Functional Data Analysis in R by fda.usc package

Manuel Oviedo de la Fuente
Universidade de Santiago de Compostela

I XORNADAS DE USUARIOS DE R EN GALICIA,
Santiago de Compostela, 10 de Outubro de 2013

State of the art

State of the art in Functional Data Analysis in R

- The package fda is a basic reference to work in R with functional data, [Ramsay and Silverman, 2005].
- [Ferraty and Vieu, (2006)] processed FD from a nonparametric point of view (normed or semi–normed functional spaces). http://www.lsp.ups-tlse.fr/staph/
- Other packages: the package ftsa for functional time series analysis, package fds with functional data sets and rainbow for functional data display and outlier detection. refund allows computing functional penalized regression.

Description Package: fda.usc

Title: Functional Data Analysis and Utilities for Statistical Computing (fda.usc)

Version: 1.0.5;

Date: 2013-06-01

Author: Manuel Febrero-Bande, Manuel Oviedo de la Fuente

Maintainer: Manuel Oviedo de la Fuente <manuel.oviedo@usc.es>

Depends: R (>= 2.10), fda, splines, MASS, mgcv

Description: This package carries out exploratory and descriptive analysis of functional data exploring its most important features: such as depth measurements or functional outliers detection, among others. It also helps to explain and model the relationship between a dependent variable and independent (regression models) and make predictions. Methods for supervised or unsupervised classification of a set of functional data regarding a feature of the data are also included. Finally, it can perform analysis of variance model (ANOVA) for functional data.

URL: `http://www.jstatsoft.org/v51/i04/`

Some definitions of Functional Data

Functional data analysis is a branch of statistics that analyzes data providing information about curves, surfaces or anything else varying over a continuum. The continuum is often time, but may also be spatial location, wavelength, probability, etc.
Functional data analysis is a branch of statistics concerned with analysing data in the form of functions, [Ferraty and Vieu, (2006)].

- **Definition 2.1**. A random variable $\mathcal{X}$ is called a functional variable if it takes values in a functional space $\mathcal{E}$ –complete normed (or seminormed) space–.
- **Definition 2.2**. A functional dataset $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ is the observation of n functional variables $\mathcal{X}_1, \ldots, \mathcal{X}_n$ identically distributed as $\mathcal{X}$.

Example of functional dataset in fda.usc

Load the library fda.usc [Febrero-Bande and Oviedo de la Fuente (2012)]

```
library(fda.usc)
```

Aemet data. Series of daily summaries of 73 spanish weather stations selected for the period 1980-2009. The dataset contains geographic information of each station and the average for the period 1980-2009 of dayly temperature, daily precipitation and daily wind speed. Meteorological State Agency of Spain (AEMET),Government of Spain.

```
data(aemet)
names(aemet)

## [1] "df"         "temp"       "wind.speed" "logprec"

names(aemet$df)

## [1] "ind"       "name"      "province"  "altitude"  "year.ini"  "year.end"
## [7] "longitude" "latitude"

class(aemet$temp)

## [1] "fdata"
```

In fda.usc: "The data are curves". An object called *fdata* as a list of the following components:

- **data**: typically a matrix of *(n x m)* dimension which contains a set of *n* curves discretized in *m* points or **argvals**.
- **argvals**: locations of the discretization points, by default: $\{t_1 = 1, \ldots, t_m = \mathbf{m}\}$.
- **rangeval**: rangeval of discretization points.
- **names**: (optional) list with three components: main, an overall title, xlab, a title for the x axis and ylab, a title for the y axis.

```
names(aemet$temp)

## [1] "data"     "argvals"  "rangeval" "names"

dim(aemet$temp)

## [1]  73 365

head(aemet$temp$argvals)

## [1] 0.5 1.5 2.5 3.5 4.5 5.5

aemet$tem$rangeval

## [1]   0 365
```

S3 method for class 'fdata'

Basic operations for fdata class objects:

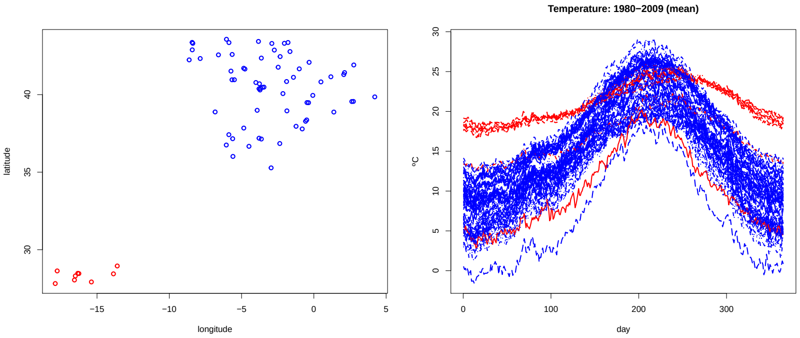
- Group “Math”: abs, sign, sqrt, floor, ceiling, trunc, round, signif, exp, log, expm1, log1p, cos, sin, tan, acos, asin, atan cosh, sinh, tanh, acosh, asinh, atanh, lgamma, gamma, digamma, trigamma, cumsum, cumprod, cummax, cummin.
- Group “Ops”: “+”, “-”, “*”, “/”, “ ^ ”, “%%”, “%/%”, “&”, “|”, “!”, “==”, “!=”, “<”, “<=”, “>=”.
- Group “Summary”: all, any, sum, prod, min, max, range.
- Other operations:“**[]**”, **is.fdata()**, **c()**, **dim()**, **ncol()**, **nrow()**.

```
range(aemet$temp)

## [1] -1.613 29.054

is.fdata(aemet$df)

## [1] FALSE

is.fdata(aemet$temp)

## [1] TRUE
```

In fda.usc: "The data are curves".

$X_i(t)$ represents the mean temperature (averaged over 1980-2009 years) at the ith weather station in Spain, and at time time t during the year.

```
par(mfrow = c(1, 2))
col1 = ifelse(aemet$df$latitude < 31, "red", "blue")
plot(aemet$df[, c("longitude", "latitude")], col = col1, lwd = 2)
plot(aemet$temp, col = col1, lwd = 2)
```



Some utilities of fda.usc package: Convert the class

- The new class **fdata** only uses the evaluations at the discretization points.
- The **fdata2fd()** converts **fdata** object to **fd** object (using the basis representation).
- Inversely, the **fdata()** converts object of class: **fd**, **fds**, **fts**, **sfts**, **vector**, **matrix**, **data.frame** to an object of class **fdata**.

```
temp.fd = fdata2fd(aemet$temp, type.basis = "fourier", nbasis = 15)
temp.fdata = fdata(temp.fd)  #back to fdata
class(temp.fd)

## [1] "fd"

names(temp.fd)

## [1] "coefs"   "basis"   "fdnames"

class(temp.fdata)

## [1] "fdata"
```

Resume by smoothing

If we supposed that our functional data $Y(t)$ is observed through the model and $\varepsilon(t)$ is the noise the regression is: $Y(t_i) = X(t_i) + \varepsilon(t_i)$
We can to get back the original signal $X(t)$ using a linear smoother

$$\hat{X}(t_i) = \sum_{i=1}^{n} s_i(t_j)Y(t_i) \Rightarrow \hat{\mathbf{X}} = \mathbf{SY}$$

where $s_i(t_j)$ is the weight that the point t_j gives to the point t_i.

Basis Let $X(t) \in \mathcal{L}_2$,

$$X(t) = \sum_{k \in \mathbb{N}} c_k \phi_k(t) \approx \sum_{k=1}^{K} c_k \phi_k(t) = c^\top \mathbf{\Phi}$$

The smoothing matrix is given by: $\mathbf{S} = \mathbf{\Phi}(\mathbf{\Phi}^\top W \mathbf{\Phi} + \lambda R)^{-1} \mathbf{\Phi}^\top W$ where λ is the penalty parameter.

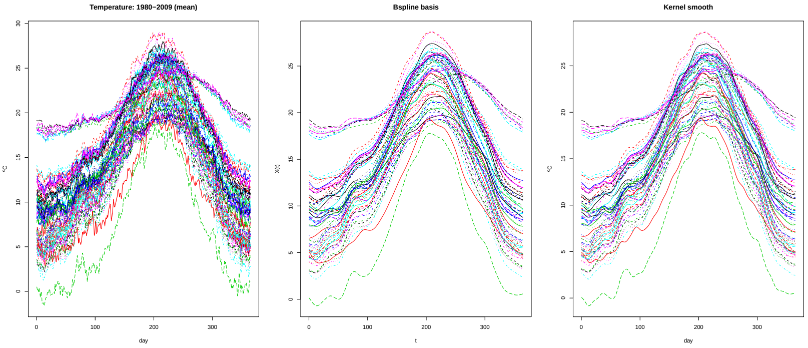
Kernel

$$\mathbf{S}_h = (s_{ij}) = \frac{K\left(\frac{t_i - t_j}{h}\right)}{\sum_{i=1}^{\top} K\left(\frac{t_i - t_j}{h}\right)}$$

where h is the bandwidth and $K()$ the Kernel function such as Normal Kernel, cosine Kernel, Epanechnikov Kernel, Triweight Kernel, Quartic Kernel.

Raw data, Smoothing by Fixed Basis and Kernel

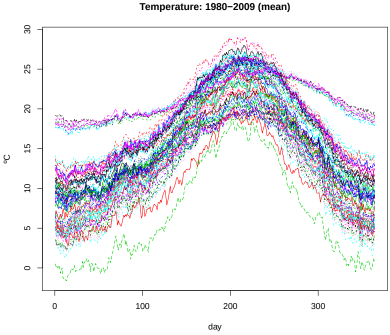
```
par(mfrow = c(1, 3))
plot(aemet$temp)
plot(min.basis(aemet$temp, lambda = 1000)$fdata.est, main = "Bspline basis")
plot(min.np(aemet$temp)$fdata.est, main = "Kernel smooth")
```

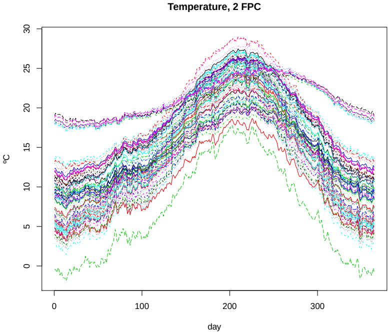


Data-driven basis: Principal Components

Optimize the amount of variance in the data.

```
par(mfrow = c(1, 2))
plot(aemet$temp)
plot((pc <- create.pc.basis(aemet$temp, l = 1:2))[[1]], main = "Temperature, 2 FPC")
```





Data-driven basis: Principal Components

```
pc <- fdata2pc(aemet$temp)
summary(pc)

##
##      - SUMMARY:   fdata2pc  object   -
##
## -With 2  components are explained  98.78 %
##  of the variability of explicative variables.
##
## -Variability for each component (%):
##    PC1   PC2
## 85.57 13.21
```

Smoothing: How estimate the smoothing parameter?

Using the validation criteria: CV or GCV criteria.

- Cross Validation (CV):

$$CV(\nu) = \frac{1}{n}\sum_{i=1}^{n} \frac{\left(y_i - \hat{r}_i^{\nu}(x_i)\right)^2}{1 - S_{ii}} w(x_i)$$

- Generalized Cross Validation (GCV):

$$GCV(\nu) = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \hat{r}_i^{\nu}(x_i)\right)^2 w(x_i)\Xi$$

Definition

- Introduction
- Resume by smoothing

Exploring Functional Data

- Tools for summarize functional data
- Functional Outlier Detection

Functional regression models (FRM)

- FRM with functional covariate
- Nonparametric approach
- FRM with functional and non functional covariate
- Generalized regression model

Functional classification

- Supervised classification

Depth measures

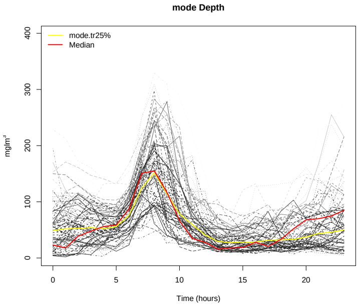
Tools for summarize functional data are included: **func.mean()**, **func.var()**, **fdata2pc()** for computing the mean, the marginal variance, principal eigenfunctions. This package includes tools based on depth measures [Cuevas et al., 2007].

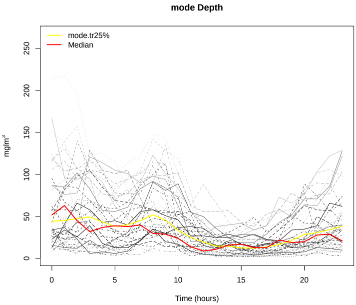
The depth is a concept emerged in the literature of robustness which measures how deep (or central) is a datum respect to a population.

- **depth.FM()**: the depth measure is based on the median, [Fraiman and Muniz, 2001].
- **depth.mode()**: the depth measure is based on how surrounded the curves are respect to a metric or a semimetric distance.
- **depth.RP()**: the depth measure is calculated through random projections (RP).
- **depth.RPD()**: the depth measure is calculated through random projections of the curves and theirs derivatives.

Poblenou data set collects 115 curves of NO_x levels measured every hour $\{t_i\}_{0:23}$ in Poblenou (Barcelona).

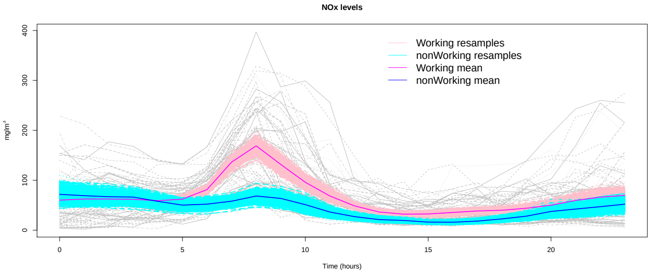
```
par(mfrow = c(1, 2))
data(poblenou)
dd <- as.integer(poblenou$df$day.week)
working = poblenou$nox[poblenou$df$day.festive == 0 & dd < 6]
nonworking = poblenou$nox[poblenou$df$day.festive == 1 | dd > 5]
depth.mode(working, draw = T)
depth.mode(nonworking, draw = T)
```





Bootstrap replies

The dispersion of a central tendency of a functional data can be estimated by smoothed bootstrap, [Cuevas et al., 2006].

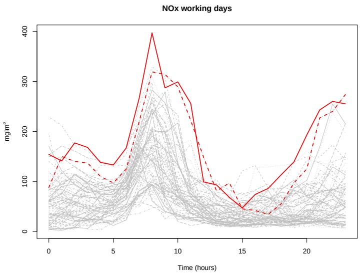


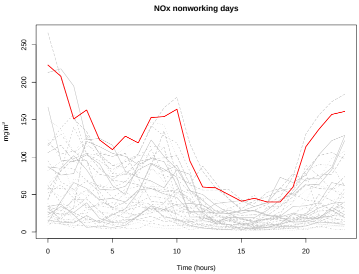
Functional Outliers: data(poblenou)

In order to identify outliers in functional datasets, [Febrero-Bande, et al., 2008] make use that depth and outlyingness are inverse notions, so that if an outlier is in the dataset, the corresponding curve will have a significant low depth.

- **outliers.depth.pond()**, based on weighting.
- **outliers.depth.trim()**, based on trimming.

```
out = outliers.depth.trim(working, dfunc = depth.FM, nb = 100, smo = 0.1, trim = 0.05)
out2 = outliers.depth.trim(nonworking, dfunc = depth.FM, nb = 100, smo = 0.1,
    trim = 0.05)
```





Electric hourly demand (in Spain)

FRM in fda.usc

Parametric approach:

- FRM with basis representation with a functional covariate.
- Functional PC with a functional covariate.
- Functional PLS with a functional covariate.
- Functional LM with more than one functional and nonfunctional covariate.

Other approach:

- Nonparametric (NP) functional regression via kernel smoothing with a functional covariate.
- Semi-partial linear model (PLM) with a functional and nonfunctional covariate.

Generalized approach:

- Generalized Functional Linear Model (GFLM).
- Generalized Functional Spectral Additive Model (GFSAM).
- Generalized Functional Kernel Additive Model (GFKAM).

FRM with functional covariate

This section assumes that the relationship between the scalar response y and the functional covariate $X(t)$ has a linear structure.

$$y_i = \langle X, \beta \rangle + \varepsilon_i = \int_T X_i(t)\beta(t)dt + \varepsilon_i$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product on $\mathcal{L}_2$

- [Ramsay and Silverman, 2005] uses fixed basis representation of $X(t)$ and $\beta(t)$: B–spline, Fourier, Wavelets..
- [Cardot et al., 1999] uses so-called functional principal components regression (FPC).
- [Preda et al., 2007] uses so-called functional partial least squares components regression (FPLS).

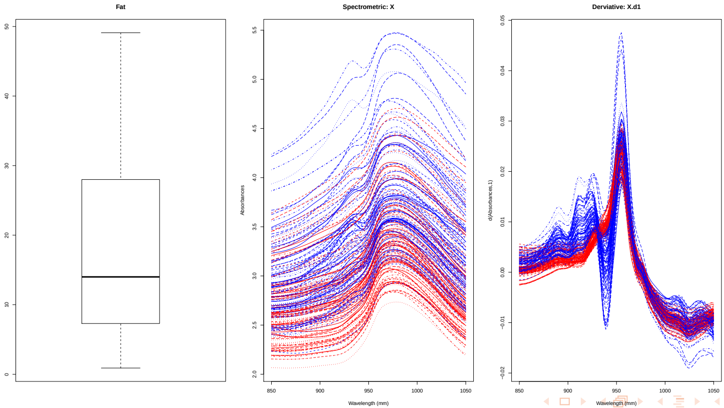
Parameter selection

- Choice the number of basis elements (Predictive Cross–Validation, Model Selection criteria (AIC),....).
- (optionally) [Ramsay and Silverman, 2005] uses a roughness penalty (λ), [Cardot et al., 1 uses the ridge parameter r_n in (FPC) and Kramer 2011 uses nonlinear regression methods based on PLS and Penalization Techniques.

Example tecator dataset

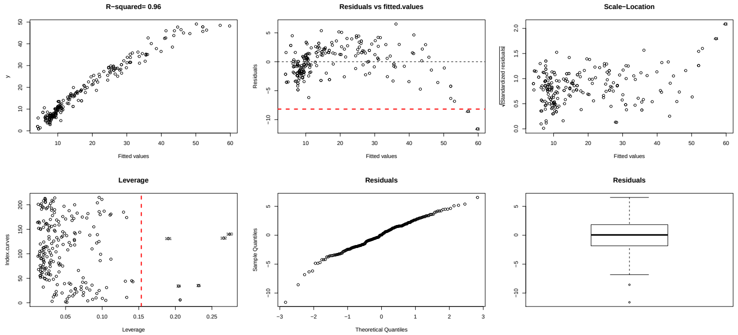
215 spectrometric curves of meat with Fat, Water and Protein contents. Explain the fat content through spectrometric curves.

```
data(tecator)
par(mfrow = c(1, 3))
fat15 <- ifelse((y <- tecator$y$Fat) < 15, 2, 4)
boxplot(y, main = "Fat")
plot((X <- tecator$absorp), col = fat15, main = "Spectrometric: X")
plot((X.d1 <- fdata.deriv(tecator$absorp, 1)), col = fat15, main = "Derviative: X.d1")
```



Example tecator dataset: regression

```
summary(fregre.basis(X.d1, y))
```




```
# fregre.pc(X.d1,y);fregre.pls(X.d1,y);fregre.np(X.d1,y)
```

fregre.np Nonparametric functional model with functional covariate $\mathcal{X}$, [Ferraty and Vieu, (2006)]:

$$y_i = r(\mathcal{X}_i(t)) + \varepsilon_i$$

where the unknown smooth real function r is estimated using kernel estimation by means of:

$$\hat{r}(X) = \frac{\sum_{i=1}^{n} K(h^{-1} d(\mathcal{X}, \mathcal{X}_i)) y_i}{\sum_{i=1}^{n} K(h^{-1} d(\mathcal{X}, \mathcal{X}_i))}$$

where K is an asymmetric kernel function, h is the smoothing parameter and d is a metric or a semimetric.

fregre.plm Semi-partial linear model (PLM) with a functional $\mathcal{X}$ and nonfunctional $\mathbb{Z} = \{Z_j\}_{j=1}^{J}$ covariates, [Aneiros-Pérez and Vieu, (2006)]:

$$y_i = f(\mathbb{Z}, \mathcal{X}(t)) + \varepsilon_i = \mathbb{Z}_i \beta + r(\mathcal{X}_i(t)) + \varepsilon_i$$

r is estimated by means of: $\hat{r}_h(t) = \sum_{i=1}^{n} w_{n,h}(t, X_i)(Y_i - Z_i^T \hat{\beta}_h)$
where $w_{n,h}(t, X_i) = \frac{K(d(t,\mathcal{X}_i)/h)}{\sum_{j=1}^{n} K(d(t,\mathcal{X}_j)/h)}$

Arguments for fregre.np() and fregre.plm() function

1. **Ker** = type of asymmetric kernel function, by default asymmetric normal kernel (cosine, epanechnicov, quadratic,....).
2. **metric**: type of metric or semimetric, by default $\mathcal{L}_2$ (**metric.lp(...,p=2)**).
   - **metric.lp()**
   - **semimetric.basis()**
   - **semimetric.pca()**
   - **semimetric.mplsr()**
   - **semimetric.deriv()**
   - **semimetric.fourier()**
3. **type.S**: type of smoothing matrix **S**: **S:NW**, **S.LLR**, **S.KNN**.

```
np <- fregre.np(X, y, metric = semimetric.deriv, nderiv = 1, type.S = S.KNN)
plm <- fregre.plm(Fat ~ Water + absorp.fdata, tecator, metric = semimetric.deriv,
    nderiv = 1, type.S = S.KNN)
```

```
c(np$h.opt, plm$h.opt)

## [1] 5 5

c(np$r2, plm$r2)

## [1] 0.9183 0.9853
```

Again, it has also implemented the function **fregre.np.cv** to estimate the smoothing parameter h by the validation criteria.

Functional Linear Model: fregre.lm() function

The scalar response y is estimated by functional $\{\mathcal{X}_q(t)\}_{q=1}^{Q}$ and also non–functional $\mathbb{Z} = \{Z_j\}_{j=1}^{J}$ covariates by:

$$y_i = \alpha + \mathbb{Z}_i\beta + \sum_{q=1}^{Q} \langle \mathcal{X}_i^q(t), \beta_q(t) \rangle + \varepsilon_i$$

where ε_i are random errors with mean zero and finite variance σ^2.

For Tecator data example, the content of **Fat** is estimated from absorbances curves and the content of **Water** by **fregre.lm** function.

```
f = Fat ~ Water + absorp.fdata
basis.x1 = list(absorp.fdata = create.pc.basis(X, l = c(1, 3)))
fregre.lm(f, tecator, basis.x = basis.x1)

##
## Call:
## lm(formula = pf, data = XX)
##
## Coefficients:
##     (Intercept)             Water  absorp.fdata.PC1  absorp.fdata.PC3
##         103.928            -1.357            -0.121             1.984
```

The fitted object returned can be used in other functions of the *"lm"* class such as: **summary**, **coefficients** or **predict**, among other (S3 methods).

One natural extension of LM model is the generalized functional linear regression model (GFLM) [Müller and Stadtmüller, 2005] which allows various types of the response. In the GLM framework it is generally assumed that $y_i|X_i$ can be chosen within the set of distributions belonging to the exponential family with probability density function:

$$f(\theta, \phi, y) = exp\left\{ \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi) \right\} \tag{1}$$

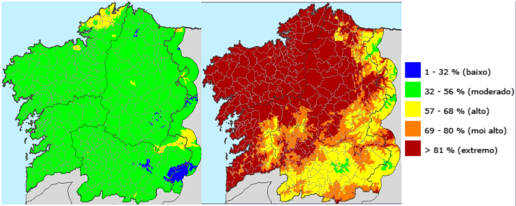
where ϕ represents a scale parameter and θ is the canonical parameter of distribution and the functions $a()$, $b()$ and $c()$ are known. The model is specified as follows:

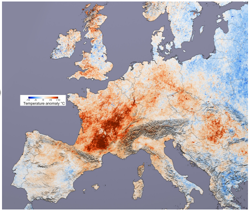
$$E[y|X] = b'(\theta) = \mu$$
$$Var[y|X] = b''(\theta)a(\phi) = V(\mu)\phi$$
$$g(\mu) = \left(\int_T X\beta + dt \right) + Z\beta$$

where μ is the expected value of response, $g()$ is the link function that specified the dependence between μ and the regressors, $V[\mu]$ is the conditional variance.

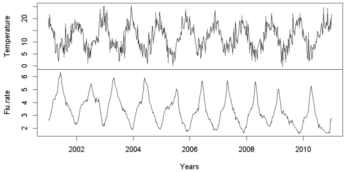
Distribution	ψ	$E(\mu)$	$V(\mu)$	Canonical link; in **R**
Binomial/n	1/n	μ	$\mu(1-\mu)$	$log(\mu(1-\mu))$; **logit**
Poisson	1	μ	μ	$log(\mu)$; **log**
Negative Binomial	1	$log\left(\frac{\mu}{1+1/\phi}\right)$	$\mu + \frac{\mu^2}{\phi}$	$log(\mu(\phi+\mu))$; **log**
Normal	σ^2	μ	1	μ; **identity**
Gamma	$1/\upsilon$	$-1/\upsilon$	μ^2	μ^{-1}; **inverse**

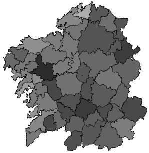
Table : Principal distributions used in GLMs.

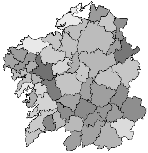




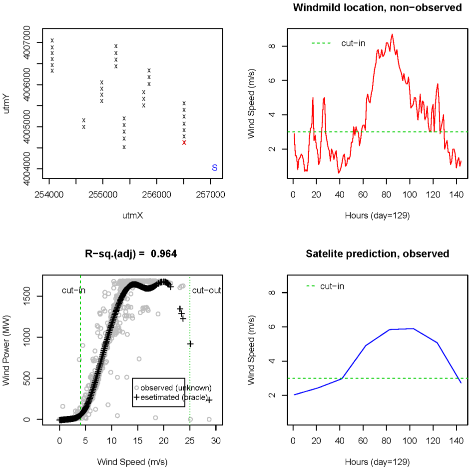
Distribution	Topic	Response	Covariate
Binomial/n	Wildfires	Fire risk	Temperature
Poisson	Epidemiology–Heat wave	Mortality incidence	Temperature
Negative Binomial	Epidemiology–epidemic	Flu incidence	Temperature
Normal	Health risks	Air pollution level	Temperature

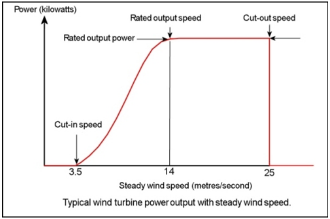






Wind power


Windmild location, non−observed
utmY
utmX
S
cut−in
Wind Speed (m/s)
Hours (day=129)
R−sq.(adj) = 0.964
cut−in
cut−out
observed (unknown)
esetimated (oracle)
Wind Power (MW)
Wind Speed (m/s)
Satelite prediction, observed
cut−in
Wind Speed (m/s)
Hours (day=129)


Power (kilowatts)
Rated output speed
Cut-out speed
Rated output power
Cut-in speed
3.5
14
25
Steady wind speed (metres/second)
Typical wind turbine power output with steady wind speed.

Generalized Functional Linear Model

The scalar response y is estimated by functional $\{\mathcal{X}_q(t)\}_{q=1}^{Q}$ and also non–functional $\mathbb{Z} = \{Z_j\}_{j=1}^{J}$ covariates by:

Generalized Functional Linear Model (FGLM), Müller and Stadtmüller,(2005),

$$y_i = g^{-1}\left(\alpha + \mathbb{Z}_i\beta + \sum_{q=1}^{Q}\left\langle \mathcal{X}_i^q(t), \beta_q(t)\right\rangle\right) + \varepsilon_i$$

$g()$ is the inverse link function and ε_i are random errors with mean zero and finite variance σ^2.

```
f = fat15 ~ Water + absorp.fdata
tecator$df$fat15 <- ifelse(tecator$df$Fat < 15, 0, 1)
fregre.glm(fat15 ~ absorp.d1, data = tecator, family = binomial())

##
## Call:  glm(formula = pf)
##
## Coefficients:
##      (Intercept)  absorp.d1.bspl4.1  absorp.d1.bspl4.2
##             8.82            -720.85             890.75
## absorp.d1.bspl4.3  absorp.d1.bspl4.4  absorp.d1.bspl4.5
##          -720.80             362.44            -184.16
##
## Degrees of Freedom: 214 Total (i.e. Null);  209 Residual
## Null Deviance:     298
## Residual Deviance: 28.5  AIC: 40.5
```

Generalized Functional Additive Model

fregre.gsam Generalized Functional Spectral Additive Linear Model (FGSAM), Müller and Yao, (2008),

$$y_i = g^{-1}\left(\alpha + \sum_{j=1}^{J} f_j\left(Z_i^j\right) + \sum_{q=1}^{Q} s_q\left(\mathcal{X}_i^q(t)\right)\right) + \varepsilon_i$$

where $f(\cdot), s(\cdot)$ are the smoothed functions.

```
fregre.gsam(fat15 ~ s(absorp.d1), data = tecator, family = binomial())
```

fregre.gkam Generalized Functional Kernel Additive Linear Model (FGKAM), Febrero–Bande and Gonzalez–Manteiga, (2012),

$$y_i = g^{-1}\left(\alpha + \sum_{q=1}^{Q} \mathcal{K}\left(\mathcal{X}_i^q(t)\right)\right) + \varepsilon_i,$$

where $\mathcal{K}(\cdot)$ is the kernel estimator.

```
fregre.gkam(fat15 ~ absorp.d1, data = tecator, family = binomial(), control = list(maxit = 10))
```

Functional supervised classification

Let $\{X_i, y_i\}_{i=1}^{n}$ be a sample of n independent pairs, identically distrbuted as (χ, Y).

Aim: How predict the class y_i of a functional variable X_i

Bayes rule: Given a sample X, the aim is to estimate the posterior probability of belonging to each group:

$$p_g(X) = P(y = g \,|\chi = X)$$

The classification rule is to assign a new functional observation that group with the maximum a posteriori probability.

$$\hat{y} = arg\ max \hat{p}_g(X)$$

The estimate of the posterior probability $p_g(X)$ can be calculated by

- k-Nearest Neighbor Classifier: **classif.knn**
- Kernel Classifier: **classif.kernel**
- Logistic Classifier (linear model): **classif.glm**
- Logistic Classifier (additive model): **classif.gsam** and **classif.gkam**
- Distance Classifier: **classif.dist**
- Maximum Depth Classifier: **classif.depth**

classification

```
out <- classif.glm(fat15 ~ absorp.d1, data = tecator)
summary(out)

##      - SUMMARY -
##
## -Probability of correct classification by group (prob.classification):
##      0      1
## 0.9554 0.9709
##
## -Confusion matrix between the theoretical groups (by rows)
##   and estimated groups (by column)
##
##       0   1
##   0 107   5
##   1   3 100
##
## -Probability of correct classification:  0.9628
```

Example Phoneme dataset

The data represent the amplitudes, plotted against time, for 250 recordings of men pronouncing five phonemes (50 recordings of each phoneme): "sh" 1 as in "she", "iy" 2 as the vowel in "she", "dcl" 3 as in "dark","aa" 4 as the vowel in "dark", and "ao" 5 as the first vowel in "water".

Functional supervised classification

```
summary.classif(out1 <- classif.knn(glearn, mlearn, knn = 3))

##      - SUMMARY -
##
## -Probability of correct classification by group (prob.classification):
##  Warning: h.opt is the minimum value of bandwidths
##    provided, range(h)= 3 3
## y
##    1    2    3    4    5
## 1.00 0.96 1.00 0.72 0.74
##
## -Confusion matrix between the theoretical groups (by rows)
##   and estimated groups (by column)
##
##      1  2  3  4  5
##   1 50  0  0  0  0
##   2  0 48  2  0  0
##   3  0  0 50  0  0
##   4  0  0  0 36 14
##   5  0  0  1 12 37
##
## -Vector of probability of correct classification
##    by number of neighbors (knn):
##     3
## 0.884
##
## -Optimal number of neighbors: knn.opt= 3
## with highest probability of correct classification max.prob=  0.884
```

Functional supervised classification

Once these classifiers have been trained, they can be used to predict on new functional data.

```
mtest <- phoneme[["test"]]
gtest <- phoneme[["classtest"]]
pred1 = predict.classif(out1, mtest)
table(pred1, gtest)

##      gtest
## pred1  1  2  3  4  5
##     1 50  0  0  0  0
##     2  0 49  0  0  0
##     3  0  1 50  0  0
##     4  0  0  0 42 15
##     5  0  0  0  8 35
```

Also implemented in fda.usc

- FLM: penalized, influence measures, Goodness-of-fit test and bootstrap
- Functional nonsupervised classification (clustering))
- Functional ANOVA

Incomming

- Functional Data Definition (wavelet basis, 1d curve, 2d surface, 3d)
- Functional Data Representation and registration, sparse data
- Functional Regression with Functional Response
- Functional Regression with Dependent Errors: temporal, spatial or both
- Functional Supervised Classification: DDplot

Aneiros-Pérez G and Vieu P (2006). Semi-functional partial linear regression. *Statistics and Probability Letters*, **76**, 1102-1110.

Cardot H, Ferraty F and Sarda P (1999). Functional Linear Model. *Statistics and Probability Letters*, **45**(1), 11-22.

Cuevas A, Febrero-Bande M and Fraiman R (2006). On the use of bootstrap for estimating functions with functional data. *Computational Statistics and Data Analysis*, **51**(2), 1063-1074.

Cuevas A, Febrero-Bande M and Fraiman R (2007). Robust estimation and classification for functional data via projection-based depth notions. *Computational Statistics*, **22**(3), 481-496.

Febrero-Bande M, Galeano P and González-Manteiga W (2008). Outlier detection in functional data by depth measures with application to identify abnormal NOx levels. *Environmetrics*, **19**:331-345.

Febrero-Bande M, Galeano P and González-Manteiga W (2010). Measures of influence for the functional linear model with scalar response. *Journal of Multivariate Analysis*, **101**(2), 327-339.

Ferraty F and Vieu P (2006). *Nonparametric functional data analysis.* Springer Series in Statistics, New York.

- Fraiman R and Muniz G (2007). Trimmed means for functional data. *Test*, **10**, 419-440.
- Preda C, Saporta G, Lévéder CL. (2007).
  PLS classification of functional data.
  Comput. Stat, **22**(2), 223–235.
- Ramsay JO and Silverman BW (2005). *Applied Functional Data Analysis*. 2nd ed., Springer, New York.
- Ramsay JO and Silverman BW (2006). *Functional Data Analysis*. 2nd ed., Springer, New York.
- Müller, H.G. and Yao, F. (2008),
  Functional additive models. Journal of the American Statistical Association 103, 1534–1544.
- Febrero-Bande, M. and Gonzalez-Manteiga, W. (2012).
  Generalized Additive Models for Functional Data. TEST. Springer-Velag.
- Müller, H.G. and Stadtmüller, U. (2005).
  Generalized functional linear models. Ann Stat, 33, 774–805.