Package 'fdANOVA'

September 27, 2026

Type Package

Title Analysis of Variance for Univariate and Multivariate Functional Data

Version 0.1.3

Description Performs analysis of variance testing procedures for univariate and multivariate functional data (Cuesta-Albertos and Febrero-Bande (2010) <doi:10.1007/s11749-010-0185-3>, Gorecki and Smaga (2015) <doi:10.1007/s00180-015-0555-0>, Gorecki and Smaga (2017) <doi:10.1080/02664763.2016.1247791>, Zhang et al. (2018) <doi:10.1016/j.csda.2018.05.0

License LGPL-2 | LGPL-3 | GPL-2 | GPL-3

Depends R (>= 2.10)

Imports fda, doParallel, ggplot2, doBy, MASS, magic, parallel, foreach

Encoding UTF-8

NeedsCompilation no

Author Tomasz Gorecki [aut], Lukasz Smaga [aut, cre]

Maintainer Lukasz Smaga <ls@amu.edu.pl>

Repository CRAN

Date/Publication 2026-09-27 09:10:34 UTC

Contents

fanova.tests 2
fmanova.ptbfr 8
fmanova.trp 11
plot.fanovatests 14
plot.fmanovatrp 15
plotFANOVA 17
print.fanovatests 19
print.fmanovaptbfr 20
print.fmanovatrp 21
summary.fanovatests 22
summary.fmanovaptbfr 23
summary.fmanovatrp 24

Index **26**

`fanova.tests` *Tests for FANOVA Problem*

Description

Performs the testing procedures for the one-way analysis of variance for (univariate) functional data (FANOVA). See Section 2.1 of the vignette file (`vignette("fdANOVA", package = "fdANOVA")`), for details of the tests.

We consider the l groups of independent random functions $X_{ij}(t)$, $i = 1, \ldots, l$, $j = 1, \ldots, n_i$ defined over a closed and bounded interval $I = [a, b]$. Let $n = n_1 + \cdots + n_l$. These groups may differ in mean functions, i.e., we assume that $X_{ij}(t)$, $j = 1, \ldots, n_i$ are stochastic processes with mean function $\mu_i(t)$, $t \in I$ and covariance function $\gamma(s, t)$, $s, t \in I$, for $i = 1, \ldots, l$. Of interest is to test the following null hypothesis

$$H_0 : \mu_1(t) = \cdots = \mu_l(t),\ t \in I.$$

The alternative is the negation of the null hypothesis. We assume that each functional observation is observed on a common grid of $\mathcal{T}$ design time points equally spaced in I (see Section 3.1 of the vignette file, `vignette("fdANOVA", package = "fdANOVA")`).

Usage

```
fanova.tests(x = NULL, group.label, test = "ALL",
             params = NULL,
             parallel = FALSE, nslaves = NULL)

# more detailed usage of params:
# params = list(paramFP = list(int, B.FP = 1000,
#                              basis = c("Fourier", "b-spline", "own"),
#                              own.basis, own.cross.prod.mat,
#                             criterion = c("BIC", "eBIC", "AIC", "AICc", "NO"),
#                              commonK = c("mode", "min", "max", "mean"),
#                         minK = NULL, maxK = NULL, norder = 4, gamma.eBIC = 0.5)
#               paramCH = 10000,
#               paramCS = 10000,
#               paramL2b = 10000,
#               paramFb = 10000,
#               paramFmaxb = 10000,
#               paramTRP = list(k = 30, projection = c("GAUSS", "BM"),
#                               permutation = FALSE, B.TRP = 10000,
#                               independent.projection.tests = TRUE))
```

Arguments

`x`	a $\mathcal{T} \times n$ matrix of data, whose each column is a discretized version of a function and rows correspond to design time points. Its default values is `NULL`, since if

the FP test is only used, we can give a basis representation of the data instead of raw observations (see the list `paramFP` below). For any of the other testing procedures, the raw data are needed.

`group.label` a vector containing group labels.

`test` a kind of indicator which establishes a choice of FANOVA tests to be performed. Its default value means that all testing procedures of Section 2.1 of the vignette file will be used. When we want to use only some tests, the parameter `test` is an appropriate subvector of the following vector of tests' labels `c("FP", "CH", "CS", "L2N", "L2B", "L2b", "FN", "FB", "Fb", "GPF", "Fmaxb", "TRP")`, where `"FP"` - permutation test based on basis function representation (Gorecki and Smaga, 2015); `"CH"` and `"CS"` - L2-norm-based parametric bootstrap tests for homoscedastic and heteroscedastic samples, respectively (Cuevas et al., 2004); `"L2N"` and `"L2B"` - L2-norm-based test with naive and bias-reduced method of estimation, respectively (Faraway, 1997; Zhang and Chen, 2007; Zhang, 2013); `"L2b"` - L2-norm-based bootstrap test (Zhang, 2013); `"FN"` and `"FB"` - F-type test with naive and bias-reduced method of estimation, respectively (Shen and Faraway, 2004; Zhang, 2011); `"Fb"` - F-type bootstrap test (Zhang, 2013); `"GPF"` - globalizing the pointwise F-test (Zhang and Liang, 2014); `"Fmaxb"` - Fmax bootstrap test (Zhang et al., 2018); `"TRP"` - tests based on random projections (Cuesta-Albertos and Febrero-Bande, 2010).

`params` a list of additional parameters for the FP, CH, CS, L^2b, Fb, Fmaxb tests and the tests based on random projections. It can contain all or a part of the elements `paramFP`, `paramCH`, `paramCS`, `paramL2b`, `paramFb`, `paramFmaxb` and `paramTRP` for passing the parameters for the FP, CH, CS, L^2b, Fb, Fmaxb tests and tests based on random projections, respectively, to the function `fanova.tests`. They are described below. The default value of `params` means that these tests are performed with their default values.

`paramFP` a list containing the parameters for the FP test.

`int` a vector of two elements representing the interval $I = [a, b]$. When it is not specified, it is determined by a number of design time points.

`B.FP` a number of permutation replicates for the FP tests.

`basis` a choice of basis of functions used in the basis function representation of the data.

`own.basis` if `basis = "own"`, a $K \times n$ matrix with columns containing the coefficients of the basis function representation of the observations.

`own.cross.prod.mat`
if `basis = "own"`, a $K \times K$ cross product matrix corresponding to a basis used to obtain the matrix `own.basis`.

`criterion` a choice of information criterion for selecting the optimum value of K.
`criterion = "NO"` means that K is equal to the parameter `maxK` defined below. We have

$$\texttt{BIC}(X_{ij}) = \mathcal{T}\log(\mathbf{e}_{ij}^\top \mathbf{e}_{ij}/\mathcal{T}) + K\log\mathcal{T},$$

$$\texttt{eBIC}(X_{ij}) = \mathcal{T}\log(\mathbf{e}_{ij}^\top \mathbf{e}_{ij}/\mathcal{T}) + K[\log\mathcal{T} + 2\gamma\log(K_{\max})],$$

$$\texttt{AIC}(X_{ij}) = \mathcal{T}\log(\mathbf{e}_{ij}^\top \mathbf{e}_{ij}/\mathcal{T}) + 2K$$

and

$$\texttt{AICc}(X_{ij}) = \texttt{AIC}(X_{ij}) + 2K(K+1)/(n-K-1),$$

where

$$\mathbf{e}_{ij} = (e_{ij1}, \ldots, e_{ij\mathcal{T}})^\top,$$

$$e_{ijr} = X_{ij}(t_r) - \sum_{m=1}^{K} \hat{c}_{ijm}\varphi_m(t_r),$$

$t_1, \ldots, t_\mathcal{T}$ are the design time points, $\gamma \in [0, 1]$, $K_{\max}$ is a maximum K considered and $\log$ denotes the natural logarithm.

`commonK` a choice of method for selecting the common value for all observations from the values of K corresponding to all processes.

`minK` a minimum value of K. When `basis = "Fourier"`, it has to be an odd number. If `minK = NULL`, we take `minK = 3`. For `basis = "b-spline"`, `minK` has to be greater than or equal to `norder` defined below. If `minK = NULL` or `minK < norder`, then we take `minK = norder`.

`maxK` a maximum value of K. When `basis = "Fourier"`, it has to be an odd number. If `maxK = NULL`, we take `maxK` equal to the largest odd number smaller than the number of design time points. If `maxK` is greater than or equal to the number of design time points, `maxK` is taken as above. For `basis = "b-spline"`, `maxK` has to be smaller than or equal to the number of design time points. If `maxK = NULL` or `maxK` is greater than the number of design time points, then we take `maxK` equal to the number of design time points.

`norder` if `basis = "b-spline"`, an integer specifying the order of b-splines.

`gamma.eBIC` a $\gamma \in [0, 1]$ parameter in the eBIC.

`paramCH` a number of discretized artificial trajectories for generating Gaussian processes for the CH test.

`paramCS` a number of discretized artificial trajectories for generating Gaussian processes for the CS test.

`paramL2b` a number of bootstrap samples for the L^2b test.

`paramFb` a number of bootstrap samples for the Fb test.

`paramFmaxb` a number of bootstrap samples for the Fmaxb test.

`paramTRP` a list containing the parameters of the tests based on random projections.

`k` a vector of numbers of projections.

`projection` a method of generating Gaussian processes in step 1 of the tests based on random projections presented in Section 2 of the vignette file. If `projection = "GAUSS"`, the Gaussian white noise is generated as in the function `anova.RPm` from the R package **fda.usc**. In the second case, the Brownian motion is generated.

`permutation` a logical indicating whether to compute p-values of the tests based on random projections by permutation method.

`B.TRP` a number of permutation replicates for the tests based on random projections.

`independent.projection.tests`
: a logical indicating whether to generate the random projections independently or dependently for different elements of vector `k`. In the first case, the random projections for each element of vector `k` are generated separately, while in the second one, they are generated as chained subsets, e.g., for `k = c(5, 10)`, the first 5 projections are a subset of the last 10. The second way of generating random projections is faster than the first one.

`parallel`
: a logical indicating whether to use parallelization.

`nslaves`
: if `parallel = TRUE`, a number of slaves. Its default value means that it will be equal to a number of logical processes of a computer used.

Details

To perform step 3 of the projection procedure given in Section 2.1 of the vignette file, we use five tests: the standard (`paramTRP$permutation = FALSE`) and permutation (`paramTRP$permutation = TRUE`) tests based on ANOVA F-test statistic and ANOVA-type statistic (ATS) proposed by Brunner et al. (1997), as well as the testing procedure based on Wald-type permutation statistic (WTPS) of Pauly et al. (2015).

Value

A list with class "fanovatests" containing the following components (|k| denotes the length of vector `k`):

`FP`
: a list containing value of test statistic `statFP`, p-value `pvalueFP` and used parameters for the FP test. The chosen optimal length of basis expansion `K` is also given there.

`CH`
: a list containing value of test statistic `statCH`, p-value `pvalueCH` and used parameter `paramCH` for the CH test.

`CS`
: a list containing value of test statistic `statCS`, p-value `pvalueCS` and used parameter `paramCS` for the CS test.

`L2N`
: a list containing value of test statistic `statL2`, p-value `pvalueL2N` and values of estimators `betaL2N` and `dL2N` used in approximation of null distribution of test statistic for the L^2N test.

`L2B`
: a list containing value of test statistic `statL2`, p-value `pvalueL2B` and values of estimators `betaL2B` and `dL2B` used in approximation of null distribution of test statistic for the L^2B test.

`L2b`
: a list containing value of test statistic `statL2`, p-value `pvalueL2b` and used parameter `paramL2b` for the L^2b test.

`FN`
: a list containing value of test statistic `statF`, p-value `pvalueFN` and values of estimators `d1FN` and `d2FN` used in approximation of null distribution of test statistic for the FN test.

`FB`
: a list containing value of test statistic `statF`, p-value `pvalueFB` and values of estimators `d1FB` and `d2FB` used in approximation of null distribution of test statistic for the FB test.

`Fb`
: a list containing value of test statistic `statF`, p-value `pvalueFb` and used parameter `paramFb` for the Fb test.

GPF	a list containing value of test statistic statGPF, p-value pvalueGPF and values of estimators betaGPF and dGPF used in approximation of null distribution of test statistic for the GPF test.
Fmaxb	a list containing value of test statistic statFmax, p-value pvalueFmaxb and used parameter paramFmaxb for the Fmaxb test.
TRP	a list containing the following elements: vectors pvalues.anova, pvalues.ATS, pvalues.WTPS of length |k| containing p-values for tests based on random projections and for numbers of projections given in k; if independent.projection.tests = TRUE, a list data.projections of length |k|, whose ith element is an $n\times$ k[i] matrix with columns being projections of the data; when independent.projection.tests = FALSE, an $n \times \max(\mathsf{k})$ matrix data.projections with columns being projections of the data; used parameters for the tests based on random projections.

and the values of other used parameters: data = x, group.label, etc.

Author(s)

Tomasz Gorecki, Lukasz Smaga

References

Brunner E, Dette H, Munk A (1997). Box-Type Approximations in Nonparametric Factorial Designs. Journal of the American Statistical Association 92, 1494-1502.

Cuesta-Albertos JA, Febrero-Bande M (2010). A Simple Multiway ANOVA for Functional Data. Test 19, 537-557.

Cuevas A, Febrero M, Fraiman R (2004). An Anova Test for Functional Data. Computational Statistics & Data Analysis 47, 111-122.

Faraway J (1997). Regression Analysis for a Functional Response. Technometrics 39, 254-261.

Gorecki T, Smaga L (2015). A Comparison of Tests for the One-Way ANOVA Problem for Functional Data. Computational Statistics 30, 987-1010.

Gorecki T, Smaga L (2017). Multivariate Analysis of Variance for Functional Data. Journal of Applied Statistics 44, 2172-2189.

Pauly M, Brunner E, Konietschke F (2015). Asymptotic Permutation Tests in General Factorial Designs. Journal of the Royal Statistical Society Series B 77, 461-473.

Shen Q, Faraway J (2004). An F Test for Linear Models with Functional Responses. Statistica Sinica 14, 1239-1257.

Zhang JT (2011). Statistical Inferences for Linear Models with Functional Responses. Statistica Sinica 21, 1431-1451.

Zhang JT (2013). Analysis of Variance for Functional Data. Chapman & Hall, London.

Zhang JT, Chen JW (2007). Statistical Inferences for Functional Data. The Annals of Statistics 35, 1052-1079.

Zhang JT, Cheng MY, Wu HT, Zhou B (2018). A New Test for Functional One-way ANOVA with Applications to Ischemic Heart Screening. Computational Statistics and Data Analysis https://doi.org/10.1016/j.csda.2018.05

Zhang JT, Liang X (2014). One-Way ANOVA for Functional Data via Globalizing the Pointwise F-Test. Scandinavian Journal of Statistics 41, 51-71.

See Also

fmanova.ptbfr, fmanova.trp, plotFANOVA, plot.fanovatests

Examples

```
# Some of the examples may run some time.

# gait data (the first feature)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- as.matrix(gait.data.frame[, 1:39])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# all FANOVA tests with default parameters
set.seed(123)
(fanova1 <- fanova.tests(x = x.gait, group.label = group.label.gait))
summary(fanova1)
# data projections generated in the test based on random projections
fanova1$TRP$data.projections

# only three tests with non-default parameters
set.seed(123)
fanova2 <- fanova.tests(x.gait, group.label.gait,
                        test = c("FP", "GPF", "Fmaxb"),
                        params = list(paramFP = list(int = c(0.025, 0.975),
                                                     B.FP = 1000, basis = "b-spline",
                                                     criterion = "eBIC",
                                                     commonK = "mean",
                                                     minK = 5, maxK = 19,
                                                     norder = 4, gamma.eBIC = 0.7),
                                      paramFmaxb = 1000))
summary(fanova2)

# the FP test with predefined basis function representation
library(fda)
fbasis <- create.bspline.basis(rangeval = c(0.025, 0.975), 19, norder = 4)
own.basis <- Data2fd(seq(0.025, 0.975, length = 20), x.gait, fbasis)$coefs
own.cross.prod.mat <- inprod(fbasis, fbasis)
set.seed(123)
fanova3 <- fanova.tests(group.label = group.label.gait, test = "FP",
                        params = list(paramFP = list(B.FP = 1000, basis = "own",
                                                     own.basis = own.basis,
                                                   own.cross.prod.mat = own.cross.prod.mat)))
summary(fanova3)

# the tests based on random projections with the Gaussian white noise generated for projections
set.seed(123)
fanova4 <- fanova.tests(x.gait, group.label.gait, test = "TRP",
                        parallel = TRUE, nslaves = 2,
                        params = list(paramTRP = list(k = c(10, 20, 30), B.TRP = 1000)))
```

```
summary(fanova4)
set.seed(123)
fanova5 <- fanova.tests(x.gait, group.label.gait, test = "TRP",
                        parallel = TRUE, nslaves = 2,
                        params = list(paramTRP = list(k = c(10, 20, 30),
                                                      permutation = TRUE, B.TRP = 1000)))
summary(fanova5)

# the tests based on random projections with the Brownian motion generated for projections
set.seed(123)
fanova6 <- fanova.tests(x.gait, group.label.gait, test = "TRP",
                        parallel = TRUE, nslaves = 2,
                      params = list(paramTRP = list(k = c(10, 20, 30), projection = "BM",
                                                    B.TRP = 1000)))
summary(fanova6)
set.seed(123)
fanova7 <- fanova.tests(x.gait, group.label.gait, test = "TRP",
                        parallel = TRUE, nslaves = 2,
                      params = list(paramTRP = list(k = c(10, 20, 30), projection = "BM",
                                                    permutation = TRUE, B.TRP = 1000)))
summary(fanova7)
```

`fmanova.ptbfr` *Permutation Tests Based on a Basis Function Representation for FMANOVA Problem*

Description

Performs the permutation tests based on a basis function representation for multivariate analysis of variance for functional data, i.e., the W, LH, P and R tests. For details of the tests, see Section 2.2 of the vignette file (`vignette("fdANOVA", package = "fdANOVA")`).

We consider independent vectors of random functions

$$\mathbf{X}_{ij}(t) = (X_{ij1}(t), \ldots, X_{ijp}(t))^\top \in SP_p(\boldsymbol{\mu}_i, \boldsymbol{\Gamma}),$$

$i = 1, \ldots, l$, $j = 1, \ldots, n_i$ defined over the interval $I = [a, b]$, where $SP_p(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ is a set of p-dimensional stochastic processes with mean vector $\boldsymbol{\mu}(t)$, $t \in I$ and covariance function $\boldsymbol{\Gamma}(s, t)$, $s, t \in I$. Let $n = n_1 + \cdots + n_l$. In the multivariate analysis of variance problem for functional data (FMANOVA), we have to test the null hypothesis as follows:

$$H_0 : \boldsymbol{\mu}_1(t) = \cdots = \boldsymbol{\mu}_l(t),\ t \in I.$$

The alternative is the negation of the null hypothesis. We assume that each functional observation is observed on a common grid of $\mathcal{T}$ design time points equally spaced in I (see Section 3.1 of the vignette file, `vignette("fdANOVA", package = "fdANOVA")`).

Usage

```
fmanova.ptbfr(x = NULL, group.label, int, B = 1000,
              parallel = FALSE, nslaves = NULL,
              basis = c("Fourier", "b-spline", "own"),
              own.basis, own.cross.prod.mat,
              criterion = c("BIC", "eBIC", "AIC", "AICc", "NO"),
              commonK = c("mode", "min", "max", "mean"),
              minK = NULL, maxK = NULL, norder = 4, gamma.eBIC = 0.5)
```

Arguments

x	a list of $\mathcal{T} \times n$ matrices of data, whose each column is a discretized version of a function and rows correspond to design time points. The mth element of this list contains the data of mth feature, $m = 1, \ldots, p$. Its default values is NULL, because a basis representation of the data can be given instead of raw observations (see the parameter own.basis below).
group.label	a vector containing group labels.
int	a vector of two elements representing the interval $I = [a, b]$. When it is not specified, it is determined by a number of design time points.
B	a number of permutation replicates.
parallel	a logical indicating whether to use parallelization.
nslaves	if parallel = TRUE, a number of slaves. Its default value means that it will be equal to a number of logical processes of a computer used.
basis	a choice of basis of functions used in the basis function representation of the data.
own.basis	if basis = "own", a list of length p, whose elements are $K_m \times n$ matrices ($m = 1, \ldots, p$) with columns containing the coefficients of the basis function representation of the observations.
own.cross.prod.mat	if basis = "own", a $KM \times KM$ cross product matrix corresponding to a basis used to obtain the list own.basis.
criterion	a choice of information criterion for selecting the optimum value of K_m, $m = 1, \ldots, p$. criterion = "NO" means that K_m are equal to the parameter maxK defined below. Further remarks about this argument are the same as for the function fanova.tests.
commonK	a choice of method for selecting the common value for all observations from the values of K_m corresponding to all processes.
minK	a minimum value of K_m. Further remarks about this argument are the same as for the function fanova.tests.
maxK	a maximum value of K_m. Further remarks about this argument are the same as for the function fanova.tests.
norder	if basis = "b-spline", an integer specifying the order of b-splines.
gamma.eBIC	a $\gamma \in [0, 1]$ parameter in the eBIC.

Value

A list with class "fmanovaptbfr" containing the following components:

`W`	a value of the statistic W.
`pvalueW`	p-value for the W test.
`LH`	a value of the statistic LH.
`pvalueLH`	p-value for the LH test.
`P`	a value of the statistic P.
`pvalueP`	p-value for the P test.
`R`	a value of the statistic R.
`pvalueR`	p-value for the R test,

the values of parameters used and eventually

`data`	a list containing the data given in `x`.
`Km`	a vector $(K_1, \ldots, K_p)$.
`KM`	a maximum of a vector `Km`.

Author(s)

Tomasz Gorecki, Lukasz Smaga

References

Gorecki T, Smaga L (2015). A Comparison of Tests for the One-Way ANOVA Problem for Functional Data. Computational Statistics 30, 987-1010.

Gorecki T, Smaga L (2017). Multivariate Analysis of Variance for Functional Data. Journal of Applied Statistics 44, 2172-2189.

See Also

`fanova.tests`, `fmanova.trp`, `plotFANOVA`

Examples

```
# Some of the examples may run some time.

# gait data (both features)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- vector("list", 2)
x.gait[[1]] <- as.matrix(gait.data.frame[, 1:39])
x.gait[[2]] <- as.matrix(gait.data.frame[, 40:78])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# the tests based on a basis function representation with default parameters
```

```
set.seed(123)
(fmanova1 <- fmanova.ptbfr(x.gait, group.label.gait))
summary(fmanova1)

# the tests based on a basis function representation with non-default parameters
set.seed(123)
fmanova2 <- fmanova.ptbfr(x.gait, group.label.gait, int = c(0.025, 0.975), B = 5000,
                          basis = "b-spline", criterion = "eBIC", commonK = "mean",
                          minK = 5, maxK = 19, norder = 4, gamma.eBIC = 0.7)
summary(fmanova2)

# the tests based on a basis function representation
# with predefined basis function representation
library(fda)
fbasis <- create.fourier.basis(c(0, nrow(x.gait[[1]])), 17)
own.basis <- vector("list", 2)
own.basis[[1]] <- Data2fd(1:nrow(x.gait[[1]]), x.gait[[1]], fbasis)$coefs
own.basis[[2]] <- Data2fd(1:nrow(x.gait[[2]]), x.gait[[2]], fbasis)$coefs
own.cross.prod.mat <- diag(rep(1, 17))
set.seed(123)
fmanova3 <- fmanova.ptbfr(group.label = group.label.gait,
                          B = 1000, basis = "own",
                          own.basis = own.basis,
                          own.cross.prod.mat = own.cross.prod.mat)
summary(fmanova3)

library(fda)
fbasis <- create.bspline.basis(c(0, nrow(x.gait[[1]])), 19, norder = 4)
own.basis <- vector("list", 2)
own.basis[[1]] <- Data2fd(1:nrow(x.gait[[1]]), x.gait[[1]], fbasis)$coefs
own.basis[[2]] <- Data2fd(1:nrow(x.gait[[2]]), x.gait[[2]], fbasis)$coefs
own.cross.prod.mat <- inprod(fbasis, fbasis)
set.seed(123)
fmanova4 <- fmanova.ptbfr(group.label = group.label.gait,
                          B = 1000, basis = "own",
                          own.basis = own.basis,
                          own.cross.prod.mat = own.cross.prod.mat)
summary(fmanova4)
```

`fmanova.trp` *Tests Based on Random Projections for FMANOVA Problem*

Description

Performs the testing procedures based on random projections for multivariate analysis of variance for functional data, i.e., the Wp, LHp, Pp and Rp tests. For details of the tests, see Section 2.2 of the vignette file (`vignette("fdANOVA", package = "fdANOVA")`).

We consider independent vectors of random functions

$$\mathbf{X}_{ij}(t) = (X_{ij1}(t), \ldots, X_{ijp}(t))^\top \in SP_p(\boldsymbol{\mu}_i, \boldsymbol{\Gamma}),$$

$i = 1, \ldots, l$, $j = 1, \ldots, n_i$ defined over the interval $I = [a, b]$, where $SP_p(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ is a set of p-dimensional stochastic processes with mean vector $\boldsymbol{\mu}(t)$, $t \in I$ and covariance function $\boldsymbol{\Gamma}(s, t)$, $s, t \in I$. Let $n = n_1 + \cdots + n_l$. In the multivariate analysis of variance problem for functional data (FMANOVA), we have to test the null hypothesis as follows:

$$H_0 : \boldsymbol{\mu}_1(t) = \cdots = \boldsymbol{\mu}_l(t),\ t \in I.$$

The alternative is the negation of the null hypothesis. We assume that each functional observation is observed on a common grid of $\mathcal{T}$ design time points equally spaced in I (see Section 3.1 of the vignette file, `vignette("fdANOVA", package = "fdANOVA")`).

Usage

```
fmanova.trp(x, group.label, k = 30, projection = c("GAUSS", "BM"),
            permutation = FALSE, B = 1000,
            independent.projection.tests = TRUE,
            parallel = FALSE, nslaves = NULL)
```

Arguments

`x`	a list of $\mathcal{T} \times n$ matrices of data, whose each column is a discretized version of a function and rows correspond to design time points. The mth element of this list contains the data of mth feature, $m = 1, \ldots, p$.
`group.label`	a vector containing group labels.
`k`	a vector of numbers of projections.
`projection`	a method of generating Gaussian processes in step 1 of the test based on random projections presented in Section 2 of the vignette file. If `projection = "GAUSS"`, the Gaussian white noise is generated as in the function `anova.RPm` from the R package **fda.usc**. In the second case, the Brownian motion is generated.
`permutation`	a logical indicating whether to compute p-values by permutation method.
`B`	a number of permutation replicates.
`independent.projection.tests`	a logical indicating whether to generate the random projections independently or dependently for different elements of vector `k`. In the first case, the random projections for each element of vector `k` are generated separately, while in the second one, they are generated as chained subsets, e.g., for `k = c(5, 10)`, the first 5 projections are a subset of the last 10. The second way of generating random projections is faster than the first one.
`parallel`	a logical indicating whether to use parallelization.
`nslaves`	if `parallel = TRUE`, a number of slaves. Its default value means that it will be equal to a number of logical processes of a computer used.

Value

A list with class "fmanovatrp" containing the following components (|k| denotes the length of vector k):

`pvalues`	a $4\times$ |k| matrix of p-values of the tests.

`data.projections`
: if `independent.projection.tests = TRUE`, a list of length |k|, whose elements are lists of $n \times p$ matrices of projections of the observations, while when `independent.projection.tests = FALSE`, a list of length $\max(k)$, whose elements are $n \times p$ matrices of projections of the observations,

and the values of used parameters.

Author(s)

Tomasz Gorecki, Lukasz Smaga

References

Cuesta-Albertos JA, Febrero-Bande M (2010). A Simple Multiway ANOVA for Functional Data. Test 19, 537-557.

Gorecki T, Smaga L (2017). Multivariate Analysis of Variance for Functional Data. Journal of Applied Statistics 44, 2172-2189.

See Also

`fanova.tests`, `fmanova.ptbfr`, `plotFANOVA`, `plot.fmanovatrp`

Examples

```
# Some of the examples may run some time.

# gait data (both features)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- vector("list", 2)
x.gait[[1]] <- as.matrix(gait.data.frame[, 1:39])
x.gait[[2]] <- as.matrix(gait.data.frame[, 40:78])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# the tests based on random projections with the Gaussian white noise generated for projections
set.seed(123)
fmanova1 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20))
summary(fmanova1)
fmanova2 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20),
                        permutation = TRUE, B = 1000,
                        parallel = TRUE, nslaves = 2)
summary(fmanova2)

# the tests based on random projections with the Brownian motion generated for projections
set.seed(123)
fmanova3 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20), projection = "BM")
summary(fmanova3)
fmanova4 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20), projection = "BM",
                        permutation = TRUE, B = 1000,
```

```
                        parallel = TRUE, nslaves = 2)
summary(fmanova4)
```

`plot.fanovatests` *Plot P-values of Tests Based on Random Projections for FANOVA Problem*

Description

The p-values of the tests based on random projections for the one-way analysis of variance for (univariate) functional data against the number of projections are plotted.

Usage

```
## S3 method for class 'fanovatests'
plot(x, y, ...)
```

Arguments

`x`	an "`fanovatests`" object. More precisely, a result of the function `fanova.tests` for the standard tests based on random projections.
`y`	an "`fanovatests`" object. More precisely, a result of the function `fanova.tests` for the permutation tests based on random projections.
`...`	additional arguments not used.

Note

We can use only one of the arguments `x` and `y`, or both simultaneously.

Author(s)

Tomasz Gorecki, Lukasz Smaga

See Also

`fanova.tests`, `plot.fmanovatrp`

Examples

```
# Some of the examples may run some time.

# gait data (the first feature)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- as.matrix(gait.data.frame[, 1:39])

# vector of group labels
```

```
group.label.gait <- rep(1:3, each = 13)

set.seed(123)
fanova4 <- fanova.tests(x.gait, group.label.gait, test = "TRP",
                        parallel = TRUE, nslaves = 2,
                        params = list(paramTRP = list(k = c(10, 20, 30), B.TRP = 1000)))
set.seed(123)
fanova5 <- fanova.tests(x.gait, group.label.gait, test = "TRP",
                        parallel = TRUE, nslaves = 2,
                        params = list(paramTRP = list(k = c(10, 20, 30),
                                                      permutation = TRUE, B.TRP = 1000)))
plot(x = fanova4)
plot(y = fanova5)
plot(x = fanova4, y = fanova5)

set.seed(123)
fanova6 <- fanova.tests(x.gait, group.label.gait, test = "TRP",
                        parallel = TRUE, nslaves = 2,
                       params = list(paramTRP = list(k = c(10, 20, 30), projection = "BM",
                                                     B.TRP = 1000)))
set.seed(123)
fanova7 <- fanova.tests(x.gait, group.label.gait, test = "TRP",
                        parallel = TRUE, nslaves = 2,
                       params = list(paramTRP = list(k = c(10, 20, 30), projection = "BM",
                                                     permutation = TRUE, B.TRP = 1000)))
plot(x = fanova6)
plot(y = fanova7)
plot(x = fanova6, y = fanova7)
```

`plot.fmanovatrp` *Plot P-values of Tests Based on Random Projections for FMANOVA Problem*

Description

The p-values of the tests based on random projections for multivariate analysis of variance for functional data against the number of projections are plotted.

Usage

```
## S3 method for class 'fmanovatrp'
plot(x, y, withoutRoy = FALSE, ...)
```

Arguments

`x`	an "`fmanovatrp`" object. More precisely, a result of the function `fmanova.trp` for the standard tests based on random projections.
`y`	an "`fmanovatrp`" object. More precisely, a result of the function `fmanova.trp` for the permutation tests based on random projections.

withoutRoy	a logical indicating whether to plot the p-values of the Rp test.
...	additional arguments not used.

Note

We can use only one of the arguments x and y, or both simultaneously.

Author(s)

Tomasz Gorecki, Lukasz Smaga

See Also

fmanova.trp, plot.fanovatests

Examples

```
# Some of the examples may run some time.

# gait data (both features)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- vector("list", 2)
x.gait[[1]] <- as.matrix(gait.data.frame[, 1:39])
x.gait[[2]] <- as.matrix(gait.data.frame[, 40:78])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# the tests based on random projections with the Gaussian white noise generated for projections
set.seed(123)
fmanova1 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20))
fmanova2 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20),
                        permutation = TRUE, B = 1000, parallel = TRUE, nslaves = 2)
plot(x = fmanova1)
plot(x = fmanova1, withoutRoy = TRUE)
plot(y = fmanova2)
plot(x = fmanova1, y = fmanova2)
plot(x = fmanova1, y = fmanova2, withoutRoy = TRUE)

# the tests based on random projections with the Brownian motion generated for projections
set.seed(123)
fmanova3 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20), projection = "BM")
fmanova4 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20), projection = "BM",
                        permutation = TRUE, B = 1000, parallel = TRUE, nslaves = 2)
plot(x = fmanova3)
plot(x = fmanova3, withoutRoy = TRUE)
plot(y = fmanova4)
plot(x = fmanova3, y = fmanova4)
plot(x = fmanova3, y = fmanova4, withoutRoy = TRUE)
```

plotFANOVA *Plot Univariate Functional Data*

Description

Univariate functional observations with or without indication of groups as well as mean functions of samples are plotted. We assume that n univariate functional observations are observed on a common grid of $\mathcal{T}$ design time points equally spaced in $I = [a, b]$ (see Section 3.1 of the vignette file, `vignette("fdANOVA", package = "fdANOVA")`).

Usage

```
plotFANOVA(x, group.label = NULL, int = NULL, separately = FALSE,
           means = FALSE, smooth = FALSE, ...)
```

Arguments

`x`	a $\mathcal{T} \times n$ matrix of data, whose each column is a discretized version of a function and rows correspond to design time points.
`group.label`	a character vector containing group labels. Its default value means that all functional observations are drawn without division into groups.
`int`	a vector of two elements representing the interval $I = [a, b]$. When it is not specified, it is determined by a number of design time points.
`separately`	a logical indicating how groups are drawn. If `separately = FALSE`, groups are drawn on one plot by different colors. When `separately = TRUE`, they are depicted in different panels.
`means`	a logical indicating whether to plot only group mean functions.
`smooth`	a logical indicating whether to plot reconstructed data via smoothing splines instead of raw data.
`...`	additional arguments not used.

Author(s)

Tomasz Gorecki, Lukasz Smaga

See Also

`fanova.tests`, `fmanova.ptbfr`, `fmanova.trp`

Examples

```
# Some of the examples may run some time.

# gait data (both features)
library(fda)
gait.data.frame <- as.data.frame(gait)
```

```
x.gait <- vector("list", 2)
x.gait[[1]] <- as.matrix(gait.data.frame[, 1:39])
x.gait[[2]] <- as.matrix(gait.data.frame[, 40:78])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

plotFANOVA(x = x.gait[[1]], int = c(0.025, 0.975))
plotFANOVA(x = x.gait[[1]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975))
plotFANOVA(x = x.gait[[1]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), separately = TRUE)
plotFANOVA(x = x.gait[[1]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), means = TRUE)

plotFANOVA(x = x.gait[[1]], int = c(0.025, 0.975), smooth = TRUE)
plotFANOVA(x = x.gait[[1]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), smooth = TRUE)
plotFANOVA(x = x.gait[[1]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), separately = TRUE, smooth = TRUE)
plotFANOVA(x = x.gait[[1]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), means = TRUE, smooth = TRUE)

plotFANOVA(x = x.gait[[2]], int = c(0.025, 0.975))
plotFANOVA(x = x.gait[[2]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975))
plotFANOVA(x = x.gait[[2]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), separately = TRUE)
plotFANOVA(x = x.gait[[2]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), means = TRUE)

plotFANOVA(x = x.gait[[2]], int = c(0.025, 0.975), smooth = TRUE)
plotFANOVA(x = x.gait[[2]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), smooth = TRUE)
plotFANOVA(x = x.gait[[2]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), separately = TRUE, smooth = TRUE)
plotFANOVA(x = x.gait[[2]], group.label = as.character(group.label.gait),
           int = c(0.025, 0.975), means = TRUE, smooth = TRUE)

# Canadian Weather data (both features)
library(fda)
x.CW <- vector("list", 2)
x.CW[[1]] <- CanadianWeather$dailyAv[,,1]
x.CW[[2]] <- CanadianWeather$dailyAv[,,2]

# vector of group labels
group.label.CW <- rep(c("Eastern", "Western", "Northern"), c(15, 15, 5))

plotFANOVA(x = x.CW[[1]])
plotFANOVA(x = x.CW[[1]], group.label = as.character(group.label.CW))
plotFANOVA(x = x.CW[[1]], group.label = as.character(group.label.CW),
           separately = TRUE)
plotFANOVA(x = x.CW[[1]], group.label = as.character(group.label.CW),
```

```
           means = TRUE)

plotFANOVA(x = x.CW[[1]], smooth = TRUE)
plotFANOVA(x = x.CW[[1]], group.label = as.character(group.label.CW),
           smooth = TRUE)
plotFANOVA(x = x.CW[[1]], group.label = as.character(group.label.CW),
           separately = TRUE, smooth = TRUE)
plotFANOVA(x = x.CW[[1]], group.label = as.character(group.label.CW),
           means = TRUE, smooth = TRUE)

plotFANOVA(x = x.CW[[2]])
plotFANOVA(x = x.CW[[2]], group.label = as.character(group.label.CW))
plotFANOVA(x = x.CW[[2]], group.label = as.character(group.label.CW),
           separately = TRUE)
plotFANOVA(x = x.CW[[2]], group.label = as.character(group.label.CW),
           means = TRUE)

plotFANOVA(x = x.CW[[2]], smooth = TRUE)
plotFANOVA(x = x.CW[[2]], group.label = as.character(group.label.CW),
           smooth = TRUE)
plotFANOVA(x = x.CW[[2]], group.label = as.character(group.label.CW),
           separately = TRUE, smooth = TRUE)
plotFANOVA(x = x.CW[[2]], group.label = as.character(group.label.CW),
           means = TRUE, smooth = TRUE)
```

`print.fanovatests` *Print "fanovatests" object*

Description

Prints the brief summary of the FANOVA tests.

Usage

```
## S3 method for class 'fanovatests'
print(x, ...)
```

Arguments

`x`	an "`fanovatests`" object.
`...`	additional arguments not used.

Details

The function prints out the values of test statistics and p-values of the FANOVA tests performed by the `fanova.tests` function.

Author(s)

Tomasz Gorecki, Lukasz Smaga

See Also

fanova.tests, summary.fanovatests

Examples

```
# This example may run some time.

# gait data (the first feature)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- as.matrix(gait.data.frame[, 1:39])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# all FANOVA tests with default parameters
set.seed(123)
fanova1 <- fanova.tests(x.gait, group.label.gait)
print(fanova1)
```

---

print.fmanovaptbfr *Print "fmanovaptbfr" object*

Description

Prints the brief summary of the permutation tests based on a basis function representation for multivariate analysis of variance for functional data, i.e., the W, LH, P and R tests.

Usage

```
## S3 method for class 'fmanovaptbfr'
print(x, ...)
```

Arguments

x	an "fmanovaptbfr" object.
...	additional arguments not used.

Details

The function prints out the values of test statistics and p-values of the permutation tests based on a basis function representation for FMANOVA problem performed by the fmanova.ptbfr function.

Author(s)

Tomasz Gorecki, Lukasz Smaga

See Also

fmanova.ptbfr, summary.fmanovaptbfr

Examples

```
# gait data (both features)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- vector("list", 2)
x.gait[[1]] <- as.matrix(gait.data.frame[, 1:39])
x.gait[[2]] <- as.matrix(gait.data.frame[, 40:78])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# the tests based on a basis function representation with default parameters
set.seed(123)
fmanova1 <- fmanova.ptbfr(x.gait, group.label.gait)
print(fmanova1)
```

---

print.fmanovatrp *Print "fmanovatrp" object*

Description

Prints the brief summary of the testing procedures based on random projections for multivariate analysis of variance for functional data, i.e., the Wp, LHp, Pp and Rp tests.

Usage

```
## S3 method for class 'fmanovatrp'
print(x, ...)
```

Arguments

x	an "fmanovatrp" object.
...	additional arguments not used.

Details

The function prints out the p-values of the tests based on random projections for FMANOVA problem performed by the fmanova.trp function.

Author(s)

Tomasz Gorecki, Lukasz Smaga

See Also

`fmanova.trp`, `summary.fmanovatrp`

Examples

```
# gait data (both features)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- vector("list", 2)
x.gait[[1]] <- as.matrix(gait.data.frame[, 1:39])
x.gait[[2]] <- as.matrix(gait.data.frame[, 40:78])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# the tests based on random projections with the Gaussian white noise generated for projections
set.seed(123)
fmanova1 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20))
print(fmanova1)
```

---

summary.fanovatests *Print "fanovatests" object*

Description

Prints the summary of the FANOVA tests.

Usage

```
## S3 method for class 'fanovatests'
summary(object, ...)
```

Arguments

`object`	an "`fanovatests`" object.
`...`	additional arguments not used.

Details

The function prints out the information about the data, the values of test statistics, the p-values and used parameters for the FANOVA tests performed by the `fanova.tests` function.

Author(s)

Tomasz Gorecki, Lukasz Smaga

See Also

fanova.tests, print.fanovatests

Examples

```
# This example may run some time.

# gait data (the first feature)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- as.matrix(gait.data.frame[, 1:39])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# all FANOVA tests with default parameters
set.seed(123)
fanova1 <- fanova.tests(x.gait, group.label.gait)
summary(fanova1)
```

---

summary.fmanovaptbfr *Print "fmanovaptbfr" object*

Description

Prints the summary of the permutation tests based on a basis function representation for multivariate analysis of variance for functional data, i.e., the W, LH, P and R tests.

Usage

```
## S3 method for class 'fmanovaptbfr'
summary(object, ...)
```

Arguments

object	an "fmanovaptbfr" object.
...	additional arguments not used.

Details

The function prints out the information about the data, the values of test statistics, the p-values and used parameters for the permutation tests based on a basis function representation for FMANOVA problem performed by the fmanova.ptbfr function.

Author(s)

Tomasz Gorecki, Lukasz Smaga

See Also

fmanova.ptbfr, print.fmanovaptbfr

Examples

```
# gait data (both features)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- vector("list", 2)
x.gait[[1]] <- as.matrix(gait.data.frame[, 1:39])
x.gait[[2]] <- as.matrix(gait.data.frame[, 40:78])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# the tests based on a basis function representation with default parameters
set.seed(123)
fmanova1 <- fmanova.ptbfr(x.gait, group.label.gait)
summary(fmanova1)
```

---

summary.fmanovatrp *Print "fmanovatrp" object*

Description

Prints the summary of the testing procedures based on random projections for multivariate analysis of variance for functional data, i.e., the Wp, LHp, Pp and Rp tests.

Usage

```
## S3 method for class 'fmanovatrp'
summary(object, ...)
```

Arguments

object	an "fmanovatrp" object.
...	additional arguments not used.

Details

The function prints out the information about the data, the p-values and used parameters for the tests based on random projections for FMANOVA problem performed by the fmanova.trp function.

Author(s)

Tomasz Gorecki, Lukasz Smaga

See Also

fmanova.trp, print.fmanovatrp

Examples

```
# gait data (both features)
library(fda)
gait.data.frame <- as.data.frame(gait)
x.gait <- vector("list", 2)
x.gait[[1]] <- as.matrix(gait.data.frame[, 1:39])
x.gait[[2]] <- as.matrix(gait.data.frame[, 40:78])

# vector of group labels
group.label.gait <- rep(1:3, each = 13)

# the tests based on random projections with the Gaussian white noise generated for projections
set.seed(123)
fmanova1 <- fmanova.trp(x.gait, group.label.gait, k = c(1, 5, 10, 15, 20))
summary(fmanova1)
```

Index

∗ **Multivariate**
  fanova.tests, 2
  fmanova.ptbfr, 8
  fmanova.trp, 11
∗ **Plot**
  plot.fanovatests, 14
  plot.fmanovatrp, 15
  plotFANOVA, 17
∗ **Printing**
  print.fanovatests, 19
  print.fmanovaptbfr, 20
  print.fmanovatrp, 21
  summary.fanovatests, 22
  summary.fmanovaptbfr, 23
  summary.fmanovatrp, 24
∗ **Test**
  fanova.tests, 2
  fmanova.ptbfr, 8
  fmanova.trp, 11

fanova.tests, 2, *9, 10*, *13, 14*, *17*, *19, 20*, *22, 23*
fmanova.ptbfr, *7*, 8, *13*, *17*, *20, 21*, *23, 24*
fmanova.trp, *7*, *10*, 11, *15–17*, *21, 22*, *24, 25*

plot.fanovatests, *7*, 14, *16*
plot.fmanovatrp, *13, 14*, 15
plotFANOVA, *7*, *10*, *13*, 17
print.fanovatests, 19, *23*
print.fmanovaptbfr, 20, *24*
print.fmanovatrp, 21, *25*

summary.fanovatests, *20*, 22
summary.fmanovaptbfr, *21*, 23
summary.fmanovatrp, *22*, 24