

Registration and Display of Functional Data

Mahdi Bakhshi

A Thesis
In
The Department
of
Mathematics and Statistics

Presented in Partial Fulfilment of the Requirements
for the degree of Master of Science (Mathematics) at
Concordia University
Montreal, Quebec, Canada

August, 2018

©Mahdi Bakhshi, 2018

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mahdi Bakhshi**

Entitled: **Registration and Display of Functional Data**

and submitted in partial fulfillment of the requirements for the degree of

Master of Science (Mathematics)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Examiner
Dr. Debaraj Sen

_____ Examiner
Dr. Lisa Kakinami

_____ Thesis Supervisor
Dr. Arusharka Sen

Approved by _____
Chair of Department or Graduate Program Director

Dean of Faculty

Date _____

CONCORDIA UNIVERSITY

Abstract

Registration and Display of Functional Data

Department of Mathematics and Statistics

Master of Science (Mathematics)

Mahdi Bakhshi

Functional data refer to data which are in the form of functions or smooth curves that are assessed at a finite, but large subset of some interval. In this thesis, we explore methods of functional data analysis, especially curve registration, in the context of climate changes in a group of 16 cities of the United States. In the first step, spline functions were developed in order to convert the raw data into functional objects. Data are available in function forms, but the mean function which was obtained by the unregistered curve fails to produce a satisfactory estimator. This means that the mean function does not resemble any of the observed curves. A significant problem with most functional data analyses is that of misaligned curves (Ramsay & Silverman, 2005). Curve registration is one method in functional data analysis that attempts to solve this problem. In the second step, we used curve registration method based on “landmarks alignment” and “continuous monotone registration” in order to construct a precise measurement of the average temperature. The results show the differences between unregistered data and registered data and a significant rise of the temperature in U.S. cities within the last few decades.

Acknowledgements

I wish to acknowledge my appreciation to all the people who have helped me in the completion of this thesis. I am grateful to my supervisor, Dr. Arusharka Sen, who has been extremely helpful to me during the period of my studies in the faculty of Mathematics and Statistics, at Concordia University.

I'd like to express my thanks to my beloved wife, Farnoosh, for encouraging me throughout this experience. I could never have accomplished this thesis without her constant support, and understanding.

I must also thank my parents for their support and patience throughout my life. Thank you both for giving me strength to follow and reach my dreams.

Continuing thanks is extended to all the concerned faculty members and friends, who directly and indirectly were involved in the preparation of this thesis.

Contents

Abstract	iii
Acknowledgements	iv
List of Figures	vii
1 Introduction	1
1.1 What is Functional Data Analysis?	1
1.2 Fourier basis functions	3
1.3 Spline basis functions	4
1.4 Number K of basis functions	4
1.5 Roughness Penalty	5
1.6 Curve Registration	8
1.7 Warping function	10
1.8 Landmark Registration	11
1.9 Continuous Registration	12
2 Methods	14
2.1 Data Collection	14
2.2 Transforming the Data into Functions	15
2.3 Registration	20
2.4 Landmark Registration Method	22
2.5 Continuous Registration Method	22
3 Results	24
3.1 Unregistered data	24
3.2 Registered data	31
3.3 Time Warping function and registration	35
4 Conclusion	42
A R Codes	43

References **55**

References **55**

List of Figures

1.1	Three curves vary only in amplitude.	9
1.2	Three curves vary only in phase.	9
2.1	Number of basis functions for the Fourier series	16
2.2	Three curves vary only in Phase.	17
2.3	GCV versus the smoothing parameter λ to penalize the functions obtained by the Fourier series	18
2.4	GCV versus the smoothing parameter λ to penalize the functions obtained by the B-splines	19
2.5	Black curve shows the temperature of the Burlington and red curve shows the temperature of Fairbanks during a year.	20
2.6	Dashed line is the mean function. The black,blue and red line rep- resent the functions for cities of Burlington, Indianapolis and San Francisco respectively.	21
3.1	Fourier series for first decade	25
3.2	Order 6 Splines for first decade	27
3.3	Splines of all cities using deseasoned data	29
3.4	Mean temperature of deseasoned data	30
3.5	Temperature of all unregistered data for the first year	31
3.6	The top panel reproduces the unregistered temperature curves. The landmark registered curves corresponding to those are shown in the bottom panel, where the landmarks were the maximum and minimum temperatures. The dashed line in each panel indicates the mean curve for the curves in that panel.	33
3.7	Comparing the mean function of unregistered and landmark regis- tered data for the city of Indianapolis	34
3.8	Temperature curves and corresponding time warping functions.	35
3.9	Warping function $h(t)$ for all cities.	36
3.10	Continuous registered curves	37
3.11	Continuous registration of three city and corresponding warping functions.	39
3.12	Plots of unregistered, landmark and registered data.	40
3.13	Mean of unregistered and registered data	41

Chapter 1

Introduction

1.1 What is Functional Data Analysis?

In Functional Data Analysis (FDA) we work with a function rather than a single data point. FDA attempts to analyze the information of curves or functions with statistical techniques. It also provides a richer set of analyses than just comparing means (Ramsay & Silverman, 2005). In other words, one advantage of functional data analysis is the possibility of using information on the rate of change or derivatives of the curves (Ramsay, 2013). In the weather data, derivative function illustrates how quickly climate change is accelerating.

The first step in functional data analysis is transforming data points to continuous functions. These functions are represented as:

$$y_{ij} = x_i(t_{ij}) + \epsilon_{ij} \quad i = 1, \dots, n \quad j = 1, \dots, n_i \quad (1.1)$$

where t is time (or another continuous variable), y_{ij} is the j th observation of the i th sample function, and x_i is a smooth function. Also, these smooth functions are defined as linear expansion:

$$x_i(t_{ij}) = \sum_{k=1}^K \phi_k(t_{ij})c_{ik} \quad (1.2)$$

The $\phi_k(t)$ are basis functions that are mathematically independent of each other. The c_{ik} are the coefficients associated with these K basis. Ramsay (2005) uses this linear combination of basis functions as the main method for representing the functions. Ramsay (2005) also determines the coefficients of the expansion c_k by minimizing the least squares criterion:

$$SSE(y|c) = \sum_{i=1}^n \sum_{j=1}^{n_i} [y_{ij} - \sum_{k=1}^K c_k \phi_k(t_{ij})]^2 \quad (1.3)$$

The criterion in matrix form is expressed as:

$$SSE(y|c) = (\mathbf{y} - \mathbf{\Phi}\mathbf{c})'(\mathbf{y} - \mathbf{\Phi}\mathbf{c}) \quad (1.4)$$

Taking the derivative of this criterion with respect to \mathbf{c} and solving the $2\mathbf{\Phi}\mathbf{\Phi}'\mathbf{c} - 2\mathbf{\Phi}'\mathbf{y} = 0$ for \mathbf{c} , provides the least square solution:

$$\hat{\mathbf{c}} = (\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi}'\mathbf{y} \quad (1.5)$$

The vector $\hat{\mathbf{y}}$ of fitted value is:

$$\hat{\mathbf{y}} = \mathbf{\Phi}\hat{\mathbf{c}} \quad (1.6)$$

Moreover, Fourier basis functions and B-Spline basis functions are two types of basis functions that we are use in our analyses.

1.2 Fourier basis functions

Fourier basis is a system that developed in early nineteenth century in France. The set of basis functions for Fourier series includes one constant function and pairs of sine and cosine functions to capture the variation in phase (Ramsay & Silverman, 2005). It is given as follows:

$$\phi_1(t) = 1$$

$$\phi_2(t) = \sin(t\omega)$$

$$\phi_3(t) = \cos(t\omega)$$

$$\phi_k(t) = \sin\left(\frac{k}{2}t\omega\right)$$

$$\phi_{k+1}(t) = \cos\left(\frac{k}{2}t\omega\right)$$

where the constant ω is related to the period T by the relation $\omega = \frac{2\pi}{T}$.

Because of how we define ω , each basis function repeats itself after T time units have passed. Consequently, T is the period of the function and we tend to use this basis for periodic functions.

1.3 Spline basis functions

Splines are piece-wise polynomial functions defined over sub-intervals. First defining a spline is to divide the interval of observation into sub-intervals with boundaries at points called knots (Ramsay, Hooker, & Graves, 2009). Over any sub-interval, the spline function is a polynomial of fixed degree (order), and the key property of spline functions is that they are continuous at the knots. The order of a polynomial is the number of constants required to define it, and it is one more than its degree (the highest power) (Ramsay & Silverman, 2005). For splines, the number of basis functions is:

the number of knots + order of the spline (Ramsay & Silverman, 2005).

Using order six splines guarantees that the first and the second derivatives can be estimated well (Ramsay et al., 2009). The B-spline basis system developed by de Boor (2001) is the most popular spline, and code for working with B-splines is available in R, S-PLUS and MATLAB .

1.4 Number K of basis functions

Determining the number of basis functions is a subtle process. The larger K provides better fit to the data, but risk of introducing noise or variation will increase. On the other hand, smaller K eliminates notable features of the smooth function $x(t)$ (Ramsay & Silverman, 2005). Comparing the number of basis function with the variability of total squared errors $\sum_i^{n_i} s_i^2$ is a common technique that help us to

specify the number of K . The minimum total square of the residuals corresponds to the best K . To have more information in terms of variability we should penalize the residuals by the number of basis n_i (King, 2014). The residual calculation is as follows:

$$s_i^2 = \frac{1}{n_i - K} \sum_{j=1}^{n_i} [y_{ij} - \hat{x}_i(t_{ij})]^2 \quad (1.7)$$

Where K is the number of basis and n_i is the number of observations for the i th curve. To determine an appropriate number of basis we plot $\sum s_i^2$ versus K . Where the plot starts to flatten, adding basis function beyond that does not improve model fit.

1.5 Roughness Penalty

When data are smoothed using either Fourier series or splines, it is very crucial to avoid both over-smoothing, thereby missing important features of the curve, and under-smoothing, thereby introducing noise. Fitting basis expansions by least square criterion (1.3) does not provide a precise estimation of the curves (Ramsay & Silverman, 2005).

Roughness penalty introduces a powerful option for approximating discrete data by a function. Additionally, roughness penalty approaches can be applied to a much wider range of smoothing problems than simply estimating a curve x from observations of $x(t_j)$ for certain points t_j (Green & Silverman, 1994).

First, we define a measure of the roughness of the fitted curve, and then minimize a fitting criterion that trades off curve roughness against lack of data fit. The square of the second derivative $[D^2x(t)]^2$ of a function x at argument value t is often called its curvature at t (Ramsay et al., 2009). Penalty term is a measure of a functions roughness that is the integrated squared second derivative or total curvature:

$$PEN_2(x) = \int [D^2x(t)]^2 dt. \quad (1.8)$$

Whenever the function is highly variable, the curvature is large and the penalty is large too. We can apply this concept to derivative estimation, as well, by penalizing the curvature of the second derivative.

For periodic functions such as mean temperature curves, suppose that the functions are represented by the first three terms in the Fourier series for some known period $\omega = \frac{2\pi}{T}$ as follows:

$$x(t) = c_0 + a_1 \sin \omega t + b_1 \cos \omega t$$

Ramsay (2005) defines the differential operator $L = \omega^2 D + D^3$ as the harmonic acceleration operator. If we apply this harmonic acceleration operator to higher-order terms in a Fourier series:

$$L[a_j \sin j\omega t + b_j \cos j\omega t] = \omega^2 j(1 - j^2)[a_j \cos j\omega t - b_j \sin j\omega t]. \quad (1.9)$$

For $j = 1$ it is zero and it increases with the cube of j . The integral of the square

of this harmonic acceleration operator may be a suitable measure of roughness for periodic data like the temperatures curves (Ramsay et al., 2009):

$$PEN_L(x) = \int [Lx(s)]^2 ds. \quad (1.10)$$

Because this expression is proportional to $j^2(1-j^2)^2$, the term with $j = 1$ does not get penalized at all, but higher-order terms in the Fourier approximation receive substantially higher penalties.

Regardless of what roughness penalty was used, we add some multiple of it to the error sum of squares to define the compound fitting criterion.

$$PenSSE = \sum_{j=1}^{n_i} (y_{ij} - x_i(t_{ij}))^2 + \lambda \int [D^2x(t)]^2 dt. \quad (1.11)$$

The parameter λ is a smoothing parameter. The first term is residual sum of squares that shows the rate of exchange between fit to the data. The second term is penalizing curvature that shows the variability of function x . The smoothing parameter λ measures the rate of exchange between first term and second term. As $\lambda \uparrow \infty$, curvature becomes increasingly penalized and $x(t)$ becomes smooth. When $\lambda \downarrow 0$ the function x free to fit the data as closely as possible (Ramsay et al., 2009).

Determining the roughness penalty is not an exact science and should be evaluated in more than one way to make sure results are consistent.

To determine the best value for smoothing parameter λ we use the generalized

cross-validation measure GCV developed by Craven and Wahba (1979).

$$GCV(\lambda) = \left(\frac{n}{n - df(\lambda)}\right)\left(\frac{SSE}{n - df(\lambda)}\right) \quad (1.12)$$

where $df()$ measures the effective number of parameters used to estimate $x(t)$ (Ramsay & Silverman, 2007) plotting the function $GCV(\lambda)$ in any case will inform us about the curvature of near its minimum (Ramsay et al., 2009).

1.6 Curve Registration

After transforming our observations to functional forms, we want to analyze the standard summary measures that are used in functional data analysis, including the mean and co-variance functions. We are not quite ready, however, there are systematic variations such as amplitude and phase variation among functional observations. Therefore, the direct averaging to produce a cross-sectional mean function over curves that are not aligned will produce poor estimators of the true process average and co-variance functions. Registration of the functional data, involving transformations of the argument t rather than the values $x(t)$, will help us to optimize the estimations (Ramsay & Silverman, 2005).

There are two types of variation between function values $x_i(t_i)$. One is amplitude variation or vertical variation, where $x_1(t)$ and $x_2(t)$ differ at points of time t but exhibit the same shape features at that time. Figure 1.1 shows this amplitude variation:

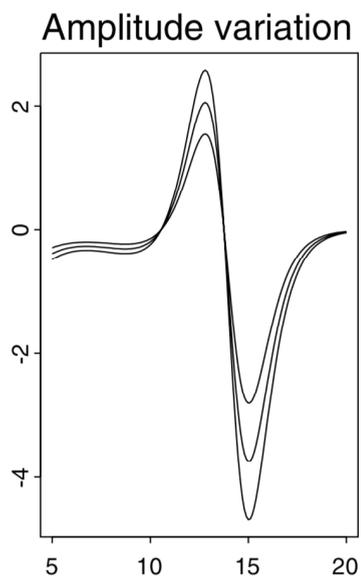


FIGURE 1.1: Three curves vary only in amplitude.

The other one is phase variation where x_1 and x_2 are not exhibiting the same behavior at the same time t . Therefore, to compare these two functions we need to transform the time scale horizontally as you can see in Figure 1.2:

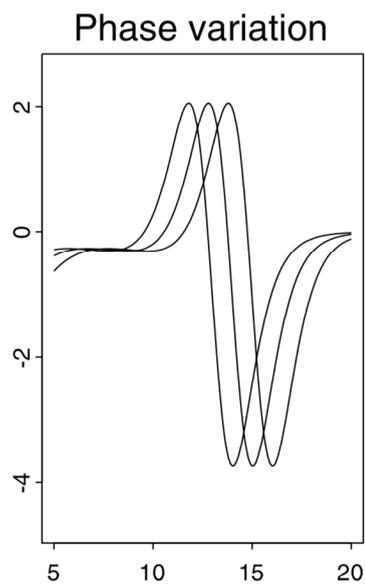


FIGURE 1.2: Three curves vary only in phase.

There are several elements that affect the weather temperature of the cities. Some of these factors are altitude or height above sea level, distance from the sea and ocean currents. Consequently, winter comes earlier in some cities than others or arrives late in some years. In temperature data case the shift is an important feature of each curve (Ramsay & Silverman, 2005).

Without thinking about the phase variation, the cross-sectional mean is at risk of underestimating the local maxima and overestimating the local minima (Zhong, 2008). The sample cross-sectional mean will be more successful in estimation of average maximum temperature or minimum temperature, when the certain features of curves are aligned to the same time.

1.7 Warping function

A warping function works as a time transformation function. Composition of the smooth function and the warping function gives a registered curve (Zhong, 2008). Let N functions x_i be defined on closed real intervals $[0, T_i]$. The upper boundaries may either vary randomly or be fixed. Also, let $h_i(t)$ be a transformation of time t for case i . Since timing of events remains in the same order and $h_i(t)$ should not affect that, the time warping function should be strictly increasing, *i.e.* $h_i(t_1) > h_i(t_2)$ for $t_1 > t_2$ (Ramsay & Silverman, 2007). Hence, transformation is one to one and warping function is invertible. The inverse of a warping function transforms observation time to a synchronized time scale where certain features on different curves occur simultaneously (Zhong, 2008).

The registered functions are $x_i^*(t) = x_i[h^{-1}(t)]$, where the aligning function $h(t)$ satisfies the following equation:

$$h^{-1}[h(t)] = t \tag{1.13}$$

1.8 Landmark Registration

One procedure to align the curves is landmark registration that makes use of warping function. A landmark is a certain feature of a curve with location such as minimum, maximum or crossing of zero.

We are going to align the curves by transforming t for each curve, so that the features of interest occur at the same transformed times.

For each curve x_i we identify the argument values t_{if} , $f = 1, \dots, F$ related to each of F features. The goal is to find a set of warping functions that satisfy:

$$h_i(0) = 0, \quad h_i(T_0) = T_i$$

where T_0 is the synchronized ending time and T_i is the observed ending time for sample path i and $h_i(t_{0f}) = t_{if}$, $f = 1, \dots, F$, where t_{0f} is timing of landmarks for mean function.

The landmarks of each converted curves occur at the same time as in the mean function (Ramsay & Silverman, 2005). Now that a warping function h has been

estimated from landmark registration, we can calculate the registered function:

$$x_i^*(t) = x_i[h^{-1}(t)]. \quad (1.14)$$

1.9 Continuous Registration

One problem of using landmark registration is that occasionally landmarks are not visible in all curves, and sometimes, its not easy to precisely find the location of the landmarks. Identifying their timing may involve tedious interactive graphical procedures. Thus, we might prefer a fully automatic method (Ramsay et al., 2009). Moreover, using just a few landmarks could devaluate other aspects of the curves and leave them unregistered. Ramsay(1998) proposed continuous monotone registration method where a warping function is selected from a smooth monotone family of functions.

The idea behind this method is that if an arbitrary sample registered curve $x[h(t)]$ and target curve $x_0(t)$ differ only in terms of amplitude variation, *i.e.* $x_0(t) = Cx[h(t)]$, C is a constant, then their values are proportional to each other. If we plot the function values $x_0(t)$ and $x[h(t)]$ against each other, we will see a straight line passes through the origin. Let the n by two matrix X contain pairs of $(x_0(t), x[h(t)])$ where both the target function $x_0(t)$ and the registered function $x[h(t)]$ are evaluated at a fine mesh of n values of t . Now we are going to analyze the two-by-two cross-product matrix $X'X$ by principle components. The functional

analogue of the cross-product matrix $X'X$ is an order two matrix :

$$T(h) = \begin{bmatrix} \int \{x_0(t)\}^2 dt & \int x_0(t)x[h(t)]dt \\ \int x_0(t)x[h(t)]dt & \int \{x[h(t)]\}^2 dt \end{bmatrix}$$

The principle components analysis of $T(h)$ of integrated products of target curve and registered curves should essentially reveal one component, and the smallest eigenvalue should be near zero (Ramsay et al., 2009). Ramsay defines the minimum eigenvalue criterion as follows:

$$MINEIG(h) = \mu_2[T(h)], \quad (1.15)$$

where the function μ_2 is the size of the smallest eigenvalue of an order two symmetric matrix. When $MINEIG(h) = 0$, h is the warping function for registration.

Chapter 2

Methods

2.1 Data Collection

In this study we are going to work on the daily temperature of sixteen cities of the United State between 1950 and 2013. (Atlanta, GA, Boston, MA ,Burlington, VT, Los Angeles, CA, Portland, OR, Miami, FL, Salt Lake City, UT, Nashville, TN, New York City, NY, San Antonio, TX, Indianapolis, IN, Minneapolis, MN, Green Bay, WI, Missoula, MT, Fairbanks, AK, San Francisco, CA).

The data was collected from *NOAA's* National Climatic Data Center. This data center provided the minimum and maximum temperatures of different weather stations and we used the average of minimum and maximum as y_{ij} for the analyses. Totally there are 23376 time points (days) for each city (n_i are equal). We have also used the package `fda` in R that created by J. O. Ramsay for the most part of our analyses.

2.2 Transforming the Data into Functions

To have a precise study of changes between temperature of cities, seasonal and random variations need to be removed. We can subtract the average median temperature from daily temperature to seasonal adjustment (King, 2014).

Using functional data techniques such as, fitting the Fourier series to the full data, is an appropriate method (Ramsay & Silverman, 2005) (Ramsay & Silverman, 2007).

The Fourier series with a period of 365.25 days (there are 16 leap years in the data) and five basis functions were used in this thesis. The residuals of the predicted and the observed data points were minimized to acquire a suitable and acceptable number of basis functions.

Plot of the number of basis functions versus the residuals in Figure 2.1 indicates that the biggest decline occurs when number of basis functions K is five and the sum of the residuals is 962.85. Far more than the range of K in the Figure 2.1, the Fourier series with 131 basis functions has residuals that sum up to 950.26. This shows that adding more basis functions does not considerably decreases the sum of the residuals and it is not worth the cost of complicating the model.

K versus total of squared residuals

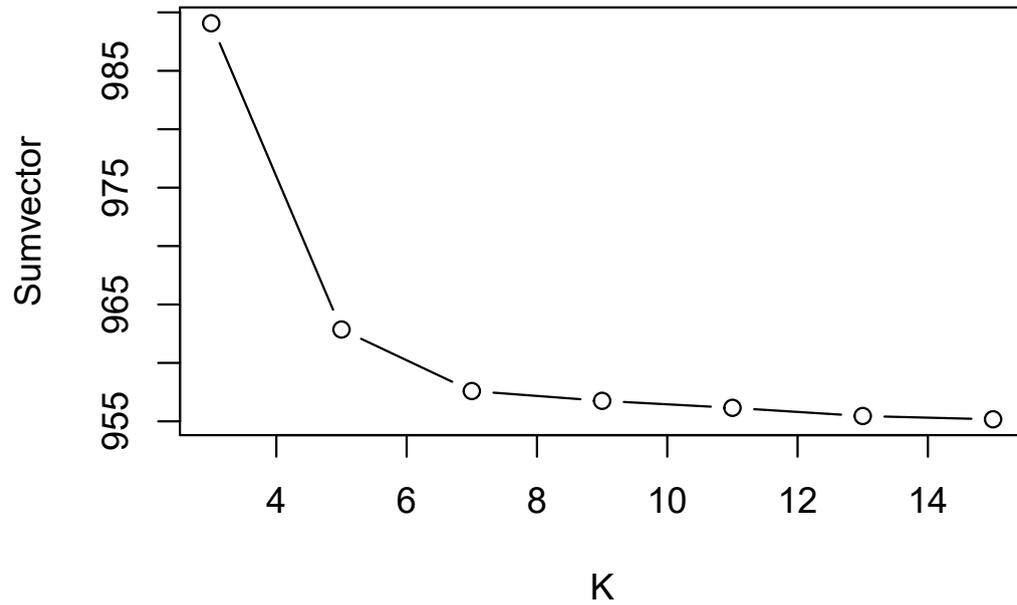


FIGURE 2.1: Number of basis functions for the Fourier series

In this study we also used B-spline basis system to fit the data. An order of six spline was fit to the original data with 133 basis functions. The calculation to determine the acceptable number of basis functions is similar to the calculation used for the Fourier series.

Figure 2.2 shows that the sum of the residuals decreases significantly with 133 basis functions.

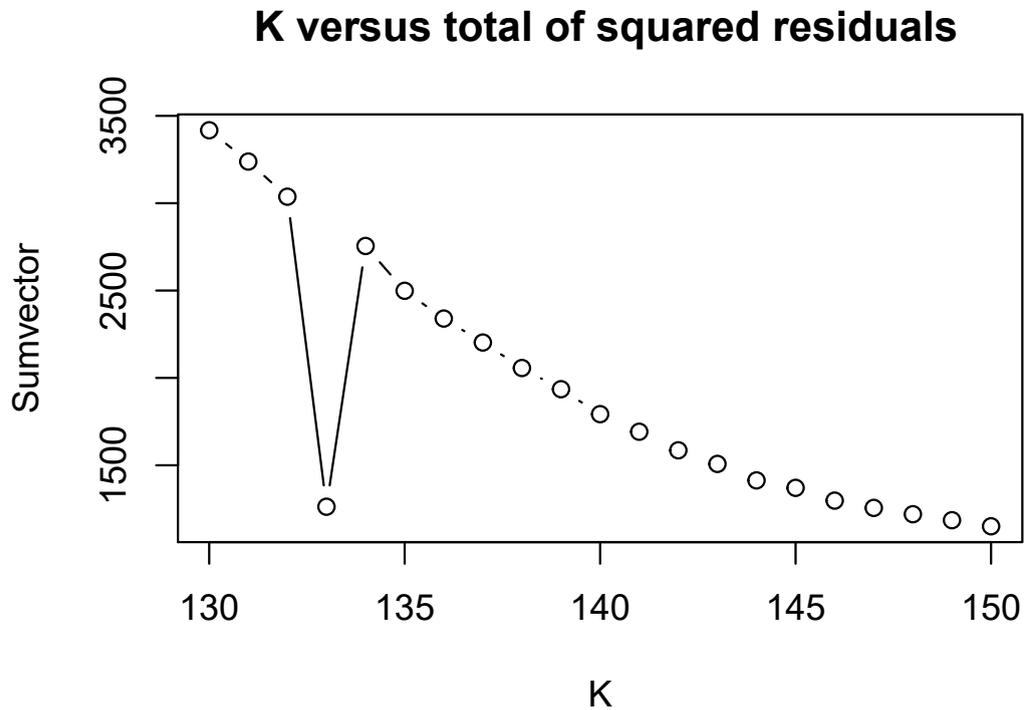


FIGURE 2.2: Three curves vary only in Phase.

One possible explanation for this drop at 133 basis functions is that this number is twice the number of years (64 years) plus the degree of the spline (King, 2014). It can be assumed that each year has two knots considering maximum and minimum temperature. In the next chapter we will compare both Fourier series and B-spline transformation.

Now its time to smooth these functional data with the roughness penalty.

For the functions obtained by the Fourier series we smoothed the data using a harmonic acceleration roughness penalty, equation (1.10), which was defined in chapter one. We also determined a range of smoothing parameter λ values and examined the degrees of freedom and values of the generalized cross-validation

coefficient GCV associated with each value of λ . Figure 2.3 shows the plot of the $\log_{10} GCV$ versus the $\log_{10} \lambda$ since the only interest is the scale of the number of digits.

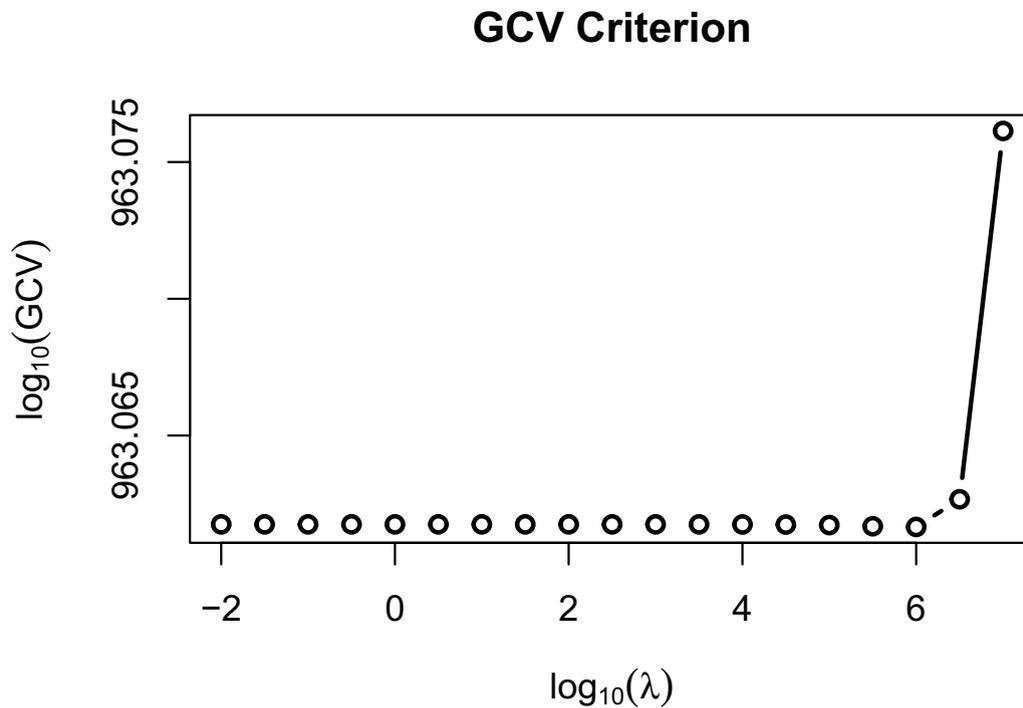


FIGURE 2.3: GCV versus the smoothing parameter λ to penalize the functions obtained by the Fourier series

According to this figure, GCV value is minimized at $\log_{10} \lambda = 6$. This was the smoothing parameter used to minimizing the penalized least squares criterion, equation (1.11), and smoothing the curves.

The process to find smoothing parameter λ for the data fitted with B-spline basis is same as the Fourier series. However, in this case, the roughness penalty is based

on the equation (1.8). Minimized GCV happened at $\log_{10} \lambda = 10$. It can be found in Figure 2.4:

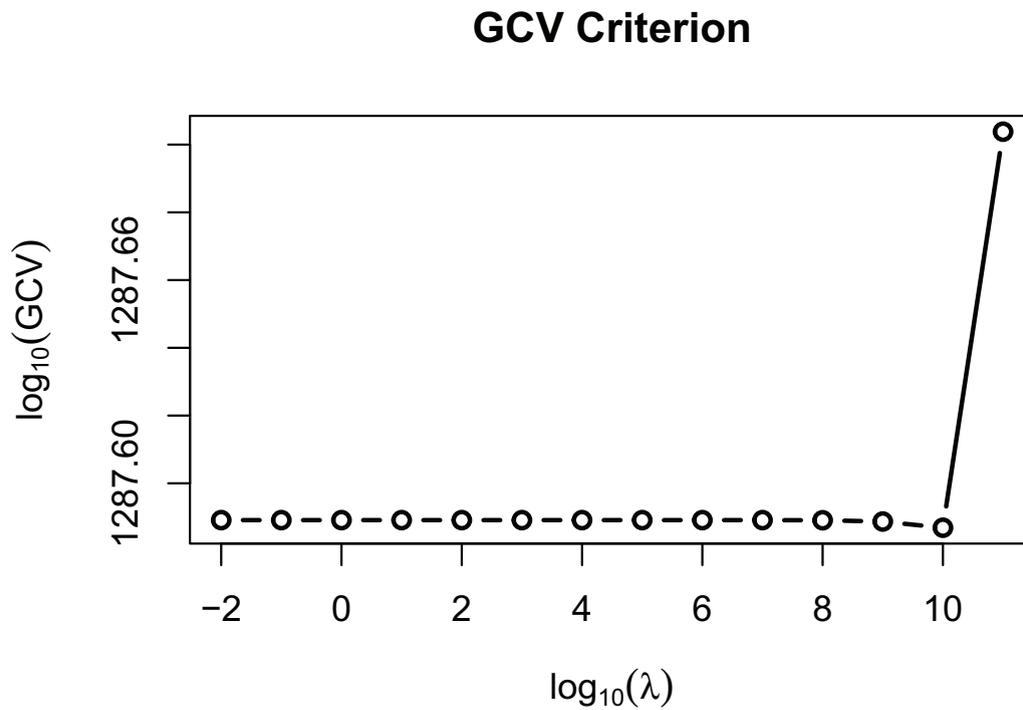


FIGURE 2.4: GCV versus the smoothing parameter λ to penalize the functions obtained by the B-splines

After smoothing the functional data and evaluating both the Fourier series and spline functions at every time point, the seasonal fits, which were acquired by the Fourier basis, were subtracted of spline fits to obtain deseasonalized data. An order six spline with ten basis function applied to smooth the residuals.

2.3 Registration

The comparison of the curves obtained in the preceding chapters shows that the warmest or coldest temperatures of cities occur in different times. Figure 2.5 shows the maximum and minimum temperature of the two cities within a year.

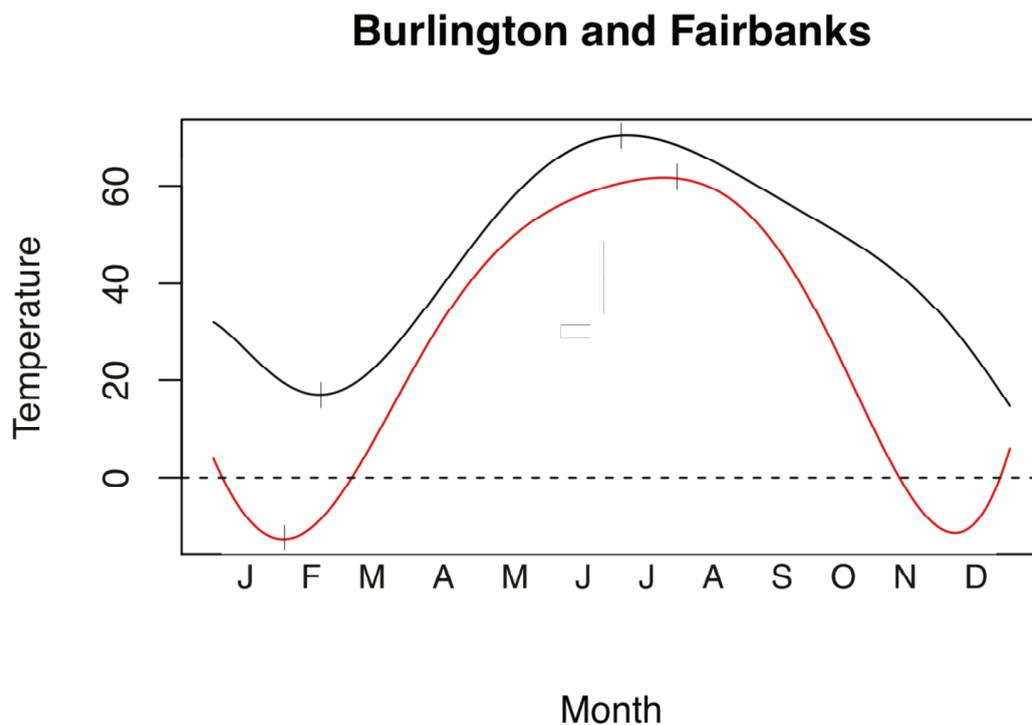


FIGURE 2.5: Black curve shows the temperature of the Burlington and red curve shows the temperature of Fairbanks during a year.

Burlington reaches the highest temperature of the year earlier than Fairbanks, but winter comes later in Burlington. This means that, based on differences in geographical areas, the arrival time of winter or summer is different. Winter comes earlier in some cities than others. Consequently, the cross sectional mean function or calculated average temperature does not resemble any of the observed curves,

and it also does not provide an accurate measure of the average temperature.

Figure 2.6 shows the mean function curve (dashed line) and unregistered functions of three cities of Burlington, Indianapolis and San Francisco. The average of these temperature curves displays an image that is different from any of the individual curve.

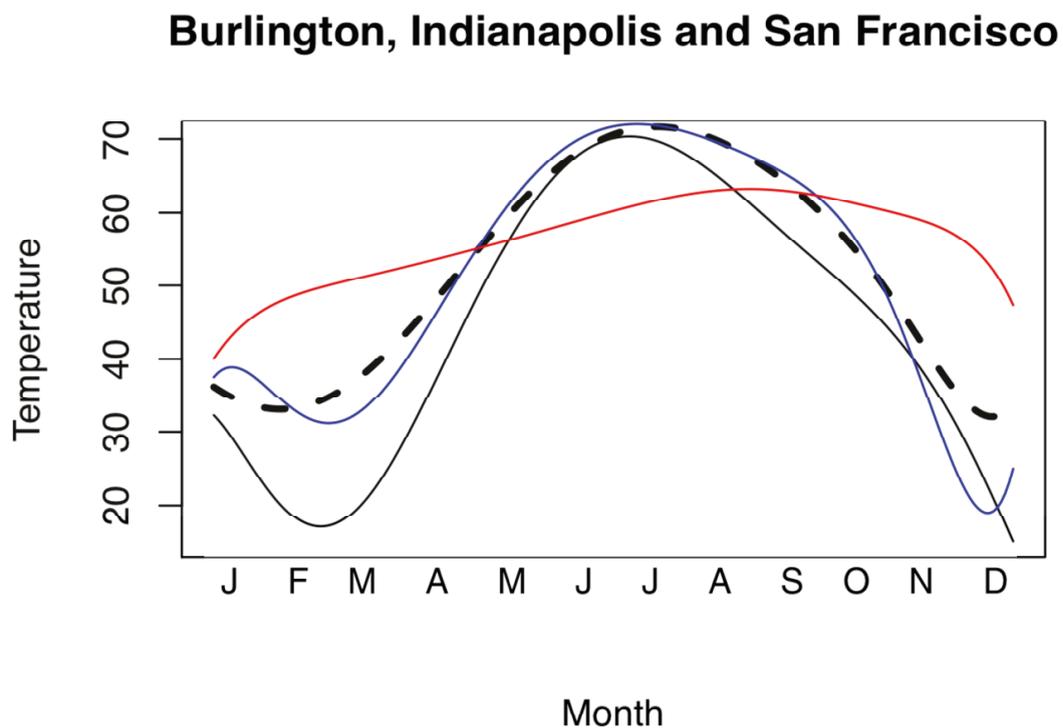


FIGURE 2.6: Dashed line is the mean function. The black, blue and red line represent the functions for cities of Burlington, Indianapolis and San Francisco respectively.

We used both landmark registration and continuous registration method to align the curves and solve this problem.

2.4 Landmark Registration Method

We found the maximum temperature of each city during a year as a landmark, and the mean of these values was calculated. The goal is to align the landmarks with their mean. This process happened similarly for the minimum temperatures to not leave these aspects unregistered.

To identify these landmarks we used interactive graphical procedures provided by locator function in R program. In the landmark method, t is transformed so that the landmarks for each curve closely approach the landmarks of the mean function. That means, all of the maximum and minimum temperatures respectively got close to maximum and minimum of mean function. In this case, we tried to show that how warm the average summer is at the time the average temperature reaches its peak, rather than at any fixed time.

We will discuss the result of this method in chapter three.

2.5 Continuous Registration Method

The timings of a fixed set of landmarks provide one way of describing how similar the shapes of two curves are. Continuous registration method is an automatic method that use the entire curves for registration (Ramsay & Silverman, 2007). The function of `register.fd` (in R program) was used to improve the temperature curves that have already been registered using function of `landmarkreg`.

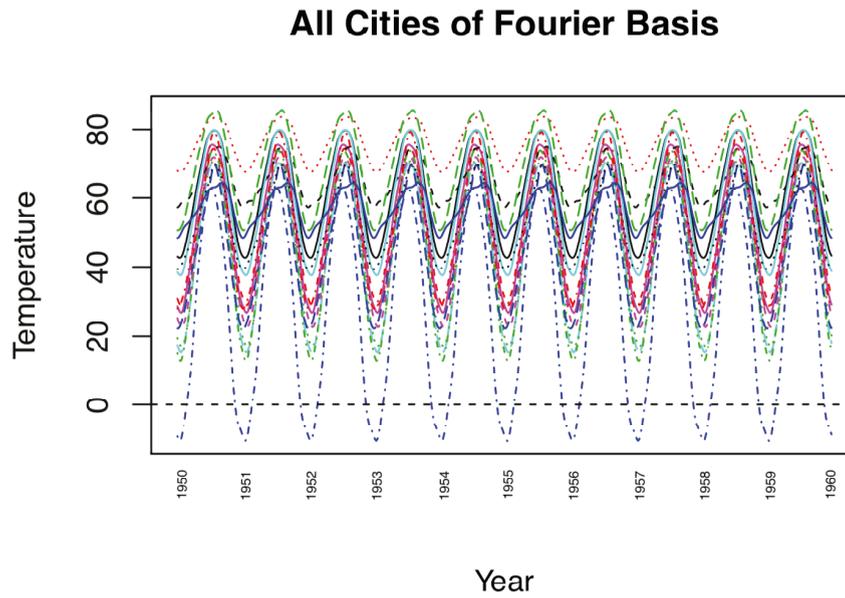
A basis function is combined with a roughness penalty for defining the warping function. Also, there is a target function so that important features such as maximum and minimum occur at about the same argument values of that target function. In this study we used the mean of registered curves obtained by landmark registration as the target function. The continuous registration process requires iterative numerical optimization techniques (Ramsay et al., 2009). To optimize the registration process and have a better estimation, we used the calculated mean obtained by this method as a new target function and again started to register the curves.

Chapter 3

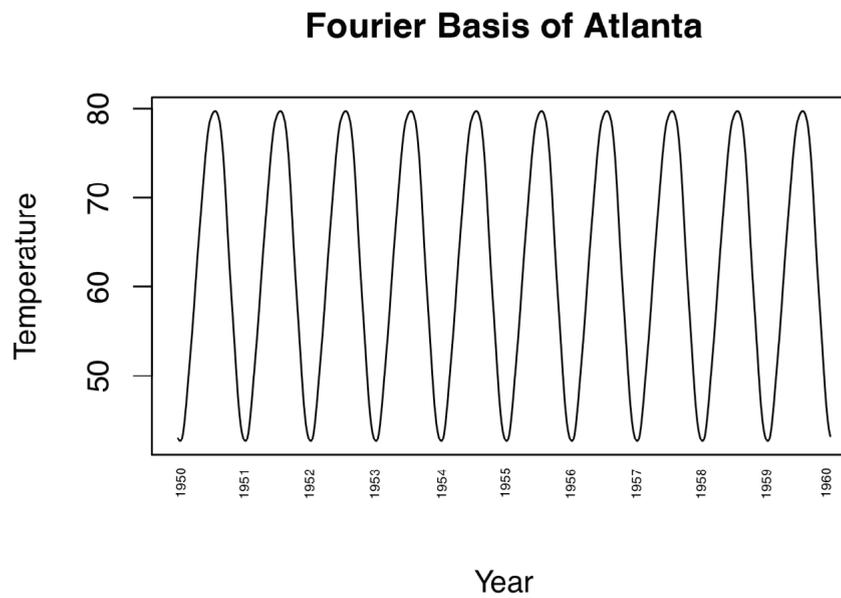
Results

3.1 Unregistered data

The Fourier series which consist of five basic functions were used to model the seasonal variation. Figure 3.1(a) shows the Fourier fits for all cities only for the first decade, and figure 3.1(b) shows the Fourier series for only the city of Atlanta.



(a) Fourier series using five basis functions. Only first decade is shown and each color represent one city



(b) Fourier series of Atlanta using five basis functions

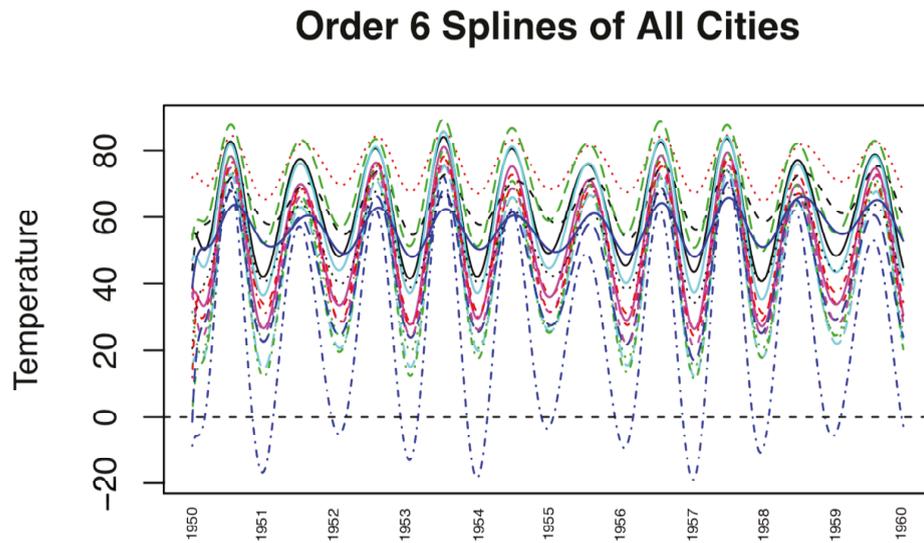
FIGURE 3.1: Fourier series for first decade

Ramsay declares that the Fourier series are useful for the functions, where the curvature tends to be of the same order everywhere. Ideally, the periodicity of the Fourier series should be reflected to some degree in the temperature data.

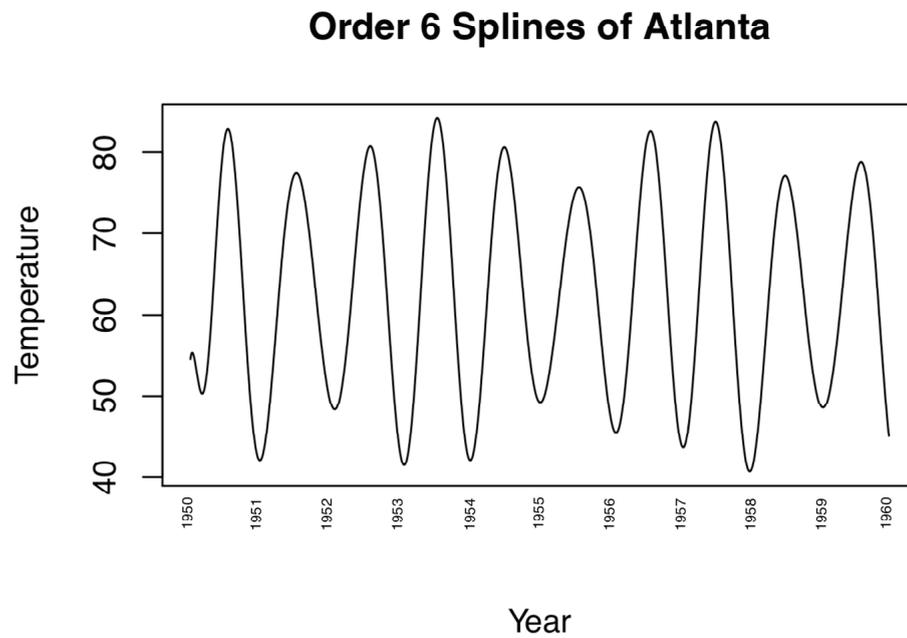
From these graphs, it is easily observed that the functions are purely periodic. Specific features such as maximum and minimum values do not vary in its period from year to year.

In accordance with previous criteria, 133 basis functions were used with order 6 splines on the original data. Figure 3.2(a) illustrates the graph of the first decade of these splines. Seasonality of the curves is still visible, but unlike the Fourier series the seasonality varies from year to year.

The maximum and minimum values of the functions change in each different year. It is shown in Figure 3.2(b) for city of Alberta.



(a) B-spline using 133 basis functions. Only first decade is shown and each color represent one city

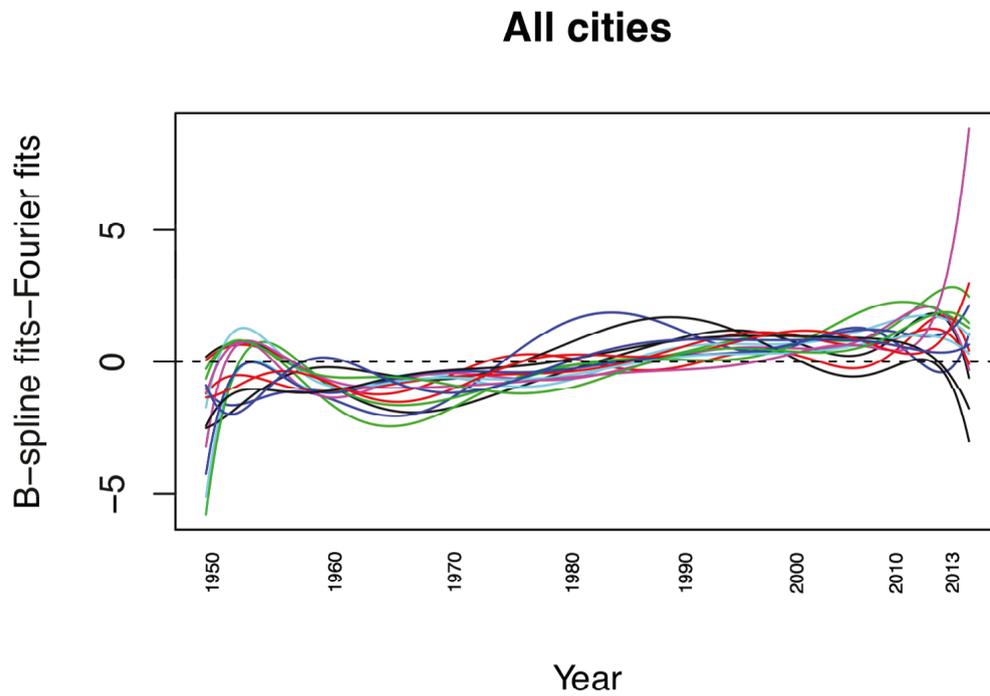


(b) B-spline of Atlanta using 133 basis functions

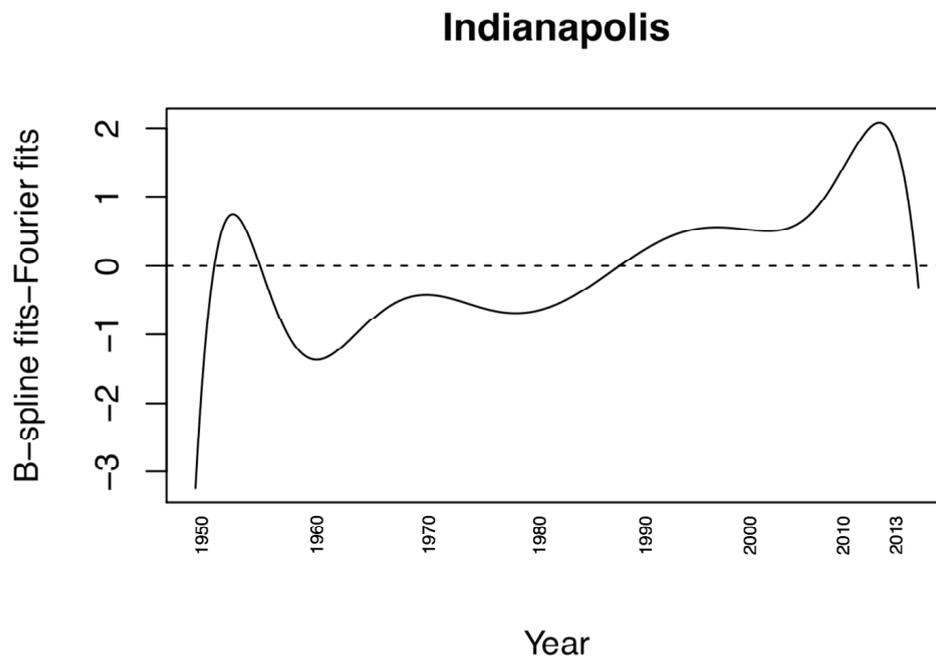
FIGURE 3.2: Order 6 Splines for first decade

After presenting the Fourier and B-spline fits, the differences of these two functional objects were calculated in order to obtain the data without seasonal variations (King, 2014). For each city a spline was fit to the differences of the Fourier series and the spline of the original data. The total number of 10 basis functions were used to have a fairly smooth trend.

These functions also were penalized using $\log \lambda = 10$ as the smoothing parameter. Figure 3.3(a) shows the splines of all the cities with the seasonal variation removed, while the Figure 3.3(b) shows only the city of Burlington, VT. These graphs mainly show an upward trend in the temperature through the entire time frame. Accordingly, mean temperature of the cities is increasing over the time.



(a) Order 6 splines of the data of all cities with seasonal variation removed.



(b) Order 6 splines of the Indianapolis with seasonal variation removed.

FIGURE 3.3: Splines of all cities using deseasoned data

The major variations of the temperature during 1950s could just be due to measurement error since recording temperatures were not regulated until later (Karl, Melillo, Peterson, et al., 2009).

Also, there is another interpretation about the high variations of 1950s. Since spline functions are less accurate around the endpoints, the difference between these two functional objects (Fourier and Spline) are significant (King, 2014).

We also found the average temperature of all of these cities. The functional data objects were averaged together using the sum of the functions and dividing by 16 (number of functions contributing to the mean). Figure 3.4 indicates how mean temperature is changing between 1950 and 2013. The mean function is not a straight line and it is possible to see that the warming is not linear.

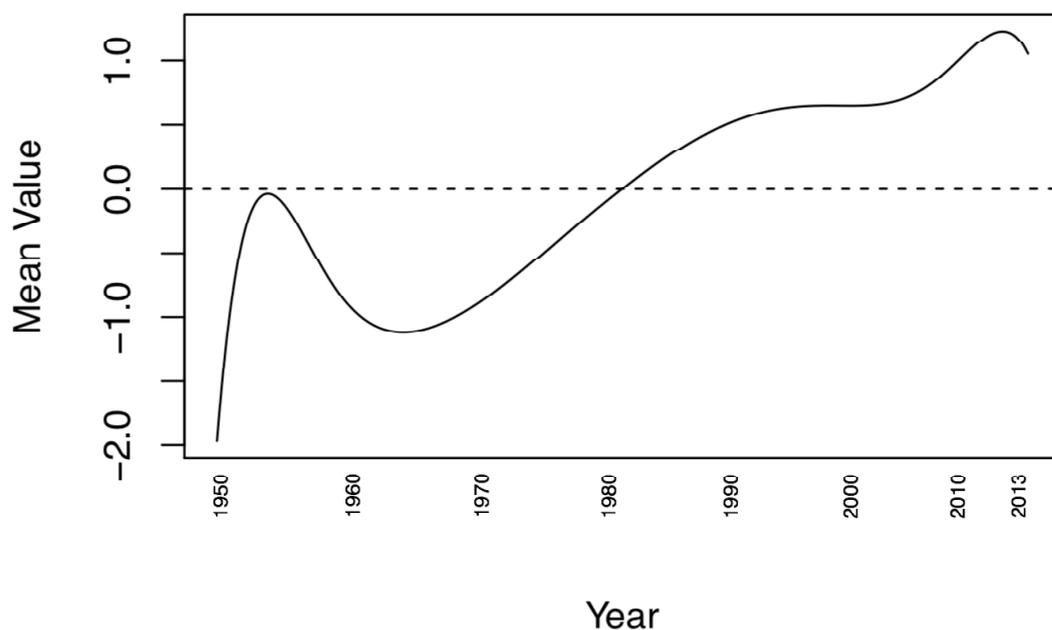


FIGURE 3.4: Mean temperature of deseasoned data

3.2 Registered data

Looking at the temperatures of the cities we can see that timing of the warmest or coldest day of a year is variable.

Figure 3.5 shows the unregistered curves during the first year and the dashed line indicates the mean of these sixteen curves.

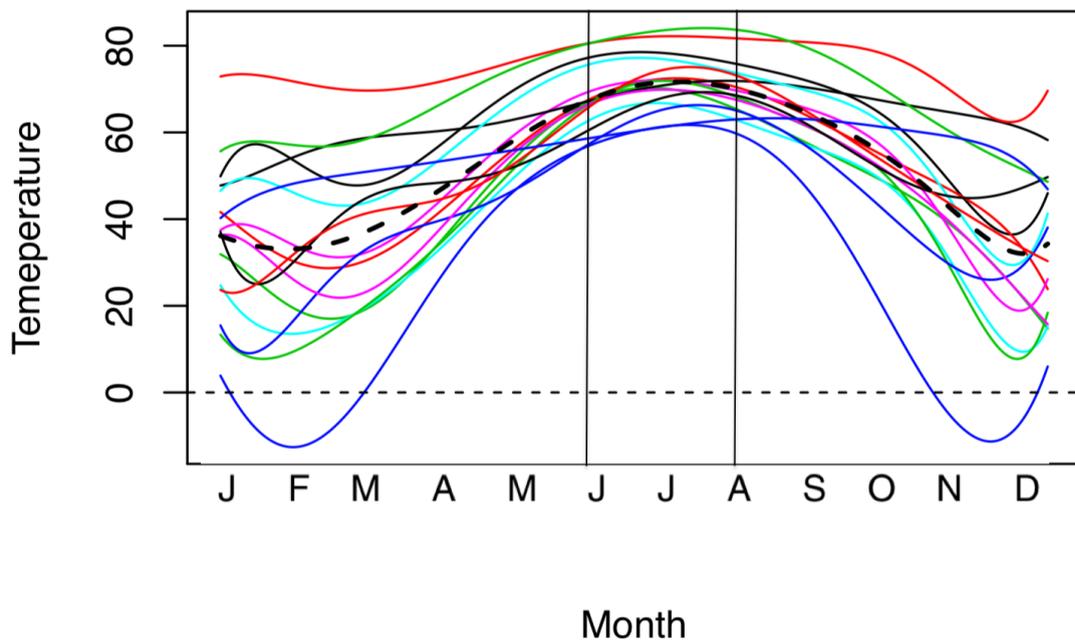


FIGURE 3.5: Temperature of all unregistered data for the first year

We can see that peak of temperature of the mean curve occurs around the last week of July. Using the land mark registration method, the time of maximum temperature of each city was specified by R programming locator code. This was a manual maximum temperature identification procedure requiring a mouse click at the point where the temperature curve reaches the peak.

We used a single landmark t_i being the day for each city (i) at which its temperature is maximum. Mean of these t_i values is 204.85 which is on 24th of July. Some cities experience their warmest day of the year before July 24th and some cities after that. This analysis should make it fairly easy to understand why the average of these temperature curves displays an image that is different from any of individual curves. Let us define $t_0 = 204.85$ as a time specified for the middle of the average temperature. Then we specify time-warping functions h_i by fitting a smooth function to the three points $(1, 1)$, (t_0, t_i) , and $(365, 365)$. Because in this method we just found maximum temperatures as landmarks, we left the minimum temperatures unregistered. To solve this, we found new landmark t_i (the day for each city (i) at which its temperature is minimum) and did the same procedure to find the average of the minimum temperature. The mean value of these new landmarks is 32.46 which is on first of February.

Unregistered data was registered using these specific landmarks. The bottom panel in Figure 3.6 uses landmark registration to align these curves, so the maximum and minimum temperature for all cities happens fairly at the same time.

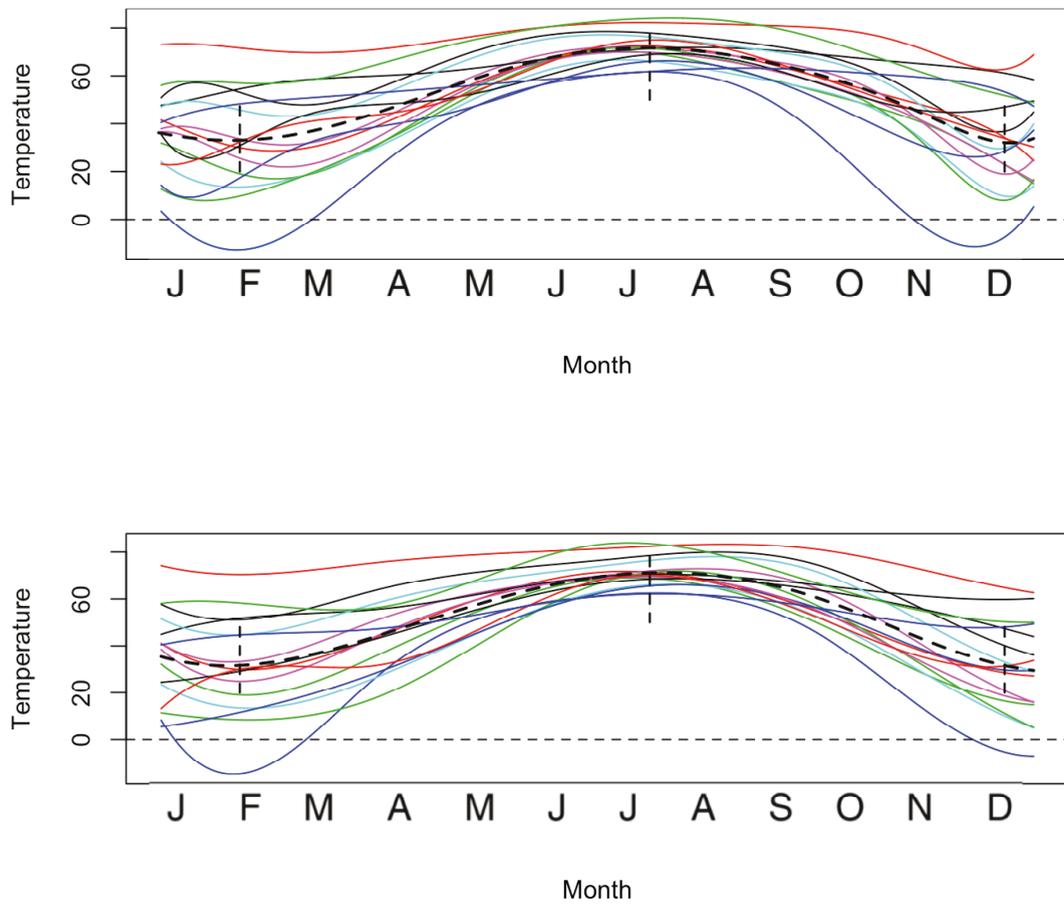


FIGURE 3.6: The top panel reproduces the unregistered temperature curves. The landmark registered curves corresponding to those are shown in the bottom panel, where the landmarks were the maximum and minimum temperatures. The dashed line in each panel indicates the mean curve for the curves in that panel.

Comparison of mean of unregistered data and registered data indicates that mean of registered data tends to resemble much more closely in most of the sample curves. In Figure 3.7, mean of unregistered and registered data for city of Indianapolis is shown. The mean function of registered data gives a better display of observed data specifically at the beginning and end of the year.

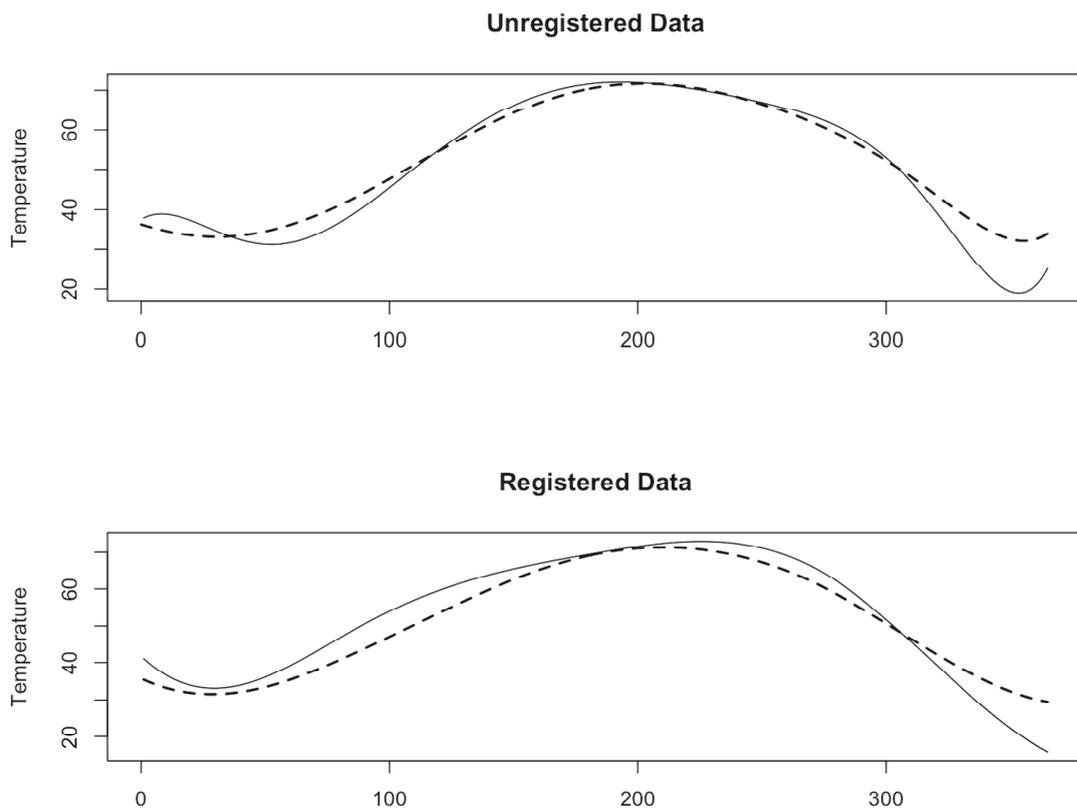


FIGURE 3.7: Comparing the mean function of unregistered and landmark registered data for the city of Indianapolis

3.3 Time Warping function and registration

Figure 3.8 shows the temperature curves on the left and the corresponding warping function $h(t)$ on the right for the cities of San Francisco and Burlington. The vertical dashed line is the mean of warmest day of the year calculated in the previous section.

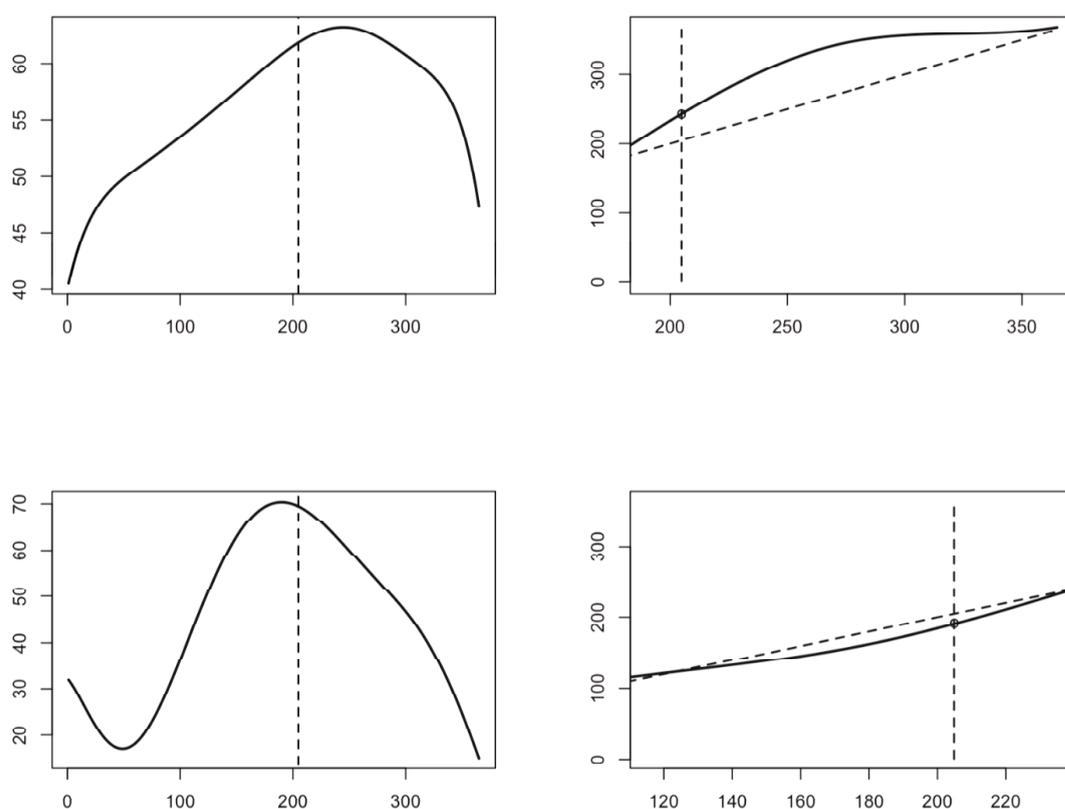


FIGURE 3.8: Temperature curves and corresponding time warping functions.

If, at any time t , $h_i(t) > t$, then warped time is growing faster than clock time and our observed process is running late (Ramsay & Silverman, 2005). This means that the temperature reaches the peak later than average at that clock time.

Similarly, if $h(t) < t$, clock time is being slowed down for a process that is

running ahead of some target. In this case, the temperature reaches the peak earlier compared to the average. According to Figure 3.8 summer comes earlier in city Burlington compared to city San Francisco. Figure 3.9 also illustrates the time warping function $h(t)$ for all sixteen cities.

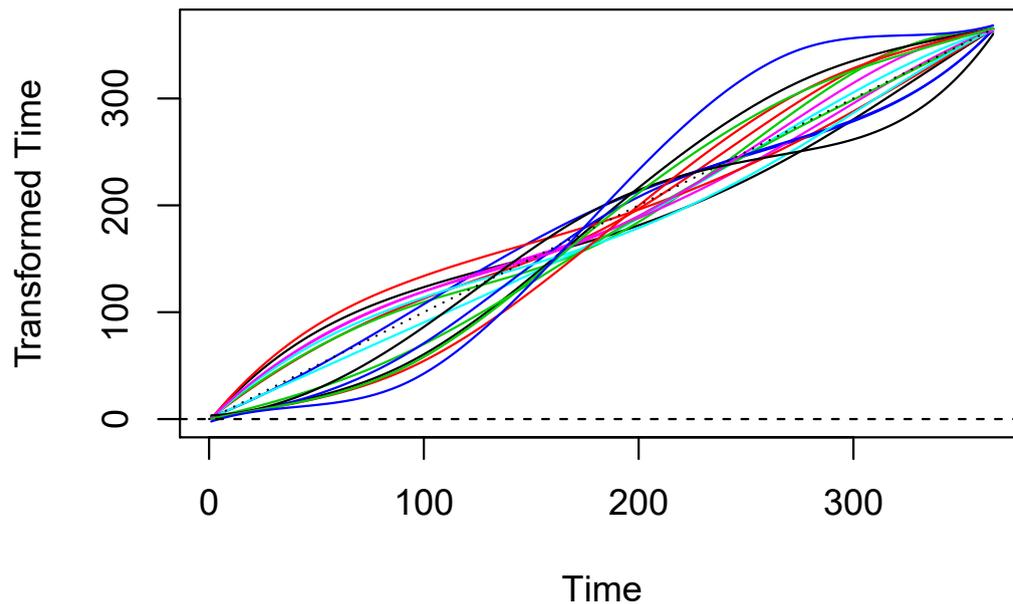
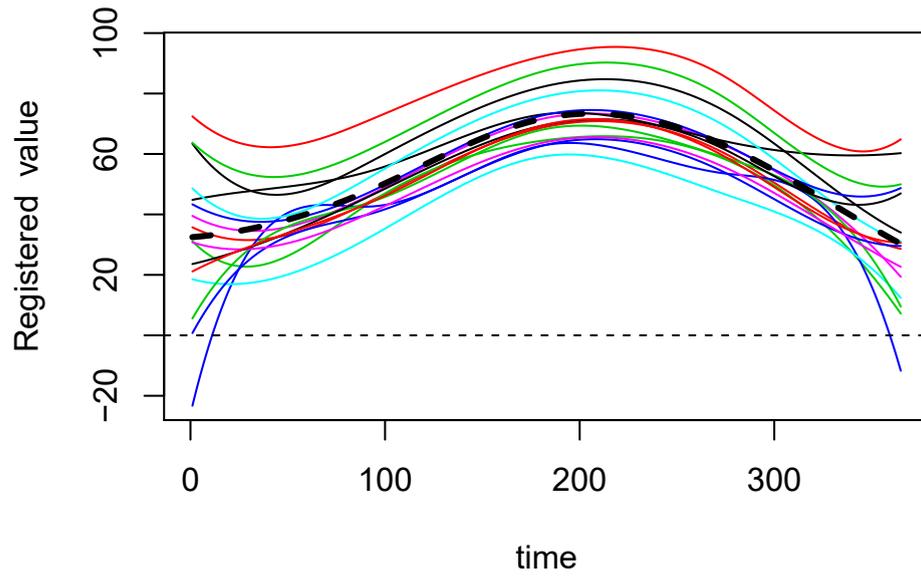
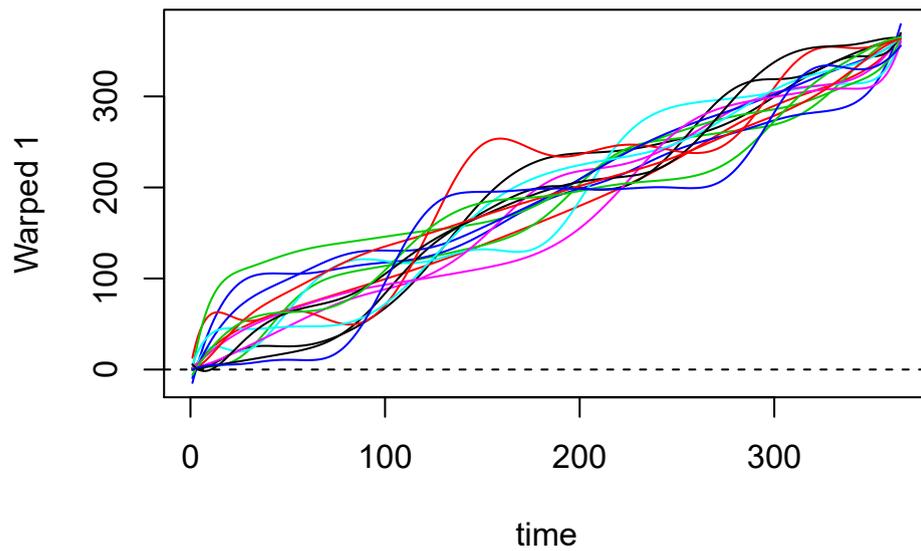


FIGURE 3.9: Warping function $h(t)$ for all cities.

For the next step we used continuous registration method to register the temperature curves. Figure 3.10(a) shows that the continuously registered temperature curves are now aligned over the entire maximum and minimum values relative to the landmark registered curves. Figure 3.10(b) also shows the time warping functions $h(t)$ of all cities.



(a) Continuous registered curves. Dashed line is mean function.



(b) Warping function $h(t)$ for all cities.

FIGURE 3.10: Continuous registered curves

The calculated mean obtained by this method was used as a new target function and the process of continuous registration was operated again. However, the result of this iteration is not significantly different from previous registered curves. For each curve we also found a series of plots, each containing two side-by-side panels. The left panel contains the unregistered curve (dashed blue line), the target function (dashed red line) and the registered curve (solid blue line). The right panel includes the warping function $h(t)$ and the linear function corresponding to no warping. Fig 3.11 shows these plots for Burlington, Indianapolis, and San Francisco cities.

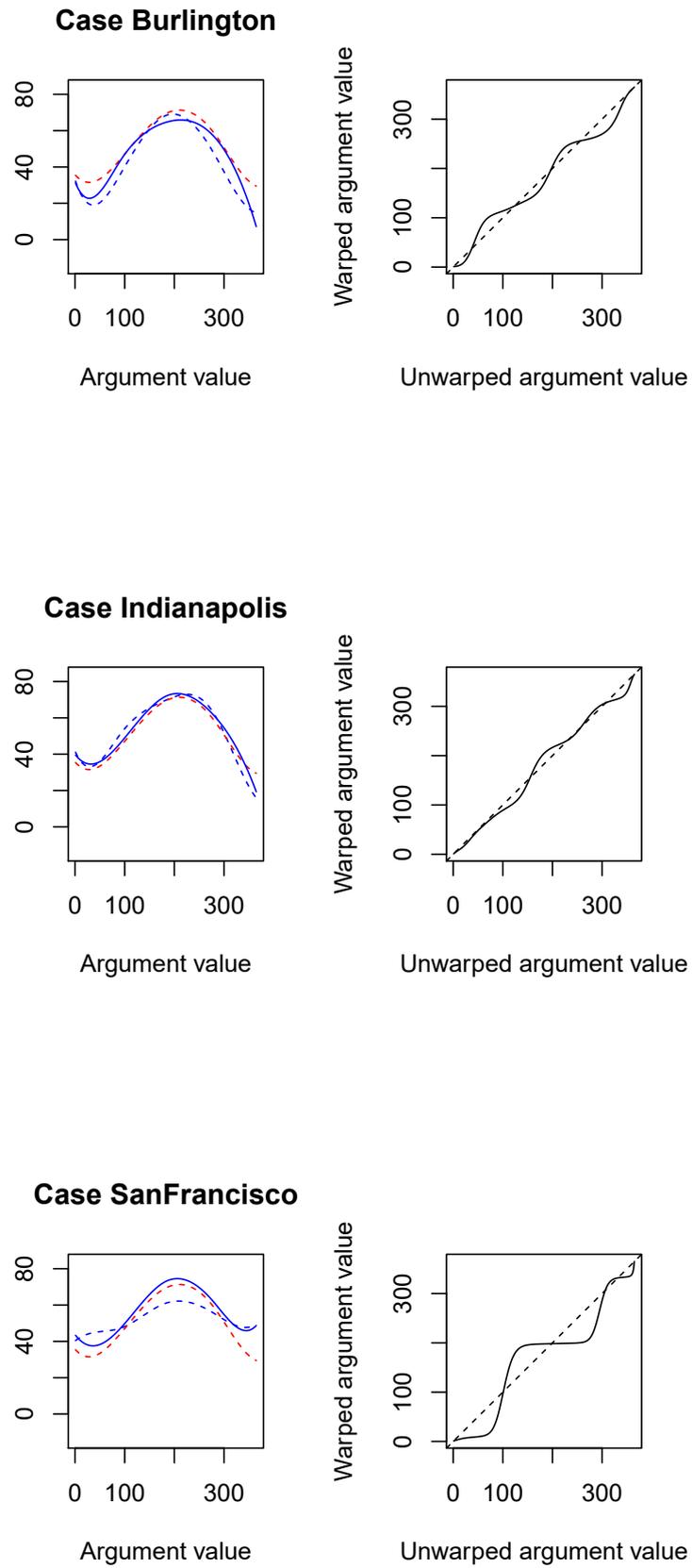


FIGURE 3.11: Continuous registration of three city and corresponding warping functions.

To determine the impacts of two types of registration (landmark and continuous methods), we plot all unregistered, landmark and continuously registered curves in Figure 3.12.

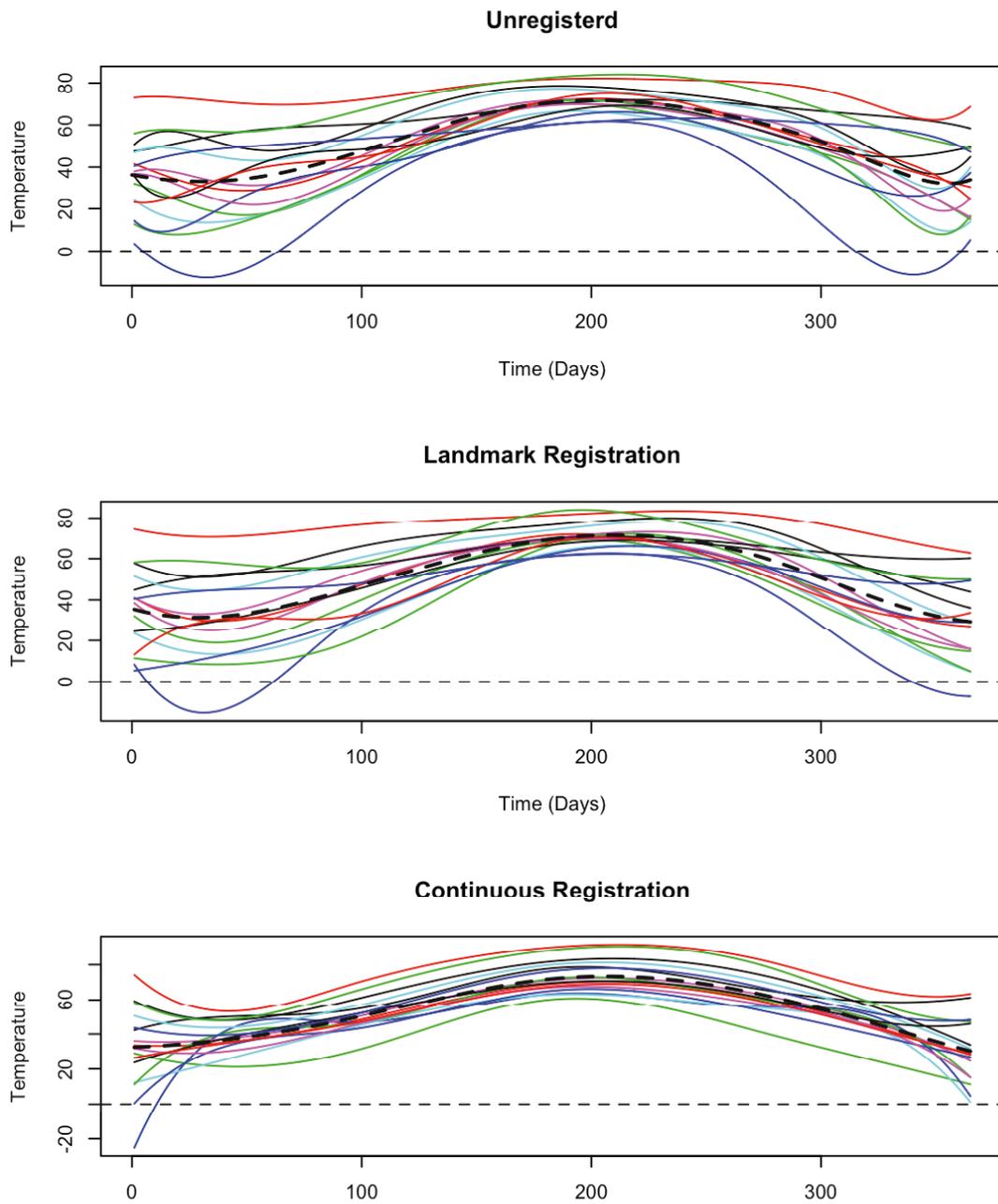


FIGURE 3.12: Plots of unregistered, landmark and registered data.

Figure 3.13 shows the mean of unregistered and registered data. The mean of the continuously registered acceleration curves is shown as black solid line, while that of the landmark registered curves is a dotted blue line. The light dashed line is the mean of the unregistered curves.

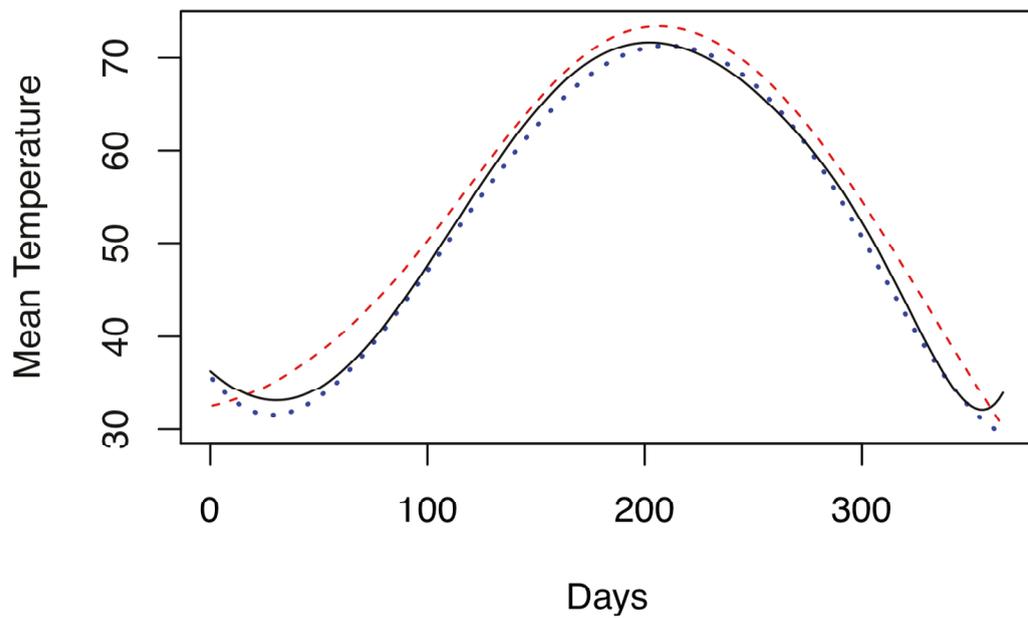


FIGURE 3.13: Mean of unregistered and registered data

Chapter 4

Conclusion

In this study we have tried to transform the discrete data to the functional form with close attention to functional data analysis methods. These methods were applied to weather data of sixteen cities of the United States during a 64 year interval. In this process, bases such as the Fourier series and splines were defined and to reach the functional form we used least square and generalized cross validation criterion. After forming the functional objects, we started to register the curves. The main reason for registration of weather data was the variation of the timing of special features (warmest or coldest day of a year) in different cities. Landmark and continuous registration methods were applied to the data in order to move these considerable features towards each other and have a more precise comparison about the temperature of different cities. According to the information acquired from registered curves, it is seen that weather temperature of all cities has experienced an upward trend during last 64 years.

Appendix A

R Codes

```
##### Load the fda library #####
```

```
library(fda)
```

```
##### Data set #####
```

```
USweather =
```

```
read.csv("/Users/Mahdi/Documents/Uni/Thesis/data/Datasets/mydatasetmatrix.csv", TRUE,  
",")
```

```
USweatherMatrix= as.matrix(USweather)
```

```
UScitynames = read.csv("/Users/Mahdi/Documents/Uni/Thesis/data/Datasets/UScitylist.csv",  
TRUE, ",", colClasses = "character" )
```

```
##### Creating Fourier basis #####
```

```
yearRng= 23376
```

```
Tempbasis1 = create.fourier.basis(yearRng, 5, period = 365.25)
```

```
Tempfd1 = smooth.basis(argvals = (1:23376)-0.5,
```

```
USweatherMatrix, Tempbasis1)$fd
```

```
Tempfd1$fdnames= list("Day", "Weather Station", "Mean temperature")
```

```
#####mean and std of functions
```

```
mean1= mean(Tempfd1)
```

```
sd1=sd.fd(Tempfd1)
```

```
##### Creating B-spline basis #####
```

```
Tempbasis2= create.bspline.basis(yearRng, 133, 6)
```

```
Tempfd2 = smooth.basis(argvals = (1:23376)-0.5,
```

```
USweatherMatrix, Tempbasis2)$fd
```

```
Tempfd2$fdnames= list("Day", "Weather Station", "Mean temperature")
```

```
#####mean and std of functions
```

```
mean2=mean(Tempfd2)
```

```
sd2=sd.fd(Tempfd2)
```

```
##### Plots Figure 9(a) 9(b) and Figure 10(a) and 10(b)
```

```
#####
```

```
plot(Tempfd1, xlim = c(1,3650), main="All Cities of Fourier Basis",  
xlab="Year", ylab="Temperature",xaxt='n')
```

```
plot(Tempfd2, xlim = c(1,3650), main="Order 6 Splines of All Cities",  
xlab="Year", ylab="Temperature",xaxt='n')
```

```
plot(Tempfd1[1], xlim = c(1,3650), main="Fourier Basis of Atlanta",  
xlab="Year", ylab="Temperature",xaxt='n')
```



```

tvec= c(1:23376)-0.5
coefficient2= coef(Tempfd2)
basismatrix2= eval.basis(tvec, Tempbasis2)
xhat2= basismatrix2%%coefficient2
diffmatrix2= (USweatherMatrix-xhat2)
diffmatrix22= (diffmatrix2)^2
squaredMatrix2 = apply(diffmatrix2, c(1,2), function(x) x^2)
sse= sum(squaredMatrix2)
ssen= (1/(23376-k))*sse
print(ssen)
}

```

```

ssp=c(3417.574, 3238.291, 3037.186, 1262.666, 2755.128, 2498.695, 2339.635, 2202.555,
      2056.651, 1935.361, 1792.426, 1691.904, 1585.551,1507.677,1413.343,1370.666,1297.635,
      1255.901,1219.691,1185.431,1150.567)
ksp= c(130:150)
plot(ksp,ssp, type = "b", main = " K versus total of squared residuals",
     xlab = "K", ylab = "Sumvector")

```

harmonic accelerator and penalized functions

```

harmaccelLfd = vec2Lfd(c(0,(2*pi/23376)^2,0), c(0, 23376))
Tempbasis1 = create.fourier.basis(yearRng, 5, period = 365.25)

```

```

loglam1 = seq(-2,7,0.5)
nlam1 = length(loglam1)
dfsave1 = rep(NA,nlam1)
names(dfsave1)= loglam1
gcvsave1 = dfsave1

```

```

for (ilam in 1:nlam1) {
  cat(paste('log10 lambda =',loglam1[ilam],'\n'))
  lambda = 10^loglam1[ilam]
  fdParobj1 = fdPar(Tempbasis1, harmaccelLfd, lambda)
  smoothlist1 = smooth.basis(argvals = (1:23376), USweatherMatrix,
                             fdParobj1)
  dfsave1[ilam] = smoothlist1$df
  gcvsave1[ilam] = sum(smoothlist1$gcv)
}

```

```

plot(loglam1, gcvsave1, type='b', lwd=2, main ='GCV Criterion',
     xlab=expression(log[10](lambda)),ylab =expression(log[10](GCV)) )

```

```

lambda1 = 1e6
fdParobj1 = fdPar(Tempbasis1, harmaccelLfd, lambda1)
logprec.fit1 = smooth.basis(argvals = 1:23376, USweatherMatrix, fdParobj1)
logprec.fd1 = logprec.fit1$fd
fdnames = list("Day ",
               "Weather Station" ,
               "Log 10 Temperature")

```

```
logprec.fd1$fdnames = fdnames
plot(logprec.fd1, xlim = c(0,3650),lty=1, lwd=1, cex=2)
```

```
##### penalized B-spline #####
```

```
Tempbasis2= create.bspline.basis(yearRng, 133, 6)
```

```
loglam2 = seq(-2,11,1)
nlam2 = length(loglam2)
dfsave2 = rep(NA,nlam2)
names(dfsave2)= loglam2
gcvsave2 = dfsave2
```

```
for (ilam in 1:nlam2) {
  cat(paste('log10 lambda =',loglam2[ilam],'\n'))
  lambda = 10^loglam2[ilam]
  fdParobj2 = fdPar(Tempbasis2, 4, lambda)
  smoothlist2 = smooth.basis(argvals = (1:23376), USweatherMatrix,
                             fdParobj2)
  dfsave2[ilam] = smoothlist2$df
  gcvsave2[ilam] = sum(smoothlist2$gcv)
}
```

```
plot(loglam2, gcvsave2, type='b', lwd=2, main='GCV Criterion',
      xlab=expression(log[10](lambda)),ylab = expression(log[10](GCV)) )
```

```
lambda2 = 1e10
fdParobj2 = fdPar(Tempbasis2, 4 , lambda2)
logprec.fit2 = smooth.basis(argvals = 1:23376, USweatherMatrix, fdParobj2)
logprec.fd2 = logprec.fit2$fd
fdnames = list("Day ",
               "Weather Station" ,
               "Log 10 Temperature")
logprec.fd2$fdnames = fdnames
plot(logprec.fd2, xlim = c(0,3650),lty=1, lwd=1, cex=2)
```

```
#####
###Deseasonalized data was obtained by subtracting
#the Fourier fits from a smooth of the original data.
```

```
differences=eval.fd(evalarg = (1:23376)-0.5, Tempfd2)-eval.fd(evalarg = (1:23376)-0.5,
Tempfd1)
```

```
Tempbasisdiff= create.bspline.basis(yearRng, 10,6)
Tempfddiff= smooth.basis(argvals = (1:23376)-0.5,
                         differences, Tempbasisdiff)$fd
```

```
##### splines of all the cities with the seasonal variation removed
```

```

plot(Tempfddiff,lty=1, lwd=1, cex=2, xlab="Year", ylab="B-spline fits-Fourier fits",main="All
cities",xaxt='n')
plot(Tempfddiff[6],lty=1, lwd=1, cex=2, xlab="Year", ylab="B-spline fits-Fourier fits",
main="Indianapolis",xaxt='n')

#####Splines of all the cities with seasonal variation removed penalized
#####against the 4th derivative using 1e10 as lambda, the smoothing parameter, and 10 basis
functions

lamdadiff= 1e10
fdParobjdiff = fdPar(Tempbasisdiff, 4, 1e10)
logprec.fitdiff = smooth.basis(argvals = 1:23376, differences, fdParobjdiff)
logprec.fdiff = logprec.fitdiff$fd

plot(logprec.fdiff,lty=1, lwd=1, cex=2)
plot(mean(logprec.fdiff), xlab="Year", ylab="Mean Value", xaxt='n')
#It is clear from these graphs temperature is increasing over time.

#####
#####
#####
#####
# Registration of data for just first year

#####
USweather5 = read.csv("/Users/Mahdi/Documents/Uni/Thesis/data/Datasets/firstyear.csv",
TRUE, ",")
USweatherMatrix5= as.matrix(USweather5)

##### Creating Bspline basis #####
yearRgn5=365
Tempbasis5= create.bspline.basis(yearRgn5,9,6)
Tempfd5 = smooth.basis(argvals = (1:365)-0.5, USweatherMatrix5, Tempbasis5)$fd
mean5= mean(Tempfd5)
Tempfd5$fdnames= list("Day", "Weather Station", "Mean temperature")

plot(Tempfd5, lty=1, lwd=1, cex=2, xaxt='n',ylab="Temeperature" ,xlab="Month")
lines(mean5,col=1, lwd=2, lty=2 )

plot(Tempfd5[3:4], lty=1, lwd=1, cex=2 , main="Burlington and Fairbanks", xlab="Month",
ylab="Temperature", xaxt='n' )

lambda5 = 1e8
fdParobj5 = fdPar(Tempbasis5, 4 , lambda5)
logprec.fit5 = smooth.basis(argvals = 1:365, USweatherMatrix5, fdParobj5)
logprec.fd5 = logprec.fit5$fd
fdnames = list("Day ","Weather Station" ,"Log 10 Temperature")
logprec.fd5$fdnames = fdnames

```

```
logmean=mean(logprec.fd5)
```

```
plot(logprec.fd5, lty=1, lwd=1, cex=2 )
```

```
plot(fdobject, lty=1, lwd=1, cex=2 )
```

```
fdobject
```

```
lines(logmean,col=1, lwd=2, lty=2 )
```

```
plot(logprec.fd5[3], lty=1, lwd=1, cex=2, xlab="Month",
```

```
ylab="Temperature" , main="Burlington, Indianapolis and San Francisco", xaxt='n')
```

```
lines(logmean,col=1, lwd=3, lty=2 )
```

```
lines(logprec.fd5[6], lty=1, lwd=1, cex=2, col=36)
```

```
lines(logprec.fd5[16], lty=1, lwd=1, cex=2, col=26)
```

```
#####
```

```
#####
```

```
##### Landmark registration
```

```
#####
```

```
#####
```

```
rooz= c(1:365)
```

```
city = 1:16
```

```
PGSctr1 = rep(0,length(city))
```

```
for (icase in city) {
```

```
  accveci5 = eval.fd(evalarg=(1:365),logprec.fd5[icase])
```

```
  plot(rooz,accveci5,"I",
```

```
    xlab="Year", ylab="Height Accel.",
```

```
    main=paste("Case",icase))
```

```
  lines(c(1,365),c(0,0),lty=2)
```

```
  PGSctr1[icase] = locator(1)$x
```

```
}
```

```
PGSctr2 = rep(0,length(city))
```

```
for (icase in city) {
```

```
  accveci5 = eval.fd(evalarg=(1:365),logprec.fd5[icase])
```

```
  plot(rooz,accveci5,"I",
```

```
    xlab="Year", ylab="Height Accel.",
```

```
    main=paste("Case",icase))
```

```
  lines(c(1,365),c(0,0),lty=2)
```

```
  PGSctr2[icase] = locator(1)$x
```

```
}
```

```
PGSctr3 = rep(0,length(city))
```

```
for (icase in city) {
```

```
  accveci5 = eval.fd(evalarg=(1:365),logprec.fd5[icase])
```

```
  plot(rooz,accveci5,"I",
```

```
    xlab="Year", ylab="Height Accel.",
```

```
    main=paste("Case",icase))
```

```
  lines(c(1,365),c(0,0),lty=2)
```

```
PGSctr3[icase] = locator(1)$x
```

```
}
```

```
PGSctr1
```

```
PGSctr2
```

```
PGSctr3
```

```
mypgstr= cbind(PGSctr2,PGSctr1,PGSctr3 )
```

```
mypgstr
```

```
PGSctrmean55= colMeans(mypgstr)
```

```
class(PGSctrmean55)
```

```
##### Mean of Landmarks #####
```

```
PGSctrmean5=mean(PGSctr1)
```

```
PGSctrmean52=mean(PGSctr2)
```

```
PGSctrmean53=mean(PGSctr3)
```

```
wbasisLM5 = create.bspline.basis(c(1,365), 7,4,breaks=seq(1, 365, len=5) )
```

```
WfdLM5 =fd(matrix(0,wbasisLM5$nbasis,1),wbasisLM5)
```

```
WfdParLM5 = fdPar(WfdLM5,1,1e-4)
```

```
fdoject= smooth.basis(argvals = 1:365, USweatherMatrix5,wbasisLM5)$fd
```

```
regListLM5 = landmarkreg(fdoject, mypgstr, PGSctrmean55,
```

```
WfdParLM5, TRUE)
```

```
accelfdLM50 = regListLM5$warpdf
```

```
accelfdLM5 = regListLM5$regfd
```

```
accelmeanfdLM5= mean(accelfdLM5)
```

```
##### plot of landmark registered data
```

```
plot(accelfdLM5, xlim=c(1,365), lty=1, lwd=1,
```

```
cex=2, xlab="Day", ylab="Mean Temperature")
```

```
lines(accelmeanfdLM5, col=1, lwd=2, lty=2)
```

```
lines(c(PGSctrmean5,PGSctrmean5), c(50,80), lty=2, lwd=1.5)
```

```
lines(c(PGSctrmean52,PGSctrmean52), c(20,50), lty=2, lwd=1.5)
```

```
lines(c(PGSctrmean53,PGSctrmean53), c(20,50), lty=2, lwd=1.5)
```

```
##### plot of all warping functions
```

```
plot(accelfdLM50, xlim=c(1,365), lty=1, lwd=1,
```

```
cex=2, xlab="Time", ylab="Transformed Time")
```

```
lines(x = c(0,365), y = c(0,365), lty=3)
```

```
##### Plots of Unregistered and Registered data
```

```
meanfun0= mean(logprec.fd5[city])
```

```
meanln0 =mean (accelfdLM5[city])
```

```
op = par(mfrow=c(2,1))
plot(logprec.fd5[city], xlim=c(1,365), lty=1, lwd=1,
      cex=2, xlab="Month", ylab="Temperature",xaxt='n')
lines(meanfun0, col=1, lwd=2, lty=2)
lines(c(PGSctrmean5,PGSctrmean5), c(50,80), lty=2, lwd=1.5)
lines(c(PGSctrmean52,PGSctrmean52), c(20,50), lty=2, lwd=1.5)
lines(c(PGSctrmean53,PGSctrmean53), c(20,50), lty=2, lwd=1.5)
plot(accelfdLM5[city], xlim=c(1,365), lty=1, lwd=1,
      cex=2, xlab="Month", ylab="Temperature",xaxt='n')
lines(meanln0, col=1, lwd=2, lty=2)
lines(c(PGSctrmean5,PGSctrmean5), c(50,80), lty=2, lwd=1.5)
lines(c(PGSctrmean52,PGSctrmean52), c(20,50), lty=2, lwd=1.5)
lines(c(PGSctrmean53,PGSctrmean53), c(20,50), lty=2, lwd=1.5)
par(op)
```

```
##### plot warping functions for cases 3 and 16
```

```
accelfdLM50 = regListLM5$warpdf
warpmatLM = eval.fd(evalarg=(1:365), accelfdLM50)
```

```
op = par(mfrow=c(2,2))
```

```
plot(logprec.fd5[16], xlim=c(1,365), lty=1, lwd=2,
      xlab="", ylab="")
lines(c(PGSctrmean5,PGSctrmean5), c(-20,180), lty=2, lwd=1.5)
```

```
plot((1:365), warpmatLM[,16], "l", lty=1, lwd=2, col=1, cex=1.2,xlim = c(190,365),
      xlab="", ylab="")
lines((180:365), (180:365), lty=2, lwd=1.5)
lines(c(PGSctrmean5,PGSctrmean5), c(1,365), lty=2, lwd=1.5)
text(PGSctrmean5+0.1, warpmatLM[205,16]+0.2, "o", lwd=2)
```

```
plot(logprec.fd5[3], xlim=c(1,365), lty=1, lwd=2,
      xlab="", ylab="")
lines(c(PGSctrmean5,PGSctrmean5), c(-20,80), lty=2, lwd=1.5)
```

```
plot((1:365), warpmatLM[,3], "l", lty=1, lwd=2, col=1, cex=1.2,xlim = c(115,235),
      xlab="", ylab="")
lines((100:365), (100:365), lty=2, lwd=1.5)
lines(c(PGSctrmean5,PGSctrmean5), c(1,365), lty=2, lwd=1.5)
text(PGSctrmean5+0.1, warpmatLM[185,10]+0.01, "o", lwd=2)
```

```
par(op)
```

```
##### h(t)>t; warped time is growing faster than clock time,
#and this is what we want if our observed process is running late
```

```
##### h(t)<t; clock time is being slowed down for a process that
#is running ahead of some target.
```

```
#####
```

```
#####
```

```
##### Continuous registration
```

```
#####
```

```
#####
```

```
wbasisCR = create.bspline.basis(c(1,365), 15, 5)
```

```
WfdOCR = fd(matrix(0,15,16),wbasisCR)
```

```
WfdParCR = fdPar(WfdOCR, 1, 1)
```

```
registerlistCR = register.fd(accelmeanfdLM5, accelfdLM5, WfdParCR)
```

```
accelfdCR = registerlistCR$regfd
```

```
warpfdCR = registerlistCR$warpfd
```

```
WfdCR = registerlistCR$Wfd
```

```
shiftCR = registerlistCR$shift
```

```
accelmeanfdCR = mean(accelfdCR)
```

```
plot(accelfdCR, xlim=c(1,365), lty=1, lwd=1,
```

```
  cex=2)
```

```
lines(accelmeanfdCR, xlim=c(1,365), lty=2, lwd=3,
```

```
  cex=2)
```

```
plot(warpfdCR, xlim=c(1,365), lty=1, lwd=1,
```

```
  cex=2)
```

```
# The left panel contains:
```

```
# -- the unregistered curve (dashed blue line)
```

```
# -- the target function (dashed red line) MEAN OF LANDMARK
```

```
# -- the registered curve (solid blue line)
```

```
# The right panel contains:
```

```
# -- the warping function h(t)
```

```
# -- the linear function corresponding to no warping
```

```
plotreg.fd(registerlistCR)
```

```
#####
```

```
#####Change the target to the mean of the registered functions
```

```
#####
```

```
registerlistCR1 = register.fd(accelmeanfdCR, accelfdLM5, WfdParCR)
```

```
accelfdCR1 = registerlistCR1$regfd
```

```
warpfdCR1 = registerlistCR1$warpfd
```

```
WfdCR1 = registerlistCR1$Wfd
shiftCR1 = registerlistCR1$shift
```

```
accelmeanfdCR1 = mean(accelfdCR1)
```

```
op = par(mfrow=c(2,1))
plot(accelfdCR, xlim=c(1,365), lty=1, lwd=1,
     cex=2)
lines(accelmeanfdCR, xlim=c(1,365), lty=2, lwd=3,
     cex=2)
```

```
plot(accelfdCR1, xlim=c(1,365), lty=1, lwd=1,
     cex=2)
lines(accelmeanfdCR1, xlim=c(1,365), lty=2, lwd=3,
     cex=2)
```

```
par(op)
```

```
plotreg.fd(registerlistCR1)
```

```
#####
##### plot Unregisterd and continuously registered curves
#####
```

```
meanln0 =mean (accelfdLM5[city])
```

```
accelmeanfdCR10 = mean(accelfdCR[city])
```

```
op = par(mfrow=c(3,1))
plot(logprec.fd5[city], xlim=c(1,365), lty=1, lwd=1,
     cex=2, xlab="Time (Days)", ylab="Temperature", main="Unregisterd")
lines(mean(logprec.fd5), col=1, lwd=2, lty=2)
```

```
plot(accelfdLM5[city], xlim=c(1,365), lty=1, lwd=1,
     cex=2, xlab="Time (Days)", ylab="Temperature", main="Landmark Registration")
lines(mean(accelfdLM5), col=1, lwd=2, lty=2)
```

```
plot(accelfdCR1[city], xlim=c(1,365), lty=1, lwd=1,
     cex=2, xlab="Time (Days)", ylab="Temperature", main="Continuous Registration")
lines(accelmeanfdCR10, col=1, lwd=2, lty=2)
```

```
par(op)
```

Average of temperature in unregistered data does not resemble to most of the observed curves,
#but in registered data ,average temperature shows a good illustration of observed curves.

Mean of Unregistered, landmark and continuous registered data
#####

```
accelmeanfdUN = mean(logprec.fd5)
accelmeanfdLM = mean(accelfdLM5)
accelmeanfdCR = mean(accelfdCR)
```

```
plot(accelmeanfdCR, xlim=c(1,365),col=2, lty=2, lwd=1,
     cex=1.2, xlab="Days", ylab="Mean Temperature",
     main = "Mean of unregisterde, landmark and continuous registered curves")
lines(accelmeanfdLM, col=4, lty=3, lwd=2)
lines(accelmeanfdUN, col=1, lty=1, lwd=1)
title(main = "mean of unregisterde , landmark and continuous registered",
      sub = "Red line is for unregisterde, Blue is for landmark and black is for continuous registerd")
```

References

- Green, P., & Silverman, B. (1994). *Nonparametric regression and generalized linear models. number 58 in monographs on statistics and applied probability.* Chapman & Hall.
- Karl, T. R., Melillo, J., Peterson, T., et al. (2009). Global climate change impacts in the united states. *Global Climate Change Impacts In The United States.*
- King, K. S. (2014). Functional data analysis with application to united states weather data.
- Ramsay, J. O. (2013). *Functional data analysis.* Retrieved from <http://www.psych.mcgill.ca/misc/fda/index.html>
- Ramsay, J. O., Hooker, G., & Graves, S. (2009). *Functional data analysis with r and matlab* (1st ed.). Springer Publishing Company, Incorporated.
- Ramsay, J. O., & Silverman, B. W. (2005). *Functional data analysis.* Springer.
- Ramsay, J. O., & Silverman, B. W. (2007). *Applied functional data analysis: methods and case studies.* Springer.
- Zhong, Z. (2008). *Curve registration in functional data analysis.* Arizona State University.