Package ‘denstest’

May 16, 2025

Title Density Equality Testing

Version 1.0.0

Description Methods for testing the equality between groups of estimated density functions. The package implements FDET (Fourier-based Density Equality Testing) and MDET (Moment-based Density Equality Testing), two new approaches introduced by the author. Both methods extend an earlier testing approach by Delicado (2007), ``Functional k-sample problem when data are density functions'' <doi:10.1007/s00180-007-0047-y>, which is referred to as DET (Density Equality Testing) in this package for clarity. FDET compares groups of densities based on their global shape using Fourier transforms, while MDET tests for differences in distributional moments. All methods are described in Anarat, Krutmann and Schwender (2025), ``Testing for Differences in Extrinsic Skin Aging Based on Density Functions'' (Submitted).

License GPL-3

VignetteBuilder knitr

Suggests knitr, rmarkdown

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Akin Anarat [aut, cre]

Maintainer Akin Anarat <akin.anarat@hhu.de>

Repository CRAN

Date/Publication 2025-05-16 09:50:10 UTC

Contents

compute_B *Computation of Test Statistics for Density Equality Tests*

Description

Computes the test statistics used in the denstest package for assessing equality between groups of estimated density functions. It supports all implemented methods (FDET, DET, and MDET), but does not perform permutation-based inference. The function is intended for users who want to inspect, compare, or further process the raw values of the test statistics without calculating p-values.

Usage

```
compute_B(
  L,
  group_sizes,
  N.max = 10000,
  a = NULL,
  b = NULL,
  m = 100,
  seed = NULL,
  density.weights = NULL,
  test = c("FDET.regular", "FDET.residual", "FDET.regular.real.imag",
    "FDET.residual.real.imag", "DET.regular", "DET.residual", "MDET.regular",
    "MDET.residual"),
  distance = c("LP", "Hellinger", "TF"),
  moment = c("expectation", "variance", "skewness", "kurtosis", "combined"),
  interpolation = c("linear", "spline"),
  p = 2,
  eps = 0.01,
  tau = 0.01,
  Lmax = 5000,
  ft.lp.weight = c("none", "AbsRoot"),
  real.imag.weights = c(0.5, 0.5),
  moment.weights = rep(0.25, 4),
  plot = FALSE,
 legend = c("topright", "topleft", "bottomright", "bottomleft", "top", "bottom", "left",
    "right", "center")
)
```

Arguments

L	A list of estimated density objects, where each element is a list with numeric vectors x and y. x contains the evaluation points, and y the corresponding estimated density values for a single observation.
group_sizes	A vector indicating the number of densities in each group.
N.max	Maximum number of permutations for the test (default: 10000).

`a, b` Evaluation range endpoints; if NULL, determined from data.

`m` Number of evaluation points (default: 100).

`seed` Random seed for reproducibility.

`density.weights`
Optional weights for densities.

`test` The test to use. One of "FDET.regular", "FDET.residual", "FDET.regular.real.imag", "FDET.residual.real.imag", "DET.regular", "DET.residual", "MDET.regular", or "MDET.residual".

`distance` The distance measure to use for FDET or DET. One of "LP", "Hellinger", or "TF".

`moment` Moment type used for MDET. One of "expectation", "variance", "skewness", "kurtosis", or "combined".

`interpolation` Method for interpolating densities. One of "linear" or "spline".

`p` Parameter for L^p distances (default: 2).

`eps` Cut-off parameter for the Fourier transforms.

`tau` Step size parameter of the Fourier Transforms.

`Lmax` Maximum size of the vectors containing the values of the individual Fourier transforms.

`ft.lp.weight` Weights for the Fourier transforms. One of "none" or "AbsRoot".

`real.imag.weights`
Weights for the real and imaginary parts of the Fourier transforms.

`moment.weights` Internal parameters for specific methods.

`plot` Logical. If TRUE, plots the density functions in `L`.

`legend` Position of the legend in the plot.

Value

Test statistic value indicating the degree of separation between groups of estimated density functions.

Author(s)

Akin Anarat <`akin.anarat@hhu.de`>

References

Anarat A., Krutmann, J., and Schwender, H. (2025). Testing for Differences in Extrinsic Skin Aging Based on Density Functions. Submitted.

Delicado, P. (2007). Functional k-sample problem when data are density functions. Computational Statistics, 22, 391–410. doi:10.1007/s001800070047y

Examples

```
n1 <- 5; n2 <- 5; n3 <- 5
group_sizes <- c(n1, n2, n3)
sample_size <- 500

densities_group1 <- lapply(1:n1, function(i) {
  data <- rnorm(sample_size, 0, 0.3)
  d <- density(data)
  list(x = d$x, y = d$y)
})

densities_group2 <- lapply(1:n2, function(i) {
  data <- rnorm(sample_size, 0, 0.32)
  d <- density(data)
  list(x = d$x, y = d$y)
})

densities_group3 <- lapply(1:n3, function(i) {
  data <- rnorm(sample_size, 0.02, 0.28)
  d <- density(data)
  list(x = d$x, y = d$y)
})

L <- c(densities_group1, densities_group2, densities_group3)

compute_B(L, group_sizes, ft.lp.weight = "AbsRoot")
```

---

denscomp *Density Equality Testing*

Description

This is the main function of the denstest package. It performs statistical tests for the equality between groups of estimated density functions using FDET, DET, or MDET.

Usage

```
denscomp(
  L,
  group_sizes,
  N.max = 10000,
  a = NULL,
  b = NULL,
  m = 100,
  seed = NULL,
  density.weights = NULL,
  test = c("FDET.regular", "FDET.residual", "FDET.regular.real.imag",
    "FDET.residual.real.imag", "DET.regular", "DET.residual", "MDET.regular",
    "MDET.residual"),
  distance = c("LP", "Hellinger", "TF"),
  moment = c("expectation", "variance", "skewness", "kurtosis", "combined"),
  interpolation = c("linear", "spline"),
  p = 2,
  eps = 0.01,
  tau = 0.01,
  Lmax = 5000,
  ft.lp.weight = c("none", "AbsRoot"),
  real.imag.weights = c(0.5, 0.5),
  moment.weights = rep(0.25, 4),
  plot = FALSE,
  legend = c("topright", "topleft", "bottomright", "bottomleft", "top", "bottom", "left",
    "right", "center")
)
```

Arguments

`L` A list of estimated density objects, where each element is a list with numeric vectors `x` and `y`. `x` contains the evaluation points, and `y` the corresponding estimated density values for a single observation.

`group_sizes` A vector indicating the number of densities in each group.

`N.max` Maximum number of permutations for the test (default: 10000).

`a, b` Evaluation range endpoints; if NULL, determined from data.

`m` Number of evaluation points (default: 100).

`seed` Random seed for reproducibility.

`density.weights`
Optional weights for densities.

`test` The test to use. One of "FDET.regular", "FDET.residual", "FDET.regular.real.imag", "FDET.residual.real.imag", "DET.regular", "DET.residual", "MDET.regular", or "MDET.residual".

`distance` The distance measure to use for FDET or DET. One of "LP", "Hellinger", or "TF".

`moment` Moment type used for MDET. One of "expectation", "variance", "skewness", "kurtosis", or "combined".

`interpolation` Method for interpolating densities. One of "linear" or "spline".

`p` Parameter for L^p distances (default: 2).

`eps` Cut-off parameter for the Fourier transforms.

`tau` Step size parameter of the Fourier Transforms.

`Lmax` Maximum size of the vectors containing the values of the individual Fourier transforms.

`ft.lp.weight` Weights for the Fourier transforms. One of "none" or "AbsRoot".

`real.imag.weights`
Weights for the real and imaginary parts of the Fourier transforms.

`moment.weights` Internal parameters for specific methods.

`plot` Logical. If TRUE, plots the density functions in `L`.

`legend` Position of the legend in the plot.

Value

A p-value indicating the significance of group differences.

Author(s)

Akin Anarat <`akin.anarat@hhu.de`>

References

Anarat A., Krutmann, J., and Schwender, H. (2025). Testing for Differences in Extrinsic Skin Aging Based on Density Functions. Submitted.

Delicado, P. (2007). Functional k-sample problem when data are density functions. Computational Statistics, 22, 391–410. doi:10.1007/s001800070047y

Examples

```
n1 <- 5; n2 <- 5; n3 <- 5
group_sizes <- c(n1, n2, n3)
sample_size <- 500

densities_group1 <- lapply(1:n1, function(i) {
  data <- rnorm(sample_size, 0, 0.3)
  d <- density(data)
  list(x = d$x, y = d$y)
})

densities_group2 <- lapply(1:n2, function(i) {
  data <- rnorm(sample_size, 0, 0.32)
  d <- density(data)
  list(x = d$x, y = d$y)
})

densities_group3 <- lapply(1:n3, function(i) {
  data <- rnorm(sample_size, 0.02, 0.28)
  d <- density(data)
  list(x = d$x, y = d$y)
})

L <- c(densities_group1, densities_group2, densities_group3)

denscomp(L, group_sizes, ft.lp.weight = ”AbsRoot”)
```

Index