Package ‘r4lineups’

October 14, 2022

Title  Statistical Inference on Lineup Fairness
Version 0.1.1
Maintainer Colin Tredoux <colin.tredoux@uct.ac.za>

Description  Since the early 1970s eyewitness testimony researchers have recognised the importance of estimating properties such as lineup bias (is the lineup biased against the suspect, leading to a rate of choosing higher than one would expect by chance?), and lineup size (how many reasonable choices are in fact available to the witness? A lineup is supposed to consist of a suspect and a number of additional members, or foils, whom a poor-quality witness might mistake for the perpetrator). Lineup measures are descriptive, in the first instance, but since the earliest articles in the literature researchers have recognised the importance of reasoning inferentially about them. This package contains functions to compute various properties of laboratory or police lineups, and is intended for use by researchers in forensic psychology and/or eyewitness testimony research. Among others, the r4lineups package includes functions for calculating lineup proportion, functional size, various estimates of effective size, diagnosticity ratio, homogeneity of the diagnosticity ratio, ROC curves for confidence x accuracy data and the degree of similarity of faces in a lineup.

Depends  R (>= 3.4.0)
Imports  boot, dplyr, ggplot2, ggrepel, here, magick, magrittr, pROC, psych, purrr, stats, utils
License  CC0
Encoding  UTF-8
LazyData  true
RoxygenNote  6.0.1.9000
Suggests  knitr, rmarkdown, pander
VignetteBuilder  knitr
NeedsCompilation  no
Author  Colin Tredoux [aut, cre],
        Tamsyn Naylor [aut]
Repository  CRAN
Date/Publication  2018-07-18 13:20:02 UTC
R topics documented:

allfoilbias .................................................. 3
allfoil_cihigh ................................................. 4
allprop ......................................................... 4
chi_diag ....................................................... 5
compare_eff_sizes.boot .................................. 6
datacheck1 ..................................................... 7
datacheck2 ..................................................... 8
datacheck3 ..................................................... 9
diag_param .................................................... 9
diag_ratio_T ................................................ 11
diag_ratio_W ................................................ 12
d_bar .......................................................... 13
d_weights ...................................................... 15
effsize_compare ........................................... 16
eff_size_per_foils ........................................ 18
esize_m ......................................................... 18
esize_m_boot .................................................. 20
esize_T ........................................................ 21
esize_T_boot ................................................. 22
esize_T_ci_n .................................................. 23
face_sim ....................................................... 24
func_size ..................................................... 24
func_size.boot ............................................... 25
func_size_report .......................................... 26
gen_boot_propci ........................................... 27
gen_boot_propmean_se .................................. 28
gen_boot_samples ......................................... 29
gen_boot_samples_list .................................. 30
gen_esize_m .................................................. 30
gen_esize_m_ci .............................................. 31
gen_lineup_prop ........................................... 32
gen_linevec .................................................. 33
homog_diag .................................................. 34
homog_diag_boot ........................................... 35
i_esize_T ...................................................... 36
line73 ......................................................... 37
lineup_boot_allprop ..................................... 38
lineup_prop_boot .......................................... 39
lineup_prop_tab ............................................ 40
lineup_prop_vec ............................................ 40
ln_diag_ratio ............................................... 41
makevec_prop ............................................... 43
make_roc ...................................................... 43
make_rocdata ................................................ 44
mickwick ..................................................... 45
mockdata ..................................................... 45
**Description**

Function to compute bias for each lineup member (assuming foil is suspect, from Malpass, 1981)

**Usage**

```r
allfoilbias(lineup_table, target_pos, k)
```

**Arguments**

- `lineup_table`: A table of lineup choices
- `target_pos`: A scalar, representing target position in lineup. Must be declared by user
- `k`: Nominal size (i.e., total number of members in lineup). Must be specified by user (scalar).

**References**


**Examples**

```r
lineup_vec <- round(runif(100, 1, 6))
lineup_table <- table(lineup_vec)
x <- allfoilbias(lineup_table, 5, 6)
```
**allfoil_cihigh**  
*Confidence Intervals for Proportion*

**Description**
Function to compute ci high for each foil in a lineup

**Usage**
```r
allfoil_cihigh(linetabprops, sumlineup)
```

**Arguments**
- `linetabprops`: A dataframe of bootstrapped lineup proportions
- `sumlineup`: Number of members in a lineup

**References**

---

**allprop**  
*Lineup proportion for all lineup members*

**Description**
Computes lineup proportion for each member in a lineup

**Usage**
```r
allprop(lineup_vec, k)
```

**Arguments**
- `lineup_vec`: A numeric vector of lineup choices
- `k`: Number of members in lineup. Must be specified by user (scalar).

**Value**
Returns a vector containing lineup proportion for each lineup member

**References**
Examples

# Data:
lineup_vec <- round(runif(100, 1, 6))

# Call:
x <- allprop(lineup_vec, k = 6)

chi_diag

Chi-squared estimate of homogeneity of diagnosticity ratio

Description

Function for getting chi-squared value for homogeneity of diagnosticity ratios

Usage

chi_diag(df)

Arguments

df A dataframe containing: ln(d), variance of ln(d), d weights

Details

- To compute linedf, use the diag_param helper function.
- To compute df, apply ln_diag_ratio, var_lnd & d_weights functions to linedf, then bind results into one dataframe (see Examples)
- The order in which the estimates are bound together (i.e., their position in the dataframe) is important, and should always be as follows: row 1: var, row 2: lnd, row 3: wi.

Value

Chi squared estimate of homogeneity of diagnosticity ratios for k independent lineups

References


Examples

# Target present data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_pres_list <- list(A, B, C)
rm(A, B, C)

# Target absent data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_abs_list <- list(A, B, C)
rm(A, B, C)

# Pos list
lineup1_pos <- c(1, 2, 3, 4, 5, 6)
lineup2_pos <- c(1, 2, 3, 4, 5)
lineup3_pos <- c(1, 2, 3, 4)
pos_list <- list(lineup1_pos, lineup2_pos, lineup3_pos)
rm(lineup1_pos, lineup2_pos, lineup3_pos)

# Nominal size:
k <- c(6, 5, 4)

# Use diag param helper function to get data (n11, n21, n12, n22):
linedf <- diag_param(lineup_pres_list, lineup_abs_list, pos_list, k)

# Get ln(d), variance of ln(d) & d weights:
ratio <- ln_diag_ratio(linedf)
var <- var_lnd(linedf)
wi <- d_weights(linedf)

# Bind estimates into one df of 3 rows & x observations
# (see Details above)
df <- t(cbind(ratio, var, wi))

# Call:
chi_diag(df)

---

**compare_eff_sizes.boot**

*Comparing Effective Size: Base function for bootstrapping*

---

**Description**

A base function for bootstrapping a dataframe of choices for 2 independent lineups
**Usage**

```r
compare_eff_sizes.boot(linedf, d)
```

**Arguments**

- `linedf`: A dataframe of lineup data. Must consist of 2 columns, each containing data for 2 independent lineups.
- `d`: Indices for bootstrap sample. Argument used by boot function to select samples for bootstrapping.

**Details**

The approach here is to compute the effective size of each lineup separately, and to take the difference between them. This is then bootstrapped, and if the bootstrap does not contain 0, we conclude the effective size estimates are different at \( p = \alpha \).

**References**


---

**datacheck1**

**Helper function**

**Description**

Checks that number of lineup choices contained in a vector is accurate.

**Usage**

```r
datacheck1(lineup_vec, k)
```
Arguments

- **lineup_vec** A numeric vector of lineup choices
- **k** Number of members in lineup

Details

This function ensures that a non-selected lineup member is not accidentally omitted from the dataframe due to lack of selection by all mock witnesses. It functions as a check that the total number of lineup members is accurate.

datacheck2  

Helper function

Description

Checks that number of lineup choices contained in a vector is accurate

Usage

datacheck2(lineup_pres, lineup_abs, k1, k2)

Arguments

- **lineup_pres** A numeric vector of lineup choices for a lineup in which the target was present
- **lineup_abs** A numeric vector of lineup choices for a lineup in which the target was absent
- **k1** Number of targets in TP lineup
- **k2** Number of targets in TA lineup

Details

This function ensures that a non-selected lineup member is not accidentally omitted from the dataframe due to lack of selection by all mock witnesses. It functions as a check that the total number of lineup members is accurate.
**datacheck3**

*Helper function*

**Description**
Checks that number of lineup choices contained in a data table is accurate

**Usage**

datacheck3(lineup_table, k)

**Arguments**

- **lineup_table**: A table of lineup choices
- **k**: Nominal size (i.e., total number of members in lineup)

**Details**
This function ensures that a non-selected lineup member is not accidentally omitted from the data due to lack of selection by all mock witnesses. It functions as a check that the total number of lineup members is accurate.

**diag_param**

*Parameters for diagnosticity ratio*

**Description**
This function calculates the parameters needed to calculate the diagnosticity ratio for several lineup pairs.

**Usage**

diag_param(lineup_pres_list, lineup_abs_list, pos_list, k)

**Arguments**

- **lineup_pres_list**: A list containing k vectors of lineup choices for k lineups, in which the target was present
- **lineup_abs_list**: A list containing k vectors of lineup choices for k lineups, in which the target was absent
- **pos_list**: A list containing k numeric vectors indexing lineup member positions for each lineup pair
- **k**: A vector indexing number of members in each lineup pair. Must be specified by user (scalar).
Details

- Lineup pairs consist of one lineup in which the target was present (TP) and one lineup in which the target was absent (TA).
- Each lineup pair must occupy corresponding positions in the TA and TP lists.
  
  Example:
  For a lineup pair A that consists of (1) TP lineup and (2) TA lineup: A(1) is the first vector in the TP list A(2) is the first vector in the TP list
- The order in which nominal size for each lineup pair is listed must also correspond with the positions of each respective lineup in the lineup lists (i.e., if lineup 1 has k = 6, then the first element of vector ’k’ = 6)
- Data must be in a list format. This allows the function to compare lineups in which the number of choices and number of lineup members differs.
- The following warning will appear if vectors comprising lineup lists are of different lengths: longer object length is not a multiple of shorter object length. This does not affect the accuracy of the function and can be ignored.

Value

Returns a dataframe containing:

- \(n11\): Number of mock witnesses who identified the suspect in the target present condition
- \(n21\): Number of mock witnesses who did not identify the suspect in the target present condition
- \(n12\): Number of mock witnesses who identified the suspect in the target absent condition
- \(n13\): Number of mock witnesses who did not identify the suspect in the target absent condition

References


Examples

```r
# Target present data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
```
C <- round(runif(20,1,4))
lineup_pres_list <- list(A, B, C)
rm(A, B, C)

#lineup absence
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_abs_list <- list(A, B, C)
rm(A, B, C)

#Pos list
lineup1_pos <- c(1, 2, 3, 4, 5, 6)
lineup2_pos <- c(1, 2, 3, 4, 5)
lineup3_pos <- c(1, 2, 3, 4)
pos_list <- list(lineup1_pos, lineup2_pos, lineup3_pos)
rm(lineup1_pos, lineup2_pos, lineup3_pos)

#Nominal size:
k <- c(6, 5, 4)

#Call:
linedf <- diag_param(lineup_pres_list, lineup_abs_list, pos_list, k)

---

diag_ratio_T

**Diagnosticy Ratio (Tredoux, 1998)**

**Description**

Computes Wells’s adjusted diagnosticity ratio for one lineup pair (see: Tredoux, 1998)

**Usage**

diag_ratio_T(lineup_pres, lineup_abs, pos_pres, pos_abs, k1, k2)

**Arguments**

- **lineup_pres**: A numeric vector of lineup choices for a lineup in which the target was present.
- **lineup_abs**: A numeric vector of lineup choices for a lineup in which the target was absent.
- **pos_pres**: A scalar, representing target position in TP lineup. Must be declared by user.
- **pos_abs**: A scalar, representing target position in TA lineup. Must be declared by user.
- **k1**: Number of targets in TP lineup. Must be specified by user (scalar).
- **k2**: Number of targets in TA lineup. Must be specified by user (scalar).
References


Examples

#Data:
lineup_pres <- round(runif(100, 1, 6))
lineup_abs <- round(runif(70, 1, 5))
pos_pres <- 3
pos_abs <- 5

#Call:
diag_ratio_T(lineup_pres, lineup_abs, pos_pres, pos_abs, 6, 5)
diag_ratio_T(lineup_pres, lineup_abs, 3, 5, 6, 5)

---

**diag_ratio_W**

*Diagnosticity Ratio (Wells & Lindsay, 1980; Wells & Turtle, 1986)*

**Description**

Computes Wells’ diagnosticity ratio for one lineup pair

**Usage**

`diag_ratio_W(lineup_pres, lineup_abs, pos_pres, pos_abs, k1, k2)`

**Arguments**

- `lineup_pres`: A numeric vector of lineup choices for a lineup in which the target was present
- `lineup_abs`: A numeric vector of lineup choices for a lineup in which the target was absent
- `pos_pres`: A scalar, representing target position in TP lineup. Must be declared by user
- `pos_abs`: A scalar, representing target position in TA lineup. Must be declared by user
- `k1`: Number of targets in TP lineup. Must be specified by user (scalar).
- `k2`: Number of targets in TA lineup. Must be specified by user (scalar).
d_bar

References


Examples

```r
# Data:
lineup_pres <- round(runif(100, 1, 6))
lineup_abs <- round(runif(70, 1, 5))
pos_pres <- 3
pos_abs <- 5

# Call:
diag_ratio_W(lineup_pres, lineup_abs, pos_pres, pos_abs, 6, 5)
diag_ratio_W(lineup_pres, lineup_abs, 3, 5, 6, 5)
```

---

**d_bar**  
Mean diagnosticity ratio for k lineup pairs

**Description**

Function for computing pooled estimator from a set of k diagnosticity ratios

**Usage**

d_bar(df)

**Arguments**

df  
A dataframe containing rows: ln(d), variance of ln(d), d weights

**Details**

The order in which the estimates are bound together (i.e., their position in the dataframe) is important, and should always be as follows:

- row 1: var
- row 2: lnd
- row 3: wi

**Value**

Mean diagnosticity ratio for k independent lineups
References


Examples

```r
# Target present data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_pres_list <- list(A, B, C)
rm(A, B, C)

# Target absent data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_abs_list <- list(A, B, C)
rm(A, B, C)

# Pos list
lineup1_pos <- c(1, 2, 3, 4, 5, 6)
lineup2_pos <- c(1, 2, 3, 4, 5)
lineup3_pos <- c(1, 2, 3, 4)
pos_list <- list(lineup1_pos, lineup2_pos, lineup3_pos)
rm(lineup1_pos, lineup2_pos, lineup3_pos)

# Nominal size:
k <- c(6, 5, 4)

# Use diag param helper function to get data (n11, n21, n12, n22):
linedf <- diag_param(lineup_pres_list, lineup_abs_list, pos_list, k)

# Compute ln(d):
ratio <- ln_diag_ratio(linedf)

# Compute variance of ln(d):
var <- var_lnd(linedf)

# Compute weights for pooled estimator:
```
wi <- d_weights(linedf)

# Bind ln(d), variance of ln(d) and weights into one df (of 3 rows & x observations)
# (see Details above):
df <- t(cbind(ratio, var, wi))

# Call:
d_bar(df)

d_weights  
Diagnosticity ratio weights

Description
Function to compute weights of each diagnosticity ratio for k lineup pairs

Usage
d_weights(linedf)

Arguments
linedf  
A dataframe of parameters for computing diagnosticity ratio

Details
In order to obtain a pooled estimate of a set of diagnosticity ratios, we use a weight for each ratio that is equal to the inverse of its variance.

To get linedf, use the diag_param helper function
diag_param returns a dataframe containing the following:

- \(n11\): Number of mock witnesses who identified the suspect in the target present condition
- \(n21\): Number of mock witnesses who did not identify the suspect in the target present condition
- \(n12\): Number of mock witnesses who identified the suspect in the target absent condition
- \(n13\): Number of mock witnesses who did not identify the suspect in the target absent condition

Value
A dataframe with one column containing weights for each pair of lineups for which diagnosticity ratio is being calculated.

References
Examples

# Target present data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_pres_list <- list(A, B, C)
rm(A, B, C)

# Target absent data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_abs_list <- list(A, B, C)
rm(A, B, C)

# Pos list
lineup1_pos <- c(1, 2, 3, 4, 5, 6)
lineup2_pos <- c(1, 2, 3, 4, 5)
lineup3_pos <- c(1, 2, 3, 4)
pos_list <- list(lineup1_pos, lineup2_pos, lineup3_pos)
rm(lineup1_pos, lineup2_pos, lineup3_pos)

# Nominal size:
k <- c(6, 5, 4)

# Use diag param helper function to get data (n11, n21, n12, n22):
linedf <- diag_param(lineup_pres_list, lineup_abs_list, pos_list, k)

# Call:
wi <- d_weights(linedf)

effsize_compare 

Master Function: Comparing Effective Size

Description

Function for comparing effective size of two independent lineups (Tredoux, 1998)

Usage

effsize_compare(linedf)

Arguments

linedf A dataframe of lineup data. Contains 2 columns, each of which hold data for 2 independent lineups
Details

This function is a master function, calling other functions it needs, and reporting results in some detail.

Value

Effective size, significance level, and confidence intervals (95 normal theory, percentile & bias-corrected)

References


See Also

boot: https://cran.r-project.org/web/packages/boot/boot.pdf

Examples

#Data:

```r
lineup_vec1 <- round(runif(100, 1, 6))
lineup_vec2 <- round(runif(100, 1, 6))
linedf <- as.data.frame(cbind(lineup_vec1, lineup_vec2))
```

#Call:

```r
x <- effsize_compare(linedf)
```
**eff_size_per_foils**    
*Effective Size per Foils*

**Description**
Function for computing effective size (Malpass, 1981) counting foils who fall within the CI for chance guessing

**Usage**
```
eff_size_per_foils(lineup_vec, target_pos, k, conf = 0.95)
```

**Arguments**
- `lineup_vec`: A numeric vector of lineup choices
- `target_pos`: A numeric vector indexing all lineup members
- `k`: Number of members in lineup. Must be specified by user (scalar).
- `conf`: Desired level of alpha. Defaults to 0.95. May be specified by user (scalar).

**References**

**Examples**
```
#Data:
lineup_vec <- round(runif(100, 1, 6))
target_pos <- c(1, 2, 3, 4, 5, 6)

#Call:
eff_size_per_foils(lineup_vec, target_pos, 6)
eff_size_per_foils(lineup_vec, target_pos, conf = 0.95, 6)
```

---

**esize_m**    
*Effective Size*

**Description**
Function for computing Effective Size

**Usage**
```
esize_m(lineup_table, k, both = FALSE)
```
Arguments

- lineup_table: A table of lineup choices
- k: Number of members in lineup. Must be specified by user (scalar).
- both: Defaults to FALSE. Returns Tredoux’s adjusted effective size estimate. If TRUE, provides both Malpass’s (1981) and Makpass’s adjusted (see: Tredoux, 1998) calculations of effective size.

Details

Reduces the size of a lineup from a (corrected) nominal starting value by the degree to which members are, in sum, chosen below the level of chance expectation.

Value

Malpass’s original & adjusted estimates of effective size

References


Examples

```r
#Data:
lineup_vec <- round(runif(100, 1, 6))

#Call:
esize_m(lineup_vec, 6, both = TRUE)
esize_m(lineup_vec, 6)
```
esize_m_boot

**Bootstrapped Effective Size**

### Description
Base function for computing bootstrapped effective size

### Usage
```
esize_m_boot(lineup_vec, d, k)
```

### Arguments
- `lineup_vec`: A vector of lineup choices
- `d`: Indices for bootstrap resampling
- `k`: A vector indexing number of members in each lineup pair. Must be specified by user (scalar).

### Details
Function to call when bootstrap resampling using boot function (in package 'boot')

### Value
If printarg=FALSE, provides only Malpass's original calculation of effective size

### References

### See Also
- `boot`: https://cran.r-project.org/web/packages/boot/boot.pdf
Examples

```r
#Data:
lineup_vec <- round(runif(100, 1, 6))

#Get boot object:
bootobject <- boot::boot(lineup_vec, esize_m_boot, k = 6, R=1000)
bootobject

#To get confidence intervals:
cis <- boot::boot.ci(bootobject, conf = 0.95, type = "all")
```

---

### esize_T

**Effective Size (Tredoux, 1998)**

#### Description

Function for generating Effective Size (Tredoux, 1998)

#### Usage

```r
esize_T(lineup_table)
```

#### Arguments

- `lineup_table`: A table of lineup choices

#### Details

Reduces the size of a lineup from a (corrected) nominal starting value by the degree to which members are, in sum, chosen below the level of chance expectation.

#### References


Examples

#Data:
lineup_vec <- round(runif(100, 1, 6))
lineup_table <- table(lineup_vec)

#Call:
e <- esize_T(lineup_table)

esize_T_boot

---

Description

Base function for generating boostrapped Effective Size (Tredoux, 1998)

Usage

esize_T_boot(lineup_table, d)

Arguments

- lineup_table: A table of lineup choices
- d: Indices for bootstrap resampling

Details

Reduces the size of a lineup from a (corrected) nominal starting value by the degree to which members are, in sum, chosen below the level of chance expectation.

References


esize_T_ci_n

See Also

boot: https://cran.r-project.org/web/packages/boot/boot.pdf

esize_T_ci_n  Effective Size with Confidence Intervals from Normal Theory (Tredoux, 1998)

Description

Function for generating Effective Size (Tredoux, 1998) with CIs from normal theory

Usage

esize_T_ci_n(lineup_table, alpha)

Arguments

lineup_table  A table of lineup choices
alpha  Alpha level to be declared by user (scalar)

Details

Reduces the size of a lineup from a (corrected) nominal starting value by the degree to which members are, in sum, chosen below the level of chance expectation

References


Examples

#Data:
lineup_vec <- round(runif(100, 1, 6))
lineup_table <- table(lineup_vec)

#Call:
e_ci <- esize_T_ci_n(lineup_table, .95)
**face_sim**  
*Compute similarity of faces in a lineup; experimental function*

**Description**  
Function to compute the degree to which each face in a set of faces loads onto a common factor computed from the faces.

**Usage**  
```r
face_sim()
```

**Details**

- The faces need to be standardized for inter-pupil distance, and for pupil location prior to running the function
- The user will be asked to choose a set of faces through a dialog box. These should be jpeg files.
- There is no argument to the function; it gets what it needs from the dialog box. The function prints the lineup array to the viewer pane in R, and reports the loading of each face on the first common factor in a factor analysis (using fa in package psych)

**References**


---

**func_size**  
*Functional Size*

**Description**  
This function computes functional size (N/D), where N = number of mock witnesses, & D = number of mock witnesses choosing the target

**Usage**  
```r
func_size(lineup_vec, target_pos)
```

**Arguments**

- `lineup_vec` A numeric vector of lineup choices
- `target_pos` A scalar, representing target position in lineup. Must be declared by user
**func_size.boot**

**Details**

An estimation of the number of plausible lineup members in a given lineup.

**Value**

Functional size of lineup

**References**


**Examples**

```r
# Data:
lineup_vec <- round(runif(100, 1, 6))
target_pos <- 3

# Call:
x <- func_size(lineup_vec, target_pos)
x <- func_size(lineup_vec, 3)
```

---

**func_size.boot**

*Bootstrapped Functional Size*

**Description**

This function is a base function for the bootstrapping that ensues to compute bootstrapped confidence intervals for the estimate of functional size.

**Usage**

```r
func_size.boot(lineup_vec, d = d, target_pos)
```

**Arguments**

- `lineup_vec` A numeric vector of lineup choices
- `d` Indices for bootstrap resampling
- `target_pos` A scalar, representing position of target in lineup. Must be declared by user
Details

Function is called bootstrap resampling using boot function (in package `boot`) This function is never called by the user - it is called in the functional size master function.

References


See Also

- `boot`: [https://cran.r-project.org/web/packages/boot/boot.pdf](https://cran.r-project.org/web/packages/boot/boot.pdf)

---

**Description**

This function is a master function, calling other functions it needs, and reporting results in some detail.

**Usage**

```r
func_size_report(lineup_vec, target_pos, k, R)
```

**Arguments**

- `lineup_vec` A numeric vector of lineup choices
- `target_pos` A scalar, representing target position in lineup. Must be declared by user
- `k` Number of members in lineup. Must be specified by user (scalar).
- `R` Number of bootstrap samples. Defaults to 1000

**Details**

Function depends on functions from package `boot`
References


See Also

*boot*: https://cran.r-project.org/web/packages/boot/boot.pdf

Examples

```r
#Data:
lineup_vec <- round(runif(100, 1, 6))
target_pos <- 3

#Call:
x <- func_size_report(lineup_vec, target_pos, 6)
x <- func_size_report(lineup_vec, 3, 6)
```

---

**Description**

Function for computing arbitrary percentile of bootstrapped lineup proportion

**Usage**

```r
gen_boot_propci(lineuprops, perc = 0.05)
```

**Arguments**

- `lineuprops`: A vector of bootstrapped lineup proportions
- `perc`: Percentile to be computed. Must be declared by user (scalar)
  
  Defaults to .05

**Details**

Can be used to calculate confidence intervals at desired level of alpha
Value

Arbitrary percentile of bootstrapped lineup proportion

Examples

```r
# Data:
lineup_vec <- round(runif(100, 1, 6))
target_pos <- 3
k <- 6

# Bootstrap data:
lineup_boot_df <- gen_boot_samples(lineup_vec, 1000)

# Compute proportion for bootstrap samples:
lineups <- gen_lineup_prop(lineup_boot_df, target_pos, 6)

# Get boot CIs:
prop_bootci_lower <- gen_boot_propci(lineups, perc = .025)
prop_bootci_upper <- gen_boot_propci(lineups, perc = .975)
```

---

**gen_boot_propmean_se**  
*Descriptive statistics for bootstrapped lineup proportion*

Description

Function for computing mean, median and se of boot proportion

Usage

```r
gen_boot_propmean_se(lineups)
```

Arguments

- `lineups` A dataframe of bootstrapped lineup proportions

Value

Mean, median, standard deviation, standard error & 95 CIs of lineup proportion across a bootstrapped dataframe

Examples

```r
# Data:
lineup_vec <- round(runif(100, 1, 6))
target_pos <- 3

# Bootstrap data:
lineup_boot_df <- gen_boot_samples(lineup_vec, 1000)
```
#Compute proportion for bootstrap samples:
lineuprops <- gen_lineup_prop(lineup_boot_df, target_pos = 3, k = 6)

#Call:
gen_boot_propmean_se(lineuprops)

#OR:

lineuprops <- boot::boot(lineup_vec, lineup_prop_boot, target_pos = 3, R = 1000)
gen_boot_propmean_se(lineuprops$t)

gen_boot_samples	Bootstrap resampling

Description

Function for generating bootstrapped samples from 1 vector of lineup data

Usage

gen_boot_samples(lineup_vec, bootno)

Arguments

lineup_vec	A numeric vectors of lineup choices
bootno	Number of bootstrap samples

Value

A dataframe of bootstrapped lineup data

Examples

#Data:
lineup_vec <- round(runif(100, 1, 6))
bootno <- 1000

#Call:
bootdf <- gen_boot_samples(lineup_vec, bootno)
gen_boot_samples_list  Bootstrapped resampling

Description

Function for generating bootstrapped samples for of k vectors of lineup choices

Usage

```r
gen_boot_samples_list(lineup_list, bootno)
```

Arguments

- `lineup_list`: A list containing k vectors of lineup choices for k lineups, in which the target was either absent or present
- `bootno`: Number of bootstrap samples

Value

A list of bootstrapped lineup data Length of list = no. of bootstrap sample draws

Examples

```r
#Data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
linelist <- list(A, B, C)
rm(A, B, C)

bootno <- 1000

#Call:
bootdata <- gen_boot_samples_list(linelist, bootno)
```

gen_esize_m  Effective Size (across a dataframe)

Description

Function for computing Effective Size (Tredoux, 1998) on lineups contained as columns in a df, usually from a bootstrapped sample

Usage

```r
gen_esize_m(lineup_boot_df, k)
```
Arguments

- `lineup_boot_df` A dataframe containing bootstrapped samples of lineup data
- `k` Number of members in lineup. Must be specified by user (scalar).

Details

This function computes effective size for k lineups simultaneously.

Value

A vector of effective size calculations for each lineup in bootstrapped df

References


Examples

```r
#Data:
lineup_vec <- round(runif(100,1,6))
bootdf <- gen_boot_samples(lineup_vec, 1000)

#Call:
esize_vec <- gen_esize_m(bootdf, 6)
```

Description

Function for computing Effective Size (Malpass, 1981, as adjusted by Tredoux, 1998) with CIs from bootstrap df of lineups

Usage

```r
gen_esize_m_ci(lineupsizes, perc = 0.05)
```
Arguments

- **lineupsizes**: A vector of bootstrapped effective sizes
- **perc**: Defaults to .05. Can be specified by user, according to desired level of alpha (scalar)

Value

Confidence intervals for effective size

Examples

```r
#Data:
lineup_vec <- round(runif(100, 1, 6))
k <- 6

#Use gen_boot_samples to get bootstrapped data:
bootdata <- gen_boot_samples(lineup_vec, 1000)

#Compute effective size over df of bootstrapped data:
lineupsizes <- gen_esize_m(bootdata, 6)

#Call:
gen_esize_m_ci(lineupsizes)
gen_esize_m_ci(lineupsizes, perc = .025)
gen_esize_m_ci(lineupsizes, perc = .975)
```

---

**gen_lineup_prop**

Lineup proportion over dataframe

Description

Function for computing lineup proportion over a dataframe

Usage

(gen_lineup_prop(lineup_boot_df, target_pos, k)

Arguments

- **lineup_boot_df**: Dataframe of lineup choices (usually a bootstrapped set)
- **target_pos**: A scalar, representing target position in lineup. Must be declared by user.
- **k**: A vector indexing number of members in each lineup pair (nominal size). Must be specified by user (scalar).

Value

A vector of bootstrapped proportions, indicating the frequency with which a target was identified in a lineup. Length of vector = number of bootstrap sample draws
Examples

#Data
lineup_vec <- round(runif(100, 1, 6))
target_pos <- 3

#Get bootstrapped data:
lineup_boot_df <- gen_boot_samples(lineup_vec, 1000)

#Call:
lineuprops <- gen_lineup_prop(lineup_boot_df, 3, 6)

desc

Description

A function that takes a table of lineup choices and generates a vector of raw lineup data

Usage

gen_linevec(lineup_table, k)

Arguments

lineup_table A table of lineup choices
k Nominal size (i.e., total number of lineup members). Must be declared by user

Value

Returns a vector of lineup choices

Examples

#Data:
lineup_vec <- round(runif(100, 1, 6))
lineup_table <- table(lineup_vec)

#Call:
lineup_vec <- gen_linevec(lineup_table, 3)
**homog_diag**

*Master function: Homogeneity of diagnosticity ratio*

**Description**

This function provides assesses the homogeneity of the diagnosticity ratio of \( k \) lineup pairs.

**Usage**

```r
homog_diag(lineup_pres_list, lineup_abs_list, pos_list, k)
```

**Arguments**

- **lineup_pres_list**: A list containing \( k \) vectors of lineup choices for \( k \) lineups, in which the target was present
- **lineup_abs_list**: A list containing \( k \) vectors of lineup choices for \( k \) lineups, in which the target was absent
- **pos_list**: A numeric vector indexing lineup member positions for the target present & absent conditions.
- **k**: A vector indexing number of members in each lineup pair (nominal size). Must be specified by user (scalar).

**Details**

Master function for assessing homogeneity of diagnosticity ratio for \( k \) independent lineups.

**Value**

Computes diagnosticity ratio with chi-squared estimate and significance level for \( k \) lineup pairs

**References**


Examples

#Target present data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_pres_list <- list(A, B, C)
rm(A, B, C)

#Target absent data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_abs_list <- list(A, B, C)
rm(A, B, C)

#Pos list
lineup1_pos <- c(1, 2, 3, 4, 5, 6)
lineup2_pos <- c(1, 2, 3, 4, 5)
lineup3_pos <- c(1, 2, 3, 4)
pos_list <- list(lineup1_pos, lineup2_pos, lineup3_pos)
rm(lineup1_pos, lineup2_pos, lineup3_pos)

#Nominal size:
k <- c(6, 5, 4)

#Call:
homog_diag(lineup_pres_list, lineup_abs_list, pos_list, k)

---

homog_diag_boot  

Homogeneity of diagnosticity ratio with bootstrapped CIs

Description

Function for computing bootstrapped estimates of homogeneity of diagnosticity ratio

Usage

homog_diag_boot(lineup_pres_list, lineup_abs_list, k, R = 100)

Arguments

lineup_pres_list
A list containing k vectors of lineup choices for k lineups, in which the target was present

lineup_abs_list
A list containing k vectors of lineup choices for k lineups, in which the target was absent

k  
Number of members in lineup. Must be specified by user (scalar).

R  
Number of bootstrap replications. Defaults to R = 100.
Details

Computes bootstrapped diagnosticity ratio with chi-squared estimate, significance level and confidence intervals for k lineup pairs.

References


Examples

```r
#Target present data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_pres_list <- list(A, B, C)
rm(A, B, C)

#Target absent data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_abs_list <- list(A, B, C)
rm(A, B, C)
```

---

### i_esize_T

**Description**

Function for generating I component of effective size

**Usage**

`i_esize_T(lineup_table)`
Arguments

lineup_table A table of lineup choices

References


Examples

```r
#Data:
lineup_vec <- round(runif(100, 1, 6))
lineup_table <- table(lineup_vec)

#Call:
i <- i_esize_T(lineup_table)
```

Description

A dataframe of lineup choices for 1 trial; real data from a Doob & Kirshenbaum (1973) article. This article was the first to discuss lineup size 1 column x 42 observations

42 mock witnesses Nominal size is 6 Target position is 4

Usage

data(line73)

Format

An object of class data.frame with 42 rows and 1 columns.

References

lineup_boot_allprop  Confidence intervals for lineup proportion

Description
Computes bootstrapped confidence intervals for lineup proportion

Usage
lineup_boot_allprop(lineup_vec, k, conf = 0.95)

Arguments
lineup_vec  A numeric vector of lineup choices
k  Number of targets in lineup. Must be specified by user (scalar).
conf  Desired level of alpha. Defaults to 0.95. May be specified by user (scalar).

Details
Function that computes bootstrapped lineup proportion using 1000 bootstrap draws Calls 'boot function in 'boot' package

Value
Returns a vector of bias corrected confidence intervals for lineup proportion for each member in a lineup

References

See Also
boot: https://cran.r-project.org/web/packages/boot/boot.pdf

Examples
#Data:
lineup_vec <- round(runif(100, 1, 6))

#Call:
lineuprops_ci <- lineup_boot_allprop(lineup_vec, k = 6)
lineuprops_ci <- lineup_boot_allprop(lineup_vec, k = 6, conf = 0.975)
lineup_prop_boot

Bootstrapped lineup proportion

Description
Base function for computing bootstrapped lineup proportion for a lineup member

Usage
lineup_prop_boot(lineup_vec, d, target_pos)

Arguments
lineup_vec A numeric vector of lineup choices
d Indices for bootstrap sample. Argument used by boot function to select samples for bootstrapping
target_pos A scalar, representing target position in lineup. Must be declared by user

Details
Function to call when bootstrap resampling using boot function

References

See Also
boot: https://cran.r-project.org/web/packages/boot/boot.pdf

Examples
```r
lineup_vec <- round(runif(100, 1, 6))
bootobject <- boot::boot(lineup_vec, lineup_prop_boot, target_pos = 3, R = 1000)
cis <- boot::boot.ci(bootobject, conf = 0.95, type = "all")
```
### lineup_prop_tab

**Lineup proportion**

**Description**
Computes the proportion of mock witnesses identifying a particular lineup member

**Usage**
```r
lineup_prop_tab(lineup_table, target_pos)
```

**Arguments**
- `lineup_table`: A table of lineup choices
- `target_pos`: A scalar, representing target position in lineup. Must be declared by user

**Value**
Returns a proportion indicating the frequency with which a lineup member was selected

**References**

**Examples**
```r
#Data:
lineup_vec <- round(runif(100, 1))
lineup_table <- table(lineup_vec)

#Call:
lineup_prop_tab(lineup_table, 3)
lineup_prop_tab(table(lineup_vec), 2)
```

---

### lineup_prop_vec

**Lineup proportion**

**Description**
Computes the proportion of mock witnesses identifying a particular lineup member

**Usage**
```r
lineup_prop_vec(lineup_vec, target_pos, k)
```
**ln_diag_ratio**

Arguments

- `lineup_vec` A numeric vector of lineup choices
- `target_pos` Suspect/lineup member position. Must be declared by user (scalar).
- `k` Nominal size (i.e., total number of members in lineup). Must be specified by user (scalar).

Value

Returns a proportion indicating the frequency with which a target was identified in a lineup

References


Examples

```r
# Data:
lineup_vec <- round(runif(100, 1, 6))

# Call:
lineup_prop_vec(lineup_vec, 3, 6)
```

---

**ln_diag_ratio**  
*Ln of Diagnosticity Ratio*

Description

Computes ln of diagnosticity ratio: ln(d)

Usage

```r
ln_diag_ratio(linedf)
```

Arguments

- `linedf` A dataframe of parameters for computing diagnosticity ratio

Details

To get `linedf`, use the `diag_param` helper function

`diag_param` returns a dataframe containing the following:

- `n11`: Number of mock witnesses who identified the suspect in the target present condition
- `n21`: Number of mock witnesses who did not identify the suspect in the target present condition
- `n12`: Number of mock witnesses who identified the suspect in the target absent condition
- `n13`: Number of mock witnesses who did not identify the suspect in the target absent condition
References


Examples

```r
#Target present data:
A <- round(runif(100, 1, 6))
B <- round(runif(70, 1, 5))
C <- round(runif(20, 1, 4))
lineup_pres_list <- list(A, B, C)
rml(A, B, C)

#Target absent data:
A <- round(runif(100, 1, 6))
B <- round(runif(70, 1, 5))
C <- round(runif(20, 1, 4))
lineup_abs_list <- list(A, B, C)
rml(A, B, C)

#Pos list
lineup1_pos <- c(1, 2, 3, 4, 5, 6)
lineup2_pos <- c(1, 2, 3, 4, 5)
lineup3_pos <- c(1, 2, 3, 4)
pos_list <- list(lineup1_pos, lineup2_pos, lineup3_pos)
rml(lineup1_pos, lineup2_pos, lineup3_pos)

#Nominal size:
k <- c(6, 5, 4)

#Use diag param helper function to get data (n11, n21, n12, n22):
linedf <- diag_param(lineup_pres_list, lineup_abs_list, pos_list, k)

#Call:
ln <- ln_diag_ratio(linedf)
```
makevec_prop

**Helper functions**

**Description**

Several helper functions that compute bootcits from proportions

**Usage**

`makevec_prop(prop, n)`

**Arguments**

- **prop**
  
  A dataframe of bootstrapped proportions

- **n**
  
  Number of lineup members

---

**make_roc**

*Compute and plot ROC curve for lineup accuracy ~ confidence*

**Description**

Function to compute and plot an ROC curve for data from an eyewitness experiment, where accuracy is recorded for target present and target absent lineups

**Usage**

`make_roc(df_confacc)`

**Arguments**

- **df_confacc**
  
  A dataframe with two columns, named confidence and accuracy (where accuracy = binary accuracy)

**Details**

The approach is outlined in several papers by Mickes, Wixted, Gronlund, Clark, and others (see References)

This function is a user level function. It chains the two roc functions together. The user must pass a dataframe, with one column indicating confidence, and another accuracy, and these must be named as such.

The approach is outlined in several papers by Mickes, Wixted, Gronlund, Clark, and others (see references)

**Value**

An ROC object of package pROC
make_rocdata

References


Examples

```r
#Data:
data(mickwick)

#Call:
make_roc(mickwick)
```

---

**make_rocdata**

*Helper functions: Compute and plot ROC curve for lineup accuracy ~ confidence*

Description

Function to compute and plot an ROC curve for data from an eyewitness experiment, where accuracy is recorded for target present and target absent lineups

Usage

```r
make_rocdata(df_confacc)
```

Arguments

- `df_confacc`: A df with two columns, confidence and accuracy, where accuracy = binary accuracy.

Value

An ROC object of package pROC
Confidence & Accuracy data (Mickes & Wixted)

Description

Data capturing confidence and accuracy for a lineup trial

Usage

data(mickwick)

Format

An object of class tbl_df (inherits from tbl, data.frame) with 100 rows and 2 columns.

Details

mickwick

mockdata

Description

A dataframe containing real lineup data.
3 columns x 94 observations.
The first column simply provides the participant number, whilst the second and third contain data for each participant’s choice (i.e., the position of the lineup member that they identified) and a confidence score for each choice.
In the field of eyewitness testimony, it is common for participants to rate how confident they are that they have accurately identified the perpetrator. Here, confidence is recorded as a percentage.
This lineup has the following parameters:

- 94 mock witness choices
- Target position is 7
- Nominal size is 9

Usage

data(mockdata)

Format

An object of class tbl_df (inherits from tbl, data.frame) with 94 rows and 3 columns.
Description
A data file containing lineup data for 3 independent lineups. 133 rows x 3 columns. Each lineup trial consisted of 133 mock witnesses & 8 lineup members.

- : Target position for lineup 1 is 3
- : Target position for lineup 2 is 6
- : Target position for lineup 3 is 2

Usage
data(nortje2012)

Format
An object of class tbl_df (inherits from tbl, data.frame) with 133 rows and 3 columns.

Description
Helper function that repeats a number k times

Usage
rep_index(index, num)

Arguments
index Number to be repeated
num Number of times to repeat index

Details
Can be used to create vectors of lineup data

Examples
rep_index(1, 10)
rot_vector

Description
Helper function to rotate vector

Usage
rot_vector(lineup_vec)

Arguments
lineup_vec A vector of lineup data

show_lineup

Description
Helper function

Usage
show_lineup(file_list)

Arguments
file_list A selection of image files (raw data)

var_diag_ratio

Description
Computes the variance of the diagnosticity ratio for a lineup pair

Usage
var_diag_ratio(lineup_pres, lineup_abs, pos_pres, pos_abs, k1, k2)
Arguments

- lineup_pres: A numeric vector of lineup choices for a lineup in which the target was present
- lineup_abs: A numeric vector of lineup choices for a lineup in which the target was absent
- pos_pres: A scalar, representing target position in TP lineup. Must be declared by user
- pos_abs: A scalar, representing target position in TA lineup. Must be declared by user
- k1: Number of targets in TP lineup. Must be specified by user (scalar).
- k2: Number of targets in TA lineup. Must be specified by user (scalar).

References


Examples

```r
#Data:
lineup_pres <- round(runif(100, 1, 6))
lineup_abs <- round(runif(100, 1, 6))
pos_pres <- 3
pos_abs <- 4

#Call:
var_d <- var_diag_ratio(lineup_pres, lineup_abs, pos_pres, pos_abs, 6, 6)
var_d <- var_diag_ratio(lineup_pres, lineup_abs, 3, 4, 6, 6)
```

**Description**

Function to compute variance of ln(d) for k lineup pairs

**Usage**

`var_lnd(linedf)`
Arguments
linedf A dataframe of parameters for computing diagnosticity ratio

Details
To get linedf, use the diag_param helper function
diag_param returns a dataframe containing the following:

• n11: Number of mock witnesses who identified the suspect in the target present condition
• n21: Number of mock witnesses who did not identify the suspect in the target present condition
• n12: Number of mock witnesses who identified the suspect in the target absent condition
• n13: Number of mock witnesses who did not identify the suspect in the target absent condition

Value
A dataframe containing ln of the variance of the diagnosticity ratio for each lineup.

References

Examples
#Target present data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_pres_list <- list(A, B, C)
rm(A, B, C)

#Target absent data:
A <- round(runif(100,1,6))
B <- round(runif(70,1,5))
C <- round(runif(20,1,4))
lineup_abs_list <- list(A, B, C)
rm(A, B, C)
# Pos list
lineup1_pos <- c(1, 2, 3, 4, 5, 6)
lineup2_pos <- c(1, 2, 3, 4, 5)
lineup3_pos <- c(1, 2, 3, 4)
pos_list <- list(lineup1_pos, lineup2_pos, lineup3_pos)
rm(lineup1_pos, lineup2_pos, lineup3_pos)

# Nominal size:
k <- c(6, 5, 4)

# Use diag_param helper function to get data (n11, n21, n12, n22):
linedf <- diag_param(lineup_pres_list, lineup_abs_list, pos_list, k)
# Call:
var <- var_lnd(linedf)
Index

* datasets
  line73, 37
  mickwick, 45
  mockdata, 45
  nortje2012, 46
  allfoil_cihigh, 4
  allfoilbias, 3
  allprop, 4
  boot, 17, 20, 23, 26, 27, 38, 39
  chi_diag, 5
  compare_eff_sizes.boot, 6
  d_bar, 13
  d_weights, 15
  datacheck1, 7
  datacheck2, 8
  datacheck3, 9
  diag_param, 9
  diag_ratio_T, 11
  diag_ratio_W, 12
  eff_size_per_foils, 18
  effsize_compare, 16
  esize_m, 18
  esize_m.boot, 20
  esize_T, 21
  esize_T.boot, 22
  esize_T.ci_n, 23
  face_sim, 24
  func_size, 24
  func_size.boot, 25
  func_size_report, 26
  gen_boot_propci, 27
  gen_boot_propmean_se, 28
  gen_boot_samples, 29
  gen_boot_samples_list, 30
  gen_esize_m, 30
  gen_esize_m_ci, 31
  gen_lineup_prop, 32
  gen_linevec, 33
  homog_diag, 34
  homog_diag_boot, 35
  i_esize_T, 36
  line73, 37
  lineup_boot_allprop, 38
  lineup_prop.boot, 39
  lineup_prop_tab, 40
  lineup_prop_vec, 40
  ln_diag_ratio, 41
  make_roc, 43
  make_rocdata, 44
  makevec_prop, 43
  mickwick, 45
  mockdata, 45
  nortje2012, 46
  rep_index, 46
  rot_vecor, 47
  show_lineup, 47
  var_diag_ratio, 47
  var_lnd, 48