How Does Climate Change Affect Forest Fire Rate in British Columbia?

by

Bin Zhao

B.Sc., University of British Columbia, 2013

Project Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2015

Approval

Name:	Bin Zhao
Degree:	Master of Science
Title:	*How Does Climate Change Affect Forest Fire Rate in British Columbia?*
Examining Committee:	**Dr. Tim Swartz** (chair) Professor

Dr. Jiguo Cao
Senior Supervisor
Associate Professor _____

Dr. Jian Pei
Supervisor
Professor _____

Dr. David Campbell
Internal Examiner
Associate Professor _____

Date Defended: 20 August 2015

Abstract

Climate change is known to be an important risk of forest fire. Studies have shown an increased risk of fire because of rising temperatures, drier conditions, more lightning from stronger storms, added dry fuel for fires and a longer fire season and "global warming makes forests more susceptible to fire." In this paper, we use modern functional data analysis methods to explore the variations of forest fire rate in British Columbia, Canada among 63 consecutive years (1950-2012), and to investigate the historical effect of temperature and precipitation on forest fire rate. Functional principle component analysis shows that forest fire rate has increased since 2004 compared to years before that. Historical functional linear model shows that the concurrent effect of temperature and precipitation are both strong. Higher temperature and less precipitation lead to more forest fire. Temperature from January to July has a historical effect on forest fire rate from August to November, while only short term effect of precipitation up to two months is detected.

Keywords: Functional data analysis; Smoothing; Functional principle component analysis; Historical functional linear model

Acknowledgements

I would like to thank Canadian Forest Service for the Canadian National Fire Database and Environment Canada for the Homogenized Temperatures and Precipitation for Canada.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The data we are going to analyze include two parts: The Canadian National Fire Database ([2]) and Second Generation of Homogenized Temperatures and Precipitation for Canada ([13] and [7]). These data can be accessed from http://cwfis.cfs.nrcan.gc.ca/ha/nfdb and http://www.ec.gc.ca/dccha-ahccd/, respectively. The Canadian National Fire Database is a collection of forest fire locations and fire perimeters as provided by Canadian fire management agencies including provinces, territories, and Parks Canada. It provides the location of each fire and the specific date of occurrence for 63 consecutive years (1950-2012). Several studies have been conducted with this database. For example, [5] talks about the severity of large fires, which varies across boreal North America, and thus can be viewed as an agent of ecological diversity. [8] talks about the variation in risk factors of forest fire that also affects spatial fire patterns using analysis of variance and Pearson's correlation. [10] talks about the percentage of each ecozones area burnt annually across Canada. [3] talks about carbon emissions due to forest fire in Canada.

The Homogenized Temperatures and Precipitation for Canada are prepared for climate trends analysis in Canada. It provides daily and monthly maximum, minimum, median and mean temperature and precipitation for 338 weather stations across Canada. The non-climatic shifts, which are mainly due to the relocation of the station, changes in observing practices and automation ([12]), are identified and adjusted using regression models ([11]).

There are many papers focusing on association between climate change and forest fire. Climate change is known to be an important risk of forest fire. [1] shows an increased risk of fire because of rising temperatures, drier conditions, more lightning from stronger storms, added dry fuel for fires and a longer fire season and "global warming makes forests more susceptible to fire." [9] talks about the temperature-driven global fire regime in the 21st century, which is an unprecedentedly fire-prone environment.

Although many papers have studied direct association between climate change and forest fire, there are few investigations on the historical effect of temperature and precipitation on forest fire rate. For example, given the trajectory of temperature for the first 4 months,

what would the trajectory of forest fire rate be for the remaining 8 months?

In this paper, we investigate the effect of temperature and precipitation on forest fire rate by applying functional data analysis (FDA) methods. Since the data we are going to analyze are weekly fire counts, weekly mean temperature and weekly mean precipitation for the years 1950-2012, all of them can be viewed as functions of time (week). They are called functional data in FDA. There are FDA methods that can be applied to these functional data. For example, functional principal component analysis (FPCA) can be used on forest fire data to explore the variations of forest fire rate among years. Other methods, such as historical functional linear model, can be used to estimate the historical effect of temperature and precipitation on forest fire rate.

To obtain weekly forest fire counts in BC, we calculate the number of fires occurring in each week according to their occurrence date. As for temperature data, we take the average of 7 consecutive days' mean temperature as the weekly measurement. More details will be discussed in Section 2.1.1 and Section 2.1.2.

The rest of paper is organized as follows. In Chapter 2, we discuss the statistical methods, including missing data imputation, data smoothing, functional principle component analysis and historical functional linear model. In Chapter 3, we explore the variations of forest fire rate among years using functional principal component analysis, and estimate the effect of temperature and precipitation on forest fire rate using historical functional linear model. Conclusions are given in Chapter 4.

Chapter 2

Methods

In this chapter we discuss the statistical methods used in this paper. Missing values are replaced with random samples from observations that are available and satisfy certain criteria. The criteria we use are different for forest fire data and temperature data. Next, we transform our data into weekly-based form so that each year would have 52 observations. After data manipulation, we represent 52 discrete points in each year with smooth curves. Poisson process model will be used on forest fire data while the traditional Gaussian model will be used on temperature and precipitation data. Then we apply the functional principle component analysis on smoothed fire data. Finally we build a historical functional linear regression model between forest fire and temperature + precipitation.

2.1 Data Manipulation And Missing Value Imputation

2.1.1 The Canadian National Fire Database

In forest fire data, the information we need is number of fires happened in each week of year 1950 to 2012. However, some fire records only have reported year but no month or day. Table 2.1 summarizes the number of such records in the dataset.

We ignore these records for now and assign a "week number" for each of the rest records according to date they occurred. For example, fires occurred during January 1st to January 7th are consdiered week "1". There are exactly 52 weeks in one year when we treat February 29th as a day in week "9" and December 31st as a day in week "52". Next, we draw a random sample from $\{1, 2, ..., 52\}$ for each year in Table 2.1 with replacement. The sample size is determined by how many missing values are there in each year. The weight for each number

Table 2.1: Number of fire records without reported month or day in each year

Year	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2009	2010	2011	2012
Number of records	18	2	191	127	40	161	66	10	9	2	14	7	6	11

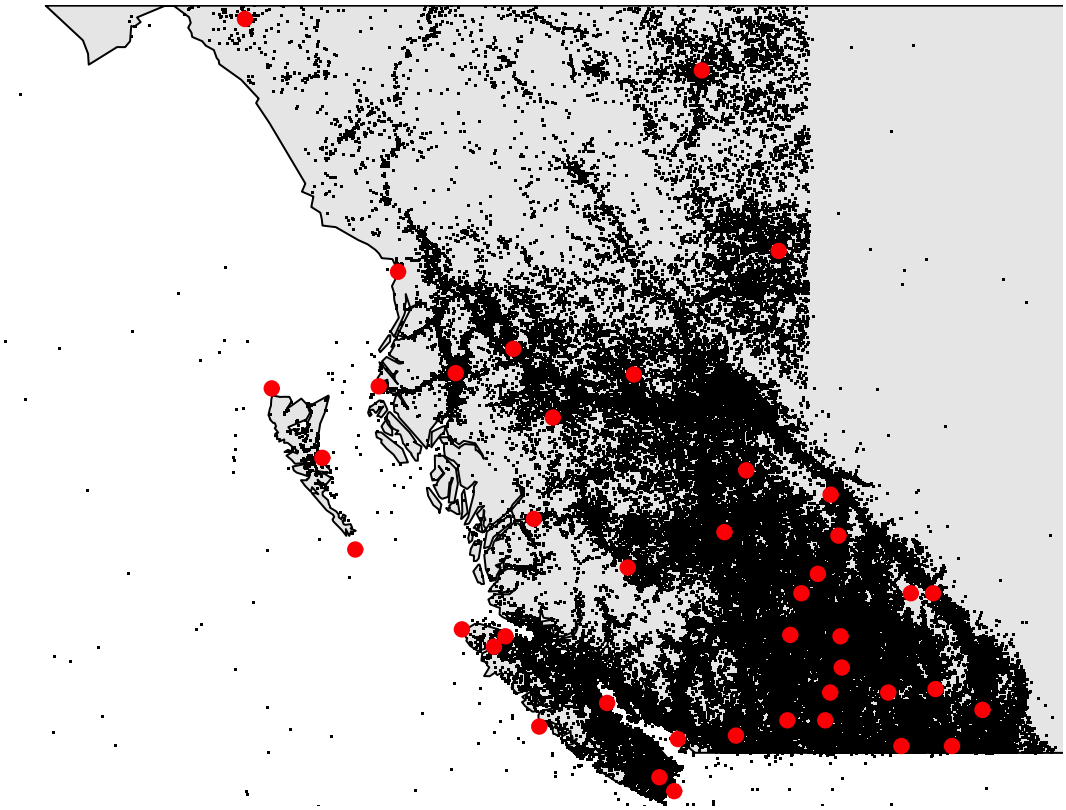

Figure 2.1: Map of British Columbia, Canada. Forest fire recorded within British Columbia during 1950-2012 are marked as small dots. Large dots are weather stations with available temperature data during that year range.

in $\{1, 2, ..., 52\}$ is determined by number of fires recorded in the corresponding week and year. Finally we replace the missing week numbers with sampled numbers.

2.1.2 Homogenized Temperatures and Precipitation for Canada

The Homogenized Temperatures and Precipitation for Canada contains daily mean temperatures and precipitation for a large number of weather stations across Canada in multiple years. Figure 2.1 shows the location of fires and weather stations where temperature and precipitation data are collected on the map of British Columbia. 42 stations within British Columbia and have records during year 1950 to 2012 are kept for analysis. For each missing temperature, we take a random sample of size 1 from those temperature records that have the same day, month and weather station. We also take the average for each day across stations so that there is only one record left for each day. Finally, we take the average for every 7 days as a weekly record to transform the temperature data to a weekly basis. The same method is used to deal with precipitation.

2.2 Data Smoothing: From Discrete Points to Smoothed Curves

Our data consist of 52 weekly fire count records, 52 weekly average temperature records and 52 weekly average precipitation records in each year for 63 consecutive years (1950-2012). We need to represent the data in 63 years with one single curve. The Gaussian model is used for temperature and precipitation data, while the Poisson Process model is used for forest fire data.

2.2.1 The Gaussian Model for Temperature and Precipitation Data

Since we use the exact same technique on temperature data and precipitation data, only temperature is discussed in this section. Let z_i represents the average temperature in a week, where $i = 1, 2, ..., 3276$. We smooth the 3276 discrete data points by removing measurement errors and represent them as a continuous function of time t. z_i can be modeled as

$$z_i = x(t_i) + \epsilon_i, \tag{2.1}$$

where $x(t)$ is the function, t_i represents the ith week and ϵ_i is the independent and identically distributed (i.i.d.) random error in Normal$(0, \sigma^2)$.
$x(t)$ is then approximated as a linear combination of K basis functions

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t) = \boldsymbol{\phi}(t)^T \mathbf{c}, \tag{2.2}$$

where $\boldsymbol{\phi}(t) = (\phi_1(t), ..., \phi_K(t))^T$ is a vector containing K basis functions and $\mathbf{c} = (c_1, ..., c_K)^T$ contains corresponding coefficients of basis functions.
We choose B-spline basis system for the temperature data. The basis coefficients c_k's are estimated by minimizing the sum of squared errors

$$\text{SSE} = \sum_{i=1}^{3276} \left[z_i - \sum_{k=1}^{K} c_k \phi_k(t_i) \right]^2 \tag{2.3}$$

Let $\mathbf{z}$ be a vector of length 3276 of observed weekly average temperatures z_i and $\boldsymbol{\Phi}$ be a $3276 \times K$ matrix containing values $\phi_k(t_i)$. Equation (2.3) can be written in matrix form as

$$(\mathbf{z} - \boldsymbol{\Phi}\mathbf{c})^T (\mathbf{z} - \boldsymbol{\Phi}\mathbf{c}). \tag{2.4}$$

The number of basis functions K need to be chosen. Generally speaking, the more basis functions we choose, the closer the fitted curve will be compared to the discrete data points. However, if we choose too many basis functions, the fitted curves may be too rough, thus we

overfit the data. Overfitting makes it difficult to interpret results derived from rough curves. In addition, since a lot of random errors are included in the curves, the results become questionable. We want the fitted curves to catch the trend of data without overfitting it. In order to achieve that, instead of controlling K, we add a roughness penalty term. We choose more basis functions than needed that will overfit the data, then combine the SSE with a penalty term, and finally minimize the sum of these two. A roughness penalty term is usually a term that is related to the roughness (or smoothness) of the fitted curve. For example, a commonly used penalty term is the integral of the square of the second derivative of the curve (L^2 *norm*). The more rough the fitted curve is, the larger the penalty term will be. On the contrary, SSE will get smaller as the fitted curve get rough. By minimizing the sum of these two terms, we are able to find a fitted curve that is not as rough as what we will end up with when only minimizing the likelihood because of the penalty term. We can also control how much weight to put on the penalty term by multiplying the penalty term with a constant μ. It is called "tuning parameter" or "smoothing parameter". When choosing a large μ, a small increase in roughness of the curve will increase the overall sum a lot. Thus the final fitted curve will be smoother. Two extreme cases will happen when $\mu = 0$ and $\mu = \infty$. For $\mu = 0$, there is no penalty on roughness, so the fitted curve is the same as what we get from minimizing negative likelihood only. When $\mu = \infty$, we will have the smoothest curve possible, which is a straight line when using L^2 *norm*.
We use L^2 *norm* as penalty for our temperature data. It is expressed as

$$\begin{aligned} \text{PEN} &= \mu \int [x''(t)]^2 dt \\ &= \mu \int \left[\sum_{k=1}^{K} c_k \phi_k''(t) \right]^2 dt, \end{aligned} \tag{2.5}$$

where $\phi_k''(t)$'s are second derivatives of basis functions.
We re-express the roughness penalty PEN in matrix terms as follows.

$$\text{PEN} = \mu \int \left[\boldsymbol{\phi}''^T(t)\mathbf{c} \right]^2 dt = \mu \mathbf{c}^T \mathbf{R} \mathbf{c} \tag{2.6}$$

where $\boldsymbol{\phi}''(t) = (\phi_1''(t), ..., \phi_K''(t))^T$ is a vector containing the secondderivatives of K basis functions and

$$\mathbf{R} = \int \boldsymbol{\phi}''(t) \boldsymbol{\phi}''^T(t) dt. \tag{2.7}$$

Composite Simpson's rule ([1]) is used to approximate the integral involved in equation 2.7 numerically. We have

$$\mathbf{R} = \sum_{r=1}^{R} v_r \left[\boldsymbol{\phi}''(u_r) \boldsymbol{\phi}''^T(u_r) \right], \tag{2.8}$$

where $u_r = a + (r-1)g$, $g = \frac{b-a}{R-1}$,

$$v_r = \begin{cases} \frac{g}{3} & r = 1 \text{ or } r = R \\ \frac{4g}{3} & r = 2, 4, 6... \\ \frac{2g}{3} & r = 3, 5, 7... \end{cases}$$

on the support of $t \in (a, b)$ and R is an integer.
The new criterion to be minimized, namely the penalized sum of squared errors, is

$$\text{PENSSE} = \text{SSE} + \text{PEN} \tag{2.9}$$

Equation 2.9 can be written in matrix terms as

$$\text{PENSSE} = (\mathbf{z} - \mathbf{\Phi c})^T(\mathbf{z} - \mathbf{\Phi c}) + \mu \mathbf{c}^T \mathbf{R c} \tag{2.10}$$

Finally, the estimate $\hat{\mathbf{c}}_i$ that minimizes (2.10) is

$$\hat{\mathbf{c}} = (\mathbf{\Phi}^T\mathbf{\Phi} + \mu\mathbf{R})^{-1}\mathbf{\Phi}^T\mathbf{z}. \tag{2.11}$$

The estimated smooth function is then

$$\hat{x}(t) = \boldsymbol{\phi}(t)^T\hat{\mathbf{c}} \tag{2.12}$$

2.2.2 The Poisson Process Model for Forest Fire Data

Since number of fires in each week is a count that can only take non-negative integer values, another possible choice to smooth the data is using Poisson process model. Assume fires that occur in year i follows an inhomogeneous Poisson process $N_i(t)$, which denotes the total number of fires occurring during time t in year i. A Poisson process is a stochastic process that counts the number of events(fires) and the time points at which these events occur in a given time interval. The time to which the next event occurs is independent of the other events and the numbers of events occurred in disjoint intervals are independent of each other. The Poisson process is inhomogeneous in the sense that events occur at a variable rate as time t varies. Let the rate parameter $\lambda(t)$ be a function of t. It is what we use to smooth the forest fire data.

2.2.3 The Maximum Likelihood Approach to the Rate Function

We use the maximum likelihood approach to estimate the rate function $\lambda(t)$. For our inhomogeneous Poisson process model, let $N_{a,b}$ be the expected number of events between

time a and b. Then

$$N_{a,b} = \int_a^b \lambda(t)dt. \tag{2.13}$$

Thus the number of fires in the time interval $[a, b]$, given as $N(b) - N(a)$, follows a Poisson distribution with associated parameter $N_{a,b}$

$$\Pr[N(b) - N(a) = k] = \frac{e^{-N_{a,b}}(N_{a,b})^k}{k!}, \quad k = 0, 1, ... \tag{2.14}$$

Considering our forest fire data, all 52 observations in each year follow Poisson distributions with rate $N_{j-1,j}$ for $j = 1, 2, ..., 52$, respectively.
Let L be the likelihood function and l be the log-likelihood function.

$$L = \prod_{j=1}^{52} \frac{e^{(-N_{j-1,j})}(N_{j-1,j})^{k_j}}{k_j!} \tag{2.15}$$

where $N_{j-1,j} = \int_{j-1}^{j} \lambda(t)dt$ is the rate of fire in week j and k_j is the number of fire observed in week j. One constraint on $\lambda(t)$ is that it has to be non-negative over t. To apply this constraint to the maximum likelihood function, we substitute $\lambda(t)$ with $e^{\rho(t)}$. The likelihood then becomes

$$L = \prod_{j=1}^{52} \frac{e^{[-\int_{j-1}^{j} \exp(\rho(t))dt]}[\int_{j-1}^{i} \exp(\rho(t))dt]^{k_j}}{k_j!} \tag{2.16}$$

The log-likelihood is

$$l = -\sum_{j=1}^{52} \int_{j-1}^{j} \exp(\rho(t))dt + \sum_{j=1}^{52} k_j \log[\int_{j-1}^{j} \exp(\rho(t))dt] - \sum_{j=1}^{52} log(k_j!) \tag{2.17}$$

Again, we write $\rho(t)$ as a linear combination of Fourier basis functions:
$1, \sin(\omega t), \cos(\omega t), \sin(2\omega t), \cos(2\omega t), ...$
where ω is the frequency. Let $\boldsymbol{\phi}(t) = (\phi_1(t), \phi_2(t), ...)^T$ be a vector containing these basis functions, the log-likelihood function becomes

$$l = -\int_0^{52} \exp(\mathbf{c}^T\boldsymbol{\phi}(t))dt + \sum_{j=1}^{52} k_j \log[\int_{j-1}^{j} \exp(\mathbf{c}^T\boldsymbol{\phi}(t))dt] - \sum_{j=1}^{52} log(k_j!) \tag{2.18}$$

where $\mathbf{c} = (c_1, c_2, ...)^T$ is the vector containing coefficients of basis functions.
To estimate the integrals involved in the above equation, we again use composite Simpson's rule. The log-likelihood function becomes

$$l = -\sum_{r=1}^{R} v_r \exp(\mathbf{c}^T\boldsymbol{\phi}(u_r)) + \sum_{i=1}^{52} k_i \log\{\sum_{q=1}^{Q}[w_q \exp(\mathbf{c}^T\boldsymbol{\phi}(s_{iq}))]\} - \sum_{i=1}^{52} log(k_i!) \tag{2.19}$$

where $u_r = 0 + (r-1)g$, $g = \frac{52}{R-1}$,

$$v_r = \begin{cases} \frac{g}{3} & r = 1 \text{ or } r = R \\ \frac{4g}{3} & r = 2, 4, 6... \\ \frac{2g}{3} & r = 3, 5, 7... \end{cases}$$

and $s_{iq} = i - 1 + (q-1)h$, $h = \frac{1}{Q-1}$,

$$w_q = \begin{cases} \frac{h}{3} & q = 1 \text{ or } q = Q \\ \frac{4h}{3} & q = 2, 4, 6... \\ \frac{2h}{3} & q = 3, 5, 7... \end{cases} . \tag{2.20}$$

We still use harmonic acceleration operator as penalty, which is expressed as

$$\mu \int_0^{52} [\lambda'''(t) + \omega\lambda'(t)]^2 dt = \mu \mathbf{c}^T \left\{ \sum_{r=1}^{R} v_r \left[\boldsymbol{\phi}'''(u_r) + \omega \boldsymbol{\phi}'(u_r) \right] \left[\boldsymbol{\phi}'''^T(u_r) + \omega \boldsymbol{\phi}'^T(u_r) \right] \right\} \mathbf{c} \tag{2.21}$$

One problem we have here is that there is no closed form for $\mathbf{c}$, so numerical method has to be used to find the maximum of likelihood. Similar to the trick we applied to SSE, we add the negative of log-likelihood (2.19) and harmonic acceleration operator (2.21) together and minimize it to find $\hat{\mathbf{c}} = \text{argmin}_{\mathbf{c}}(-(2.19) + (2.21))$.

2.2.4 Difficulties in Finding the Global Minimum

It is difficult to find the global minimum of a function in high dimension. We use Simulated annealing, an algorithm for global optimum searching, and carefully choose the initial point of iterations. A reasonable choice of initial point is $\hat{\mathbf{c}}_N$, the estimated coefficients under traditional Gaussian model, since $\hat{\mathbf{c}}_N$ and $\hat{\mathbf{c}}$ should be close to each other.

2.2.5 Functional Principal Component Analysis on Forest Fire Data

We use functional principal component analysis (FPCA) to study the variations in forest fire data. Before doing FPCA, the mean curve is usually subtracted. Let $\bar{y}(t) = \frac{1}{n}\sum_{i=1}^{n} y_i(t)$ and $r_i(t) = y_i(t) - \bar{y}(t)$. We will conduct FPCA on $r_i(t)$'s.
Let $\xi_1(t)$ be the first functional principal component (FPC). It is estimated by maximizing

$$\sum_{i=1}^{n} f_{i1}^2, \tag{2.22}$$

subject to $\|\xi_1\|^2 = \int \xi_1^2(s)ds = 1$, where $f_{i1} = \int \xi_1(s) r_i(s) ds$ is the first PC score of the i-th curve $r_i(t)$.
Similarly, the second FPC $\xi_2(t)$ is estimated by maximizing $\sum_{i=1}^{n} f_{i2}^2$, subject to $\|\xi_2\|^2 =$

$\int \xi_2^2(s)ds = 1$ and $\int \xi_1(s)\xi_2(s)ds = 0$, where $f_{i2} = \int \xi_2(s)r_i(s)ds$ is the second FPC score of the i-th curve $r_i(t)$. The subsequent FPCs, $\xi_3(t), ..., \xi_M(t)$, can be estimated similarly with additional constraints $\int \xi_u(s)\xi_v(s)ds = 0$ for all u, v where $1 \leq u < v \leq Q$.
Let

$$v(s,t) = \sum_{i=1}^{n} r_i(s)r_i(t) \tag{2.23}$$

be the variance-covariance function for $r_i(t)$'s. All FPCs can be calculated as the eigenfunctions of the following functional eigenequations

$$\int v(s,t)\xi_m(s)ds = \rho_m\xi_m(t), \tag{2.24}$$

where ρ_m is the corresponding eigenvalue, $q = 1, ..., M$ and $\rho_1 \geq ... \geq \rho_M$. Each eigenfunction $\xi_m(t)$ takes account $\frac{\rho_1}{\sum_{m=1}^{M}\rho_m}$ of the total variations among n curves. Usually only the first few eigenfunctions need to be calculated such that they take account more than 95% of the total variations.

2.2.6 Historical Functional Linear Regression

Let $y_i(t)$, $x_i(t)$ and $z_i(t)$, $t \in [0, T]$, be the forest fire rate, temperature and precipitation over time t in the i-th year, respectively. We use the historical functional linear model ([6]) to test the effect of temperature and precipitation on forest fire rate. The model is written as

$$y_i(t) = \beta_0(t) + \int_0^t \beta_1(s,t)x_i(s)ds + \int_0^t \beta_2(s,t)z_i(s)ds + \epsilon_i(t). \tag{2.25}$$

$\beta_1(s,t)$ represents the effect of temperature $x_i(s)$ at time s on the forest fire rate $y_i(t)$ at time t. The integral is from 0 to t such that only x_i before t will affect y_i at t. In this way the model avoids backwards causation. And as a result, the support of $\beta_1(s,t)$ becomes a triangle rather than a rectangle. Figure 2.2 shows an example of our forest fire versus temperature case. We are going to estimate $\beta_1(s,t)$ over this triangle support and test the significance of it. Similarly, $\beta_2(s,t)$ represents the effect of precipitation $z_i(s)$ at time s on the forest fire rate $y_i(t)$ at time t. The same procedure is applied to $\beta_2(s,t)$ as $\beta_1(s,t)$.
In order to simplify the estimation of model (2.25), we only consider temperature effect for now and ignore precipitation. The model then becomes

$$y_i(t) = \beta_0(t) + \int_0^t \beta_1(s,t)x_i(s)ds + \epsilon_i(t). \tag{2.26}$$

To further simplify the notations, we drop the intercept function $\beta_0(t)$ by subtracting the mean curve from our data. Let $y_i^*(t) = y_i(t) - \bar{y}(t)$ and $x_i^*(t) = x_i(t) - \bar{x}(t)$, where $\bar{y}(t) =$

$\frac{1}{n}\sum_{i=1}^{n} y_i(t)$ and $\bar{x}(t) = \frac{1}{n}\sum_{i=1}^{n} x_i(t)$. We obtain

$$y_i^*(t) = \int_0^t \beta_1(s,t) x_i^*(s) ds + \epsilon_i(t). \tag{2.27}$$

In addition, we drop the asterisk in what follows.
Let $\beta_1(s,t)$ be approximated by $\hat{\beta}_1(s,t)$, a linear combination of known 2-dimentional basis functions $\phi_k(s,t)$, namely

$$\hat{\beta}_1(s,t) = \sum_{k=1}^{K} b_k \phi_k(s,t). \tag{2.28}$$

We construct the 2-dimentional basis function system by taking the tensor product of two 1-dimentional basis function systems. Let $\eta_1(s), \eta_2(s), ..., \eta_{K_1}(s)$ and $\theta_1(t), \theta_2(t), ..., \theta_{K_2}(t)$ be the two 1-dimentional basis function systems. Define the 2-dimentional basis function system as the set of basis functions

$$\{\eta_{k_1}(s)\theta_{k_2}(t)\}, \tag{2.29}$$

where $k_1 \in \{1, ..., K_1\}, k_2 \in \{1, ..., K_2\}$ and $k_1 < k_2$. The constraint $k_1 < k_2$ arises because of the triangular shape of the support of $\beta_1(s,t)$. Figure 2.2 shows one possible basis function system that is consist of the tensor product of two 1-dimentional basis function of size 10. The total number of 2-dimentional basis functions $K = \frac{K_1(K_2+1)}{2}$ when $K_1 = K_2$.
We continue working on the estimation of $\beta(s,t)$. Define

$$\psi_{ik}(t) = \int_0^t x_i(s)\phi_k(s,t) ds. \tag{2.30}$$

Equation 2.25 becomes

$$y_i(t) = \sum_{k=1}^{K} b_k \int_0^t x_i(s)\phi_k(s,t) ds + \int_0^t x_i(s)\epsilon_a(s,t) ds + \epsilon_i(t) = \sum_{k=1}^{K} b_k \psi_{ik}(t) + \epsilon_i'(t), \tag{2.31}$$

where $\epsilon_a(s,t) = \beta_1(s,t) - \hat{\beta}_1(s,t)$ is the error due to approximation of $\beta_1(s,t)$ and $\epsilon_i'(t)$ is the combined error.
Let $\mathbf{y}(t)$ and $\mathbf{e}(t)$ be the vectors of length n containing values of $y_i(t)$ and $\epsilon_i'(t)$, respectively. Let $\mathbf{\Psi}(t)$ be the $n \times K$ matrix containing values of $\psi_{ik}(t)$ and let $\mathbf{b} = (b_1, ..., b_K)^T$ be the coefficient vector. We re-express equation (2.25) as

$$\mathbf{y}(t) = \mathbf{\Psi}(t)\mathbf{b} + \mathbf{e}(t). \tag{2.32}$$

We wish to minimize

$$\text{SSE} = \int_0^T \sum_{i=1}^{n} \{\epsilon_i'(t)\}^2 dt, \tag{2.33}$$

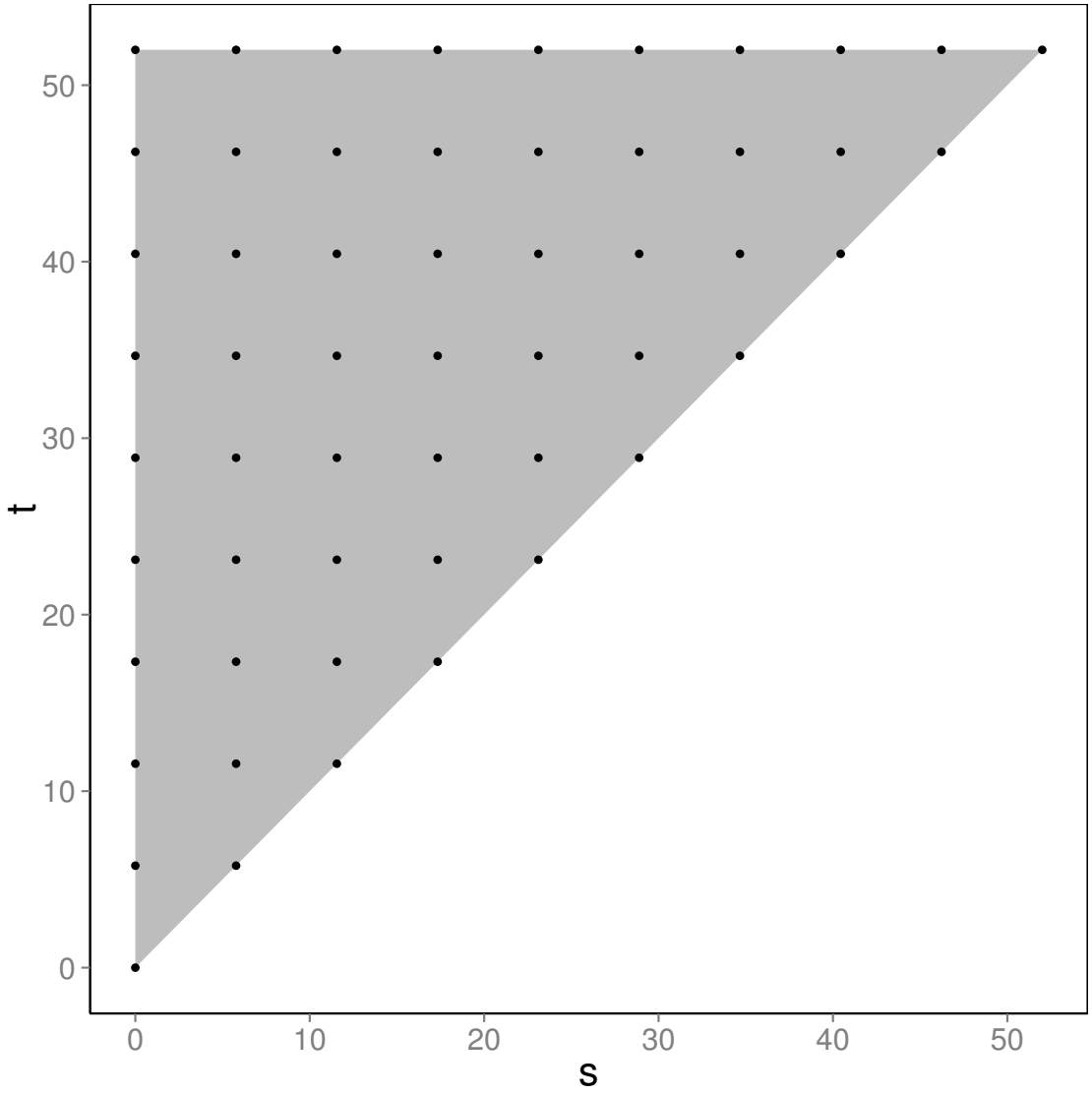

Figure 2.2: An example of triangular support and basis functions. The grey shadowed area is the triangular support of $\beta_1(s,t)$. Each black dot indicates a 2-dimentional basis function. In this example, s and t represent time (in weeks) of which mean temperature and number of forest fire are recorded, respectively, thus $s \in (0, 52)$ and $t \in (0, 52)$. These basis functions are constructed by the tensor product of two 1-dimentional basis function systems of size 10. Getting rid of those basis functions that are out of the support, we end up with 55 of them.

which is equivalent to solving for the normal equations

$$\Big\{ \int_0^T \mathbf{\Psi}^T(t)\mathbf{\Psi}(t)dt \Big\}\mathbf{b} = \int_0^T \mathbf{\Psi}^T(t)\mathbf{y}(t)dt. \tag{2.34}$$

Similar to section 2.2.1 and 2.2.2, we need to take roughness penalty into consideration. For our 2-dimentional basis functions, we consider roughness in each dimension separately. Let

$$\text{PEN}_1 = \lambda_1 \iint \Big[\frac{D^3\beta_1(s,t)}{ds^3} + \omega^2 \frac{D\beta_1(s,t)}{ds} \Big]^2 dsdt \tag{2.35}$$

and

$$\text{PEN}_2 = \lambda_2 \iint \Big[\frac{D^3\beta_1(s,t)}{dt^3} + \omega^2 \frac{D\beta_1(s,t)}{dt} \Big]^2 dsdt \tag{2.36}$$

be the harmonic acceleration operators along s and t, respectively, where λ_1 and λ_2 are smoothing parameters and ω is the frequency.

$$\begin{aligned} \text{PEN}_1 &= \lambda_1 \iint \Big[\sum_{k=1}^K b_k \phi'''_{k(ds)}(s,t) + \omega^2 \sum_{k=1}^K b_k \phi'_{k(ds)}(s,t) \Big]^2 dsdt \\ &= \lambda_1 \iint \Big[\sum_{k=1}^K b_k \Big(\phi'''_{k(ds)}(s,t) + \omega^2 \phi'_{k(ds)}(s,t) \Big) \Big]^2 dsdt, \end{aligned} \tag{2.37}$$

where $\phi'''_{k(ds)}(s,t)$ and $\phi'_{k(ds)}(s,t)$ represent the third and first derivatives w.r.t. s.
Let $\boldsymbol{\phi}_1(s,t)$ be a length K vector containing functions $\phi'''_{k(ds)}(s,t) + \omega^2\phi'_{k(ds)}(s,t)$ for $k = 1, 2, ..., K$. Equation (2.37) can be written in matrix terms as

$$\begin{aligned} \text{PEN}_1 &= \lambda_1 \iint \big[\mathbf{b}^T\boldsymbol{\phi}_1(s,t)\big]^2 dsdt \\ &= \lambda_1 \iint \big[\mathbf{b}^T\boldsymbol{\phi}_1(s,t)\boldsymbol{\phi}_1^T(s,t)\mathbf{b}\big] dsdt \\ &= \lambda_1 \mathbf{b}^T \Big\{ \iint \big[\boldsymbol{\phi}_1(s,t)\boldsymbol{\phi}_1^T(s,t)\big] dsdt \Big\}\mathbf{b} \\ &= \lambda_1 \mathbf{b}^T \mathbf{R}_1 \mathbf{b}, \end{aligned} \tag{2.38}$$

where $\mathbf{R}_1 = \iint \big[\boldsymbol{\phi}_1(s,t)\boldsymbol{\phi}_1^T(s,t)\big] dsdt$ is an $K \times K$ matrix.
Similarly, we can work out the matrix form of PEN_2

$$\text{PEN}_2 = \lambda_2 \mathbf{b}^T \mathbf{R}_2 \mathbf{b}, \tag{2.39}$$

where $\mathbf{R}_2$ resembles $\mathbf{R}_1$ except that derivatives of basis functions involved are taken w.r.t. t.
Now we wish to minimize the summation of SSE and two penalty terms, denoted as

$$\text{PENSSE} = \text{SSE} + \text{PEN}_1 + \text{PEN}_2, \tag{2.40}$$

which is equivalent to solving for the normal equations

$$\left\{ \int_0^T \mathbf{\Psi}^T(t)\mathbf{\Psi}(t)dt + \lambda_1\mathbf{R}_1 + \lambda_2\mathbf{R}_2 \right\}\mathbf{b} = \int_0^T \mathbf{\Psi}^T(t)\mathbf{y}(t)dt. \tag{2.41}$$

There are three types of integrals involved in equation (2.41): The large integrals over $(0, T)$ that we can see directly on both sides of equation; small integrals in $\psi_{ik}(t)$ contained in $\mathbf{\Psi}(t)$; and double integrals in $\mathbf{R}_1$ and $\mathbf{R}_2$. We approximate the latter two types using numerical method such as the composite Simpson's rule. As for the first type of integrals, we approximate them by a multivariate linear model. Evaluate $y_i(t)$ and $\psi_{i1}(t), \psi_{i2}(t), ..., \psi_{iK}(t)$ at a finite set of time points $t_q, q = 0, ..., Q$. This gives us

$$\mathbb{E}(\mathbf{y}_i) = \mathbf{\Psi}_i\mathbf{b}, \tag{2.42}$$

where $\mathbf{y}_i = (y_i(t_0), ..., y_i(t_Q))^T$ and

$$\mathbf{\Psi}_i = \begin{bmatrix} \psi_{i1}(t_0) & \cdots & \psi_{ik}(t_0) & \cdots & \psi_{iK}(t_0) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \psi_{i1}(t_Q) & \cdots & \psi_{ik}(t_Q) & \cdots & \psi_{iK}(t_Q) \end{bmatrix} .$$

Stacking these matrices $\mathbf{y}_i$ and $\mathbf{\Psi}_i$ on top of each other, we obtain the $n(Q+1) \times 1$ and $n(Q+1) \times K$ matrices $\mathbf{y}$ and $\mathbf{\Psi}$, respectively. The normal equations are

$$\{\mathbf{\Psi}^T\mathbf{\Psi} + \lambda_1\mathbf{R}_1 + \lambda_2\mathbf{R}_2\}\mathbf{b} = \mathbf{\Psi}^T\mathbf{y}, \tag{2.43}$$

thus the estimated coefficients are

$$\hat{\mathbf{b}} = \{\mathbf{\Psi}^T\mathbf{\Psi} + \lambda_1\mathbf{R}_1 + \lambda_2\mathbf{R}_2\}^{-1}\mathbf{\Psi}^T\mathbf{y}. \tag{2.44}$$

To add the precipitation effect back to our model, we do the following procedure.

1. Choose another set of 2-dimentional basis functions.
2. Define the precipitation version of "$\mathbf{\Psi}$" matrix, combine it with $\mathbf{\Psi}$ by column. Denote the combined matrix as $\mathbf{\Delta}$.
3. Define two penalty terms for precipitation as $\lambda_3\mathbf{R}_3$ and $\lambda_4\mathbf{R}_4$.
4. The estimated coefficients, namely $\hat{\mathbf{c}}$, is then $\hat{\mathbf{c}} = \{\mathbf{\Delta}^T\mathbf{\Delta} + \lambda_1\mathbf{R}_1 + \lambda_2\mathbf{R}_2 + +\lambda_3\mathbf{R}_3 + \lambda_4\mathbf{R}_4\}^{-1}\mathbf{\Delta}^T\mathbf{y}$.

Chapter 3

Results

3.1 Smoothed Curves of Temperature Data

The discrete temperature data are smoothed using 1827 B-spline basis functions. The smoothing parameter is chosen to be $\mu = 100$. Figure 3.1 shows the results of smoothing the temperature data for a selection of years using Gaussian model. It shows that mean weekly temperatures in BC are mostly below zero in January, mid-November and December. The highest temperature appears around week 30, which corresponds to late July.

3.2 Smoothed Curves of Precipitation Data

We also use 1827 B-spline basis functions to smooth precipitation data. The smoothing parameter is chosen to be 100. Figure 3.2 shows the results of smoothing the precipitation data for a selection of years using Gaussian model. The precipitation has more variations between weeks compared to temperature. There is more precipitation in winters than summer. November has the highest precipitation in a year.

3.3 Smoothed Curves of Forest Fire Data

For the forest fire data, we use the Poisson process model to do the smoothing. The smoothing parameter we choose is $\mu = 100$. Results for selected years are shown in figure 3.3. Most years have 0 forest fire recorded in January, February, November and December. June and July have the most forest fires recorded and the number reaches the peak at the end of July.

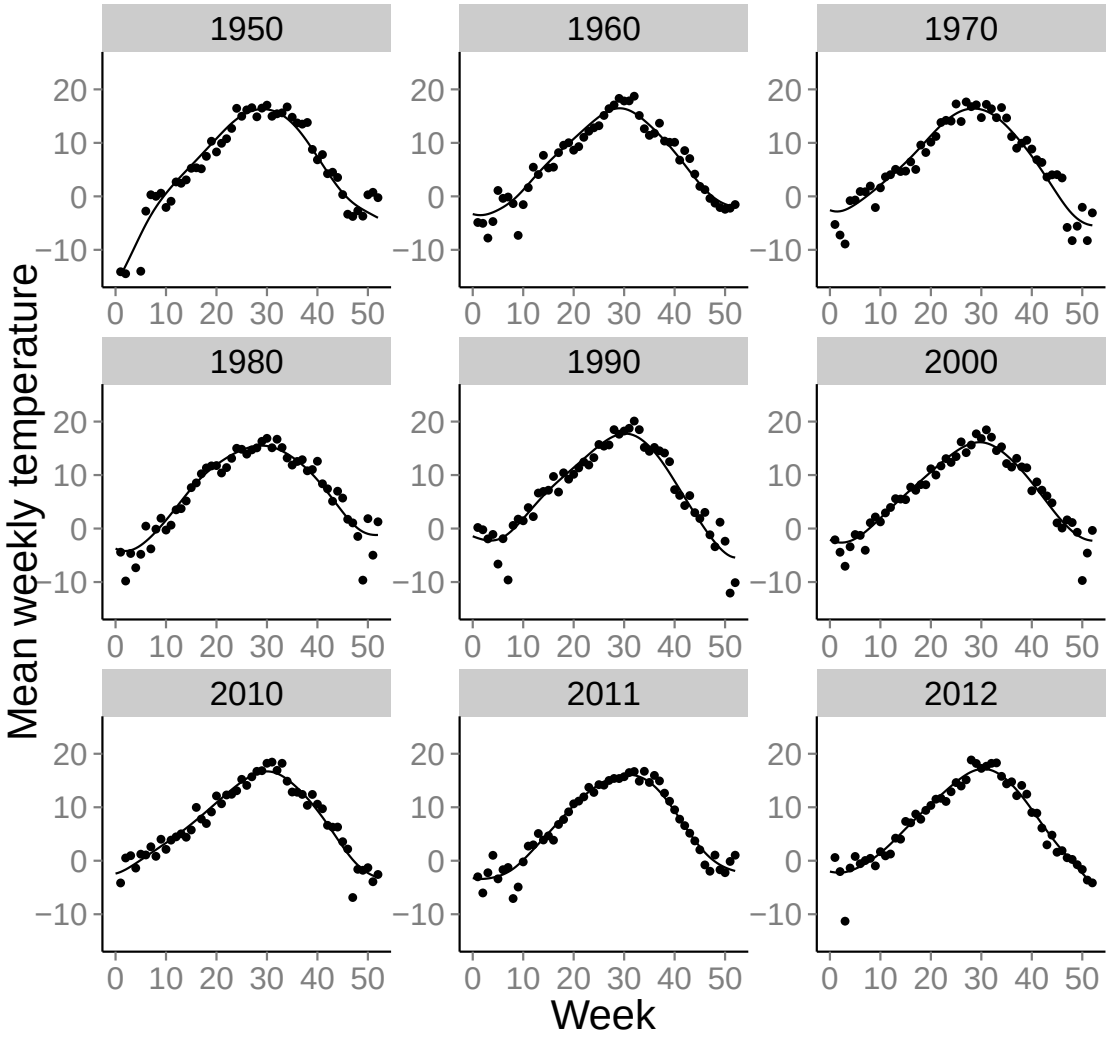

Figure 3.1: Discrete points are original records of mean temperatures in each week. Smoothed curves are derived using the Gaussian model. A selection of years is shown here.

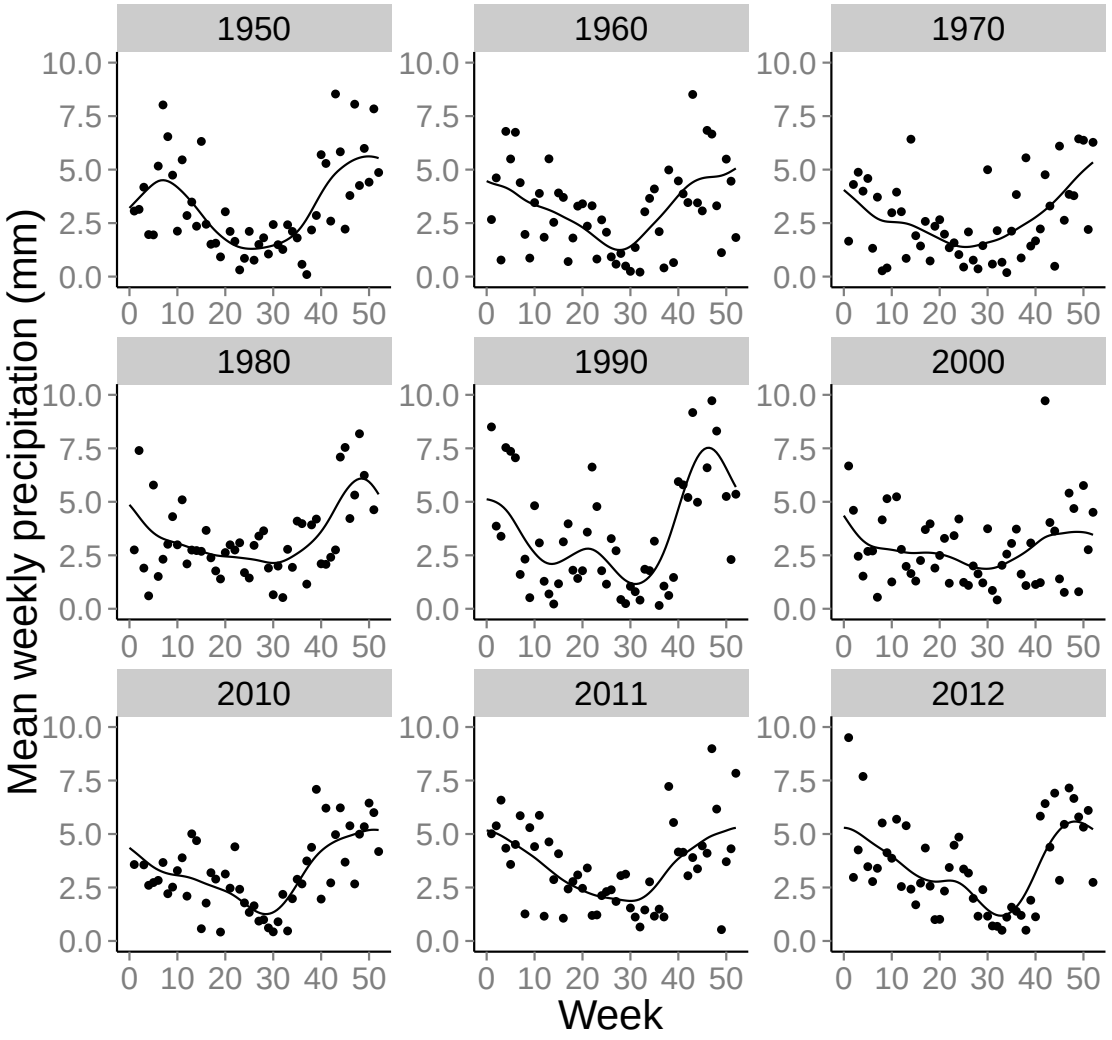

Figure 3.2: Discrete points are original records of mean precipitation in each week. Smoothed curves are derived using the Gaussian model. A selection of years is shown here.

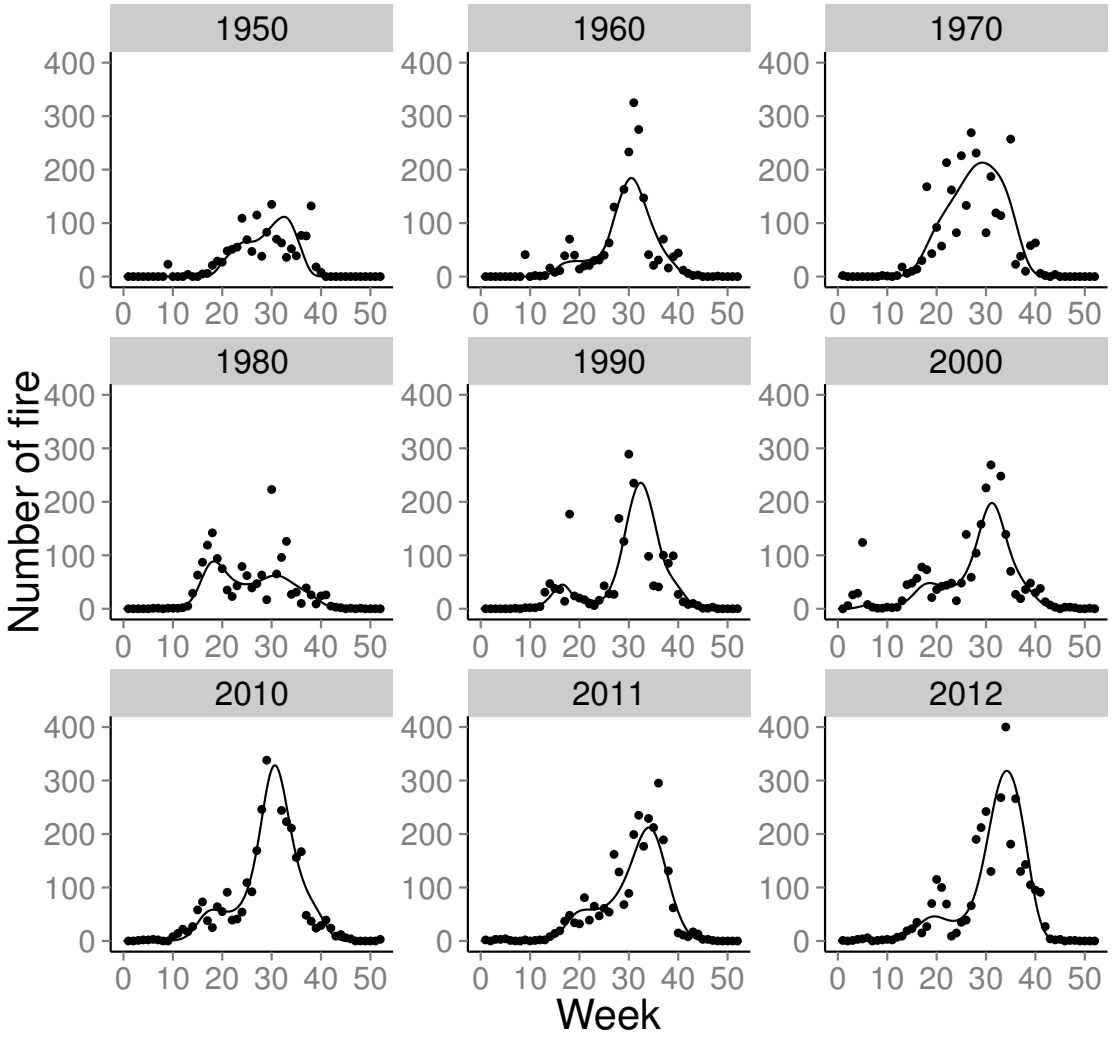

Figure 3.3: Discrete points are original records of number of fires in each week. Smoothed curves are derived using the Poisson process model. Results of a selection of years are shown here.

3.4 FPCA

Functional principal component analysis (FPCA) is used here to detect the variations in the number of forest fires recorded within British Columbia among 63 years. We choose the first four functional principal components so that more than 95% of the total variations is covered. Figure 3.4 shows the four FPCs. Each of them takes account 72.6%, 14.3%, 6.8% and 3.4% of the total variations, respectively. FPC1 is positive throughout the year, but the FPC1 value in winter is close to 0, while it rises through spring and peaks in summer. It means that forest fires are most variable in summer and very stable in winter. Year 2009 and 2008 have very high number of fire recorded in the summer and thus have very large FPC1 scores. FPC2 is positive through April to July (early summer), negative through August to October (late summer) and 0 otherwise. It means FPC2 score measures the change of forest fire between early summer and late summer. For example, there is much less forest fire recorded in early summer compared to late summer in year 2012 , thus it has the smallest FPC2 score (-377). In year 2004, there is much more forest fire recorded in early summer compared to late summer, thus it has the largest FPC2 score (238). FPC3 and FPC4 are not very interpretable. Figure 3.5 displays the scatterplot of PFC2 score versus PFC1 score for each year. We can see clearly that year 2004 and 2012 have extreme FPC2 scores, which means they have large difference between early summer and late summer, and year 2009 has the largest FPC1 score, which indicates large number of forest fires in summer. In addition, most years after 2000 are on the right side of panel, which means there have been more forest fire in recent years than in the past during summer.

3.5 Historical Functional Linear Model

The effect of temperature and precipitation on forest fire rate is modelled by the historical functional linear model. Two sets of 19 Fourier basis functions are chosen to construct the 2-dimentional basis functions, making it a total of 190 basis functions. Two smoothing parameters are chosen to be 1e6. An identical setup is applied to precipitation as well.
Figure 3.6 shows the predicted forest fire curves from the linear model for some years, along with the observed data and smoothed curve. Our model predicts the trend well during off-summer. However, for some years in the summer where extreme number of fires are observed, our model performs conservatively.

3.5.1 Estimated Temperature/Precipitation Effect And Permutation Tests

Figure 3.7 is a heat map showing the estimated effect of temperature on forest fire. In addition, in order to know whether the effect differs from 0 significantly, we use permutation tests to find point-wise confidence intervals at a series of time points. We perform the following procedure:

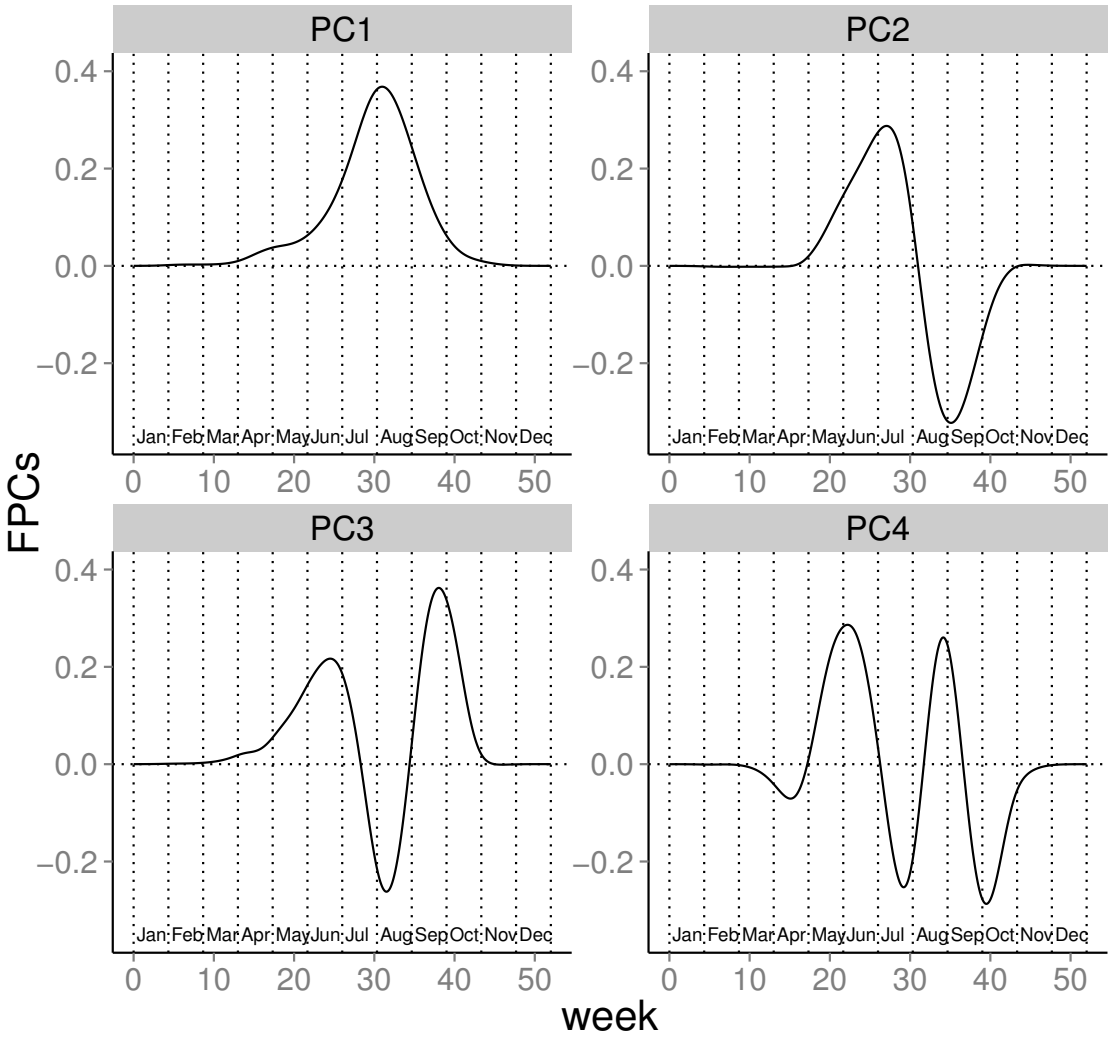

Figure 3.4: Four functional principal components for the forest fire data are plotted. Each of them takes account 72.6%, 14.3%, 6.8% and 3.4% of the total variations, respectively.

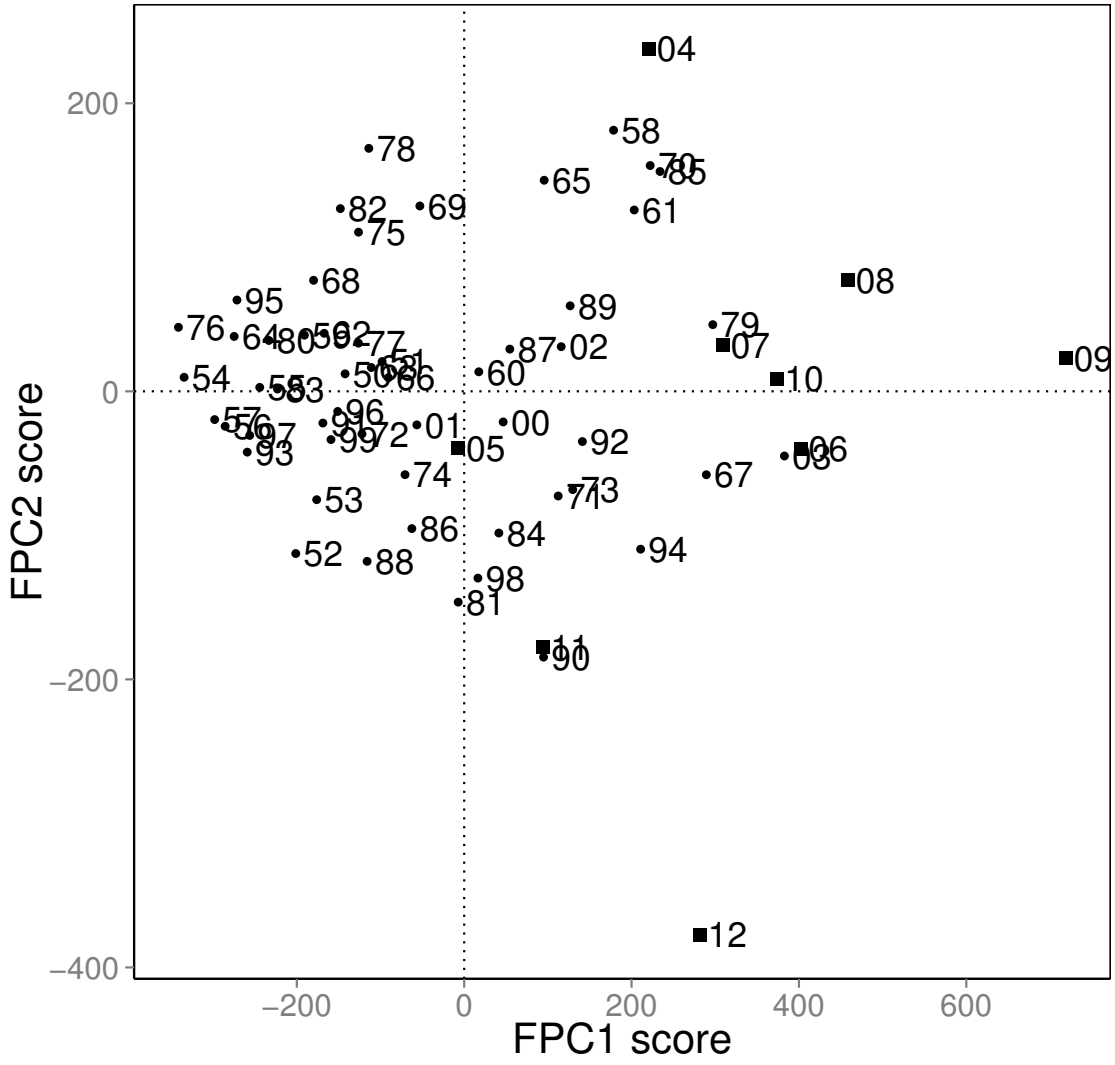

Figure 3.5: A scatterplot of FPC2 score versus FPC1 score.

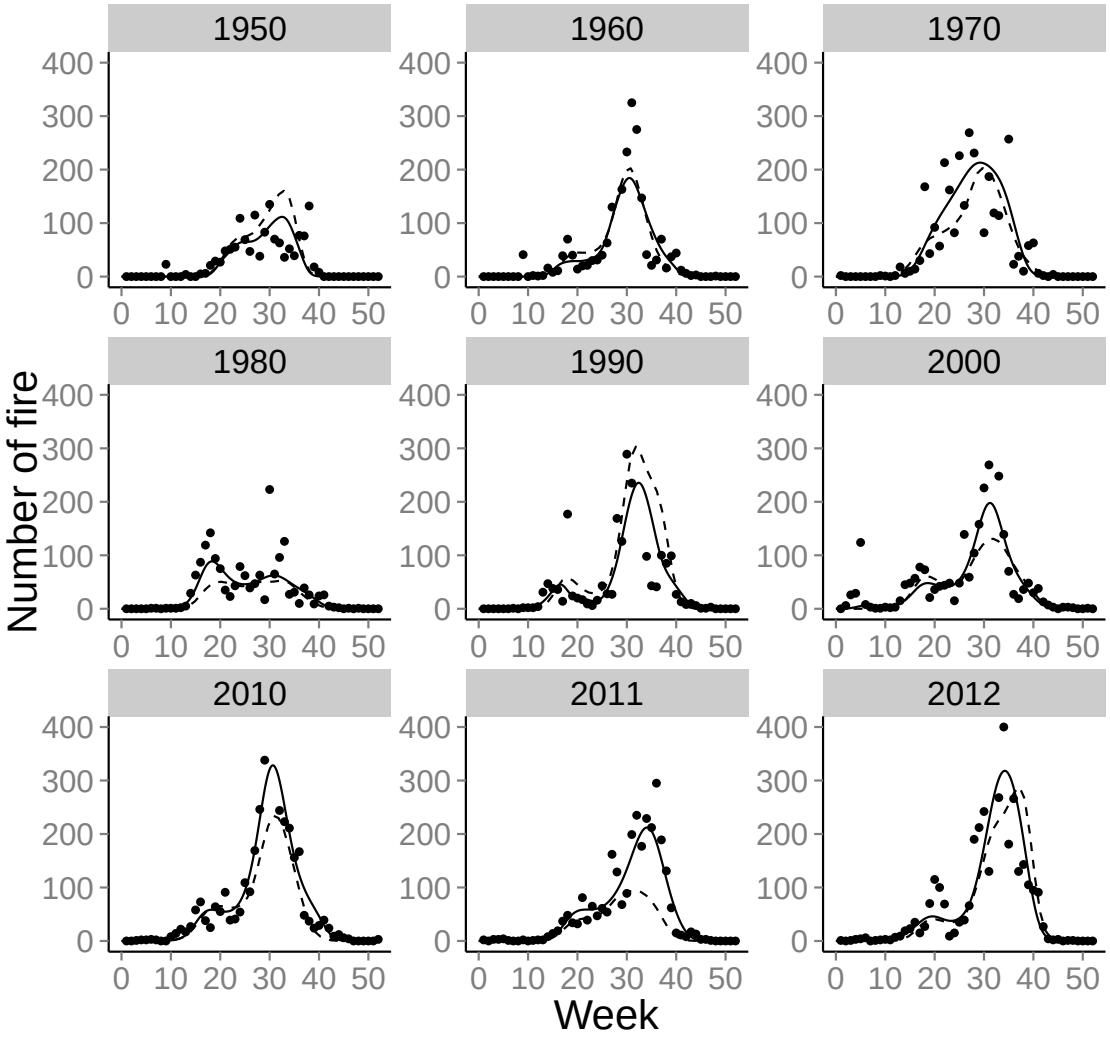

Figure 3.6: Discrete points shows original number of fires in each week. Solid curves are smoothed fire data using Poisson process model. Dashed curves are predicted fire curves from temperature using historical functional linear model.

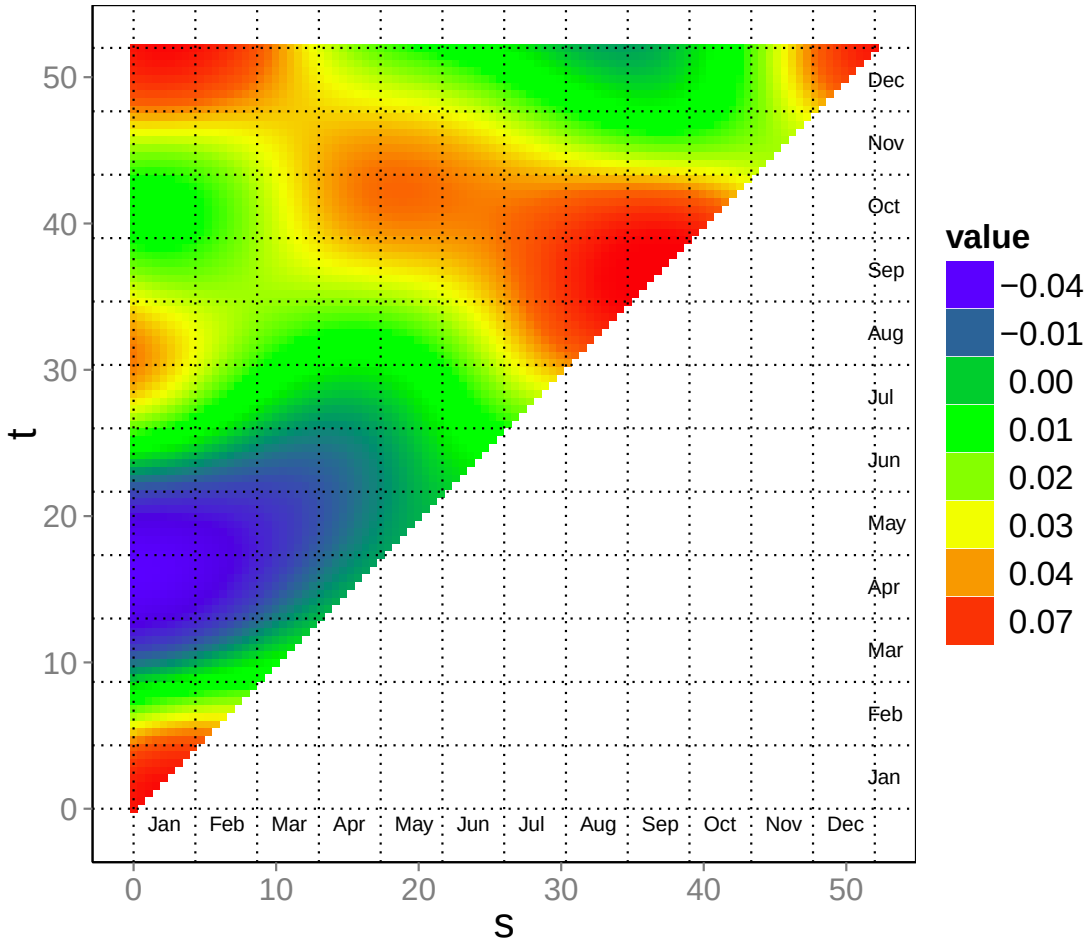

Figure 3.7: A heat map showing the estimated effect of temperature on forest fire, $\hat{\beta}_1(s,t)$.

1. Randomly shuffle the labels of the curves of forest fire. Keep the labels of the curves of temperature unchanged.

2. Pair each curve of forest fire and temperature according to their new labels and redo the historical functional linear regression to get a new estimate of $\beta_1(s,t)$, denoted as $\hat{\beta}_{1m}(s,t)$.

3. Evaluate $\hat{\beta}_{1m}(s,t)$ at (s_i,t_j), where $i=1,...,I, j=1,...,J$ and $s_i<t_j$. Let the total number of time points be T.

4. Repeat step 1, 2 and 3 M times. At each time point (s_i,t_j), find out which quantile $\hat{\beta}_1(s_i,t_j)$ is among $\hat{\beta}_{11}(s_i,t_j), \hat{\beta}_{12}(s_i,t_j),..., \hat{\beta}_{1M}(s_i,t_j)$.

5. $\hat{\beta}_1(s_i,t_j)$'s that fall below 5% or above 95% are considered significant.

We choose $I=100, J=100$, thus $T=5050$ and $M=1000$. Figure 3.8 shows the locations of these significant points.

Figure 3.7 and 3.8 shows that a higher temperature through the year always has positive effect on forest fire rate from August to November. This indicates that there exists positive historical effect of temperature on forest fire rate. We also find that the effect is most influential when s is close to t in January, August, September, October and December, which is a sign of strong concurrent effect of temperature. However, no significant association is found between temperature and forest fire rate from March to June.

Figure 3.9 and 3.10 shows the effect of precipitation on forest fire rate. There exists strong negative concurrent effect and moderate negative short-term effect up to approximately two months.

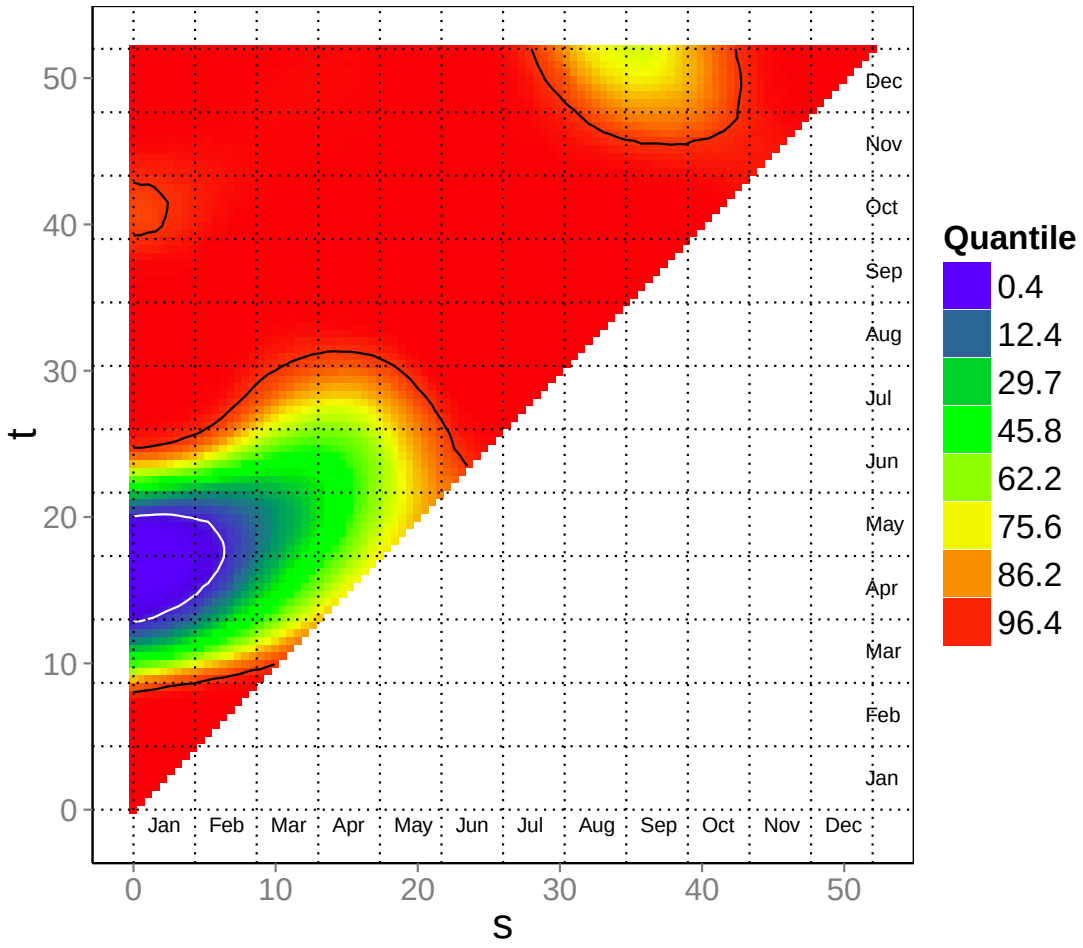

Figure 3.8: The quantile each $\hat{\beta}_1(s_i, t_j)$ corresponds to. White line and Black line show the contours of 5th and 95th quantiles, respectively.

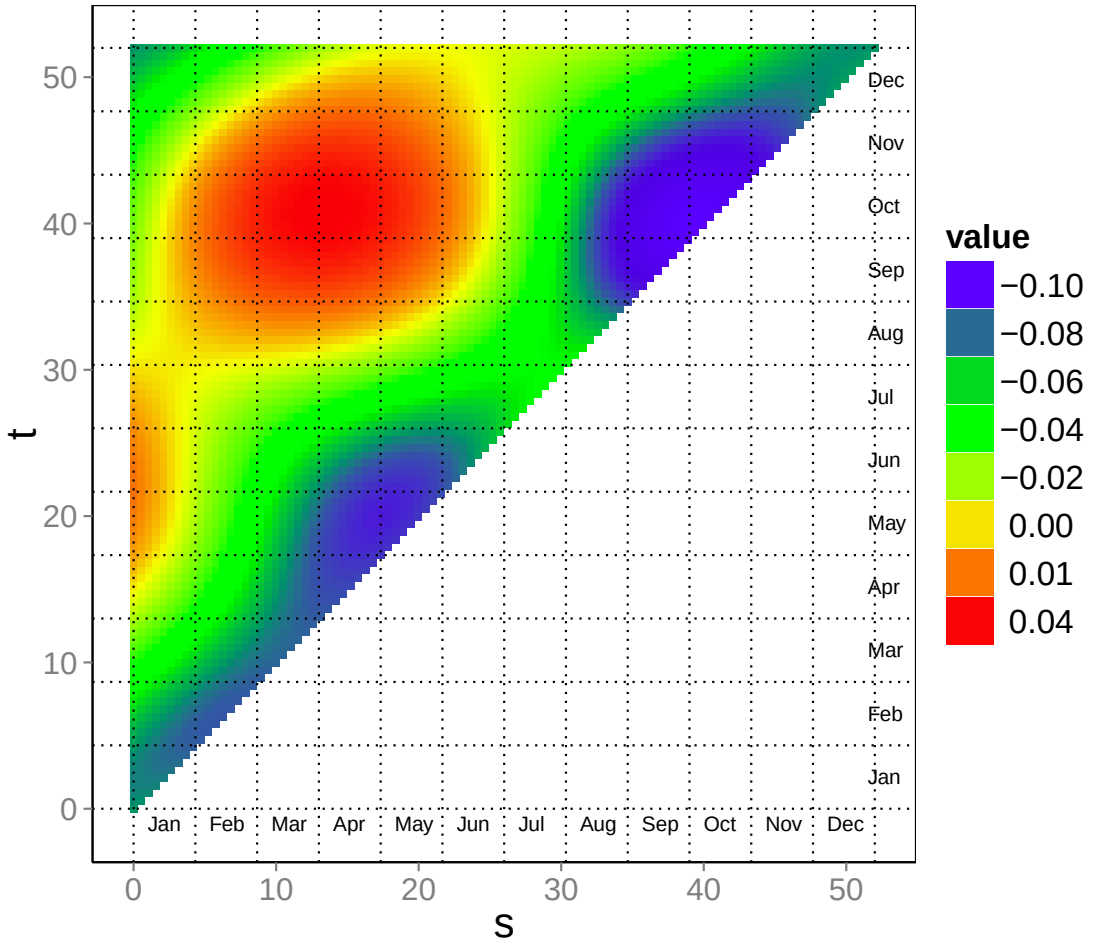

Figure 3.9: A heat map showing the estimated effect of precipitation on forest fire, $\hat{\beta}_2(s,t)$.

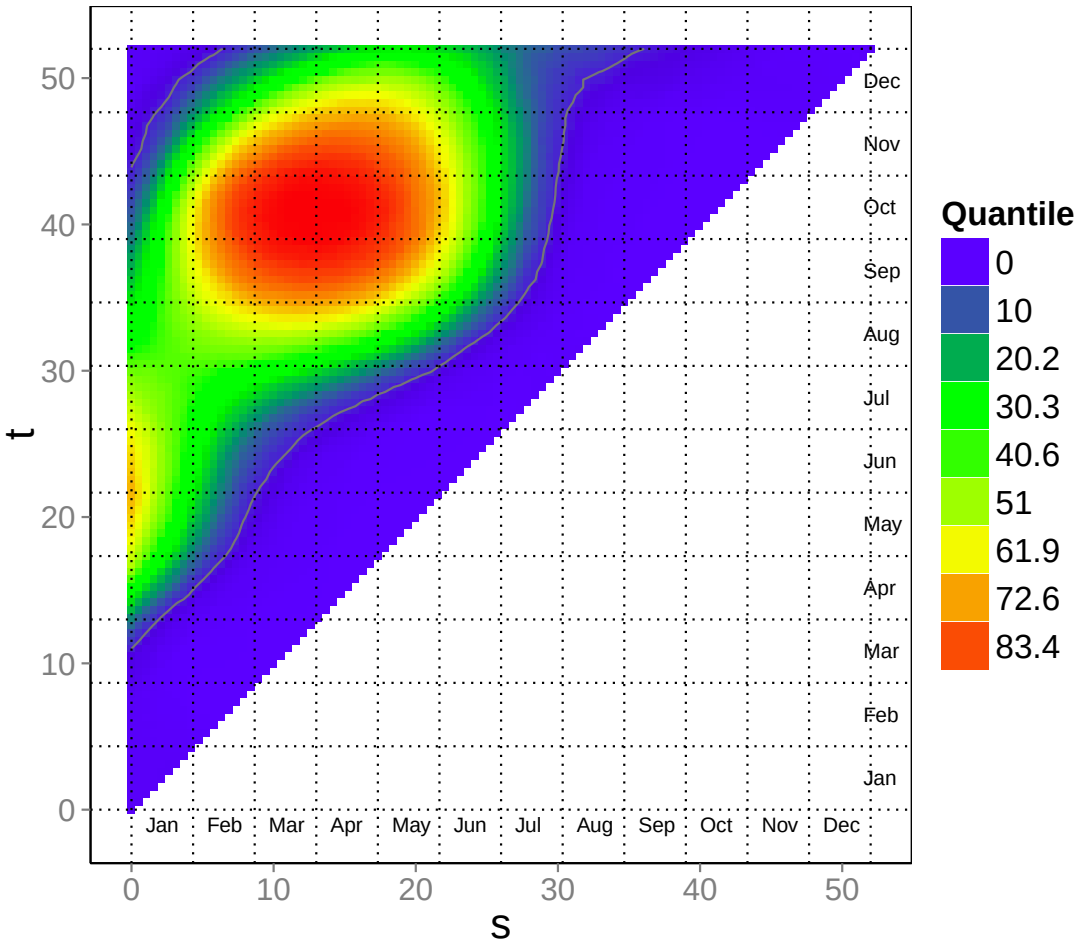

Figure 3.10: The quantile each $\hat{\beta}_2(s_i, t_j)$ corresponds to. White line and Black line show the contours of 5th and 95th quantiles, respectively.

Chapter 4

Conclusions

Functional data analysis is used to explore the fluctuation and variation of weekly forest fire rate in British Columbia among 63 years. The effect of weekly average temperature on forest fire rate is also investigated.

Functional principal component analysis shows that forest fire rate is most variable in summer and very stable during winter time. We also find out that forest fire rate is higher and less stable in the last decade than before. Year 2008 and 2009 have the most recorded forest fire among 63 years, and year 2004 and 2012 have the largest fire rate difference.

Historical functional linear model shows that there exists significant effect of temperature on forest fire rate. In particular, a higher temperature through the year always has positive effect on forest fire rate from August to November, which is a sign of historical effect of temperature. Compared to this historical effect, concurrent effect of temperature is stronger in certain months. The effect of precipitation is mostly negative and concurrent with a 2-month short-term effect detected.

Bibliography

[1] Increased risk of catastrophic wildfires: Global warming's wake-up call for the western united states, 2008.

[2] *Canadian National Fire Database-Agency Fire Data*. Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre, Edmonton, Alberta., 2013.

[3] B.D. Amiro, J.B. Todd, B.M. Wotton, K.A. Logan, M.D Flannigan, B.J. Stocks, J.A. Mason, D.L. Martell, and K.G. Hirsch. Direct carbon emissions from canadian forest fires, 1959 to 1999. *Canadian Journal of Forest Research*, 31:512–525, 2001.

[4] K.E. Atkinson. *An Introduction to Numerical Analysis*. Wiley, 1989.

[5] P.J. Burton, M.A. Parisien, J.A. Hicke, R.J. Hall, and J.T. Freeburn. Large fires as agents of ecological diversity in the north american boreal forest. *International Journal of Wildland Fire*, 16(6):754–767, 2008.

[6] N Malfait and J Ramsay. The historical functional linear model. *The Canadian Journal of Statistics*, 31:115–128, 2003.

[7] E. Mekis and L.A. Vincent. An overview of the second generation adjusted daily precipitation dataset for trend analysis in canada. *Atmosphere-Ocean*, 49(2):163–177, 2011.

[8] M.A. Parisien, V.S. Peters, Y. Wang, J.M. Little, E.M. Bosch, and B.J. Stocks. Spatial patterns of forest fires in canada 1980–1999. *International Journal of Wildland Fire*, 15:361–374, 2006.

[9] O Pechony and D.T Shindell. Driving forces of global wildfires over the past millennium and the forthcoming century. *Proceedings of the National Academy of Sciences*, 107:19167–19170, 2010.

[10] B.J. Stocks, J.A. Mason, J.B. Todd, E.M. Bosch, B.M. Wotton, B.D. Amiro, M.D. Flannigan, K.G. Hirsch, K.A. Logan, D.L. Martell, and W.R. Skinner. Large forest fires in canada, 1959–1997. *Journal of Geophysical Research*, 108:1–12, 2003.

[11] L.A Vincent. A technique for the identification of inhomogeneities in canadian temperature series. *Journal of Climate*, 11:1094–1104, 1998.

[12] L.A Vincent and D Gullett. Canadian historical and homogeneous temperature datasets for climate change analyses. *International Journal of Climatology*, 19:1375–1388, 1999.

[13] L.A Vincent, X.L Wang, E.J Milewska, H Wan, F Yang, and V Swail. A second generation of homogenized canadian monthly surface air temperature for climate trend analysis. *Journal of Geophysical Research*, 117(D18), 2012.

Appendix A

Code

```
##############################visualize raw data###############
library(maps)
library(mapdata)
#read in forest fire data
dat=read.csv("D:\\research\\forest_fire\\data\\NFDB_point_20131108
.txt",quote="\"",head=T,sep=",")
#keep only fires in BC
datbc=dat[dat[,2]=="BC",]

#read in weather station information where temperature data were col
lected
stations = read.csv("C:\\Users\\Administrator\\Dropbox\\ongoing for
est fire\\data\\Homog_daily_me
an_temp\\Homog_temperature_stations_v2014.csv", head=T)
#keep only stations within BC and have data between 1950 and 2012
stations=subset(stations,stations$Prov=="BC" & stations$beg.yr<=195
0 & stations$end.yr>=2012)

#fire and weather station distribution on map of BC
#BC map
map("worldHires","Canada",xlim=c(-140,-114),ylim=c(48,60),fill=T,co
l="gray90")
#fire
points(datbc$LONGITUDE,datbc$LATITUDE,pch=".")
#weather stations where temperature data are collected
points(stations$long..deg.,stations$lat..deg.,col=2,pch=19)
#add vancouver and victoria label to the map
text(x=-122,y=48.45,label="Victoria",cex=1,font=2,col=1)
text(x=-122.3,y=49.55,label="Vancouver",col=5,cex=1,font=2)
##################################################OK
```

```
######fire numbers in 52 weeks of a year############
#since there are some obs. without date (only year), "doy" will h
ave some NA in it. We replace it with 0
#so we can carry out calculation in the loop. Use the result fro
m the loop to generate week number
s to replace NA in doy.

#transfer date into days of a year (1-365)
doy=strptime(datbc$REP_DATE, "%Y-%m-%d %H:%M:%S")$yda
y+1
#replace "na" with "0"
doy[is.na(doy)]=0
fire_wk=matrix(nrow=63,ncol=52)

#the loop counts the number of fires occurred in each week
 of a year for 1950-2012.
#The result is a 63 by 52 matrix. Rows correspond to year
s, columns correspond to weeks.
for(i in 1:63){
for(j in 1:52){
fire_wk[i,j]=sum(as.numeric(datbc$YEAR_==(1949+i) &
(1+(j-1)*7)<=doy & doy<=(j*7)))
}
}
#add row names and column names
rownames(fire_wk,do.NULL=F)
rownames(fire_wk)=c(1950:2012)
colnames(fire_wk)=c(1:52)

#sample missing data
write.csv(as.data.frame(fire_wk),file="C:\\Users\\Admi
nistrator\\Desktop\\2.csv")
sample2012=sample(c(1:52),11,replace=T,prob=fire_
wk["2012",])
sample2011=sample(c(1:52),6,replace=T,prob=fire_
wk["2011",])
sample2010=sample(c(1:52),7,replace=T,prob=fire_
wk["2010",])
sample2009=sample(c(1:52),14,replace=T,prob=fire
_wk["2009",])
sample2007=sample(c(1:52),2,replace=T,prob=fire_
wk["2007",])
sample2006=sample(c(1:52),9,replace=T,prob=fire_
wk["2006",])
sample2005=sample(c(1:52),10,replace=T,prob=fire_wk["2005",])
sample2004=sample(c(1:52),66,replace=T,prob=fire_wk["2004",])
```

```
sample2003=sample(c(1:52),161,replace=T,prob=fire_wk["2003",])
sample2002=sample(c(1:52),40,replace=T,prob=fire_wk["2002",])
sample2001=sample(c(1:52),127,replace=T,prob=fire_wk["2001",])
sample2000=sample(c(1:52),191,replace=T,prob=fire_wk["2000",])
sample1999=sample(c(1:52),2,replace=T,prob=fire_wk["1999",])
sample1998=sample(c(1:52),18,replace=T,prob=fire_wk["1998",])
#replace NA in doy with sampled week numbers
doy=strptime(datbc$REP_DATE, "%Y-%m-%d %H:%M:%S")$yday+1
doy[datbc$YEAR_==2012 & datbc$MONTH_==0]=sample2012
doy[datbc$YEAR_==2011 & datbc$MONTH_==0]=sample2011
doy[datbc$YEAR_==2010 & datbc$MONTH_==0]=sample2010
doy[datbc$YEAR_==2009 & datbc$MONTH_==0]=sample2009
doy[datbc$YEAR_==2007 & datbc$MONTH_==0]=sample2007
doy[datbc$YEAR_==2006 & datbc$MONTH_==0]=sample2006
doy[datbc$YEAR_==2005 & datbc$MONTH_==0]=sample2005
doy[datbc$YEAR_==2004 & datbc$MONTH_==0]=sample2004
doy[datbc$YEAR_==2003 & datbc$MONTH_==0]=sample2003
doy[datbc$YEAR_==2002 & datbc$MONTH_==0]=sample2002
doy[datbc$YEAR_==2001 & datbc$MONTH_==0]=sample2001
doy[datbc$YEAR_==2000 & datbc$MONTH_==0]=sample2000
doy[datbc$YEAR_==1999 & datbc$MONTH_==0]=sample1999
doy[datbc$YEAR_==1998 & datbc$MONTH_==0]=sample1998

#final calculation of number of fires in each week
fire_wk=matrix(nrow=63,ncol=52)
for(i in 1:63){
for(j in 1:52){
fire_wk[i,j]=sum(as.numeric(datbc$YEAR_==(194
9+i) & (1+(j-1)*7)<=doy & doy<=(j*7)))
}
}
rownames(fire_wk,do.NULL=F)
rownames(fire_wk)=c(1950:2012)
colnames(fire_wk)=c(1:52)

#plot one curve for each year
plot(c(1:52),fire_wk[1,],type="l",ylim=c(0,1200),ma
in="fire numbers in each week by year",
xlab="week",ylab="number of fire")
for(i in 1:62){
lines(c(1:52),fire_wk[i+1,],col=(i+1))
}

#plot a selection of years using ggplot
###
library(ggplot2)
library(reshape2)
```

```
library(plyr)
fire_wk_plotdat=melt(fire_wk)
fire_wk_plotdat=subset(fire_wk_plotdat,Var1=
=1950|Var1==1960|Var1==1970|Var1==
1980|Var1==1990|Var1==2000|Var1==201
0|Var1==2011|Var1==2012)
fire_wk_plotdat=rename(fire_wk_plotdat,c("Va
r1"="year"))

plotobj=ggplot()
plotobj=plotobj+geom_point(data = fire_wk_
plotdat, aes(x = Var2, y = value),size=2)+fac
et_wrap(~year,ncol=3,scale="free")
plotobj=plotobj+ xlab("Week") + ylab("Num
ber of fire")+xlim(0,52)+ylim(0,400)
plotobj=plotobj+ theme(axis.text.y = eleme
nt_text(size=15),axis.text.x = element_text(si
ze=15),text= element_text(size=20),panel.g
rid.major = element_blank(), panel.grid.mino
r = element_blank(), panel.background = e
lement_blank(), axis.line = element_line(colou
r = "black"))
plotobj
####################################



#####Poisson process to smooth forest fire data#
#use exp(alpha(t))=lambda(t) since we have constr
aint "lambda(t)>0"

library(fda)
#define 29 fourier basis functions
weekbasis51 <- create.fourier.basis(c(0, 52), nb
asis=29, period=52)

#one set of Simpson's method variables
R=1000
g=52/(R-1)
U=(c(1:R)-1)*g
V=c(g/3,rep(c(4*g/3,2*g/3),(R-2)/2),g/3)

#another set of Simpon's method variables
Q=50
h=1/(Q-1)
W=c(h/3,rep(c(4*h/3,2*h/3),(Q-2)/2),h/3)
```

```
#define basis matrices that will be used later
basismat=eval.basis(U, weekbasis51)
basismat3=eval.basis(U, weekbasis51,3)
basismat2=eval.basis(U, weekbasis51,2)
basismat1=eval.basis(U, weekbasis51,1)
t=c(1:52)

#define an empty matrix to contain coefficien
ts of basis functions
Chat=matrix(ncol=63,nrow=29)

#a loop to calculate the coefficient of basis f
unctions
for(I in 1:63){
K=fire_wk[I,]
#take the log of fire data. If original fire coun
t=0 then let it be 0.001 so that log(fire)=-6.9
result <- smooth.basis(t, log(K+0.001), week
basis51)

#smoothing the forest fire data using normal
model.
#This is done so that we could get a better ini
tial value for simulated annealing method
#to find the penalized MLE of poisson process
Kfd  <- result$fd
KfdPar <- fdPar(Kfd, vec2Lfd(c(0,(2*pi/52)^2
,0), c(0, 52)), 1e2)
Kfd1 <- smooth.fd(Kfd, KfdPar)

#smoothing parameter is chosen to be 1e2
miu=1e2
#a function to calculate the penalizied likeli
hood of poisson process
#it is the sum of negative likelihood and pe
nalty. When rough, negative likelihood gets sm
aller, penalty gets larger.
ppmle=function(c){
p1=-exp(t(c)%*%t(basismat))%*%V
p2=c()
for(i in 1:52){
S=i-1+(c(1:Q)-1)*h
Phi=eval.basis(S, weekbasis51)
p2[i]=K[i]*log((exp(t(c)%*%t(Phi))%*%W))#
}
p2=sum(p2)
#p3=-sum(log(factorial(K)))
```

```
p4=exp(t(c)%*%t(basismat))^2*((t(c)%*%t(ba
sismat1))^3+3*t(c)%*%t(basismat1)*t(c)%*%t(b
asismat2)+t(c)%*%t(basismat3)+2*pi/52
*t(c)%*%t(basismat1))^2
p4=-miu*p4%*%V
-(p1+p2+p4)
}

#find minimum of sum of negative likeliho
od and penalty
chat=optim(matrix(Kfd1$coefs,ncol=1),
ppmle,method = "SANN")$par
Chat[,I]=chat
}

#store it as "Chat_1_63_b29_1e2"
Chat_1_63_b29_1e2=Chat

#plot of discrete data and smoothed lines
yhat=basismat%*%chat
plot(c(1:52),K,main="year 1951")
lines(U,exp(yhat))
plot(U,yhat)


par(mfrow=c(2,2))
for(i in 1:63){
K=fire_wk[i,]
chat=Chat_1_63_b29_1e2[,i]
yhat=basismat%*%chat
plot(c(1:52),K,xlab="week",ylab="number o
f fire",main=seq(1950,2012,1)[i],ylim=c(0,
600))
lines(U,exp(yhat))
}

#add smoothed curve to original discrete d
ata plot using ggplot
ypredicted=exp(basismat%*%Chat_1_63_b29
_1e2)
rownames(ypredicted)=seq(0,52,length=1000)
colnames(ypredicted)=1950:2012
smoothed_fire_wk_plotdat=melt(ypredicted)
smoothed_fire_wk_plotdat=subset(smoothed
_fire_wk_plotdat,Var2==1950|Var2==1960
|Var2==1970|Var2==1980|Var2==1990|Va
r2==2000|Var2==2010|Var2==2011|Var2=
```

```
=2012)
smoothed_fire_wk_plotdat=rename(smooth
ed_fire_wk_plotdat,c("Var2"="year"))

plotobj=plotobj+geom_line(data=smoothe
d_fire_wk_plotdat, aes(Var1, value))
plotobj
###################ok


#############FPCA using curves smooth
ed from Poisson process#######
library(fda)
newyhat=basismat%*%Chat_1_63_b29_1e2
newy=exp(newyhat)
#use 101 basis functions to interpolate poin
ts
weekbasis101=create.fourier.basis(c(0, 52)
, nbasis=101, period=52)
newyfd <- smooth.basis(argvals=seq(0,52,l
ength=1000), y=newy,weekbasis101, fdnam
e
s=list("week", "year", "count"))$fd
new_weekfirepcaobj=pca.fd(center.fd(newyf
d), nharm = 4,centerfns = TRUE)
weekbasis29 <- create.fourier.basis(c(0, 52)
, nbasis=29, period=52)
new_weekfirefd=fd(Chat_1_63_b29_1e2, we
ekbasis29,fdnames=list("week", "year", "co
unt"))
new_harmaccelLfd <- vec2Lfd(c(0,(2*pi/5
2)^2,0), c(0, 52))
#harmonic parameter, used to define func
tional version of our month fire data
new_harmfdPar     <- fdPar(weekbasis29,
new_harmaccelLfd, lambda=1e2)#will try differ
ent lambda below
new_weekfirepcaobj=pca.fd(center.fd(ne
w_weekfirefd), nharm = 4,new_harmfdPa
r,cente
rfns = TRUE)

#plotting results
op <- par(mfrow=c(2,2))
par(op)

#plot mean curve+-pc
```

```
plot.pca.fd(new_weekfirepcaobj, cex.mai
n=0.9)
#plot PC
plot(new_weekfirepcaobj$harmonics)
#plot pc in seperate graphs
par(mfrow=c(2,2))
par(pty="s")
plot(new_weekfirepcaobj$harmonics[1],x
lab="week",main=paste("PC",1,sep=""),yl
im=c
(-.4,.4))
plot(new_weekfirepcaobj$harmonics[2],x
lab="week",main=paste("PC",2,sep=""),ylim=
c(-.4,.4))
plot(new_weekfirepcaobj$harmonics[3],
xlab="week",main=paste("PC",3,sep=""),ylim=
c(-.4,.4))
plot(new_weekfirepcaobj$harmonics[4],
xlab="week",main=paste("PC",4,sep=""),ylim=
c(-.4,.4))
#ggplot for pc
pcvalue=eval.fd(seq(0,52,length=1000),
new_weekfirepcaobj$harmonics)
rownames(pcvalue)=seq(0,52,length=10
00)
pcvalue_plotdat=melt(pcvalue)
plotobj2=ggplot(data=pcvalue_plotdat,ae
s(x=Var1,y=value))+geom_line()+facet_w
rap(~Var2,ncol=2,scale="free")
plotobj2=plotobj2+theme(axis.text.y =
element_text(size=15),axis.text.x = elemen
t_text(size=15),text= element_text(size=
20),panel.grid.major = element_blank(), pa
nel.grid.minor = element_blank(), panel.
background = element_blank(), axis.lin
e = element_line(colour = "black"))
plotobj2=plotobj2+ylim(-0.35,0.4)+xlab
("week")+ylab("FPCs")
plotobj2=plotobj2+geom_hline(yinterce
pt=0, linetype="dotted")
plotobj2=plotobj2+geom_vline(xinterce
pt=seq(0,52,length=13),linetype="dot
ted")+annotate("text",x=seq(0,52,length
=13)[1:12]+0.4,y=-0.35,label=c("Jan","F
eb","Mar","Apr","May","Jun","Jul","Aug","Se
p","Oct","Nov","Dec"),hjust=0,size=3)
plotobj2
```

```
#scores 1 vs scores 2
plotscores(new_weekfirepcaobj,scores=c
(1,2),pch=".")
text(new_weekfirepcaobj$scores[,1],new_
weekfirepcaobj$scores[,2],labels=c(c(
50:99),"00","01","02","03","04","05","06"
,"07","08","09","10","11","12"))
#ggplot of scores 1 vs scores 2
score12=new_weekfirepcaobj$scores[,1
:2]
score12=cbind(1950:2012,score12)
colnames(score12)=c("Year","FPC1_score
","FPC2_score")
score12=data.frame(score12)
score12$Year=substring(score12$Ye
ar,3)
score12=cbind((c(rep(0,54),rep(1,9))),
score12)
colnames(score12)[1]="grp"
plotobj5=ggplot(data=score12)+geo
m_point(aes(x=FPC1_score,y=FPC2_score)
,size=2,shape=16)
plotobj5=plotobj5+geom_point(aes(
x=FPC1_score[grp==1],y=FPC2_score[grp

==1]),size=3,shape=15)
plotobj5=plotobj5+theme(axis.text.y
 = element_text(size=15),axis.text.x = el
ement_text(size=15),text= element_
text(size=20),panel.grid.major = element
_blank(), panel.grid.minor = element
_blank(), panel.background = element_blan
k(), axis.line = element_line(colour
= "black"),panel.border = element_r
ect(colour = "black", fill=NA))
plotobj5=plotobj5+labs(x="FPC1 sc
ore",y="FPC2 score")
plotobj5=plotobj5+geom_text(aes(
FPC1_score,FPC2_score,label=Year), size=
6,hjust=-0.2)
plotobj5=plotobj5+geom_vline(xinterc
ept=0,linetype="dotted")+geom_hli
ne(yintercept=0,linetype="dotted")
plotobj5
#plot scores: scatterplot
plot(rep(0,63),new_weekfirepcaobj$sc
```

```
ores[,1],xlab="",ylab="PC1 scores
",ylim=c(-400,500))

plot(rep(0,63),new_weekfirepcaobj$
scores[,2],xlab="",ylab="PC2 scores
",ylim=c(-400,500))
plot(rep(0,63),new_weekfirepcaobj$
scores[,3],xlab="",ylab="PC3 score
s",ylim=c(-400,500))
plot(rep(0,63),new_weekfirepcaobj$
scores[,4],xlab="",ylab="PC4 scor
es",ylim=c(-400,500))
library(ggplot2)
library(reshape2)
plotdat=melt(as.matrix(new_weekfire
pcaobj$scores))
plotobj3=ggplot(data = plotdat, aes
(x = Var2, y = value))+ geom_poi
nt()+ xlab("PC") + ylab("value")
#plot scores: boxplot
boxplot(new_weekfirepcaobj$scores,
use.cols = TRUE,xlab="PC",ylab=
"scores")
library(ggplot2)
library(reshape2)
plotdat=melt(as.matrix(new_weekfi
repcaobj$scores))
plotobj4=ggplot(data = plotdat, a
es( x=factor(Var2),y = value))+ ge
om_boxplot()+ xlab("PC") + ylab("v
alue")
############################ok

#########temperature data######################

#read in station information
stations = read.csv("C:\\Users\\Admini
strator\\Dropbox\\ongoing forest fire\\
data\\Homo
g_daily_mean_temp\\Homog_temperatu
re_stations_v2014.csv", head=T)
stations=subset(stations,stations$Prov=
="BC" & stations$beg.yr<=1950 & station
s$end.yr>=2012)
stations

#read in file names
```

```
filelist=list.files(path="C:\\Users\\Admi
nistrator\\Dropbox\\ongoing forest fire\
\data\\Ho
mog_daily_mean_temp\\Homog_daily_m
ean_temp_v2014")

#read in temperature data
readtempdata=function(id){
read.table(paste("C:\\Users\\Administra
tor\\Dropbox\\ongoing forest fire\\data
\\Homog
_daily_mean_temp\\daily_temp\\",grep(i
d,filelist,value=T),sep=""),head=F)
}
tempdata=lapply(stations$stnid,readte
mpdata)

#add column names
addcolnames=function(x){
x=as.data.frame(x)
colnames(x)=c("yr","mo",paste("day",1:3
1,sep=""))
return(x)
}
tempdata_=lapply(tempdata,addcoln
ames)

#clean the data

samplebyday=function(x){
x=as.matrix(x)
if(length(x[x!="M"])!=0){
for(i in 1:length(x)){
if(x[i]=="M" & any(x[1:i]!="M")){
x[i]=sample(x[1:i][x[1:i]!="M"],1,rep
lace=T)
}
}
return(x)
}
else{return(x)}
}

cleandata=function(x){
#delete a and e
x_=as.matrix(as.data.frame(x))
x_[substr(x_, nchar(x_), nchar(x_))=
```

```
="a" | substr(x_, nchar(x_), nchar(x_
))=="E"]=substr(x
_, 1, nchar(x_)-1)[substr(x_, nchar(x_
), nchar(x_))=="a" | substr(x_, nchar(
x_), nchar(x_))=="E"]
#replace missing values with a rando
m sample from corresponding mont
h and day
x_=as.data.frame(x_)
month1=x_[as.numeric(as.characte
r(x_$mo))==1,]
nomdata=as.data.frame(lapply(mo
nth1,samplebyday))
for(j in 2:12){
monthly=x_[as.numeric(as.charac
ter(x_$mo))==j,]
nomlist=lapply(monthly,sampleb
yday)
nomdata=rbind(nomdata,as.data
.frame(nomlist))
}

#some dates do not exist. Set th
em to NA
#nomdata[as.numeric(as.charact
er(nomdata$mo))==2,]$day29=
"NA"
#nomdata[as.numeric(as.charac
ter(nomdata$mo))==2,]$day30
="NA"
#nomdata[as.numeric(as.chara
cter(nomdata$mo))==2,]$day
31="NA"
#nomdata[as.numeric(as.charac
ter(nomdata$mo))==4,]$day30="NA"
#nomdata[as.numeric(as.charac
ter(nomdata$mo))==6,]$day30="NA"
#nomdata[as.numeric(as.charac
ter(nomdata$mo))==9,]$day30="NA"
#nomdata[as.numeric(as.chara
cter(nomdata$mo))==11,]$day30="NA"
return(nomdata)
}

cleanedtempdata=lapply(tem
pdata_,cleandata)
```

```
#combine weather stations toge
ther
cleanedtempdata_=cleanedtem
pdata[[1]]
for(i in 2:42){
cleanedtempdata_=rbind(cleaned
tempdata_,cleanedtempdata[[i]])
}

#keep only year 1950 to 2012
cleanedtempdata__=subset(clean
edtempdata_,as.numeric(as.charac
ter(cleanedtempdat
a_$yr))>=1950 & as.numeric(as.c
haracter(cleanedtempdata_$yr))<=2012)

#some stations has incomplete 20
12 data. find out which ones
incomplete2012=function(x){
sum(as.numeric(as.numeric(as.char
acter(x$yr))==2012))
}
incomp2012=lapply(cleanedtempd

ata,incomplete2012)#station 10 has 11. station 15 ha
s 11. station 24 has 5. station 26 has 5.
stations$stnid[10];stations$stnid[1
5];stations$stnid[24];stations$stnid[26]

#change all "M" into NA
cleanedtempdata__[cleanedtempdata__=="M"]=NA

#change all numbers into numeric
chartonum=function(x){
y=as.numeric(as.character(x))
return(y)
}
tempdata=as.data.frame(do.call(cbin
d,lapply(cleanedtempdata__,chartonum)))

#change all Feb 29th data to NA
tempdata$day29[tempdata$mo==2]=NA

#sort data by year and month
tempdata=tempdata[with(tempdata, or
der(yr,mo )), ]
```

```
#check no NA
any(is.na(tempdata[,1:32][tempdata$mo==4,]))

#calculate daily average data
avgdata=matrix(nrow=63*12,ncol=33)
for (i in 1:63){
for(j in 1:12){
sbst=tempdata[tempdata$mo==j
& tempdata$yr==(1949+i),]
avgdata[12*(i-1)+j,]=do.call(cbind,l
apply(sbst,mean))
}
}

#change daily data to weekly data
avgdata=as.data.frame(avgdata)
avgdata_wk=matrix(nrow=63,ncol=52)
for(i in 1:63){
alldata=subset(avgdata,avgdata[,1]==(1949+i))
jan=as.matrix(subset(alldata,alldata[,2]==1)[,3:33])
feb=as.matrix(subset(alldata,alldata[,2]==2)[,3:33])
mar=as.matrix(subset(alldata,alldata[,2]==3)[,3:33])
apr=as.matrix(subset(alldata,alldata[,2]==4)[,3:33])
may=as.matrix(subset(alldata,alldata[,2]==5)[,3:33])
jun=as.matrix(subset(alldata,alldata[,2]==6)[,3:33])
jul=as.matrix(subset(alldata,alldata[,2]==7)[,3:33])
aug=as.matrix(subset(alldata,alldata[,2]==8)[,3:33])
sep=as.matrix(subset(alldata,alldata[,2]==9)[,3:33])
oct=as.matrix(subset(alldata,alldata[,2]==10)[,3:33])
nov=as.matrix(subset(alldata,alldata[,2]==11)[,3:33])
dec=as.matrix(subset(alldata,alldata[,2]==12)[,3:33])
avgdata_wk[i,1]=mean(jan[1:7])
avgdata_wk[i,2]=mean(jan[8:14])
avgdata_wk[i,3]=mean(jan[15:21])
avgdata_wk[i,4]=mean(jan[22:28])
avgdata_wk[i,5]=mean(c(jan[29:31],feb[1:4]))
avgdata_wk[i,6]=mean(feb[5:11])
avgdata_wk[i,7]=mean(feb[12:18])
avgdata_wk[i,8]=mean(feb[19:25])
avgdata_wk[i,9]=mean(c(feb[26:28],mar[1:4]))
avgdata_wk[i,10]=mean(mar[5:11])
avgdata_wk[i,11]=mean(mar[12:18])
avgdata_wk[i,12]=mean(mar[19:25])
avgdata_wk[i,13]=mean(c(mar[26:31],apr[1]))
avgdata_wk[i,14]=mean(apr[2:8])
avgdata_wk[i,15]=mean(apr[9:15])
avgdata_wk[i,16]=mean(apr[16:22])
```

```
avgdata_wk[i,17]=mean(apr[23:29])
avgdata_wk[i,18]=mean(c(apr[30],may[1:6]))
avgdata_wk[i,19]=mean(may[7:13])
avgdata_wk[i,20]=mean(may[14:20])
avgdata_wk[i,21]=mean(may[21:27])
avgdata_wk[i,22]=mean(c(may[28:31],jun[1:3]))
avgdata_wk[i,23]=mean(jun[4:10])
avgdata_wk[i,24]=mean(jun[11:17])
avgdata_wk[i,25]=mean(jun[18:24])
avgdata_wk[i,26]=mean(c(jun[25:30],jul[1]))
avgdata_wk[i,27]=mean(jul[2:8])
avgdata_wk[i,28]=mean(jul[9:15])
avgdata_wk[i,29]=mean(jul[16:22])
avgdata_wk[i,30]=mean(jul[23:29])
avgdata_wk[i,31]=mean(c(jul[30:31],aug[1:5]))
avgdata_wk[i,32]=mean(aug[6:12])
avgdata_wk[i,33]=mean(aug[13:19])
avgdata_wk[i,34]=mean(aug[20:26])
avgdata_wk[i,35]=mean(c(aug[27:31],sep[1:2]))
avgdata_wk[i,36]=mean(sep[3:9])
avgdata_wk[i,37]=mean(sep[10:16])
avgdata_wk[i,38]=mean(sep[17:23])
avgdata_wk[i,39]=mean(sep[24:30])
avgdata_wk[i,40]=mean(oct[1:7])
avgdata_wk[i,41]=mean(oct[8:14])
avgdata_wk[i,42]=mean(oct[15:21])
avgdata_wk[i,43]=mean(oct[22:28])
avgdata_wk[i,44]=mean(c(oct[29:31],nov[1:4]))
avgdata_wk[i,45]=mean(nov[5:11])
avgdata_wk[i,46]=mean(nov[12:18])
avgdata_wk[i,47]=mean(nov[19:25])
avgdata_wk[i,48]=mean(c(nov[26:30],dec[1:2]))
avgdata_wk[i,49]=mean(dec[3:9])
avgdata_wk[i,50]=mean(dec[10:16])
avgdata_wk[i,51]=mean(dec[17:23])
avgdata_wk[i,52]=mean(dec[24:30])
}

#store data as "tempdata_wk"
tempdata_wk=avgdata_wk

#smooth the temp data
weekbasis29 <- create.fourier.basis(c
(0, 52), nbasis=29, period=52)
harmaccelLfd <- vec2Lfd(c(0,(2*pi/52)^2
,0), c(0, 52))
harmfdPar     <- fdPar(weekbasis29, harma
```

```
ccelLfd, lambda=1e2)
weektempfd <- smooth.basis(argvals=c(1:5
2), y=t(tempdata_wk),fdParobj=harmfdPar,
fdnames=list("week", "year", "temperature"))
$fd

#plot temp data and smoothed curve using ggplot
library(ggplot2)
library(reshape2)
library(plyr)

yhattemp=eval.fd(seq(0,52,length=1000),weektempfd)
rownames(yhattemp)=seq(0,52,length=1000)
colnames(yhattemp)=1950:2012
yhattemp_plotdat=melt(yhattemp)
yhattemp_plotdat=rename(yhattemp_plotdat,c
("Var2"="year"))
yhattemp_plotdat=subset(yhattemp_plotdat,y
ear==1950|year==1960|year==1970|
year==1980|year==1990|year==2000|year=
=2010|year==2011|year==2012)

tempdata_wk_=tempdata_wk
rownames(tempdata_wk_)=1950:2012
colnames(tempdata_wk_)=1:52
tempdata_wk_plotdat=melt(tempdata_wk_)
tempdata_wk_plotdat=rename(tempdata_wk_p
lotdat,c("Var1"="year"))
tempdata_wk_plotdat=subset(tempdata_wk_pl
otdat,year==1950|year==1960|year==
1970|year==1980|year==1990|year==2000|ye
ar==2010|year==2011|year==2012)

plotobj1=ggplot()+geom_line(data=yhattemp_
plotdat, aes(Var1, value))
plotobj1=plotobj1+geom_point(data = tempda
ta_wk_plotdat, aes(x = Var2, y = value
),size=2)+facet_wrap(~year,ncol=3,scale="free")
plotobj1=plotobj1+ xlab("Week") + ylab("Mean
 weekly temperature (??C)")+xlim(0,52)
+ylim(-15,25)
plotobj1=plotobj1+ theme(axis.text.y = eleme
nt_text(size=15),axis.text.x = elemen
t_text(size=15),text= element_text(size=20),p
anel.grid.major = element_blank(), p
anel.grid.minor = element_blank(), panel.backg
round = element_blank(), axis.line
```

```
= element_line(colour = "black"))
plotobj1
########################################ok



#####################
#historical functional linear model
#use log fire count
#include intercept function, but no penalty, use only 19 basis
#beta(s,t) use 29 fourier basis by 29 fourier basis, both with
HAO penalty lambda 1e5
#plot exp(fitted)
###################

#historical functional linear model
weekbasis29=create.fourier.basis(c(0, 52), nbasis=29, period=52)

#empty matrix
Psimatrix=matrix(nrow=50*63,ncol=435)

#evaluate the integral at 50 points
tpoints=seq(0,52,length=50)

psitmat=eval.basis(tpoints,weekbasis29)
all1vec=rep(1,29*29)
psimatrix=matrix(nrow=50,ncol=435)
for(i in 1:28){
all1vec[(29*i+1):(29*i+i)]=0
}
for(i in 1:63){
for(q in 1:50){
tpoint=tpoints[q]#the given t_0 to t_Q
targ=seq(0,tpoint,length=50)#t used to estimate integral in each psi
gap=tpoint/49
psismat=eval.basis(targ,weekbasis29)
xvalues=eval.fd(targ,weektempfd)
xvalue=xvalues[,i]
xcbind=replicate(29,xvalue)
aa=replicate(29,colSums(xcbind*psismat)*gap)
bb=as.vector(aa)
cc=rep(psitmat[q,],each=29)
psimatrix[q,]=t(bb*cc)[all1vec==1]
}
Psimatrix[(1+(i-1)*50):(50+(i-1)*50),]=psimatrix
}
```

```
#y values
Y=as.vector(eval.fd(tpoints,logfire))

#build alpha(t) matrix, use 19 basis functions
weekbasis19=create.fourier.basis(c(0, 52), nbasis=19, period=52)
alphamat=do.call(rbind, replicate(63, eval.basis(tpoints,w
eekbasis19), simplify=FALSE))


Y2Y2T=yyt(Y2[1,])
for(i in 2:10000){
Y2Y2T=Y2Y2T+yyt(Y2[i,])
}#need to add up the matrices multiply by gap=(52/99)^2
lambda1=1e5
lambda2=1e5
pen1=Y1Y1T*(52/99)^2*lambda1
pen2=Y2Y2T*(52/99)^2*lambda2

#solve for coefficients
Psimat=cbind(alphamat,Psimatrix)
coefab=solve(t(Psimat)%*%Psimat+pen1+pe
n2)%*%t(Psimat)%*%Y
coefa=coefab[1:19]
coefb=coefab[20:454]

#alpha function
alphafd=fd(coefa,weekbasis19)

#plot alpha function
plot(seq(0,52,length=1000),exp(eval.fd(seq(0,5
2,length=1000
),alphafd)),type="l")

#try to use ggplot to plot y(t),yhat(t) and origi
nal discrete data points
#these are copied from above
plotobj=ggplot()+geom_line(data=smoothed_fire_wk_p
lotda
t, aes(Var1, value))
plotobj=plotobj+geom_point(data = fire_wk_plotdat,
 aes(x = Var2, y = value),size=2)
plotobj=plotobj+ xlab("Week") + ylab("Number
of fire")+xlim(0,52)+ylim(0,400)
plotobj=plotobj+ theme(axis.text.y = element_te
xt(size=15),axis.text.x = element_
text(size=15),text= element_text(size=20),panel.
grid.major = element_blank(), pa
```

```
nel.grid.minor = element_blank(), panel.backgrou
nd = element_blank(), axis.line =
 element_line(colour = "black"))

#we add the yhat(t) to the plot above
yhat_hlm=Psimatrix%*%coefb
yhatdat=exp(rep(eval.fd(tpoints,alphafd),63)+yhat_hlm)
yhatdat=cbind(rep(1950:2012,each=50),tpoints,yhatdat)
colnames(yhatdat)=c("year","week","value")
yhatdat=as.data.frame(yhatdat)
yhatdat=subset(yhatdat,year==1950|year==1960|
year==1970|year==1980|year=
=1990|year==2000|year==2010|year==2011|year==2012)

plotobj=plotobj+geom_line(data=yhatdat,aes(x=w
eek,y=value),linetype = 2)+face
t_wrap(~year,ncol=3,scale="free")
plotobj

#plot triangular support of s,t
plotd=data.frame(cbind(c(0,0,52),c(0,52,52)))
pobj=ggplot()+geom_polygon(data=plotd,aes(x=
X1,y=X2),fill="grey")
pobj=pobj+ theme(axis.text.y = element_text(siz
e=15),axis.text.x = element_tex
t(size=15),text= element_text(size=20),panel.gri
d.major = element_blank(), pan
el.grid.minor = element_blank(), panel.backgroun
d = element_blank(), axis.line
= element_line(colour = "black"),panel.border =
element_rect(colour = "black",
fill=NA),legend.position="none")
pobj=pobj+xlab("s")+ylab("t")
pobj
all1vec_=rep(1,100)
for(i in 1:9){
all1vec_[(10*(i-1)+1+i):(10*i)]=0
}
basisfnc=data.frame(cbind(1:55,rep(seq(0,52,leng
th=10),10)[all1vec_==1],rep(
seq(0,52,length=10),each=10)[all1vec_==1]))
colnames(basisfnc)=c("b","s","t")
pobj=pobj+geom_point(data=basisfnc,aes(x=s,y=t))
pobj

#plot beta(s,t), heat map
sdat=eval.basis(seq(0,52,length=100),weekbasis29)
```

```
tdat=sdat
all1vec2=rep(1,10000)
for(i in 1:99){
all1vec2[(100*i+1):(100*i+i)]=0
}
bmat=kronecker(sdat,tdat)
colnames(bmat)=1:841
rownames(bmat)=1:10000
bmat=bmat[all1vec2==1,all1vec==1]
tpts=seq(0,52,length=100)

scoord=rep(0,100)
for(i in 2:100){
tpt=tpts[i]
scoord=c(scoord,rep(tpt,101-i))
}

tcoord=tpts
for(i in 2:100){
tpt=tpts[i]
tcoord=c(tcoord,tpts[i:100])
}

betastvalue=bmat%*%coefb




####################################ok
```