Package ‘opa’
August 7, 2022

Type Package
Title An Implementation of Ordinal Pattern Analysis
Version 0.5.2
Description Quantifies hypothesis to data fit for repeated measures and longitudinal data, as described by Thorngate (1987) <doi:10.1016/S0166-4115(08)60083-7> and Grice et al., (2015) <doi:10.1177/2158244015604192>. Hypothesis and data are encoded as pairwise relative orderings which are then compared to determine the percentage of orderings in the data that are matched by the hypothesis.
License GPL (>= 3)
URL https://github.com/timbeechey/opa
BugReports https://github.com/timbeechey/opa/issues
Encoding UTF-8
RoxygenNote 7.2.1
LinkingTo cpp11, Rcpp
Imports Rcpp
Suggests rmarkdown, knitr, testthat (>= 3.0.0)
Config/testthat/edition 3
SystemRequirements C++11
NeedsCompilation yes
Author Timothy Beechey [aut, cre] (<https://orcid.org/0000-0001-8858-946X>)
Maintainer Timothy Beechey <tim.beechey@proton.me>
Repository CRAN
Date/Publication 2022-08-07 14:30:02 UTC

R topics documented:

  compare_conditions ................................................................. 2
cval_plot ................................................................. 3
**compare_conditions**

Calculates PCCs and c-values based on pairwise comparison of conditions.

### Usage

```r
call_conditions(
  result,
  cval_method = "exact",
  nreps = 1000L,
  progress = FALSE
)
```

### Arguments

- **result**: an object of class "opafit" produced by a call to `opa()`.
- **cval_method**: a string, either "exact" or "stochastic"
- **nreps**: an integer, ignored if `cval_method = "exact"
- **progress**: a boolean indicating whether to display a progress bar

### Value

`compare_conditions` returns a list with the following elements:

- **pccs**: An upper triangle matrix containing PCCs calculated from each pairing of data columns, indicated by the matrix row and column names.
- **cvals**: An upper triangle matrix containing c-values calculated from each pairing of data columns, indicated by the matrix row and column names.
cval_plot

Examples

dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11),
                  t4 = c(10, 5, 11, 12))
opamod <- opa(dat, 1:4)
compare_conditions(opamod)

cval_plot

Plot individual chance values

Description

Plot individual chance values

Usage

cval_plot(m, threshold = NULL, title = TRUE, legend = TRUE)

Arguments

m an object of class "opafit"
threshold a boolean indicating whether to plot a threshold abline
title a boolean indicating whether to include a plot title
legend a boolean indicating whether to include a legend when n groups > 1

Value

No return value, called for side effects.

Examples

dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
cval_plot(opamod)
cval_plot(opamod, threshold = 0.1)
group_results

**Group-level PCC and chance values.**

**Description**

Group-level PCC and chance values.

**Usage**

`group_results(m, digits)`

**Arguments**

- `m`: an object of class "opafit" produced by `opa()`.
- `digits`: a positive integer.

**Details**

If the model was fitted with no grouping variable, a single PCC and c-value are returned. If a grouping variable was specified in the call to `opa` then PCCs and c-values are returned for each factor level of the grouping variable.

**Value**

a matrix with 1 row per group.

**Examples**

```r
dat <- data.frame(t1 = c(9, 4, 8, 10),
                   t2 = c(8, 8, 12, 10),
                   t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
group_results(opamod)
```

individual_results

**Individual-level PCC and chance values.**

**Description**

Individual-level PCC and chance values.

**Usage**

`individual_results(m, digits)`
**opa**

**Arguments**

- **m**: an object of class "opafit" produced by `opa()`
- **digits**: an integer

**Details**

If the model was fitted with no grouping variable, a matrix of PCCs and c-values are returned corresponding to the order of rows in the data. If the `opa` model was fitted with a grouping variable specified, a table of PCCs and c-values is returned ordered by factor level of the grouping variable.

**Value**

a matrix containing a column of PCC values and a column of c-values with 1 row per row of data.

**Examples**

```r
dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
individual_results(opamod)
```

**Description**

`opa` is used to fit ordinal pattern analysis models by computing the percentage of pair orderings in each row of data which are matched by corresponding pair orderings in a hypothesis, in addition the chance of a permutation of the data producing a percentage match as great.

**Usage**

```r
opa(
  dat,
  hypothesis,
  group = NULL,
  pairing_type = "pairwise",
  diff_threshold = 0,
  cval_method = "stochastic",
  nreps = 1000L,
  progress = FALSE
)
```
Arguments

dat a data frame  
hypothesis a numeric vector  
group an optional factor vector  
pairing_type a string  
diff_threshold a positive integer or floating point number  
cval_method a string, either "exact" or "stochastic  
nreps an integer, ignored if `cval_method = "exact"  
progress a boolean indicating whether to display a progress bar

Details

Data is expected in wide format with 1 row per individual and 1 column per measurement condition. Data must contain only columns consisting of numerical values of the dependent variable. The length of the hypothesis must be equal to the number of columns in the dependent variable data.frame `dat`. 

Any independent variable must be passed separately as a vector with the group keyword. The grouping vector must be a factor. 

pairing_type must be either "pairwise" or "adjacent". The "pairwise" option considered the relative ordering of every pair of observations in the data and every pair of elements of the hypothesis. The "adjacent" option considered the ordering of adjacent pairs only. If unspecified, the default is "pairwise". 

diff_threshold may be a positive integer or double. If unspecified a default zero threshold is used. The diff_threshold is never applied to the hypothesis. 

cval_method is either "stochastic" or "exact". The "stochastic" option generates random reorderings of each data row. The "exact" method generates every possible permutation of each data row. Care must be taken using the "exact" method since the number of permutations is the factorial of the number of columns in the data. For large numbers of data columns it is best to use the default "stochastic" method to sample orderings. 

nreps specifies the number of random reorderings to generate when using the "stochastic" method for computing chance values. The default value of nreps is 1000. If the cval_method = "exact" option is specified, nreps is ignored.

Value

opa returns an object of class "opafit". An object of class "opafit" is a list containing the following components:

**group_pcc** the percentage of pairwise orderings from all pooled data rows which were correctly classified by the hypothesis. 

**individual_pccs** a vector containing the percentage of pairwise orderings that were correctly classified by the hypothesis for each data row. 

**condition_pccs** a matrix containing PCCs for each pair of conditions, or a list containing such a matrix for each group level if a grouping variable is passed to opa.
**correct_pairs** an integer representing the number of pairwise orderings pooled across all data rows that were correctly classified by the hypothesis.

**total_pairs** an integer, the number of pair orderings contained in the data.

**group_cval** the group-level chance value.

**individual_cvals** a vector containing chance values for each data row

**n_permutations** an integer, the number of permutations of the data used to compute chance values.

**pccs_geq_observed** an integer, the number of permutations which generated PCC values at least as great as the PCC of the observed data.

**pcc_replicates** a matrix containing PCC values, one column per data row, computed from all permutations used to compute chance values.

**call** the matched call

---

**References**


---

**Examples**

```r
 dat <- data.frame(group = c("a", "b", "a", "b"),
   t1 = c(9, 4, 8, 10),
   t2 = c(8, 8, 12, 10),
   t3 = c(8, 5, 10, 11))
 dat$group <- factor(dat$group, levels = c("a", "b"))
opamod <- opa(dat[,2:4], 1:3)
opamod <- opa(dat[,2:4], 1:3, nreps = 500)
opamod <- opa(dat[,2:4], 1:3, cval_method = "exact")
opamod <- opa(dat[,2:4], 1:3, pairing_type = "adjacent")
opamod <- opa(dat[,2:4], 1:3, diff_threshold = 1)
opamod <- opa(dat[,2:4], 1:3, group = dat$group)
```

---

**pcc_plot**

*Plot individual PCCs.*

---

**Description**

Plot individual PCCs.

**Usage**

```r
 pcc_plot(m, threshold = NULL, title = TRUE, legend = TRUE)
```
Arguments

- **m**: an object of class "opafit"
- **threshold**: a boolean indicating whether to plot a threshold abline
- **title**: a boolean indicating whether to include a plot title
- **legend**: a boolean indicating whether to include a legend when n groups > 1

Value

No return value, called for side effects.

Examples

```r
dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
pcc_plot(opamod)
pcc_plot(opamod, threshold = 85)
```

```
# Description

Plots individual PCCs relative to a user-supplied PCC threshold value.

# Usage

```
pcc_threshold_plot(m, pcc_threshold = 75)
```

Arguments

- **m**: an object of class "opafit"
- **pcc_threshold**: a numeric scalar

Value

No return value, called for side effects.

Examples

```r
dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
pcc_threshold_plot(opamod)
pcc_threshold_plot(opamod, pcc_threshold = 85)
```
plot.opafit

Plots individual-level PCCs and chance-values.

### Usage

```r
## S3 method for class 'opafit'
plot(x, pcc_threshold = NULL, cval_threshold = NULL, ...)
```

### Arguments

- **x**: an object of class "opafit" produced by `opa()`
- **pcc_threshold**: a number used as the x-intercept to plot a PCC threshold abline
- **cval_threshold**: a number used as the x-intercept to plot a c-value threshold abline
- **...**: ignored

### Value

No return value, called for side effects.

### Examples

```r
dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
plot(opamod)
```

plot_hypothesis

Plot a hypothesis.

### Usage

```r
plot_hypothesis(h, title = TRUE)
```

### Arguments

- **h**: a numeric vector
- **title**: a boolean indicating whether to include a plot title
Value

No return value, called for side effects.

Examples

h <- c(1,2,3,3,3)
plot_hypothesis(h)

summary.opafit

Prints a summary of results from a fitted ordinal pattern analysis model.

Description

Prints a summary of results from a fitted ordinal pattern analysis model.

Usage

## S3 method for class 'opafit'
summary(object, ..., digits = 2L)

Arguments

object an object of class "opafit".
... ignored
digits an integer used for rounding values in the output.

Value

No return value, called for side effects.

Examples

dat <- data.frame(t1 = c(9, 4, 8, 10),
                  t2 = c(8, 8, 12, 10),
                  t3 = c(8, 5, 10, 11))
opamod <- opa(dat, 1:3)
summary(opamod)
summary(opamod, digits = 3)
Index

compare_conditions, 2
compare_conditions, 2
cval_plot, 3
group_results, 4
individual_results, 4
opa, 5
pcc_plot, 7
pcc_threshold_plot, 8
plot.opafit, 9
plot_hypothesis, 9
summary.opafit, 10