

Package ‘pdt’

July 23, 2025

Type Package

Title Permutation Distancing Test

Version 0.0.2

Author Jan Houtveen [aut, cre],
Anouk Vroegindeweij [ctb]

Maintainer Jan Houtveen <janhoutveen@gmail.com>

Depends R (>= 4.1.0)

Description Permutation (randomisation) test for single-case phase design data with two phases (e.g., pre- and post-treatment). Correction for dependency of observations is done through stepwise resampling the time series while varying the distance between observations. The required distance 0,1,2,3.. is determined based on repeated dependency testing while stepwise increasing the distance.
In preparation: Vroegindeweij et al. ``A Permutation distancing test for single-case observational AB phase design data: A Monte Carlo simulation study".

License GPL-3

Repository CRAN

NeedsCompilation no

Imports stats, graphics

Encoding UTF-8

Language en-US

RoxygenNote 7.2.3

Date/Publication 2023-01-12 18:20:02 UTC

Contents

AB_permutation_distancing_test	2
AB_permutation_test	4
AB_permutation_test_plot	5
detrend_A	7
detrend_B	7
do_remove_NA_series_above_k	8
insert_NA_and_try_to_shift	9

AB_permutation_distancing_test
AB_permutation_distancing_test

Description

Performs a randomisation test for two phases (A and B) that corrects for dependency. The correcting is done through stepwise resampling the time series while varying the distance between observations. The required distance 0,1,2,3.. is determined based on repeated dependency testing while stepwise increasing the distance. The input x and y values should be equidistant (with NA's included) using insert_NA_and_try_to_shift. The distance per cycles = k-1. k_max should be max 25 If de_A_trend=TRUE, phase A will be first de_A_trended. If de_B_trend=TRUE, phase B will be first de_B_trended. If detrend_x_position = "first" : take predicted value for first valid observation. If detrend_x_position = "center": take predicted value for center observation. If detrend_x_position = "last" : take predicted value for last valid observation. The p-value returned corresponds with the lowest Ljung-Box test (minimal) p-value found. The statistic returned correspond with median chi-square k with p_box larger than alpha_p_box_test or statistic_box smaller than max_statistic_p_box_test (i.e., not dependent).

Usage

```
AB_permutation_distancing_test(
  x,
  y,
  test_statistic = "*",
  test_statistic_function = "mean",
  reps_max = 2000,
  k_max = NULL,
  alpha_p_box_test = 0.1,
  max_statistic_p_box_test = 2.7,
  no_duplicates = FALSE,
  remove_NA_series_above_k = TRUE,
  de_A_trend = FALSE,
  detrend_A_position = "center",
  de_B_trend = FALSE,
  detrend_B_position = "center",
  show_plot = FALSE,
  show_plot_header = ""
)
```

Arguments

x	factor vector to indicate conditions or phases (e.g., "A" and "B")
y	numerical vector with the observed y-values
test_statistic	character how to compute the test statistic c("A-B", "B-A", "*") *=two-sided

test_statistic_function	character compute and compare "mean" or "median" for A and B
reps_max	numerical maximum number of permutation replications (the theoretical number= n!)
k_max	numerical maximum k value
alpha_p_box_test	numerical see above
max_statistic_p_box_test	numerical see above
no_duplicates	boolean do a permutation test without duplicates (makes it much slower)
remove_NA_series_above_k	boolean first clean the data by skipping repeated NA's
de_A_trend	boolean de-trend A first (optional)
detrend_A_position	character c("first", "center", "last"), see detrend_A
de_B_trend	boolean de-trend B first (optional)
detrend_B_position	character c("first", "center", "last"), see detrend_B
show_plot	boolean show test plot of statistical test
show_plot_header	character header of test plot

Value

List with the permutation distancing test results: de_A_trend setting in call, detrend_A_position in call, de_B_trend setting in call, detrend_B_position in call, ar1 = vector of computed ar1 values per distancing step (0,1,2, etc), p_box = vector of computed box-test p-values per distancing step, statistic_box = vector of box-test statistics per distancing step, observed_test_statistic = computed overall AB test statistic (before distancing), effect_size_overall = computed overall effect size (before distancing), p = vector of computed permutation test p-values per distancing step, effect_size = vector of computed permutation test effect-sizes per distancing step, p_fitted = vector of lm-fitted line p-values through p_box, k_max = k_max setting in call or computed based on the number of observations, k_selected_based_on_Box_test = selected k values, p_selected_based_on_Box_test = selected p-value, effect_size_selected_based_on_Box_test = selected effect-size values.

Examples

```
pdt::AB_permutation_distancing_test(
  as.factor(c(rep("A",20), rep("B",20))),
  c(rnorm(20), rnorm(20)+2),
  test_statistic="B-A",
  test_statistic_function="mean",
  reps_max=1000,
  k_max=NULL,
  alpha_p_box_test=0.1,
  max_statistic_p_box_test=2.7,
  no_duplicates=FALSE,
```

```

remove_NA_series_above_k=TRUE,
de_A_trend=FALSE,
detrend_A_position="center",
de_B_trend=FALSE,
detrend_B_position="center",
show_plot=FALSE,
show_plot_header="")

```

AB_permutation_test *AB_permutation_test*

Description

Performs a regular permutations test for two conditions or phases (A and B).

Usage

```

AB_permutation_test(
  x,
  y,
  test_statistic = "*",
  test_statistic_function = "mean",
  reps_max = 2000,
  no_duplicates = FALSE,
  show_plot = FALSE,
  show_plot_header = ""
)

```

Arguments

x	factor vector to indicate conditions or phases (e.g., "A" and "B")
y	numerical vector with the observed y-values
test_statistic	character how to compute the test statistic c("A-B", "B-A", "*") *=two-sided
test_statistic_function	character compute and compare "mean" or "median" for A and B
reps_max	numerical maximum number of permutation replications (the theoretical number= n!)
no_duplicates	boolean do a permutation test without duplicates (makes it much slower)
show_plot	boolean show test plot of statistical test
show_plot_header	character header of test plot

Value

List with the permutation test results: observed_test_statistic = computed test statistic, effect_size = computed effect size (similar to Cohen's d), random_assignments, p_randomization_AB = p value randomization AB test, one_sided_p = one-sided p-value in case of B-A or A-B.

Examples

```
pdt::AB_permutation_test(
  as.factor(c(rep("A",20), rep("B",20))),
  c(rnorm(20), rnorm(20)+2),
  test_statistic="B-A",
  test_statistic_function="mean",
  reps_max=1000,
  no_duplicates=FALSE,
  show_plot=FALSE,
  show_plot_header="")
```

```
AB_permutation_test_plot
      AB_permutation_test_plot
```

Description

Creates a permutation distancing test plot. Several plot options are available, e.g., to show both the observed and detrended lines.

Usage

```
AB_permutation_test_plot(
  x,
  x_values,
  y,
  test_statistic_function = "mean",
  de_A_trend = FALSE,
  detrend_A_position = "center",
  show_de_A_trended = FALSE,
  de_B_trend = FALSE,
  detrend_B_position = "center",
  show_de_B_trended = FALSE,
  show_plot_header = "",
  xlab = "",
  ylab = "",
  ylim = NULL,
  labels = NULL,
  line_colors = c("blue", "red", "blue", "red", "blue", "cyan"),
  show_legend = TRUE
)
```

Arguments

x	factor vector to indicate conditions or phases (e.g., "A" and "B")
x_values	numerical vector with distance (time markers) between observations

y	numerical vector with the observed y-values
test_statistic_function	character compute and compare "mean" or "median" for phase A and B
de_A_trend	boolean de-trend A (optional)
detrend_A_position	character c("first", "center", "last"), see detrend_A
show_de_A_trended	boolean show de-trend A line (optional)
de_B_trend	boolean de-trend B (optional)
detrend_B_position	character c("first", "center", "last"), see detrend_B
show_de_B_trended	boolean show de-trend B line (optional)
show_plot_header	character header
xlab	character x-axis label
ylab	character y-axis label
ylim	numerical vector of y-axis limits
labels	character vector of labels
line_colors	character vector with colors of the succeeding lines c("blue", "red", "blue", "red", "blue", "cyan"),
show_legend	boolean show legend

Value

NULL.

Examples

```
pdt::AB_permutation_test_plot(
  as.factor(c(rep("A", 20), rep("B", 20))),
  1:40,
  c(rnorm(20), rnorm(20)+2),
  test_statistic_function="mean",
  de_A_trend=TRUE,
  detrend_A_position="center",
  show_de_A_trended=TRUE,
  de_B_trend=TRUE,
  detrend_B_position="center",
  show_de_B_trended=TRUE,
  show_plot_header="",
  xlab="",
  ylab="",
  ylim=NULL,
  labels=NULL,
  line_colors=c("blue", "red", "blue", "red", "blue", "cyan"),
  show_legend=TRUE)
```

detrend_A	<i>detrend_A</i>
-----------	------------------

Description

Detrends the phase A part of time series y. detrend A is optional and not validated. The mean of the detrended signal will be set to the predicted value based on detrend_A_position: detrend_A_position = "first" : take predicted value for first valid observation detrend_A_position = "center" : take predicted value for center observation detrend_A_position = "last" : take predicted value for last valid observation.

Usage

```
detrend_A(x, x_values, y, detrend_A_position = "center")
```

Arguments

- x factor vector to indicate conditions or phases (e.g., "A" and "B")
- x_values numerical vector with distance (time markers) between observations
- y numeric vector with the observed y-values
- detrend_A_position character to indicate the mean

Value

List with the trend and the detrended y-values: x_values_A_trend = vector with distance (time markers) between A-detrended signal, y_A_trend = vector with computed A-trend, y_detrended = vector with computed A-detrended y values.

Examples

```
pdt::detrend_A(as.factor(c(rep("A",20), rep("B",20))), 1:40,
  c(rnorm(20), rnorm(20)+2), detrend_A_position="center")
```

detrend_B	<i>detrend_B</i>
-----------	------------------

Description

Detrends the phase B part of time series y. detrend B is optional and not validated. The mean of the detrended signal will be set to the predicted value based on detrend_B_position: detrend_B_position = "first" : take predicted value for first valid observation detrend_B_position = "center" : take predicted value for center observation detrend_B_position = "last" : take predicted value for last valid observation.

Usage

```
detrend_B(x, x_values, y, detrend_B_position = "center")
```

Arguments

x factor vector to indicate conditions or phases (e.g., "A" and "B")

x_values numerical vector with distance (time markers) between observations

y numeric vector with the observed y-values

detrend_B_position character to indicate the mean

Value

List with the trend and the detrended y values: **x_values_B_trend** = vector with distance (time markers) between B-detrended signal, **y_B_trend** = vector with computed B-trend, **y_detrended** = vector with computed B-detrended y values.

Examples

```
pdt::detrend_B(as.factor(c(rep("A",20), rep("B",20))), 1:40,
  c(rnorm(20), rnorm(20)+2), detrend_B_position="center")
```

```
do_remove_NA_series_above_k
      do_remove_NA_series_above_k
```

Description

Remove series of more than k succeeding NA's in x, y, and x_values. This function is recommended before performing a permutation distancing test.

Usage

```
do_remove_NA_series_above_k(x, y, k, x_values = NULL)
```

Arguments

x factor vector to indicate conditions or phases (e.g., "A" and "B")

y numeric vector with the observed y-values

k maximum allowed number of NA's

x_values numerical vector with distance (time markers) between observations

Value

List with the modified x, y, x_values: x = factor vector with conditions (e.g., "A" and "B"). y = vector with observed values. x_values = vector with distance (time markers) between observations x,y.

Examples

```
pdt::do_remove_NA_series_above_k(as.factor(c("A", "A", "A", "B", "B", "B")),
  c(1.1, NA, NA, 7.1, 8.3, 9.8), 1, c(1, 2, 4, 5, 6, 8))
```

```
insert_NA_and_try_to_shift
      insert_NA_and_try_to_shift
```

Description

Makes the input time series equidistant. This is recommended before performing a permutation distancing test. This function first inserts NA's for missing x_values, then it tries to shift double value to previous or next NA's, finally it aggregates the remaining identical x_values.

Usage

```
insert_NA_and_try_to_shift(x, x_values, y)
```

Arguments

x	factor vector to indicate conditions or phases (e.g., "A" and "B")
x_values	numerical vector with distance (time markers) between observations
y	numeric vector with the observed y-values

Value

List with the modified x, x_values, y: x = factor vector with conditions or phases (e.g., "A" and "B"). x_values = (optional) vector with distance (time markers) between observations. y = vector with observed values.

Examples

```
pdt::insert_NA_and_try_to_shift(as.factor(c("A", "A", "A", "B", "B", "B")),
  c(1, 2, 4, 5, 6, 8), c(1.1, 3.2, 5.3, 7.1, 8.3, 9.8))
```

Index

AB_permutation_distancing_test, [2](#)

AB_permutation_test, [4](#)

AB_permutation_test_plot, [5](#)

detrend_A, [7](#)

detrend_B, [7](#)

do_remove_NA_series_above_k, [8](#)

insert_NA_and_try_to_shift, [9](#)