Package 'Dogoftest'

July 21, 2025

Title Distributed Online Goodness-of-Fit Tests for Distributed Datasets

Date 2025-06-16

Version 0.2

Description Distributed Online Goodness-of-Fit Test can process the distributed datasets. The philosophy of the package is described in Guo G.(2024) <doi:10.1016/j.apm.2024.115709>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Guangbao Guo [aut, cre] (ORCID: <https://orcid.org/0000-0002-4115-6218>), Di Chang [aut]

Maintainer Guangbao Guo <ggb11111111@163.com>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-06-24 09:50:08 UTC

Contents

`AD2gof` *Two-Sample Anderson-Darling Test (Bootstrap Version)*

Description

Performs a two-sample Anderson-Darling (AD) goodness-of-fit test using bootstrap resampling to compare whether two samples come from the same distribution. This test is sensitive to differences in both location and shape between the two distributions.

Usage

```
AD2gof(
  x,
  y,
  alternative = c("two.sided", "less", "greater"),
  nboots = 2000,
  keep.boots = FALSE
)
```

Arguments

`x`	A numeric vector of data values from the first sample.
`y`	A numeric vector of data values from the second sample.
`alternative`	Character string specifying the alternative hypothesis. One of ‘"two.sided"‘ (default), ‘"less"‘, or ‘"greater"‘.
`nboots`	Integer. Number of bootstrap replicates to compute the null distribution (default: 2000).
`keep.boots`	Logical. If ‘TRUE‘, returns the full vector of bootstrap statistics (default: ‘FALSE‘).

Details

The test computes the Anderson-Darling statistic using the pooled empirical distribution functions (ECDFs) of the two samples. A bootstrap procedure resamples the group labels to approximate the null distribution and compute a p-value. If ‘p.value = 0‘, it is adjusted to ‘1 / (2 * nboots)‘ for stability.

Value

A list of class ‘"htest"‘ containing:

statistic The observed Anderson-Darling test statistic.

p.value The estimated bootstrap p-value.

alternative The alternative hypothesis used.

method A character string describing the test.

bootstraps (Optional) A numeric vector of bootstrap statistics if ‘keep.boots = TRUE‘.

Examples

```
set.seed(123)
x <- rnorm(100, mean = 0, sd = 4)
y <- rnorm(100, mean = 2, sd = 4)
AD2gof(x, y)
```

`ADgof` *Anderson-Darling Goodness-of-Fit Test for a Specified Distribution*

Description

Performs the Anderson-Darling (AD) goodness-of-fit test for a given univariate distribution. The function computes the AD statistic and returns an approximate p-value based on adjusted formulas.

Usage

```
ADgof(
  x,
  dist = c("norm", "exp", "unif", "lnorm", "weibull", "gamma", "t", "chisq"),
  ...,
  eps = 1e-15
)
```

Arguments

`x`	A numeric vector of sample observations.
`dist`	A character string specifying the null distribution. Options are `"norm"`, `"exp"`, `"unif"`, `"lnorm"`, `"weibull"`, `"gamma"`, `"t"`, and `"chisq"`.
`...`	Additional named parameters passed to the corresponding distribution functions (e.g., `mean`, `sd`, `rate`, `df`, etc.).
`eps`	A small positive constant to avoid log(0) during computation (default: `1e-15`).

Details

This implementation supports several common distributions. Parameters of the null distribution must be supplied via .... The p-value is calculated using the approximations suggested by Stephens (1986) and other refinements. For small samples or custom distributions, a bootstrap version may be preferred.

Value

A list of class "htest" with components:

statistic The value of the Anderson-Darling test statistic.

p.value The approximate p-value computed using adjustment formulas.

method A description of the test performed.

data.name The name of the input data.

Examples

```
set.seed(123)
x1 <- rnorm(500, mean = 5, sd = 2)
ADgof(x1, dist = "norm", mean = 5, sd = 2)

x2 <- rexp(400, rate = 1.5)
ADgof(x2, dist = "exp")
ADgof(x2, dist = "exp", rate = 1.5)

x3 <- runif(300, min = -2, max = 4)
ADgof(x3, dist = "unif", min = -2, max = 4)
```

CVM2gof *Two-Sample Cramér–von Mises Test (Bootstrap Version)*

Description

Performs a nonparametric two-sample Cramér–von Mises test using a permutation-based bootstrap method to assess whether two samples come from the same distribution.

Usage

```
CVM2gof(
  x,
  y,
  alternative = c("two.sided", "less", "greater"),
  nboots = 2000,
  keep.boots = FALSE
)
```

Arguments

x	Numeric vector of observations from the first sample.
y	Numeric vector of observations from the second sample.
alternative	Character string specifying the alternative hypothesis. Must be one of "two.sided" (default), "less", or "greater".
nboots	Number of bootstrap replicates to approximate the null distribution (default: 2000).
keep.boots	Logical. If TRUE, the bootstrap statistics will be returned (default: FALSE).

Details

The test compares two empirical cumulative distribution functions (ECDFs). The bootstrap procedure permutes group labels to generate the null distribution. Tailored one-sided tests use one-sided squared differences of ECDFs.

Value

An object of class "htest" with elements:

statistic Observed Cramér–von Mises test statistic.

p.value Bootstrap-based p-value.

alternative The alternative hypothesis used.

method A description of the test.

bootstraps (Optional) Vector of bootstrap test statistics if keep.boots = TRUE.

Examples

```
set.seed(123)
x <- rnorm(100, mean = 0, sd = 4)
y <- rnorm(100, mean = 2, sd = 4)
CVM2gof(x, y)

# One-sided test
CVM2gof(x, y, alternative = "greater")

# Store bootstrap replicates
res <- CVM2gof(x, y, keep.boots = TRUE)
hist(res$bootstraps, main = "Bootstrap Distribution", xlab = "Test Statistic")
```

cvmgof *Perform the Cramer-von Mises Goodness-of-Fit Test for Normality*

Description

Perform the Cramer-von Mises Goodness-of-Fit Test for Normality

Usage

```
cvmgof(x)
```

Arguments

x	A numeric vector containing the sample data.

Value

statistic	The value of the Cramer-von Mises test statistic.
p.value	The p-value for the test.
method	A character string describing the test.

Examples

```
# Example usage:
set.seed(123)
x <- rnorm(100)  # Generate a sample from a normal distribution
result <- cvmgof(x)
print(result)

# Example with non-normal data:
y <- rexp(100)  # Generate a sample from an exponential distribution
result <- cvmgof(y)
print(result)
```

CVMgof2 *One-Sample Cramér–von Mises Goodness-of-Fit Test*

Description

Performs the one-sample Cramér–von Mises goodness-of-fit (GoF) test to assess whether a sample comes from a specified distribution using asymptotic p-value approximations.

Usage

```
CVMgof2(
  x,
  dist = c("norm", "exp", "unif", "lnorm", "weibull", "gamma", "t", "chisq"),
  ...,
  eps = 1e-15
)
```

Arguments

x	A numeric vector of observations.
dist	A character string specifying the theoretical distribution. Must be one of "norm", "exp", "unif", "lnorm", "weibull", "gamma", "t", or "chisq".
...	Distribution parameters passed to the corresponding p functions (e.g., mean, sd, rate, df, etc.).
eps	A small value to truncate extreme p-values (default is 1e-15).

Details

The test uses the Cramér–von Mises statistic to assess how well the empirical distribution function (EDF) of the sample agrees with the cumulative distribution function (CDF) of the specified theoretical distribution. The p-value is computed using approximation formulas derived from the asymptotic distribution of the test statistic.

Value

An object of class "htest" with the following components:

statistic The computed Cramér–von Mises test statistic.

p.value The asymptotic p-value.

method A description of the test and distribution.

data.name The name of the data vector.

Examples

```
set.seed(123)
x1 <- rnorm(500, mean = 0, sd = 1)
CVMgof2(x1, dist = "norm", mean = 0, sd = 1)

x2 <- rexp(500, rate = 2)
CVMgof2(x2, dist = "exp", rate = 2)

x3 <- runif(200, min = -1, max = 3)
CVMgof2(x3, dist = "unif", min = -1, max = 3)
```

KS2gof *Two-Sample Kolmogorov–Smirnov Test with Bootstrap*

Description

Performs a two-sample Kolmogorov–Smirnov (KS) test using a bootstrap method to assess whether two independent samples come from the same distribution.

Usage

```
KS2gof(
  x,
  y,
  alternative = c("two.sided", "less", "greater"),
  nboots = 5000,
  keep.boots = FALSE
)
```

Arguments

x, y	Numeric vectors of data values for the two independent samples.
alternative	Character string specifying the alternative hypothesis, must be one of "two.sided", "less", or "greater".
nboots	Number of bootstrap resamples used to approximate the null distribution (default: 5000).
keep.boots	Logical; if TRUE, return the vector of bootstrap statistics.

Details

This implementation performs a nonparametric KS test for equality of distributions by resampling under the null hypothesis. It supports one-sided and two-sided alternatives.

If keep.boots = TRUE, the function returns all bootstrap statistics, which can be used for further analysis (e.g., plotting).

If the p-value is zero due to no bootstrap statistic exceeding the observed value, it is adjusted to 1 / (2 * nboots) to avoid a zero p-value.

Value

An object of class "htest" with the following components:

statistic The observed KS statistic.

p.value The p-value based on the bootstrap distribution.

alternative The alternative hypothesis.

method Description of the test used.

Examples

```
set.seed(123)
x <- rnorm(100, mean = 0, sd = 4)
y <- rnorm(100, mean = 2, sd = 4)
KS2gof(x, y)
```

---

ksgof *Perform the Lilliefors (Kolmogorov-Smirnov) Goodness-of-Fit Test for Normality*

Description

Perform the Lilliefors (Kolmogorov-Smirnov) Goodness-of-Fit Test for Normality

Usage

```
ksgof(x)
```

Arguments

x	A numeric vector containing the sample data.

Value

statistic	The value of the Lilliefors (Kolmogorov-Smirnov) test statistic.
p.value	The p-value for the test.
method	A character string describing the test.

Examples

```
# Example usage:
set.seed(123)
x <- rnorm(100)  # Generate a sample from a normal distribution
result <- ksgof(x)
print(result)

# Example with non-normal data:
y <- rexp(100)  # Generate a sample from an exponential distribution
result <- ksgof(y)
print(result)
```

KSgof2 *One-sample Kolmogorov-Smirnov goodness-of-fit test*

Description

Performs the one-sample Kolmogorov-Smirnov test for a specified theoretical distribution.

Usage

```
KSgof2(
  x,
  dist = c("norm", "exp", "unif", "lnorm", "weibull", "gamma", "t", "chisq"),
  ...,
  eps = 1e-15
)
```

Arguments

x	Numeric vector of observations.
dist	Character string specifying the distribution to test against. One of "norm", "exp", "unif", "lnorm", "weibull", "gamma", "t", or "chisq".
...	Additional parameters passed to the distribution's cumulative distribution function (CDF). For example, mean and sd for the normal distribution.
eps	Numeric lower and upper bound for tail probabilities to avoid numerical issues (default: 1e-15).

Details

The test compares the empirical distribution function of x with the cumulative distribution function of a specified theoretical distribution using the Kolmogorov-Smirnov statistic. For large sample sizes, a p-value approximation based on the asymptotic distribution is used.

A correction is applied when sample size exceeds 100, adjusting the test statistic to approximate a fixed sample size. For very small or very large statistics, piecewise polynomial approximations are used to compute the p-value.

Value

An object of class "htest" containing the test statistic, p-value, method description, and data name.

Examples

```
set.seed(123)
x <- rnorm(1000, mean = 5, sd = 2)
KSgof2(x, dist = "norm", mean = 5, sd = 2)

y <- rexp(500, rate = 0.5)
KSgof2(y, dist = "exp", rate = 0.5)
```

```
u <- runif(300, min = 0, max = 10)
KSgof2(u, dist = "unif", min = 0, max = 10)
```

`Kuiper2gof` *Two-Sample Kuiper Test with Bootstrap*

Description

Performs a two-sample Kuiper test using bootstrap resampling to test whether two independent samples come from the same distribution.

Usage

```
Kuiper2gof(
  x,
  y,
  alternative = c("two.sided", "less", "greater"),
  nboots = 2000,
  keep.boots = FALSE
)
```

Arguments

`x, y`	Numeric vectors of data values for the two samples.
`alternative`	Character string indicating the alternative hypothesis. Must be one of `"two.sided"`, `"less"`, or `"greater"`.
`nboots`	Integer. Number of bootstrap resamples to compute the empirical null distribution (default: 2000).
`keep.boots`	Logical. If `TRUE`, returns all bootstrap test statistics.

Details

The Kuiper test is a nonparametric test similar to the Kolmogorov–Smirnov test, but sensitive to discrepancies in both location and shape between two distributions. This implementation uses bootstrap resampling to estimate the p-value.

The `two.sided` test uses the sum of maximum positive and negative ECDF differences. The `greater` and `less` options use one-sided variations.

If the observed test statistic exceeds all bootstrap values, the p-value is set to `1 / (2 * nboots)` to avoid zero.

Value

An object of class "htest" containing:

statistic The observed Kuiper statistic.

p.value The p-value computed from the bootstrap distribution.

alternative The specified alternative hypothesis.

method A character string describing the test.

bootstraps (If requested) A numeric vector of bootstrap statistics.

Examples

```
set.seed(123)
x <- rnorm(100, 0, 4)
y <- rnorm(100, 2, 4)
Kuiper2gof(x, y)
```

---

qCvMgof *Calculate the Quantile of the Cramer-von Mises Goodness-of-Fit Statistic*

Description

This function calculates the quantile of the Cramer-von Mises goodness-of-fit statistic using the ‘uniroot‘ function to find the root of the given function.

Usage

```
qCvMgof(X, p)
```

Arguments

X	A numeric vector containing the sample data.
p	A numeric value representing the desired quantile probability.

Value

root	The quantile value corresponding to the given probability.

Examples

```
# Example usage:
set.seed(123)
X <- rnorm(100)  # Generate a sample from a normal distribution
p <- 0.95        # Desired quantile probability
result <- qCvMgof(X, p)
print(result)
```

simpleCvMgof — *Perform a Simple Cramer-von Mises Goodness-of-Fit Test*

Description

This function performs a simple Cramer-von Mises goodness-of-fit test to assess whether a given sample comes from a uniform distribution. The test statistic and p-value are calculated based on the sorted sample data.

Usage

```
simpleCvMgof(X)
```

Arguments

X	A numeric vector containing the sample data.

Value

statistic	The value of the Cramer-von Mises test statistic.
pvalue	The p-value for the test.
statname	The name of the test statistic.

Examples

```
# Example usage:
set.seed(123)
X <- runif(100)  # Generate a sample from a uniform distribution
result <- simpleCvMgof(X)
print(result)

# Example with non-uniform data:
Y <- rnorm(100)  # Generate a sample from a normal distribution
result <- simpleCvMgof(Y)
print(result)
```

Wgof — *Watson goodness-of-fit test Performs the Watson test for goodness-of-fit to a specified distribution.*

Description

Watson goodness-of-fit test Performs the Watson test for goodness-of-fit to a specified distribution.

Usage

```
Wgof(x, dist = c("norm", "exp", "unif", "lnorm", "gamma"), ..., eps = 1e-15)
```

Arguments

x	Numeric vector of observations.
dist	Character string specifying the distribution to test against. One of "norm", "exp", "unif", "lnorm", or "gamma".
...	Additional parameters passed to the distribution's cumulative distribution function (CDF). For example, mean and sd for the normal distribution.
eps	Numeric tolerance for probability bounds to avoid extremes (default: 1e-15).

Details

The Watson test is a modification of the Cramér–von Mises test, adjusting for mean deviations. It measures the squared distance between the empirical distribution function of the data and the specified theoretical cumulative distribution function, with a correction for location.

Value

An object of class "htest" containing the test statistic, p-value, method description, data name, and any distribution parameters used.

Examples

```
set.seed(123)
x_norm <- rnorm(1000, mean = 5, sd = 2)
Wgof(x_norm, dist = "norm", mean = 5, sd = 2)

x_exp <- rexp(500, rate = 0.5)
Wgof(x_exp, dist = "exp", rate = 0.5)

x_unif <- runif(300, min = 0, max = 10)
Wgof(x_unif, dist = "unif", min = 0, max = 10)

x_lnorm <- rlnorm(200, meanlog = 0, sdlog = 1)
Wgof(x_lnorm, dist = "lnorm", meanlog = 0, sdlog = 1)

x_gamma <- rgamma(400, shape = 1, rate = 1)
Wgof(x_gamma, dist = "gamma", shape = 1, rate = 1)
```

Index