

# P-MHD: Multivariate Methods

## Project Assignment

1 March 2019

```
# First set your working directory (with data file CanadianWeather.rda),
# and install the required packages
setwd("~/dropbox/education/PMHD/Project1819/")
#install.packages("ldr")
library(ldr)
```

```
## Loading required package: GrassmannOptim
```

```
## Loading required package: Matrix
```

This text contains an introduction to the data and to **Functional Data Analysis**. The details of the assignment can be found at the end of this text.

## 1. Introduction

### 1.1. Data

We have data on average daily rainfall (mm/day) for the 365 days in the year and for 35 Canadian cities.

First we read the data

```
load("CanadianWeather.rda")

da<-CanadianWeather[[1]]
da<-da[, "Precipitation.mm"] # precipitation data
head(da)
```

```
##      St. Johns Halifax Sydney Yarmouth Charlottvl Fredericton Scheffervll
## jan01      5.2      6.0      5.3      5.6      4.6      4.0      1.1
## jan02      5.8      5.3      5.2      3.7      4.4      3.2      1.3
## jan03      3.9      2.6      2.1      2.8      2.3      3.3      1.2
## jan04      4.3      5.3      5.0      5.3      4.8      3.3      1.3
## jan05      6.2      6.0      7.3      3.8      5.1      2.7      1.0
## jan06      3.4      2.1      2.2      2.4      1.5      0.8      1.3
##      Arvida Bagottville Quebec Sherbrooke Montreal Ottawa Toronto London
## jan01      2.6      3.0      4.1      2.9      2.9      2.5      1.8      2.4
## jan02      1.2      1.8      2.3      2.9      1.2      1.1      0.9      1.4
## jan03      2.1      1.3      2.6      1.9      1.4      1.3      0.9      1.8
## jan04      2.3      2.5      4.3      2.9      3.6      3.1      1.5      2.9
## jan05      1.7      2.1      2.3      2.1      1.6      1.3      0.8      1.1
## jan06      2.0      1.6      1.5      0.8      1.1      1.3      1.0      1.4
##      Thunder Bay Winnipeg The Pas Churchill Regina Pr. Albert
## jan01      0.7      0.5      0.5      0.5      0.2      0.1
## jan02      1.9      0.6      0.9      0.6      0.3      0.9
## jan03      0.8      0.3      0.7      0.5      0.6      0.6
## jan04      0.3      0.5      0.5      0.4      0.3      0.3
## jan05      0.8      0.4      0.2      0.4      0.8      0.2
## jan06      1.7      0.7      0.9      0.2      0.5      0.3
```

```
##      Uranium City Edmonton Calgary Kamloops Vancouver Victoria Pr. George
## jan01      0.3      0.4      0.3      0.6      5.5      5.3      2.2
## jan02      0.4      0.8      0.1      0.4      6.6      5.2      1.9
## jan03      1.3      1.1      0.3      1.2      6.8      5.4      1.9
## jan04      0.6      1.1      0.6      1.3      5.1      4.5      1.8
## jan05      0.8      1.0      1.0      1.2      3.8      4.6      1.1
## jan06      1.1      0.8      0.2      0.5      2.5      2.6      1.2
##      Pr. Rupert Whitehorse Dawson Yellowknife Iqaluit Inuvik Resolute
## jan01      6.0      0.5      0.9      0.6      1.1      0.8      0.1
## jan02      5.0      0.8      0.6      0.7      0.9      0.9      0.1
## jan03      6.7      1.1      0.8      0.3      0.8      0.8      0.0
## jan04      7.1      0.2      0.8      0.5      0.7      0.4      0.2
## jan05      6.1      0.6      1.0      0.7      0.9      0.8      0.2
## jan06      8.1      0.7      1.0      0.5      0.2      0.4      0.2
```

Here is a list of the 35 cities

```
colnames(da)
```

```
## [1] "St. Johns"      "Halifax"      "Sydney"      "Yarmouth"
## [5] "Charlottvl"    "Fredericton"  "Scheffervll" "Arvida"
## [9] "Bagottville"   "Quebec"      "Sherbrooke"  "Montreal"
## [13] "Ottawa"        "Toronto"     "London"      "Thunder Bay"
## [17] "Winnipeg"      "The Pas"     "Churchill"   "Regina"
## [21] "Pr. Albert"    "Uranium City" "Edmonton"    "Calgary"
## [25] "Kamloops"      "Vancouver"   "Victoria"    "Pr. George"
## [29] "Pr. Rupert"    "Whitehorse"  "Dawson"      "Yellowknife"
## [33] "Iqaluit"       "Inuvik"      "Resolute"
```

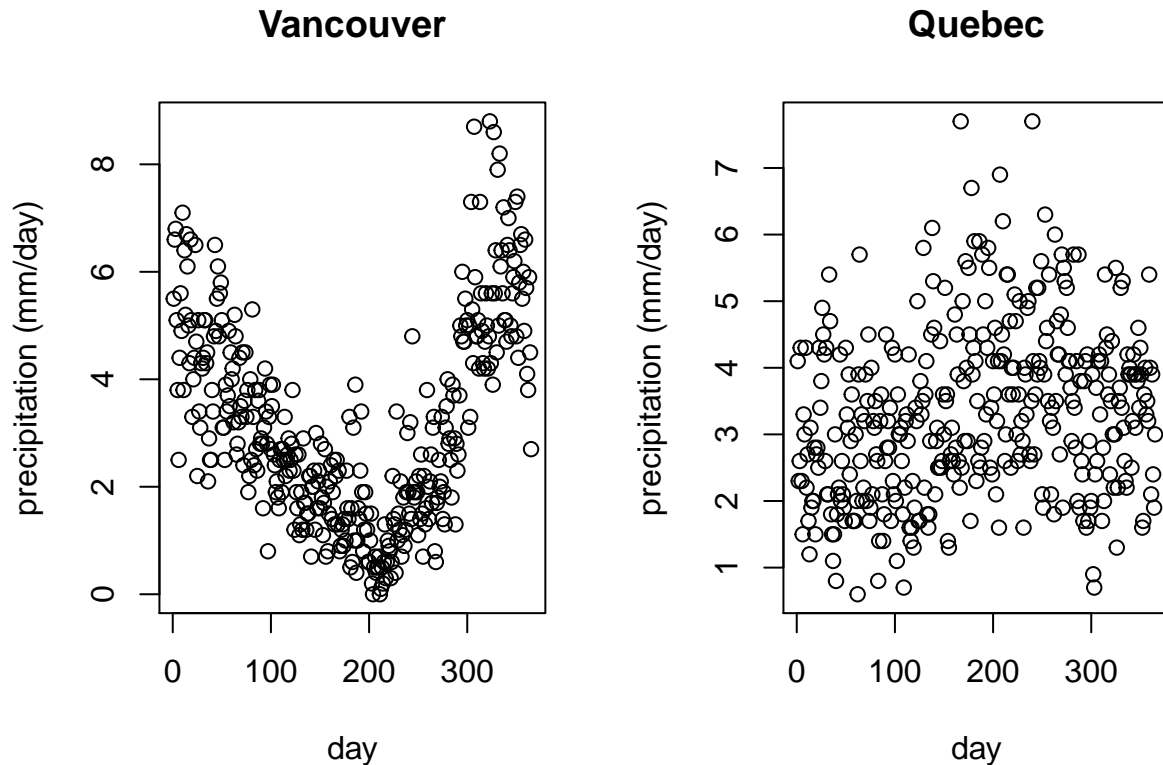
The data set also contains extra information about the cities. For example, the regions, provinces and the coordinates are included.

```
MetaData<-data.frame(city=colnames(da), region=CanadianWeather$region, province=CanadianWeather$province,
head(MetaData)
```

```
##      city      region      province coord.N.latitude
## St. Johns    St. Johns Atlantic Newfoundland      47.34
## Halifax      Halifax Atlantic   Nova Scotia      44.39
## Sydney        Sydney Atlantic   Nova Scotia      46.09
## Yarmouth      Yarmouth Atlantic   Nova Scotia      43.50
## Charlottvl    Charlottvl Atlantic   Ontario        42.48
## Fredericton   Fredericton Atlantic New Brunswick    45.58
##      coord.W.longitude
## St. Johns      52.43
## Halifax        63.36
## Sydney          60.11
## Yarmouth        66.07
## Charlottvl      80.25
## Fredericton     66.39
```

I show you the data of Vancouver and Quebec.

```
par(mfrow=c(1,2))
plot(1:365,da[, "Vancouver"], main="Vancouver", xlab="day", ylab="precipitation (mm/day)")
plot(1:365,da[, "Quebec"], main="Quebec", xlab="day", ylab="precipitation (mm/day)")
```



```
par(mfrow=c(1,1))
```

## 1.2. Research Question

The objective is to discover which cities have similar precipitation patterns, and which have dissimilar patterns. We need a 2-dimensional graph that shows each city as a point, such that cities with similar precipitation patterns are close to one another. We also want to understand the difference in rainfall patterns: in what sense do they differ (e.g overall more rainfall in Western Canada, or less rain in summer, ...).

## 1.3. Functional Data Analysis

### 1.3.1. Introduction

From the research question we conclude that a MDS is an appropriate method.

One way of looking at the data is to consider it as a multivariate data set with  $n = 35$  rows (cities) and  $p = 365$  columns (days). However, this is not the approach that we take. In this section I will introduce a functional data analysis (FDA) approach.

In FDA we typically consider functions as observations. For example, for each of the 35 cities we have observations on a precipitation function. To take this approach it is necessary for each city to first convert the 365 data entries to a single function. This function will contain parameter estimates (typically less than the original number of data entries, say  $q < p = 365$ ). Thus each city will have its set of  $q$  parameter estimates, and thus an  $n \times q$  data matrix can be constructed. These parameter estimates form now the input for the MDS. To give a meaningful interpretation to the results, at the end we need to back-transform our solution from the parameter space to the function space.

### 1.3.2. Transformation to functions

Let  $Y_i(t)$  denote the outcome of observation  $i = 1, \dots, n$  (here: average daily rainfall) at time  $t \in [1, 365]$ . For observation  $i$  we have data on times  $t_{ij}$ ,  $j = 1, \dots, p_i$ . This setting thus allows for irregularly spaced measurements: each observation  $i$  may come with its own set of time points  $t_{ij}$  with measurements  $Y_i(t_{ij})$ . Our data set, however, contains regularly spaced data: for each city  $i$  precipitations are available for  $t = 1, 2, \dots, 365$ .

Consider the non-linear statistical model

$$Y_i(t_{ij}) = f_i(t_{ij}) + \varepsilon_{ij} \quad i = 1, \dots, n; j = 1, \dots, p_i$$

for some smooth functions  $f_i(\cdot)$  and with  $\varepsilon_{ij}$  i.i.d. with mean 0 and constant variance  $\sigma^2$ . If  $f_i(\cdot)$  is a parametric function, then non-linear least squares can be used to estimate the parameters.

Functional approximation theory (mathematical theory) says that a sufficiently smooth function  $f(\cdot)$  can be arbitrary well approximated by the expansion

$$f_i(t) = \sum_{k=0}^{\infty} \theta_{ik} \phi_k(t)$$

where the  $\theta_{ik}$  are parameters, and the  $\phi_k(\cdot)$  form an set of orthonormal basis functions. In practice the expansion (sum) is truncated after a few terms, say  $m$ . We will thus use

$$f_i(t) = \sum_{k=0}^m \theta_{ik} \phi_k(t).$$

For this homework it is not important to exactly understand the meaning of an orthonormal basis. Classical examples of orthonormal basis functions: Fourier functions, polynomial basis functions, wavelet functions. For this homework you may choose between a Fourier or a polynomial basis.

Functions from a polynomial basis are of the form

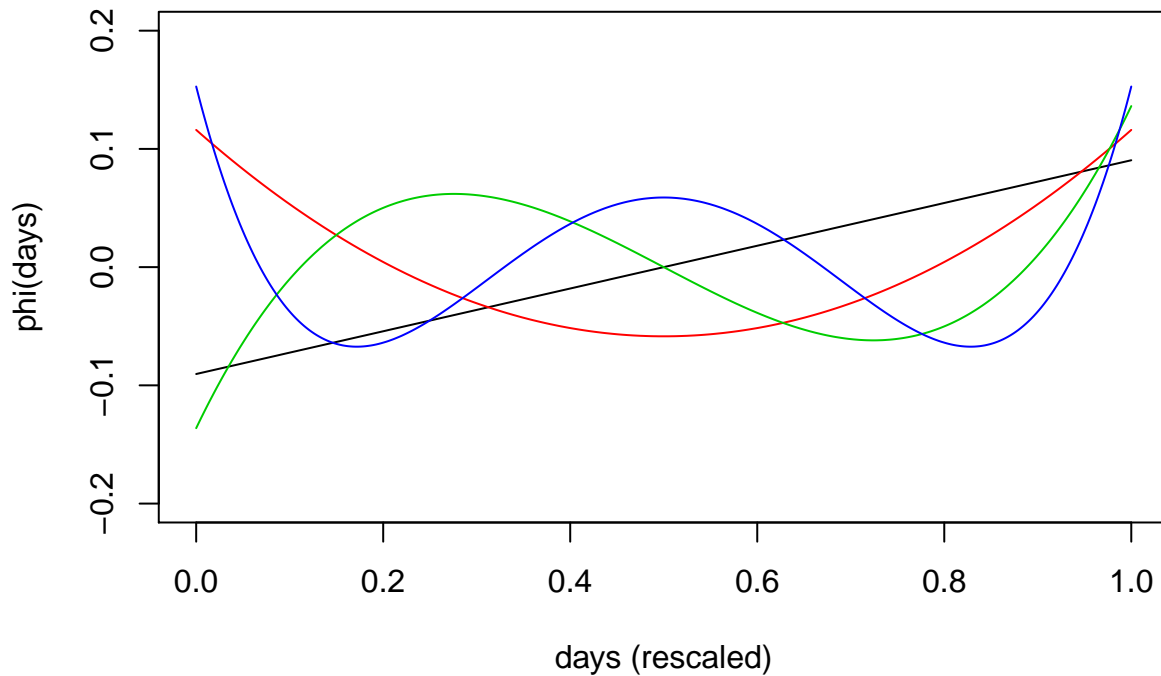
$$\phi_k(t) = \sum_{j=0}^k \alpha_j t^j,$$

i.e.  $\phi_k(\cdot)$  is a polynomial function in  $t$  of degree  $k$ . The  $\alpha_j$ 's are constants (no need to estimate).

The next chunk of R code illustrates the generation of a polynomial basis. Note that I rescaled the 365 days to the  $[0, 1]$  interval. This is only to avoid numerical problems when computing  $\text{days}^j$  for a large  $j$ .

```
days<-1:365
days<-(days-min(days))/(diff(range(days))) # rescaling to [0,1]
phi<-poly(days,degree=4)
# the i-th column of phi contains the i-th order
# polynomial evaluated in "days"

plot(days,phi[,1],type="l",ylim=c(-0.2,0.2), xlab="days (rescaled)", ylab="phi(days)")
lines(days,phi[,2],type="l",col=2)
lines(days,phi[,3],type="l",col=3)
lines(days,phi[,4],type="l",col=4)
```



```
# the next line demonstrates that the polynomials are normalised
# (no need to exactly understand this)
colMeans(scale(phi,center=T,scale=F)^2)
```

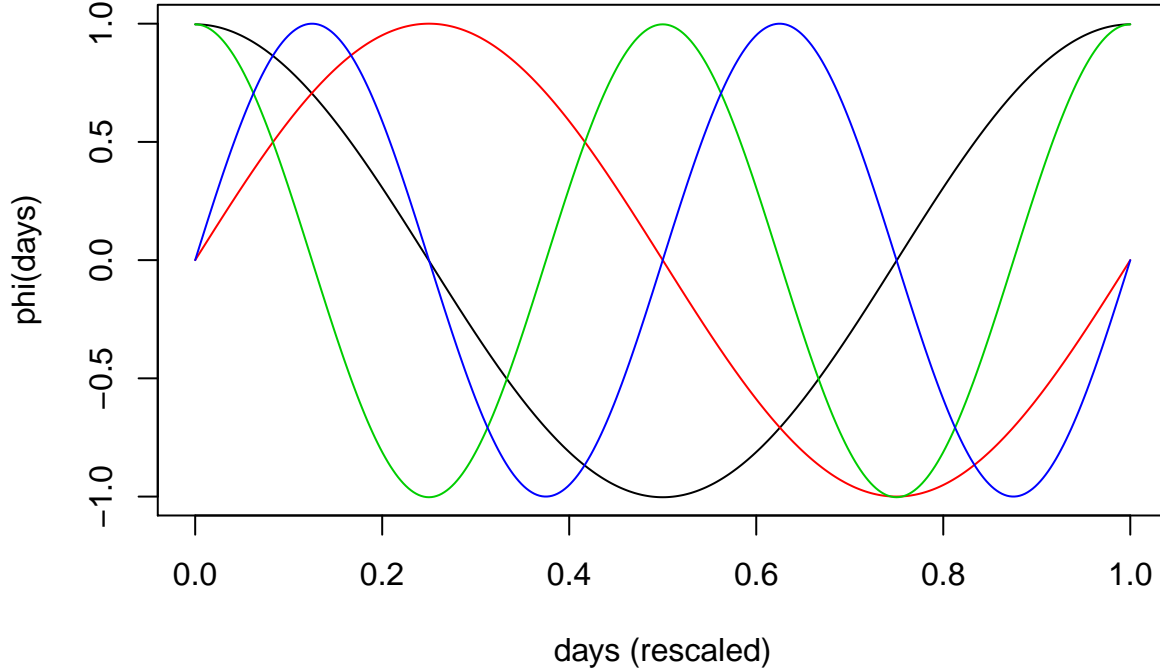
```
##           1           2           3           4
## 0.002739726 0.002739726 0.002739726 0.002739726
```

Now an example with the Fourier basis.

```
library(ldr) # needed for the bf function for generation of Fourier basis

phi<-bf(days,case="fourier",degree=4)
# the i-th column of phi contains the i-th order
# Fourier basis function evaluated in "days"

plot(days,phi[,1],type="l",ylim=c(-1,1), xlab="days (rescaled)", ylab="phi(days)")
lines(days,phi[,2],type="l",col=2)
lines(days,phi[,3],type="l",col=3)
lines(days,phi[,4],type="l",col=4)
```



```
# the next line demonstrates that the polynomials are normalised
# (no need to exactly understand this)
colMeans(scale(phi,center=T,scale=F)^2)
```

```
## [1] 0.5013624 0.4986301 0.5013624 0.4986301 0.5013624 0.4986301 0.5013624
## [8] 0.4986301
```

The R code has demonstrated that the `poly` and `bf` functions transform a vector with the days at which measurements are available, to a matrix. For a given city (i.e. for a given  $i$ ), the number of rows of the matrix equals the number of days, and each column corresponds to a basis function. The  $(j, k)$ th element of the matrix equals the  $k$ th basis function evaluated in the  $j$ th day  $t_{ij}$ . Let  $x_{ijk} = \phi_k(t_{ij})$  denote this element.

We now rewrite the statistical model for  $Y_i(t_{ij})$ ,

$$Y_i(t_{ij}) = \sum_{k=0}^m \theta_{ik} \phi_k(t_{ij}) + \varepsilon_{ij}$$

or

$$Y_i(t_{ij}) = \sum_{k=0}^m \theta_{ik} x_{ijk} + \varepsilon_{ij}.$$

For a given  $i$ , this has the structure of a linear regression model with outcomes  $Y_i(t_{ij})$ ,  $j = 1, \dots, n$ , and  $q = m + 1$  regressors  $x_{ijk}$ .

Finally, we write the statistical model for city  $i$  in matrix notation.

$$\mathbf{Y}_i = \boldsymbol{\theta}_i^t \mathbf{X}_i + \boldsymbol{\varepsilon}_i$$

with  $\mathbf{Y}_i$  the vector with the outcomes of observation  $i$  (one for each day  $t_{ij}$ ),  $\boldsymbol{\theta}_i$  the vector with the  $\theta_{ik}$  (one for each basis function  $k$ ),  $\mathbf{X}_i$  the matrix with the  $x_{ijk}$  (days  $j$  in the rows, basis function index  $k$  in columns), and  $\boldsymbol{\varepsilon}_i$  the vector with the i.i.d. error terms.

The parameters  $\theta_{ik}$  can be estimated by means of least squares. This is illustrated in the next R code for a single city (i.e. a single  $i$ ). The process need to be repeated for all cities  $i = 1, \dots, n$ . Once the parameters are estimated they can be used to plot the fitted function.

```

# Using the polynomial basis functions up to degree 3

days<-1:365
days<-(days-min(days))/(diff(range(days))) # rescaling to [0,1]
phi<-poly(days,degree=3)
dim(phi)

## [1] 365    3

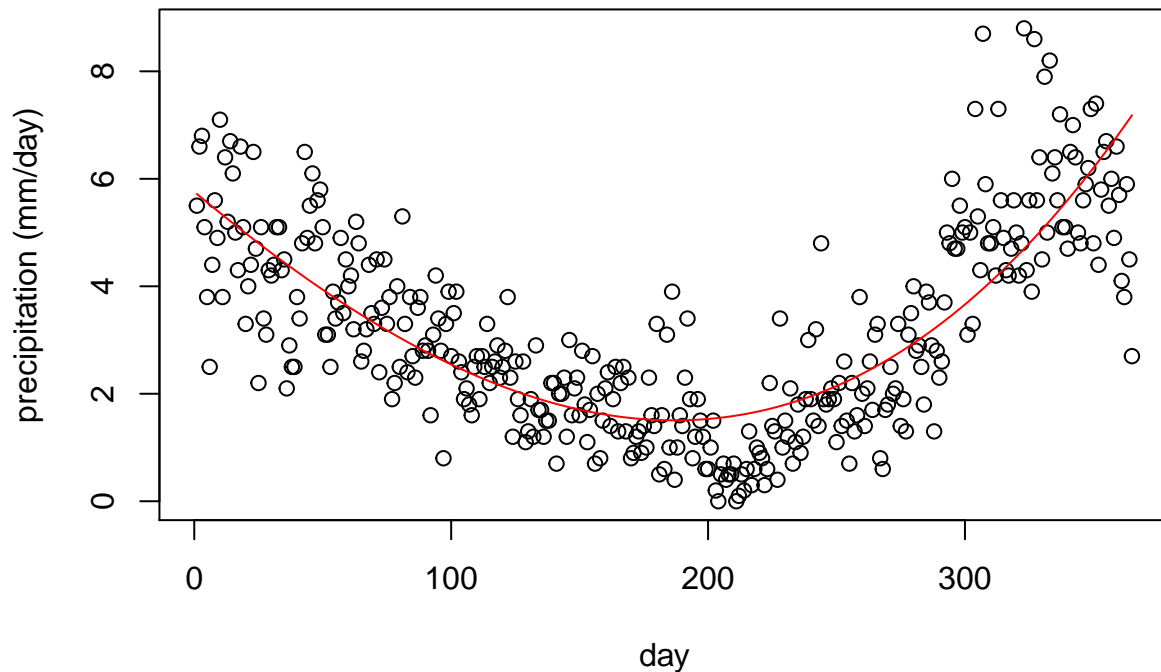
# estimation of the theta parameters for Vancouver
m.Vancouver<-lm(da[, "Vancouver"]~phi)
summary(m.Vancouver)

##
## Call:
## lm(formula = da[, "Vancouver"] ~ phi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.4831 -0.7312 -0.1068  0.6815  4.7459
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.1647     0.0592  53.455 < 2e-16 ***
## phi1          3.9234     1.1311   3.469 0.000586 ***
## phi2         28.3115     1.1311  25.031 < 2e-16 ***
## phi3          2.7691     1.1311   2.448 0.014833 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.131 on 361 degrees of freedom
## Multiple R-squared:  0.641, Adjusted R-squared:  0.638
## F-statistic: 214.9 on 3 and 361 DF,  p-value: < 2.2e-16

# plot of fitted function
plot(1:365,da[, "Vancouver"],main="Vancouver (m=3)",xlab="day", ylab="precipitation (mm/day)")
lines(1:365,m.Vancouver$fitted.values,type="l", col=2)

```

## Vancouver (m=3)



```
# Using the polynomial basis functions up to degree 10
```

```
days<-1:365
days<-(days-min(days))/(diff(range(days))) # rescaling to [0,1]
phi<-poly(days,degree=10)
dim(phi)
```

```
## [1] 365 10
```

```
# estimation of the theta parameters for Vancouver
```

```
m.Vancouver<-lm(da[, "Vancouver"]~phi)
summary(m.Vancouver)
```

```
##
```

```
## Call:
```

```
## lm(formula = da[, "Vancouver"] ~ phi)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.8201 -0.6551 -0.0621  0.5794  3.8571
```

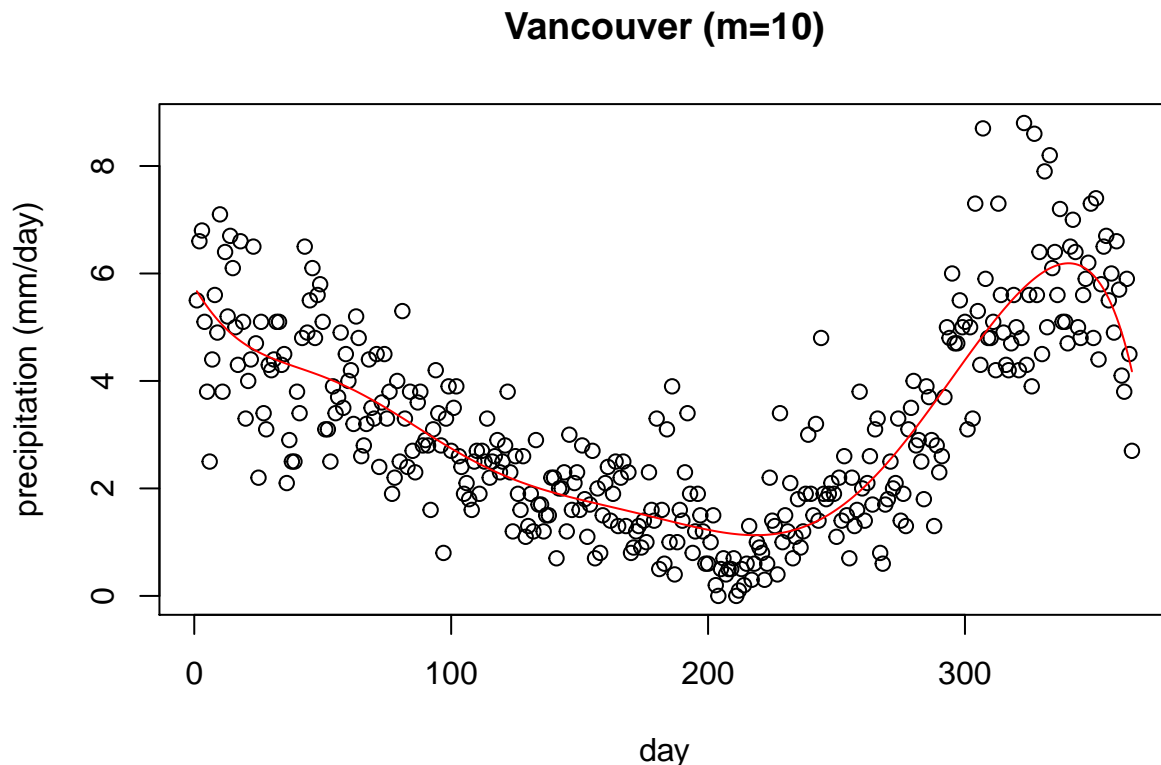
```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.1647     0.0523  60.511 < 2e-16 ***
## phi1          3.9234     0.9992   3.927 0.000104 ***
## phi2         28.3115     0.9992  28.335 < 2e-16 ***
## phi3          2.7691     0.9992   2.771 0.005878 **
## phi4         -6.7529     0.9992  -6.758 5.76e-11 ***
## phi5         -6.9005     0.9992  -6.906 2.32e-11 ***
## phi6         -3.4009     0.9992  -3.404 0.000741 ***
## phi7         -1.2774     0.9992  -1.278 0.201913
```



```
## phi8          1.2035      0.9992      1.205 0.229201
## phi9         -0.4364      0.9992     -0.437 0.662548
## phi10        -0.6000      0.9992     -0.601 0.548537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9992 on 354 degrees of freedom
## Multiple R-squared:  0.7253, Adjusted R-squared:  0.7175
## F-statistic: 93.46 on 10 and 354 DF,  p-value: < 2.2e-16
# plot of fitted function
plot(1:365,da[, "Vancouver"],main="Vancouver (m=10)", xlab="day", ylab="precipitation (mm/day)")
lines(1:365,m.Vancouver$fitted.values,type="l", col=2)
```



The R code demonstrates that the quality of the fit improves with increasing number of basis functions. Later in the course we will see methods for choosing an appropriate number of basis functions. For this homework I suggest that you choose an  $m \leq 20$  that makes sense to you. I'm not claiming that this is a good method, but for now this will do.

### 1.3.3. Multidimensional Scaling of Functions

Let  $\hat{\theta}_i$  denote the vector with the parameter estimates. Given the set of basis functions,  $\hat{\theta}_i$  contain the information (shape) of the precipitation functions for observation  $i$ . The estimates for all cities can be collected into a single new  $n \times (m + 1)$  data matrix  $\Theta$ . The  $i$ th row of  $\Theta$  is  $\hat{\theta}_i^t$ . The matrix  $\Theta$  contains now all information on the shape of the precipitation functions. Since  $\Theta$  has the structure of an ordinal data matrix, classical multivariate techniques can be applied to it. We will apply a MDS to  $\Theta$ , so that we can construct a 2-dimensional plot with each point representing a city. The distances between the points in the 2-dimensional MDS space are approximations of the distances between the rows of  $\Theta$ , and hence can be interpreted as distances between the precipitation functions.

The MDS thus starts from the truncated SVD of  $\Theta$  (or after column-centering!),

$$\Theta_k = U_k D_k V_k^t$$

(note that the index  $k$  now refers to the number of components in the truncated SVD and not to the index of the basis functions).

### 3. Functional Biplot

A biplot directly constructed from the SVD of  $\Theta$  is hard to interpret. A better interpretation arises from backtransforming the SVD to the original function space.

From Section 1.3.2 we have the statistical model for city  $i$  in matrix notation.

$$\mathbf{Y}_i = \mathbf{X}_i \boldsymbol{\theta}_i + \boldsymbol{\varepsilon}_i$$

with  $\mathbf{Y}_i$  the vector with the outcomes of observation  $i$  (one for each day  $t_{ij}$ ,  $j = 1, \dots, p$ ),  $\boldsymbol{\theta}_i$  the vector with the  $\theta_{ik}$  (one for each basis function  $k$ ),  $\mathbf{X}_i$  the matrix with the  $x_{ijk}$  (days  $j$  in the rows, basis function index  $k$  in columns), and  $\boldsymbol{\varepsilon}_i$  the vector with the i.i.d. error terms.

The  $\boldsymbol{\theta}_i$  parameters estimates are denoted by  $\hat{\boldsymbol{\theta}}_i$ , and as before the  $i$ th row of  $\Theta$  is  $\hat{\boldsymbol{\theta}}_i^t$ . The fitted model for all  $n$  cities may then be simultaneously written as

$$\hat{\mathbf{Y}} = \Theta \mathbf{X}^t$$

with  $\hat{\mathbf{Y}}$  the  $n \times p$  matrix with  $i$ th row  $\mathbf{Y}_i^t$ , and  $\mathbf{X}$  the  $p \times (m+1)$  matrix  $\mathbf{X}_i$  as defined before (note that all  $\mathbf{X}_i$  are equal because for all cities  $i$  the measurements were obtained at the same  $p$  time points (days)).

After substituting  $\Theta$  with its truncated SVD (after  $k$  terms) we get the model fit

$$\hat{\mathbf{Y}}_k = \Theta_k \mathbf{X}^t = U_k D_k V_k^t \mathbf{X}^t = \mathbf{Z}_k V_k^t \mathbf{X}^t.$$

The latter expression relates an approximate model fit ( $\hat{\mathbf{Y}}_k$ ) for the complete data set of  $n$  cities to the scores of the SVD ( $\mathbf{Z}_k$ ) and the loadings ( $\mathbf{V}_k$ ). The relation is established through the matrix  $\mathbf{X}$  which was used to relate the basis functions to the  $\theta$ -parameters. In particular, the  $(j, k)$  element of  $\mathbf{X}$  was given by  $\phi_k(t_j)$  (before we wrote  $t_{ij}$  but since for all cities  $i$  the measurements were taken at the same times, we simplified the notation from  $t_{ij}$  to  $t_j$ ). The model fit of the previous equation may thus also be written in functional form

$$\hat{Y}_{ki}(t) = \sum_{j=1}^k \sum_{r=0}^m z_{kij} v_{rj} \phi_r(t)$$

where  $z_{kij}$  is the  $(i, j)$ th element of  $\mathbf{Z}_k$  and  $v_{rj}$  is the  $(r, j)$ th element of  $\mathbf{V}_k$  (or, equivalently, of  $\mathbf{V}$ ).

For a given set of basis functions  $\phi_r(t)$  and the SVD of the matrix with parameter estimates  $\Theta$ , the expression for  $\hat{Y}_{ki}(t)$  may be used for plotting informative graphs to express the interpretation of the SVD of  $\Theta$  in terms of the original functions.

Some motivation/suggestions for informative plots for  $k = 2$ :

- $\hat{Y}_{2i}(t)$  is written as a function of the scores  $z_{2ij}$ , and these scores are plotted as points  $(z_{2i1}, z_{2i2})$  in a two-dimensional graph (e.g. MDS plot)
- to understand the meaning of the first dimension,  $\hat{Y}_{2i}(t)$  can be plotted as a function of  $t$  with  $z_{2i1}$  varying between a small and a large score, and with  $z_{2i2}$  fixed at a meaningful constant
- to understand the meaning of the second dimension,  $\hat{Y}_{2i}(t)$  can be plotted as a function of  $t$  with  $z_{2i2}$  varying between a small and a large score, and with  $z_{2i1}$  fixed at a meaningful constant

Many other variants are possible.

Using the package will not give you deep insight into how an FDA works.

## 4. Assignment

You may work in groups (same groups as for the other project).

For this project you are asked to perform a MDS on the average rainfall functions for the Canadian cities. In particular:

- choose a set of basis functions (polynomial or Fourier basis with some  $m \leq 20$ )
- transform the data series for each Canadian city to a fitted function (estimated  $\theta$  parameters)
- collect all estimated  $\theta$  parameters into a  $\Theta$  matrix
- perform a MDS on the (column-centered)  $\Theta$  matrix
- make one or two informative graphs
- interpret the graph(s) (you can e.g. use the metadata) and write a clear and concise conclusion

You are asked to present your results as an R markdown file (you may use this file as a template – you may also use the R code from this file). Briefly comment your R code, motivate your choices and provide interpretation of your final results. The report (excl. R code, R output and graphs) should not count more than 3 pages, and it should include a clear and concise conclusion. Please limit the number of important graphs on which your conclusions are based, to 4.

There is an R package that contains functions for an FDA. However, you are explicitly asked NOT to use this package. Using the package will not give you deep insight into how an FDA works.

Note that this document is quite lengthy, but it includes a detailed description of the methods and it includes R code that you may use. Once you have read the methodology, I expect that your work effort will not take more than one working day.

At the final exam, you will be asked a few questions related to this project and to the multivariate methods discussed in class.

## 5. Timeline

- March 1: project assignment is posted on Blackboard
- March 14: in the lecture there will be time for Q&A
- June 5, 16h00: deadline for submission of final report