Advance Statistics course
Some explanations on fda package

Haik Daniel

University Autonoma of Madrid

Professor : Antonio Cuevas

May 2018

Contents

1 Data processing

In functional data analysis our data $(X_1,...,X_n)$ are some iid realisation of a random variable X taking value in a functional space $\mathcal{X}$, $X : \Omega \to \mathcal{X}$.

But in pratice we can only observe a finite number of point, and our observations can be seen as a matrix of size n x m where n is the number of data and m is the amount of points that we have observed.

$\forall\ i=1,...,n,\ X_i=(X_i(t_1),...,X_i(t_m)),\ t_1,..,t_m \in \mathbb{R}_+$.

$$\begin{pmatrix} X_1 \\ ... \\ X_n \end{pmatrix} = \begin{pmatrix} X_1(t_1)... & X_1(t_m) \\ ... & ... \\ X_n(t_1)... & X_n(t_m) \end{pmatrix}$$

We obviously want to have a number of data m large enough.
With this data we want to have the hole function on a given interval.

Reference : (8) on fdata object in fda package.

In our case $\mathcal{X} = \mathcal{L}^2([a,b])$ ($[a,b]$ the definition interval $a \leq b$), so a function can be express as :

$\forall\ t \in [a,b],\ X(t) = \sum_k \alpha_k \phi_k(t)$ where $\phi_k(t)$ are some basis functions of $\mathcal{L}^2([a,b])$
For example Fourier basis : $(2sin(k\pi t))_{k\geq 1}$ and α_k are the Fourier coefficient of the function.

In practice we compute $\hat{X}(t) = \sum_{k=1}^{K} \alpha_k \phi_k(t)$ the pointwise estimator of the function for K a given integer.

Reference : (1) 2.2 Basis representation page 8

Since the basis $(\phi_1(t),...,\phi_K(t))$ is known we only have to estimate $\alpha = (\alpha_1,...,\alpha_K) \in \mathbb{R}^K$ by least squared :

Indeed, if we write the model we have :

$$X(t) = \hat{X}(t) + \epsilon(t) = \sum_{k=1}^{K} \hat{\alpha}(k)\phi_k(t) + \epsilon(t)$$

where $X(t)$ is the real value of the function, $\epsilon(t)$ the error assume to be centered,uncorrelated and with finite variance.

The model can be rewrite as follow considering the data observations t_1,..,$t_m \in \mathbb{R}_+$:

$$\begin{pmatrix} X(t_1) \\ ... \\ X(t_m) \end{pmatrix} = \begin{pmatrix} \phi_1(t_1)... & \phi_K(t_1) \\ ... & ... \\ \phi_1(t_m)... & \phi_K(t_m) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ ... \\ \alpha_K \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ ... \\ \epsilon_K \end{pmatrix}$$

$$X = \phi\ \alpha + \epsilon$$

with matrix notation, ϕ is the m x K matrix of basis function, $\alpha = (\alpha_1,...,\alpha_K)'$ Then we minimize the sum of squared error,

$$\hat{\alpha} = \underset{\alpha \in \mathbb{R}^K}{min} \sum_{i=1}^{m} \left(X(t_i) - \sum_{k=1}^{K} \alpha_k\ \phi_k(t_i) \right)^2$$

If the matrix ϕ satisfy $\mathrm{rg}(\phi) = K$ then we have an unique solution of the problem given by $\hat{\alpha} = (\phi\ '\ \phi)^{-1}\ \phi\ '\ X$

However the problem is not always well solved because we have to choose K the number of basis functions and also the order of these functions who appears in the equation of $\hat{X}(t)$ or over-fitted problem.
We can add a "penalisation" in order to find another estimator for α by minimizing the quantity :

$$\underset{\alpha \in \mathbb{R}^K}{min} \sum_{i=1}^{m} \left(X(t_i) - \sum_{k=1}^{K} \alpha_k\ \phi_k(t_i) \right)^2 + \lambda\ ||\ \frac{d^2X}{dt^2}\ ||^{\ 2}$$

where $\frac{d^2X}{dt^2}$ denote the second derivative of X which will be estimated with our first estimation of α given above.
λ a smoothing parameter which must also be determined by the Cross validation.

Then once we have calculated $\hat{\alpha}$ we can estimate pointwise the function for a given t in the interval, $\hat{X}(t) = \sum_k^K \hat{\alpha_k}\phi_k(t)$.

Reference : (5) part "Generalized cross-validation".

2 Example with a data set in fda package

In the fda package **fdata** is an object representing only the data matrix n x m for n functions observated at 100 points that we saw in the first page.

Thanks to smoothing we create a new object called **fd** wich contains the basis functions ϕ, the coefficient α and the new function.

There exists a several choice of basis in fda.package wich can be used with respect to the behaviour of the data :
• create.polygonal.basis
• create.exponential.basis,
• create.fourier.basis,
• create.bspline.basis .

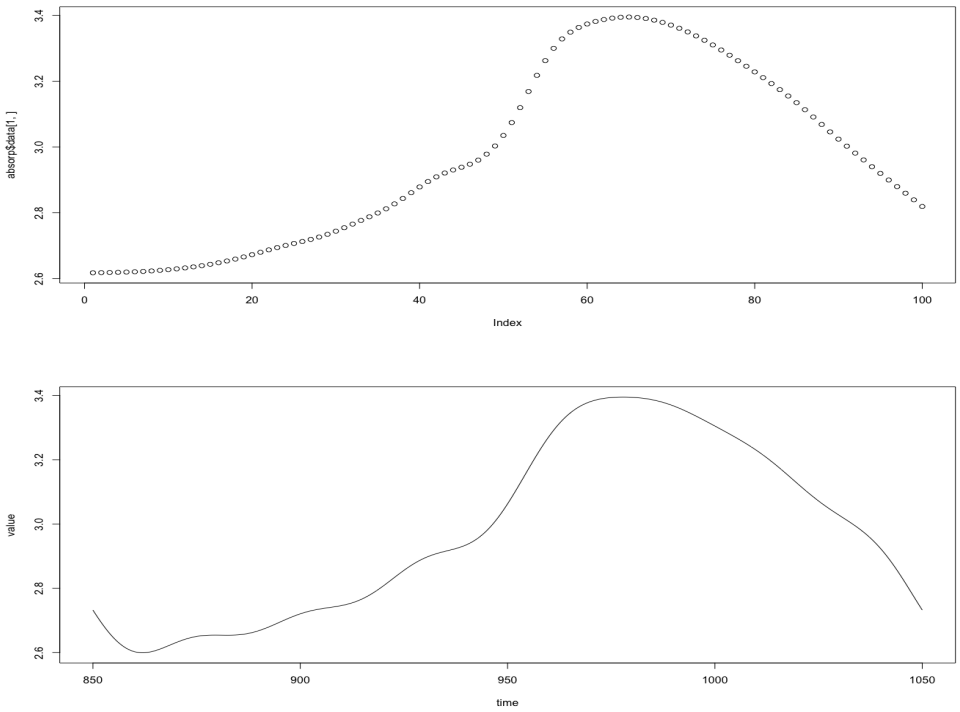
We upload a set of data called "tecator".
This example deals with spectrometric data, we observe the infrared spectrum of 215 piece of meat.
Here we have a matrice n x m, $n = 215$, $m = 100$, so 215 functions evaluated at 100 points.
As an example we want to smooth the first data X_1.

We begin by generate the basis functions thanks to the commande :
• aBS $\leftarrow$ create.fourier.basis(c(1,100))

Then we construct the matric ϕ corresponding so a matrix $m = 100$ x K=number of basis.
We write :
• absorpX $\leftarrow$ predict(aBS,absorpdata[1,])
We print:
dim(absorpX) : " 100 3 "

Then we get the coefficient α estimated by :
• aB $\leftarrow$ solve(crossprod(absorpX),crossprod(absorpX,absorpdata[1,]))

In the first graphic below we plot the raw function $X_1 = (X_{t_1},...,X_{t_m})$ and in the second one the new function $X_1(t)$, $t \in [850,1050]$ obtained thanks to Fourier basis.



Once we have a smooth function we can work on it.

Estimate the mean function by $\hat{m}(t) = \frac{1}{n}\sum_{i=1}^{n} X_i(t)$ with the commande mean(object of type fd)

Then the variance : $\hat{var}(t) = \frac{1}{n-1}\sum_{i=1}^{n} (X_i(t) - m(t))^2$ and $\sigma(t) = \sqrt{var(t)}$ with the commande std.fd(object of type fd).

The covariance function $\hat{K}(s,t) = \frac{1}{n-1}\sum_{i=1}^{n} (X_i(t) - m(t))(X_i(s) - m(s))$, s , $t \in [850,1050]$.
We can remark that all these quantities are random and depend only on the observations.

Reference : (10), presentation of the set of data "tecator".

3 Functional linear regression methods

From a functionnal data $X \in \mathcal{L}^2([a,b])$ we want to explain the effect of a response Y which could be a scalar, a finite vector or a function, here we only study the case of vector response.

In the case in our data X lives in $\mathbb{R}^n$ the model is :

$$Y = < X \ , \beta > + \epsilon = \sum_{k=1}^{n} X_k \ \beta_k + \epsilon.$$

where $< \ , >$ denote the inner product in $\mathbb{R}^n$, ϵ the error and we have to estimate β.

In the functional case it is a generalisation since we have an inner product in $\mathcal{L}^2([a,b])$. And we have the following model we scalar response :

$$Y = < X \ , \beta > + \epsilon = Y = \int_a^b X(s) \ \beta(s) \ dt + \epsilon \ (1)$$

where $\beta \in \mathcal{L}^2([a,b])$, ϵ is the error with finite variance and $\mathbb{E}(X_t \ \epsilon)=0 \ \forall \ t \in [a,b]$.

Reference : (4) page 15 "Functional regression models with scalar response".

Definition :
Let K be the covariance function of a square integrable process X.
Then we define T_K the linear operator in $\mathcal{L}^2([a,b])$ defined by : $\forall \ f \in \mathcal{L}^2([a,b])$, $T_K(f) = \int \ K(s,t) \ f(s) \ ds$

Karhunen–Loève Theorem :
If X is a square integrable process $\in \mathcal{L}^2([a,b])$ with zero mean and continuous covariance function $K : \ s,t \rightarrow \text{cov}(X_s,X_t)$ then there exists an unique decomposition of X given by :

$$\forall \ t \in [a,b], \ X(t) = \sum_{k \geq 1} Z_k v_k(t)$$

where $(v_k)_{k \geq 1}$ an orhonormal basis of $\mathcal{L}^2([a,b])$ formed by the eigen functions of the covariance operator T_K with eigen values λ_k ie $\forall \ k \geq 1$, $T_K(v_k) = \lambda_k \ v_k$.
And Z_k are random elements with zero mean, uncorellated and with variance λ_k.
Z_k are compute as $Z_k = \int_a^b X(t) v_k(t) \ dt$.

Reference : (4) page 10 "Functional Principal Component Analysis".

Going back to the linear model in (1), we have that $X_t \ Y = \int \ X_t \ X_s \ \beta(s) \ ds$ $+ \ X_t \ \epsilon$ then $\mathbb{E}(X_t \ Y) = \int \ K(s,t) \beta(s) \ ds$ (2).
We want to find a solution $\beta \in \mathcal{L}^2([a,b])$ for (2) equation.

Since $\beta \in \mathcal{L}^2([a,b])$ we can express it with $(v_k)_{k\geq 1}$ the basis functions given in the Karhunen-Loeve decomposition of X.
Then $\beta(t) = \sum_{k=1}^{\infty} \beta_k \, v_k(t)$, substitute this decomposition in (2) we have :

$$\begin{aligned}
\mathbb{E}(X_t Y) &= \int K(s,t) \sum_{k=1}^{\infty} \beta_k v_k(s) ds \\
&= \sum_{k=1}^{\infty} \beta_k \int K(s,t) v_k(s) ds \\
&= \sum_{k=1}^{\infty} \beta_k \lambda_k v_k(t)
\end{aligned}$$

On the other hand, using the Karhunen–Loève decomposition,
$X_t = \sum_{k\geq 1} Z_k v_k(t)$, we have :

$$\begin{aligned}
\mathbb{E}(X_t Y) &= \mathbb{E}(\sum_{k=1}^{\infty} Z_k v_k(t) Y) \\
&= \sum_{k=1}^{\infty} \mathbb{E}(Z_k Y v_k(t)) \\
&= \sum_{k=1}^{\infty} \mathbb{E}(\int X_s Y v_k(s) ds) v_k(t) \\
&= \sum_{k=1}^{\infty} (\int \mathbb{E}(X_s Y) v_k(s) ds) v_k(t) \\
&= \sum_{k=1}^{\infty} c_k v_k(t)
\end{aligned}$$

where $\forall\ k \geq 1$, $c_k = \int \mathbb{E}(X_s\ Y)\ v_k(s)\ ds$.

Then by uniqueness of the decomposition, by identification we must have
$\forall\ k \geq 1$, $\beta_k = \frac{c_k}{\lambda_k}$.

Thus we can express the solution of the regression problem,

$$\forall\ t \in [a,b],\ \beta(t) = \textstyle\sum_{k\geq 1} \frac{c_k}{\lambda_k}\ v_k(t).$$

However this solution does not always exist and there is not always an unique one. We give some condition below.

Picard Theorem:
In the linear model with scalar response $Y = \int_a^b X(s)\ \beta(s)\ dt + \epsilon$, if X is a square integrale process with zero mean and denoting by c_k and λ_k the quantities introduced above.
Then there exists an unique solution β in $\mathcal{L}^2([a,b])$ for the equation $\mathbb{E}(X_t\ Y) = \int K(s,t)\beta(s)\ ds$ if and only if $\sum_{k\geq 1} \frac{c_k^2}{\lambda_k^2} < \infty$.

Reference (french) : (9) page 10 "Regression sur données fonctionelles" used for finding the coefficients of beta function.

Reference (french) : (6) page 6, 3.1 , "Le modèle fonctionnelle de régression linéaire. L'approche PLS"
I used that Wiener-Hopf equation is equivalent to least square criterium.

Reference : (1) page 21 of papers "5.1 Eigen-decomposition of lameness data".

Since we are working in a Hilbert space, normal convergence implies convergences and $\sum_{k\geq 1} \frac{c_k}{\lambda_k}\ v_k(t)$ converge if $\sum_{k\geq 1} \mid\mid \frac{c_k}{\lambda_k}\ v_k(t) \mid\mid^2$ converges,
$\sum_{k\geq 1} \mid\mid \frac{c_k}{\lambda_k}\ v_k(t) \mid\mid^2 = \sum_{k\geq 1} (\frac{c_k}{\lambda_k})^2 \mid\mid v_k(t) \mid\mid^2 = \sum_{k\geq 1} \frac{c_k^2}{\lambda_k^2}$
Then we need to have that $\sum_{k\geq 1} \frac{c_k^2}{\lambda_k^2} < \infty$ to make sure that such a β function given above does exist.

In practice we need to estimate a lot of quantities,the eigen function v_k are unknown.

Now we have given some theorical result on β we can see the problem in a different way.
We express β in the same basis (v_k) as before so $\beta(t) = \sum_{k\geq 1}\ \beta_k\ v_k(t) \sim \sum_{k=1}^K\ \beta_k\ v_k(t)$.
And the model becomes,

$$Y = < X\ ,\ \beta > + \epsilon = < \sum_{k=1}^{\infty}\ Z_k\ v_k(t)\ ,\ \sum_{l=1}^{\infty}\ \beta_l\ v_l(t) > + \epsilon,$$
$$Y = \sum_{k=1}^{\infty}\ \sum_{l=1}^{\infty}\ Z_k\ \beta_l < v_k, v_l > + \epsilon = \sum_{k=1}^{\infty}\ Z_k\ \beta_k + \epsilon$$

since (v_k) is an orthonormal basis.

Reference : (4) page 15-16, how to transform the functional model with Karhunen-Loeve transformation.

Thus, we can express the model by $Y = \sum_{k=1}^{K} Z_k \, \beta_k + \epsilon$ with a given integer K, since the coefficients Z_k can be estimated we are back in the classical regression model for a data $X \in \mathbb{R}^K$ and the coefficients β_k are given by least squared in order to find the hole function β.

In n dimension, we have $(X_1,...X_n)$ n functions and $Y \in \mathbb{R}^n$,
$Y = (Y_1,...,Y_n)' = (\int_a^b X_1(t) \, \beta(t) \, dt,...,\int_a^b X_n(t) \, \beta(t) \, dt)' + \epsilon$.

We are thus looking for the " best " function $\beta \in \mathcal{L}^2([a,b])$ that explains the model.
We apply the Karhunen–Loève decomposition for each function X_i, $i = 1,...,n$ and we denote by $Z_{i,k}$ $k = 1,...,K$ the coefficients corresponding to the i-th function X_i. And let Z the n x K matrix, $(Z)_{i,j} = Z_{i,j}$.
The model can be written as

$$Y = Z \, \beta + \epsilon$$

where Z n x K matrix and $\beta \in \mathbb{R}^K$. Then the solution for the coefficient for β are given by $\hat{\beta}=(Z' \, Z)^{-1} \, Z' \, Y$ if $\text{rank}(Z) = K$ otherwise $Z' \, Z$ is not invertible.

We can add to the problem a penalisation on β. If β can be written on a set of basis function of $\mathcal{L}^2([a,b])$,
The B-spline basis is often use to represents β ,$\hat{\beta}(t) = \sum_{k=1}^{K} b_k \, \phi_k(t)$.

The estimator for b with the penalisation can be written as :

$$\hat{b} = \underset{b \in \mathbb{R}^K}{argmin} \, \frac{1}{n} \sum_{i=1}^{n} (Y_i - < X_i \, , \, b^t \, \phi >)^2 + \lambda \, || \, b^t \, \phi^{(2)} \, ||^2$$

where $||.||$ is the $\mathcal{L}^2$ norm, $\phi = (\phi_1,...,\phi_K)'$ and $\phi_k^{(2)}$ the second derivative of the function ϕ_k and $\lambda \geq 0$ a parameter.

Reference : (5) part 1 "Generals" equation (3) for penalization criterium.

Reference : (1) page 9 of papers, "3.3 LASSO", I have adapted Lasso criterum to penalization with the second derivative.

Using that the basis of function is orthonormal as before we then have that $|| \sum_{k=1}^{K} b_k \phi_k ||^2 = < \sum_{k=1}^{K} b_k \phi_k, \sum_{j=1}^{K} b_j \phi_j > = \sum_{k,j=1}^{K} b_k b_j < \phi_k, \phi_j > = \sum_{k=1}^{K} b_k^2$

$$\hat{b} = \underset{b \in \mathbb{R}^K}{argmin} \frac{1}{n} \left(\sum_{i=1}^{n} (Y_i - \sum_{k=1}^{K} Z_{i,k} b_k)^2 + \lambda \sum_{k=1}^{K} b_k^2 \right)$$

Computing the Lagrangien function of the minimization problem :

$$\mathcal{L}(b) = \sum_{i=1}^{n} (Y_i - \sum_{k=1}^{K} Z_{i,k} b_k)^2 + \lambda \sum_{k=1}^{K} b_k^2$$

Then $\frac{d\mathcal{L}(b)}{db_l} = -2 < Y - Z b, Z_l > + 2 \lambda b_l, \forall l = 1,...,K$, here we use the euclidien inner product.

$$\frac{d\mathcal{L}(b)}{db_l} = 0 \Longleftrightarrow - < Y - Z b, Z_l > + \lambda b_l = 0, \forall l = 1,...,K$$

With matrix form, $- < Y - Z \hat{b}, Z > + \lambda \hat{b} = 0 \Longleftrightarrow \hat{b}' (Z' Z + \lambda I) = Y' Z$ $\Rightarrow$ the estimator is $\hat{b} = (Z' Z + \lambda I)^{-1} Z' Y$ if the invert makes sense.

Then $\hat{Y} = Z \hat{b} = \mathcal{P}_\lambda Y$, by denoting $\mathcal{P}_\lambda := Z(Z' Z + \lambda I)^{-1} Z'$ the n x n matrix.

How to choose λ - parameter ?

We define the estimation of Y without the data y_k by $\hat{Y}_\lambda^{[k]} = Z \hat{b}^{[k]}$

where $\hat{b}^{[k]} = \underset{b \in \mathbb{R}^K}{argmin} \frac{1}{n} \left(\sum_{i=1, i \neq k}^{n} (Y_i - \sum_{k=1}^{K} Z_{i,k} b_k)^2 + \lambda \sum_{k=1}^{K} b_k^2 \right)$
The ordinary cross validation (OCV) is the function of square error :

$$V(\lambda) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_{\lambda,i}^{[i]})^2$$

This is natural to minimize this function in λ in order to get our parameter which reduce the variance of the error.

Lemme : If we denote $a_{i,k}(\lambda)$ the coefficient of the matrix $\mathcal{P}_\lambda$, we have

$$Y_i - \hat{Y}_{\lambda,i}^{[i]} = \frac{Y_i - \hat{Y}_i}{1 - a_{ii}(\lambda)}, \ i=1,...,n$$

Then the ordinary cross validation becomes,

$$V(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{Y_i - \hat{Y}_i}{1 - a_{ii}(\lambda)} \right)^2 \ (*)$$

Generalized Cross validation :

We define the function :

$$\text{GCV}(\lambda) = \frac{\frac{1}{n}||Y - \mathcal{P}_\lambda Y||^2}{(\frac{1}{n} tr(I - \mathcal{P}_\lambda))^2} \ , \ \lambda \geq 0.$$

Then the parameter λ can be chosen as : $\overline{\lambda} = \underset{\lambda \geq 0}{argmin}\ \text{GCV}(\lambda)$

The reason why we do not use the first definition of (OCV) is that it requires too much computations.

The generalized cross validation function is obtained by changing in $(*)$ $a_{ii}(\lambda)$ into $\frac{1}{n}\ \text{tr}(\mathcal{P}_\lambda)$, each diagonal element is subtituted by the mean of the diagonal element.

Reference : (5) part "Generalized cross-validation" I have adapted with my model the OCV and GCV criterium from this source.

Regression with data

We use spectrometric data where 215 piece of meat have been observed.
For the i-th piece of meat we know the fat content Y_i.
Given a spectromatric curve, can we estimate the fat content of this piece of meat ?
Indeed, the analysis of the fat is more time-consuming that the spectrometric process .
We want to know the link between the variable Y_i and the curve X_i corresponding to the observation $i \in$ [1,215].
We build the model :

$$Y_i = \int X_i(t)\beta(t)dt + \epsilon_i,\ i = 1,...,215.$$

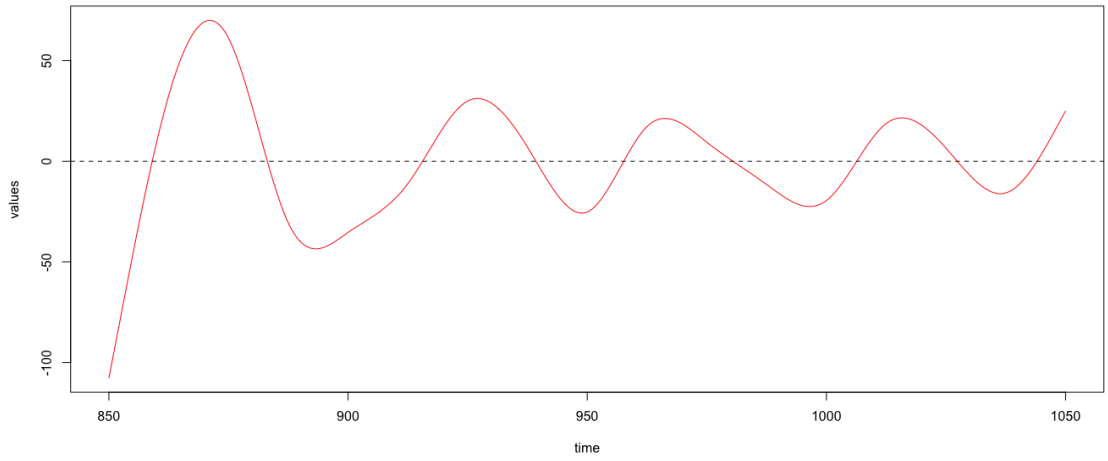
We have to estimate the functional parameter β by solving the following minimization problem :

$$\underset{\beta \in \mathcal{B}}{argmin}\ \frac{1}{n} \sum_{i=1}^{n} (Y_i - \int X_i(t)\beta(t)dt)^2 + \lambda\ ||\ \beta^{(2)}\ ||^2$$

$\mathcal{B}$ is all the functions in $\mathcal{L}^2$ written in a B-spline basis.

We have seen so far that we can express X_i and β on two differents basis or with the same basis computed from the eigen functions of X.

The function *fregre.basis* let us choose the basis for X_i and β,
whereas *fregre.pc* compute the regression with the eigen functions basis of X_i.



We get the function β with cross-validation criterium $\lambda = 7{,}39$.

Reference : (11), presentation and example of the function fregre.pls to compute regression with gcv criterium .

Conclusion : From finite dimension observations we are able to build the hole function and to compute linear model, however the price to pay is that a lot of unknown parameters come into account.
For example the number of functions basis that we choose to express the function, the right basis adapted to the function, the best penalization criterium for penalty and many others.
Study of functional data involves a lot of contraints since we are working in an infinite dimension space however more and more answers are emerging nowadays.
I would like to thank our teacher for opening our mathematical mind.

4 Annex

```
 data("tecator")
 names(tecator)

# create smoothing function from the beggining

x2 <- create.fdata.basis(tecator$absorp.fdata[1,])
x <- create.fourier.basis(c(1,100))
 X <- predict(x,absorp$data[1,])
aB <- solve(crossprod(X),crossprod(X,absorp$data[1,]))
smfunction <- fd(aB,x)

#Regression

 X <- tecator$absorp.fdata
 y <- tecator$y$Fat
 rang<- X$rangeval
 basisx <- create.bspline.basis(rang,nbasis=15)
 basisb <- create.bspline.basis(rang,nbasis=7)
 ans <- fregre.basis(X,y,basis.x=basisx,basis.b = basisb) # regression with no penalization
using b-spline basis
 plot(ans$beta.est,col="red",main="Beta function")

 ans2 <- fregre.basis.cv(X,y,lambda=TRUE) # regression with GCV criterium
ans2$gcv.opt
plot(ans2$beta.est,col=" red »,main="Beta function « )

x1 <- create.fdata.basis(tecator%absorp.fdata[1,]) # build the first functional x1
plot(x1)
```

5 Bibliography

http://web.math.ku.dk/noter/filer/phd16snm.pdf (1)
http://www-bcf.usc.edu/~gareth/research/AOS641.pdf (2)
http://www3.stat.sinica.edu.tw/sstest/oldpdf/A22n14.pdf (3)
http://anson.ucdavis.edu/~mueller/Review151106.pdf (4)

http://www.csc.kth.se/~szepessy/inversfor/marten.pdf (5)
http://www.csm.ro/reviste/Revue_Mathematique/pdfs/2010/6/Apostol.pdf (6)

https://www.rdocumentation.org/ (7)
https://www.rdocumentation.org/packages/fda.usc/versions/1.4.0/topics/fdata (8)

Multicolinéarité et régression PLS - Mistis (9)
https://www.rdocumentation.org/packages/fda.usc/versions/1.4.0/topics/tecator (10)
https://www.rdocumentation.org/packages/fda.usc/versions/1.4.0/topics/fregre.pls.cv (
(11)