Package ‘tiefightR’

July 13, 2020

Title Calculate Preference Positions

Version 0.0.0.9000

Year 2020

Maintainer Steven R. Talbot <`talbot.steven@mh-hannover.de`>

Description The tiefightR package is for preference test evaluation and simulation. Its goal is to rank commodity positions
obtained from preference test experiments. Special attention goes into the analysis of intransitivities in the data and
resulting tie evaluation after data binarization.

License GPL-3

Encoding UTF-8

LazyData true

Suggests knitr,
rmarkdown

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.0

URL `https://github.com/mytalbot/tiefightR`

BugReports `https://github.com/mytalbot/tiefightR/issues`

Imports magrittr,
tibble,
dplyr,
reshape2,
prefmod,
gnm,
ggplot2,
ggpubr,
parallel,
doParallel,
foreach,
viridis,
ggsci,
Rmisc,
doRNG

Depends R (>= 2.10)

R topics documented:

`human` *tiefightR - human data*

Description

Two sets of seven pictures each were taken for preference ranking. The subjects for the first set were 32 (16 male, 16 female) persons aged between 19 and 46 years (average: 25.5 years). The second set of pictures was ranked by 63 persons (33 male, 29female, 1 other) with an average age of 29.1 years (ranging from 19-70). 73 of the probands were recruited at the campus of the University of Göttingen, Germany and conducted the test in a controlled laboratory environment. The other 22 persons conducted the internet-based test somewhere else. The first set of pictures was tested a second time during the course of a graduate spring school by 16 participants (age range from 26 to 40 years, mean 31.1 years; 12 female, 4 male).

Usage

```
data(human)
```

Format

A 2331 x 7 data frame with binary response data (pref_img1).

Examples

```
data(human)
head(human)
```

`mouse` *tiefightR - mouse data*

Description

Eleven female C57BL/6J mice were tested. The mice were purchased from Charles River Sulzfeld, Germany and arrived at the institute at the age of 21 days. All mice were implanted with an RFID chip (Planet ID, FDX-B transponder according to ISO 11784/85) holding a unique ID for individual differentiation at an age of 33 or 56 days, a procedure performed under anesthesia (Isofluran). Two hours before the transplantation, all mice were given an analgesic (Meloxicam). At the start of the first run, the mice were seven months old and weighed between 25.0 and 29.0 g. At the start of the second run, the mice were 14 month old and weighed between 25.5 and 37.0 g.

Usage

```
data(mouse)
```

Format

A 880 x 10 data frame with continouse response data (numOF_visits_with_Licks).

Examples

```
data(mouse)
head(mouse)
```

`rhesus` *tiefightR - rhesus data*

Description

Six male rhesus macaques (Macaca mulatta) at an age range of 6-19 years (mean = 13.09) living in same-sexual groups of 2 - 4 at the Cognitive Neuroscience Laboratory, German Primate Centre, took part in this study. Monkeys were housed in indoor rooms equipped with a multitude of toys and wooden structures as well as natural and artificial light. The space per monkey exceeded all applicable German and European regulations (Berger et al. 2018). Indoor rooms were temperature-controlled and connected by a tunnel with rooms at ambient outdoor temperature and lighting, but protected from precipitation. On test days, monkeys had free access to water for at least 4 hours (typically much more: for definitions of access to water see (Pfefferle et al. 2018)) and received monkey chow ad libitum. On non-test days, the monkeys had free access to water and received monkey chow ad libitum, supplemented with dried fruits, fresh fruits and vegetables. The health of the monkeys was monitored daily by the animal care staff, veterinarians, and the laboratory researchers who were all highly experienced with these animals.

Usage

```
data(rhesus)
```

Format

A 240 x 7 data frame with continouse response data (amountDrank).

Examples

```
data(rhesus)
head(rhesus)
```

---

tie_binarize *Binarize continous data*

Description

Uses `tie_import` non-binary outcome data as input. The outcome variable is randomized for ties in the continous data.

Usage

```
tie_binarize(
  xdata = NULL,
  SV = NULL,
  RF = NULL,
  CF = NULL,
  id = NULL,
  RV = NULL,
  datalabel = "binarized",
  compiled_studies = NULL,
  setseed = TRUE,
  prefLimit = 50,
  sidevar = "side",
  refval = "refValue",
  oval = "otherValue",
  aid = "animalID",
  fldrnk = "fluid_drunk"
)
```

Arguments

`xdata`	imported (binarized) data frame
`SV`	name of the side variable
`RF`	name of the reference fluid variable
`CF`	name of the combination fluid variable
`id`	subject IDs
`RV`	name of the response variable
`datalabel`	universal study label for the binarized data
`compiled_studies`	label of the compiled sub study (used for filtering)
`setseed`	TRUE/FALSE for seeding
`prefLimit`	preference limit for binarization threshold
`sidevar`	the name of the standardized side variable in the org data
`refval`	the name of the standardized reference variable in the org data
`oval`	the name of the standardized combination variable in the org data
`aid`	the name of the standardized animal id column in the org data
`fldrnk`	the name of the standardizedresponse variable in the org data

Value

binarized data in the default format (colum headers)

`tie_cicheck` *Commodity position and confidence interval check for a discrete number of randomizations*

Description

The `tie_cicheck` is a wrapper for checking the confidence intervals for data with ties. The function calculates the worth values for a specific number of randomizations and reports the confidence intervals for the commodity means.

Usage

```
tie_cicheck(
  data = tiefightR::mouse,
  R = NULL,
  ciLvl = 0.95,
  seed = TRUE,
  SV = NULL,
  RF = NULL,
  CF = NULL,
  id = NULL,
  RV = NULL,
  ord = NULL,
  prefLimit = 50,
  compstudy = NULL,
  default = NULL,
  showplot = TRUE,
  showstats = FALSE,
  ylim = c(0.1, 0.35)
)
```

Arguments

`R`	number of maximum randomization steps
`ciLvl`	Level of confidence (default: 0.95)
`seed`	TRUE/FALSE for constant seeding
`SV`	name of the side variable
`RF`	name of the reference fluid variable
`CF`	name of the combination fluid variable
`id`	subject IDs
`RV`	name of the response variable
`ord`	item category order
`prefLimit`	preference limit for binarization threshold
`compstudy`	label of the compiled sub study (used for filtering)

`default`	default item in worth value estimation (usually the lowest worth value)
`showplot`	show the errorplot with confidence intervals
`showstats`	calculate ANOVA1 and Tukey's test for the commodities
`dat`	imported raw data (should be binary, if not, will be binarized automatically)

Value

Exports random binarize response for distance cutoff selection

`tie_cores` *CPU Core Detection*

Description

The `tie_cores` detects the number of available CPUs for parallel computing. Don't overdo it!

Usage

```
tie_cores()
```

Value

No of CPUs on your machine

`tie_cutoff` *Cutoff determination for increasing number of randomizations*

Description

The `tie_cutoff` function calculates the mean Euclidean distance between commodity worth values. This becomes relevant when ties are present in the data. Depending on how ties are resolved (see prefLimit argument in the function), the items' position will change a lot. Since their relative positions are a function of the number of ties, more randomizations will stabilize their means and thus commodity positions. Increasing the number of randomizations usually leads not only to a stabilized mean but also to smaller confidence intervals. By defining a relative cutoff (e.g., 5 or 10%) for the range of the CIs regarding the maximum range in the data, a cutoff for the number of randomizations can be found.

Usage

```
tie_cutoff(
  data = tiefightR::mouse,
  R = 50,
  ciLvl = 0.95,
  cutoff = 0.1,
  cpus = 2,
  SV = NULL,
  RF = NULL,
  CF = NULL,
  id = NULL,
  RV = NULL,
  ord = NULL,
  prefLimit = 50,
  compstudy = NULL,
  default = NULL,
  showplot = FALSE,
  showCutoff = FALSE
)
```

Arguments

`R`	number of maximum randomization steps
`cutoff`	Percent cutoff level (default: 0.10) - means CI range < than cutoff value
`cpus`	No. of used local CPUs for parallel computing (you should have more than 2)
`SV`	name of the side variable
`RF`	name of the reference fluid variable
`CF`	name of the combination fluid variable
`id`	subject IDs
`RV`	name of the response variable
`ord`	item category order
`prefLimit`	preference limit for binarization threshold
`compstudy`	label of the compiled sub study (used for filtering)
`default`	default item in worth value estimation (usually the lowest worth value)
`showplot`	show the plot for randomization cutoff determination
`showCutoff`	show vertical line of the cutoff
`dat`	imported raw data (should be binary, if not, will be binarized automatically)
`standardize`	standardize on the maximum CI value?

Value

Exports cutoff value and plots

`tie_import` *Import Function*

Description

The `tie_import` function loads the raw data as a data frame. This function can be skipped when one of the three internal data sets (human, mouse, rhesus) are used. For a user who wants to import own data this function is a good start. Make sure that the imported data has at least the following information: data, subset (even if none is there), SV (side variable, left/right), RF (reference fluid/item), CF (combination fluid), id (animal id), RV (response variable)).

Usage

```
tie_import(path = NULL, valenceset = NULL)
```

Arguments

`path`	path to the raw data
`valenceset`	subset filtering argument

Value

data.frame with the filtered subset

`tie_intrans` *Calculate Intransitivity*

Description

The `tie_intrans` function loads the raw data

Usage

```
tie_intrans(
  mydata = NULL,
  idcolumn = "ID",
  I1 = "img1",
  I2 = "img2",
  response = "pref_img1"
)
```

Arguments

`mydata`	input data frame
`idcolumn`	name of the ID column in the input data
`I1`	name of the test image column in the input data
`I2`	name of the other (tested) items column in the input data
`response`	name of the response variable

Value

intranscount intransitivity counts

tie_rwalk	*Tie random walk function*

Description

The tie_rwalk function prepares binary and continuous data for tiefightR analysis.

Usage

```
tie_rwalk(
  dat = NULL,
  SV = NULL,
  RF = NULL,
  CF = NULL,
  id = NULL,
  RV = NULL,
  ord = NULL,
  prefLimit = 50,
  setseed = FALSE,
  compstudy = NULL,
  default = NULL,
  R = NULL
)
```

Arguments

dat	imported (binarized) data frame
SV	name of the side variable
RF	name of the reference fluid variable
CF	name of the combination fluid variable
id	subject IDs
RV	name of the response variable
ord	item category order
prefLimit	preference limit for binarization threshold
setseed	BOOLEAN; set a random seed TRUE/FALSE?
compstudy	label of the compiled sub study (used for filtering)
default	default item in worth value estimation (usually the lowest worth value)
R	number of randomizations

Value

Exports random binarize response for distance cutoff selection

tie_sim *Tie Simulation Function*

Description

The `tie_sim` function starts a simulation of item pairings and introduces random pairs for the remaining combinations. In parallel, the intransitivity of triple pairings can be calculated to estimate their position quality. Good transitiviy and massed localization in a position will improve the Likelihood of a good fit for the item.

Usage

```
tie_sim(
  xdata = NULL,
  R = 2,
  SV = "side_img1",
  RF = "img1",
  CF = "img2",
  id = "ID",
  RV = "pref_img1",
  intrans = TRUE,
  compstudy = "LagreValenceRange_SpringSchool",
  default = "War",
  cpus = 2,
  ord = NULL,
  v1 = NULL
)
```

Arguments

`xdata`	imported (binarized) data frame
`R`	No. of randomization steps
`SV`	name of the side variable
`RF`	name of the reference fluid variable
`CF`	name of the combination fluid variable
`id`	subject IDs
`RV`	name of the response variable
`intrans`	calculate intransitivities (calculation intense!)
`compstudy`	label of the compiled sub study (used for filtering)
`default`	default item in worth value estimation (usually the lowest worth value)
`cpus`	No. of used local CPUs for parallel computing (you should have more than 2)
`ord`	item category order
`v1`	testing variable (can be one item from the item list)

Value

data.frame with the simulation results

`tie_simrep` *Simulation report*

Description

The `tie_simrep` function prepares a frequency table and plots for analysing the simulation output. Can also be used for saving the output to file when a path is provided.

Usage

```
tie_simrep(res = NULL, v1 = NULL, path = NULL)
```

Arguments

`res`	result or output from the simulation
`v1`	test variable
`path`	path to where the report shall be stored (inluding plots)

Value

A frequency table for item positions during the simulation; Tukey's HSD Test for positions; Position Bubble Plot

`tie_test` *Test Function*

Description

The `tie_test` function can be used for individual item testing.

Usage

```
tie_test(
  xdata = NULL,
  R = NULL,
  intrans = TRUE,
  compstudy = NULL,
  default = NULL,
  ord = NULL,
  seed = TRUE,
  testme = NULL,
  against = NULL
)
```

Arguments

`xdata`	imported (binarized) data frame
`R`	number of randomizations
`intrans`	calculate intransitivities (calculation intense!)
`compstudy`	label of the compiled sub study (used for filtering)
`default`	default item in worth value estimation (usually the lowest worth value)
`ord`	item category order
`seed`	BOOLEAN; set a random seed TRUE/FALSE?
`testme`	test this variable against...
`against`	any other single variable from the commodity list (or combinations); vectorize if necessary c("item1","item2")

Value

A frequency table for item positions during the simulation; Position Bubble Plot

---

`tie_worth` *Main preference function*

Description

The `tie_import` function prepares binary and continuous data form import into the tiefightR analysis. The user has to specicy the names of the input columns (if they deviate from the default values in the function argument list). The function randomizes the response variable for any non chosen item test combination and reports the worth values.

Usage

```
tie_worth(
  xdata = NULL,
  esti = "worth",
  SV = "side_img1",
  RF = "img1",
  CF = "img2",
  id = "ID",
  RV = "pref_img1",
  default = NULL,
  showplot = FALSE,
  intrans = FALSE,
  compstudy = NULL,
  ordn = NULL,
  r1 = NULL,
  r2 = NULL,
  ymin = 0,
  ymax = 0.5
)
```

Arguments

`xdata`	imported (binarized) data frame
`esti`	worth estimator (default, "worth", alt: "estimator")
`SV`	name of the side variable
`RF`	name of the reference fluid variable
`CF`	name of the combination fluid variable
`id`	subject IDs
`RV`	name of the response variable
`default`	default item in worth value estimation (usually the lowest worth value)
`showplot`	show worth plot TRUE/FALSE
`intrans`	calculate intransitivities (calculation intense!)
`compstudy`	label of the compiled sub study (used for filtering)
`ordn`	item category order
`r1`	label of the test item (e.g., "Lake")
`r2`	label(s) of the remaining item(s)
`ymin`	minimum y-scale of the worth plot
`ymax`	maximum y-scale of the worth plot

Value

Exports the results of the worth value calculation, including the GNM analysis.

Index