

Package ‘groupcompare’

May 8, 2026

Type Package

Title Comparing Two Groups Using Various Descriptive Statistics

Version 1.0.1

Date 2025-6-25

Author Zeynel Cebeci [aut, cre],
A. Firat Ozdemir [aut],
Engin Yildiztepe [aut]

Maintainer Zeynel Cebeci <zcebeci@cu.edu.tr>

Description Comparing two independent or paired groups across a range of descriptive statistics, enabling the evaluation of potential differences in central tendency (mean, median), dispersion (variance, interquartile range), shape (skewness, kurtosis), and distributional characteristics (various quantiles). The analytical framework incorporates parametric t-tests, non-parametric Wilcoxon tests, permutation tests, and bootstrap resampling techniques to assess the statistical significance of observed differences.

Depends R (>= 4.5.0)

License GPL (>= 2)

LazyData true

Imports boot, vioplot

Suggests knitr, rmarkdown, prettydoc

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2025-06-25 19:50:05 UTC

Contents

groupcompare-package	2
bivarplot	4
bootstrap	5
calhubermeandif	6
calcquantdif	8

calcquantile	9
calcstatdif	11
ci2df	12
descstats	13
ghdist	15
groupcompare	16
groupdata	18
hdqe	19
intnorm	20
levene.test	21
long2wide	22
permtest	23
quail	25
wide2long	25
Index	27

groupcompare-package *Comparing Two Groups with Various Descriptive Statistics*

Description

The ‘groupcompare’ package performs various statistical tests to compare two groups. It calculates descriptive statistics and quantile statistics and then conducts some normality tests and variance homogeneity tests. Based on these assumptions checks, it provides results from t-test or Wilcoxon rank sum test, permutation tests, and bootstrap confidence intervals.

Details

The main function ‘groupcompare’ of the package is designed to compare two independent or paired groups using various statistical tests. It calculates descriptive statistics and quantile statistics. Then it performs Shapiro-Wilk normality tests, variance homogeneity test (Levene’s test), t-test, Wilcoxon signed-rank sum test (or Mann-Whitney U test), permutation tests, and bootstrap confidence intervals.

groupcompare The main function which compares descriptive statistics of two groups using a variety of statistical tests.

bivarplot Generates various plots to visualize and compare the distribution and characteristics of two variables.

bootstrap Calculates bootstrap confidence intervals for the descriptive statistics or any statistic implemented in a custom function.

calchubermeandif Computes the difference between Huber’s M-estimator of location of two groups in long data format.

calcquantdif Calculates the differences between specified quantiles for grouped data.

calcquantile Calculates the quantiles (percentiles) for a given vector of data at specified fractions.

[calcstatdif](#) Calculates the differences in multiple statistics (mean, median, IQR, variance) for grouped data.

[ci2df](#) Converts a list of confidence intervals into a data frame.

[descstats](#) Calculates the common and robust descriptive statistics.

[ghdist](#) Generates a random sample from the g-and-h (gh) distribution with specified parameters.

[groupdata](#) A data set contains seven data frames with two variables from various distributions.

[hdqe](#) Computes the Harrell-Davis quantile estimator for given quantile levels.

[intnorm](#) Performs an inverse normal transformation on non-normally distributed data.

[levene.test](#) Performs the Levene test to check the homogeneity of variances across groups.

[long2wide](#) Converts long-format data to wide-format data by splitting based on groups.

[permtest](#) Performs a permutation test on long-format data to evaluate differences between two groups using a specified test statistic.

[quail](#) A data frame containing daily weight gains (in grams) of two quail breeds during a fattening period.

[wide2long](#) Converts wide-format data to long-format data.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

See Also

[bootstrap](#), [permtest](#), [ghdist](#), [bivarplot](#)

Examples

```
# Sample dataset in long format
set.seed(123)
group1 <- rnorm(30, mean=50, sd=2)
group2 <- rnorm(30, mean=51, sd=3)
df <- data.frame(value=c(group1, group2), group=rep(c("A", "B"), each=30))

# Compare the groups using various descriptive statistics
result <- groupcompare(df, cl=0.95, alternative="two.sided",
  q=c(0.25, 0.5, 0.75), qt=0, R=500, out=FALSE, verbose=TRUE)
result

# Compare the groups using Huber's M-estimator of location with bootstrap
bshubermean <- bootstrap(df, statistic=calchubermeandif,
  alternative="two.sided", alpha=0.05, R=500)
bshubermean

permhubermean <- permtest(df, statistic=calchubermeandif,
  alternative="two.sided", R=500)
permhubermean$pval
```

bivarplot*Plots for Two Variables*

Description

Generates various plots to visualize and compare the distribution and characteristics of two variables.

Usage

```
bivarplot(ds)
```

Arguments

<code>ds</code>	A data frame or matrix containing the input data. The first column should be the variable of interest, and the second column should be the grouping variable, if data is in long format.
-----------------	--

Details

This function generates a series of plots to compare two variables, including density plots, ECDF plots, boxplots, violin plots, QQ plots, symmetry plots, and empirical shift plots. If data is in long format, the function uses the second column of `ds` for the group label.

Value

Generates a series of plots to the current graphical device.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

Examples

```
# Create data for two independent groups, each with 50 observations
set.seed(1)
df1 <- data.frame(value = rnorm(100), group = rep(1:2, each = 50))
head(df1)

# Plots for visualization
bivarplot(df1)

# Convert to long data to wide data
df2 <- long2wide(df1)
head(df2)

# Plots for visualization
bivarplot(df2)
```

bootstrap

*Bootstrap for Various Statistics***Description**

Conducts a bootstrap procedure to calculate confidence intervals for various statistics.

Usage

```
bootstrap(x, statistic, alternative="two.sided", alpha = 0.05,
         Q = c(0.25, 0.5, 0.75), qt = 7, R = 3000)
```

Arguments

x	A data frame or matrix containing the data.
statistic	Name of the function to compute the statistic of interest.
alternative	Type of hypothesis test. The default is "two.sided".
alpha	A numeric value specifying the significance level for confidence intervals. The default is 0.05.
Q	A numeric vector or a number specifying the probabilities used in quantile calculations. The default is c(0.25, 0.5, 0.75) for P25, P50, P75 (aka Q1, Q2 and Q3).
qt	A numeric value specifying the type of quantile calculation. The default is 7.
R	An integer specifying the number of bootstrap replicates. The default is 3000.

Details

This function performs a bootstrap procedure to calculate confidence intervals for various statistics. It is mainly used to evaluate the differences between various statistics for two groups based on a specified function. The function calculates confidence intervals using different methods, including normal, basic, percentile, and bias-corrected and accelerated (BCa) intervals. It allows users to pass custom statistics (via `statistic`) that include parameters like quantiles (via `Q`) and types of quantiles (`qt`), making it versatile for non-standard use cases. In this way, the function extends the capabilities of `boot::boot.ci` (R's suggested package) by supporting more advanced statistical needs and customizable interval calculations. This function also incorporates the argument `alternative` can be set to "greater" or "less" for one-tailed confidence intervals, whereas `boot::boot.ci` primarily focuses on two-tailed intervals. Therefore, it is particularly useful for specialized applications. The data can be provided in long format, and the test uses a specified number of bootstrap replicates to calculate the empirical distribution of the test statistic under the null hypothesis.

Value

A list containing the data frames for the following components for each statistic:

normal	Lower and upper limits of the confidence interval computed with the normal method.
--------	--

basic	Lower and upper limits of the confidence interval computed with the basic method.
percent	Lower and upper limits of the confidence interval computed with the percent method.
bca	Lower and upper limits of the confidence interval computed with the BCa method.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

See Also

[permtest](#)

Examples

```
# Load the quail dataset
data(quail)

# Bootstrap for the difference of various basic statistics
# Increase R for real-world applications

bootres <- bootstrap(quail, statistic = calcstatdif, R=200)
bootres

# Arrange the results as a data frame
ci2df(bootres)

# Bootstrap for the differences of quantiles
bootresq <- bootstrap(quail, statistic = calcquantdif, R=200)
bootresq

# Arrange the results as a data frame
ci2df(bootresq)
```

calchubermeandif

The difference between Huber's M-estimator of location

Description

Computes the difference between Huber's M-estimator of location of two groups in long data format.

Usage

```
calchubermeandif(x, indices, ...)
```

Arguments

x	A data frame or matrix containing the input data. The first column should be the variable of interest, and the second column should be the grouping variable.
indices	Optional; specific rows to be considered. If not provided, all rows are used.
...	Additional arguments passed to the internal hubermean function.

Details

This function demonstrates the structure of a user-defined statistic function to use with bootstrap and permutation test. The function calculates the difference between Huber's M-estimator of location of two groups using the iterative weighted mean method. Huber's M-estimator of location is robust to outliers and is computed using an iterative re-weighting procedure. The internal function follows:

- Initialize with the median of the data.
- Compute weights based on deviations from the current mean.
- Update the mean iteratively until convergence is reached.

Value

A numeric value representing the difference between Huber's M-estimator of location of the two groups.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

See Also

[median](#)

Examples

```
df <- data.frame(value = rnorm(100), group = rep(1:2, each = 50))
bivarplot(df)

# Bootstrap confidence intervals for the difference of
# Huber's M-estimator of location of two groups
# Increase R for real-world applications
bshubermean <- bootstrap(df, statistic=calchubermeandif, alpha=0.05,
  alternative="less", R=200)
bshubermean
```

`calcquantdif`*Quantile Differences*

Description

Calculates the differences between specified quantiles for grouped data.

Usage

```
calcquantdif(x, indices, Q=seq(0.1, 0.9, 0.1), qt=7)
```

Arguments

<code>x</code>	A data frame or matrix containing the input data. The first column should be the variable of interest, and the second column should be the grouping variable.
<code>indices</code>	Optional; specific rows to be considered. If not provided, all rows are used.
<code>Q</code>	A numeric vector specifying the quantiles to be computed. The default is <code>seq(0.1, 0.9, 0.1)</code> .
<code>qt</code>	An integer specifying the quantile type from 0 to 9. The default is type 7, as discussed by Hyndman and Fan (1996)<doi:10.2307/2684934>.

Details

This function calculates the differences between specified quantiles for groups defined by the second column of the input data. It uses the specified quantile type to compute the quantiles. Types of quantiles are:

- **0**: Harrell-Davis estimator (not available in `stats::quantile` function).
- **1**: Inverse of the empirical distribution function.
- **2**: Similar to Type 1 but with averaging at discontinuities.
- **3**: Empirical distribution with sampling.
- **4**: Linear interpolation of the empirical distribution function.
- **5**: Linear interpolation of the expectations for the order statistics.
- **6**: Linear interpolation of the modes for the order statistics.
- **7**: The default in the `stats::quantile` function.
- **8**: Median-unbiased estimator.
- **9**: Normal-unbiased estimator.

For the details on types, see the [quantile](#) and [hdqe](#) function.

Value

A numeric vector containing the differences between the specified quantiles for each group.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

References

Hyndman, R. J. and Fan, Y. (1996) Sample quantiles in statistical packages, *American Statistician*, 50, 361-365. <doi:10.2307/2684934>.

See Also

[quantile](#), [hdqe](#), [calcstatdif](#)

Examples

```
df <- data.frame(value = rnorm(100), group = rep(1:2, each = 50))

# Difference between the medians
mediandif <- calcquantdif(df, Q=0.5)
mediandif

# Differences between the quartiles
quandifs <- calcquantdif(df, Q=c(0.25, 0.5, 0.75))
quandifs

# Differences between the percentiles from P10 to P90 using the method 5
quants <- seq(0.1, 0.9, 0.1)
quandifs <- calcquantdif(df, Q=quants, qt=5)
quandifs

# Differences between the percentiles from P10 to P90 using the method Harrell-Davis
quants <- seq(0.1, 0.9, 0.1)
quandifs <- calcquantdif(df, Q=quants, qt=0)
quandifs
```

calcquantile

Sample Quantiles

Description

Calculates the quantiles (percentiles) for a given vector of data at specified fractions.

Usage

```
calcquantile(x, indices, Q = seq(0.1, 0.9, 0.1), qt = 7)
```

Arguments

x	Numeric vector containing the values to calculate quantiles.
indices	Optional; vector containing the indices for which the calculation will be performed.
Q	Probabilities for quantile levels. The default is <code>seq(0.1, 0.9, 0.1)</code>
qt	Type of quantile calculation. Integer between 0 and 9. Default: 7

Details

This function calculates the quantiles at specified fractions of the given data set. If `qt` is 0, the [hdqe](#) function is used.

- **0**: Harrell-Davis estimator (not available in `stats::quantile` function).
- **1**: Inverse of the empirical distribution function.
- **2**: Similar to Type 1 but with averaging at discontinuities.
- **3**: Empirical distribution with sampling.
- **4**: Linear interpolation of the empirical distribution function.
- **5**: Linear interpolation of the expectations for the order statistics.
- **6**: Linear interpolation of the modes for the order statistics.
- **7**: The default in the `stats::quantile` function.
- **8**: Median-unbiased estimator.
- **9**: Normal-unbiased estimator.

For the details on types, see the [quantile](#) and [hdqe](#) functions.

Value

Returns a numeric vector containing the calculated quantiles.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

References

Hyndman, R. J. and Fan, Y. (1996) Sample quantiles in statistical packages, *American Statistician* **50**, 361–365. <doi:10.2307/2684934>.

See Also

[quantile](#), [hdqe](#)

Examples

```
x <- rnorm(100)
calcquantile(x)
calcquantile(x, qt=9)
calcquantile(x, qt = 0)
```

Description

Calculates the differences in multiple statistics (mean, median, IQR, variance) for grouped data.

Usage

```
calcstatdif(x, indices, ...)
```

Arguments

x	A data frame or matrix containing the input data. The first column should be the variable of interest, and the second column should be the grouping variable.
indices	Optional; specific rows to be considered. If not provided, all rows are used.
...	Optional arguments;

Details

This function calculates the differences in multiple statistics (mean, median, interquartile range (IQR), and variance) for groups defined by the second column of the input data. The function is used in permutation test and bootstrapping as a statistical function.

Value

A named numeric vector containing the differences in the specified statistics for each group. The names of the vector elements are "MEAN", "MED", "IQR", "VAR", "SKEW", and "KURT" for mean, median, interquartile range, variance, skewness, and kurtosis, respectively.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

See Also

[calcquantdif](#)

Examples

```
# Generate example data
set.seed(1199)
grp1 <- rnorm(20, 45, 5)
grp2 <- c(rnorm(10, 45, 10), rnorm(10, 52, 20))
df <- data.frame(cbind(grp1=grp1, grp2=grp2))
head(df)
bivarplot(df)
```

```
# Reshape the data into long format
df <- wide2long(df)
head(df)

# Differences between the basic stats
calcstatdif(df)
```

`ci2df`*Convert List to Data Frame*

Description

Converts a list of confidence intervals into a data frame.

Usage

```
ci2df(x)
```

Arguments

`x` A list where each element is a list of data frames or matrices containing confidence interval data as the result of the function [bootstrap](#).

Details

This function takes a list of confidence intervals and converts it into a data frame. Each row represents a method, and each column represents a statistic. Confidence intervals are formatted as strings in the form `[lower, upper]`.

Value

A data frame with rows representing methods and columns representing statistics, containing the confidence intervals.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

See Also

[bootstrap](#)

Examples

```
ciresults <- list(
  method1 = data.frame(lower = c(-0.1, 0.2), upper = c(0.3, 0.4), row.names = c("stat1", "stat2")),
  method2 = data.frame(lower = c(0.2, 0.3), upper = c(0.4, 0.5), row.names = c("stat1", "stat2"))
)
ciresults

cidf <- ci2df(ciresults)
cidf
```

descstats

Descriptive Statistics

Description

This function calculates various descriptive statistics for a given numeric vector. These statistics include measures of central tendency, dispersion, skewness, kurtosis, and some robust estimators.

Usage

```
descstats(x, trim = 0.1, k = 1.5)
```

Arguments

x	A numeric vector.
trim	The fraction (0 to 0.5) of observations to be trimmed from each end of the vector is used to calculate the trimmed mean and winsorized mean. The default is 0.1.
k	The robustness parameter for the Huber M-estimator. The default is 1.5.

Details

In order to determine an appropriate k value for the Huber M-estimator some experiments might be needed. In the literature, commonly used k values typically range from 1.5 to 2. Users can start by choosing any value within this range. However, to determine an appropriate k within a given range, it is also selected by performing Huber estimations for each k value within this range, as shown in the example below. In the output, the estimated Huber M-estimator values can be checked on a plot. Select k values where a smooth trend or plateau is reached. If the Huber M-estimator values stabilize after a certain k value, that k value may be appropriate. Finally, if there are outliers and you want to reduce their impact, you can use smaller k values.

Value

A list containing the computed descriptive statistics, including:

n	The number of observations
min	The minimum value
max	The maximum value

mean	The mean
se	The standard error of the mean
sd	The standard deviation
trmean	The trimmed mean
med	The median
mad	The median absolute deviation (MAD), a robust statistic for measuring variability in data.
skew	The skewness
kurt	The excess kurtosis measures how peaked or flat a distribution is compared to a normal distribution. Subtracting 3 centers the measure relative to the kurtosis of a normal distribution, which is always 3.
winsmean	The Winsorized mean
hubermean	The Huber's M-estimator of location
range	The range
iqr	The interquartile range

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

Examples

```
set.seed(123)
x <- rnorm(100, mean=50, sd=5)
descriptives <- descstats(x)
as.data.frame(descriptives)

descriptives$mean
descriptives$se

# Determining the appropriate k in a given set of different k values.
# This parameter is used to calculate the Huber M-estimator of the location
# Array of k values for testing
k <- seq(0, 5, by = 0.1)
k <- k[k > 0]
result <- sapply(k, function(y) descstats(x, k = y)$hubermean)
names(result) <- paste0("k=", k)
result

plot(k, result, type = "b", col = "blue", pch = 19, ylab = "Huber's mean")

descstats(x, k=2, trim=0.05)$hubermean
```

`ghdist`*Random Sample from G-H Distribution*

Description

Generates a random sample from the G-and-H (GH) distribution with specified parameters.

Usage

```
ghdist(n = 30, A = 0, B = 1, g = 0, h = 0)
```

Arguments

<code>n</code>	An integer specifying the sample size. The default is 30.
<code>A</code>	A numeric value specifying the location parameter. The default is 0.
<code>B</code>	A numeric value specifying the scale parameter. The default is 1. Must be positive.
<code>g</code>	A numeric value specifying the skewness parameter. The default is 0.
<code>h</code>	A numeric value specifying the kurtosis parameter. The default is 0. Must be zero or positive.

Details

The gh distribution is a flexible distribution defined by four parameters: A (location), B (scale), g (skewness), and h (kurtosis). The parameter B must be positive, and h must be zero or positive. This function generates random samples from the gh distribution using these parameters.

The GH distribution was introduced by John W. Tukey in 1977 as a way to model data with varying degrees of skewness and kurtosis. The distribution is defined by transforming standard normal random variables using the g and h parameters to control skewness and kurtosis, respectively.

Value

A numeric vector of length n containing the generated random samples.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

References

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

See Also

[rnorm](#)

Examples

```

set.seed(50)
A <- ghdist(100, 50, 2, g=0, h=0)
B <- ghdist(100, 50, 3, g=0.5, h=0.5)
ds <- data.frame(A=A, B=B)
head(ds)
bivarplot(ds)

```

groupcompare

*Comparing Two Groups with Various Descriptive Statistics***Description**

Performs a variety of statistical tests and generates plots to compare two groups.

Usage

```

groupcompare (ds, paired=FALSE, cl=0.95, alternative="two.sided",
  qtest=TRUE, q=seq(0.1, 0.9, by=0.1), qt=7, lognorm=TRUE,
  R=3000, plots=TRUE, out=FALSE, verbose=TRUE)

```

Arguments

ds	A data frame in long format, where the first column represents the observations and the second column the group names or labels.
paired	A logical flag indicating whether the group data is paired (i.e., related). Set to TRUE for paired groups.
cl	Confidence level for the interval. The default is 0.95. The significance level (Alpha, Type I error) equals 1-cl.
alternative	Type of alternative hypothesis: <code>"two.sided"</code> , <code>"less"</code> , or <code>"greater"</code> . The default is <code>"two.sided"</code> , corresponding to a two-tailed test.
qtest	Logical; if <code>TRUE</code> , performs a quantile test. The default is <code>TRUE</code> .
q	A vector of quantile probabilities specifying the quantiles to be compared. The default is <code>seq(0.1, 0.9, by = 0.1)</code> , corresponding to the 10th through 90th percentiles (P10 to P90).
qt	An integer between 0 and 9 specifying the quantile calculation method. The default is 7.
lognorm	Logical; if <code>TRUE</code> , performs a logarithmic transformation on data. The default is <code>TRUE</code> .
R	An integer indicating the number of permutations or bootstrap samples. The default is 3000.
plots	Logical; if <code>TRUE</code> , generates bivariate plots for the data. The default is <code>TRUE</code> .
out	Logical; if <code>TRUE</code> , writes the result to a file named <code>'groupcompare.txt'</code> .
verbose	Logical; if <code>TRUE</code> displays the details for each test. The default is <code>TRUE</code> .

Details

This function calculates descriptive statistics and performs a t-test or Wilcoxon test on two groups of data, depending on the variance homogeneity test and normality test. It also generates various plots, including density plots, ECDF plots, boxplots, violin plots, QQ plots, symmetry plots, and empirical shift plots, to visualize the data.

Value

A list that contains the results of the statistical tests and the bivariate plots, if `plots` is TRUE.

<code>descriptives</code>	A data frame containing summary statistics, such as mean, standard deviation, and sample size, for each group.
<code>quantiles</code>	A data frame with calculated quantile values for specified probabilities for each group.
<code>test</code>	The outcome of the appropriate statistical test (e.g., t-test or Wilcoxon test), including test statistic and p-value.
<code>quantest</code>	A data frame containing quantile comparison results if <code>qtest</code> is set to TRUE.
<code>inference</code>	A string summarizing the conclusion of the statistical test performed, indicating whether the null hypothesis was rejected or not.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

See Also

[bootstrap](#), [permtest](#), [bivarplot](#)

Examples

```
# Test with default values
df <- data.frame(grp1 = rnorm(50), grp2 = rnorm(50))
df <- wide2long(df)
result <- groupcompare(df)
print(result)

# Test with daily weight gain of two quail breeds
data(quail)
result <- groupcompare(quail, alternative="two.sided", cl=0.95,
  R=200, plots=TRUE, out=FALSE, verbose=TRUE)
print(result)

# Test with milk trace mineral data
filepath <- system.file("extdata", "milkcomp.csv", package = "groupcompare")
milktrace <- read.table(filepath, header=TRUE, sep=",")
head(milktrace)
milkzinc <- as.data.frame(cbind(milktrace$Zn, milktrace$grp))
colnames(milkzinc) <- c("zn", "grp")
head(milkzinc)
tail(milkzinc)
```

```
result <- groupcompare(milkzinc, cl=0.99, alternative="greater",
  R=200, plots=TRUE, out=FALSE, verbose=TRUE)
print(result)
```

groupdata

Synthetic Data for Two Independent Groups

Description

This data set contains seven data frames with two variables from various distributions.

Usage

```
data(groupdata)
```

Format

A list containing 7 data frames:

df1 Normally distributed, equal means, and equal variances. Columns: A, B.

df2 Normally distributed, unequal means, and unequal variances. Columns: A, B.

df3 Normally distributed, equal means, and unequal variances. Columns: A, B.

df4 A normal, B right-skewed, equal means, and unequal variances. Columns: A, B.

df5 A normal, B right-skewed, equal means, and unequal variances. Columns: A, B.

df6 A normal, B right-skewed, equal means, and unequal variances. Columns: A, B.

df7 A normal, B right-skewed, equal means, and unequal variances. Columns: A, B.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

Examples

```
data(groupdata)
df1 <- groupdata$df1
head(df1)
bivarplot(df1)

df4 <- groupdata$df4
head(df4)
bivarplot(df4)
```

hdqe	<i>Harrell-Davis Quantile Estimator</i>
------	---

Description

Computes the Harrell-Davis quantile estimator for given quantile levels.

Usage

```
hdqe(x, Q = c(0.25, 0.5, 0.75))
```

Arguments

x	Numeric vector of data values.
Q	A numeric vector of quantile levels to estimate, between 0 and 1. Defaults to <code>c(0.25, 0.5, 0.75)</code> for the 25th, 50th, and 75th percentiles (Q1, Q2, Q3).

Details

The function computes the Harrell-Davis quantile estimator, which estimates data quantiles by calculating a weighted average of order statistics. The weights are based on the beta distribution.

Value

A numeric vector containing estimated quantiles.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

Examples

```
set.seed(1)
x <- sample(1:10, 50, replace=TRUE)
quantile(x, probs=c(0.25, 0.5, 0.75), type=1) # quantiles with Type 1 in stats::quantile
quantile(x, probs=c(0.25, 0.5, 0.75), type=7) # quantiles with Type 7 (default) in stats::quantile
hdqe(x, Q=c(0.25, 0.5, 0.75)) # quantiles with Harrell-Davis Estimator
```

`intnorm`*Inverse Normal Transformation Function*

Description

This function performs an inverse normal transformation on non-normally distributed data. Using the provided rank information, it can also revert the transformed data back to its original scale.

Usage

```
intnorm(x)
```

Arguments

`x` A numeric vector of data to be transformed.

Value

A numeric vector of the transformed values.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

Examples

```
set.seed(1)
xnonnormal <- c(rexp(25, rate = 1), rexp(25, rate = 0.2))
xnormal <- intnorm(xnonnormal)

# Plot density and perform the Shapiro-Wilk test for non-normal and normalized data
opar <- par(mfrow = c(1, 2))

# Non-normal data plot
density_xnonnormal <- density(xnonnormal)
plot(density_xnonnormal, main = "Density Plot of Non-Normal Data",
     xlab = "Value", ylab = "Density", col = "blue", lwd=2)
polygon(density_xnonnormal, col = rgb(1, 0, 0, 0.3))
shapirotest1 <- shapiro.test(xnonnormal)
mtext(paste("p-value:", round(shapirotest1$p.value, 4)),
      side = 3, line = 0.5, at = mean(xnonnormal), col = "black")
shapirotest1

# Normalized data plot
density_xnormal <- density(xnormal)
plot(density_xnormal, main = "Density Plot of Inverse Normalized Data",
     xlab = "Value", ylab = "Density", col = "blue", lwd=2)
polygon(density_xnormal, col = rgb(1, 0, 0, 0.3))
shapirotest2 <- shapiro.test(xnormal)
```

```
mtext(paste("p-value:", round(shapirotest2$p.value, 4)),
      side = 3, line = 0.5, at = mean(xnormal), col = "black")
shapirotest2

par(opar)
```

`levene.test`*Levene Test for Homogeneity of Variances*

Description

This function performs the Levene test to check the homogeneity of variances across groups.

Usage

```
levene.test(ds)
```

Arguments

`ds` A data frame with two columns: the first column contains numeric values, and the second column contains group labels.

Details

The function calculates the absolute deviations from the group means, performs an ANOVA on these deviations, and returns the p-value of the Levene test.

Value

A single numeric value named `Levene.p`, representing the p-value of the Levene test.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

Examples

```
df <- data.frame(values = c(1.1, 2.3, 2.5, 3.7, 1.2, 2.1, 3.4, 3.9),
                 groups = c("A", "A", "B", "B", "A", "A", "B", "B"))
levene.test(df)
```

`long2wide`*Convert Long-Format Data to Wide-Format Data*

Description

Converts long-format data to wide-format data by splitting based on groups.

Usage

```
long2wide(x)
```

Arguments

`x` A data frame or matrix with two columns. The first column must contain observations, and the second column must contain the group levels as factor.

Details

This function converts long-format data to wide format by splitting it based on unique groups in the second column. The resulting data frame has columns for each group, where each column contains the values of the first column for the corresponding group.

Value

A data frame with columns corresponding to unique groups and rows containing the values for each group.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

See Also

[wide2long](#)

Examples

```
# Normal distributed groups of data with different means and different variances
set.seed(21)
obs1 <- rnorm(20, 50, 5)
obs2 <- rnorm(20, 55, 3)
obs <- c(obs1, obs2)
grp <- c(rep(as.factor("A"), 20), rep(as.factor("B"), 20))
ds1 <- data.frame(obs = obs, group = grp)
head(ds1)

# Reshape data to long format
ds2 <- long2wide(ds1)
head(ds2)
bivarplot(ds2)
```

permtest *Permutation Test on Long-Format Data*

Description

Performs a permutation test on long-format data to evaluate differences between two groups using a specified test statistic.

Usage

```
permtest(x, y = NULL, statistic, alternative = "two.sided",
         Q = seq(0.1, 0.9, by = 0.1), qt = 7, R = 10000)
```

Arguments

x	A data frame or matrix containing the data in long format.
y	An optional second data frame or matrix. If provided, the data will be combined with x and converted to long format.
statistic	The function name to compute the statistic of interest.
alternative	Type of alternative hypothesis: "two.sided", "less", or "greater". The default is "two.sided", corresponding to a two-tailed test.
Q	A vector of quantile probabilities specifying the quantiles to be compared. The default is seq(0.1, 0.9, by = 0.1), corresponding to the 10th through 90th percentiles (P10 to P90).
qt	An integer between 0 and 9 specifying the quantile calculation method. The default is 7.
R	An integer indicating the number of permutations. The default is 10000.

Details

The function allows researchers to perform robust statistical testing by utilizing permutations. This approach does not rely on distributional assumptions and is particularly useful when the sample size is small or the data distribution is unknown. The test generates an empirical distribution of the test statistic by repeatedly permuting group labels and recalculating the statistic, providing p-values based on these permutations.

Value

A list containing the following components:

t0	The estimated statistics of the differences.
t	A matrix of the permuted values of the statistics.
pval	A numeric vector of p-values for each statistic.
alternative	The specified alternative hypothesis.
R	The number of permutations.

pvalsk	A numeric vector of p-values for skewness and kurtosis with two-sided alternative. These are used to check the similarity of distributions to decide whether to use a bootstrap and a permutation test for inference.
call	The matched call.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

See Also

[bootstrap](#)

Examples

```
# Group1 normal, Group 2 right skewed with equal means and different variances
set.seed(1199)
grp1 <- rnorm(20, 45, 5)
grp2 <- c(rnorm(10, 45, 10), rnorm(10, 52, 20))
df <- data.frame(cbind(grp1=grp1, grp2=grp2))
head(df)
bivarplot(df)

# Reshape the data into long format
df <- wide2long(df)
head(df)

# Differences between the basic stats
calcstatdif(df)

# Permutation test for the differences between the basic stats of two groups
result <- permtest(df, statistic = calcstatdif, alternative = "two.sided", R = 500)
result$pval

# Permutation with custom statistics
# A custom function to compute the differences between the group means
meancomp <- function(x, ...){
  meandif <- diff(rev(tapply(x[,1], x[,2], mean)))
  return(meandif)
}

# Permutation test with meancomp function
result <- permtest(x = df, statistic = meancomp, alternative="less", R=500)
result$pval
```

`quail`*Daily Weight Gains of Quail Breeds*

Description

A data frame containing daily weight gains (in grams) of two quail breeds during a fattening period.

Usage

```
data(quail)
```

Format

A data frame with 16 observations on the following 2 variables:

dwg A numeric vector containing daily weight gains (g).

genotype A factor with levels A and B indicating breeds or genotypes.

Details

The dataset is used to compare the daily weight gain performances of different breeds of quails.

Examples

```
data(quail)
head(quail)
```

`wide2long`*Convert Wide-Format Data to Long-Format Data*

Description

Converts wide-format data to long-format data.

Usage

```
wide2long(x, y = NULL, grpnames=NULL)
```

Arguments

<code>x</code>	A data frame or matrix containing two columns for the observations in groups.
<code>y</code>	Optional; a vector containing data to combine with <code>x</code> . If <code>NULL</code> , the second column of <code>x</code> will be used as <code>y</code> .
<code>grpnames</code>	Optional character string vector for the names of groups, e.g. <code>c("A", "B")</code> .

Details

This function converts wide-format data to long format by combining two columns of data into a single column and creating a grouping variable to distinguish the original data sources.

Value

A data frame with two columns: Obs containing the combined values of x and y, and group indicating the original data source.

Author(s)

Zeynel Cebeci, A. Firat Ozdemir, Engin Yildiztepe

See Also

[long2wide](#)

Examples

```
# Normal distributed groups of data with equal means and different variances
set.seed(2)
grp1 <- rnorm(20, 50, 5)
grp2 <- rnorm(20, 50, 9)
ds1 <- data.frame(A=grp1, B=grp2)
head(ds1)
bivarplot(ds1)

# Convert to long data
ds2 <- wide2long(ds1)
head(ds2)
```

Index

- * **G&H distribution**
 - ghdist, 15
- * **Huber mean**
 - calchubermeandif, 6
- * **analysis**
 - groupcompare, 16
- * **basic statistics**
 - calcstatdif, 11
- * **bivariate statistics**
 - groupcompare-package, 2
- * **bivariate**
 - bivarplot, 4
- * **bootstrapping**
 - bootstrap, 5
- * **bootstrap**
 - bootstrap, 5
 - groupcompare-package, 2
- * **comparative analysis**
 - bivarplot, 4
- * **comparison**
 - bivarplot, 4
- * **confidence interval**
 - ci2df, 12
- * **data analysis**
 - groupcompare, 16
 - groupcompare-package, 2
- * **data frame**
 - ci2df, 12
- * **data reshape**
 - long2wide, 22
 - wide2long, 25
- * **data shape**
 - long2wide, 22
 - wide2long, 25
- * **data structure**
 - long2wide, 22
 - wide2long, 25
- * **data visualization**
 - bivarplot, 4
- * **decile**
 - calcquantdif, 8
 - hdqe, 19
- * **descriptive statistics**
 - calcstatdif, 11
- * **difference**
 - calcstatdif, 11
- * **distribution**
 - ghdist, 15
- * **grouped data analysis**
 - calcstatdif, 11
- * **group**
 - calcstatdif, 11
- * **harrell-davis estimator**
 - hdqe, 19
- * **list conversion**
 - ci2df, 12
- * **list**
 - ci2df, 12
- * **long data**
 - long2wide, 22
 - wide2long, 25
- * **mean**
 - calchubermeandif, 6
- * **non-parametric tests**
 - groupcompare-package, 2
- * **non-parametric**
 - bootstrap, 5
 - permtest, 23
- * **outlier**
 - calchubermeandif, 6
- * **parametric tests**
 - groupcompare-package, 2
- * **percentile**
 - calcquantdif, 8
 - hdqe, 19
- * **permutation test**
 - groupcompare-package, 2
 - permtest, 23

- * **permutation**
 - groupcompare-package, 2
 - permttest, 23
- * **plotting**
 - bivarplot, 4
- * **plot**
 - bivarplot, 4
- * **probability distribution**
 - ghdist, 15
- * **quantile difference**
 - calcquantdif, 8
- * **quantiles comparison**
 - groupcompare-package, 2
- * **quantiles**
 - calcquantdif, 8
 - hdqe, 19
- * **quantile**
 - calcquantdif, 8
 - hdqe, 19
- * **quartile**
 - calcquantdif, 8
 - hdqe, 19
- * **random sampling**
 - ghdist, 15
- * **resampling statistics**
 - bootstrap, 5
 - permttest, 23
- * **resampling**
 - bootstrap, 5
- * **robust statistics**
 - calchubermeandif, 6
- * **sampling**
 - ghdist, 15
- * **statistical testing**
 - bootstrap, 5
 - groupcompare, 16
 - permttest, 23
- * **statistical tests**
 - groupcompare-package, 2
- * **statistical test**
 - bootstrap, 5
 - permttest, 23
- * **statistics**
 - calcstatdif, 11
 - ghdist, 15
- * **visualization**
 - bivarplot, 4
 - groupcompare, 16
- * **wide data**
 - long2wide, 22
 - wide2long, 25
- bivarplot, 2, 3, 4, 17
- bootstrap, 2, 3, 5, 12, 17, 24
- calchubermeandif, 2, 6
- calcquantdif, 2, 8, 11
- calcquantile, 2, 9
- calcstatdif, 3, 9, 11
- ci2df, 3, 12
- descstats, 3, 13
- ghdist, 3, 15
- groupcompare, 2, 16
- groupcompare-package, 2
- groupdata, 3, 18
- hdqe, 3, 8–10, 19
- intnorm, 3, 20
- levene.test, 3, 21
- long2wide, 3, 22, 26
- median, 7
- permttest, 3, 6, 17, 23
- quail, 3, 25
- quantile, 8–10
- rnorm, 15
- wide2long, 3, 22, 25