Package 'pseudorank'

July 23, 2025

Title Pseudo-Ranks

Version 1.0.4

Date 2025-02-18

Maintainer Martin Happ <statistics@happ.co.at>

Description Efficient calculation of pseudo-ranks and (pseudo)-rank based test statistics. In case of equal sample sizes, pseudo-ranks and mid-ranks are equal. When used for inference mid-ranks may lead to paradoxical results. Pseudo-ranks are in general not affected by such a problem. See Happ et al. (2020, <doi:10.18637/jss.v095.c01>) for details.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports Rcpp (>= 0.12.16), doBy

LinkingTo Rcpp

URL https://github.com/happma/pseudorank/

BugReports https://github.com/happma/pseudorank/issues/

RoxygenNote 7.3.2

Suggests testthat

NeedsCompilation yes

Author Martin Happ [aut, cre] (ORCID: <https://orcid.org/0000-0003-0009-2665>),
Georg Zimmermann [aut],
Arne C. Bathke [aut],
Edgar Brunner [aut]

Repository CRAN

Date/Publication 2025-02-20 17:20:02 UTC

Contents

`pseudorank-package` *Pseudo-Ranks*

Description

This packge provides functions to calculate pseudo-ranks. Rank based test statistics (e.g. Kruskal-Wallis test) may lead to paradoxical results as the weighted relative effects (based on ranks) depend on the sample sizes (Brunner, 2018). Pseudo-ranks do not have these problems.

Author(s)

Maintainer: Martin Happ <martin.happ@aon.at>

References

Brunner, E., Konietschke, F., Bathke, A. C., & Pauly, M. (2018). Ranks and Pseudo-Ranks-Paradoxical Results of Rank Tests. arXiv preprint arXiv:1802.05650.

Brunner, E., Bathke, A.C., and Konietschke, F. (2018a). Rank- and Pseudo-Rank Procedures for Independent Observations in Factorial Designs - Using R and SAS. Springer Series in Statistics, Springer, Heidelberg. ISBN: 978-3-030-02912-8.

Happ M, Zimmermann G, Brunner E, Bathke AC (2020). Pseudo-Ranks: How to Calculate Them Efficiently in R. Journal of Statistical Software, Code Snippets, *95*(1), 1-22. doi: 10.18637/jss.v095.c01 (URL:https://doi.org/10.18637/jss.v095.c01).

`hettmansperger_norton_test`
Hettmansperger-Norton Trend Test for k-Samples

Description

This function calculates the Hettmansperger-Norton trend test using pseudo-ranks under the null hypothesis H0F: F_1 = ... F_k.

Usage

```
hettmansperger_norton_test(x, ...)

## S3 method for class 'numeric'
hettmansperger_norton_test(
  x,
  y,
  na.rm = FALSE,
  alternative = c("decreasing", "increasing", "custom"),
  trend = NULL,
  pseudoranks = TRUE,
  ...
)

## S3 method for class 'formula'
hettmansperger_norton_test(
  formula,
  data,
  na.rm = FALSE,
  alternative = c("decreasing", "increasing", "custom"),
  trend = NULL,
  pseudoranks = TRUE,
  ...
)
```

Arguments

x	vector containing the observations
...	further arguments are ignored
y	vector specifiying the group to which the observations from the x vector belong to
na.rm	a logical value indicating if NA values should be removed
alternative	either decreasing (trend k, k-1, ..., 1) or increasing (1, 2, ..., k) or custom (then argument trend must be specified)
trend	custom numeric vector indicating the trend for the custom alternative, only used if alternative = "custom"
pseudoranks	logical value indicating if pseudo-ranks or ranks should be used
formula	formula object
data	data.frame containing the variables in the formula (observations and group)

Value

Returns an object.

References

Brunner, E., Bathke, A.C., and Konietschke, F. (2018a). Rank- and Pseudo-Rank Procedures for Independent Observations in Factorial Designs - Using R and SAS. Springer Series in Statistics, Springer, Heidelberg. ISBN: 978-3-030-02912-8.

Happ M, Zimmermann G, Brunner E, Bathke AC (2020). Pseudo-Ranks: How to Calculate Them Efficiently in R. Journal of Statistical Software, Code Snippets, *95*(1), 1-22. doi: 10.18637/jss.v095.c01 (URL:https://doi.org/10.18637/jss.v095.c01).

Hettmansperger, T. P., & Norton, R. M. (1987). Tests for patterned alternatives in k-sample problems. Journal of the American Statistical Association, 82(397), 292-299

Examples

```
# create some data, please note that the group factor needs to be ordered
df <- data.frame(data = c(rnorm(40, 3, 1), rnorm(40, 2, 1), rnorm(20, 1, 1)),
  group = c(rep(1,40),rep(2,40),rep(3,20)))
df$group <- factor(df$group, ordered = TRUE)

# you can either test for a decreasing, increasing or custom trend
hettmansperger_norton_test(df$data, df$group, alternative="decreasing")
hettmansperger_norton_test(df$data, df$group, alternative="increasing")
hettmansperger_norton_test(df$data, df$group, alternative="custom", trend = c(1, 3, 2))
```

---

kruskal_wallis_test *Kruskal-Wallis Test*

Description

This function calculates the Kruskal-Wallis test using pseudo-ranks under the null hypothesis H0F: F_1 = ... F_k.

Usage

```
kruskal_wallis_test(x, ...)

## S3 method for class 'numeric'
kruskal_wallis_test(x, grp, na.rm = FALSE, pseudoranks = TRUE, ...)

## S3 method for class 'formula'
kruskal_wallis_test(formula, data, na.rm = FALSE, pseudoranks = TRUE, ...)
```

Arguments

x	numeric vector containing the data
...	further arguments are ignored
grp	factor specifying the groups
na.rm	a logical value indicating if NA values should be removed

pseudoranks	logical value indicating if pseudo-ranks or ranks should be used
formula	optional formula object
data	optional data.frame of the data

Value

Returns an object of class 'pseudorank'

References

Brunner, E., Bathke, A.C., and Konietschke, F. (2018a). Rank- and Pseudo-Rank Procedures for Independent Observations in Factorial Designs - Using R and SAS. Springer Series in Statistics, Springer, Heidelberg. ISBN: 978-3-030-02912-8.

Examples

```
x = c(1, 1, 1, 1, 2, 3, 4, 5, 6)
grp = as.factor(c('A','A','B','B','B','D','D','D','D'))

# calculate Kruskal-Wallis test using pseudo-ranks
kruskal_wallis_test(x, grp, na.rm = FALSE, pseudoranks = TRUE)
```

ParadoxicalRanks *Artifical data of 54 subjects*

Description

An artificial dataset containing data of 54 subjects where where a substance was administered in three different concentrations (1,2 and 3). This data set can be used to show the paradoxical results obtained from rank tests, i.e., the Hettmansperger-Norton test.

Usage

```
data(ParadoxicalRanks)
```

Format

A data frame with 54 rows and 2 variables.

Details

The columns are as follows:

- conc. Grouping variable specifying which concentration was used. This factor is ordered, i.e., $1 < 2 < 3$.
- score. The response variable.

References

Happ M, Zimmermann G, Brunner E, Bathke AC (2020). Pseudo-Ranks: How to Calculate Them Efficiently in R. Journal of Statistical Software, Code Snippets, *95*(1), 1-22. doi: 10.18637/jss.v095.c01 (URL:https://doi.org/10.18637/jss.v095.c01).

Examples

```
data("ParadoxicalRanks")
dat <- ParadoxicalRanks

set.seed(1)
n <- c(60, 360, 120)
x1 <- sample(subset(dat, dat$conc == 1)$score, n[1], replace = TRUE)
x2 <- sample(subset(dat, dat$conc == 2)$score, n[2], replace = TRUE)
x3 <- sample(subset(dat, dat$conc == 3)$score, n[3], replace = TRUE)


dat <- data.frame(score = c(x1, x2, x3),
  conc = factor(c( rep(1,n[1]), rep(2,n[2]), rep(5,n[3]) ), ordered=TRUE) )

# Hettmansperger-Norton test with ranks (pseudorannks = FALSE) returns a small p-value (0.011).
# In contrast, the pseudo-rank test returns a large p-value (0.42). By changing the ratio of
# group sizes, we can also obtain a significant decreasing trend with ranks, e.g.
# n <- c(260,20,260) and the same seed.
hettmansperger_norton_test(score ~ conc, data = dat, pseudoranks = FALSE,
  alternative = "increasing")
hettmansperger_norton_test(score ~ conc, data = dat, pseudoranks = TRUE,
  alternative = "increasing")
```

---

pseudorank *Calculation of Pseudo-Ranks*

Description

Calculation of (mid) pseudo-ranks of a sample. In case of ties (i.e. equal values), the average of min pseudo-ranks and max-pseudo-ranks are taken (similar to rank with ties.method="average").

Usage

```
pseudorank(x, ...)

## S3 method for class 'numeric'
pseudorank(x, y, na.last = NA, ties.method = c("average", "max", "min"), ...)

## S3 method for class 'formula'
pseudorank(
  formula,
  data,
```

```
  na.last = NA,
  ties.method = c("average", "max", "min"),
  ...
)
```

Arguments

x	vector containing the observations
...	further arguments
y	vector specifiying the group to which the observations from the x vector belong to
na.last	for controlling the treatment of NAs. If TRUE, missing values in the data are put last; if FALSE, they are put first; if NA, they are removed (recommended).
ties.method	type of pseudo-ranks: either 'average' (recommended), 'min' or 'max'.
formula	formula object
data	data.frame containing the variables in the formula (observations and group)

Value

Returns a numerical vector containing the pseudo-ranks.

References

Brunner, E., Bathke, A.C., and Konietschke, F. (2018a). Rank- and Pseudo-Rank Procedures for Independent Observations in Factorial Designs - Using R and SAS. Springer Series in Statistics, Springer, Heidelberg. ISBN: 978-3-030-02912-8.

Happ M, Zimmermann G, Brunner E, Bathke AC (2020). Pseudo-Ranks: How to Calculate Them Efficiently in R. Journal of Statistical Software, Code Snippets, *95*(1), 1-22. doi: 10.18637/jss.v095.c01 (URL:https://doi.org/10.18637/jss.v095.c01).

Examples

```
df <- data.frame(data = round(rnorm(100)), group = c(rep(1,40),rep(2,40),rep(3,20)))
df$group <- as.factor(df$group)

## two ways to calculate pseudo-ranks

# Variant 1: use a vector for the data and a group vector
pseudorank(df$data,df$group)

# Variant 2: use a formula object, Note that only one group factor can be used
# that is, in data~group*group2 only 'group' will be used
pseudorank(data~group,df)
```

psrank *Calculation of Pseudo-Ranks (Deprecated)*

Description

Calculation of (mid) pseudo-ranks of a sample. In case of ties (i.e. equal values), the average of min pseudo-ranks and max-pseudo-ranks are taken (similar to rank with ties.method="average").

Usage

```
psrank(x, ...)
```

Arguments

x	vector containing the observations
...	further arguments (see help for pseudorank)

Value

Returns a numerical vector containing the pseudo-ranks.

References

Happ M, Zimmermann G, Brunner E, Bathke AC (2020). Pseudo-Ranks: How to Calculate Them Efficiently in R. Journal of Statistical Software, Code Snippets, *95*(1), 1-22. doi: 10.18637/jss.v095.c01 (URL:https://doi.org/10.18637/jss.v095.c01).

Examples

```
df <- data.frame(data = round(rnorm(100)), group = c(rep(1,40),rep(2,40),rep(3,20)))
df$group <- as.factor(df$group)

## two ways to calculate pseudo-ranks

# Variant 1: use a vector for the data and a group vector
pseudorank(df$data,df$group)

# Variant 2: use a formula object, Note that only one group factor can be used
# that is, in data~group*group2 only 'group' will be used
pseudorank(data~group,df)
```

Index