

# Package ‘pairwiseComparisons’

December 3, 2020

**Type** Package

**Title** Multiple Pairwise Comparison Tests

**Version** 3.1.1

**Maintainer** Indrajeet Patil <patilindrajeet.science@gmail.com>

**Description** Multiple pairwise comparison tests on tidy data for one-way analysis of variance for both between-subjects and within-subjects designs. Currently, it supports only the most common types of statistical analyses and tests: parametric (Welch's and Student's t-test), nonparametric (Durbin-Conover and Dunn test), robust (Yuen's trimmed means test), and Bayes Factor (Student's t-test).

**License** GPL-3 | file LICENSE

**URL** <https://indrajeetpatil.github.io/pairwiseComparisons/>,  
<https://github.com/IndrajeetPatil/pairwiseComparisons>

**BugReports** <https://github.com/IndrajeetPatil/pairwiseComparisons/issues>

**Depends** R (>= 3.6.0)

**Imports** BayesFactor, dplyr, ipmisc, parameters (>= 0.10.0), PMCMRplus (>= 1.7.1), purrr, rlang, stats, WRS2

**Suggests** knitr, rmarkdown, spelling, testthat

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**RoxygenNote** 7.1.1

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**Config/testthat/start-first** watcher, parallel\*

**NeedsCompilation** no

**Author** Indrajeet Patil [cre, aut, cph]  
(<<https://orcid.org/0000-0003-1995-6531>>, @patilindrajeets)

**Repository** CRAN

**Date/Publication** 2020-12-03 13:10:02 UTC

## R topics documented:

bugs_long . . . . .	2
iris_long . . . . .	3
movies_long . . . . .	4
movies_wide . . . . .	5
pairwise_caption . . . . .	6
pairwise_comparisons . . . . .	6
p_adjust_text . . . . .	10

<b>Index</b>	<b>12</b>
--------------	-----------

---

bugs_long	<i>Tidy version of the "Bugs" dataset.</i>
-----------	--

---

### Description

Tidy version of the "Bugs" dataset.

### Usage

```
bugs_long
```

### Format

A data frame with 372 rows and 6 variables

- subject. Dummy identity number for each participant.
- gender. Participant's gender (Female, Male).
- region. Region of the world the participant was from.
- education. Level of education.
- condition. Condition of the experiment the participant gave rating for (**LDLF**: low frighteningness and low disgustingness; **LFHD**: low frighteningness and high disgustingness; **HFHD**: high frighteningness and low disgustingness; **HFHD**: high frighteningness and high disgustingness).
- desire. The desire to kill an arthropod was indicated on a scale from 0 to 10.

### Details

This data set, "Bugs", provides the extent to which men and women want to kill arthropods that vary in frighteningness (low, high) and disgustingness (low, high). Each participant rates their attitudes towards all anthropods. Subset of the data reported by Ryan et al. (2013).

### Source

<https://www.sciencedirect.com/science/article/pii/S0747563213000277>

**Examples**

```
dim(bugs_long)
head(bugs_long)
dplyr::glimpse(bugs_long)
```

---

iris\_long

*Edgar Anderson's Iris Data in long format.*

---

**Description**

Edgar Anderson's Iris Data in long format.

**Usage**

```
iris_long
```

**Format**

A data frame with 600 rows and 5 variables

- id. Dummy identity number for each flower (150 flowers in total).
- Species. The species are *Iris setosa*, *versicolor*, and *virginica*.
- condition. Factor giving a detailed description of the attribute (Four levels: "Petal.Length", "Petal.Width", "Sepal.Length", "Sepal.Width").
- attribute. What attribute is being measured ("Sepal" or "Petal").
- measure. What aspect of the attribute is being measured ("Length" or "Width").
- value. Value of the measurement.

**Details**

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

This is a modified dataset from datasets package.

**Examples**

```
dim(iris_long)
head(iris_long)
dplyr::glimpse(iris_long)
```

---

`movies_long`*Movie information and user ratings from IMDB.com (long format).*

---

### Description

Movie information and user ratings from IMDB.com (long format).

### Usage

```
movies_long
```

### Format

A data frame with 1,579 rows and 8 variables

- `title`. Title of the movie.
- `year`. Year of release.
- `budget`. Total budget (if known) in US dollars
- `length`. Length in minutes.
- `rating`. Average IMDB user rating.
- `votes`. Number of IMDB users who rated this movie.
- `mpaa`. MPAA rating.
- `genre`. Different genres of movies (action, animation, comedy, drama, documentary, romance, short).

### Details

Modified dataset from `ggplot2movies` package.

The internet movie database, <https://imdb.com/>, is a website devoted to collecting movie data supplied by studios and fans. It claims to be the biggest movie database on the web and is run by amazon.

Movies were are identical to those selected for inclusion in `movies_wide` but this dataset has been constructed such that every movie appears in one and only one genre category.

### Source

<https://CRAN.R-project.org/package=ggplot2movies>

### Examples

```
dim(movies_long)
head(movies_long)
dplyr::glimpse(movies_long)
```

---

movies_wide	<i>Movie information and user ratings from IMDB.com (wide format).</i>
-------------	--

---

### Description

Movie information and user ratings from IMDB.com (wide format).

### Usage

```
movies_wide
```

### Format

A data frame with 1,579 rows and 13 variables

- title. Title of the movie.
- year. Year of release.
- budget. Total budget in millions of US dollars
- length. Length in minutes.
- rating. Average IMDB user rating.
- votes. Number of IMDB users who rated this movie.
- mpaa. MPAA rating.
- action, animation, comedy, drama, documentary, romance, short. Binary variables representing if movie was classified as belonging to that genre.
- NumGenre. The number of different genres a film was classified in an integer between one and four

### Details

Modified dataset from `ggplot2movies` package.

The internet movie database, <https://imdb.com/>, is a website devoted to collecting movie data supplied by studios and fans. It claims to be the biggest movie database on the web and is run by amazon.

Movies were selected for inclusion if they had a known length and had been rated by at least one imdb user. Small categories such as documentaries and NC-17 movies were removed.

### Source

<https://CRAN.R-project.org/package=ggplot2movies>

### Examples

```
dim(movies_wide)
head(movies_wide)
dplyr::glimpse(movies_wide)
```

---

pairwise_caption	<i>Expression containing details about pairwise comparison test</i>
------------------	---

---

### Description

This returns an expression containing details about the pairwise comparison test and the  $p$ -value adjustment method. These details are typically included in the ggstatsplot package plots as a caption.

### Usage

```
pairwise_caption(
  caption,
  test.description,
  pairwise.display = "significant",
  ...
)
```

### Arguments

caption	Additional text to be included in the plot.
test.description	Text describing the details of the test.
pairwise.display	Decides which pairwise comparisons to display. Available options are "significant" (abbreviation accepted: "s") or "non-significant" (abbreviation accepted: "ns") or "everything"/"all". The default is "significant".
...	Ignored.

### Examples

```
library(pairwiseComparisons)
pairwise_caption("my caption", "Student's t-test")
```

---

pairwise_comparisons	<i>Multiple pairwise comparison tests with tidy data</i>
----------------------	--

---

### Description

Calculate parametric, non-parametric, robust, and Bayes Factor pairwise comparisons between group levels with corrections for multiple testing.

**Usage**

```
pairwise_comparisons(
  data,
  x,
  y,
  subject.id = NULL,
  type = "parametric",
  paired = FALSE,
  var.equal = FALSE,
  tr = 0.1,
  bf.prior = 0.707,
  p.adjust.method = "holm",
  k = 2L,
  ...
)
```

**Arguments**

<code>data</code>	A dataframe from which variables specified are to be taken. A matrix or tables will <b>not</b> be accepted.
<code>x</code>	The grouping variable from the dataframe <code>data</code> .
<code>y</code>	The response (a.k.a. outcome or dependent) variable from the dataframe <code>data</code> .
<code>subject.id</code>	In case of repeated measures design ( <code>paired = TRUE</code> , i.e.), this argument specifies the subject or repeated measures id. Note that if this argument is <code>NULL</code> (which is the default), the function assumes that the data has already been sorted by such an id by the user and creates an internal identifier. So if your data is <b>not</b> sorted and you leave this argument unspecified, the results can be inaccurate.
<code>type</code>	Type of statistic expected ("parametric" or "nonparametric" or "robust" or "bayes"). Corresponding abbreviations are also accepted: "p" (for parametric), "np" (nonparametric), "r" (robust), or "bf" resp.
<code>paired</code>	Logical that decides whether the experimental design is repeated measures/within-subjects or between-subjects. The default is <code>FALSE</code> .
<code>var.equal</code>	a logical variable indicating whether to treat the two variances as being equal. If <code>TRUE</code> then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
<code>tr</code>	Trim level for the mean when carrying out robust tests. If you get error stating "Standard error cannot be computed because of Winsorized variance of 0 (e.g., due to ties). Try to decrease the trimming level.", try to play around with the value of <code>tr</code> , which is by default set to 0.1. Lowering the value might help.
<code>bf.prior</code>	A number between 0.5 and 2 (default 0.707), the prior width to use in calculating Bayes factors.
<code>p.adjust.method</code>	Adjustment method for <i>p</i> -values for multiple comparisons. Possible methods are: "holm" (default), "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

k                    Number of digits after decimal point (should be an integer) (Default: k = 2L).  
 ...                    Current ignored.

### Value

A tibble dataframe containing two columns corresponding to group levels being compared with each other (group1 and group2) and p.value column corresponding to this comparison. The dataframe will also contain a p.value.label column containing a *label* for this *p*-value, in case this needs to be displayed in `ggsignif::geom_ggsignif`. In addition to these common columns across the different types of statistics, there will be additional columns specific to the type of test being run.

This function provides a unified syntax to carry out pairwise comparison tests and internally relies on other packages to carry out these tests. For more details about the included tests, see the documentation for the respective functions:

- *parametric* : `stats::pairwise.t.test()` (paired) and `PMCMRplus::gamesHowellTest()` (unpaired)
- *non-parametric* : `PMCMRplus::durbinAllPairsTest()` (paired) and `PMCMRplus::kwAllPairsDunnTest()` (unpaired)
- *robust* : `WRS2::rmmcp()` (paired) and `WRS2::lincon()` (unpaired)
- *Bayes Factor* : `BayesFactor::ttestBF()`

### Examples

```
# for reproducibility
set.seed(123)
library(pairwiseComparisons)

# show me all columns and make the column titles bold
options(tibble.width = Inf, pillar.bold = TRUE, pillar.subtle_num = TRUE)

#----- between-subjects design -----

# parametric
# if `var.equal = TRUE`, then Student's t-test will be run
pairwise_comparisons(
  data = mtcars,
  x = cyl,
  y = wt,
  type = "parametric",
  var.equal = TRUE,
  paired = FALSE,
  p.adjust.method = "none"
)

# if `var.equal = FALSE`, then Games-Howell test will be run
pairwise_comparisons(
  data = mtcars,
  x = cyl,
```



```
    y = wt,
    type = "parametric",
    var.equal = FALSE,
    paired = FALSE,
    p.adjust.method = "bonferroni"
  )

# non-parametric (Dunn test)
pairwise_comparisons(
  data = mtcars,
  x = cyl,
  y = wt,
  type = "nonparametric",
  paired = FALSE,
  p.adjust.method = "none"
)

# robust (Yuen's trimmed means t-test)
pairwise_comparisons(
  data = mtcars,
  x = cyl,
  y = wt,
  type = "robust",
  paired = FALSE,
  p.adjust.method = "fdr"
)

# Bayes Factor (Student's t-test)
pairwise_comparisons(
  data = mtcars,
  x = cyl,
  y = wt,
  type = "bayes",
  paired = FALSE
)

#----- within-subjects design -----

# parametric (Student's t-test)
pairwise_comparisons(
  data = bugs_long,
  x = condition,
  y = desire,
  subject.id = subject,
  type = "parametric",
  paired = TRUE,
  p.adjust.method = "BH"
)

# non-parametric (Durbin-Conover test)
pairwise_comparisons(
  data = bugs_long,
  x = condition,
```

```

y = desire,
subject.id = subject,
type = "nonparametric",
paired = TRUE,
p.adjust.method = "BY"
)

# robust (Yuen's trimmed means t-test)
pairwise_comparisons(
  data = bugs_long,
  x = condition,
  y = desire,
  subject.id = subject,
  type = "robust",
  paired = TRUE,
  p.adjust.method = "hommel"
)

# Bayes Factor (Student's t-test)
pairwise_comparisons(
  data = bugs_long,
  x = condition,
  y = desire,
  subject.id = subject,
  type = "bayes",
  paired = TRUE
)

```

---

p\_adjust\_text

---

*Preparing text to describe which p-value adjustment method was used*


---

### Description

Preparing text to describe which  $p$ -value adjustment method was used

### Usage

```
p_adjust_text(p.adjust.method)
```

### Arguments

p.adjust.method

Adjustment method for  $p$ -values for multiple comparisons. Possible methods are: "holm" (default), "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

### Value

Standardized text description for what method was used.

**Examples**

```
library(pairwiseComparisons)
p_adjust_text("none")
p_adjust_text("BY")
```

# Index

## \* datasets

- bugs\_long, 2
- iris\_long, 3
- movies\_long, 4
- movies\_wide, 5

BayesFactor::ttestBF(), 8

bugs\_long, 2

iris\_long, 3

movies\_long, 4

movies\_wide, 5

p\_adjust\_text, 10

pairwise\_caption, 6

pairwise\_comparisons, 6

PMCMRplus::durbinAllPairsTest(), 8

PMCMRplus::gamesHowellTest(), 8

PMCMRplus::kwAllPairsDunnTest(), 8

stats::pairwise.t.test(), 8

WRS2::lincon(), 8

WRS2::rmmcp(), 8