Package 'rmfanova'

May 9, 2026

Type Package

Title Repeated Measures Functional Analysis of Variance

Version 0.1.0

Date 2023-06-09

Description
The provided package implements the statistical tests for the functional repeated measures analysis problem (Kurylo and Smaga, 2023, <doi:10.48550/arXiv.2306.03883>). These procedures enable us to verify the overall hypothesis regarding equality, as well as hypotheses for pairwise comparisons (i.e., post hoc analysis) of mean functions corresponding to repeated experiments.

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

Imports foreach, doParallel, MASS, refund

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Katarzyna Kurylo [aut],
Lukasz Smaga [aut, cre]

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

Repository CRAN

Date/Publication 2023-06-10 13:50:02 UTC

Contents

`pointwise_f_test_statistic` *Pointwise F-type test statistic*

Description

The function `pointwise_f_test_statistic()` calculates and draws the pointwise F-type test statistic.

Usage

```
pointwise_f_test_statistic(
  x,
  plot = TRUE,
  values = FALSE,
  type = "l",
  ylab = "",
  main = "F(t)",
  ...
)
```

Arguments

`x`	a list of length ℓ with elements being $n \times p$ matrices of data corresponding to n functional observations measured in p design time points under given experimental conditions.
`plot`	a logical indicating of whether to draw the values of the pointwise F-type test statistic. The default is `TRUE`.
`values`	a logical indicating of whether to return the values of the pointwise F-type test statistic. The default is `FALSE`.
`type`	1-character string giving the type of plot desired, the same as in the `plot()` function. The default is `"l"` for lines.
`ylab`	a label for the y-axis, the same as in the `plot()` function. The default is the empty sign.
`main`	a main title for the plot, the same as in the `plot()` function. The default is `F(t)`.
`...`	other graphical parameters, the same as in the `plot()` function.

Details

For details, see the documentation of the `rmfanova()` function or the paper Kurylo and Smaga (2023).

Value

If `values = TRUE`, a vector of values of the pointwise F-type test statistic.

References

Kurylo K., Smaga L. (2023) Functional repeated measures analysis of variance and its application. Preprint https://arxiv.org/abs/2306.03883

Examples

```
# preparation of the DTI data set, for details see Kurylo and Smaga (2023)
library(refund)
data(DTI)
# MS patients
DTI_ms <- DTI[DTI$case == 1, ]
miss_data <- c()
for (i in 1:340) if (any(is.na(DTI_ms$cca[i, ]))) miss_data <- c(miss_data, i)
DTI_ms <- DTI_ms[-miss_data, ]
DTI_ms_2 <- DTI_ms[DTI_ms$Nscans == 4, ]
xx <- vector("list", 4)
for (i in 1:4) {
  xx[[i]] <- DTI_ms_2$cca[DTI_ms_2$visit == i, ]
}
xx[[1]] <- xx[[1]][-14, ]
xx[[3]] <- xx[[3]][-14, ]
yy <- xx
for (i in seq_len(4)) yy[[i]] <- yy[[i]][1:17, ]
# pointwise F-type test statistic
pointwise_f_test_statistic(yy, xlab = "t", xaxt = "n")
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
```

`pointwise_sample_mean_fun`

Pointwise sample mean functions

Description

The function `pointwise_sample_mean_fun()` calculates and draws the pointwise sample mean functions.

Usage

```
pointwise_sample_mean_fun(
  x,
  plot = TRUE,
  values = FALSE,
  type = "l",
  lty = 1,
  main = "Sample mean functions",
  ...
)
```

Arguments

x	a list of length ℓ with elements being $n \times p$ matrices of data corresponding to n functional observations measured in p design time points under given experimental conditions.
plot	a logical indicating of whether to draw the values of the pointwise sample mean functions. The default is TRUE.
values	a logical indicating of whether to return the values of the pointwise sample mean functions. The default is FALSE.
type	1-character string giving the type of plot desired, the same as in the matplot() function. The default is "l" for lines.
lty	vector of line types, the same as in the matplot() function. The default is 1 (solid lines).
main	a main title for the plot, the same as in the plot() function. The default is Sample mean functions.
...	other graphical parameters, the same as in the matplot() function.

Value

If values = TRUE, a matrix of values of the pointwise sample mean functions.

References

Kurylo K., Smaga L. (2023) Functional repeated measures analysis of variance and its application. Preprint https://arxiv.org/abs/2306.03883

Examples

```
# preparation of the DTI data set, for details see Kurylo and Smaga (2023)
library(refund)
data(DTI)
# MS patients
DTI_ms <- DTI[DTI$case == 1, ]
miss_data <- c()
for (i in 1:340) if (any(is.na(DTI_ms$cca[i, ]))) miss_data <- c(miss_data, i)
DTI_ms <- DTI_ms[-miss_data, ]
DTI_ms_2 <- DTI_ms[DTI_ms$Nscans == 4, ]
xx <- vector("list", 4)
for (i in 1:4) {
  xx[[i]] <- DTI_ms_2$cca[DTI_ms_2$visit == i, ]
}
xx[[1]] <- xx[[1]][-14, ]
xx[[3]] <- xx[[3]][-14, ]
yy <- xx
for (i in seq_len(4)) yy[[i]] <- yy[[i]][1:17, ]
# sample mean functions
oldpar <- par(mfrow = c(1, 1), mar = c(4, 4, 2, 0.1))
pointwise_sample_mean_fun(yy, values = FALSE,
                          col = 1:4, xlab = "t", ylab = "FA", xaxt = "n")
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
```

```
legend(x = 36, y = 0.64, legend = 1:4, lty = 1, col = 1:4, title = "Visit")
par(oldpar)
```

`pointwise_ssa_test_statistic` *Pointwise SSA test statistic*

Description

The function `pointwise_ssa_test_statistic()` calculates and draws the pointwise SSA test statistic.

Usage

```
pointwise_ssa_test_statistic(
  x,
  plot = TRUE,
  values = FALSE,
  type = "l",
  ylab = "",
  main = "SSA(t)",
  ...
)
```

Arguments

`x`	a list of length ℓ with elements being $n \times p$ matrices of data corresponding to n functional observations measured in p design time points under given experimental conditions.
`plot`	a logical indicating of whether to draw the values of the pointwise SSA test statistic. The default is `TRUE`.
`values`	a logical indicating of whether to return the values of the pointwise SSA test statistic. The default is `FALSE`.
`type`	1-character string giving the type of plot desired, the same as in the `plot()` function. The default is `"l"` for lines.
`ylab`	a label for the y-axis, the same as in the `plot()` function. The default is the empty sign.
`main`	a main title for the plot, the same as in the `plot()` function. The default is `SSA(t)`.
`...`	other graphical parameters, the same as in the `plot()` function.

Details

For details, see the documentation of the `rmfanova()` function or the paper Kurylo and Smaga (2023).

Value

If `values = TRUE`, a vector of values of the pointwise SSA test statistic.

References

Martinez-Camblor P., Corral N. (2011) Repeated Measures Analysis for Functional Data. Computational Statistics & Data Analysis 55, 3244–3256.

Kurylo K., Smaga L. (2023) Functional repeated measures analysis of variance and its application. Preprint https://arxiv.org/abs/2306.03883

Examples

```
# preparation of the DTI data set, for details see Kurylo and Smaga (2023)
library(refund)
data(DTI)
# MS patients
DTI_ms <- DTI[DTI$case == 1, ]
miss_data <- c()
for (i in 1:340) if (any(is.na(DTI_ms$cca[i, ]))) miss_data <- c(miss_data, i)
DTI_ms <- DTI_ms[-miss_data, ]
DTI_ms_2 <- DTI_ms[DTI_ms$Nscans == 4, ]
xx <- vector("list", 4)
for (i in 1:4) {
  xx[[i]] <- DTI_ms_2$cca[DTI_ms_2$visit == i, ]
}
xx[[1]] <- xx[[1]][-14, ]
xx[[3]] <- xx[[3]][-14, ]
yy <- xx
for (i in seq_len(4)) yy[[i]] <- yy[[i]][1:17, ]
# pointwise SSA test statistic
pointwise_ssa_test_statistic(yy, xlab = "t", xaxt = "n")
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
```

`rmfanova` *Repeated measures functional analysis of variance*

Description

The function `rmfanova()` calculates the tests based on three test statistics $\mathcal{C}_n$, $\mathcal{D}_n$, and $\mathcal{E}_n$ for the problem of comparing ℓ-samples of repeated measures for functional data. The tests are based on five resampling methods, i.e., two permutation and three bootstrap ones. The overall and local hypotheses are considered.

Usage

```
rmfanova(
  x,
  method = "bonferroni",
  n_perm = 1000,
  n_boot = 1000,
  parallel = FALSE,
  n_cores = NULL,
  multi_gen = FALSE
)
```

Arguments

x	a list of length ℓ with elements being $n \times p$ matrices of data corresponding to n functional observations measured in p design time points under given experimental conditions.
method	the correction method to be used for pairwise comparisons. Options are "bonferroni" (default) and those given in the vector p.adjust.methods (as for the p.adjust() function).
n_perm	a number of permutation replicates. The default is 1000.
n_boot	a number of bootstrap replicates. The default is 1000.
parallel	a logical indicating of whether to use parallel computing. The default is FALSE.
n_cores	if parallel = TRUE, a number of processes used in parallel computation. Its default value (NULL) means that it will be equal to a number of cores of a computer used.
multi_gen	a logical indicating of whether to use separate multiple generations of Gaussian processes for the parametric bootstrap tests. The default is FALSE, which means that the processes will be generated once in a big matrix. This method is much faster, but for larger n and p the generated data can be too large for RAM. In such a case, we suggest using separate generation (multi_gen = TRUE), which is slower, but possible to calculate.

Details

The function rmfanova() concerns the tests for the functional repeated measures analysis problem. The details are presented in Kurylo and Smaga (2023), where in particular, some recommendations for using tests are given. Here we present only some summary of the problem and its solutions implemented in the package.

We have n subjects subjected to $\ell \geq 2$ (possibly) different conditions. The results of the experiments are functional observations. Let the subjects be represented by a functional sample consisting of independent stochastic processes $Y_1, \ldots, Y_n$ defined on the interval $[0, \ell]$, which satisfy the following model proposed by Martinez-Camblor and Corral (2011):

$$Y_j(t) = \mu(t) + e_j(t),\ j = 1, \ldots, n,\ t \in [0, \ell],$$

where μ is a fixed mean function, and e_j is a random process with zero mean function. In this notation, $t \in [0, 1]$ corresponds to the first experimental condition, $t \in [1, 2]$ to the second, and

so on. Thus, in this model, we ignore the possible time periods between repetitions of the experiment, but this does not mean that they do not exist. We are interested in testing the equality of ℓ mean functions corresponding to experimental conditions; namely, the global null hypothesis is as follows:

$$\mathcal{H}_0 : \mu(t) = \mu(t+1) = \cdots = \mu(t+(\ell-1)) \ \ \forall t \in [0,1].$$

For the global null hypothesis $\mathcal{H}_0$, the tests given by Martinez-Camblor and Corral (2011) used the pointwise sum of squares due to the hypothesis:

$$\mathrm{SSA}_{point}(t) = n\sum_{i=1}^{\ell}(\bar{Y}_{i\cdot}(t) - \bar{Y}(t))^2, \ t \in [0,1],$$

where

$$\bar{Y}_{i\cdot}(t) = n^{-1}\sum_{j=1}^{n} Y_j(t+(i-1)), \ \bar{Y}(t) = N^{-1}\sum_{i=1}^{\ell}\sum_{j=1}^{n} Y_j(t+(i-1)),$$

$i = 1, \ldots, \ell$. In the package, it is calculated and drawn by the `pointwise_ssa_test_statistic()` function. The other option is the following pointwise F-type test statistic proposed in Kurylo and Smaga (2023):

$$F_{point}(t) = \frac{\mathrm{SSA}_{point}(t)/(\ell-1)}{\mathrm{SSR}_{point}(t)/((\ell-1)(n-1))}, \ t \in [0,1],$$

where

$$\mathrm{SSR}_{point}(t) = \sum_{i=1}^{\ell}\sum_{j=1}^{n}(Y_j(t+(i-1)) - \bar{Y}_{i\cdot}(t) - \bar{Y}_{\cdot j}(t) + \bar{Y}(t))^2$$

is the pointwise sum of squares due to residuals, and

$$\bar{Y}_{\cdot j}(t) = \ell^{-1}\sum_{i=1}^{\ell} Y_j(t+(i-1)), \ j = 1, \ldots, n.$$

F_{point} is calculated and drawn by the `pointwise_f_test_statistic()` function.

To obtain global test statistics for $\mathcal{H}_0$, Martinez-Camblor and Corral (2011) proposed the following test statistic:

$$\mathcal{C}_n(\ell) = \int_0^1 \mathrm{SSA}_{point}(t)dt.$$

On the other hand, Kurylo and Smaga (2023) proposed the following two test statistics:

$$\mathcal{D}_n(\ell) = \int_0^1 F_{point}(t)dt, \quad \mathcal{E}_n(\ell) = \sup_{t\in[0,1]} F_{point}(t).$$

To construct the tests, five resampling strategies are proposed by Kurylo and Smaga (2023). For details, we refer to this paper. Here we just note the two permutation tests and three bootstrap tests are denoted by P1, P2, B1, B2, and B3 in the output of the `summary.rmfanova()` function.

When $\ell > 2$, by rejecting the global null hypothesis $\mathcal{H}_0$, we determine the presence of significant differences in the mean functions corresponding to the experimental conditions. However, we do not know which conditions are significantly different and which are not. To solve this problem, one needs to perform a post hoc analysis. More precisely, we would like to test the family of hypotheses:

$$\begin{cases} \mathcal{H}_0^{rs} : \mu(t+(r-1)) = \mu(t+(s-1)) \ \forall t \in [0,1], \\ \mathcal{H}_1^{rs} : \mu(t+(r-1)) \neq \mu(t+(s-1)) \text{ for some } t \in [0,1], \end{cases}$$

for $r, s = 1, \ldots, \ell$, $r \neq s$. These hypotheses are also named pairwise comparisons. To test this family of local hypotheses, we propose the following procedure:

1. Test each of the hypotheses $\mathcal{H}_0^{rs}$ using the data for the r-th and s-th objects, i.e., $Y_1(t), \ldots, Y_n(t)$ for $t \in [r-1, r]$ and $t \in [s-1, s]$ respectively, and the chosen test from those presented above. Let p_{rs} denote the p-values obtained.

2. Make a final decision using the Bonferroni method, i.e., reject $\mathcal{H}_0^{rs}$ if $p_{rs}^{Bonf} \leq \alpha$, where $p_{rs}^{Bonf} = m \cdot p_{rs}$ are the corrected p-values, α is the significance level and m is the number of null hypotheses considered.

In the paper Kurylo and Smaga (2023), the Bonferroni method was used only. However, in the package, there is a possibility to use other correction methods, which are available in the vector `p.adjust.methods`.

The results of testing the global and local hypotheses are given separately in the output of the `summary.rmfanova()` function for the convenience of the user.

Value

A list of class `rmfanova` containing the following 7 components:

`n`	a number n of functional observations.
`p`	a number p of design time points.
`l`	a number ℓ of repeated samples.
`method`	an argument `method`.
`test_stat`	values of the test statistics $\mathcal{C}_n$, $\mathcal{D}_n$, and $\mathcal{E}_n$.
`p_values`	p-values for the global null hypothesis.
`p_values_pc`	p-values of the pairwise comparisons.

References

Martinez-Camblor P., Corral N. (2011) Repeated Measures Analysis for Functional Data. Computational Statistics & Data Analysis 55, 3244–3256.

Kurylo K., Smaga L. (2023) Functional repeated measures analysis of variance and its application. Preprint https://arxiv.org/abs/2306.03883

Ramsay J.O., Silverman B.W. (2005) Functional Data Analysis, 2nd Edition. New York: Springer.

Zhang J.T. (2013) Analysis of Variance for Functional Data. London: Chapman & Hall.

Examples

```
# Some of the examples may run some time.
# preparation of the DTI data set, for details see Kurylo and Smaga (2023)
library(refund)
data(DTI)
# MS patients
DTI_ms <- DTI[DTI$case == 1, ]
miss_data <- c()
for (i in 1:340) if (any(is.na(DTI_ms$cca[i, ]))) miss_data <- c(miss_data, i)
DTI_ms <- DTI_ms[-miss_data, ]
```

```
DTI_ms_2 <- DTI_ms[DTI_ms$Nscans == 4, ]
xx <- vector("list", 4)
for (i in 1:4) {
  xx[[i]] <- DTI_ms_2$cca[DTI_ms_2$visit == i, ]
}
xx[[1]] <- xx[[1]][-14, ]
xx[[3]] <- xx[[3]][-14, ]
yy <- xx
for (i in seq_len(4)) yy[[i]] <- yy[[i]][1:17, ]
# data trajectories for four visits
oldpar <- par(mfrow = c(1, 4), mar = c(4, 4, 4, 0.1))
matplot(t(yy[[1]]), type = "l", col = 1, lty = 1, xlab = "t", ylab = "FA",
        main = "Visit 1", xaxt = "n", ylim = c(0.29, 0.73))
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
matplot(t(yy[[2]]), type = "l", col = 1, lty = 1, xlab = "t", ylab = "FA",
        main = "Visit 2", xaxt = "n", ylim = c(0.29, 0.73))
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
matplot(t(yy[[3]]), type = "l", col = 1, lty = 1, xlab = "t", ylab = "FA",
        main = "Visit 3", xaxt = "n", ylim = c(0.29, 0.73))
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
matplot(t(yy[[4]]), type = "l", col = 1, lty = 1, xlab = "t", ylab = "FA",
        main = "Visit 4", xaxt = "n", ylim = c(0.29, 0.73))
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
par(oldpar)
# sample mean functions
oldpar <- par(mfrow = c(1, 1), mar = c(4, 4, 2, 0.1))
pointwise_sample_mean_fun(yy, values = FALSE,
                          col = 1:4, xlab = "t", ylab = "FA", xaxt = "n")
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
legend(x = 36, y = 0.64, legend = 1:4, lty = 1, col = 1:4, title = "Visit")
par(oldpar)
# pointwise SSA and F-type test statistics
oldpar <- par(mfrow = c(1, 2), mar = c(4, 2, 2, 0.1))
pointwise_ssa_test_statistic(yy, xlab = "t", xaxt = "n")
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
pointwise_f_test_statistic(yy, xlab = "t", xaxt = "n")
axis(1, c(1, 15, 30, 45, 60, 75, 93), labels = c(1, 15, 30, 45, 60, 75, 93))
par(oldpar)

# testing without parallel computing and multiple generation of Gaussian processes
res <- rmfanova(yy)
summary(res, digits = 3)
# testing without parallel computing and with multiple generation of Gaussian processes
res <- rmfanova(yy, multi_gen = TRUE)
summary(res, digits = 3)
# testing with parallel computing and without multiple generation of Gaussian processes
res <- rmfanova(yy, parallel = TRUE, n_cores = 2)
summary(res, digits = 3)
# testing with parallel computing and with multiple generation of Gaussian processes
res <- rmfanova(yy, parallel = TRUE, multi_gen = TRUE, n_cores = 2)
summary(res, digits = 3)
```

summary.rmfanova *Print "rmfanova" object*

Description

Prints the summary of the repeated measures functional analysis of variance.

Usage

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

Arguments

object	a "rmfanova" object.
...	integer indicating the number of decimal places to be used to present the numerical results. It can be named digits as in the round() function (see examples).

Details

The function prints out the information about the number of samples ℓ, number of observations n, number of design time points p, adjustment method for pairwise comparison tests (if $\ell > 2$), test statistics, and p-values of tests performed by the rmfanova() function.

Value

No return value, called for side effects.

Examples

```
# Some of the examples may run some time.
# preparation of the DTI data set, for details see Kurylo and Smaga (2023)
library(refund)
data(DTI)
# MS patients
DTI_ms <- DTI[DTI$case == 1, ]
miss_data <- c()
for (i in 1:340) if (any(is.na(DTI_ms$cca[i, ]))) miss_data <- c(miss_data, i)
DTI_ms <- DTI_ms[-miss_data, ]
DTI_ms_2 <- DTI_ms[DTI_ms$Nscans == 4, ]
xx <- vector("list", 4)
for (i in 1:4) {
  xx[[i]] <- DTI_ms_2$cca[DTI_ms_2$visit == i, ]
}
xx[[1]] <- xx[[1]][-14, ]
xx[[3]] <- xx[[3]][-14, ]
yy <- xx
for (i in seq_len(4)) yy[[i]] <- yy[[i]][1:17, ]
```

```
# testing without parallel computing and multiple generation of Gaussian processes
res <- rmfanova(yy)
summary(res, digits = 3)
```

Index