Package ‘reappraised’

October 6, 2023

Title Statistical Tools for Assessing Publication Integrity of Groups of Trials

Version 0.1.1

Description Takes user-provided baseline data from groups of randomised controlled data and assesses whether the observed distribution of baseline p-values, numbers of participants in each group, or categorical variables are consistent with the expected distribution, as an aid to the assessment of integrity concerns in published randomised controlled trials.
References (citations in PubMed format in details of each function):
Bolland MJ, Avenell A, Gamble GD, Grey A. (2016) <doi:10.1212/WNL.0000000000003387>.
Bolland MJ, Gamble GD, Avenell A, Grey A, Lumley T. (2019) <doi:10.1016/j.jclinepi.2019.05.006>.
Bolland MJ, Gamble GD, Avenell A, Grey A. (2019) <doi:10.1016/j.jclinepi.2019.03.001>.
Bolland MJ, Gamble GD, Grey A, Avenell A. (2020) <doi:10.1111/anae.15165>.
Bolland MJ, Gamble GD, Avenell A, Cooper DJ, Grey A. (2021) <doi:10.1016/j.jclinepi.2020.11.012>.
Bolland MJ, Gamble GD, Avenell A, Grey A. (2021) <doi:10.1016/j.jclinepi.2021.05.002>.
Bolland MJ, Gamble GD, Avenell A, Cooper DJ, Grey A. (2023) <doi:10.1016/j.jclinepi.2022.12.018>.
Carlisle JB, Loadsman JA. (2017) <doi:10.1111/anae.13650>.
Carlisle JB. (2017) <doi:10.1111/anae.13938>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports boot, broom, data.table, dplyr, epitools, flextable, ggplot2, ggpubr, magrittr, officer, purrr, readxl, rlang, stats, tidyr, utils, vcd, vcdExtra

Depends R (>= 2.10)

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Mark Bolland [aut, cre, cph] (<`https://orcid.org/0000-0003-0465-2674`>)

Maintainer Mark Bolland <m.bolland@auckland.ac.nz>

Repository CRAN

Date/Publication 2023-10-06 04:20:02 UTC

R topics documented:

---

anova_fn — *Compares differences between baseline means using Carlisle's montecarlo anova method*

Description

Creates plots of distribution of p-values for differences in baseline means calculated using Carlisle's montecarlo anova method.

Usage

```
anova_fn(
  df = anova_data,
  method = "alt",
  seed = 0,
  sims = -1,
  btsp = 500,
  title = "",
  verbose = TRUE
)
```

Arguments

`df` dataframe generated from load_clean function

`method` "orig" is adapted from original code; "alt" avoids using loops in the code (see details)

`seed` the seed to use for random number generation, default 0 = current date and time. Specify seed to make repeatable.

`sims` number of simulations, default -1 = function selects based on number of variables and sample size

`btsp` number of bootstrap repeats used to generate 95% confidence interval around AUC

`title` optional title for plots

`verbose` TRUE or FALSE indicates whether progress bar and comments show and prints plot

Details

Method is from Carlisle JB, Loadsman JA. Evidence for non-random sampling in randomised, controlled trials by Yuhji Saitoh. Anaesthesia. 2017;72:17-27.
R code is in appendix to paper. This function is adapted from that code.
The function has two methods. The published code selects each variable from each study then generates simulations for that variable using a row-wise approach with several loops. The adapted method is method = "orig". The method = "alt" generates all the simulations at once and initially I thought was considerably faster, but in practice the time savings are small.
The results from the two approaches will not be identical even if the same random number seed is used because they use the generated random numbers in different orders but the p-values generated differ by about <0.1. Usually the differences are close to 0.01 (although this depends on the number of simulations- more simulations = smaller differences). The code that generates the p-value for each variable from the simulated means is essentially the same.

Returns a list containing 3 objects and (if verbose = TRUE) prints the plot anova_ecdf

Value

list containing 3 objects as described

- anova_ecdf = plot of cumulative distribution of calculated p-values compared to the expected uniform distribution
- anova_pvalues = plots of distribution of calculated p-values and AUC, as for pval_cont_fn()
- anova_all_results = list containing
  - anova_data = data frame of baseline data, with calculated p-values
  - anova_pvals = plot of distribution of calculated p-values from anova_pvalues
  - anova_auc = plot of AUC of calculated p-values from anova_pvalues

Examples

```
# load example data
anova_data <- load_clean(import= "no", file.cont = "SI_pvals_cont",anova= "yes",
format.cont = "wide")$anova_data


# run function (takes only a few seconds)
anova_fn(seed=10, sims = 100, btsp = 100)$anova_ecdf

# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                    mustWork = TRUE)
# delete file name from path
path <- sub("/[^/]+$", "", path)

# load data
anova_data <- load_clean(import= "yes", anova = "yes", dir = path,
     file.name.cont = "reappraised_examples.xlsx", sheet.name.cont = "SI_pvals_cont",
     range.name.cont = "A:O", format.cont = "wide")$anova_data
```

---

cat_all_fn *Compares observed and expected distribution of all categorical (binomial) variables*

Description

Creates plots of observed to expected numbers and ratios for the binomial variables and/or compares reported and calculated p-values for the variables
Reference: Bolland MJ, Gamble GD, Avenell A, Cooper DJ, Grey A. Distributions of baseline categorical variables were different from the expected distributions in randomized trials with integrity concerns. J Clin Epidemiol. 2023;154:117-124

Usage

```
cat_all_fn(
  df = cat_all_data,
  comp.pvals = "no",
  fisher.sim = "y",
  fish.n.sims = 10000,
  binom = "no",
  two_levels = "no",
  del.disparate = "yes",
  excl.level = "yes",
  seed = 0,
  title = "",
  verbose = TRUE
)
```

Arguments

`df` data frame generated from load_clean function

`comp.pvals` "yes" or "no" indicator whether reported and calculated p-values should be compared

`fisher.sim` "yes" or "no" indicator whether to allow fisher test to simulate p-values for >2*2 tables

`fish.n.sims` number of simulations to use in Fisher test, default 10,000

`binom` "yes" or "no" indicator whether observed to expected distributions of binomial variables should be calculated

`two_levels` "yes" or "no" indicator whether variables with more than 2 levels should be collapsed to 2 levels

`del.disparate` if yes, data in which the absolute difference between group sizes is >20% are deleted

`excl.level` "yes" or "no" indicator whether one level of a variable should be deleted. Deleted level is chosen randomly using seed parameter.

`seed` seed for random number generator, default 0 = current date and time. Specify seed to make repeatable.

`title` title name for plots (optional)

`verbose` TRUE or FALSE indicates whether progress bar and comments show and flextable or plot or both are printed

Details

Returns a list containing objects described below and (if verbose = TRUE) prints the flextable cat_all_diff_calc_rep_ft and/or graph cat_all_graph depending on options chosen

Value

list containing objects as described

if p-value comparison used:

- cat_all_pvals = data frame of data for comparison of reported and calculated p-values
- cat_all_diff_calc_rep_ft = flextable of comparison of reported and calculated p-values
- cat_all_diff_calc_rep_data = data frame used to make flextable
- cat_all_diff_thresh_ft = flextable of comparison of reported and calculated p-values when only threshold given
- cat_all_diff_thresh_data = data frame used to make flextable for p-value thresholds

if comparing categorical variables used

- cat_all_graph = plot of observed to expected numbers and differences between groups, top panels are the absolute numbers, bottom panels are the differences between trial arms in two arm studies
- cat_all_graph_pc = plot of observed to expected numbers expressed as percentages and differences between groups, top panels are the percentages, bottom panels are the differences between trial arms in two arm studies
- cat_all_data_abs = data frame of data for absolute numbers
- cat_all_data_df = data frame of data for difference between groups in two arm studies
- cat_all_dataset_abs = data frame of dataset used for all trials
- cat_all_dataset_df = data frame of dataset used for two arm trials
- cat_all_all_graphs list containing
  - abs = plot for absolute numbers only
  - df = plot for difference between groups in two arm studies only
  - pc = plot for percentages only
  - all_pc = composite plot of percentages and absolute numbers
  - individual_graphs list of 6 individual plots making up composite figures

Examples

```
# load example data
cat_all_data <- load_clean(import= "no", file.cat = "SI_cat_all", cat_all= "yes",
format.cat = "wide")$cat_all_data


# run function comparing p-values only (takes only a few seconds)
cat_all_fn (comp.pvals = "yes")$cat_all_diff_calc_rep_ft

# run function comparing distribution of binomial variables only

# to speed example up limit to 12 2-arm trials with 20 variables
# (takes close to 5 secs)

cat_all_data <- cat_all_data [1:41, c(1:8,10:11,13:15)]

cat_all_fn (binom = "yes", two_levels = "yes", del.disparate = "yes",
excl.level = "yes", seed = 10)$cat_all_graph


# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                   mustWork = TRUE)
# delete file name from path
path <- sub("/[^/]+$", "", path)

# load data
```

```
cat_all_data <- load_clean(import= "yes", cat_all = "yes", dir = path,
   file.name.cat = "reappraised_examples.xlsx", sheet.name.cat = "SI_cat_all",
   range.name.cat = "A:N", format.cat = "wide")$cat_all_data
```

---

cat_fn *Compares observed and expected distribution of a categorical (binomial) variable*

Description

Creates plots of observed to expected numbers and ratios for the specified binomial variable

Usage

```
cat_fn(
  df = cat_data,
  x_title = "",
  prefix = "",
  del.disparate = "yes",
  title = "",
  verbose = TRUE
)
```

Arguments

`df`	data frame generated from load_clean function
`x_title`	name of the variable for use on the x-axis
`prefix`	letter for variable columns in data frame
`del.disparate`	if yes, data in which the absolute difference between group sizes is >20% are deleted
`title`	title name for plots (optional)
`verbose`	TRUE or FALSE indicates whether to print plot

Details

An example is for trial withdrawls in Bolland 2021
Bolland MJ, Gamble GD, Avenell A, Cooper DJ, Grey A. Participant withdrawals were unusually distributed in randomized trials with integrity concerns: a statistical investigation. J Clin Epidemiol 2021;131:22-29.

Returns a list containing 4 objects and (if verbose = TRUE) prints the plot cat_graph

Value

list containing 4 objects as described

- cat_graph = plot of observed to expected numbers and differences between groups, top panels are the absolute numbers, bottom panels are the differences between trial arms in two arm studies
- cat_data_abs = data frame of data for absolute numbers
- cat_data_df = data frame of data for difference between groups in two arm studies
- cat_all_graphs = list containing
  - abs = plot for absolute numbers only
  - df = plot for difference between groups in two arm studies only
  - individual_graphs list of 4 individual plots making up composite figures

Examples

```
# load example data
cat_data <- load_clean(import= "no", file.cat = "SI_cat", cat= "yes",
format.cat = "wide", cat.names = c("n", "w"))$cat_data


# run function (takes only a few seconds)
cat_fn(x_title= "withdrawals", prefix="w", del.disparate = "yes")$cat_graph


# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                    mustWork = TRUE)
# delete file name from path
path <- sub("/[^/]+$", "", path)

# load data
cat_data <- load_clean(import= "yes", cat = "yes", dir = path,
    file.name.cat = "reappraised_examples.xlsx", sheet.name.cat = "SI_cat",
    range.name.cat = "A:G", cat.names = c("n", "w"), format.cat = "wide")$cat_data
```

cohort_fn *Compares proportions of matching summary statistics in different cohorts*

Description

Creates flextable of probability of matching mean, SD, and mean and SD for each variable in different cohorts in the specified number of simulations

Usage

```
cohort_fn(
  df = cohort_data,
  seed = 0,
  sims = -1,
  n_vars = 10,
  popn = "",
  title = "",
  verbose = TRUE
)
```

Arguments

`df` data frame generated from load_clean function

`seed` the seed to use for random number generation, default 0 = current date and time. Specify seed to make repeatable.

`sims` number of simulations, default -1 = function selects based on number of variables and sample size.

`n_vars` restrict analyses to variables in at least (>=) this number of cohorts, default = 10 (ie variable has mean in 10 or more cohorts).

`popn` if dataset contains studies in different sub-populations, code this in cohort_data$population and studies are subsetted if match in this variable. 'All' overrides this and uses all data regardless of information in this variable.

`title` title name for plots (optional)

`verbose` TRUE or FALSE indicates whether progress bar and comments show and flextable is printed

Details

Reference data is from Bolland 2021
Bolland MJ, Gamble GD, Avenell A, Grey A. Identical summary statistics were uncommon in randomized trials and cohort studies. J Clin Epidemiol 2021;136:180-188.

Returns a list containing 6 objects and (if verbose = TRUE) prints the flextable cohort_ft

Value

list containing 6 objects as described

- cohort_ft = flextable of results
- cohort_graph = plot of observed to expected numbers of matches per cohort for mean; SD; and mean and SD
- all_graphs = list containing
  - all_graphs = all plots on single plot
  - both_graphs = list of 3 plots row by row used to form all_graphs
  - individual_graphs= list of 6 individual plots used to form all_graphs

- cohort_cohort_data = data frame used to generate results data
- cohort_prob_data = data frame used to make flextable
- cohort_oe_data= data frame used to make observed to expected plots

Examples

```
# load example data
cohort_data <- load_clean(import= "no", file.cont = "SI_cohort", cohort= "yes",
format.cont = "long")$cohort_data


# run function (takes close to 5 seconds)
cohort_fn(seed=10, sims = 100)$cohort_ft

# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                     mustWork = TRUE)
# delete file name from path
path <- sub("/[^/]+$", "", path)

# load data
cohort_data <- load_clean(import= "yes", cohort = "yes", dir = path,
     file.name.cont = "reappraised_examples.xlsx", sheet.name.cont = "SI_cohort",
     range.name.cont = "A1:F101", format.cont = "long")$cohort_data
```

final_digit_fn *Compares proportions of final digits from summary statistics*

Description

Creates graph of proportion of final digits for summary statistics of specified variables

Usage

```
final_digit_fn(
  df = generic_data,
  vars = "",
  dec.pl = "no",
  dec.pl.vars = "",
  title = "",
  verbose = TRUE
)
```

Arguments

`df` data frame generated from load_clean function

`vars` vector of the summary statistics to be used

`dec.pl` "yes" or "no" indicator whether columns for decimal places are included (yes) or should be calculated (no)

`dec.pl.vars` vector of the names of the columns for decimal places for each statistics

`title` title name for plots (optional)

`verbose` TRUE or FALSE indicates whether print plot

Details

This approach is still in development and needs validation and discussion about its place in integrity assessment.

Requires data frame containing columns for study, variable (named var), summary statistic(s) (named with single letter eg m or s), and optional columns for decimal places for each statistic (named dp_* eg dp_m, dp_s). Data can be imported using the generic option of load_clean function

Returns a list containing 5 objects and prints the plot digit_graph

Value

list containing 5 objects as described

- digit_graph = plot of proportions of final digits
- digit_ft = flextable of results
- digit_table = data frame of results
- digit_dataset = data frame of data set used to generate results data
- digit_data = results of analyses used to generate results data

Examples

```
# load example data
generic_data <- load_clean(import= "no", file.cont = "SI_pvals_cont", generic= "yes",
gen.vars.del = c("p"), format.cont = "wide")$generic_data


# run function (takes only a few seconds)
final_digit_fn(vars = c("m","s"), dec.pl = "n")$digit_graph

# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                    mustWork = TRUE)
# delete file name from path
```

```
path <- sub("/[^/]+$", "", path)

# load data
generic_data <- load_clean(import= "yes", generic = "yes", dir = path,
    file.name.cont = "reappraised_examples.xlsx", sheet.name.cont = "SI_pvals_cont",
    range.name.cont = "A1:O51", gen.vars.del = c("p"),
    format.cont = "wide")$generic_data
```

load_clean *Load data then clean and format it*

Description

Function loads and cleans data for the nine functions

Usage

```
load_clean(
  import = "yes",
  file.cont = "",
  file.cat = "",
  dir = "",
  file.name = "",
  pval_cont = "no",
  match = "no",
  cohort = "no",
  anova = "no",
  dir.cont = "",
  file.name.cont = "",
  sheet.name.cont = "Sheet1",
  range.name.cont = "",
  format.cont = "wide",
  cat = "no",
  sr = "no",
  cat_all = "no",
  pval_cat = "no",
  cat.names = c("n"),
  dir.cat = "",
  file.name.cat = "",
  sheet.name.cat = "Sheet1",
  range.name.cat = "",
  format.cat = "wide",
  generic = "",
  gen.vars.keep = "",
  gen.vars.del = "",
  verbose = TRUE
)
```

Arguments

`import` ’yes’ indicates import excel file. ’no’ indicates takes dataset already loaded into R as data frame

`file.cont` If import = ’no’, name of data frame containing continuous data

`file.cat` If import = ’no’, name of data frame containing categorical data

`dir` If import = ’yes’, path to location of excel file for continuous and categorical data

`file.name` If import = ’yes’, file name of excel file containing continuous and categorical data

`pval_cont` ’yes’/’no’ indicating if data will be used for pval_cont_fn. Only data for 1 continuous data function can be loaded with each run of this function.

`match` ’yes’/’no’ indicating if data will be used for match_fn. Only data for 1 continuous data function can be loaded with each run of this function.

`cohort` ’yes’/’no’ indicating if data will be used for cohort_fn. Only data for 1 continuous data function can be loaded with each run of this function.

`anova` ’yes’/’no’ indicating if data will be used for anova_fn. Only data for 1 continuous data function can be loaded with each run of this function.

`dir.cont` If import = ’yes’, path to location of excel file for continuous data

`file.name.cont` If import = ’yes’, file name of excel file containing continuous data

`sheet.name.cont` Sheet name containing continuous data

`range.name.cont` Range of cells containing continuous data. Can be in format ’a1:b20’ or ’a:b’

`format.cont` ’wide’/’long’ indicating continuous data is in wide or long format

`cat` ’yes’/’no’ indicating if data will be used for cat_fn. Only data for 1 categorical data function can be loaded with each run of this function.

`sr` ’yes’/’no’ indicating if data will be used for sr_fn. Only data for 1 categorical data function can be loaded with each run of this function.

`cat_all` ’yes’/’no’ indicating if data will be used for cat_all_fn. Only data for 1 categorical data function can be loaded with each run of this function.

`pval_cat` ’yes’/’no’ indicating if data will be used for cat_all_fn. Only data for 1 categorical data function can be loaded with each run of this function.

`cat.names` names of variables to be used in cat_fn and sr_fn

`dir.cat` If import = ’yes’, path to location of excel file for categorical data

`file.name.cat` If import = ’yes’, file name of excel file containing categorical data

`sheet.name.cat` Sheet name containing categorical data

`range.name.cat` Range of cells containing categorical. Can be in format ’a1:b20’ or ’a:b’

`format.cat` ’wide’/’long’ indicating categorical data is in wide or long format

`generic` ’yes’/’no’ indicating if data to be loaded for generic use

`gen.vars.keep` Vector of variables in data to keep

`gen.vars.del` Vector of variables in data to delete

`verbose` TRUE/FALSE TRUE indicates comments will be printed during loading

Details

Function can load continuous or categorical data. Continuous data can be used for comparison of baseline p-values (pval_cont_fn), matching summary stats within a trial (match_fn), matching summary stats in different cohorts (cohort_fn), or comparing means of baseline p-values (anova_fn). Categorical data can be used for comparisons of observed with expected distributions for single variable (cat_fn), for group numbers in trials using simple randomisation (sr_fn), for all variables (cat_all_fn), and for comparison of baseline p-values (pval_cat_fn).

There is one function in development that allows assessment of proportion of final digits in summary statistics (final_digit_fn). This function works using summary statistics but could be adapted to use on raw continuous or categorical data.

Only 1 continuous and/or 1 categorical data set allowed per load to avoid clashes

Data can be imported from a file (import = "yes") or taken from an existing data frame, import = "no"

If loading from an existing data use file.cont and file.cat

If loading from common directory or file, can use dir and file.name rather than more specific dir.cont, dir.cat, file.name.cont, or file.name.cat.

Comments about each indicator: *pval_cont*
loads continuous data for pval_cont_fn, outputs as list of 1 containing named data frame pval_cont_data.

format should be study, variable or var, n, m, s, p. Can be in any order. n = sample size, m = mean, s = standard deviation, p = baseline p value (can omit if not reported)

can be in wide or long format
wide: study, var, n1, n2, n3 ..., m1, m2, m3 ... s1, s2, s3..., p
long: study, var, group, m , s, n , p

group or g or grp required for long format
separators (eg n1 n_1 n.1) are stripped and replaced

match
loads continuous data for match_fn, outputs as list of 1 containing named data frame match_data

remainder is same as for pval_cont above.
only difference between pval_cont and match is that match allows for missing mean or SD whereas pval_cont does not

format should be study, variable or var, n, m, s. Can be in any order. n = sample size, m = mean, s = standard deviation

can be in wide or long format
wide: study, var, n1, n2, m1, m2, s1, s2, p
long: study, var, group, m , s, n

group or g or grp required for long format
separators (eg n1 n_1 n.1) are stripped and replaced

cohort
loads continuous data for cohort_fn, outputs as list of 1 containing named data frame cohort_data

same as pval_cont but allows a lookup variable for variable names

format should be study, variable or var, n, m, s, p. Can be in any order. n = sample size, m = mean, s = standard deviation

can be in wide or long format
wide: study, var, n1, n2, n3 ..., m1, m2, m3 ... s1, s2, s3...
long: study, var, group, m , s, n

group or g or grp required for long format
separators (eg n1 n_1 n.1) are stripped and replaced

lookup table is var_name_final, var_name_orig and allows you to specify a list of all variables names (var_name_orig) from all studies and a lookup table of standardised names (var_name_final) allowing different names in different studies to be standardised

has optional variable 'population' which can be used to subset the data if trials in different populations are reported

anova
loads continuous data for anova_fn, outputs as list of 1 containing named data frame anova_data

same as for pval_cont above but allows for optional value for decimal place

format should be study, variable or var, n, m, s, p. Can be in any order. n = sample size, m = mean, s = standard deviation, d= decimal place of mean (if omitted, this is calculated automatically in anova_fn)

can be in wide or long format
wide: study, var, n1, n2, n3 ..., m1, m2, m3 ... s1, s2, s3..., d
long: study, var, group, m , s, n , d

group or g or grp required for long format
separators (eg n1 n_1 n.1) are stripped and replaced

cat
loads categorical data for cat_fn, outputs as list of 1 containing named data frame cat_data

format should be study, n, v. Can be in any order, n= group size, v= number with characteristic

can be in wide or long format
wide: study, n1, n2, n3 ..., v1, v2, v3...
long: study, group, n, v

group or g or grp required for long format
use cat.names to name variable eg c("n", "v") , c("n", "g") ...
separators (eg n1 n_1 n.1) are stripped and replaced

sr
loads categorial data for sr_fn, outputs as list of 1 containing named data frame sr_data

as for cat but only requires study and n

format should be study, n. n= group size

can be in wide or long format
wide: study, n1, n2, n3 ...
long: study, group, n

group or g or grp required for long format
separators (eg n1 n_1 n.1) are stripped and replaced

cat_all
loads categorical data for cat_all_fn, outputs as list of 1 containing named data frame cat_all_data

format should be study, var or variable, n, N, level, stat, recode, p. Can be in any order, n = number with characteristic, N = group size, p = baseline p value (can omit if not reported), can use "ns" for not significant or "<" or ">" to indicate threshold (eg "<0.05")

optional level - number for level of variable (eg y/n =1,2; high/med/low =1,2,3)
optional recode- for variables with >2 levels to tell how to recode into 2 groups
optional stat: statistical test used for p-value : chisq - Chisquare, chisqc- Chisquare with correction, fisher- Fisher's exact, midp - midp -calculated using two different methods, lr- likelihood ratio, mh - Mantel-Haenszel test

can be in wide or long format
wide study, var, n1, n2, n3, ... N1, N2, N3... p, stat, level, recode
long study, var, group, n, N, p, stat, level, recode

group or g or grp required for long format

if variable has 2 levels, only 1 required, other will be calculated.

separators (eg n1 n_1 n.1) are stripped and replaced

pval_cat
loads categorical data for pval_cat_fn, outputs as list of 1 containing named data frame pval_cat_data

as for cat_all but recode variable is not generated

format should be study, var or variable, n, N, p. Can be in any order, n = number with characteristic, N = group size, p = baseline p value (can omit if not reported), can use "ns" for not significant or "<" or ">" to indicate threshold (eg "<0.05")

optional level - number for level of variable (eg y/n =1,2; high/med/low =1,2,3)
optional stat: statistical test used for p-value : chisq - Chisquare, fisher- Fisher's exact

can be in wide or long format
wide study, var, n1, n2, n3, ... N1, N2, N3... p, stat, level
long study, var, group, n, N, p, stat, level

group or g or grp required for long format

if variable has 2 levels, only 1 required, other will be calculated.

separators (eg n1 n_1 n.1) are stripped and replaced

generic
loads data for use generic use, outputs as list of 1 containing named data frame generic_data

use cont suffixes for file details: dir.cont (or dir), file.name.cont (or file.name), sheet.name,cont, range.name.cont)

format should be study, var or variable, variable names

optional gen.vars.keep = vector of variables to keep
optional gen.vars.del = vector of variables to delete

can be in wide or long format
wide study, var, a1, a2..., b1, b2 ...
long study, var, group, a, b, ...

group or g or grp required for long format

separators (eg n1 n_1 n.1) are stripped and replaced
no data checking or other transformations take place

Value

list containing a named data frame containing data in suitable format for appropriate function as described in Details

Examples

```
# examples of loading data for each function are given in the individual functions.
# Here is one- for pval_cont_fn():

pval_cont_data <- load_clean(import= "no", file.cont = "SI_pvals_cont", pval_cont= "yes",
format.cont = "wide")$pval_cont_data


# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                    mustWork = TRUE)
# delete file name from path
path <- sub("/[^/]+$", "", path)

# load data
pval_cont_data <- load_clean(import= "yes", pval_cont = "yes", dir = path,
     file.name.cont = "reappraised_examples.xlsx", sheet.name.cont = "SI_pvals_cont",
     range.name.cont = "A1:O51", format.cont = "wide")$pval_cont_data
```

match_fn — *Compares proportions of matching summary statistics within two-arm randomised trials*

Description

Creates flextable of matching summary statistics by significant figures with Reference data

Usage

```
match_fn(df = match_data, verbose = TRUE)
```

Arguments

`df` data frame generated from load_clean function

`verbose` TRUE or FALSE indicates whether to print flextable

Details

Reference data is from Bolland 2021
Bolland MJ, Gamble GD, Avenell A, Grey A. Identical summary statistics were uncommon in randomized trials and cohort studies. J Clin Epidemiol 2021;136:180-188.

Returns a list containing 6 objects and (if verbose = TRUE) prints the flextable match_ft_all

Value

list containing 6 objects as described

- match_ft_all = flextable of matches with reference data
- match_ft = flextable of matches (no reference data)
- ref_match_ft = flextable of reference data
- match_match_data = data frame of results used in calculations
- match_table = data frame of matches used to make flextable
- ref_table = data frame of reference data used to make flextable

Examples

```
# load example data
match_data <- load_clean(import= "no", file.cont = "SI_pvals_cont", match= "yes",
format.cont = "wide")$match_data


# run function (takes only a few seconds)
match_fn()$match_ft_all

# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                    mustWork = TRUE)
# delete file name from path
path <- sub("/[^/]+$", "", path)

# load data
match_data <- load_clean(import= "yes", match = "yes", dir = path,
     file.name.cont = "reappraised_examples.xlsx", sheet.name.cont = "SI_pvals_cont",
     range.name.cont = "A:O", format.cont = "wide")$match_data
```

pval_cat_fn *Compares observed and expected distribution of p-values for categorical variables*

Description

Creates plots of calculated p-value distribution and AUC (area under curve)

Usage

```
pval_cat_fn(
  df = pval_cat_data,
  seed = 0,
  sims = -1,
  btsp = 500,
  title = "",
  stat = "chi_midp",
  stat.override = "no",
  fisher.sim = "y",
  fish.n.sims = 10000,
  method = "mix",
  verbose = TRUE
)
```

Arguments

`df`	data frame generated from load_clean function
`seed`	the seed to use for random number generation, default 0 = current date and time. Specify seed to make repeatable.
`sims`	number of simulations, default -1 = function selects based on number of variables.
`btsp`	number of bootstrap repeats used to generate 95% confidence interval around AUC
`title`	optional title for plots
`stat`	statistical test to be used 'chisq', 'fisher', 'midp' or 'midp.epitools' (from epitools package), 'midp.sas' (as calculated in SAS), or combinations -if chisq is not appropriate because expected cells<5, use second test: 'chi_fish', 'chi_midp' or 'chi_midp.epi','chi_midp.sas'
`stat.override`	if 'yes' then test specified in stat will be used rather than values for stat in data frame
`fisher.sim`	"yes" or "no" indicator whether to allow fisher test to simulate p-values for >2*2 tables
`fish.n.sims`	number of simulations to use in Fisher test, default 10,000

`method` 'sm', 'mix', or 'ind'. 'ind' does test on individual data, 'sm' summarises data and then does test on summary data, 'mix' does 'ind' for fisher and 'sm' for others. Duration varies with size of studies, test, and number of simulations. Experiment before running large simulations.

`verbose` TRUE or FALSE indicates whether progress bar and comments show and prints plot

Details

See also Bolland MJ, Gamble GD, Avenell A, Grey A, Lumley T. Baseline P value distributions in randomized trials were uniform for continuous but not categorical variables. J Clin Epidemiol 2019;112:67-76.

Returns a list containing 3 objects and (if verbose = TRUE) prints the plot pval_cat_calculated_pvalues

Value

list containing 3 objects as described

- pval_cat_calculated_pvalues = plots of calculated p-value distribution and AUC
- pval_cat_reported_pvalues = plots of reported p-value distribution and AUC (if p-values were reported)
- all_results = list containing
  - pval_cat_baseline_pvalues_data = data frame of all results used in calculations
  - pval_cat_reported_pvalues= plot of reported p-value distribution
  - pval_cat_auc_reported_pvalues = AUC of reported p-values
  - pval_cat_calculated_pvalues = plot of calculated p-value distribution
  - pval_cat_auc_calculated_pvalues= AUC of calculated p-values

Examples

```
# load example data
pval_cat_data <- load_clean(import= "no", file.cat = "SI_cat_all", pval_cat= "yes",
format.cont = "wide")$pval_cat_data


# run function (takes a few seconds)
pval_cat_fn(seed=10, sims = 50, btsp = 100)$pval_cat_calculated_pvalues

# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                    mustWork = TRUE)
# delete file name from path
path <- sub("/[^/]+$", "", path)

# load data
```

```
pval_cat_data <- load_clean(import= "yes", pval_cat = "yes", dir = path,
     file.name.cat = "reappraised_examples.xlsx", sheet.name.cat = "SI_cat_all",
     range.name.cat = "A:n", format.cat = "wide")$pval_cat_data
```

pval_cont_fn *Compares observed and expected distribution of p-values for continuous variables*

Description

Creates plots of calculated p-value distribution and AUC (area under curve)

Usage

```
pval_cont_fn(df = pval_cont_data, btsp = 500, title = "", verbose = TRUE)
```

Arguments

df	data frame generated from load_clean function
btsp	number of bootstrap repeats used to generate 95% confidence interval around AUC
title	optional title for plots
verbose	TRUE or FALSE indicates whether progress bar and comments show and prints plot

Details

Reference data is from (Carlisle 2017, Bolland 2021)
Carlisle JB . Data fabrication and other reasons for non-random sampling in 5087 randomised, controlled trials in anaesthetic and general medical journals. Anaesthesia 2017;72:944–52 .2017
Bolland MJ, Gamble GD, Grey A, Avenell A. Empirically generated reference proportions for baseline p values from rounded summary statistics. Anaesthesia 2020;75:1685-1687.
See also Bolland MJ, Gamble GD, Avenell A, Grey A, Lumley T. Baseline P value distributions in randomized trials were uniform for continuous but not categorical variables. J Clin Epidemiol 2019;112:67-76.
and Bolland MJ, Gamble GD, Avenell A, Grey A. Rounding, but not randomization method, non-normality, or correlation, affected baseline P-value distributions in randomized trials. J Clin Epidemiol 2019;110:50-62.

Returns a list containing 4 objects and (if verbose = TRUE) prints the plot pval_cont_calculated_pvalues

Value

list containing 4 objects as described

- pval_cont_calculated_pvalues = plots of calculated p-value distribution and AUC
- pval_cont_reported_pvalues = plots of reported p-value distribution and AUC (if p-values were reported)
- pval_cont_ft_diff_calc_rep_p = flextable of distribution of differences in calculated and reported results
- all_results = list containing
  - pval_cont_baseline_pvalues_data = data frame of all results used in calculations
  - pval_cont_diff_calc_rep_p = data frame of differences between calculated and reported p-values
  - pval_cont_reported_pvalues= plot of reported p-value distribution
  - pval_cont_auc_reported_pvalues = AUC of reported p-values
  - pval_cont_calculated_pvalues = plot of calculated p-value distribution
  - pval_cont_auc_calculated_pvalues= AUC of calculated p-values

Examples

```
# load example data
pval_cont_data <- load_clean(import= "no", file.cont = "SI_pvals_cont", pval_cont= "yes",
format.cont = "wide")$pval_cont_data


# run function (takes only a few seconds)
pval_cont_fn(btsp=100)$pval_cont_calculated_pvalues

# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                    mustWork = TRUE)
# delete file name from path
path <- sub("/[^/]+$", "", path)

# load data
pval_cont_data <- load_clean(import= "yes", pval_cont = "yes", dir = path,
     file.name.cont = "reappraised_examples.xlsx", sheet.name.cont = "SI_pvals_cont",
     range.name.cont = "A1:O51", format.cont = "wide")$pval_cont_data
```

SI_cat *Example of 20 observations for categorical analysis*

Description

Sample from Sato/Iwamoto dataset of 35 studies with withdrawal data
see Bolland MJ, Gamble GD, Avenell A, Cooper DJ, Grey A. Participant withdrawals were unusually distributed in randomized trials with integrity concerns: a statistical investigation. J Clin Epidemiol 2021;131:22-29.

Usage

```
SI_cat
```

Format

A data frame with 20 rows and 8 variables:

study study ID

n1, n2, n3 number of participants in each group

w1, w2, w3 number of withdrawals in each group

SI_cat_all *Example of 50 variables from different studies for categorical (cat_all_fn) analysis*

Description

Sample from Sato/Iwamoto dataset of 31 studies with categorical data

Usage

```
SI_cat_all
```

Format

A data frame with 106 rows and 14 variables:

study study ID

var variable

level level of variable

recode value to recode level two if collapsing variable to two levels

levels_no total number of levels for each variable

group number of trial arms

n1, n2, n3 number of participants with characteristic in each group

N1, N2, N3 number of participants in each group

p reported p-value

stat reported statistical test used to calculate p-value

`SI_cohort` *Example of 100 observations for cohort analysis*

Description

Sample from Sato/Iwamoto dataset of 226 baseline variables in 34 cohorts
see Bolland MJ, Gamble GD, Avenell A, Grey A. Identical summary statistics were uncommon in randomized trials and cohort studies. J Clin Epidemiol 2021;136:180-188.

Usage

```
SI_cohort
```

Format

A data frame with 100 rows and 6 variables:

study study ID

var variable name

n number of participants in group

m mean of variable

s sd of variable

group number of group

`SI_pvals_cont` *Example of 50 observations for p-value analysis*

Description

Sample from Sato/Iwamoto dataset of 500 baseline variables in 41 trials
for details see Bolland MJ, Avenell A, Gamble GD, Grey A. Systematic review and statistical analysis of the integrity of 33 randomized controlled trials. Neurology 2016;87:2391-2402.

Usage

```
SI_pvals_cont
```

Format

A data frame with 50 rows and 15 variables:

study study ID

var variable name

n1,n2,n3,n4 size of group 1, 2, 3, 4

m1,m2,m3,m4 mean of variable for group 1, 2, 3, 4

s1,s2,s3,s4 sd of variable for group 1, 2, 3, 4

p reported p-value

`sr_fn` *Compares observed and expected distribution of difference in numbers of participants between groups in two-arm randomised trials*

Description

Creates plot of observed to expected numbers and ratios for differences in numbers of participants between trial groups

Usage

```
sr_fn(
  df = sr_data,
  br = "no",
  block = data.frame(study = "", fbsz = "", n_fb = "", df = ""),
  title = "",
  verbose = TRUE
)
```

Arguments

`df` data frame generated from load_clean function

`br` block randomisation: ’yes’ or ’no’. If ’no’ runs function as if all trials used simple randomisation. If ’yes’ performs simple calculations as if block randomised

`block` an additional option for studies using block randomisation. block is a data frame containing columns named study (for study id); fbsz (for the final block size); n_fb (for number of participants in the final block); df (the difference between groups)

`title` title name for plots (optional)

`verbose` TRUE or FALSE indicates whether progress bar and comments in block randomisation function show and whether to print plot

Details

An example is for Sato and Iwamoto trials in Bolland 2016
Bolland MJ, Avenell A, Gamble GD, Grey A. Systematic review and statistical analysis of the integrity of 33 randomized controlled trials. Neurology 2016;87:2391-2402.

Returns a list containing 4 objects and (if verbose = TRUE) prints the plot sr_graph

Value

list containing 4 objects as described

- sr_graph = plot of observed to expected numbers for differences between numbers of participants in trial groups
- sr_graph = plot of observed to expected numbers and ratios for differences between numbers of participants in trial groups
- sr_individual_graphs = list containing 2 plots making up composite figure
- sr_data = data frame containing data for plots

Examples

```
# load example data
sr_data <- load_clean(import= "no", file.cat = "SI_cat", sr= "yes",
format.cat = "wide")$sr_data


# run function (takes only a few seconds)
sr_fn()$sr_graph

# to import an excel spreadsheet (modify using local path,
# file and sheet name, range, and format):

# get path for example files
path <- system.file("extdata", "reappraised_examples.xlsx", package = "reappraised",
                    mustWork = TRUE)
# delete file name from path
path <- sub("/[^/]+$", "", path)

# load data
sr_data <- load_clean(import= "yes", sr = "yes", dir = path,
   file.name.cat = "reappraised_examples.xlsx", sheet.name.cat = "SI_cat",
   range.name.cat = "A:D", format.cat = "wide")$sr_data

# function has an additional option for block randomisation.
#  If studies are block randomised and the final block size is known,
#  the number of participants in the final block can be determined.
#  The distribution of differences between groups for the final block
#  can be compared to the expected distribution
#
#  Few studies provide all these details so it seems unlikely this function
#  would get used often
# Example takes only a few seconds to run

sr_fn(br = ”yes”, block = data.frame(study = c(1,2,3,4,5,6,7,8,9,10),
         fb_sz= c(2,4,6,8,10,12,8,8,6,14), n_fb = c(1,1,4,5,7,8,4,6,2,10),
         df=c(1,1,0,1,3,4,2,2,0,0)))$sr_graph
```

Index