is to provide a comprehensive, intuitive and rigorous exposition of the SFPLSIM, focusing on its theoretical aspects and estimation methods. Furthermore, we aim to introduce the necessary notation and procedures in order to successfully implement the SFPLSIM-MAR, to then evaluate the model in different use cases.

To achieve these objectives, we will now give a brief introduction or 'crash course' on functional regression. Then, on Chapter 2 we will first delve into the theoretical aspects of SFPLSIM, outlining its formulation and the main assumptions it takes. This will be followed by a detailed discussion on the estimation procedures, incorporating both existing techniques and novel approaches tailored for MAR responses. In Chapter 3, the theoretical contributions will be complemented by an extensive simulation study, applying our method for the first time and drawing as much conclusions as we can. Lastly, Chapter 4 presents a real-world case study, illustrating its practical application to everyday data.

All the code and computations needed in the making of this thesis has been developed using the latest R programming language available version in July, 2024. For reproducibility, all the code and requirements are left in this github repository.

1.1. Functional Regression

Regression models are the bread and butter of statistical learning, and FDA provides a very interesting setting for this endeavour. We now present the basic knowledge the reader may need in order to fully understand the model in Chapter 2.

The raison d'être of regression problems is obtaining the relationship between some informative variables (predictors) and a variable of interest (response) given a set of observed values (sample).

All regression problems can be expressed in a general form: Given a response R, a set of predictors $X_1, \ldots, X_n$, the general model is:

$$R = m(X_1, \ldots, X_n) + \varepsilon.$$

Here $m(X_1, \ldots, X_n)$ is the relationship between the response and the predictors and it is called the regression function (typically = $\mathbb{E}[R|X_1, \ldots, X_n]$). ε is a random variable with $\mathbb{E}[\varepsilon] = 0$ that it is called the error term.

If we make these random variables to be functions we end up with a set of very interesting Functional Regression problems:

- Scalar response and functional predictors: Scalar on Function (SoF) regression.
- Functional response and scalar predictors: Function on Scalar (FoS) regression.
- Both functional predictors and response: Function on Function (FoF) regression.

Each of the previous settings derive in a lot of different regression models, each of them with their own characteristics. The Semi-Functional Partial Linear Single-Index Model belongs to the Scalar on Function set of problems.

Semi Parametric Scalar on Function regression

The mathematical skeleton of a univariate SoF regression problem may be the following:

Let $Y \in \mathbb{R}$ be a scalar single random variable. Let $\mathcal{X} \in \mathcal{H}$ be a functional predictor in a Hilbert space $\mathcal{H}$ (with inner product $\langle \cdot, \cdot \rangle$, typically $L^2(\mathbb{R})$). We assume that we have observed pairs of data $\{(Y_i, \mathcal{X}_i), i = 1, \ldots, n\}$ that are independently and identically distributed (i.i.d.) represented as $(\mathbf{Y}, \mathcal{X})$, the sample. Regression goal is to estimate $m(\mathcal{X}) = \mathbb{E}(\mathbf{Y}|\mathcal{X})$ given a random error ε with $\mathbb{E}(\varepsilon|\mathcal{X}) = 0$

In order to model this relationship, the regression function $m(\cdot)$, the literature on Functional Data Analysis has focused on three main regression strategies (Aneiros et al., 2019):

1. Parametric Regression: The Functional Linear Model (FLM) assumes a linear relationship between the functional predictor and the response (Cardot et al., 1999). This model is the immediate generalisation to FDA of standard Linear Regression. It is easy to estimate and provides interpretable results through β, but it relies on the assumption of linearity, which might not hold in some applications. The model can be expressed as:
$$Y_i = \langle \beta, \mathcal{X}_i \rangle + \varepsilon_i, \quad i = 1, \ldots, n.$$
where β is an unknown function defined on $\mathcal{H}$ and $m(\cdot) = \langle \beta, \cdot \rangle$.
2. Nonparametric Regression: To overcome the linearity limitation of FLM, we can use a Functional Nonparametric Model (FNM), which relaxes the linear assumption and assumes a smooth relationship (Ferraty and Vieu, 2006):
$$Y_i = m(\mathcal{X}_i) + \varepsilon_i, \quad i = 1, \ldots, n.$$

In this case, $m(\cdot)$ is an unknown, smooth, non-linear functional that it is estimated using nonparametric techniques. While FNM is more flexible and reliable in practice, $m(\cdot)$ is harder to interpret and estimate.

3. Semi-parametric Regression: Combining both parametric and nonparametric regression, the functional single-index model (FSIM) assumption is that the regression function is the nonparametric estimation of the functional predictor projected onto a one-dimensional subspace (Ferraty et al., 2003):

$$Y_i = r\left(\langle \theta_0, \mathcal{X}_i \rangle\right) + \varepsilon_i, \quad i = 1, \ldots, n.$$

Here, we perform a projection via $\theta_0 \in \mathcal{H}$, the functional index summarising the information in $\mathcal{X}_i$ affecting Y_i. Then, $r(\cdot)$, the smooth, non-linear function extends FLM by allowing more flexibility through the power and flexibility of FNM. This model assumes that the functional information in Y, given by $\mathcal{X}$, can be (mostly) collected by a single direction θ_0. This information is then estimated by a nonparametric function.

Semi-parametric approaches also simplify the problem by reducing its dimensionality, making it easier to interpret and represent the results. As a sample of data gets larger in dimension, it becomes more and more sparse, which makes developing statistical processes more challenging (Vieu, 2018). By applying the ideas of FSIM in our usual functional Hilbert space $\mathcal{H}$, then the problem of estimating the infinite dimensional operator $m(\cdot)$ in FNM, is now transformed into estimating θ_0 and $r(\cdot)$, both functions in a 1-dimensional space. These models, are more interpretable than the FNM approaches but harder to estimate than the FLM. Some of the most common estimation techniques will be presented in Chapter 2.

2. THE MODEL

The model that we will be using in this thesis is derived from the FSIM model presented earlier. When not only the functional covariate but also some other explanatory (linear) covariates affect our scalar response, the Semi-Functional Partial Linear Single-Index Model takes a single-index component and combines it with a one (or more) scalar linear components. We now present the model formulation in Novo et al. (2021a).

Assume we have a statistical iid sample of n vectors with the following structure $(X_{i1}, \ldots, X_{ij}, \mathcal{X}_i, Y_i)$ $\forall i = 1, \ldots, n$, $\forall j \in \mathbb{N}$. We could represent this in column matrix notation as $(X_1, \ldots, X_j, \mathcal{X}, \mathbf{Y})$. X_j with $j = 1, \ldots, j$ and $\mathbf{Y}$ are real random variables and $\mathcal{X} \in \mathcal{H}$ is a functional random variable with inner product $\langle \cdot, \cdot \rangle$. If we wanted to create a (regression) model combining the effects of both the linear predictors and the functional variable on the response (setting of SoF regression); the semi-parametric approach seems like a fairly good idea thanks to all the previously mentioned advantages. The SFPLSIM states:

$$Y_i = X_{i1}\beta_{01} + \ldots + X_{ij}\beta_{0j} + r(\langle \theta_0, \mathcal{X}_i \rangle) + \varepsilon_i \quad \forall i = 1, \ldots, n. \tag{2.1}$$

Here ε is a random error component verifying $\mathbb{E}(\varepsilon | X_1, \ldots, X_j, \mathcal{X}) = 0$. The vector $\beta_0 \in \mathbf{R}^j$ and the functional direction (single-index) $\theta_0 \in \mathcal{H}$ are the model parameters and are unknown, as well as the smooth real valued function $r(\cdot)$. All of them are the estimation target of the model.

Please note how the scalar covariates are included in the model using a parametric (standard linear regression) approach. Obviously, the relation between the scalar covariates could be non-linear, and even non-parametric, but using linear regression in this component allow us to enhance interpretability of the model, and avoids bad dimensionality effects. If we used non-parametric regression in the scalar explanatory variables, the dimensionality of the problem will increase a lot, making the decrease in dimensionality attained by the semi-parametric functional regression useless.

2.1. Model estimation procedures

In order to estimate the unknown model components we will apply some different techniques. The main idea is to use Penalised Least Squares (PeLS) to estimate β_0 and θ_0 and then Kernel or KNN nonparametric estimation in order to estimate $m(\cdot)$. In practice, some modifications to this approach are necessary, and will be explained below.

In order to explain the model parameters estimation, we first present some nonparametric functional estimation basics.

2.1.1. Nonparametric functional estimation

Nonparametric estimation is a statistical technique used to estimate relationships or patterns in data without assuming a predefined form for the underlying model. Unlike parametric methods, which rely on specific assumptions, nonparametric approaches assume nothing and offer greater flexibility by allowing the data to shape the model.

F. Ferraty and P. Vieu are the main contributors to the field of nonparametric estimation in FDA. They proposed (Ferraty and Vieu, 2006) the functional extension of the classical Nadaraya-Watson kernel estimator for non-parametric scalar estimation.

This estimator is:

$$\widehat{r_h}(\chi) = \frac{\sum_{i=1}^{n} Y_i K\left(h^{-1} d\left(\mathcal{X}_i, \chi\right)\right)}{\sum_{i=1}^{n} K\left(h^{-1} d\left(\mathcal{X}_i, \chi\right)\right)}, \forall \chi \in \mathcal{H}. \tag{2.2}$$

Here $h \in \mathbb{R}^+$, the bandwidth, is the width of the window over which the kernel function is applied [1]. d is a general semi-metric [2] giving the proximity between functional data, and K is a real-valued asymmetrical kernel, usually the Epanechnikov.

However, in some other works (Novo et al., 2021a), a K-Nearest Neighbours (KNN) version of the estimator is proposed. In this variation, for each element χ within $\mathcal{H}$, the estimator is calculated solely based on the k observations in the sample that are the closest to it. The KNN estimator for $r(\cdot)$ is defined as follows:

$$\widehat{r_k^*}(\chi) = \frac{\sum_{i=1}^{n} Y_i K\left(H_{k,\chi}^{-1} d\left(\mathcal{X}_i, \chi\right)\right)}{\sum_{i=1}^{n} K\left(H_{k,\chi}^{-1} d\left(\mathcal{X}_i, \chi\right)\right)}, \forall \chi \in \mathcal{H}, \tag{2.3}$$

where $k \in \mathbb{Z}^+$ is the parameter indicating the number of neighbours taken into account. $H_{k,\chi} = \min\left\{h \in \mathbb{R}^+ : \sum_{i=1}^{n} 1_{B(\chi,h)}\left(\mathcal{X}_i\right) = k\right\}$, is the minimum bandwidth for which the neighbour of bandwidth h and centre χ $B(\chi, h) = \{z \in \mathcal{H} : d(\chi, z) \leq h\}$ contains k observations. The fact that the smoothing parameter $H_{k,\chi}$ depends on χ and k means that the KNN estimator is taking into account the location of each observation and also its surroundings, providing more adaptability than the Kernel estimator. Moreover, k is taken from a finite set $\{1, \ldots, n\}$, which provides a much more efficient selection than choosing the continuous parameter h.

In some models, like in the FSIM (and in SFPLSIM), the semi-metric depends on a parameter (the functional direction). We could 'combine' the selection of the tuning parameter with the selection of the regression parameter of the semi-metric (Novo and

[1] It controls the degree of smoothing in the estimate: a small bandwidth leads to less smoothing, capturing more details and noise in the data, while a large bandwidth results in more smoothing, which can reduce important data features but reduce noise.

[2] Natural generalisation of a metric to functional spaces. There is no need for functions to be topologically distinguishable.

Aneiros, 2024). If we considered the natural semi-metric depending on the functional direction θ:

$$d_\theta(\chi_1,\chi_2) = |\langle\theta,\chi_1\rangle - \langle\theta,\chi_2\rangle| = |\langle\theta,\chi_1-\chi_2\rangle| \quad \text{for } \chi_1,\chi_2 \in \mathcal{H},$$

now, we could substitute this expression in (2.2) and (2.3) obtaining:

$$\widehat{r}_{h,\theta}(\chi) = \frac{\sum_{i=1}^n Y_i K\left(h^{-1} d_\theta(\mathcal{X}_i,\chi)\right)}{\sum_{i=1}^n K\left(h^{-1} d_\theta(\mathcal{X}_i,\chi)\right)}, \quad \widehat{r}^*_{k,\theta}(\chi) = \frac{\sum_{i=1}^n Y_i K\left(H^{-1}_{k,\chi,\theta} d_\theta(\mathcal{X}_i,\chi)\right)}{\sum_{i=1}^n K\left(H^{-1}_{k,\chi,\theta} d_\theta(\mathcal{X}_i,\chi)\right)}, \quad \forall \chi \in \mathcal{H} \tag{2.4}$$

Note how these expressions now become statistics, as they depend on more than one parameter. Consequently, they cannot be directly calculated and need to be, once again, estimated.

It is typical to address the estimation of both θ (model parameter) and h or k (tuning parameters) using Leave One Out Cross Validation (LOOCV). Assuming the reader is familiarised with this kind of procedures, we now need to define the minimisation problems that will be optimised. This involves considering the following objective functions:

$$CV(h,\theta) = n^{-1}\sum_{j=1}^n \left(Y_j - \widehat{r}^{(-j)}_{h,\theta}\left(\mathcal{X}_j\right)\right)^2 \text{ and } CV^*(k,\theta) = n^{-1}\sum_{j=1}^n \left(Y_j - \widehat{r}^{*(-j)}_{k,\theta}\left(\mathcal{X}_j\right)\right)^2, \tag{2.5}$$

for Kernel and KNN estimators respectively. Minimising them respect to both parameters we obtain the following estimates:

$$\hat{\theta}_h = \arg\min_{\theta\in\Theta} CV(h,\theta), \quad \hat{\theta}^*_k = \arg\min_{\theta\in\Theta} CV^*(k,\theta), \tag{2.6}$$

$$\widehat{h} = \arg\min_{h\in\mathbb{R}^+} CV\left(h,\hat{\theta}_h\right), \quad \widehat{k} = \arg\min_{1\le k\le n} CV^*\left(k,\hat{\theta}_k\right), \tag{2.7}$$

where $\Theta \subset \mathcal{H}$. Computationally, a subset $\mathcal{I} \in \mathbb{R}^+$ is considered in grid form.

As the reader may have noticed, the key aspect of solving a functional nonparametric estimation is transformed into a minimisation problem. Computationally, this involves the subset $\Theta \in \mathcal{H}$ selection and then and 2 way minimisation procedure, which is a complex issue.

Several methods are used for the proper selection of Θ (Novo and Aneiros, 2024). The most broadly used method (Ait-Saïdi et al., 2008) uses B-Spline approximations of the functional directions. This way, the infinite-dimensional optimisation problem is reduced to a multivariate one involving the basis coefficients. The main idea is that if we took a B-Spline basis $\{e_1(\cdot),\ldots,e_d(\cdot)\}$, the functional directions can be now expressed as:

$$\theta(\cdot) = \sum_{j=1}^d \alpha_j e_j(\cdot) \text{ , where } (\alpha_1,\ldots,\alpha_d) \in \mathcal{V}.$$

Reducing the subset of functions to subset of coefficients makes the minimisation problem to be:

$$\hat{\theta}_h = \arg \min_{(\alpha_1,\ldots,\alpha_d)\in\mathcal{V}} CV(h,\theta), \quad \hat{\theta}^*_k = \arg \min_{(\alpha_1,\ldots,\alpha_d)\in\mathcal{V}} CV^*(k,\theta). \tag{2.8}$$

Complete algorithm

We now present the complete algorithm solving the estimation of the functional nonparametric component that will be used in the SFPLSIM. Essentially, this is the main way of solving a FSIM. Prior to this, a simplified version to the creation method of the eligible functional direction subset Θ is also given in order to make use of it whenever it is necessary.

Algorithm 1 Θ creation method

1: Choose the dimension of the B-spline basis d.
2: For each $(\beta_1,\ldots,\beta_d) \in C^d$, where $C = \{c_1,\ldots,c_J\} \subset \mathbb{R}^J$ denotes a set of J 'seed-coefficients', build the initial functional direction $\theta_{\text{init}}(\cdot) = \sum_{j=1}^{d} \beta_j e_j(\cdot)$.
3: For each θ_{init} from Step 2 that satisfies the condition $\theta_{\text{init}}(t_0) > 0$, where t_0 is a fixed value in the domain of $\theta_{\text{init}}(\cdot)$, compute $\langle \theta_{\text{init}}, \theta_{\text{init}} \rangle$ and construct $(\alpha_1,\ldots,\alpha_d) = \frac{(\beta_1,\ldots,\beta_d)}{\langle \theta_{\text{init}}, \theta_{\text{init}} \rangle^{1/2}}$.
4: Construct $\mathcal{V}$ as the set of vectors $(\alpha_1,\ldots,\alpha_d)$ obtained in Step 3.
Thus, the final set of eligible functional directions is given by
$\Theta = \left\{ \theta(\cdot) = \sum_{j=1}^{d} \alpha_j e_j(\cdot); (\alpha_1,\ldots,\alpha_d) \in \mathcal{V} \right\}$

The general scheme of the proposed algorithm is:

Algorithm 2 Kernel or KNN Nonparametric functional estimation

Require: a set of eligible directions Θ. (If not given use Algorithm 1) and a grid $\mathcal{I}$ or subset $\{1,\ldots,n\}$.
1: **for** $\theta \in \Theta$ **do**
2: **for** $h \in \mathcal{I}$ or $k \in \{1,\ldots,n\}$ **do**
3: Compute $CV(h,\theta)$ or $CV^*(k,\theta)$ like (2.5)
4: **end for**
5: **end for**
6: Select both θ and h or k as the 2 way minimiser of (2.8) and (2.7).
7: **return** The computation of $\widehat{r}_{h,\theta}(\chi)$ or $\widehat{r}^*_{k,\theta}(\chi)$ like (2.4).

Note how, since this method requires intensive computation, it is necessary to balance the size of Θ and the performance of the estimators. It is recommended (Ait-Saïdi et al., 2008) to use the seed-coefficients $C = \{1, 0, -1\}$. In the same paper we can find the condition $\theta_{\text{init}}(t_0) > 0$ to ensure the identifiability of the model, as well as unitary norm.

Also, the B-Spline basis can add more complexity depending on the spline degree as well as the interior knots used in the estimation.

2.1.2. Model parameter estimation

Now we have a complete (and arduous) way of estimating and computing nonparametric functional estimators, we have all the knowledge necessary to estimate the SFPLSIM.

As mentioned in 1.1, the main goal of a regression problem is to obtain the relation $m(X_1, \ldots, X_n)$ as a conditional expectation. In the SFPLSIM case, we have scalar and functional covariates mixed making the computation of $\mathbb{E}[R|X_1, \ldots, X_n, \mathcal{X}]$ not immediate at all. If we conceptually separate the expectation (and consequently the effect on the response) between the linear scalar part of the model, we would end up with $\mathbb{E}[R|X_1, \ldots, X_n]$ and $\mathbb{E}[R|\mathcal{X}]$, which appear to be much more mathematically manageable.

We first extract the effect of the functional covariate from the response and the predictors, this is:

$$Y_i - \mathbb{E}(Y_i|\langle\theta_0, \mathcal{X}_i\rangle) = (X_i - \mathbb{E}(X_i|\langle\theta_0, \mathcal{X}_i\rangle))^T\beta_0 + \varepsilon_i. \tag{2.9}$$

Note how $\mathbb{E}(Y_i|\langle\theta_0, \mathcal{X}_i\rangle)$ and $\mathbb{E}(X_i|\langle\theta_0, \mathcal{X}_i\rangle))$ are unknown. Therefore, after using FSIM Algorithm 2 to estimate both values we get $\hat{r}^{Y_i}_{\theta_0,h}$ and $\hat{r}^{X_i}_{\theta_0,h}$ or $\hat{r}^{Y_i}_{\theta_0,k}$ and $\hat{r}^{X_i}_{\theta_0,k}$ (depending on Kernel or KNN estimation). Without loss of generality, assume the Kernel estimation, then we can rewrite $Y_i - \hat{r}^{Y_i}_{\theta_0,h} = \tilde{\mathbf{Y}}_{\theta_0}$ and $X_i - \hat{r}^{X_i}_{\theta_0,h} = \tilde{\mathbf{X}}_{\theta_0}$ and end up with the following expression of an approximate linear model:

$$\tilde{\mathbf{Y}}_{\theta_0} = \tilde{\mathbf{X}}_{\theta_0}\beta_0 + \varepsilon. \tag{2.10}$$

Now this problem in (2.10) can be seen as a regular linear regression problem involving only the linear part of the model. There are a lot of different methods to solve linear regression, being one of the most general the Penalised Least Squares (PeLS) technique. This means minimising the following function over $\beta \times \theta$:

$$Q(\beta, \theta) = \frac{1}{2}(\tilde{Y}_\theta - \tilde{\mathcal{X}}_\theta\beta)^T(\tilde{Y}_\theta - \tilde{\mathcal{X}}_\theta\beta) + n\sum_{j=1}^{p_n}\mathcal{P}_{\lambda_{j_n}}(|\beta_j|),$$

given a penalisation function and rate $\mathcal{P}(\cdot)$ and λ_{j_n} respectively, if any. Note how this function is not convex so minimising it will not assure a absolute minimum but a local minimum. However, its existence is assured (Novo et al., 2021a). If no penalisation function is included, given a functional direction, then an explicit expression for β_0 can be attained. No penalisation term will be used from now on[3], so the function to be minimised

[3]Please note how the algorithm proposed in (2.3) can be adapted very easily to penalisation terms. Check Novo and Aneiros (2024) for a complete implementation of the SFPLSIM with penalisation,

is:

$$Q(\beta, \theta) = \frac{1}{2}(\tilde{Y}_\theta - \tilde{\mathcal{X}}_\theta \beta)^T (\tilde{Y}_\theta - \tilde{\mathcal{X}}_\theta \beta). \tag{2.11}$$

After solving this regression problem we finally have $\hat{\beta}_0$ and $\hat{\theta}_0$ than can be plucked in (2.1). The last component that needs estimation is the smooth function $r(\cdot)$, than can be estimated again using Kernel or KNN estimators once again. Just substituting Y_i with $Y_i - \mathbf{X}_i^T \hat{\beta}_0$ in (2.4).

2.2. Missing at random responses

Missing data is a common challenge in statistical analysis. The presence of missing data can significantly impact the reliability of some models. For all statistical procedures, there is a need to investigate its robustness against missing data, and develop the necessary machinery in order to rigorously address this issue.

The absence of data can arise due to nonresponse in surveys, errors or even data corruption. The probability of an observation being missing may be entirely random and be unrelated to the sample in which case it won't introduce any bias into the analysis, only reducing its sample size. There are some cases where the missingness depends on some observed data variable (imagine a medical study where older participants are less likely to respond to follow-up surveys: the missingness depends on age, an observed variable), and bias is introduced requiring some more complex adjusting.

Now we propose a novel theoretical framework that extends the Semi-Functional Partial Linear Single-Index Model (SFPLSIM) to accommodate scenarios where responses are Missing at Random (MAR): SFPLSIM-MAR. This adaptation builds upon existing methodologies for handling missing data in statistical models, generalising them to the functional data context of SFPLSIM. Our aim is to develop and present new scientific insights that incorporate the MAR mechanism, enhancing the model robustness and applicability, since missing responses are a common occurrence in real-world data.

The theoretical development of our proposed approach, including the mathematical formulation, estimation procedures and practical implementation is the following. This represents an advancement to the field of functional data analysis, now being able to address MAR responses in the context of SFPLSIM.

Lets consider a statistical iid sample of n vectors with the following structure $(Y_i, \delta_i, X_i, \mathcal{X}_i)$ $\forall i = 1, \ldots, n$. Let Y_i, X_i and $\mathcal{X}_i$ be in the same settings as in the standard SFPLSIM. Additionally, δ_i is a binary random variable with values 0 or 1. We assume that if $\delta_i = 0$ then Y_i is missing and if $\delta_i = 1$ then Y_i has been observed. We assume that δ and Y are conditionally independent given X and $\mathcal{X}$; this is:

$$P(\delta = 1|Y, X, \mathcal{X}) = P(\delta = 1|X, \mathcal{X}).$$

The idea is to impute the missing responses in order to 'fill' the model and estimate the final model with a complete sample. We fill the sample by applying the model estimation with the reduced sample and then making the imputation of the missing observations as if they were predictions from the reduced model.

First, we create auxiliary estimators in the subsample using the complete observations. This is:

$$\{(Y_i, X_i, \mathcal{X}_i) \text{ such that } \delta_i = 1, \forall i = 1, \ldots, n\}.$$

Now we solve the SFPLSIM regression model generated by this subsample and end up with the auxiliary estimators: $\tilde{\beta}$, $\tilde{\theta}_0$ and $\tilde{r}(\cdot)$.

Then, we can impute the missing responses with this estimators as if they were new predictions. This creates the complete sample:

$$\{(\mathcal{Y}_i, X_i, \mathcal{X}_i), \forall i = 1, \ldots, n\},$$

where $\mathcal{Y}_i = \delta_i Y_i + (1 - \delta_i)(X_i^T \tilde{\beta} + \tilde{r}(\langle \tilde{\theta}_0, \mathcal{X}_i \rangle)), \forall i \in \{1, \ldots, n\}$

This ends up creating a new SFPLSIM that can be estimated again, producing the final estimators: $\hat{\beta}$, $\hat{\theta}_0$ and $\hat{r}(\cdot)$.

2.3. SFPLSIM-MAR responses estimation

In this section, we present the algorithmic formulation of the SFPLSIM-MAR estimation procedure, which integrates the various techniques and methodologies discussed in the preceding sections. This new algorithm provides a robust framework for estimating responses in the presence of incomplete data, ensuring accurate and efficient estimation. The following expression outlines the step-by-step process of implementing the SFPLSIM-MAR estimation.

Algorithm 3 Kernel or KNN SFPLSIM-MAR fitting

Require: $\{\mathbf{Y}, \delta, \mathbf{X}, \mathcal{X})\}$, a grid $\mathcal{I}$ or a subset $\{1, \ldots, n\}$.

Create the auxiliary sample $\{(Y_i, X_i, \mathcal{X}_i)$ such that $\delta_i = 1, \forall i = 1, \ldots, n\}$.
Apply the Θ generation procedure as in Algorithm 1.
for $\theta \in \Theta$ **do**
  for $h \in \mathcal{I}$ or $k \in \{1, \ldots, n\}$ **do**

    1. Extract the effect of the functional covariate by $Y_i - \mathbb{E}(Y_i|\langle\theta, \mathcal{X}_i\rangle) = (X_i - \mathbb{E}(X_i|\langle\theta, \mathcal{X}_i\rangle))^T\beta_0 + \varepsilon_i$.

    2. Estimate $\mathbb{E}(Y_i|\langle\theta, \mathcal{X}_i\rangle$ and $\mathbb{E}(X_i|\langle\theta, \mathcal{X}_i\rangle$ with Algorithm 2 and obtain auxiliary $\tilde{r}^{Y_i}_{\theta,h}$ and $\tilde{r}^{X_i}_{\theta,h}$ or $\tilde{r}^{Y_i}_{\theta,k}$ and $\tilde{r}^{X_i}_{\theta,k}$ (depending on Kernel or KNN estimation).

    3. Rename and obtain problem $\tilde{\mathbf{Y}}_\theta = \tilde{\mathbf{X}}_\theta\beta_0 + \varepsilon$.

    4. Solve the above problem using scalar regression techniques, getting $\tilde{\beta}_0$.

  end for
  Select optimal $\tilde{h}$ or $\tilde{k}$ in expression (2.11).
end for
Select optimal $\tilde{\theta}_0$ in expression (2.11).
After getting $\tilde{\beta}_0$, $\tilde{h}$ or $\tilde{k}$ and $\tilde{\theta}_0$; estimate $r(\cdot)$ with Algorithm 2 (with Kernel or KNN estimation) substituting Y_i with $Y_i - \mathbf{X}_i^T\tilde{\beta}_0$. End up with auxiliary parameters: $\tilde{\beta}$, $\tilde{\theta}_0$ and $\tilde{r}(\cdot)$.
Impute the MAR responses with the auxiliary model and create the complete sample: $\{(\mathcal{Y}_i, X_i, \mathcal{X}_i), \forall i \in 1, \ldots, n\}$.
for $\theta \in \Theta$ **do**
  for $h \in \mathcal{I}$ or $k \in \{1, \ldots, n\}$ **do**

    1. Extract the effect of the functional covariate by $\mathcal{Y} - \mathbb{E}(\mathcal{Y}|\langle\theta, \mathcal{X}_i\rangle) = (X_i - \mathbb{E}(X_i|\langle\theta, \mathcal{X}_i\rangle))^T\beta_0 + \varepsilon_i$.

    2. Estimate $\mathbb{E}(\mathcal{Y}|\langle\theta, \mathcal{X}_i\rangle$ and $\mathbb{E}(X_i|\langle\theta, \mathcal{X}_i\rangle$ with Algorithm 2 and obtain $\hat{r}^{Y_i}_{\theta,h}$ and $\hat{r}^{X_i}_{\theta,h}$ or $\hat{r}^{Y_i}_{\theta,k}$ and $\hat{r}^{X_i}_{\theta,k}$ (depending on Kernel or KNN estimation).

    3. Rename and obtain the problem $\tilde{\mathcal{Y}}_\theta = \tilde{\mathbf{X}}_\theta\beta_0 + \varepsilon$.

    4. Solve the above problem using scalar regression techniques, getting $\hat{\beta}_0$.

  end for
  Select optimal $\hat{h}$ or $\hat{k}$ in expression (2.11).
end for
Select optimal $\hat{\theta}_0$ in expression (2.11).
After getting $\hat{\beta}_0$, $\hat{h}$ or $\hat{k}$ and $\hat{\theta}_0$; estimate $r(\cdot)$ again with Algorithm 2 (with Kernel or KNN estimation) substituting $\mathcal{Y}_i$ with $\mathcal{Y}_i - \mathbf{X}_i^T\hat{\beta}_0$.
Obtain the final estimated parameters: $\hat{\beta}$, $\hat{\theta}_0$ and $\hat{r}(\cdot)$.

3. SIMULATION STUDY

In this section, we present our simulation study, which aims to evaluate the performance of the Semi-Functional Partial Linear Single-Index Model with Missing at Random responses (SFPLSIM-MAR). This study involves generating synthetic datasets, inducing MAR missingness, imputing the missing values, and assessing the performance of the model in various conditions.

3.1. Design

We have generated samples of size n from the SFPLSIM expression in (2.1) for $j = 3$. This means generating $\mathcal{D} = \{X_{i1}, X_{i2}, X_{i3}, \mathcal{X}_i, Y_i\}_{i=1}^{n+25}$ attending to:

$$Y_i = X_{i1}\beta_{01} + X_{i2}\beta_{02} + X_{i3}\beta_{03} + r(\langle\theta_0, \mathcal{X}_i\rangle) + \varepsilon_i \quad \forall i = 1, \ldots, n. \tag{3.1}$$

The functional random variable $\mathcal{X}$ we have considered is:

$$\mathcal{X}(t) = a(x - 0.5)^2 + b,$$

where a and b are univariate random variables with $a \sim U[-2, 2]$ and $b \sim N[0, 1]$. Function domain is $t \in [0, 1]$ and it has been uniformly discretised over a grid of 100 time points. An example is shown in figure 3.1.

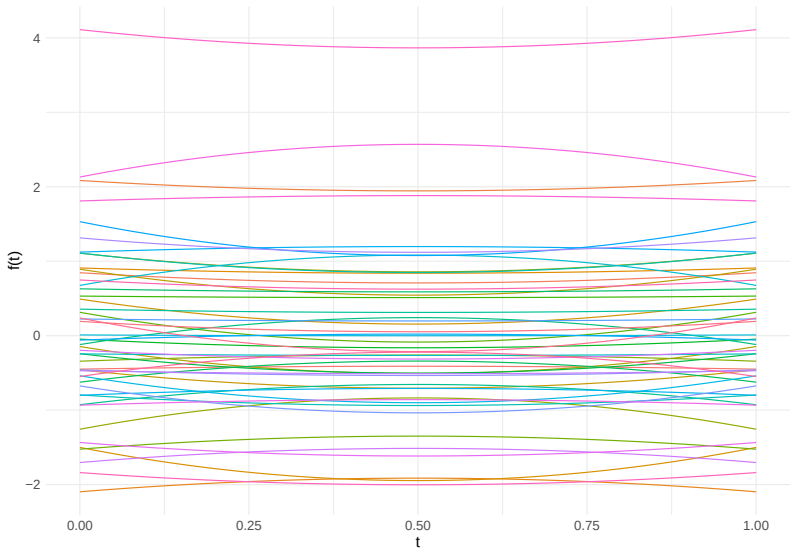


Figure 3.1: Example of $\mathcal{X}(t)$ generation

The scalar covariates are drawn from a multivariate ('3-variate') normal distribution of mean 0 and covariance matrix $cov_{ij} = \{\gamma \text{ if } i \neq j | 1 \text{ if } i = j\}$ for a certain $\gamma = \{0, 0.5\}$. We have selected the vector $\beta_0 = (-1, 0.7, 2)$.

The functional direction θ_0 has been selected as in Novo et al. (2021b) to ensure identifiability [4]. The B-Spline basis order is $d_n = 6$ (attending to order 3 and 3 interior knots). This way the selected coefficients were $\alpha_0 = (0, 1.741539, 0, 1.741539, -1.741539, -1.741539)$ and the true function was constructed as $\theta_0(t) = \sum_{q=1}^{d_n} \alpha_{0q} e_q(t)$ (where $e_q(t)$ are the B-spline basis functions).

The functional space we will be working is $\mathcal{L}^2([0, 1])$ with inner product $\langle f, g \rangle = \int_0^1 fg$. In addition, $r(\langle f, g \rangle) = \langle f, g \rangle^3$.

The sample error has been generated following a normal distribution $\varepsilon \sim N(0, c\mathbf{S}(\mathcal{R}))$ being $c = \{0.01, 0.05\}$ the signal to noise ratio and $\mathbf{S}(\mathcal{R})$ the sample standard deviation of the regression $\mathcal{R} = X_1 + X_2 + X_3 + \langle \theta_0, \mathcal{X} \rangle^3$.

Finally, the Missing at Random responses were introduced attending to the generalisation of the mechanism used in Ling et al. (2019) to 3 linear covariates, and similar to the one used in Febrero-Bande et al. (2019). It is based in the the following probabilistic value:

$$P(\delta = 0 | X_1 = x_1, X_2 = x_2, X_3 = x_3, \mathcal{X} = \chi) = p(\mathbf{x}, \chi) = \text{expit}\Big(\frac{2\alpha}{\pi}(x_1 + x_2 + x_3 + \int_0^1 \chi(t)^2 dt)\Big),$$

being $\text{expit}(u) = \frac{e^u}{1+e^u}$ and α the missing rate parameter. The observed rate is defined as $\bar{\delta} = 1 - \frac{1}{n}\sum_{i=1}^n \delta_i$. Note that this expression is independent of the response variable given all the predictors, being a suitable MAR responses mechanism. Please note how this is a probability given $\delta = 0$, so the higher the α the more missing responses are introduced.

After the sample generation, a train-test split has being performed being the size of the test split 25 in all cases. The number of simulations for each group has been set to $M = 100$ for all the 16 groups drawn from the set $n \times \alpha \times \gamma \times c = \{50, 200\} \times \{0.01, 2\} \times \{0, 0.5\} \times \{0.01, 0.05\}$. The Kernel and the KNN versions of the model estimation have been computed in every iteration.

Some performance metrics will be calculated in order to evaluate different aspects of the model. This metrics will be given for the Kernel and the KNN variations in order to compare both estimation techniques.

- $\beta_{0_{SE}} = \sum(\hat{\beta} - \beta)^2$. The Squared Error between the model estimated $\hat{\beta}$ and the true β.
- $\text{Imputation}_{MSE} = \frac{\sum_d (\mathcal{Y} - y)^2}{|D|}$. Given that $D = \{\text{elements in sample with } \delta = 0\}$, the Mean Squared Error (MSE) between the MAR imputed values $\mathcal{Y}_d$ and the true values obtained from the data generation process y_d.

[4] In essence, we are calibrating as in Novo et al. (2019) the vector $(0, 1, 0, 1, -1, -1)$ as seed coefficients for the algorithm in section 5 and $t_0 = 0.5$.

- MAR_Model$_{MSEP} = \frac{1}{25}\sum_{i=n}^{n+25}(\hat{y} - y)^2$. The Mean Squared Error in Prediction between the predicted (by the model excluding the MAR observations) response values $\hat{y}$ in test and the true test responses y .
- Fitted_Model$_{MSEP} = \frac{1}{25}\sum_{i=n}^{n+25}(\hat{y}^* - y^*)^2$. The Mean Squared Error in Prediction between the predicted (by the model including the MAR imputations) response values $\hat{y}^*$ in test and the true test responses y^*.

Finally, the mean of all metrics for all 100 simulations for each combination of parameters is calculated and will be used to discuss the results in the following section.

3.2. Results

The simulation study already presented was performed in R (version 4.4.0). All the model fitting was done using package 'fsemipar' (version 1.1.1) (Novo and Aneiros, 2024). The table containing all the aggregated mean metrics above discussed can be consulted on Tables 1 and 2 on Appendix. Some visualisations will be included in order to show the mentioned results.

We anticipated several hypotheses for this simulation. Primarily, we expected that increasing the sample size n would improve estimation values, which was confirmed as shown in Figure 3.2. The figure illustrates that, across all metrics and both estimation techniques (Kernel and KNN), larger samples ($n = 200$) consistently led to better performance. Notably, the key metrics, both MSEPs, improved, as did the estimation of the true betas and the MSE of the imputed observations. This suggests that even with more imputations in larger samples, the overall model estimation benefits from the increased sample size.

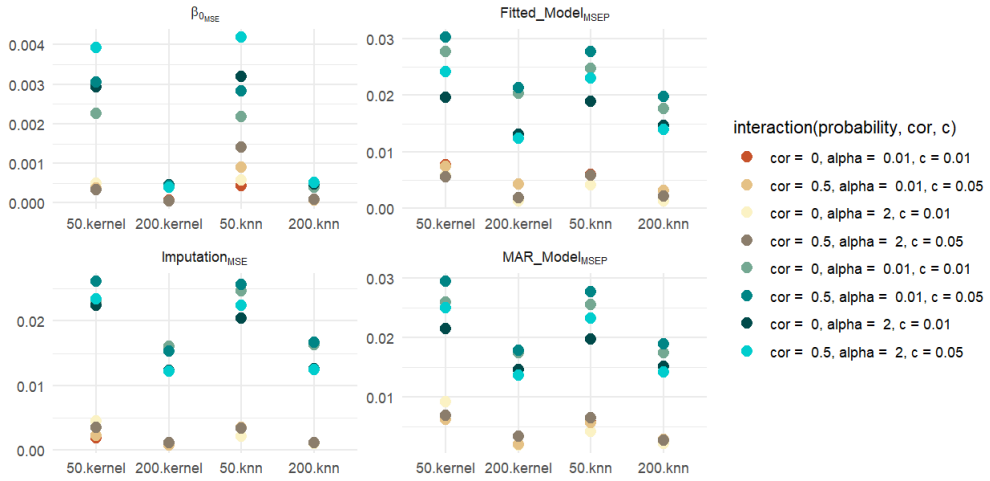


Figure 3.2: Metrics comparison by size and method

Additionally, we anticipated that the KNN estimation would yield superior results in

	size	α	γ	cor	Fitted_kernel$_{MSEP}$	Fitted_knn$_{MSEP}$	rate
1	50	0.01	0	0.01	0.007731	0.005915	-0.234873
2	50	0.01	0	0.05	0.027781	0.024805	-0.107131
3	50	0.01	0.5	0.01	0.005778	0.004186	-0.275548
4	50	0.01	0.5	0.05	0.019617	0.018914	-0.035825
5	50	2	0	0.01	0.007436	0.005836	-0.215146
6	50	2	0	0.05	0.030266	0.027727	-0.083912
7	50	2	0.5	0.01	0.005497	0.005790	0.053399
8	50	2	0.5	0.05	0.024211	0.023086	-0.046447
9	200	0.01	0	0.01	0.004228	0.003090	-0.269177
10	200	0.01	0	0.05	0.020374	0.017686	-0.131948
11	200	0.01	0.5	0.01	0.001243	0.001319	0.061540
12	200	0.01	0.5	0.05	0.013037	0.014617	0.121217
13	200	2	0	0.01	0.004297	0.003083	-0.282501
14	200	2	0	0.05	0.021300	0.019756	-0.072488
15	200	2	0.5	0.01	0.001863	0.002066	0.109056
16	200	2	0.5	0.05	0.012312	0.013924	0.130911

Table 3.1

MSEP comparision between Kernel and KNN estimation

at least some cases due to its local properties, which could justify its inclusion in the model. As shown in Table 3.1, KNN estimation generally demonstrates better performance[5]. The primary observation is that KNN estimation is particularly advantageous for smaller sample sizes, where its local properties offer significant benefits. In contrast, for larger sample sizes, the differences between KNN and Kernel estimations become less pronounced[6].

We thought that the proportion of missing responses will impact the estimation accuracy. Specifically, as the number of missing responses increases, the quality of the estimation should deteriorate. In Table 3.2, we present the rates of change[7] for both Kernel and KNN estimations across all simulation cases. Remember that when $\alpha = 2$ the missing rate is higher. It is evident that the MSEP for the imputed model generally improves, and in instances where it does not, the decline is less than 5%, making it relatively insignificant. In several cases, the improvement is quite substantial, with rates being near to 10%, 20% or even 50%. The favourable cases $(2, 4, 6, 7)$ have bigger magnitudes in the rates as the non favourable $(1, 8)$ and the doubtful $(3, 5)$.

The covariance between the linear predictors may also impact the estimation. In Figure 3.3, it is evident that $\gamma = 0.5$, makes the estimation of beta values worse. This makes sense, given the fact that the linear part of the data gets more complex. On the other hand, we find very surprising that the test MSEP got better with some correlation between the covariates than with uncorrelated ones ($\gamma = 0$) as well as the Imputation$_{MSE}$.

In a similar way, the signal-to-noise ratio c also appears to influence the estimation. As shown in Figure 3.4, an increase in the signal-to-noise ratio generally leads to worse

[5]The rate changes were calculated as (knn − kernel)/kernel, so negative values indicate better results for the KNN estimation.

[6]Actually, Kernel and KNN estimation are equivalent asymptotically as in Novo et al. (2021a)

[7]The rate changes were calculated as (alpha 2 − alpha 0.01)/alpha 0.01, where positive values indicate a worse estimation as α increases.

				Kernel			KNN		
	size	γ	c	$Fitted\_MSEP_{\alpha=0.01}$	$Fitted\_MSEP_{\alpha=2}$	rate	$Fitted\_MSEP_{\alpha=0.01}$	$Fitted\_MSEP_{\alpha=2}$	rate
1	50	0	0.01	0.007731	0.007436	-0.038081	0.005915	0.005836	-0.013281
2	50	0	0.05	0.027781	0.030266	0.089456	0.024805	0.027727	0.117788
3	50	0.5	0.01	0.005778	0.005497	-0.048633	0.004186	0.005790	0.383348
4	50	0.5	0.05	0.019617	0.024211	0.234151	0.018914	0.023086	0.220555
5	200	0	0.01	0.004228	0.004297	0.016283	0.003090	0.003083	-0.002246
6	200	0	0.05	0.020374	0.021300	0.045462	0.017686	0.019756	0.117074
7	200	0.5	0.01	0.001243	0.001863	0.498823	0.001319	0.002066	0.565913
8	200	0.5	0.05	0.013037	0.012312	-0.055608	0.014617	0.013924	-0.047442

Table 3.2

MAR ratios comparison in imputed model MSEP

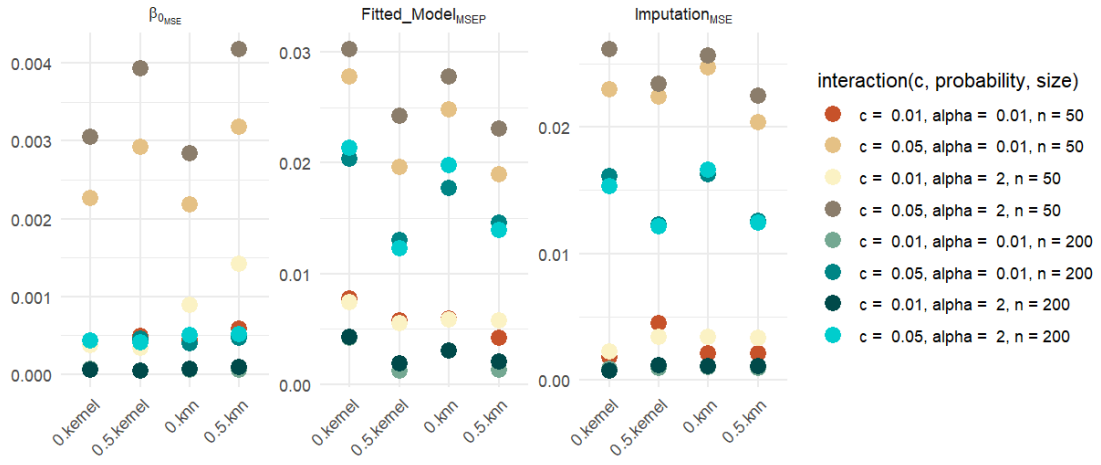


Figure 3.3: Metrics comparison by linear predictors covariance

results for most metrics.

Finally, we expected that the MSEP in the model with the imputations to be lower than the MSEP in the model that eliminates the missing responses. You can see the results summarised in Table 3.3. Some surprising results arise from this table [8]:

- In the Kernel estimation, we observed how the imputation was worse in the cases of $\gamma = 0$. Surprisingly, when covariates had $\gamma = 0.5$ the imputation worked better than eliminating the MAR responses. A possible explanation to this event may be the complexity of the data. When the linear covariates are uncorrelated, the complexity that the model has to encapsulate is less, so the information we loose when eliminating observations does not compensates the error introduced when estimating the MAR responses and imputing them. When the complexity of the data is higher, this trade-off does compensate.

- For KNN estimation, results were more heterogeneous, noticing how when the MAR model worked better the difference is very slight in magnitude (less than

[8]The rate changes were calculated as (mar − fitted)/fitted, where negative values indicate a worse estimation of the Fitted (Imputed) Model

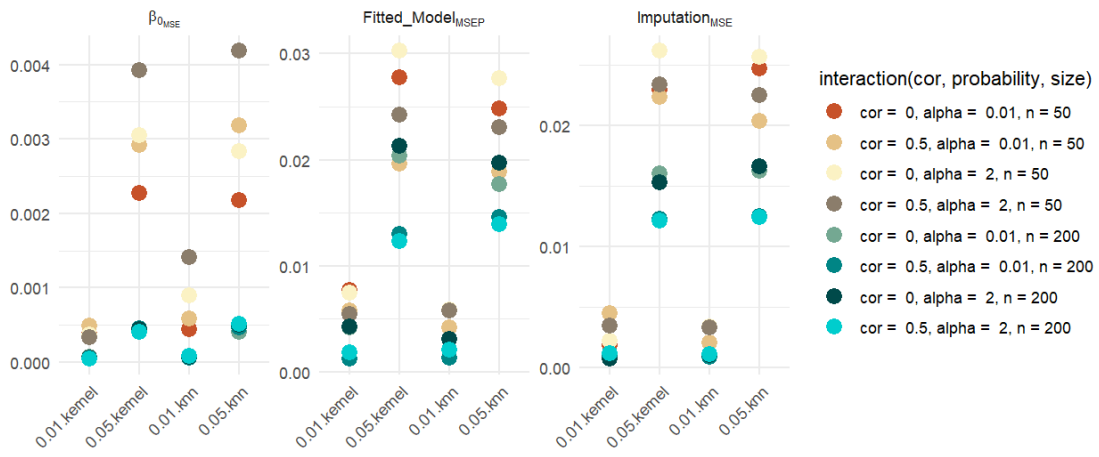


Figure 3.4: Metrics comparison by signal to noise ratio

10%) whether when the Imputation worked better the difference is noticeably bigger. This indicates us that the KNN estimation handles better the complex models than the Kernel estimation, but it is not clear which cases benefit from the imputation in KNN estimation.

					Kernel			KNN		
	size	α	γ	c	*MAR_MSEP*	*Fitted_MSEP*	rate	*MAR_MSEP*	*Fitted_MSEP*	rate
1	50	0.01	0	0.01	0.006934	0.007731	-0.103099	0.005852	0.005915	-0.010650
2	50	0.01	0	0.05	0.025907	0.027781	-0.067442	0.025598	0.024805	0.031961
3	50	0.01	0.5	0.01	0.009198	0.005778	0.591988	0.004177	0.004186	-0.001989
4	50	0.01	0.5	0.05	0.021534	0.019617	0.097692	0.019752	0.018914	0.044270
5	50	2	0	0.01	0.006169	0.007436	-0.170395	0.005651	0.005836	-0.031772
6	50	2	0	0.05	0.029468	0.030266	-0.026361	0.027709	0.027727	-0.000616
7	50	2	0.5	0.01	0.006933	0.005497	0.261330	0.006488	0.005790	0.120495
8	50	2	0.5	0.05	0.024977	0.024211	0.031649	0.023292	0.023086	0.008899
9	200	0.01	0	0.01	0.003384	0.004228	-0.199697	0.002763	0.003090	-0.105791
10	200	0.01	0	0.05	0.017431	0.020374	-0.144454	0.017459	0.017686	-0.012816
11	200	0.01	0.5	0.01	0.001866	0.001243	0.500851	0.002081	0.001319	0.577185
12	200	0.01	0.5	0.05	0.014547	0.013037	0.115870	0.015091	0.014617	0.032425
13	200	2	0	0.01	0.002022	0.004297	-0.529548	0.002864	0.003083	-0.071036
14	200	2	0	0.05	0.017833	0.021300	-0.162784	0.018972	0.019756	-0.039701
15	200	2	0.5	0.01	0.003317	0.001863	0.780372	0.002673	0.002066	0.293595
16	200	2	0.5	0.05	0.013588	0.012312	0.103632	0.014158	0.013924	0.016846

Table 3.3

MSEP ratios comparison

4. REAL CASE STUDY

In statistical modelling and data analysis, the application of methodologies to real-world datasets serves as a crucial validation of theoretical approaches. The primary objective of a real case study is to assess the performance and applicability of proposed models under real-world conditions, where data may exhibit complexities and challenges not fully captured in synthetic simulations like missing data, noise, and other anomalies that are often present in actual datasets. In our study, the focus is on evaluating the effectiveness of SFPLSIM-MAR model to four different estimation scenarios depending on the estimation technique (Kernel and KNN) and percentage of Missing at Random (MAR) responses.

Specifically, this study aims to:

- Assess the impact of missing data on model accuracy by comparing results with and without imputation.
- Compare the performance of Kernel and KNN SFPLSIM estimation in the context of MAR responses.
- Analyse how the MAR mechanism affects the estimation process and determine which method is more robust in the presence of missing data for a particular data set.

4.1. The Tecator Dataset

The Tecator dataset is a well-known benchmark in the field of FDA, particularly for studies involving regression models. It has being used in Ferraty and Vieu (2006), Shang (2014) or Ling et al. (2019), among others. The dataset originates from the Tecator Infratec Food and Feed Analyzer, which measures the absorbance spectra of meat samples across a range of wavelengths. The primary objective is to predict the fat content of these samples based on their absorbance profiles. This dataset can be found in several resources like in the 'fsemipar' package (Novo and Aneiros, 2024), but our version is the one found in the package 'fda.usc' (Febrero-Bande and Oviedo de la Fuente, 2012).

The dataset consists of 215 samples, each providing a spectrum of absorbance values at 100 different wavelengths, ranging from 850nm to 1050nm discretised into 100 points. In addition to the absorbance data, the dataset includes three more characteristics of the meat: Fat, Protein, and Water content. It is usual to take Fat as response variable and Protein and Water as explanatory variables. Some of the curves can be observed in Figure 4.1

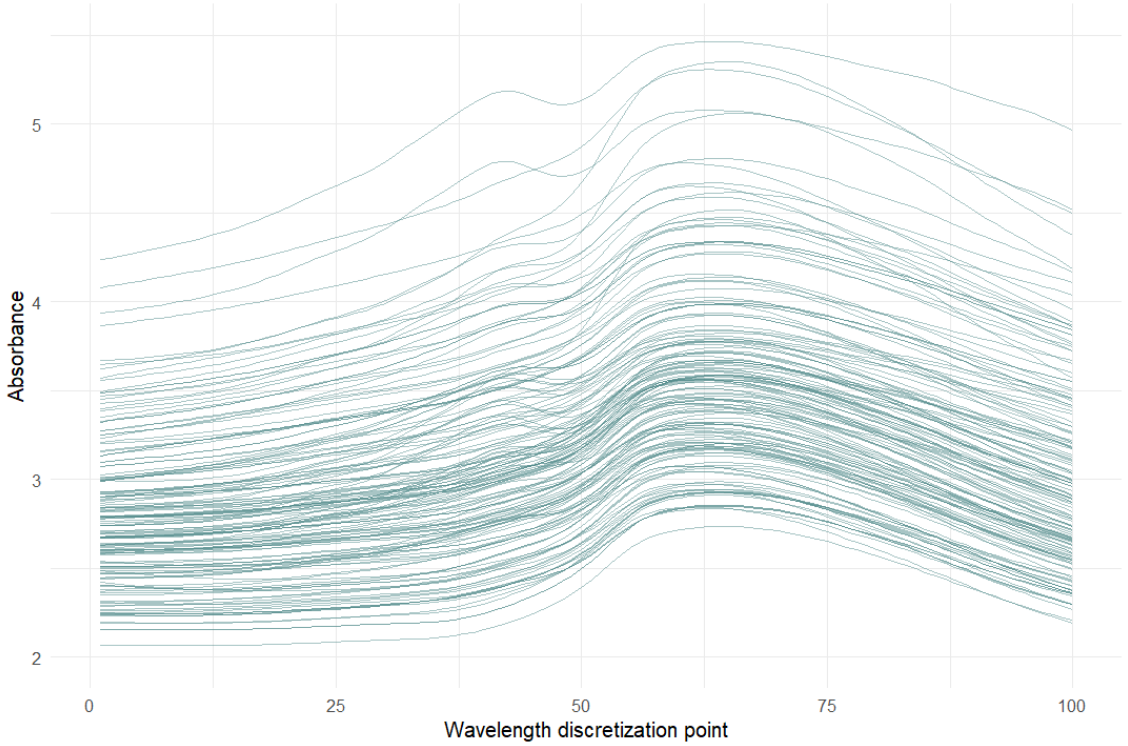


Figure 4.1: Tecator absorbance curves

Note how this dataset has no missing responses, so we will introduce some of them artificially.

4.2. Methodology

Since the Tecator dataset has no missing responses, we had to replicate a MAR mechanism. We followed the one described in Ling et al. (2019) that introduces missingness based on a logistic regression model where the probability of missing data in the response variable is modelled as a function of observed variables and additional covariates, similar as our simulation study. Specifically, the log odds of missingness are expressed as:

$$\log\left(\frac{P(\delta = 1 \mid X_1, X_2, \chi}{1 - P(\delta = 1 \mid X_1, X_2, \chi}\right) = \alpha\left(X_1 + X_2 + \int_0^{100} \chi(t)\, dt\right),$$

where δ is an indicator variable for missingness same as the simulation study, X_1 and X_2 are the observed predictors Protein and Water, and $\int_0^{100} \chi(t)\, dt$ represents the integral of the spectral data over the range 1 to 100. The parameter α controls the effect of the predictors and the integral on the probability of missing data. If α decreases, the number of missing responses increases, opposite to what happened in the simulation study in Chapter 3.

The probability of missing data in the Fat column is then calculated using the logistic function:

$$P(\delta = 1 \mid X_1, X_2, \chi) = \frac{1}{1 + \exp\left(\alpha\left(X_1 + X_2 + \int_0^{100} \chi(t)\, dt\right)\right)}$$

Prior to our MAR responses introduction, the Tecator dataset will be separated into train-test partitions. The canonical way of performing this split is making the train sample $\{X_{1i}, X_{2i}, \chi_i, Y_i\}_{i=1}^{160}$ and test $\{X_{1i}, X_{2i}, \chi_i, Y_i\}_{i=160}^{215}$.

We will evaluate the performance of the SFPLSIM-MAR attending to:

- Fitted_Model$_{MSEP}$ $= \frac{1}{55}\sum_{i=160}^{215}(\hat{y}_i^* - y_i)^2$. The Mean Squared Error in Prediction between the predicted (by the model including the imputations) response values $\hat{y}^*$ in test and the true test responses y.
- MAR_Model$_{MSEP}$ $= \frac{1}{55}\sum_{i=160}^{215}(\tilde{y}_i^* - y_i)^2$. The Mean Squared Error in Prediction between the predicted (by the model excluding the missing responses observations) response values $\hat{y}^*$ in test and the true test responses y.
- Full_Model$_{MSEP}$ $= \frac{1}{55}\sum_{i=160}^{215}(y_i^* - y_i)^2$. The Mean Squared Error in Prediction between the predicted (by the model including all observations prior to the MAR introduction) response values y^* in test and the true test responses y. This can be seen as a benchmark to evaluate the 'base' error in each method.
- Imputation$_{MSE}$ $= \frac{\sum_d(\mathcal{Y}-y)^2}{|D|}$. Given that D = {elements in sample with $\delta = 0$}, the Mean Squared Error (MSE) between the MAR imputed values $\mathcal{Y}_d$ and the true values obtained from the dataset y_d.

At the outset of this study, we made a full analysis with the full training sample and got some unsatisfactory results. Both estimation techniques results were unexpected, having better MSEPs with more missing samples, and even making the Fitted_Model$_{MSEP}$ being lower than the Full_Model$_{MSEP}$, which is unlikely. This results can be seen in Table 3 on Appendix. After a process of intensive testing, we discovered that while KNN estimation offers advantages in small samples and can provide more accurate estimates under certain conditions, it is particularly susceptible to the presence of large outliers, as was the case in this dataset. This vulnerability significantly impacted the performance of the KNN method, leading to the observed anomalies. It is for this reason that we also performed a functional outlier detection analysis in the training sample[9]. We ended up excluding 4 observations as shown in Figure 4.2.

[9]Note how the outlier detection was done by the built in trimming function with modal depth measure in package 'fda.usc' in R. Bootstrap samples set to 200, smoothing is 0.05, quantile set to 0.5 and trimming parameter is 0.01.

The model fitting was done (for all Kernel and KNN models) with a B-Spline basis of order 4, 20 interior knots for the curves and 4 interior knots for θ_0 estimation.

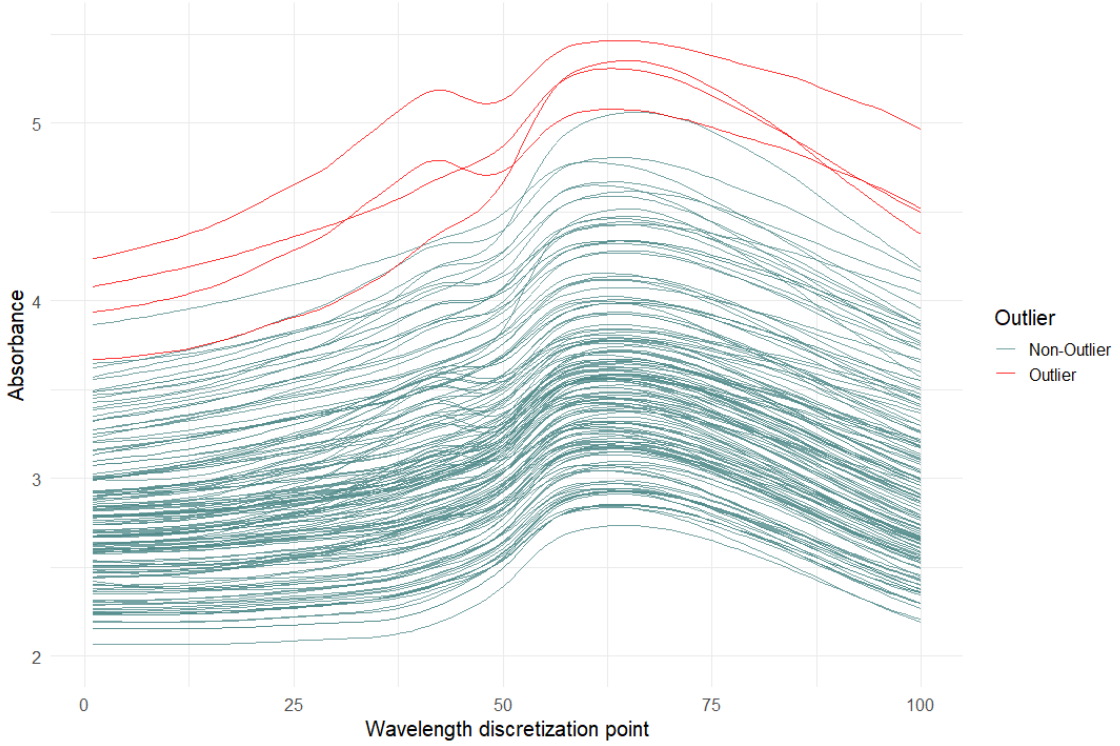


Figure 4.2: Outlier detection in Tecator dataset with

4.3. Results

The results of our analysis provide several insights into the performance of the Kernel and KNN methods under the conditions imposed by the Tecator dataset. The results can be observed in Table 4.1.

		Kernel			**KNN**		
% missing	α	Imputation$_{MSE}$	MAR$_{MSEP}$	IMP$_{MSEP}$	Imputation$_{MSE}$	MAR$_{MSEP}$	IMP$_{MSEP}$
20	0.0035	2.437615	2.400773	2.105095	0.9002344	1.433772	1.374259
44	0.0008	3.179664	2.845492	2.820679	1.635691	1.313862	1.180884
FULL MSEP		**1.3884**			**1.212723**		

Table 4.1

Comparison of Kernel and KNN estimation for different cases in Tecator without Outliers

As anticipated, the increase in the percentage of missing responses is directly correlated with a rise in the imputation error, denoted as Imputation$_{MSE}$, for both Kernel and KNN methods. A higher proportion of missing data inherently complicates the imputation process, resulting in larger discrepancies between the imputed and true values. In

more detail, the KNN method consistently outperforms the Kernel method in terms of imputation accuracy across the evaluated scenarios. This finding is consistent with the results observed in our earlier simulation studies.

However, the behaviour of the two methods diverges when evaluating the MSEPs metrics. For the Kernel method, both MAR_{MSEP} and IMP_{MSEP} exhibit an increase when the missing rate parameter is set to $\alpha = 0.0008$. This suggests that as the missingness rate increases, the Kernel method struggles to maintain prediction accuracy, possibly due to its reliance on smoothness and global features, which become less effective as the amount of missing data grows.

In contrast, the KNN method presents a different pattern. When the missing parameter is $\alpha = 0.0008$, the method unexpectedly outperforms its performance at a lower missing rate (higher α). Remarkably, in this scenario, the IMP_{MSEP} is even lower than that of the Full_Model_{MSEP}, which is derived from the complete dataset without missingness. This counterintuitive result suggests that under certain conditions, KNN's focus on local data points can lead to more accurate predictions. Also, even though this phenomenon highlights the robustness of KNN estimation in challenging datasets, it is important to approach it with caution. We think this observed metrics, far for being a trend, only respond to the specific characteristics of this dataset.

5. CONCLUSIONS AND FUTURE WORK

This thesis set out with a clear set of objectives centred on the explanation of the SFPLSIM and the development of novel methodologies for handling missing data in the context of SFPLSIM-MAR. Through the rigorous generalisation of theoretical concepts and practical simulations, all the primary objectives have been successfully achieved, addressing a gap in the Functional Data Analysis literature. The work represents a valuable advance in FDA and Functional Regression, exploring the effectiveness of Kernel and KNN estimation (and imputation) techniques in the context of Missing at Random (MAR) mechanisms, and their impact on the SFPLSIM.

The simulation and real-case study conducted in Chapters 3 and 4 provided several insights. Notably, the comparison between Kernel and KNN estimation methods demonstrated the significant influence that the choice of imputation and estimation techniques can have on the overall model performance. KNN estimation, with its local estimation capabilities, consistently showed superior imputation accuracy, particularly in scenarios with complex data structures or higher levels of missingness. Also, we noted how the rates of missing responses significantly affected the estimation procedures, generally lowering the performance of the methods, being this effects mitigated by KNN estimation in comparison to Kernel estimation, which was less robust in this context.

There are several avenues for future research that could build upon the findings presented here. First, we observed how, even though KNN imputation worked better almost in every scenario, not always the best test results were with KNN but with Kernel estimation, so we find an interesting modification to the presented procedure to impute the MAR responses with KNN imputation and then complete the Imputed Model fitting with Kernel estimation. This may result to overall better performance under certain conditions and may be explored.

Also, the study primarily focused on specific MAR mechanism, but future work could extend this analysis by testing other ones, being different probabilistic values to the presented ones or even Missing Completely at Random (MCAR) mechanisms. This would provide a more comprehensive understanding of how different types of missing data affect the performance of SFPLSIM.

The simulation study presented in Chapter 3 was very extensive. Still, since the model is very complex, we think there are some modifications or further actions to be tested.

On one hand, we encountered a computation limitation throughout the development of the study. Since these techniques are very computationally expensive, we couldn't reasonably try all the cases and situations we wanted, even though we presented in this thesis the most important ones. In future works, we could focus the simulations only in some of the cases now confirmed as favourable for this model, but increasing both the

train and test sample sizes. This is because we suspect some of the not-so-clear results in the simulation study may have been caused by the limited size of the sample. Also, we think that for this new scenarios, develop a benchmark like the Full_Model$_{MSEP}$ may be very useful to finish the discussion of the favourable and not favourable contexts for Kernel and KNN estimation in SFPLSIM-MAR.

On the other hand, some of the unusual findings in the simulation study may be related to the semiparametric aspect of the model. Future research could explore increasing the influence of the semiparametric component both in magnitude and in shape (different functions and projections), as this might help explain some of the unexpected results and provide further insights into the behaviour of the method.

In addition, in order to improve the efficiency of the model, a future study may be in the θ_0 estimation. Currently, we are estimating the parameter twice, one in the imputing process and then again in the imputed model fitting. If the first estimation is good enough (which could be evaluated under certain conditions) we may be able to use it in the second part of the process and save a lot of computational effort.

Last, but not least, the natural way to proceed now is to implement a more refined and user friendly version of the presented Algorithm 3 in R in order to enable the management of MAR responses scenarios in SFPLSIM to the general statistician. This could be done as a part of an update in a specialised package like 'fsemipar' and could end up being a future publication.

Overall, this thesis has successfully met all its objectives, delivering a comprehensive and valuable contribution to the field of Functional Regression. The innovative approaches and insightful findings presented here represent significant progress, offering practical solutions to real world challenges.

BIBLIOGRAPHY

Ait-Saïdi, A., Ferraty, F., Kassa, R., & Vieu, P. (2008). Cross-validated estimations in the single-functional index model. *Statistics*, *42*(6), 475–494.

Aneiros, G., Cao, R., Fraiman, R., Genest, C., & Vieu, P. (2019). Recent advances in functional data analysis and high-dimensional statistics [Special Issue on Functional Data Analysis and Related Topics]. *Journal of Multivariate Analysis*, *170*, 3–9.

Aneiros-Pérez, G., & Vieu, P. (2006). Semi-functional partial linear regression. *Statistics & Probability Letters*, *76*(11), 1102–1110.

Cardot, H., Ferraty, F., & Sarda, P. (1999). Functional linear model. *Statistics & Probability Letters*, *45*(1), 11–22.

Carroll, R. J., Fan, J., Gijbels, I., & Wand, M. P. (1997). Generalized partially linear single-index models. *Journal of the American Statistical Association*, *92*(438), 477–489.

Febrero-Bande, M., Galeano, P., & González-Manteiga, W. (2019). Estimation, imputation and prediction for the functional linear model with scalar response with responses missing at random [High-dimensional and functional data analysis]. *Computational Statistics Data Analysis*, *131*, 91–103.

Febrero-Bande, M., & Oviedo de la Fuente, M. (2012). Statistical computing in functional data analysis: The R package fda.usc. *Journal of Statistical Software*, *51*(4), 1–28.

Ferraty, F., & Vieu, P. (2006). *Nonparametric Functional Data Analysis: Theory and Practice*. Springer New York.

Ferraty, F., Peuch, A., & Vieu, P. (2003). Modèle à indice fonctionnel simple. *Comptes Rendus Mathematique*, *336*(12), 1025–1028.

Hardle, W., Hall, P., & Ichimura, H. (1993). Optimal smoothing in single-index models. *The Annals of Statistics*, *21*(1), 157–178.

Ling, N., Kan, R., Vieu, P., & Meng, S. (2019). Semi-functional partially linear regression model with responses missing at random. *Metrika*, *82*(1), 39–70.

Novo, S., & Aneiros, G. (2024). fsemipar: an R package for SoF semiparametric regression.

Novo, S., Aneiros, G., & Vieu, P. (2019). Automatic and location-adaptive estimation in functional single-index regression. *Journal of Nonparametric Statistics*, *31*(2), 364–392.

Novo, S., Aneiros, G., & Vieu, P. (2021a). A KNN procedure in semiparametric functional data analysis. *Statistics Probability Letters*, *171*, 109028.

Novo, S., Aneiros, G., & Vieu, P. (2021b). Sparse semiparametric regression when predictors are mixture of functional and high-dimensional variables. *Test*, *30*, 481–504.

Ramsay, J., & Silverman, B. (2005). *Functional Data Analysis*. Springer New York.

Shang, H. L. (2014). Bayesian bandwidth estimation for a semi-functional partial linear regression model with unknown error density. *Computational Statistics*, *29*, 829–848.

Speckman, P. (1988). Kernel smoothing in partial linear models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *50*(3), 413–436.

Vieu, P. (2018). On dimension reduction models for functional data [The role of Statistics in the era of big data]. *Statistics and Probability Letters*, *136*, 134–138.

Wang, G., Feng, X.-N., & Chen, M. (2016). Functional partial linear single-index model. *Scandinavian Journal of Statistics*, *43*(1), 261–274.

APPENDIX 1

n	α	$\bar{\delta}$	γ	c	β_kernel_{SE}	Imputation_kernel$_{MSE}$	MAR_Model_kernel$_{MSEP}$	Fitted_Model_kernel$_{MSEP}$
50	0.01	0.1954	0	0.01	0.000378	0.001828	0.006934	0.007731
50	0.01	0.1728	0	0.05	0.002269	0.022965	0.025907	0.027781
50	0.01	0.2418	0.5	0.01	0.000488	0.004503	0.009198	0.005778
50	0.01	0.2518	0.5	0.05	0.002925	0.022370	0.021534	0.019617
50	2	0.2554	0	0.01	0.000370	0.002213	0.006169	0.007436
50	2	0.2606	0	0.05	0.003048	0.026173	0.029468	0.030266
50	2	0.3044	0.5	0.01	0.000338	0.003424	0.006933	0.005497
50	2	0.3172	0.5	0.05	0.003928	0.023424	0.024977	0.024211
200	0.01	0.175	0	0.01	0.000072	0.000890	0.003384	0.004228
200	0.01	0.176	0	0.05	0.000439	0.016071	0.017431	0.020374
200	0.01	0.2417	0.5	0.01	0.000046	0.000938	0.001866	0.001243
200	0.01	0.2449	0.5	0.05	0.000452	0.012301	0.014547	0.013037
200	2	0.2559	0	0.01	0.000058	0.000719	0.002022	0.004297
200	2	0.2556	0	0.05	0.000438	0.015330	0.017833	0.021300
200	2	0.3065	0.5	0.01	0.000048	0.001148	0.003317	0.001863
200	2	0.3178	0.5	0.05	0.000405	0.012154	0.013588	0.012312

Table 1

Aggregated Metrics - Kernel Estimation

n	α	$\bar{\delta}$	γ	c	β_knn_{SE}	Imputation_knn$_{MSE}$	MAR_Model_knn$_{MSEP}$	Fitted_Model_knn$_{MSEP}$
50	0.01	0.1954	0	0.01	0.000442	0.002095	0.005852	0.005915
50	0.01	0.1728	0	0.05	0.002179	0.024714	0.025598	0.024805
50	0.01	0.2418	0.5	0.01	0.000584	0.002076	0.004177	0.004186
50	0.01	0.2518	0.5	0.05	0.003185	0.020387	0.019752	0.018914
50	2	0.2554	0	0.01	0.000896	0.003406	0.005651	0.005836
50	2	0.2606	0	0.05	0.002836	0.025650	0.027709	0.027727
50	2	0.3044	0.5	0.01	0.001417	0.003293	0.006488	0.005790
50	2	0.3172	0.5	0.05	0.004184	0.022486	0.023292	0.023086
200	0.01	0.175	0	0.01	0.000062	0.000972	0.002763	0.003090
200	0.01	0.176	0	0.05	0.000402	0.016264	0.017459	0.017686
200	0.01	0.2417	0.5	0.01	0.000063	0.000908	0.002081	0.001319
200	0.01	0.2449	0.5	0.05	0.000470	0.012545	0.015091	0.014617
200	2	0.2559	0	0.01	0.000068	0.001060	0.002864	0.003083
200	2	0.2556	0	0.05	0.000509	0.016630	0.018972	0.019756
200	2	0.3065	0.5	0.01	0.000090	0.001115	0.002673	0.002066
200	2	0.3178	0.5	0.05	0.000510	0.012407	0.014158	0.013924

Table 2

Aggregated Metrics - KNN estimation

APPENDIX 2

		Kernel			KNN		
% missing	α	Imputation$_{MSE}$	MAR$_{MSEP}$	IMP$_{MSEP}$	Imputation$_{MSE}$	MAR$_{MSEP}$	IMP$_{MSEP}$
20	0.003	2.515129	1.770236	1.575505	2.460702	2.286908	2.423205
44	0.0008	2.408305	1.354958	1.436377	1.7069311	1.373102	1.140815
FULL MSEP		**1.3884**			**1.212723**		

Table 3

Comparison of Kernel and KNN estimation for different cases in Tecator with Outliers