Forecasting Age-Specific Mortality Rates

Using Functional Data Analysis.

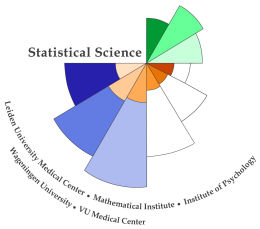
Pantelis Matsakidis (s2105330)

Thesis advisor: Prof. Dr. Thomas Nagler

MASTER THESIS

Defended on Month Day, Year

Specialization: Data Science



STATISTICAL SCIENCE
FOR THE LIFE AND BEHAVIOURAL SCIENCES

Abstract

Functional Data Analysis (FDA) has received increased attention within the demographic forecasting framework since it was first introduced by Hyndman and Ullah (2007) as a robust approach on forecasting mortality and fertility rates. In this master thesis we apply this methodology on the Dutch mortality rates for the years 1950-2018. We make a model comparison by fitting Functional Principal Components Analysis (FPCA), weighted FPCA and Functional Partial Least Squares models on the age-specific mortality rates from 1950 to 1999, for the male and female populations separately.

We obtain the results of the evaluation by calculating the Root Mean Squared Error (RMSE) between the forecasted and true mortality rates from 2000 to 2018. They show that for the male population, the weighted FPCA outperformed the other models, while for the female population the FPCA and weighted FPCA models had a very similar performance.

Contents

1 Introduction

In demographic studies, one is interested in revealing notable features that may characterize a population or a certain group of people, as well as identify important patterns that might be taking place, while trying to provide possible explanations for them. This master thesis is itself a demographic study for the population of the Netherlands, which tries to approach the previously mentioned points, among some other technical issues. The population characteristic which is of our main concern throughout this project, is the mortality rate. We wish to investigate how mortality rates differ between groups of people, as well as how they change over the course of time.

Particularly, we study the mortality rates of people from different age groups and of different sex, from 1950 to 2018 in the Netherlands. The age group distinction is very important in this case since mortality rates greatly vary with respect to age. The same is true for the populations of the two sexes. Male populations tend to have a bit different mortality rate patterns than female populations, and this is definitely something we want to consider in our analysis. Furthermore, we are interested in how the mortality rates of all the mentioned groups change through time, which basically forms the core of our project. Additionally, we apply some modeling techniques within the framework of Functional Data Analysis (FDA), which allows us to better understand our time dependent data, and also provide forecasts for the mortality rates of the different groups.

FDA basically treats observation sets as separate functions which are defined over a specific interval. Viewing our data as separate functions can in general be an important advantage of using FDA, since we can easier and more efficiently identify important information that might be present. One can also draw additional information from the derivatives of the original functions in order to investigate patterns that are not immediately observable. These functions can then be used in a similar manner as in classic statistical analysis, with only a few differences on the mathematical background, since functions have different properties than vectors. With the use of FDA, we can think of the mortality rates of every age group for one year as a function with the mortality rates on the y-axis and the age groups on the x-axis. Consequently, our data will consist of 68 mortality rate functions which are considered as our observations, one for each year from 1950 to 2018. In this context we can use these observations which are ordered by year, for exploratory analysis as well as for model fitting and forecasting for the male and female population separately. This concept is known as *functional time series* and it has various applications in demographic forecasting.

We apply different modeling methods on the mortality curves from 1950 to 1999, in order to produce forecasts for the mortality rates from 2000 to 2018 for males and females separately, which can be used for testing their forecast accuracy. This way, one can evaluate different methods on the same test data, provide a thorough comparison in relation to the overall performance of each method, and attempt to provide an explanation for any significant differences that might occur. Functional time series have been used in the past for demographic applications, notably from Hyndman and Ullah (2007), Hyndman and Shang (2009) and Hyndman and Booth (2008) which include all of the methods that we applied in our project. All of these research papers contribute to the conclusion that FDA can be a robust and flexible way of approaching demographic forecasting problems.

Section 2 contains a brief overview of the methods that we used. It includes the theoretical background of the techniques that are used for modeling the data, as well as the theory of some

methods that are used for data preparation in Section 4. Section 3, is a more detailed review of the literature on functional time series and their application in demography, with a strong focus on the methods that were developed in Hyndman and Ullah (2007) and Hyndman and Booth (2008). In Section 4 follows a description of the data along with exploratory analysis which can help visualize and identify trends that might be important. Also, it includes a description of the smoothing process required before the modeling step. Section 5 describes the models that were fitted on the mortality rate curves for the two genders, while providing goodness-of-fit measures and residual plots in order to visualize how these models differ in the way they fit the data. Section 6 contains an evaluation of their forecast accuracy on the test data, by computing the Root Mean Squared Error (RMSE) between the forecasted and true mortality rates by year and by age group. Finally, Section 7 provides an overall conclusion between the differences in performance of the models as well as possible ways on how this thesis could be extended along with its limitations.

2 Background

2.1 Functional data

In the Functional Data Analysis (FDA) framework, observation sets are considered as functions and each of them lives in an interval $[\alpha, \beta]$. In that case, one could have a sequence of functions $F = (X_1(t), X_2(t), ..., X_N(t))$, where $t \in [\alpha, \beta]$ and N is the number of functions. It is therefore helpful to think of these curves as observations, where a single realization of the i-th curve at point t is defined as $x_i(t)$. Furthermore, we can define the basic population statistics such as the mean function $\mu(t) = E[X(t)]$, $t \in [\alpha, \beta]$ and the covariance function $C(s,t) = E[(X(t) - \mu(t))(X(s) - \mu(s))]$. These population statistics have the form of functions as well, since they are computed over the population of curves, for a given value of $t \in [\alpha, \beta]$.

Additionally, if we consider a vector of functions F as we defined it above, one can derive the sample mean function, which denotes the average of the realizations for the functions contained in vector F, at a specific point t, as follows:

$$\overline{X}(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t)$$

Similarly, the sample variance function can be written as:

$$Var(X(t)) = \frac{1}{N-1} \sum_{i=1}^{N} (X_i(t) - \overline{X}(t))^2$$

Finally, the sample covariance function which can be used as a measure of association between two different points, t_1 and t_2, is defined as:

$$Cov(t_1, t_2) = \frac{1}{N-1} \sum_{i=1}^{N} (X_i(t_1) - \overline{X}(t_1))(X_i(t_2) - \overline{X}(t_2))$$

In general, the goals of FDA remain the same as in classical statistics, and in this project we will use past age-specific mortality curves $X_i(t)$, where $i = 1, 2, ..., n$ denotes the year and t the age, in order to generate future forecasts for males $X^M_{n+h}(t)$ and females $X^F_{n+h}(t)$.

In reality, the realizations of a function $X_i(t)$ $(i = 1, 2, ..., n)$, are just discrete values $x_i(t_1), ..., x_i(t_N)$ observed for different values of $t \in [t_1, t_N]$, therefore making the processes of *interpolation* and *smoothing*, one of our first concerns in FDA. The former method applies when the realizations are assumed to be observed without error, whereas the latter is used in the case where the realizations contain some form of observational error that needs to be handled. The goal of these methods is to enable us to evaluate the function and its derivatives at any arbitrary value of t, always in relation to the available data.

2.2 Basis function decomposition

From the several smoothing methods that are available, in this project we will focus on the basis function methods. Using a basis function method, one can approximate a function $X_i(t)$ by using a linear combination of K basis functions ϕ_k:

$$X_i(t) \approx \sum_{k=1}^{K} c_k \phi_k(t)$$

The number of basis functions K controls the level of smoothing that will take place. For example, a small number of K may not be able to capture most of the information present in the data that we want to smooth. A very large number of K will overfit the data, in the sense that the basis function representation will not be a smooth curve anymore. Figure 1 depicts an example, where two *B-spline* curves were fitted on the log-mortality ratios of men in the Netherlands, for the year 2000. As one may notice, the available data points represent a specific age group, therefore smoothing is an important technique in this case, if we want to be able to evaluate our functions at any arbitrary time age t. Both curves were fitted using a basis function method, with the only difference being the number of basis functions K used in each case. Although the *B-spline* method is explained in detail in the next section, the impact of the chosen number of K in each curve is obvious in Figure 1. On one hand, the curve which corresponds to $K = 4$ basis functions does not really capture the information present in the data, especially for the age groups 0-20 the trend is not being well represented. On the other hand, the black curve with $K = 50$ basis functions, looks more like a series of different lines that connect the data points between the age groups. Consequently, the choice of K should be made in such a way, that the fitted curves captures the general trend along with the most important sources of information from the original data points, without being too specific.

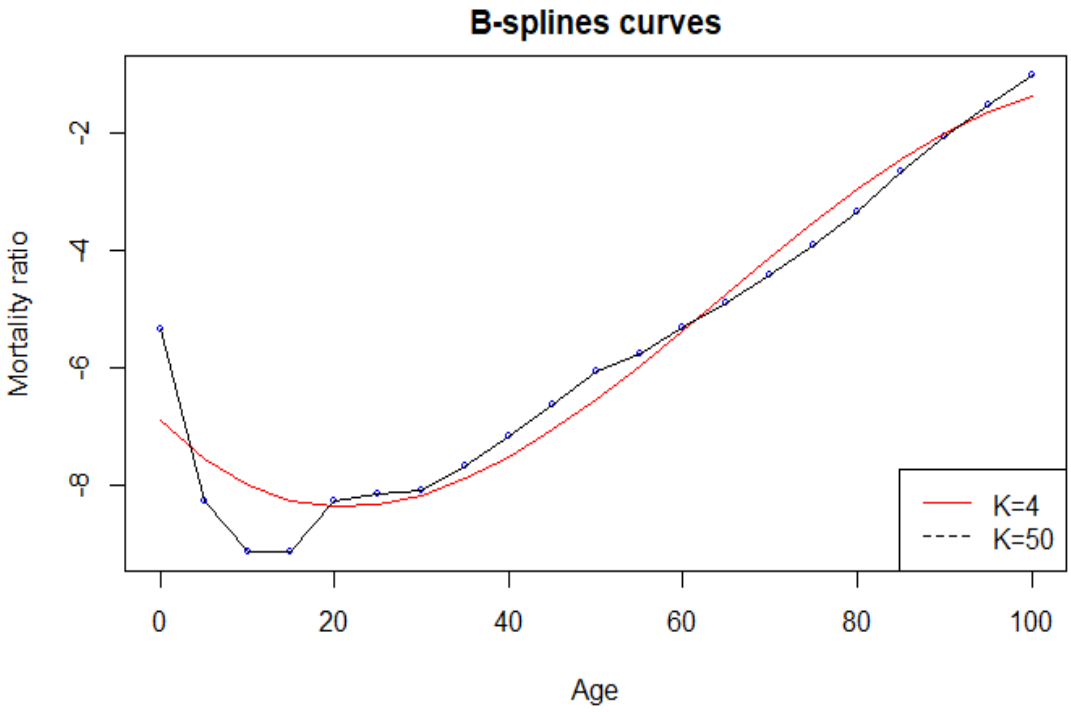


Figure 1: *Example of the impact that number of basis functions K has on the final fitted curve. Two B-splines curves are fitted, with $K = 4$ (red line) and $K = 50$ (black line).*

2.3 B-splines smoothing

The next step in our analysis, is to represent the mortality curves as smooth functions before moving into the modeling part. This is particularly important in our case, since the mortality curves are defined over the different age groups, which are 20 in number, meaning that there are 20 data points at which a mortality curve can be evaluated on. As it was also mentioned in section 2.2, the smoothing method that we applied is called *B-splines* and it will be explained in detail on the following paragraph (Aguilera and Aguilera-Morillo (2013)). Additionally, the fitting criterion along with the estimation of coefficients are going to be discussed, and finally we are going to visualize the functional data as smooth mortality curves for the male and female population.

The term *B-splines* is a short for basis splines, where a linear combination of them can be used to approximate a function, in our case the age-specific mortality rate function. Basically, the approximation involves a linear combination of different polynomials of the same degree, which are tied together at specific points called *knots*. A polynomial of degree n is a mathematical expression of the following form:

$$P(x) = c_n x^n + c_{n-1} x^{n-1} + \cdots + c_2 x^2 + c_1 x + c_0$$

The above expression can equivalently be written as $\sum_{i=0}^{n} c_k x^k$, where $c_0, \ldots, c_n$ are constants and x is a variable. Therefore, a spline function of *order* m, consists of $m-1$ degree polynomials ($order = degree + 1$), tied together at N_{knots} knots (k_j) which are non-decreasingly ordered ($k_j \leq$

k_{j+1}). Furthermore, the number of basis functions in a B-splines smoothing approach, along with the order and the number of knots need to satisfy the following relationship:

$$K = m + N_{knots} - 2 \tag{1}$$

That means, the number of basis functions K is equal to the sum of the order m and the number of interior knots (total number of knots N_{knots}, except the two exterior knots).

If we define the polynomials that constitute a B-spline function as segments of the same degree, then segments are constrained to be smooth at the joints. In other words, a B-spline is continuous at the knots we have defined. It is also important to note that B-spline functions are uniquely defined by their knots, meaning that two B-spline functions are identical, only if they are of the same order and are defined over the exact same knots.

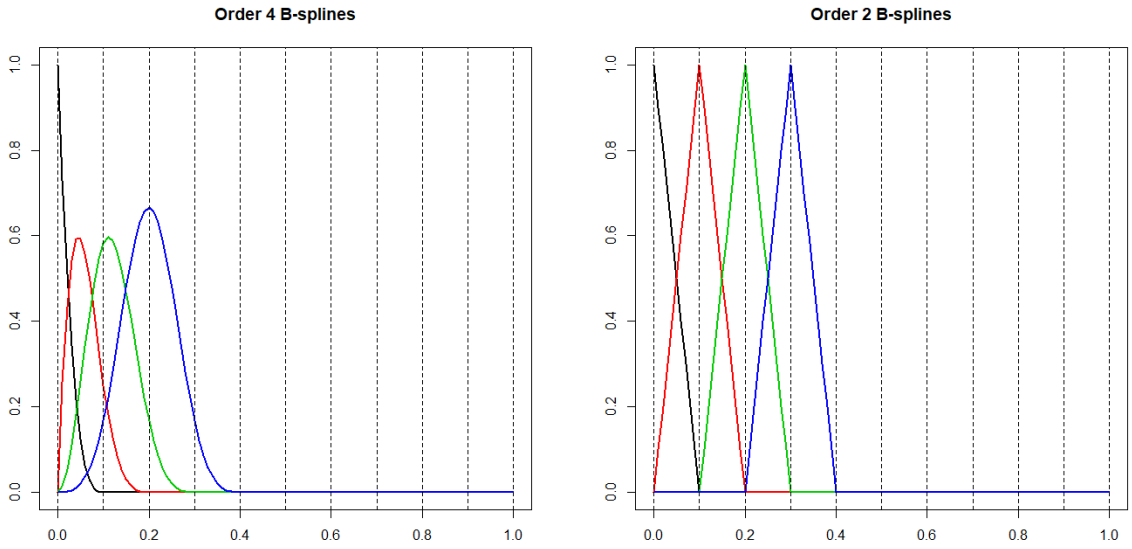


Figure 2: *B-splines example that shows the impact of the order. The first 4 basis functions are plotted in both cases. In one case the order is 4 (left plot), and in the other case the order is 2 (right plot).*

Additionally, and interesting property of the order m of B-splines, is that a spline of order m can be non-zero, at up to m segments. This property can be better understood and visualized by observing Figure 2, which presents two cases of B-splines, one of order 2 and one of order 4. In each example, the number of knots used is 11, but the order is different. Therefore, one can understand that the number of basis functions will also differ between the two cases. In the case where the order is 4, the number of basis functions will be 13, whereas in the other case where the order is 2, we will have 11 basis functions according to relationship (1). So, for the sake of simplicity and for better visualization, in this example only the first 4 basis functions are plotted in each case. Moreover, one can see that in the left plot where the order is 4, the first three basis

functions are non-zero at one, two and three segments respectively, followed by the fourth basis function which is non-zero at four segments. This is an even pattern, which also applies for the last three basis functions. In comparison, if we observe the right plot we will see that the first basis function is non-zero at one segment, but all the other basis functions are limited into being non-zero at exactly two segments.

An additional property of B-splines which is very useful in the functional data framework, is that the first $m-2$ derivatives are continuous, given that the knots are distinct (they do not coincide). Generally, in functional data analysis there is a wide interest in derivative functions because they can provide us with additional information and patterns that we wouldn't be able to recognise otherwise. The derivative of a function basically represents how sensitive to change is a function's output value, with respect to the change of its input variable. In other words, a derivative denotes rate of change, and the process of calculating the derivative of a function is called *differentiation*. Therefore, the derivative of a function f (if it exists) is written as f', the derivative of f' is written as f'' (second derivative of f), and similarly the derivative of the $(n-1)$-th derivative of f, is called the n-th derivative. Derivatives are particularly important in the smoothing process, because when estimating a smooth function from functional data, we want the the patterns of the derivative functions to be preserved as well. Generally, because the order itself does not have a big impact on the fitted curve, a very common choice of B-splines is of order 4, which is called *cubic* B-splines and it ensures that the first and second derivatives are continuous. Knots can be then experimented with, in order to achieve a suitable fit.

Generally, when experimenting with choosing the order or the knots of a B-splines basis, one should keep the tradeoff between bias and variance in mind. Assuming that we want to estimate a function $f(t)$, the bias is computed as $\text{Bias}[\hat{f}(t)] = f(t) - E[\hat{f}(t)]$ and the variance as $\text{Var}[\hat{f}(t)] = E[(\hat{f}(t) - E[\hat{f}(t)])^2]$, where $\hat{f}(t)$ is the estimated function. By increasing the order or the knots, the number of basis functions increases as well. Therefore, adding too many basis functions means small bias but large sampling variance. Similarly, too few basis functions can lead to small sampling variance but large bias.

2.4 Penalized B-splines using a roughness penalty

An additional approach on smoothing splines that slightly differs from the regular method that was discussed above, is the penalized B-splines, or roughness penalty approach. This method involves the introduction of a penalty term, in order to counter some of the limitations that the regular smoothing approach suffers from. One of those limitations is the sensitivity to the number of basis functions used for smoothing. It is difficult to define an optimal value for the order of the B-splines and number of knots to be used, because there is always the tradeoff between bias and variance.

More specifically, if one is interested in estimating a smooth function f from data points $y_j = f(t_j) + \epsilon_j$, a way of checking the fit of f on the data is by computing the sum of squared error as follows:

$$SSE = \sum_j [y_j - f(t_j)]^2$$

The difference in the penalized approach is that the above equation will include an additional penalty term λ as can be seen from the following:

$$SSE_\lambda = \sum_j [y_j - f(t_j)]^2 + \lambda * PEN_2(f)$$

where $PEN_2(f) = \int f''(s)^2 ds$ is the integrated squared second derivative of f, which is a measure of roughness that quantifies the degree to which a function deviates from being a straight line. Therefore, higher values of PEN_2 indicate that the function highly deviates from a straight line.

The smoothing parameter λ has control over the amount of penalization applied during the smoothing process. The higher the value of λ ($\lambda \to \infty$), the more the fitted curve f will approach a straight line. On the contrary, the lower the value of λ ($\lambda \to 0$), the more f will approach a curve that satisfies $f(t_j) = y_j$ for every j, which is equivalent to no penalization applied.

Consequently, the roughness penalty approach has the advantage of giving us the freedom of choosing a larger number of basis functions for smoothing, that would otherwise result in overfitting if regularization was not to be applied. Therefore, one does not need to specify an optimal number of knots to be used or basis functions in general, but has to find an optimal value for the smoothing parameter λ instead, in order to apply the right amount of regularization. The process of choosing the optimal λ value involves experimentation with different values and visualizing the results, as well as computing some measure of error between the fitted curve and the original data and the the *generalized cross validation* (GCV) statistic which is further explained on section 4.3.

2.5 Functional principal components analysis

A basis function decomposition method that is very popular in the functional data framework, is Functional Principal Components Analysis (FPCA). FPCA can identify the most important sources of variation in the functional data at hand, and can be used for exploring trends as well as for modeling. By reducing the data into a few components that are responsible for most of the variation, it gains the advantage of interpretability which can be very useful in understanding important properties of the data. More specifically, a function $X_i(t)$ can be decomposed into a mean function and a linear combination of K orthogonal functional principal components:

$$X(t) = \mu(t) + \sum_{k=1}^{\infty} \beta_k \phi_k(t) \tag{2}$$

where $\mu(t)$ is the mean function, β_k are uncorrelated functional principal component scores, and $\phi_k(t)$ are orthogonal functional principal components derived from the FPCA method.

The main concept of the FPCA method, is to identify the most important sources of variation by maximizing the variance, and it can be described in more detail through a series of steps Ramsay and Silverman (2001). The first step, is to calculate $\phi_1(t)$, which is the first *principal component*, for which the values $\beta_{i1} = \int \phi_1(t) X_i(t) dt$ have the maximum possible mean square $N^{-1} \sum_i^N \beta_{i1}^2$. The constraint $\int \phi_1(t)^2 dt = 1$ is necessary to prevent the mean square from becoming arbitrarily large. Note that in the above notation, $X_i(t)$ denotes the i-th functional variable, and the values

β_{i1} denote the *principal component scores*. Maximization of the mean square is equivalent to the maximization of the variance, if prior to performing FPCA, the mean of each variable is subtracted. In fact, given the presence of $\mu(t)$ in equation (2), the functional variables have to be centered before FPCA, otherwise the results might be misleading.

During the m-th step (where $m \leq n$), the weight function $\phi_m(t)$ is computed, such that the principal component scores β_{im} have maximum mean square, subject to the constraint $\int \phi_m(t)^2 dt = 1$. But this time, there are $m-1$ additional constraints that need to be satisfied. These are $\int \phi_k(t)\phi_m(t)dt = 0$, for $k < m$. This means, that the weight functions need to be orthogonal to those derived from the previous steps, so that they are representing a new feature. These steps can be extended up to a limit of the number of functional observations n. However, the amount of variation explained in each subsequent step will decrease, which may result in losing interest as the number of steps approaches n.

Therefore, by using a finite number of K basis functions in order to estimate $X_i(t)$, we get the estimated principal components and principal component scores, $\overline{\phi}_k(t)$ and $\overline{\beta}_k$ respectively along with the sample mean function $\overline{X}(t)$, leading us to the following equation:

$$X_i(t) \approx \overline{X}(t) + \sum_{k=1}^{K} \overline{\beta}_k \overline{\phi}_k(t) \tag{3}$$

An additional approach on FPCA which was compared to the orignal FPCA in Hyndman and Shang (2009) and is also applied in this project, is called weighted FPCA and it comes with the use of geometrically decaying weights on the computation of the principal components. This alteration from the previous method has the effect of putting more weight on selected curves in order give them more influence on the final estimation. For example, if we assume that the curves are ordered by year, then by using this method we can put more weight on the most recent data so that they have a stronger impact on the final result.

More precisely, by using the same steps that were described earlier, applying weighted FPCA leads us to a very similar expression to (3), with some slight differences that come from the use of weights:

$$X_i(t) \approx \overline{X}^*(t) + \sum_{k=1}^{K} \overline{\beta}_k \overline{\phi}_k^*(t)$$

In this case, $\overline{X}^*(t) = \sum_{i=1}^{N} w_i X_i(t)$ denotes the weighted average function and $\{w_i = \lambda(1-\lambda)^{N-i}, i = 1, \ldots, N\}$ is a set of geometrically decaying weights, where $0 < \lambda < 1$ is the weight parameter that can be estimated from the data (Hyndman and Shang (2009)). Additionally, $\overline{\phi}_k^*(t)$ denotes the weighted functional principal components. They are calculated given that their corresponding principal component scores, maximize the quantity $\sum_i \beta_{ik}^2 w_i^2$.

2.6 Univariate time series models

Before moving into the functional time series framework, we first introduce the basics of time series modelling, since they will be later needed to provide the h-step-ahead forecasts of the component

scores. A univariate time series model can be defined when we are dealing with a single time series variable y_t. That means, we observe values of y_t over a specified time interval, where $t = 1, 2, ..., m$ denotes the time parameter.

In this project, a univariate Auto-Regressive Integrated Moving Average (ARIMA) model will be used for forecasting the component scores before we are able to forecast the future mortality curves. An ARIMA model, consists of two parts which are going to be described separately, the Auto-regressive model (AR(p)) and the Moving Average model (MA(q)).

In an AR(p) model, one can make forecasts of the variable y_t by using a linear combination of past values of itself. Therefore the past values of y_t act as predictors. The parameter p denotes the order of the model, which defines the number of past values of y_t used in the model. So, an AR(p) model can be defined as follows:

$$y_t = c + \upsilon_1 y_{t-1} + \upsilon_2 y_{t-2} + \cdots + \upsilon_p y_{t-p} + \varepsilon_t \tag{4}$$

where different values of the parameters $\upsilon_1, \upsilon_2, ..., \upsilon_p$ lead to different time series patterns, and ε_t represents an independent and identically distributed random variable called white noise. The white noise term is therefore a sequence of random numbers that show no autocorrelation, which means that there is no association between ε_t and ε_l, for $l < t$.

The difference of a MA(q) model from an AR(p) model, is that instead of using past values as predictors, it uses past forecast errors. It can be specified as follows, with the parameter q having the same effect as in an AR(p) model:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}, \tag{5}$$

where the parameters $\theta_1, \theta_2, ..., \theta_p$ control the time series patterns and ε_t is white noise.

Before moving into the definition of an ARIMA model, it is important to introduce the meaning of *stationarity* first. A time series is stationary when the behavior of the observations remains the same over the whole period at which they are observed. A time series can be considered stationary when there are no obvious trends or patterns nor seasonality present, and its statistical properties such as the mean and variance remain constant over time. As one can understand, in practice the assumption of stationarity is not always satisfied. Therefore, ARIMA models are used for forecasting time series which are not stationary, but can be made so by using *differencing*.

Differencing is a transformation which is applied to non-stationary time series, in order to make them stationary by stabilizing the mean of the series. More specifically, it involves computing the differences between consecutive observations, which results in reducing the trends to a certain degree, or potentially remove them. Therefore, if differencing is applied, the transformed series can be written as:

$$y'_t = y_t - y_{t-1}$$

The above equation results in a new, transformed time series y'_t, which represents the change between consecutive observations from the initial time series. We may also note, that since the first observation of the differenced time series y'_1 is not possible to calculate, it will consist of one observation less than the original series. It may happen that even the differenced time series do not appear to be stationary, in which case a second round of differencing can be applied to the differenced series y'_t:

$$y''_t = y'_t - y'_{t-1}$$

In this case the second-order differenced series will represent the change in changes of the original series, and it will consist of two observations less than the original series since $y'_t - y'_{t-1} = y_t - 2y_{t-1} + y_{t-2}$. Theoretically, this process can be continued up to a degree d of differencing, giving the following general form:

$$y_t^{(d)} = y_t^{(d-1)} - y_{t-1}^{(d-1)}$$

Therefore, an ARIMA(p, q, d) model can be constructed from combining differencing with the two models (4) and (5), that were described above, where d is the degree of differencing:

$$y_t^{(d)} = c + \upsilon_1 y_{t-1}^{(d)} + \cdots + \upsilon_p y_{t-p}^{(d)} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{6}$$

2.7 Functional time series

Functional time series methods are of our main concern throughout this project, since we are dealing with demographic data which are measured over time. More specifically, in our framework we can define a function $y_t(a_i)$, which represents the age-specific mortality rates for age values a_i at year t. Since our mortality function is observed for discrete age points a_i, it is assumed that an underlying smooth function $f_t(a_i)$ can be extracted from the following equation:

$$y_t(a_i) = f_t(a_i) + \sigma_t(a_i)\epsilon_{t,i} \tag{7}$$

where $\epsilon_{t,i}$, is a random variable that denotes the observational error which follows the standard normal distribution, and, for fixed t and varying i, is independent and identically distributed. Furthermore, $\sigma_t(a_i)$ represents a variance term that lets the error variance change with age a_i and year t.

This approach was initially introduced by Hyndman and Ullah (2007) as a method of deriving smooth functions from the data, with application on Australian mortality and fertility rates. By applying various methods one can forecast the mortality rates curve y_{n+h}, where n is the most recent time point $(t = 1, 2, ..., n)$ and h is the chosen forecast horizon.

More specifically, $f_t(a_i)$ can be decomposed as it is shown on (3) using a basis decomposition approach such as the functional principal components analysis (FPCA) described in the previous section:

$$f_t(a) = \overline{f}(a) + \sum_{k=1}^{K} \beta_{t,k}\hat{\phi}_k(a) + \epsilon_t(a)$$

On the equation above, $\overline{f}(a)$ stands for the estimated mean function $\overline{f}(a) = 1/n \sum_{t=1}^{n} f_t(a)$, $\hat{\phi}_k(a)$ is the k-th estimated eigenfunction that shows the main regions of variation, $\beta_{t,k}$ is the k-th estimated component score for year t and $\epsilon_t(a)$ are the residuals. Also, K is the selected number

of basis functions used for decomposition. Typically, the optimal number of basis functions K, can be found in such a way that the forecast error is as small as possible, but it has been proven that when using the FPCA method, the choice of K does not have a significant impact on the forecast accuracy, and it is therefore preferable that a larger value of K (although not too large) should be chosen rather than a small one which could potentially lead to poor forecast accuracy (Hyndman and Booth (2008)).

The h-step-ahead forecasts of the component scores $\beta_{n+h,k}$ are constructed using a univariate time series model. A univariate time series model is sufficient in our case because the principal component scores $\hat{\beta}_{t,k}$ are uncorrelated for different values of $k = 1, 2, ..., K$. Therefore, we will fit a univariate time series model for each of the principal component scores series $\beta_{t,k}$, that means one model for each value of k. h-step ahead forecasts from ARIMA(p,q,d) models are computed from the following equation, for $k = 1, 2, \ldots, K$:

$$\hat{\beta}_{n+h,k} = c + \upsilon_1 \hat{\beta}_{n+h-1,k} + \upsilon_2 \hat{\beta}_{n+h-2,k} + \cdots + \upsilon_p \hat{\beta}_{n+h-p,k} + \theta_1 \varepsilon_{n+h-1} + \cdots + \theta_q \varepsilon_{n+h-q} + \varepsilon_{n+h},$$

where each subsequent $\hat{\beta}_{t,k}$ with $n < t \leq n + h$ is estimated as a one-step forecast from an ARIMA model using all the previously observed and forecasted values of t as covariates.

Finally, the h-step-ahead forecasts are provided from equation (7).

$$\hat{y}_{n+h|n}(x) = \mu(x) + \sum_{k=1}^{K} \hat{\beta}_{n+h|n,k} \hat{\phi}_k(x) + \epsilon_t(x)$$

From the equations described above, the forecast variance can be estimated in order to create prediction intervals. Note that through the FPCA approach, one approximately gets orthogonal basis functions, which means that the forecast variance can be estimated by the sum of component variances of the previous equation:

$$\mathrm{Var}[y_{n+h}(x)|f(x), B] \approx \hat{\sigma}^2_{\mu}(x) + \sum_{k=1}^{K} u_{n+h,k} \hat{\phi}^2_k(x) + u(x) + \hat{\sigma}^2_{n+h}(x)$$

On the above equation, $\hat{\sigma}^2_{\mu}(x)$ is the variance of the mean function estimated from the smoothing method, $u_{n+h,k} = \mathrm{Var}(\beta_{n+h,k}|\beta_{1,k}, \beta_{2,k}, ..., \beta_{n,k})$ is estimated from the univariate time series model, $u(x)$ is the average of $\hat{\epsilon}^2_t(x)$ for every value of x and $t = 1, 2, ..., n$, and $\hat{\sigma}^2_{n+h}(x)$ is estimated from the smoothing method as well.

3 Functional approaches on demographic forecasting: a literature review

3.1 Robust forecasting approach of mortality and fertility rates

Prior research on the topic of functional data methods, has shown that they can make reliable forecasts on demographic characteristics, hence, this is why they have been used by several countries

for the production of their official statistics (including the Netherlands). A scientific paper introduced by Hyndman and Ullah (2007) was the first demonstration of robust forecasting within the functional data framework and its research aim was to make age-specific forecasts on Australian mortality and fertility rates, for the years 2001-2020.

The methods applied on this paper, include non parametric smoothing (constrained and weighted penalized regression splines) in order to estimate the smooth curves $f_t(x)$, for $x \in (x_1, x_p)$ from the observed functional time series $x_i, y_t(x_i)$ (where $y_t(x_i)$ denotes the mortality/fertility rate for age x in year t). Next, followed the decomposition of these curves using a basis function expansion, where the basis functions were obtained from a functional PCA method.

Note also, that the model built by Hyndman and Ullah was a generalization of the Lee-Carter method (1992), which was one of the most important and distinguished methods of stochastic demographic forecasting. Their main difference is that Lee and Carter used only the first principal component during decomposition, whereas Hyndman and Ullah chose the value of K that minimizes $\sum_{t=N}^{n-h} \sum_{h=1}^{m} ISFE_n(h)$ (that was $K = 3$), where $ISFE_n(h)$ is the Integrated Squared Forecast Error. In other words, they used the optimal number of basis functions in order to achieve the highest possible forecast accuracy.

Finally, the forecasts for fertility for 2001-2020, showed a decrease in fertility rates for ages that range from 17 to 30 years old, and a slight increase in fertility for ages over 30. This result is a reflection on how society changes over the years, with more women shifting to later ages for giving birth, as they are more focused on chasing careers at a younger age. Other remarkable changes in fertility rates could be noticed from the older curves that were used for forecasting, like for example the post-war increase in fertility for all ages at around 1945.

For the mortality rates, there certainly has been improvement over the Lee-Carter method. Experimental analysis show that this method has a good performance on mortality over the period 1900-1990, because age patterns have been stable, so it could produce reliable forecasts given the continuity of similar trends in the future Lee (2000).

3.2 Stochastic population forecasts for mortality fertility and migration

The scientific paper introduced one year later by Hyndman and Booth (2008) is a very good example of the flexibility provided by functional data methods, since they are used to make forecasts for three different demographic characteristics, mortality, fertility and net migration.

In this paper, instead of using log transformations on $y_t(x)$, a Box-Cox transformation was used in order to introduce greater variability for higher rates and lower variability for lower rates. Where the Box-Cox transformation on a functional response $y_t^*(x)$ is defined as follows:

$$y_t(x) = \begin{cases} \frac{1}{\lambda}([y_t^*(x)]^\lambda - 1) & \text{if } 0 < \lambda \leq 1 \\ \ln(y_t^*(x)) & \text{if } \lambda = 0. \end{cases}$$

This has the advantage of letting the transformation differ with respect to λ, with $\lambda = 1$ giving no transformation and $\lambda = 0$ giving the common log transformation. Also, this time the data were restricted from 1950 and on, to avoid adding further complexity to the model, because one

would have to follow a more robust and complex estimation of the model parameters in order to account for the impact of war and epidemics on the mortality rates.

Another alteration from the previous research paper, was the modification of the method used for calculating the forecast variance. The variance in Hyndman and Booth (2008) was adjusted in such a way, that the one-step-ahead forecast variance, matches the in-sample one-step-ahead forecast variance, which turned out to lead into smaller variances for forecasts made for mortality and fertility rates, at many different ages. Equation (7) of the forecast variance can equivalently be written as: $\text{Var}[y_{n+h}(x)|f(x), B] = V_h(x) + \hat{\sigma}^2_{n+h}(x)$, with the term $\hat{\sigma}^2_{n+h}(x)$ representing the observational error. The in-sample empirical forecast variance can be calculated to check the validity of (7), and is given by:

$$W_h(x) = \frac{1}{n-h-m+1} \sum_{t=m}^{n-h} [f_{t+h}(x) - \hat{f}_{t,h}]^2,$$

where m is the minimum number of observations used in a model. Therefore, the adjustment made on the forecast variance was the following:

$$\text{Var}[y_{n+h}(x)|f(x), B] = V_h(x)W_1(x)/V_1(x) + \hat{\sigma}^2_{n+h}(x).$$

The result of this adjustment in mortality rates, was that the variance was reduced for almost all ages except very old ones, but for fertility rates, the variance for ages 25-35 was increased and for ages 35 and above it was decreased.

As in the previous paper, a functional PCA method was used to obtain a basis function expansion, only that this time it was known that the value of K (which denotes the number of basis functions) does not need to be optimized (Hyndman and Ullah (2007) selected K to minimize the mean integrated squared forecast error), given that it is large enough. In other words, adding more basis functions than needed, does not decrease forecast accuracy, though if K is too large the computational time will increase.

Finally, forecasts for the next years were generated using the cohort-component method, which involves the simulation of deaths, births and migrants from the functional models, in order to simulate a future population. The method described above by Hyndman and Booth (2008), accounts for many different sources of variation, therefore it forms a thorough representation of the uncertainty that lies behind the data. However, there are still other dependencies that were not addressed, like the ones between the mortality or migration of the two sexes. The whole process was even further extended in 2009 on a research by Hyndman and Shang (2009), where they proposed the introduction of geometrically decreasing weights during the functional PCA, which lead to more recent data having a larger impact on the results than the older data. Furthermore, it was shown that the use of weighted approaches, achieves a better forecasting accuracy than the unweighted ones.

4 Data Preparation

4.1 Description

In this project, we use population and mortality data by age and sex for the Netherlands. The data were collected from the online database of the Central Agency for Statistics in the Netherlands (Centraal Bureau voor de Statistiek, or CBS). More specifically, two datasets were combined in order to derive the mortality rates by age and sex. These include the population data by age and sex, and the total deaths by age and sex. According to CBS, the population data become available at the 1st of January, whereas the total number of deaths, are being published on December of the same year. The two databases consist of population and mortality data for males and females of all ages across a 68 year period, starting from 1950 up to 2018. Its also important to note that data are recorded for 21, 5-year age groups $(0,\, 1-5, 5-10, 10-15, \ldots, 90-95, 95+)$ with the only exceptions being the newborns and the age group of 95+ years old due to their distinctiveness on displaying higher mortality rates. Therefore, the mortality rates are calculated by dividing the number of deaths for a particular age group within a calendar year, by the total number of individuals in this age group for the same year. The calculation is done separately for males and females as can be seen below:

$$mr_{S,t}(\alpha_i) = \frac{D_{S,t}(\alpha_i)}{P_{S,t}(\alpha_i)}$$

where $D_{S,t}(\alpha_i)$ denotes the number of deaths for the sex S population, at calendar year t, for the age group α_i, and $P_{S,t}(\alpha_i)$ denotes the corresponding population of the target group.

4.2 Exploratory analysis of long-term trends

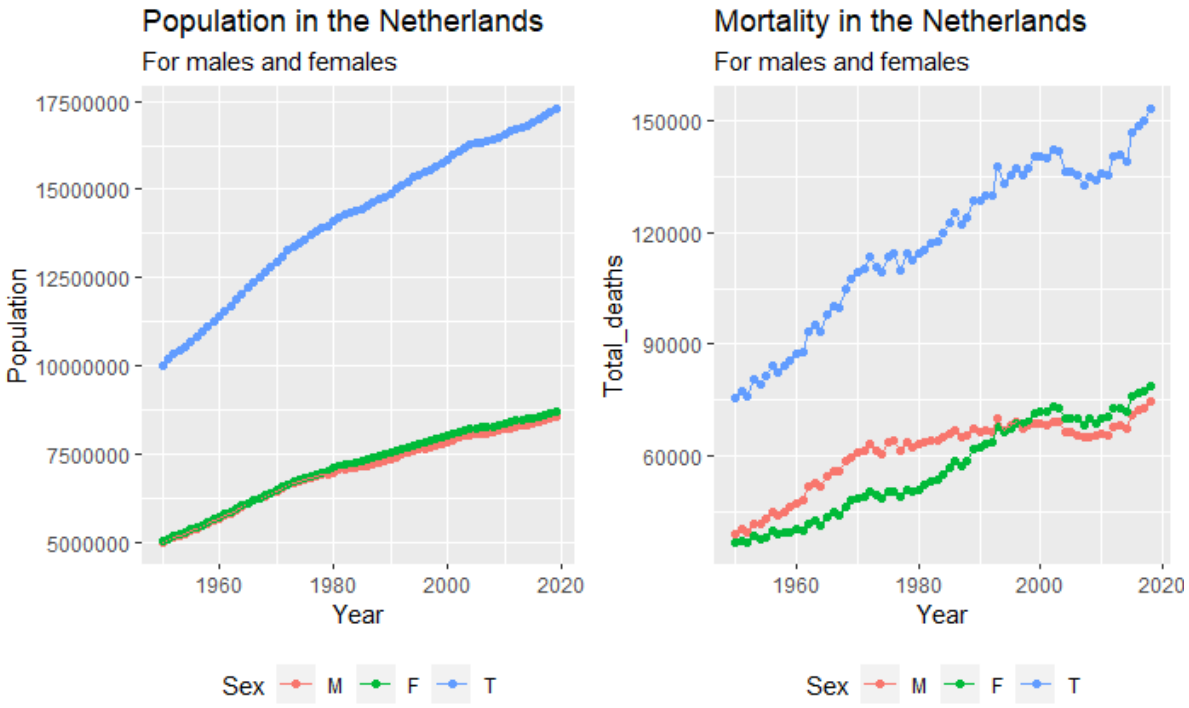


Figure 3: *Population in the Netherlands for males, females and total (left). Mortality in the Netherlands for males, females and total (right). (F: female, M: male, T: total)*

We start with an exploratory analysis of the mortality and population data that were described previously, in order to identify possible trends in the data and to better understand the problem before delving deeper into it. We mainly focus on the differences between the population and mortality of the two genders during the period 1950-2018, and try to explain them through visualization, as well as some patterns on the population change of two particular age groups.

From the two plots in Figure 3, one can get a general taste of the population and mortality data for the Netherlands, as well as how they deviate over the years. On the left plot, the fast paced population growth is clearly visible, justifying the fact that the Netherlands is one of the most densely populated countries in Europe, with an increase of 5.664.253 people over a 57 year period. One may also notice, that from around 1980, the population of females in the country had already started to surpass the population of males, while preserving a relatively steady difference through the years. Meanwhile, we are also noticing an increase in the total number of deaths for both sexes, with the female mortality curve being generally steeper. The total mortality for males was constantly higher than for females, until around the year 2000, where the two curves intersect, following a period where the majority of deaths were related to women.

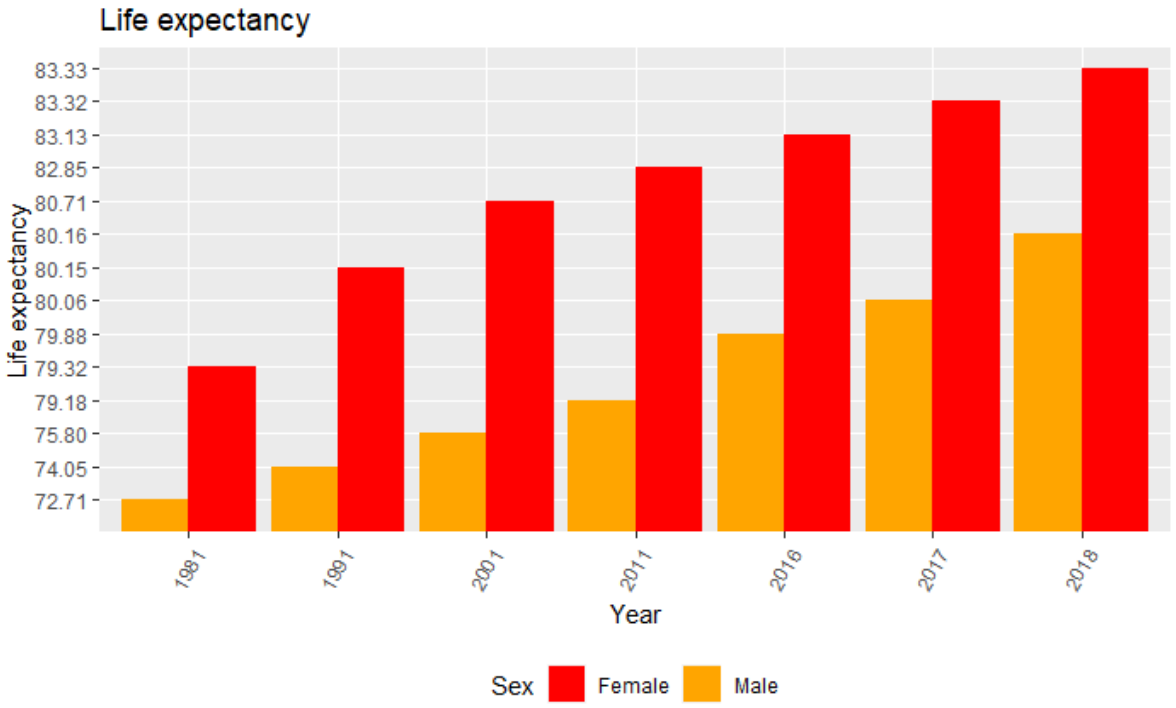


Figure 4: *Life expectancy for males and females*

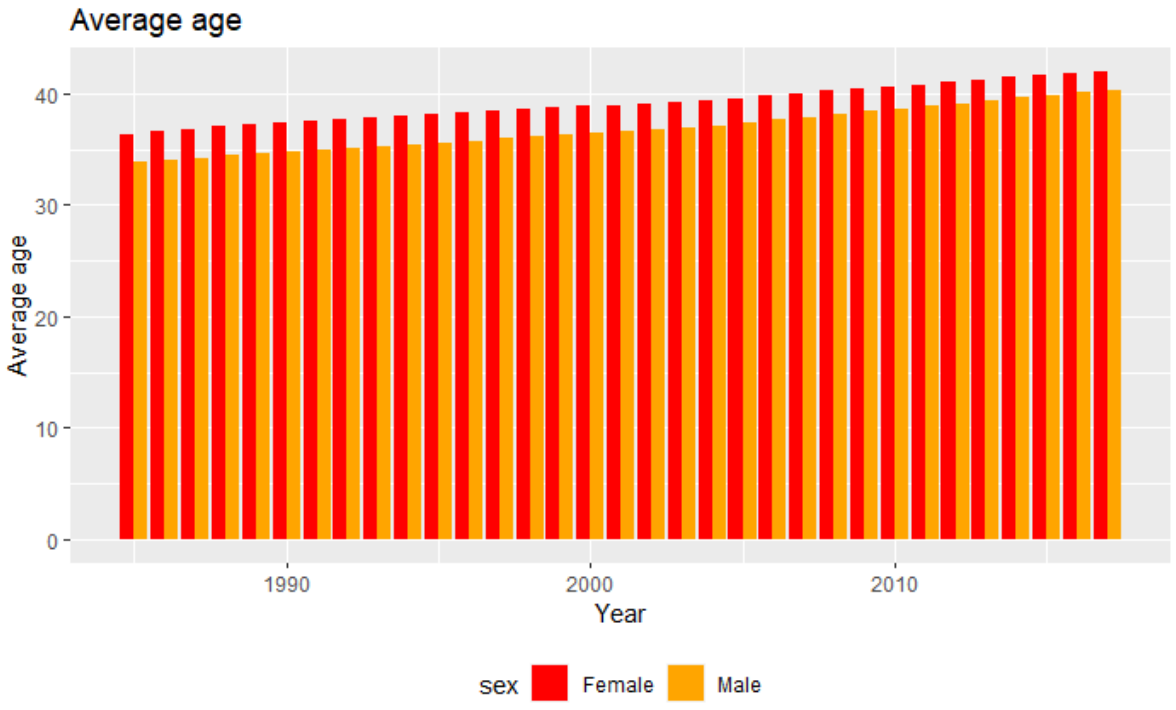


Figure 5: *Average age for males and females*

The trends that were described above based on Figure 3, could be further explained in some extent by looking at the sex-specific life expectancy at birth, as well as the average age for the two populations through the years. Looking first at Figure 4 which was produced by separate data from the Central Agency for Statistics in the Netherlands (CBS), we can see that overall, women have a higher life expectancy than men. Especially before 2011, the difference was larger compared to the recent years where the life expectancy for men started to climb, making the gap smaller. Therefore, the trend that was observed in the population plot of Figure 3 seems reasonable since women are expected on average to live longer than men. Additionally, by taking a look at Figure 5, one can identify a similar pattern. The average age for women is constantly a bit higher than for men, with a gap that is slowly closing through the recent years. That in combination with the fact that the average age for both males and females is constantly increasing, might also explain the steeper mortality curve for women, since the number of older women is constantly increasing.

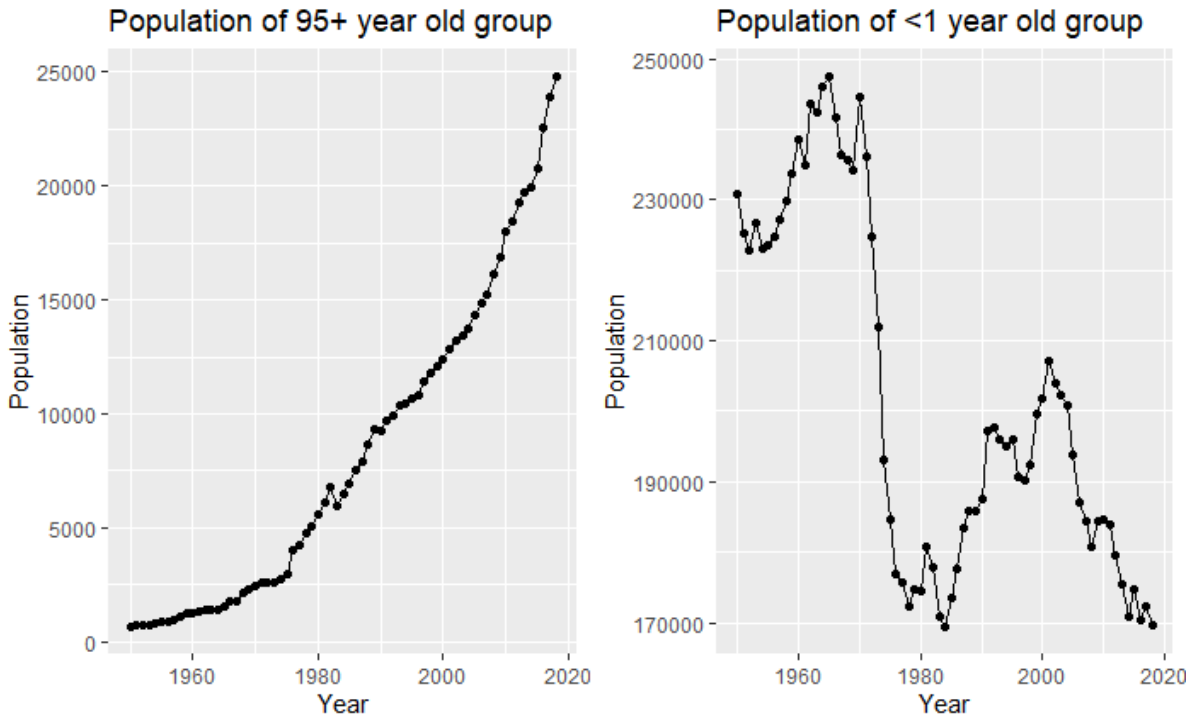


Figure 6: *Population of 95+ year old age group (left) and population of less than one year old newborns (right) for both genders together*

Another interesting trend that can be observed from the two figures that follow, is the change in population of two specific age groups over the years. These age groups involve the people that are over 95 years old, but also the less than one year old newborns. On the left plot, the increasing trend of the population is rather predictable, given the review we had on Figure 4, with the life expectancy and average age of both men and women constantly increasing over the years.

On the right plot, the first thing one can notice, is a steady increase in the population of newborns until the year 1970, which is a feature that could be explained by the increase in fertility rates that took place after the end of World War II, which can be also seen in Figure 7. During the next decade, from 1970 to 1980, a very sharp decrease in the population of newborns took place. This was the period when in most developed countries, there had been a large increase in the use of contraceptive pills, leading to the smallest fertility rates during the 1980s.

Finally, after the 1980s and during the recent years, the population of newborns slightly increased as a result of the increasing fertility rates. The fertility rates can be visualized in Figure 7, which is available through the CBS website.

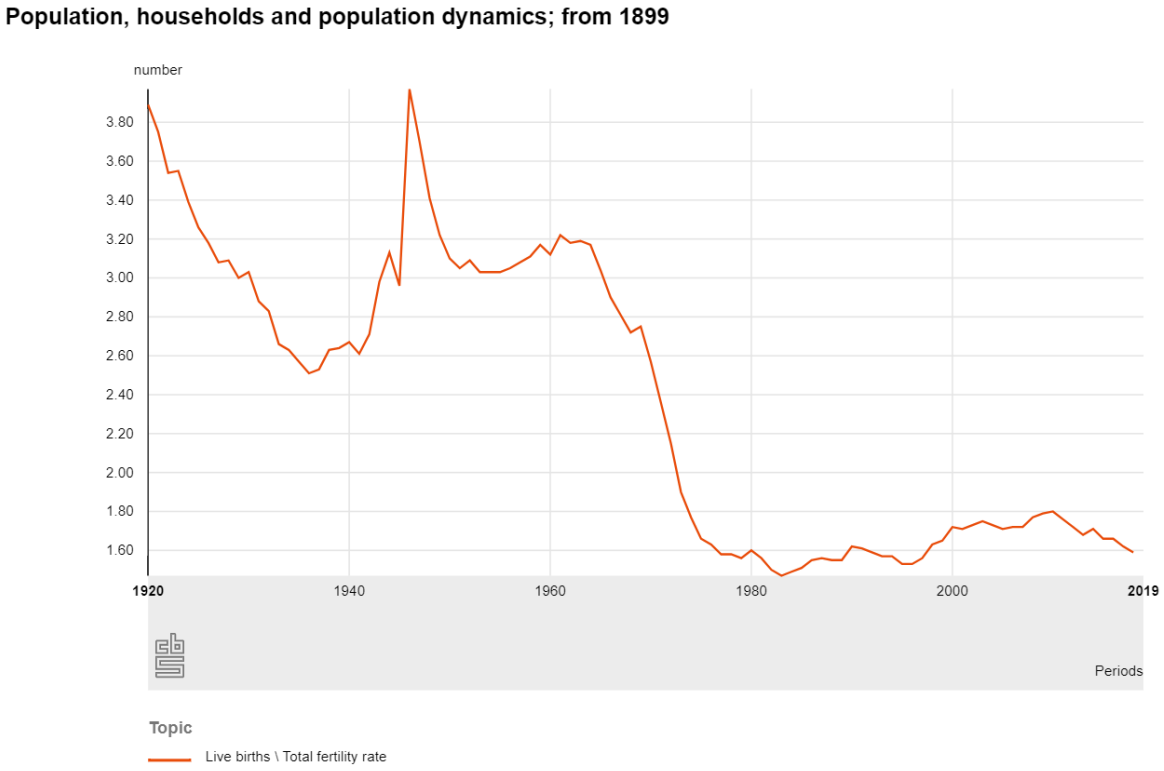


Figure 7: *Total fertility rates in the Netherlands*

An additional way of investigating the properties of our data, involves the visualization and interpretation of the principal components and their corresponding coefficients, which can be obtained from the FPCA method that we discussed in the background section. The principal components can be seen as certain features that explain most of the variation in our outcome.

Usually, the first two principal components explain the largest part of the variation, therefore they are the most informative and interpretable features of our data. In order to draw some inference from the principal components, we have plotted the fitted basis functions along with their corresponding coefficients in Figure 8 and Figure 9.

More specifically, two FPCA models were fitted, one for the male and one for the female population, and their results can be seen in Figure 8 and Figure 9 respectively. In each case only two basis functions were used, which explain 95.14% and 3.67% of the variance in the case of males, and 97.67% and 1.03% of the variance in the case of females. Therefore the first principal component is the one that explains most of the variation in mortality rates for both populations, with the second component having a relatively small impact and probably representing a secondary feature of our data.

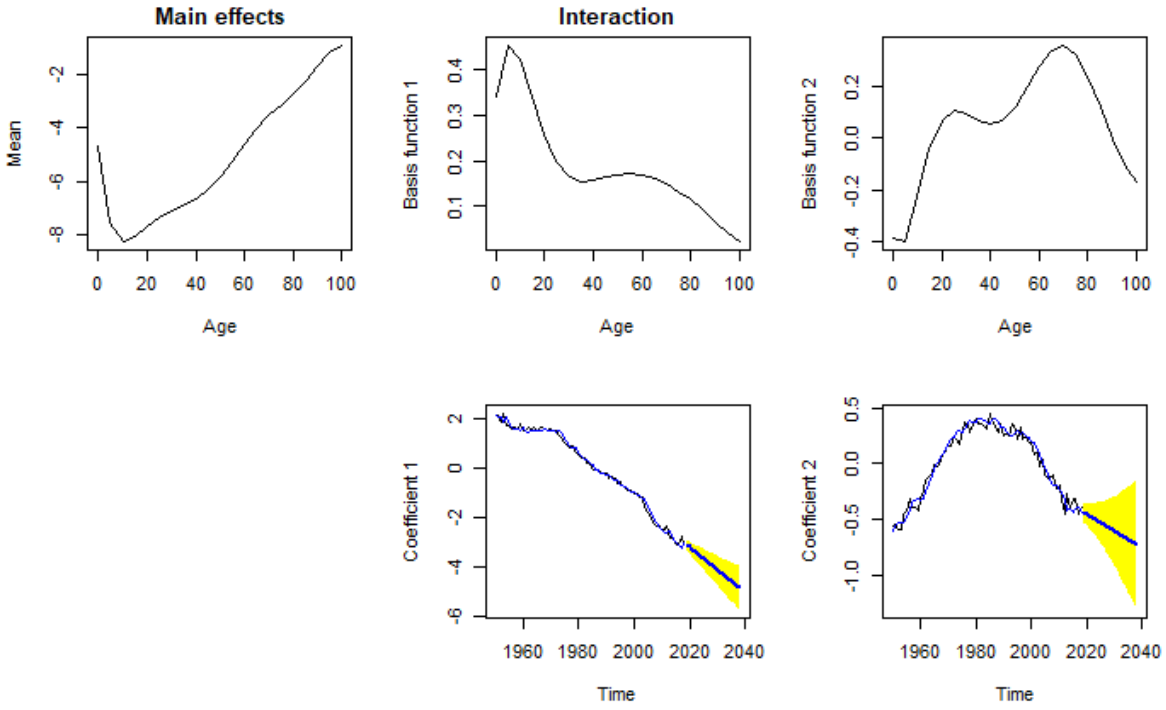


Figure 8: *Mean function (top-left), first two principal components (top-middle and top-right) and principal component scores for males (bottom-middle and bottom-right). The yellow area represents the* 80% *confidence intervals for the estimated coefficients.*

Taking a closer look at Figure 8, we can observe that the first basis function mostly models the age groups of 0 to 20 years old. The corresponding coefficient, shows a steady decreasing trend over the years, which translates to decreasing mortality rates for the respective age groups. Initially, we can observe that the coefficient's value is positive from 1950 until around 1980, which denotes that the mortality rates of the age groups modelled by the corresponding basis function, are larger than those of the mean mortality curve. However, after 1980 the coefficient's value

becomes negative and it keeps decreasing, meaning that the mortality rates after that year, fall below the mean curve.

The second basis function, mainly represents the ages 60-80, but also the ages 20-40 at a lower degree. Now looking at the second coefficient values, we observe an increase in mortality rates for these ages, that starts from 1950 and reaches its peak at around 1990, following a steady decline through the most recent years. One may also notice by the sign of the coefficient values, that approximately between 1970 and 2008, the mortality rates for these age groups were larger than the average. On the other hand, the mortality rates for these ages were below the mean function before 1970 and after 2008. This particular feature can be observed from the rainbow plots in the next section as well (Figure 10).

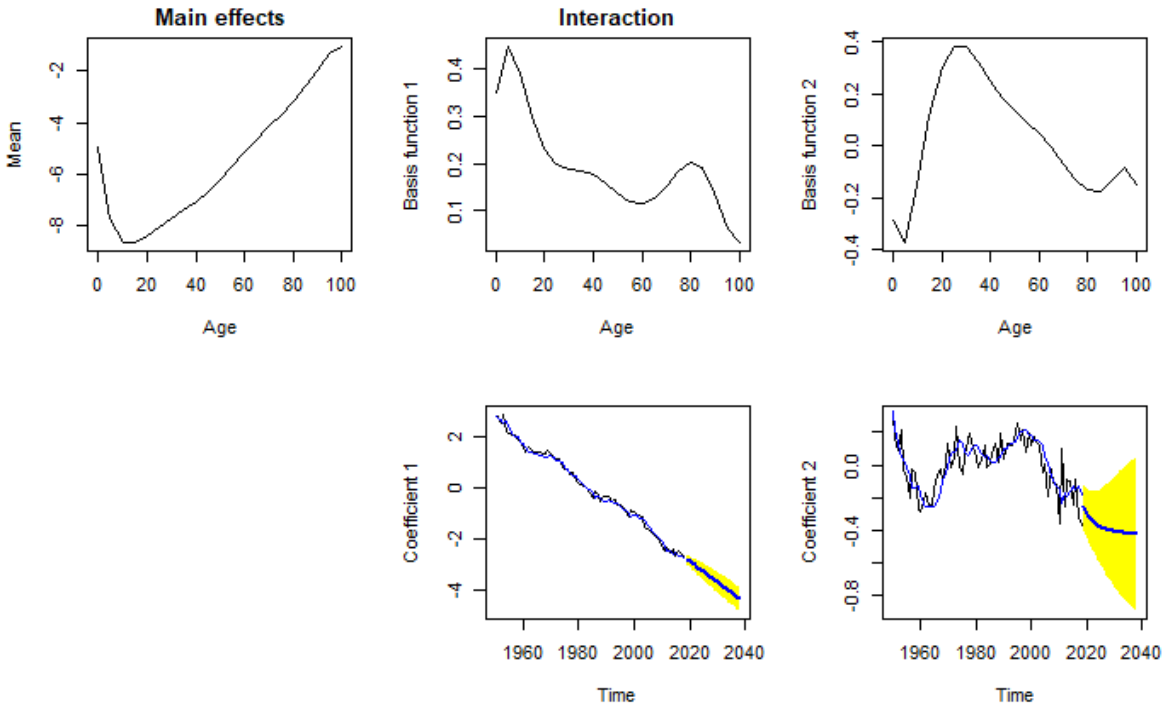


Figure 9: *Mean function (top-left), first two principal components (top-middle and top-right) and principal component scores for females (bottom-middle and bottom-right). The yellow area represents the* 80% *confidence intervals for the estimated coefficients*

Moving on to Figure 9, we can observe a similar behaviour of the first basis function as in the previous figure. In this case, the first basis function mostly corresponds to ages 0-20, but also to ages around 80 years. For these age groups the coefficient values show a constant decrease in mortality rates, which looks a bit more steady in comparison with the one for males. Also, this constant decline in mortality rates for females applies for the older ages too, which is not something that we can say for the mortality rates of males of the same age. Additionally, the mortality rates of the mentioned age groups start to fall below the mean mortality curve shortly

after 1980, when the coefficient values become negative.

The second basis function, models the age groups around 40 to 60 years. Even though the second principal component accounts for only 1% of the variance, we might be able to observe a small period of stagnation for the mortality rates for these age groups if we look at the corresponding coefficient values and the rainbow plots for females (Figure 11). The coefficient values indicate a period of non-decreasing mortality rates which starts at around 1980 and follows until the year 2000, where they start to slowly decline. This feature could be seen from Figure 11 as well.

4.3 Estimation of smooth mortality curves

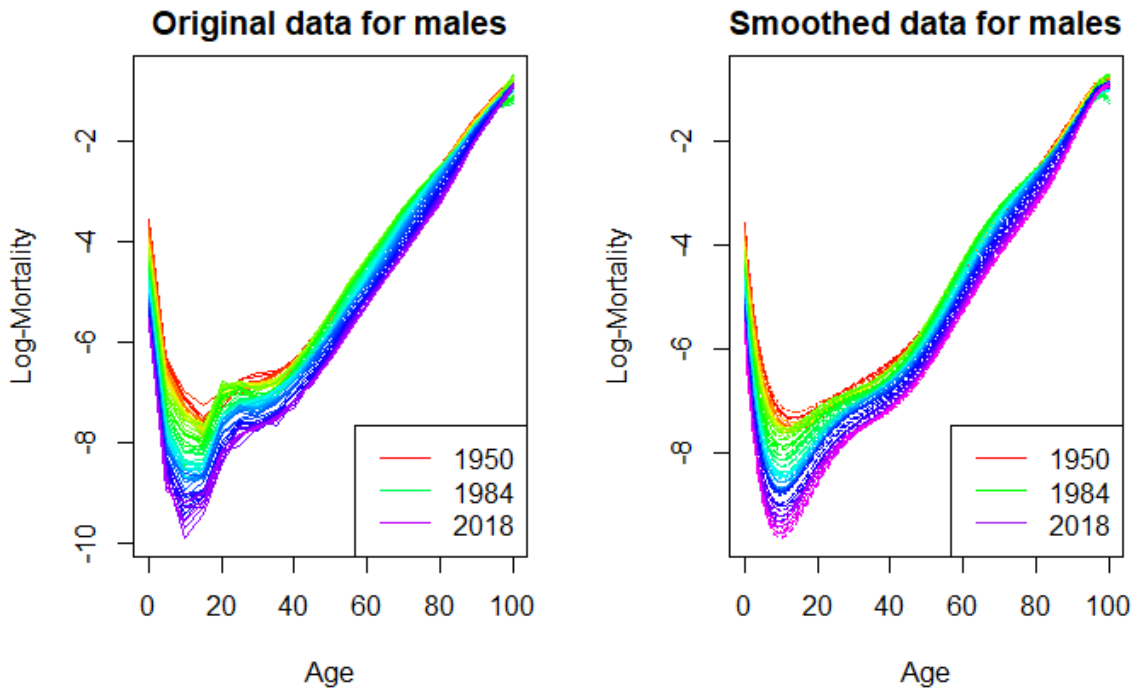


Figure 10: *Rainbow plots of original data (left plot) and smooth data (right plot) for males.*

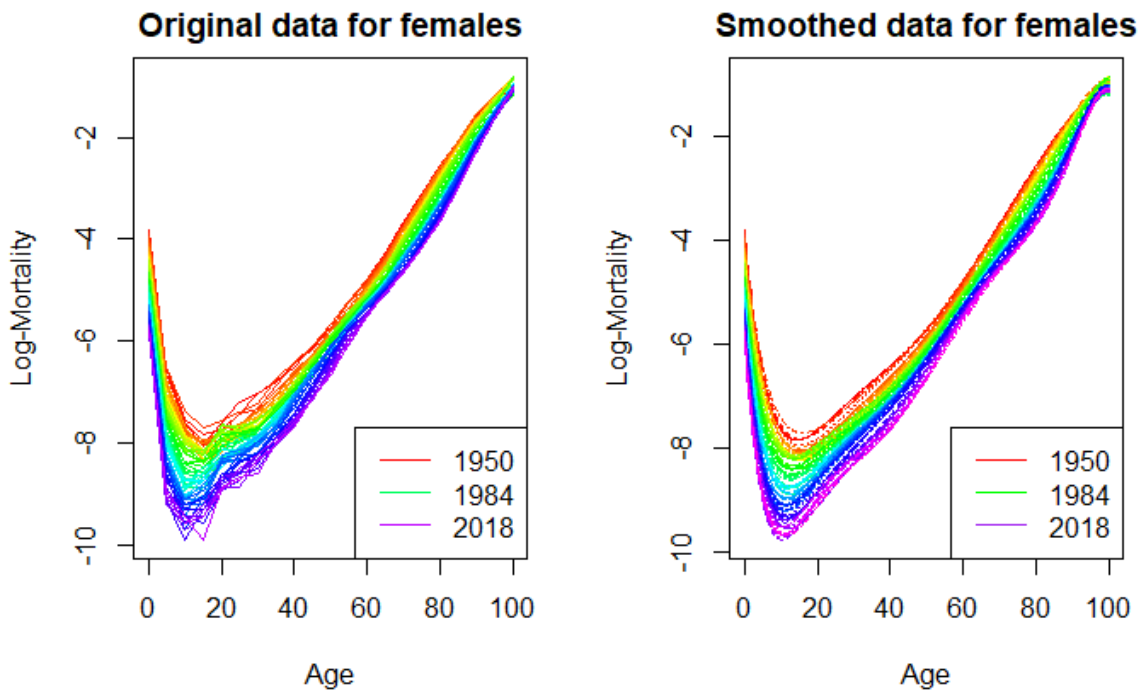


Figure 11: *Rainbow plots of original data (left plot) and smooth data (right plot) for females.*

On the figures displayed above, one can see the rainbow plots that were produced for the male and female populations separately, for the original and smoothed functional data. They depict the age-specific mortality ratios on the natural logarithmic scale, with rainbow colours used for representing the time ordering.

Some of the features that were discussed in the previous subsection using the first two principal components for inference, can be observed from the rainbow plots as well. More specifically, for the male population we can see that especially for the ages 60-80, the warmer colours (which represent the more distant past) are being covered by the curves of the following years. This trend is also observed for the ages 20-40, but with less intensity.

Additionally, if we look closely at Figure 11, we may notice that for the ages 40 to 60 (although it is not as obvious as in the case of the male population), the light blue lines which correspond to the years around 1990 are starting to fade as they are covered by the darker blue lines. This is an indication that there was a period around 1990 and 2000 were the mortality rates for these age groups had stopped decreasing.

The mortality curves are smoothed using penalized B-splines with roughness penalty as explained in Section 2. A B-spline basis of order 9 was fitted using 10 equally spaced knots, therefore consisting of 17 basis functions. That choice was made after experimenting with different values for the order as well as for the number of knots. The results for order less than 9, showed that the curves were underfitting with many of the features that are present in the original data being left out. On the contrary, values larger than 9 started to show signs of overfitting with curves

becoming too "wiggly". As for the number of knots, we observed no impact on the fit for values larger than 10. The process of choosing the optimal value for the penalty term λ, involved the initialization of a sequence of possible λ values on the log_{10} scale, then fitting the B-spline basis for each value of the sequence transformed back to the original scale, and finally extracting the degrees of freedom, generalized cross-validation (GCV) statistic and RMSE. RMSE stands for Root Mean Squared Error and it is given by the following formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_t(\alpha_i) - y_t(\alpha_i))^2} \tag{8}$$

where $\hat{y}_t(\alpha_i)$ are the fitted values of the age group α_i and year t, and $y_t(\alpha_i)$ denotes the true values. Therefore, we get an RMSE value for each year separately, and those values are then averaged to obtain a single error.

The same process is also applied for the computation of the GCV criterion value, which is given by:

$$GCV = \frac{n * SSE}{(n - df)^2}$$

where SSE is the sum of squared errors $SSE = \sum_{i=1}^{n}(\hat{y}_t(\alpha_i) - y_t(\alpha_i))^2$, n is the sample size and df are the degrees of freedom.

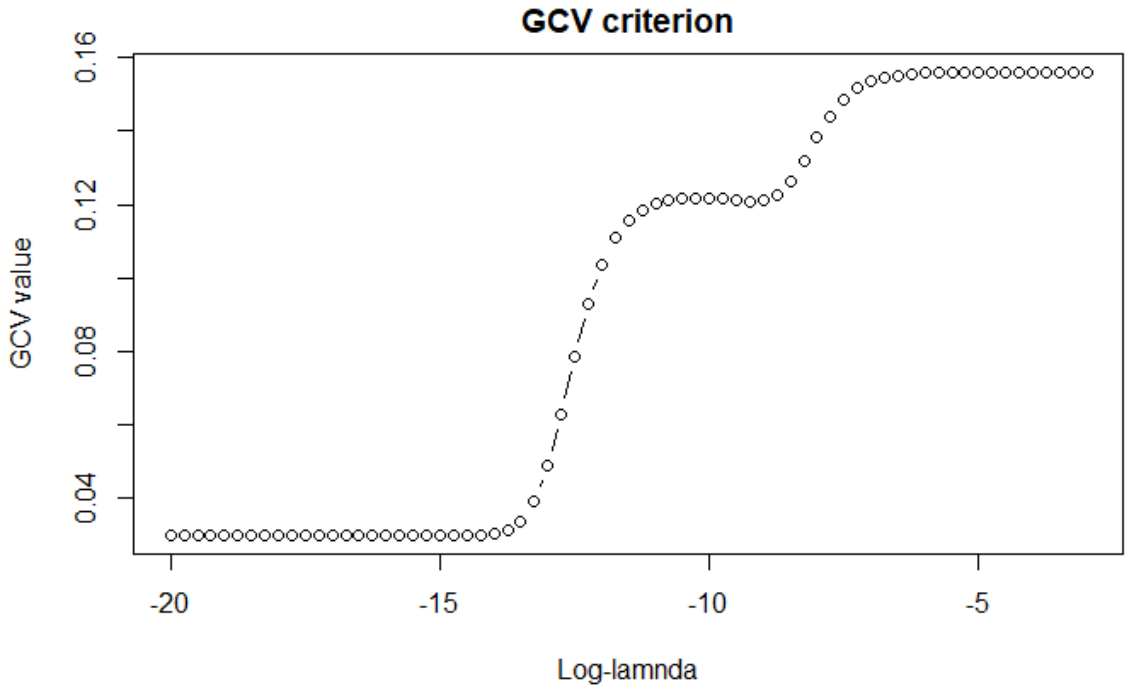


Figure 12: *GCV criterion values with respect to* $log_{10}(\lambda)$.

Both of these measures that were mentioned above, can be used as guidelines for selecting the optimal value of λ by minimizing them, while keeping a relatively low number of degrees of

freedom, as a higher number of degrees of freedom translates to increased model complexity which can lead to overfitting and less amount of actual smoothing.

From Figure 12 one can visualize the change of the GCV criterion value as the $log_{10}(\lambda)$ increases. We can see that the value of $log_{10}(\lambda)$ which minimizes the GCV value is around -15 and that it remains constant for even smaller penalties. Therefore, the optimal choice for λ is 10^{-15} with GCV = 0.03, RMSE = 0.105 and df = 8. However, such a small value for the penalty term produces equivalent results as if no penalization was used.

5 Modeling

In this section, the different models that were applied to the data are going to be discussed, in terms of structure and goodness-of-fit. Along with the FPCA and weighted FPCA models that were explained in Section 2, we also fit a Functional Partial Least Squares (FPLSR) model (Preda and Saporta (2005)). The FPLSR method is similar to the FPCA with the difference that it focuses on identifying components that have the closest relationship with the prediction of the outcome. Instead of finding the most important components based on the percentage of variance that they explain, it does so by maximizing the covariance function between functional predictors and functional responses.

The FPCA, weighted FPCA and FPLSR models were fitted on the smooth log-mortality curves for males and females separately (Hyndman and Shang (2015), Shang et al. (2013)), using the data from 1950 to 1999, while keeping the curves from 2000 to 2018 for the evaluation of their forecast accuracy. Additionally, experiments were made by fitting the models using several different values of K, but since the final results did not show any significant difference in terms of forecast accuracy, a value of $K = 6$ was chosen for all three models.

Table 1 and Table 2 include a summary of the goodness-of-fit for the three models for the male and female population separately, using as measures the Mean Squared Error (MSE) and the Integrated Squared Error (ISE). The MSE is given by the following formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_t(\alpha_i) - y_t(\alpha_i))^2$$

So, the above error measure is calculated for each year, across all the age groups. In contrast, the ISE is calculated for each age group, across all the years. The results from Table 1 and Table 2 include the ISE averaged through all the age groups.

$$ISE = \int_{\alpha_1}^{\alpha_n}(\hat{y}_t(\alpha) - y_t(\alpha))^2 d\alpha$$

From Table 1, we can see that the FPCA model gives the smallest MSE. This means that it has a slightly better fit than the other models when it comes to fitting the annual mortality curves from 1950 to 1999, averaged across all ages. Furthermore, the FPLSR model has the lowest ISE, meaning that it better fits the mortality rates of each age group, averaged across all years.

For the female population, one can see from Table 2 that the weighted FPCA model gives the best goodness-of-fit both in terms of MSE and ISE.

Error	FPCA	FPCA$_w$	FPLSR
MSE	**0,0008**	0,0016	0,0018
ISE	0,0685	0,1349	**0,0338**

Table 1: *Goodness-of-fit measures for the different models, calculated for the male population*

Error	FPCA	FPCA$_w$	FPLSR
MSE	0,0007	**0,0003**	0,0022
ISE	0,0626	**0,0317**	0,0421

Table 2: *Goodness-of-fit measures for the different models, calculated for the female population*

5.1 FPCA models

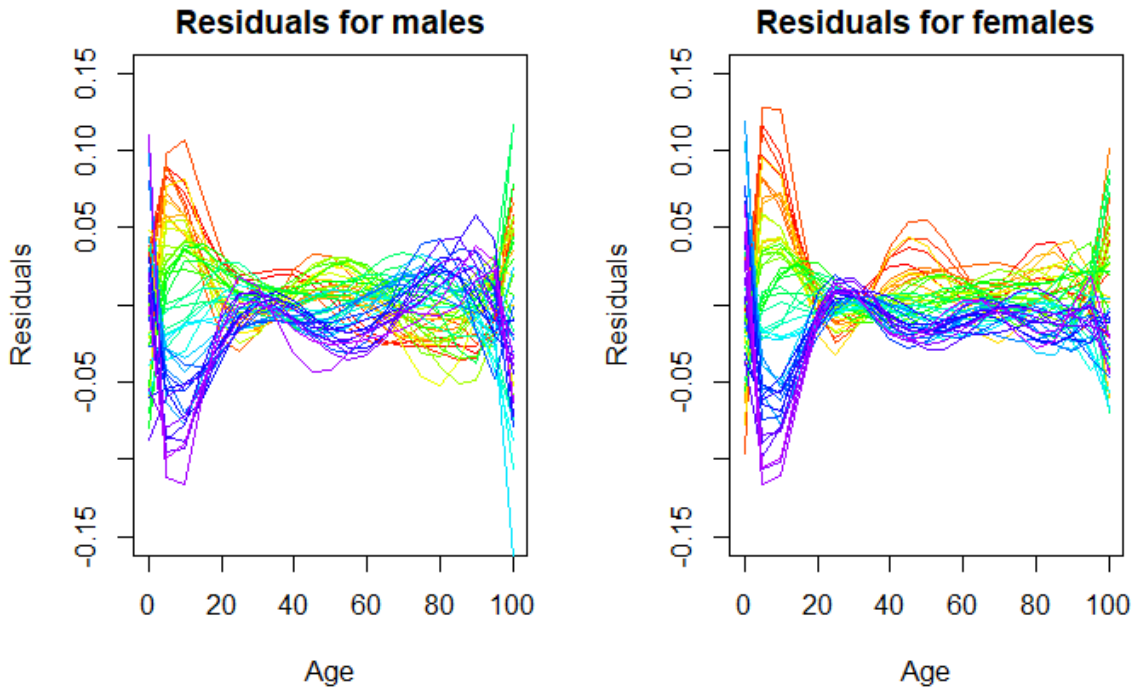


Figure 13: *Residuals for male population (1950-1999).*

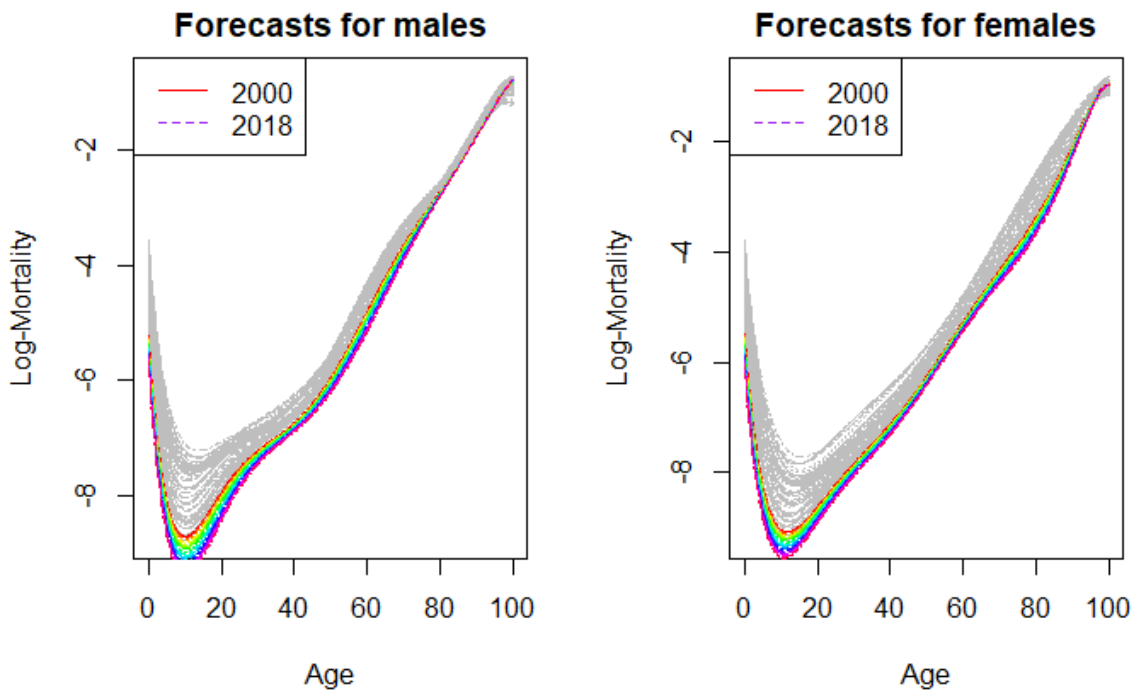


Figure 14: *Forecasts for male population.*

Starting with the FPCA models fitted for the male and female population, one can observe the residuals from Figure 13 and the 19-year ahead forecasts from Figure 14. First of all, the residuals are based on one-step forecasts and they are calculated on the training data, that is the mortality curves from 1950 to 1999, using the rainbow colours for time ordering with the red colour representing the year 1950 and the purple colour the year 1999. The model residuals between the two genders do not seem to have any remarkable differences, while they both have an age interval in common (0-20), in which the errors appear to be larger especially for the years around 1950 and those around 1999.

In Figure 14, the gray lines represent the mortality curves from 1950 to 1999 at which the models were fitted, and the rainbow coloured lines are the 19-step ahead forecasts of the mortality curves from 2000 to 2018 that were also smoothed only for visualization purposes. Overall, the forecasts suggest decreased mortality rates for both genders, and especially for the age groups around 5 to 30 years. However, the forecasts for older ages show that the mortality rates remain stable.

5.2 Weighted FPCA models

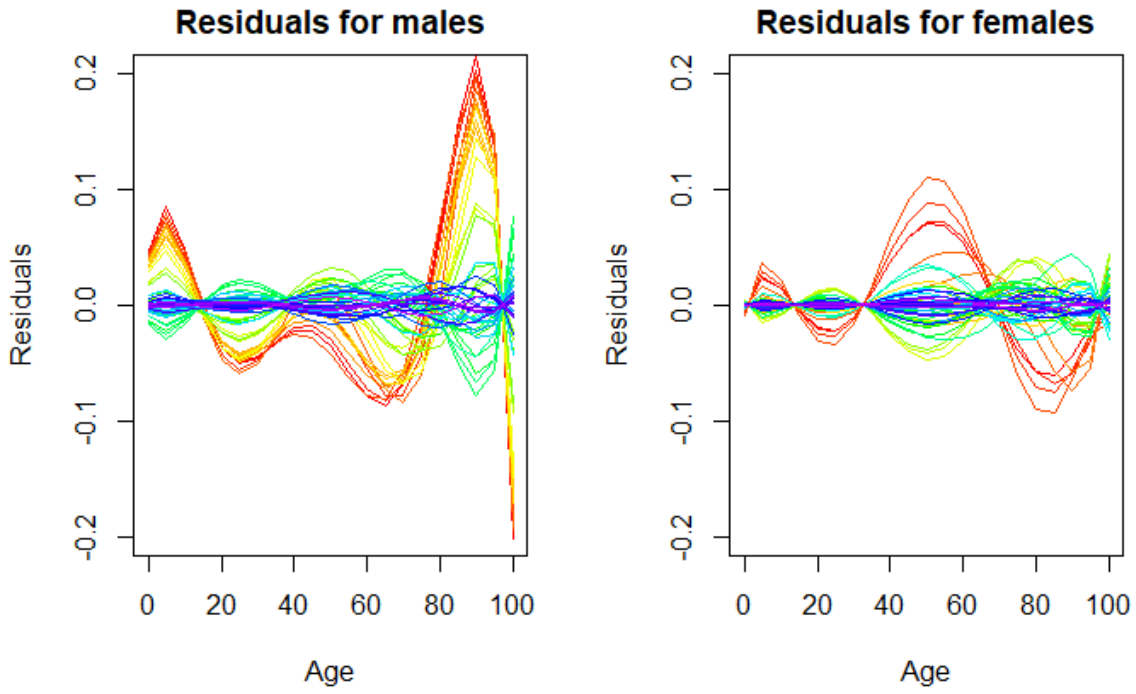


Figure 15: *Residuals for male population (1950-1999).*

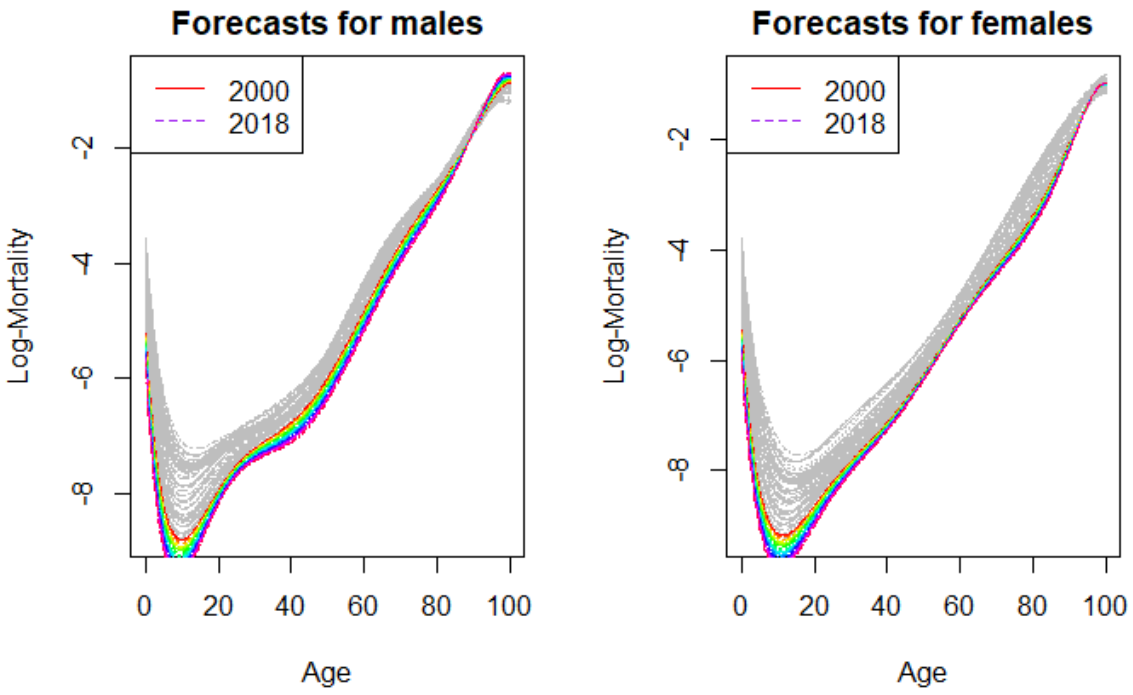


Figure 16: *Forecasts for male population.*

Figure 15 depicts the residuals for the weighted FPCA models, for the male and female population. The result of applying geometrically decaying weights to the mortality curves can be clearly visualized from this figure, as mortality curves of the more distant past receive much less weight than the most recent ones. This is why the most recent curves have residuals closer to zero, contrary to the older mortality curves which show much more unstable residuals.

From Figure 16, one can visualize the forecasts of the mortality curves for the weighted FPCA approach. We may notice that the forecasts are quite comparable to those obtained from the FPCA approach without the use of weights. Especially for the ages 40-80, the forecasts for males show more decreased mortality rates compared to the regular FPCA approach, and for females we observe the opposite. Additionally, in contrast to the regular FPCA approach, were we observed non-decreasing mortality rates for the age groups of 90 years and older males, now we observe an increase in mortality rates for the same age groups.

5.3 FPLSR models

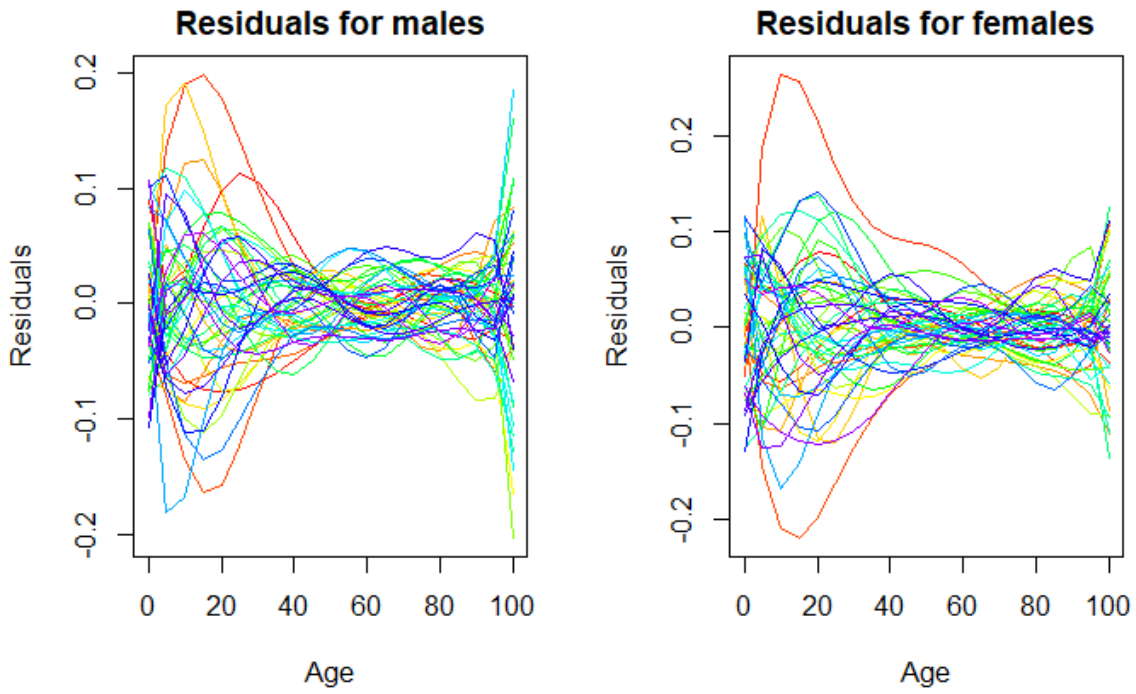


Figure 17: *Residuals for male population (1950-1999).*

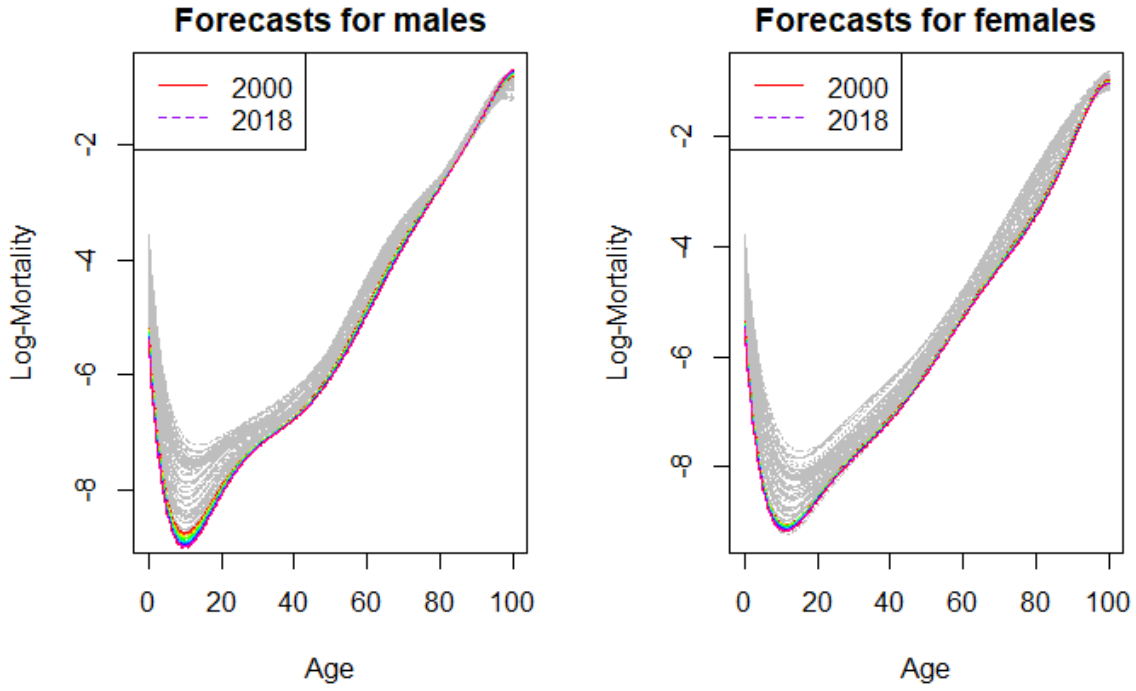


Figure 18: *Forecasts for male population.*

Finally, by looking at Figure 17, for the FPLSR models we observe generally larger residuals compared to the regular FPCA models, with most of the larger residuals concentrated around the age of 20 years for both genders.

The forecasts of the FPLSR models, contrary to the two previous approaches show very stable, non-decreasing mortality rates. More specifically, for the female population we observe almost no decline in mortality rates at all, for any of the age groups. On the other hand, for the male population only a small decrease around the ages between 0 and 20 is visible.

6 Evaluation/Comparison

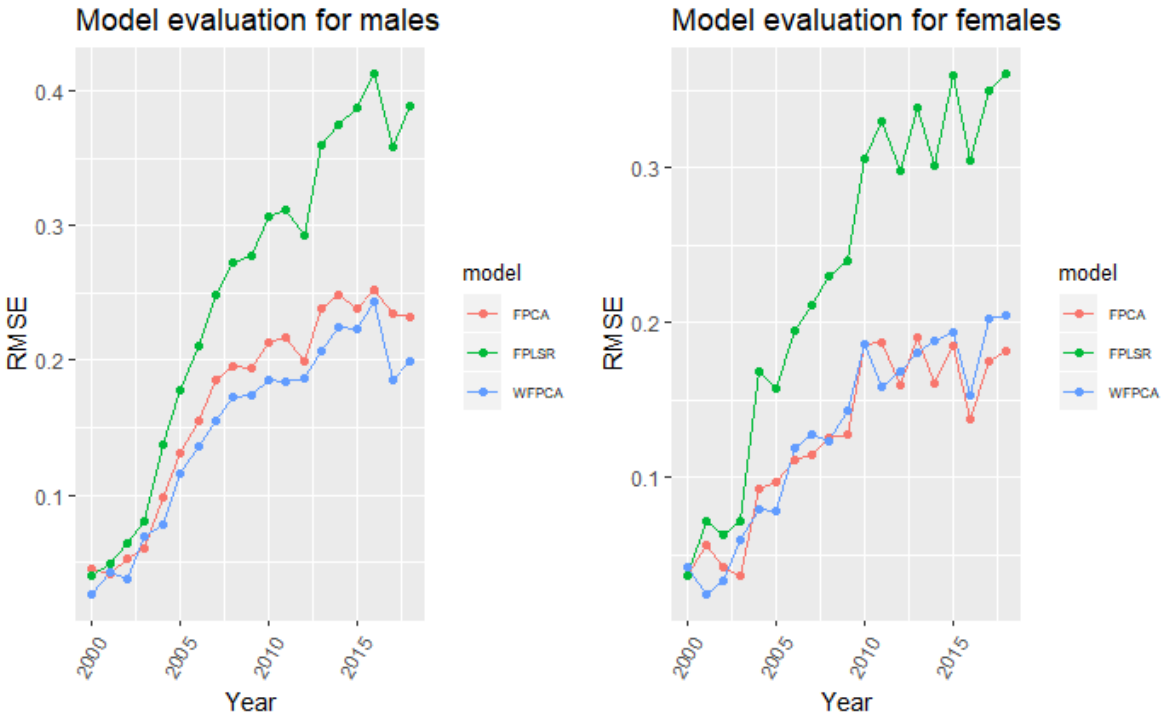


Figure 19: *RMSE calculated for the male and female populations, for each year.*

In order to evaluate the forecast accuracy of our models, we compute the RMSE from equation (8) between the estimated smooth curves and the predicted curves for the years 2000 to 2018. Figure 19 gives us a picture of the RMSE for the three models in each population, calculated for each year. This way we can make a comparison of their forecast accuracy depending on the forecast horizon.

In the case of the male population, the results show that the weighted FPCA approach gives the lowest RMSE for every year from 2000 to 2018, and therefore gives the best forecast accuracy out of the three models. However, the FPCA model has comparable performance with the weighted approach although the errors are a bit larger, and the FPLSR model has clearly the worst performance out of the three models especially for long term forecasts. Moreover, the RMSE for each model averaged over the 19 year period is 0.17 for the FPCA, 0.14 for the weighted FPCA and 0.25 for the FPLSR.

For the female population, the FPCA and weighted FPCA models have very similar accuracy over the years and again the FPLSR model gives the highest errors throughout the 19 year period. Averaging the RMSE over the years we get 0.126 for the FPCA, 0.129 for the weighted FPCA and 0.231 for the FPLSR.

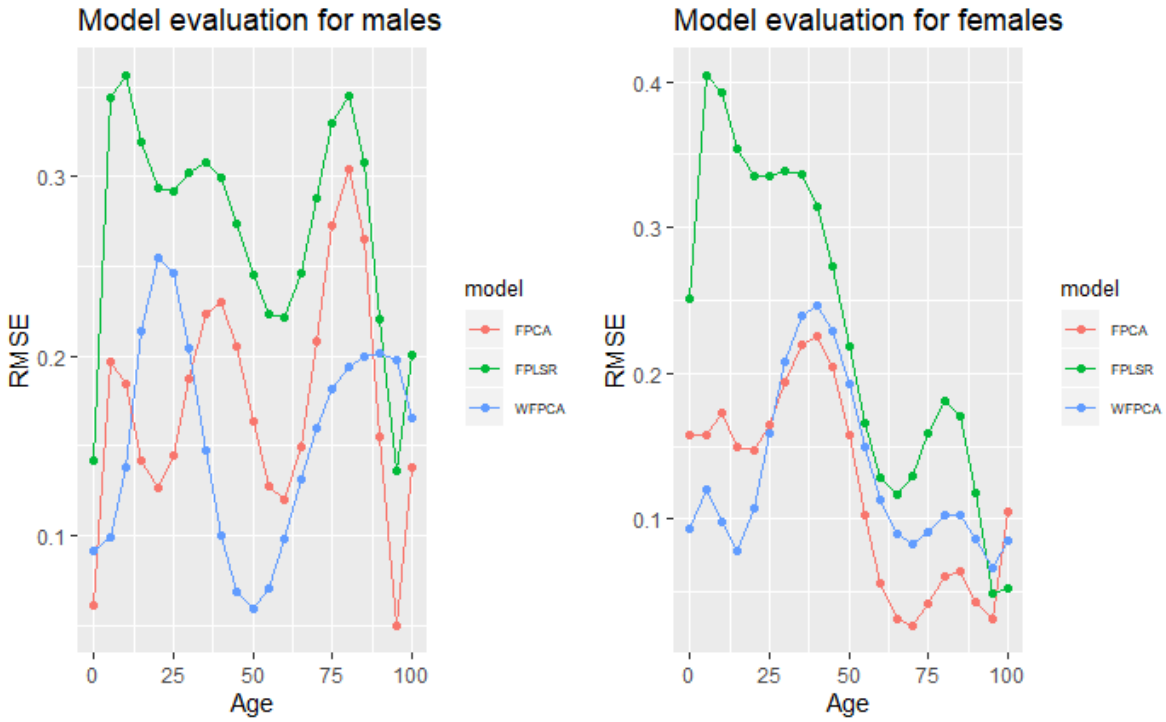


Figure 20: *RMSE calculated for the male and female population, for each year.*

Furthermore, we compare the performances of the models by each age group, as one can see from Figure 20. Again the RMSE is computed, but this time it is evaluated for each age group through the whole 19 year period.

At the left plot for the male population, we can see that for ages larger than 25, the weighted FPCA model has the lowest error. However, for the older ages groups (¿80 years) and for the age groups of 10-30 years, the FPCA model has lower error than the weighted approach. The FPLSR model still has the largest errors for all age groups. The RMSEs for each model, averaged for every age group are 0.153 for the weighted FPCA model, 0.174 for the FPCA and 0.271 for the FPLSR model.

For the female population, we can observe from the right plot that the weighted FPCA model outperforms the FPCA for the ages up to 25 years. But from this age point and on the FPCA has a bit lower errors, as it seems to be the most accurate for ages over 60 years. Again, the FPLSR model is the least accurate for all age groups. Additionally, the averaged errors are 0.119 for the FPCA, 0.13 for the weighted FPCA and 0.229 for the FPLSR.

7 Conclusion

In this project, we applied FDA methods to the age-sex specific mortality rates of the Netherlands in order to conduct exploratory analysis of various trends, fit different models on the mortality curves and to evaluate them and compare their performance.

From the analysis we conducted on the mortality rate curves we found some worth mentioning trends both for males and females. Particularly, from investigating the first two principal components for males and their corresponding coefficients, we observed an increase in mortality rates mostly for the age groups 60-80, that peaked around 1990. Janssen et al. (2003) conducted a research about the stagnation of mortality decline amongst elderly people in the Netherlands, which suggests a couple of possible causes for this fact, mostly related to particular diseases that made an important impact on their mortality. It was suggested that some causes of death such as smoking-related cancers and chronic obstructive pulmonary disease went through an increase in mortality until the 1990s. Especially for men, smoking rates peaked during the 20th century, which was a period that men who were born from 1897 to 1917, reached an old age while having a lifetime exposure to smoking. On the other hand, for the female population the age groups 40-60 also experienced a smaller period of non-decreasing mortality, which started around the 1980s and lasted until the year 2000. However, based on this research, smoking could not have been the only reason for this increase in mortality rates, but other reasons such as changes in health care or social services for elderly people could have contributed as well.

Regarding the different models that were fitted on the mortality curves, we performed a two-sided evaluation for each model and each gender. Particularly, from computing the RMSE on the test data (for the years 2000-2018) with respect to age and year separately, we found that for the male population the weighted FPCA approach gives the most accurate forecasts. In the case of females the FPCA and weighted FPCA approaches give very similar results, with the FPCA giving a slightly lower average RMSE than the weighted FPCA. For both populations the FPLSR model produced the least accurate forecasts and was clearly outperformed by the other two models. We may also note that after experimentation, the number of basis functions K used for decomposition did not have a significant impact on the final results for any of the three models, so it was fixed to $K = 6$.

Furthermore, there are some limitations in this project that should be addressed and one of those is related to the available age-specific mortality data. We used data from 1950 to 2018, from which only the mortality curves from 1950 until 1999 were used for model fitting, while the rest of them were used for evaluation. Additional data of the more distant past, could potentially help us reveal new information regarding the performance of the models. Another limitation is related to future forecasts. This is an issue that would involve a considerable amount of uncertainty, since mortality rate patterns might be affected by the recent outbreak of SARS-CoV-2, especially for specific age groups.

Additionally, in order to be able to extend the methods that were used in this master thesis, further population characteristics could be incorporated such as fertility and/or migration. That would compose a more thorough demographic study using Functional Data Analysis and it would also give us the chance to further compare the models that were fitted on the mortality curves, therefore giving us a more complete comparison. Another method that could be added to the analysis as a slightly different approach that could be compared with the already applied methods, is the product-ratio method which was introduced by Hyndman, Booth, and Yasmeen (2013) as a

method of producing coherent forecasts. This method could be applied to forecast mortality rates when two or more sub-populations are present (in our case males and females), in order to ensure that the forecasts do not unreasonably diverge between the two sub-populations.

A Appendix

A.1 Data cleaning and exploratory analysis

```
1
2 install.packages("tidyverse")
3 install.packages("Hmisc")
4 install.packages("VIM")
5 install.packages("rainbow")
6 install.packages("ftsa")
7 install.packages("fda")
8 install.packages("glmnet")
9 install.packages("fda.usc")
10 install.packages("Rfssa")
11 install.packages("gridExtra")
12
13 #Import libraries
14 library(fda.usc)
15 library(ftsa)
16 library(fda)
17 library(rainbow)
18 library(VIM)
19 library(tidyverse)
20 library(magrittr)
21 library(dplyr)
22 library(base)
23 library(ggplot2)
24 library(glmnet)
25 library(gridExtra)
26
27 #---------- READING THE DATA ----------#
28
29 Mortality <- as.data.frame(read.table("7052eng_UntypedDataSet_23032020
    _165127.csv", header = TRUE, sep = ";"))
30 Population <- as.data.frame(read.table("7461eng_UntypedDataSet_
    23032020_165214.csv", header = TRUE, sep = ";"))
31
32 #Adjust the names of some variables
33 colnames(Population)[1] <- "ID"
34 colnames(Population)[4] <- "Year"
35 colnames(Population)[5] <- "Population"
36
37
38 colnames(Mortality)[1] <- "ID"
39 colnames(Mortality)[4] <- "Year"
```

```r
colnames(Mortality)[5] <- "Total_deaths"

## Keep only the variables of interest
Population <- Population %>%
  select(c(Sex, Age, Year, Population))

Mortality <- Mortality %>%
  select(c(Sex, Age, Year, Total_deaths))

## The variable Year in both datasets needs to be converted to numeric values

# first we have to convert the variable Year from factor to character
Population$Year <- as.character(Population$Year)
Mortality$Year <- as.character(Mortality$Year)

to_number <- function(x) as.numeric(strsplit(x, "J")[[1]][1])

Population$Year <- unlist(lapply(Population$Year, FUN = to_number))
Mortality$Year <- unlist(lapply(Mortality$Year, FUN = to_number))

## Change the encoding of the Sex variable to M, F and T
levels(Population$Sex) <- c("M", "F", "T")
levels(Mortality$Sex) <- c("M", "F", "T")


## Fix Age variable encodings which are a bit different in the two datasets.
# We need to create a new age group in the Population dataset (51300),
# containing the 1 to 5 ages like in the Mortality data.
# So, we need to substract the population of newborns from the 0-5 age group.
# We want to keep the newborns as a separate group.


#levels(as.factor(Mortality$Age)) #10010: 0, 51300: 1-5
#levels(as.factor(Population$Age))#10010: 0, 70100: 0-5

Population <- Population %>%
  spread(Age, value = Population) %>%
  mutate(`51300` = `70100` - `10010`) %>%
  gather(key = "Age", value = "Population", `10000`:`51300`)


```

```r
## Merge the two datasets
# Now in the Ratio_data we have the 0-5 age group (70100) but also the 0 age group (10010)
Ratio_data <- merge(Population, Mortality, by = c("Sex", "Year", "Age"))


#Plots
p1 <-  ggplot(Population %>%
                filter(Age == 10000), aes(x = Year, y = Population)) +
  geom_point(aes(col=Sex)) +
  geom_line(aes(col=Sex)) +
  labs(title = "Population in the Netherlands", subtitle = "For males and females") +
  xlab("Year")

p2 <- ggplot(Mortality %>%
               filter(Age == 10000), aes(x = Year, y = Total_deaths)) +
  geom_point(aes(col=Sex)) +
  geom_line(aes(col=Sex)) +
  labs(title = "Mortality in the Netherlands", subtitle = "For males and females") +
  xlab("Year")

grid.arrange(p1, p2, ncol = 2)


# No missing values
summary(Ratio_data)


#---------- CALCULATING MORTALITY RATIOS ----------

# Mortality ratios
Ratio_data$Mortality_ratio <- round(Ratio_data$Total_deaths/Ratio_data$Population, digits = 5)

# checking for weird values
summary(Ratio_data$Mortality_ratio)

# log ratios
Ratio_data$log_ratio <- log(Ratio_data$Mortality_ratio)

# Plot sex-specific log ratios
```

```
ggplot(Ratio_data %>%
         filter(Age == 10000, Sex != "T"), aes(x = Year, y = log_ratio
            , group = Sex)) +
  geom_point(aes(col = Sex)) +
  geom_line(aes(col = Sex)) +
  labs(title = "Log-Mortality rates in the Netherlands", subtitle = "
     For males and females separately") +
  ylab("log-mortality rate")


# Plot population of two age groups
p3 <- ggplot(Ratio_data %>%
               filter(Age == 22000, Sex == "T" ), aes(x = Year, y =
                  Population, group = 1)) +
  geom_point() +
  geom_line() +
  labs(title = "Population of 95+ year old group") +
  xlab("Year")

p4 <- ggplot(Ratio_data %>%
               filter(Age == 10010, Sex == "T" ), aes(x = Year, y =
                  Population, group = 1)) +
  geom_point() +
  geom_line() +
  labs(title = "Population of <1 year old group") +
  xlab("Year")

grid.arrange(p3, p4, ncol = 2)


#Plot Life expectancies of the two genders
Life_exp <- as.data.frame(read.table("Health_expectancy__since_1981_
   29032020_194215.csv", header = FALSE, sep = ","))
Life_exp <- as.data.frame(Life_exp[-c(1:5, 20), ])
Life_exp <- Life_exp %>%
  separate(col = 1, into = c("Sex", "Age", "Periods", "Life.expectancy
     ", "Life_exp_in_good_health", "Life_exp_no_physical_limit", "Life
     _exp_no_chr", "mental_health", "GALI"), sep = ";") %>%
  select(c(Sex, Age, Periods, Life.expectancy))



ggplot(Life_exp,
       aes(x = as.factor(Periods), y = Life.expectancy, fill = Sex)) +
  geom_bar(position="dodge", stat="identity") +
```

```
159   labs(title = "Life expectancy") +
160   ylab("Life expectancy") +
161   theme(axis.text.x = element_text(angle=60, hjust=1)) +
162   xlab("Year") +
163   scale_fill_manual(values=c("black", "purple"))
164
165 levels(as.factor(Ratio_data$Age))
```

A.2 Estimation of smooth mortality curves

```
166 #------------------------------ PREPARATION FOR SMOOTHING
    ------------------------
167
168 # First we need to sort our data by age groups.
169 Ratio_data <- Ratio_data %>%
170   filter(Age != 10000)
171
172 # Since the age group 22000 is for 95+ year old people, we assign it
      to the largest number
173 Ratio_data[Ratio_data$Age == 22000, 3] <- 72000
174 # Sort by age
175 Ratio_data <- Ratio_data %>%
176   arrange(Age)
177
178 # For fitting least squares we need wide format
179 Ratio_wide <- Ratio_data %>%
180   select(-c(Population, Mortality_ratio, Total_deaths)) %>%
181   spread(Year, log_ratio)
182
183 ### Bsplines smoothing
184
185 #Creating the basis function decomposition using B-splines
186 # We have 20 age groups
187 #0 corresponds to newborns (10010) and 100 corresponds 95+ age group
188 age_points <- seq(from = 0, to = 100, by = 5)
189
190
191
192 #Test the impact of K, at the smoothed curve (only for visualization)
193
194
195 #Now we need to estimate the coefficients in order to produce the
      smooth function.
196 #We are doing that using OLS
```

```r
#Y is the mortality ratios of females of every age, at 2000. We want to fit a smooth curve to its data points.
Y <- Ratio_wide %>% filter(Sex == "F") %>% select('2000')
Y <- unlist(Y)


### K = 4
basis1 <- create.bspline.basis(c(0, 100), nbasis = 4)
basis_eval1 <- eval.basis(age_points, basis1)


model1 <- lm(Y ~ 0 + basis_eval1)
Yhat1 <- model1$fitted.values

### K = 50
basis2 <- create.bspline.basis(c(0, 100), nbasis = 50)
basis_eval2 <- eval.basis(age_points, basis2)


model2 <- lm(Y ~ 0 + basis_eval2)
Yhat2 <- model2$fitted.values


### Plot
plot(age_points, Y, type="n",lwd=4, col="black",
     xlab="Age", ylab="Mortality ratio",
     main = "B-splines curves")

points(age_points, Y, pch=1, cex=.5, col="blue", lwd=1)
lines(age_points, Yhat1, lwd=1, col="red")
lines(age_points, Yhat2, lwd=1, col="black")
legend("bottomright", legend = c("K=4", "K=50"), col = c("red", "black"),lty=1:2)



#---------------------------------- PENALIZED B-SPLINES SMOOTHING ----------------------------


set.seed(1993)
# Discrete mortality observations for males and females
ym <- as.matrix(Ratio_wide %>% filter(Sex == "M") %>% select(-c(Sex, Age)))
yf <- as.matrix(Ratio_wide %>% filter(Sex == "F") %>% select(-c(Sex, Age)))
```

```r


### In order to choose the best value for lambda, we compute the RMSE, GCV and degrees of freedom
# for a sequence of values for lambda.
summarise_penalties <- function(loglambdas, basis_expansion, data, argvalues, fdParobj){
  #Initializing an empty dataframe, based on the length of lambdas sequence
  results <- numeric(0)

  #Looping for every values of lambda
  for (i in 1:length(loglambdas)){
    loglambda <- loglambdas[i]
    #Fitting the penalized B-splines basis
    fdParobj$lambda <- 10^(loglambda)
    smooth_basis <- smooth.basis(argvals = argvalues, y = data, fdParobj)

    #Extracting predicted curves, df, GCV and RMSE
    smooth_curve <- smooth_basis$fd
    df <- round(smooth_basis$df, digits = 5)
    gcv <- round(mean(smooth_basis$gcv), digits = 5)
    error_set <- eval.fd(argvalues, smooth_curve) - data
    RMSE <- mean(apply(error_set, 2, function(x) round(sqrt(mean(x^2)), digits = 3)))
    #Adding a row in the results table
    results <- rbind(results, c(loglambda, df, gcv, RMSE))
  }

  results <- as.data.frame(results)
  colnames(results) <- c("log-lambda", "df", "GCV", "RMSE")
  return(results)
}


loglambdas <- seq(-20, -3, 0.25)
basis_exppen <- create.bspline.basis(c(0,100), norder = 9, breaks = 10)
fdParobj <- fdPar(basis_exppen, Lfdobj = NULL, lambda = 10^(-5))
## RESULTS
penalty_summary <- summarise_penalties(loglambdas, basis_expansion = basis_exppen, data = ym, argvalues = age_points, fdParobj)

```

```
275
276 #Plot the change of GCV value
277 plot(penalty_summary$`log-lambda`, penalty_summary$GCV, xlab = "Log-
    lamnda", ylab = "GCV value", main = "GCV criterion", type = "b")
278
279 #Optimal log lambda is equal to -9. Has the lowest GCV and RMSE values
    .
280 lambda <- 10^(-15)
281 fdParobj <- fdPar(basis_exppen, Lfdobj = NULL, lambda)
282
283
284
285
286
287 #Smooth curves for males
288 mortality_smooth_m <- smooth.basis(argvals = age_points, y = ym,
    fdParobj)
289 mortalityfd_m <- mortality_smooth_m$fd
290
291
292
293
294 #Smooth curves for females
295 mortality_smooth_f <- smooth.basis(argvals = age_points, y = yf,
    fdParobj)
296 mortalityfd_f <- mortality_smooth_f$fd
297
298
299
300
301
302
303
304 ## Unsmoothed data
305 dataM <- fts(age_points, Ratio_wide %>%
306                filter(Sex == "M") %>%
307                select(-c(Sex, Age)), xname = "Age", yname = "Log-
                   Mortality")
308
309
310
311
312 dataF <- fts(age_points, Ratio_wide %>%
313                filter(Sex == "F") %>%
314                select(-c(Sex, Age)), xname = "Age", yname = "Log-
```

```
                    Mortality")
315
316
317
318
319
320 ### Plots for males
321 par(mfrow = c(1,2))
322 plot(dataM, plot.type = "functions",
323       plotlegend = TRUE, legendpos = "bottomright",
324       main = "Original data for males")
325
326 plot.fd(mortalityfd_m, col = rainbow(80),
327          xlab = "Age", ylab = "Log-Mortality",
328          main = "Smoothed data for males")
329 legend("bottomright", legend = c("1950", "1984", "2018"), col = c("red
      ", "green", "purple"), lty = 1)
330
331
332
333 ### Plots for females
334 par(mfrow = c(1,2))
335 plot(dataF, plot.type = "functions",
336       plotlegend = TRUE, legendpos = "bottomright",
337       main = "Original data for females")
338
339 plot.fd(mortalityfd_f, col = rainbow(80),
340          xlab = "Age", ylab = "Log-Mortality",
341          main = "Smoothed data for females")
342 legend("bottomright", legend = c("1950", "1984", "2018"), col = c("red
      ", "green", "purple"), lty = 1)
343
344
345 # Smooth functional time series
346 smoothM <- fts(age_points, eval.fd(age_points, mortalityfd_m), xname =
       "Age", yname = "Log-Mortality")
347 smoothF <- fts(age_points, eval.fd(age_points, mortalityfd_f), xname =
       "Age", yname = "Log-Mortality")
```

A.3 Modeling

```
348 ###---------------------- FPCA MODELS ----------------------
349
350 ##First two principal components for visualization
```

```
plot(forecast(ftsm(smoothM, order = 2), h = 20), "components")


###Training set
y_train_m <- fts(age_points, as.data.frame(eval.fd(age_points,
    mortalityfd_m)) %>% select('1950':'1999'), xname = "Age", yname = "
    Log-Mortality_ratios")


###Forecasts for different values of K
fpca_m_4 <- forecast(ftsm(y_train_m, order = 4), h = 19)
fpca_m_6 <- forecast(ftsm(y_train_m, order = 6), h = 19)
fpca_m_8 <- forecast(ftsm(y_train_m, order = 8), h = 19)



###Goodness of fit
#For each choice of K we get slightly different errors.
summary(ftsm(y_train_m, order = 4))
summary(ftsm(y_train_m, order = 6))
summary(ftsm(y_train_m, order = 8))



###RESIDUALS
plot.fmres(residuals.fm(ftsm(y_train_m, order = 6)),
           type = "fts", xlab = "Age", main = "Residuals_for_males_(
              FPCA)")



#Smooth training data
train_smooth_m <- smooth.basis(argvals = age_points,
                               y = as.matrix(Ratio_wide %>%
                                             filter(Sex == "M") %>%
                                             select(-c(Sex, Age))
                                                %>%
                                             select('1950':'1999')),
                               fdParobj = fdParobj)$fd

#Smooth forecasts
fpca_forecasts_m_6 <- smooth.basis(argvals = age_points, y = fpca_m_6$
    mean$y,
                                   fdParobj)$fd

```

```
391
392 ### Plot data with forecasts
393 #Smooth data
394 plot.fd(train_smooth_m, col = "grey",
395         xlab = "Age", ylab = "Log-Mortality",
396         main = "Forecasts for males")
397 legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
      , lty = 1:2)
398 #Smooth forecasts
399 plot.fd(fpca_forecasts_m_6, col = rainbow(19), add = TRUE)
400
401
402
403
404
405 ###------------------------------ Weighted FPCA models
      ------------------------------
406
407 ###Forecasts for males
408 wfpca_m_4 <- forecast(ftsm(y_train_m, order = 4, weight = TRUE), h =
      19)
409 wfpca_m_6 <- forecast(ftsm(y_train_m, order = 6, weight = TRUE), h =
      19)
410 wfpca_m_8 <- forecast(ftsm(y_train_m, order = 8, weight = TRUE), h =
      19)
411
412
413 ###Goodness of fit
414 summary(ftsm(y_train_m, order = 4, weight = TRUE))
415 summary(ftsm(y_train_m, order = 6, weight = TRUE))
416 summary(ftsm(y_train_m, order = 8, weight = TRUE))
417
418
419 ### RESIDUALS
420 plot.fmres(residuals.fm(ftsm(y_train_m, order = 6, weight = TRUE)),
421            type = "fts", xlab = "Age", main = "Residuals for males (
                 Weighted FPCA)")
422
423
424 #The difference between the weighted FPCA and the regular FPCA can be
      clearly seen from the residuals
425 #The weighted FPCA focuses on most recent data which have low
      residuals, and not so much on older data
426 #which have clearly larger residuals.
427
```

```r


#Smoothing the forecasts
wfpca_forecasts_m_6 <- smooth.basis(argvals = age_points, y = wfpca_m_
    6$mean$y,
                                    fdParobj = fdParobj)$fd


### Plot data with forecasts
plot.fd(train_smooth_m, col = "grey",
        xlab = "Age", ylab = "Log-Mortality",
        main = "Forecasts for males")
legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
    , lty = 1:2)


### Apply the same methodology on the female population



###---------------------- FPCA MODELS ----------------------


##First two principal components for visualization
plot(forecast(ftsm(smoothF, order = 2), h = 20), "components")


###Training set
y_train_f <- fts(age_points, as.data.frame(eval.fd(age_points,
    mortalityfd_f)) %>% select(`1950`:`1999`), xname = "Age", yname = "
    Log-Mortality ratios")


###Forecasts for different values of K
fpca_f_4 <- forecast(ftsm(y_train_f, order = 4), h = 19)
fpca_f_6 <- forecast(ftsm(y_train_f, order = 6), h = 19)
fpca_f_8 <- forecast(ftsm(y_train_f, order = 8), h = 19)


###Goodness of fit
#For each choice of K we get slightly different errors.
summary(ftsm(y_train_f, order = 4))
summary(ftsm(y_train_f, order = 6))
summary(ftsm(y_train_f, order = 8))

```

```
469
470 ### RESIDUALS
471 plot.fmres(residuals.fm(ftsm(y_train_f, order = 6)),
472             type = "fts", xlab = "Age", main = "Residuals for females (
                FPCA)")
473
474
475 #Smoothing the forecasts
476 fpca_forecasts_f_6 <- smooth.basis(argvals = age_points, y = fpca_f_6$
    mean$y,
477                                    fdParobj = fdParobj)$fd
478
479
480 #Smooth training data
481 train_smooth_f <- smooth.basis(argvals = age_points,
482                                y = as.matrix(Ratio_wide %>%
483                                              filter(Sex == "F") %>%
484                                              select(-c(Sex, Age))
                                                 %>%
485                                              select('1950':'1999')),
486                                fdParobj = fdParobj)$fd
487
488
489 ### Plot data with forecasts
490 plot.fd(train_smooth_f, col = "grey",
491         xlab = "Age", ylab = "Log-Mortality",
492         main = "Forecasts for females")
493 legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
    , lty = 1:2)
494 plot.fd(fpca_forecasts_f_6, col = rainbow(19), add = TRUE)
495
496
497
498
499
500 ###---------------------------- Weighted FPCA models
    ----------------------------

501
502 ###Forecasts for females
503 wfpca_f_4 <- forecast(ftsm(y_train_f, order = 4, weight = TRUE), h =
    19)
504 wfpca_f_6 <- forecast(ftsm(y_train_f, order = 6, weight = TRUE), h =
    19)
505 wfpca_f_8 <- forecast(ftsm(y_train_f, order = 8, weight = TRUE), h =
    19)
```

```
506
507
508 ###Goodness of fit
509 summary(ftsm(y_train_f, order = 4, weight = TRUE))
510 summary(ftsm(y_train_f, order = 6, weight = TRUE))
511 summary(ftsm(y_train_f, order = 8, weight = TRUE))
512
513
514 ### RESIDUALS
515 plot.fmres(residuals.fm(ftsm(y_train_f, order = 6, weight = TRUE)),
516             type = "fts", xlab = "Age", main = "Residuals for females (
                Weighted FPCA)")
517
518
519 #Smoothing the forecasts
520 wfpca_forecasts_f_6 <- smooth.basis(argvals = age_points, y = wfpca_f_
    6$mean$y,
521                                     fdParobj = fdParobj)$fd
522
523
524 ### Plot data with forecasts
525 plot.fd(train_smooth_f, col = "grey",
526         xlab = "Age", ylab = "Log-Mortality",
527         main = "Forecasts for females")
528 legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
    , lty = 1:2)
529 plot.fd(wfpca_forecasts_f_6, col = rainbow(19), add = TRUE)
530 plot.fd(wfpca_forecasts_m_6, col = rainbow(19), add = TRUE)
531
532
533
534
535 ###------------------Functional Partial Least Squares
    ------------------------
536
537 ### FPLSR model fitted on the male population
538
539 ##Forecasts for different values of K
540 fplsr_m_4 <- forecastfplsr(y_train_m, components = 4, h = 19)
541 fplsr_m_6 <- forecastfplsr(y_train_m, components = 6, h = 19)
542 fplsr_m_8 <- forecastfplsr(y_train_m, components = 8, h = 19)
543
544
545 ###Goodness of fit
546 #For each choice of K we get slightly different errors.
```

```
summary(fplsr(y_train_m, order = 4))
summary(fplsr(y_train_m, order = 6))
summary(fplsr(y_train_m, order = 8))


### RESIDUALS
#Plot smooth residuals (K = 6)

fplsr_res_m <- fts(age_points, t(residuals.fm(fplsr(y_train_m, order =
    6))$z), xname = "Age", yname = "Residuals")

plot(fplsr_res_m , plot.type = "functions",
     plotlegend = TRUE, legendpos = "bottomright",
     main = "Residuals for males (FPLSR)")



# Smooth forecasts
fplsr_forecasts_m_6 <- smooth.basis(argvals = age_points, y = fplsr_m_
    6$y,
                                    fdParobj = fdParobj)$fd

#Plot data with forecasts for K=6
plot.fd(train_smooth_m, col = "grey",
        xlab = "Age", ylab = "Log-Mortality",
        main = "Forecasts for males")
legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
    , lty = 1:2)
plot.fd(fplsr_forecasts_m_6, col = rainbow(19), add = TRUE)



### FPLSR model fitted on the female population


##Forecasts for different values of K
fplsr_f_4 <- forecastfplsr(y_train_f, components = 4, h = 19)
fplsr_f_6 <- forecastfplsr(y_train_f, components = 6, h = 19)
fplsr_f_8 <- forecastfplsr(y_train_f, components = 8, h = 19)


###Goodness of fit
#For each choice of K we get slightly different errors.
summary(fplsr(y_train_f, order = 4))
summary(fplsr(y_train_f, order = 6))
```

```
589 summary(fplsr(y_train_f, order = 8))
590
591
592 ### RESIDUALS
593 #Plot smooth residuals (K = 6)
594 fplsr_res_f <- fts(age_points, t(residuals.fm(fplsr(y_train_f, order =
        6))$z), xname = "Age", yname = "Residuals")
595
596 plot(fplsr_res_f , plot.type = "functions",
597       plotlegend = TRUE, legendpos = "bottomright",
598       main = "Residuals for females (FPLSR)")
599
600
601
602 # Smooth forecasts
603 fplsr_forecasts_f_6 <- smooth.basis(argvals = age_points, y = fplsr_f_
      6$y,
604                                     fdParobj = fdParobj)$fd
605
606 #Plot data with forecasts for K=6
607 plot.fd(train_smooth_f, col = "grey",
608         xlab = "Age", ylab = "Log-Mortality",
609         main = "Forecasts for females")
610 legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
      , lty = 1:2)
611 plot.fd(fplsr_forecasts_f_6, col = rainbow(19), add = TRUE)
612
613
614
615 ##---------- PRODUCING PLOTS ----------
616
617 ###RESIDUALS FPCA
618 par(mfrow = c(1,2))
619
620 plot.fmres(residuals.fm(ftsm(y_train_m, order = 6)),
621            type = "fts", xlab = "Age", ylim = c(-0.15, 0.15), main = "
               Residuals for males")
622
623 plot.fmres(residuals.fm(ftsm(y_train_f, order = 6)),
624            type = "fts", xlab = "Age", ylim = c(-0.15, 0.15), main = "
               Residuals for females")
625
626
627 par(mfrow = c(1,2))
628 ### Plot data with forecasts FPCA
```

```
#Smooth data
plot.fd(train_smooth_m, col = "grey",
        xlab = "Age", ylab = "Log-Mortality",
        main = "Forecasts for males")
legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
    , lty = 1:2)
#Smooth forecasts
plot.fd(fpca_forecasts_m_6, col = rainbow(19), add = TRUE)

plot.fd(train_smooth_f, col = "grey",
        xlab = "Age", ylab = "Log-Mortality",
        main = "Forecasts for females")
legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
    , lty = 1:2)
plot.fd(fpca_forecasts_f_6, col = rainbow(19), add = TRUE)


par(mfrow = c(1,2))
### RESIDUALS WFPCA
plot.fmres(residuals.fm(ftsm(y_train_m, order = 6, weight = TRUE)),
           type = "fts", xlab = "Age", ylim = c(-0.2, 0.2), main = "
              Residuals for males")

plot.fmres(residuals.fm(ftsm(y_train_f, order = 6, weight = TRUE)),
           type = "fts", xlab = "Age", ylim = c(-0.2, 0.2), main = "
              Residuals for females")


par(mfrow = c(1,2))
### Plot data with forecasts WFPCA
plot.fd(train_smooth_m, col = "grey",
        xlab = "Age", ylab = "Log-Mortality",
        main = "Forecasts for males")
legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
    , lty = 1:2)
plot.fd(wfpca_forecasts_m_6, col = rainbow(19), add = TRUE)


plot.fd(train_smooth_f, col = "grey",
        xlab = "Age", ylab = "Log-Mortality",
        main = "Forecasts for females")
legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
    , lty = 1:2)
plot.fd(wfpca_forecasts_f_6, col = rainbow(19), add = TRUE)

```

```
668
669
670 # Residuals FPLSR
671 par(mfrow = c(1,2))
672 plot(fplsr_res_m , plot.type = "functions", legendpos = "bottomright",
673       main = "Residuals for males")
674
675 plot(fplsr_res_f , plot.type = "functions", legendpos = "bottomright",
676       main = "Residuals for females")
677
678
679 # FORECASTS FPLSR
680 par(mfrow = c(1,2))
681 plot.fd(train_smooth_m, col = "grey",
682          xlab = "Age", ylab = "Log-Mortality",
683          main = "Forecasts for males")
684 legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
      , lty = 1:2)
685 plot.fd(fplsr_forecasts_m_6, col = rainbow(19), add = TRUE)
686
687 plot.fd(train_smooth_f, col = "grey",
688          xlab = "Age", ylab = "Log-Mortality",
689          main = "Forecasts for females")
690 legend("topleft", legend = c("2000", "2018"), col = c("red", "purple")
      , lty = 1:2)
691 plot.fd(fplsr_forecasts_f_6, col = rainbow(19), add = TRUE)
```

A.4 Evaluation

```
751 #------------------------ MODEL EVALUATION FOR MALES
      ------------------------------

752
753 ### We compute the RMSE for every year
754
755 ##Test set
756 y_test_m <- as.data.frame(eval.fd(age_points, mortalityfd_m)) %>%
      select(`2000`:`2018`)
757
758
759 ### Compute the RMSE for each curve (2000-2018)
760 ## For every model
761
762 # FPCA models
763 fpca_m_4_rmse <- apply(fpca_m_4$mean$y - y_test_m, 2, function(x)
```

```
    round(sqrt(mean(x^2)), digits = 3))
fpca_m_6_rmse <- apply(fpca_m_6$mean$y - y_test_m, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))
fpca_m_8_rmse <- apply(fpca_m_8$mean$y - y_test_m, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))

# Weighted FPCA models
wfpca_m_4_rmse <- apply(wfpca_m_4$mean$y - y_test_m, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))
wfpca_m_6_rmse <- apply(wfpca_m_6$mean$y - y_test_m, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))
wfpca_m_8_rmse <- apply(wfpca_m_8$mean$y - y_test_m, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))

# FPLSR models
fplsr_m_4_rmse <- apply(fplsr_m_4$y - y_test_m, 2, function(x) round(
    sqrt(mean(x^2)), digits = 3))
fplsr_m_6_rmse <- apply(fplsr_m_6$y - y_test_m, 2, function(x) round(
    sqrt(mean(x^2)), digits = 3))
fplsr_m_8_rmse <- apply(fplsr_m_8$y - y_test_m, 2, function(x) round(
    sqrt(mean(x^2)), digits = 3))


eval_males <- as.data.frame(rbind(fpca_m_6_rmse, wfpca_m_6_rmse, fplsr
    _m_6_rmse))

eval_males$model <- c("FPCA", "WFPCA", "FPLSR")

apply(eval_males[ ,-20], 1, FUN = mean)


#------------------------ MODEL EVALUATION FOR FEMALES
    ------------------------------

##Test set
y_test_f <- as.data.frame(eval.fd(age_points, mortalityfd_f)) %>%
    select(`2000`:`2018`)

### Compute the RMSE for each curve (2000-2018)
## For every model
fpca_f_4_rmse <- apply(fpca_f_4$mean$y - y_test_f, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))
fpca_f_6_rmse <- apply(fpca_f_6$mean$y - y_test_f, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))
fpca_f_8_rmse <- apply(fpca_f_8$mean$y - y_test_f, 2, function(x)
```

```
    round(sqrt(mean(x^2)), digits = 3))
795
796 wfpca_f_4_rmse <- apply(wfpca_f_4$mean$y - y_test_f, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))
797 wfpca_f_6_rmse <- apply(wfpca_f_6$mean$y - y_test_f, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))
798 wfpca_f_8_rmse <- apply(wfpca_f_8$mean$y - y_test_f, 2, function(x)
    round(sqrt(mean(x^2)), digits = 3))
799
800 # FPLSR models
801 fplsr_f_4_rmse <- apply(fplsr_f_4$y - y_test_f, 2, function(x) round(
    sqrt(mean(x^2)), digits = 3))
802 fplsr_f_6_rmse <- apply(fplsr_f_6$y - y_test_f, 2, function(x) round(
    sqrt(mean(x^2)), digits = 3))
803 fplsr_f_8_rmse <- apply(fplsr_f_8$y - y_test_f, 2, function(x) round(
    sqrt(mean(x^2)), digits = 3))
804
805 eval_females <- as.data.frame(rbind(fpca_f_6_rmse, wfpca_f_6_rmse,
    fplsr_f_6_rmse))
806
807 eval_females$model <- c("FPCA", "WFPCA", "FPLSR")
808
809
810 apply(eval_females[ ,-20], 1, FUN = mean)
811
812
813
814 # MALES
815 p5 <- ggplot( eval_males %>%
816                 gather(key = "Year", value = "RMSE", `2000`:`2018`),
817               aes(x = as.numeric(Year), y = RMSE, group = model)) +
818   geom_line(aes(col = model)) +
819   geom_point(aes(col = model)) +
820   labs(title = "Model evaluation for males") +
821   ylab("RMSE") +
822   theme(axis.text.x = element_text(angle=60, hjust=1)) +
823   xlab("Year") +
824   theme(legend.title = element_text(color = "black", size = 10),
825         legend.text = element_text(color = "black", size = 6))
826
827
828 # FEMALES
829 p6 <- ggplot( eval_females %>%
830                 gather(key = "Year", value = "RMSE", `2000`:`2018`),
831               aes(x = as.numeric(Year), y = RMSE, group = model)) +
```

```
  geom_line(aes(col = model)) +
  geom_point(aes(col = model)) +
  labs(title = "Model evaluation for females") +
  ylab("RMSE") +
  theme(axis.text.x = element_text(angle=60, hjust=1)) +
  xlab("Year") +
  theme(legend.title = element_text(color = "black", size = 10),
        legend.text = element_text(color = "black", size = 6))


grid.arrange(p5, p6, ncol = 2)



### Compute the RMSE for each age group
## For every model


### MALE POPULATION

# FPCA models
fpca_m_6_rmse2 <- apply(fpca_m_6$mean$y - y_test_m, 1, function(x)
    round(sqrt(mean(x^2)), digits = 3))

# Weighted FPCA models
wfpca_m_6_rmse2 <- apply(wfpca_m_6$mean$y - y_test_m, 1, function(x)
    round(sqrt(mean(x^2)), digits = 3))

# FPLSR models
fplsr_m_6_rmse2 <- apply(fplsr_m_6$y - y_test_m, 1, function(x) round(
    sqrt(mean(x^2)), digits = 3))

eval_males2 <- as.data.frame(rbind(fpca_m_6_rmse2, wfpca_m_6_rmse2,
    fplsr_m_6_rmse2))
colnames(eval_males2) <- as.character(age_points)
eval_males2$model <- c("FPCA", "WFPCA", "FPLSR")

apply(eval_males2[ ,-22], 1, FUN = mean)

### FEMALE POPULATION

# FPCA models
fpca_f_6_rmse2 <- apply(fpca_f_6$mean$y - y_test_f, 1, function(x)
    round(sqrt(mean(x^2)), digits = 3))

```

```r
# Weighted FPCA models
wfpca_f_6_rmse2 <- apply(wfpca_f_6$mean$y - y_test_f, 1, function(x) round(sqrt(mean(x^2)), digits = 3))

# FPLSR models
fplsr_f_6_rmse2 <- apply(fplsr_f_6$y - y_test_f, 1, function(x) round(sqrt(mean(x^2)), digits = 3))

eval_females2 <- as.data.frame(rbind(fpca_f_6_rmse2, wfpca_f_6_rmse2, fplsr_f_6_rmse2))
colnames(eval_females2) <- as.character(age_points)
eval_females2$model <- c("FPCA", "WFPCA", "FPLSR")

apply(eval_females2[ ,-22], 1, FUN = mean)

# MALES
p7 <- ggplot( eval_males2 %>%
                gather(key = "Age", value = "RMSE", `0`:`100`),
              aes(x = as.numeric(Age), y = RMSE, group = model)) +
  geom_line(aes(col = model)) +
  geom_point(aes(col = model)) +
  labs(title = "Model evaluation for males") +
  ylab("RMSE") +
  xlab("Age") +
  theme(legend.title = element_text(color = "black", size = 10),
        legend.text = element_text(color = "black", size = 6))


# FEMALES
p8 <- ggplot( eval_females2 %>%
                gather(key = "Age", value = "RMSE", `0`:`100`),
              aes(x = as.numeric(Age), y = RMSE, group = model)) +
  geom_line(aes(col = model)) +
  geom_point(aes(col = model)) +
  labs(title = "Model evaluation for females") +
  ylab("RMSE") +
  xlab("Age") +
  theme(legend.title = element_text(color = "black", size = 10),
        legend.text = element_text(color = "black", size = 6))


grid.arrange(p7, p8, ncol = 2)
```

References

Lee, Ronald (2000). "The Lee-Carter method for forecasting mortality, with various extensions and applications". In: *North American actuarial journal* 4.1, pp. 80–91.

Ramsay, JO and Bernard W Silverman (2001). "Functional data analysis". In:

Janssen, Fanny et al. (2003). "Stagnation in mortality decline among elders in the Netherlands". In: *The Gerontologist* 43.5, pp. 722–734.

Preda, Cristian and Gilbert Saporta (2005). "Clusterwise PLS regression on a stochastic process". In: *Computational Statistics & Data Analysis* 49.1, pp. 99–108.

Hyndman, Rob J and Md Shahid Ullah (2007). "Robust forecasting of mortality and fertility rates: a functional data approach". In: *Computational Statistics & Data Analysis* 51.10, pp. 4942–4956.

Hyndman, Rob J and Heather Booth (2008). "Stochastic population forecasts using functional data models for mortality, fertility and migration". In: *International Journal of Forecasting* 24.3, pp. 323–342.

Hyndman, Rob J and Han Lin Shang (2009). "Forecasting functional time series". In: *Journal of the Korean Statistical Society* 38.3, pp. 199–211.

Aguilera, Ana M and MC Aguilera-Morillo (2013). "Comparative study of different B-spline approaches for functional data". In: *Mathematical and Computer Modelling* 58.7-8, pp. 1568–1579.

Hyndman, Rob J, Heather Booth, and Farah Yasmeen (2013). "Coherent mortality forecasting: the product-ratio method with functional time series models". In: *Demography* 50.1, pp. 261–283.

Shang, Hanlin et al. (2013). "ftsa: An R package for analyzing functional time series". In:

Hyndman, R and Han Lin Shang (2015). "ftsa: Functional time series analysis". In: *R package version* 4.