Package ‘dataepi’

June 2, 2021

Version 0.1

Date 2021-04-17

Title An R Package For Health Dataset

Description A set of functions for automatically generating a draft report from a health dataset in tabular form with a binary variable for the disease and any type of variable (categorical or binary, continuous or discrete) for the exposures which are the factors considered in the study. The package should be used with cautious by checking while completing the obtained output.

Author R. Priam <rpriam@gmail.com>

Maintainer R. Priam <rpriam@gmail.com>

Imports MASS, car, plyr, pracma, pwr, officer, xtable

License GPL-3

RoxygenNote 7.1.1

NeedsCompilation no

R topics documented:

`data_prepare` *A function for preparing the data.frame before the analysis*

Description

This function takes as input a data.frame and the names of the variables in order check the variables and their r types.

Usage

```
data_prepare(
  A,
  var_y = NULL,
  vars_cont = NULL,
  vars_disc = NULL,
  vars_int = NULL,
  var_id = NULL
)
```

Arguments

`A`	The data.frame with the variables to test.
`var_y`	The variable for the disease yes/not.
`vars_cont`	The names of the variables with continuous values.
`vars_disc`	The names of the variables with categorical values.
`vars_int`	The names of the variables with integer (ordered) values.
`var_id`	The name of the variable for the unique identifier per row.

Value

A list with the following entries.

A The data.frame from the dataset after checking and updating.

var_disc_from_cont The names of the discretized variables from continuous ones (not implemented).

vars_disc The vectors of names received from the input parameters.

vars_cont The vectors of names received from the input parameters.

var_y The same name received from the input parameters.

Examples

```
data(DebTrivedi)
A          <- DebTrivedi[,c("age","ofp","gender","region","health")]
A$id       <- 1:nrow(A)
A$hospbin  <- DebTrivedi$hosp>0
vars_cont  <- c("age","ofp")
vars_disc  <- c("gender","region","health")
var_id     <- "id"
var_y      <- "hospbin"
fp = data_prepare(A,var_y,vars_cont,vars_disc,var_id)
print(head(fp$A))
```

data_rename *A function for renaming the modalities of categorical variables*

Description

This function take as input a data.frame and the names of the categorical variables in order to rename the modalities with strings.

Usage

```
data_rename(A, vars_disc_to_recode)
```

Arguments

`A` The data.frame with the variables to rename.

`vars_disc_to_recode`
The vector with the string names of the variables to rename.

Value

A A data.frame for the dataset after renaming the modalities.

dico A list with the correspondences between the old and new characters strings as modalities of the selected categorical variables.

Examples

```
data(DebTrivedi)
A <- DebTrivedi
A$id       <- 1:nrow(A)
A$hospbin  <- as.integer(A$hosp>0)
var_id     = "id"
vars_cont = c("age","ofp","ofnp","opp","opnp","emer","numchron","hosp","school")
vars_disc = c("health","adldiff","region","black","gender","married","employed",
              "privins","medicaid")
vars_int  = NULL
```

```
var_y     = "hospbin" #binary 0/1
label_y   = "hospibin"
A <- data_prepare(A,var_y,vars_cont,vars_disc,var_id)$A
vars_disc_to_recode  <- c("health","gender","region")
resu_     <- data_rename(A, vars_disc_to_recode)
for (nv in vars_disc_to_recode) {
  resu_nv <-  data.frame(resu_$dico[[nv]])
  rownames(resu_nv) <- nv
  print(resu_nv)
}
```

DebTrivedi *Dataset of 4406 individuals aged 66 and over with 19 variables*

Description

Deb and Trivedi (1997) dataset of 4406 individuals, aged 66 and over, who are covered by Medicare, a public insurance program. Originally prepared for an R package accompanying Kleiber and Zeileis (2008) paper and also available as with Zeileis (2006) paper.

Usage

```
data(DebTrivedi)
```

Format

An object of class "data.frame".

Source

JSTAT (http://www.jstatsoft.org/v27/i08/paper)

References

A. Zeileis et al. (2008) Journal of Statistical Software 27(8):1-25.
Deb, P. and Trivedi, P. (1997). Demand for medical care by the elderly: A finite mixture approach. Journal of Applied Econometrics, 12:313-36.

Examples

```
data(DebTrivedi)
head(DebTrivedi)
```

rep_compute — *A function for generating the tables for the report from a data.frame*

Description

This function returns tables from a variable for disease which is binary and a set of variables which are categorical for exposures. The descriptive statistics and statistical tests are computed and aggregated in several data.frames.

Usage

```
rep_compute(A, var_y, vars_x, vars_cont, vars_disc, vars_int, var_id)
```

Arguments

`A`	The data.frame for the analysis.
`var_y`	The variable for the disease yes/not.
`vars_x`	The variables for the table with glm for or and rr.
`vars_cont`	The name of the variables with continuous values.
`vars_disc`	The name of the variables with categorical values.
`vars_int`	The name of the variables with integer (ordered) values.
`var_id`	The name of the variable for the unique identifier per row.

Value

A list with the following entries.

Anew The new matrix A after pre-traitment with data_prepare().

desc_all The result from tab_contents().

desc_cont The result from tab_desc_cont().

desc_disc The result from tab_desc_disc().

desc_biv The result from tab_desc2class_cont().

test_tt The result from tab_tt2classes_cont().

test_anova The result from tab_ttanova_cont().

test_chi2 The result from tab_chi2all().

or The result from tab_all2x2() and stat_oddsratio().

rr The result from tab_all2x2() and stat_relativerisk().

gg The result from tab_glmorr().

fv The result from viz_all2x2().

args The variables at the call of the function.

Examples

```
## Not run:
data(DebTrivedi)
A=DebTrivedi
A$id=1:nrow(A)
A$hospbin  = as.integer(A$hosp>0)
var_id    = "id"
vars_cont = c("age","ofp","ofnp","opp","opnp","emer","numchron","hosp","school")
vars_disc = c("health","adldiff","region","black","gender","married","employed",
              "privins","medicaid")
vars_int  = NULL
var_y     = "hospbin" #binary 0/1
label_y   = "hospibin"
A0 = A
fp = data_prepare(A,var_y,vars_cont,vars_disc,var_id)#,discretize_=TRUE)
A  = fp$A
A$age_3cl = as.character(1*(A$age<7.1)+2*(A$age>=7.1&A$age<7.7)+3*(A$age>=7.7))
A$age_3cl[A$age_3cl=="1"]="[6.6, 7.1)"
A$age_3cl[A$age_3cl=="2"]="[7.1, 7.7)"
A$age_3cl[A$age_3cl=="3"]="[7.7,10.9]"
vars_disc = c(vars_disc,"age_3cl")
vars_x =c(vars_disc,fp$var_disc_from_cont)[c(1:9,10)] #only discrete variables relevant
au = dataepi::rep_compute(A, var_y, vars_x, vars_cont, vars_disc, vars_int, var_id)

## End(Not run)
```

---

rep_write *A function for writing the full report from a data.frame*

Description

This function allows to generate a report from a binary variable for a disease, and a set of categorical variables for the exposures.

Usage

```
rep_write(
  fullpathfile = NULL,
  formatfile = "docx",
  A,
  var_y,
  vars_x,
  vars_cont,
  vars_disc,
  vars_int,
  var_id,
  list_supp = NULL,
  add_ORpca_ = FALSE
)
```

Arguments

`fullpathfile` A character string with the full path for report saving.

`formatfile` A character string for the format of the file with ”doc”,”docx” or ”rtf” for a word document and ”tex” for a latex one.

`A` The data.frame for the analysis.

`var_y` The variable for the disease yes/not.

`vars_x` The variables for the table with glm for or and rr.

`vars_cont` The name of the variables with continuous values.

`vars_disc` The name of the variables with categorical values.

`vars_int` The name of the variables with integer (ordered) values.

`var_id` The name of the variable for the unique identifier per row.

`list_supp` A list with supplementary information for adding to the report, with optional entries.

- **where** A brief descriptive of the place(s) where the study took place.
- **who** A brief descriptive of the population targeted.
- **objective** A brief descriptive of the objectives and purposes.
- **disease** The name of the disease.
- **descriptive** A brief descriptive of the disease.
- **project** For the type of project, for instance ”descriptive”.
- **keywords** A list of key words corresponding to the study or analysis.
- **inex** A descriptive for the criteria for the inclusion and exclusion.
- **topics** A list with terms to classify the variables and each subset of variable names corresponding, for instance, ”biological”, with blood test results, ”socio-demographic” with age, gender, etc.

`add_ORpca_` A boolean variable for including or not including ORpca in the table with odds ratio and relative risk.

Value

A list with following entries.

au The resulting output from rep_compute().

fullpathfile The copy of the variable from the parameters with the same name for the full path for report saving.

note Message to user not null if the file exists already in order to avoid file loss.

Examples

```
## Not run:
data(DebTrivedi)
A=DebTrivedi
A$id=1:nrow(A)
A$hospbin  = as.integer(A$hosp>0)
var_id    = "id"
```

```
vars_cont = c("age","ofp","ofnp","opp","opnp","emer","numchron","hosp","school")
vars_disc = c("health","adldiff","region","black","gender","married","employed",
              "privins","medicaid")
vars_int  = NULL
var_y     = "hospbin" #binary 0/1
label_y   = "hospibin"
A0 = A
fp = data_prepare(A,var_y,vars_cont,vars_disc,var_id)#,discretize_=TRUE)
A  = fp$A
A$age_3cl = as.character(1*(A$age<7.1)+2*(A$age>=7.1&A$age<7.7)+3*(A$age>=7.7))
A$age_3cl[A$age_3cl=="1"]="[6.6, 7.1)"
A$age_3cl[A$age_3cl=="2"]="[7.1, 7.7)"
A$age_3cl[A$age_3cl=="3"]="[7.7,10.9]"
vars_disc = c(vars_disc,"age_3cl")
vars_x =c(vars_disc,fp$var_disc_from_cont)[c(1:9,10)] #only discrete variables relevant
#au = dataepi::rep_compute(A, var_y, vars_x, vars_cont, vars_disc, vars_int, var_id)
wr = dataepi::rep_write("./report_dataepi.docx","docx",
                         A, var_y, vars_x, vars_cont, vars_disc, vars_int, var_id)

## End(Not run)
```

stat_oddsratio *A function for computing the odds ratio from a table for an exposure*

Description

This function computes the odds ratio of a binary variable from the crosstable or contingency table between a disease and an exposure.

Usage

```
stat_oddsratio(X)
```

Arguments

X	The object of type table of size 2x2.

Value

A list with the following entries.

stat The odds ratio from the table X.

SE The standard-deviation of the odds ratio from the table X.

I.95left The left part of the confidence interval at 0.95%.

I.95right The right part of the confidence interval at 0.95%.

name The name of the statistics, ”OR”.

warning A boolean value for or for not having a=0 or b=0 or c=0 or d=0.

a The value a from the 2x2 input table.

b The value b from the 2x2 input table.

c The value c from the 2x2 input table.

d The value d from the 2x2 input table.

table2x2 The table from the input parameter.

Examples

```
X=matrix(c(2534,459,487,142),ncol=2,byrow=TRUE)
X=as.table(X)
colnames(X)<-c("0","1")
rownames(X)<-c("0","1")
print(X)
resu_=stat_oddsratio(X)
cat(resu_$name,"=",round(resu_$stat,2),
paste("(",round(resu_$stdstat,2),")",sep=""),
"\n")
```

`stat_relativerisk` *A function for computing the relative risk from a table for an exposure*

Description

This function computes the relative risk of a binary variable from the crosstable or contingency table between a disease and an exposure.

Usage

```
stat_relativerisk(X)
```

Arguments

`X`	The object of type table of size 2x2.

Value

stat The relative risk from the table X

SE The standard-deviation of the odds ratio from the table X.

I.95left The left part of the confidence interval at 0.95%.

I.95right The right part of the confidence interval at 0.95%.

name The name of the statistics, ”OR”.

warning A boolean value for or for not having a=0 or b=0 or c=0 or d=0.

a The value a from the 2x2 input table.

b The value b from the 2x2 input table.

c The value c from the 2x2 input table.

d The value d from the 2x2 input table.

table2x2 The table from the input parameter.

Examples

```
X=matrix(c(2534,459,487,142),ncol=2,byrow=TRUE)
X=as.table(X)
colnames(X)<-c("0","1")
rownames(X)<-c("0","1")
print(X)
resu_=stat_relativerisk(X)
cat(resu_$name,"=",round(resu_$stat,2),
paste("(",round(resu_$stdstat,2),")",sep=""),
"\n")
```

tab_all2x2 *A function for generating a table with statistics such as odds ratios*

Description

This function returns the whole table with the statistics odd ratio, risk ratio of any other defined by the user from a set of binary variables (exposures) vs one binary variable (disease).

Usage

```
tab_all2x2(A, vars_disc, var_y, stat_f = NULL)
```

Arguments

`A`	The data.frame with the data.
`vars_disc`	The name of the variables with categorical values.
`var_y`	The name of the categorical variable.
`stat_f`	A function such as stat_oddsratio,stat_relativerisk,...

Value

A list with following entries.

tabstat A data.frame with eight columns: the variable name, the modality or level name, the contents of the 2x2 contingency table with A for a, B for b, C for c and D for D, and the corresponding statistics from the function stat_... in the list of parameters, for instance OR and SE_OR for the odds ratios and their standard-deviations.

binarized_1_s The binarized variable for or not the variable equal to modality, in a list of lists.

binarized_0_s The binarized variable for or not the variable not equal to modality, in a list of lists.

tables2x2 The contingency tables for all the variables and all the modalities in the format of a list a lists, from the function stat_f in the parameters.

Examples

```
library(dataepi)
data(DebTrivedi)
A         <- DebTrivedi
A$hospbin <- as.integer(A$hosp>0)
vars_disc <- c("health","gender","region")
var_y     <- "hospbin"
resu_or   <- dataepi::tab_all2x2(A,vars_disc,var_y,dataepi::stat_oddsratio)
print(resu_or$tabstat)
```

`tab_chi2all` *A function for computing the chi2 test from several variables of a data.frame*

Description

This function take as input a data.frame and the names of the categorical variables in order to compute the tests and aggregates them in a data.frame. The resulting data.frame contains the chi-square tests for each variable in vars_disc minus the last which appears in the second column. There are two loops: k in (1;p-1) while l in (k+1;p) in order to compute the upper part to the diagonal.

Usage

```
tab_chi2all(A, vars_disc, pvalue_seuil_ = 0.015)
```

Arguments

`A`	The data.frame with the variables to test.
`vars_disc`	The vector with the names of the variables to test.
`pvalue_seuil_`	The maximal p-value for keeping the pairs of variables.

Value

A list with three entries

tabchi2 The data.frame with the chi2 tests by pairs of variables with the following columns

- **row** The variable from the rows of the contingency table.

- **col** The variable from the cols of the contingency table.
- **nbr** The number of modalities of the first variable.
- **nbc** The number of modalities of the second variable.
- **chi2** The statistics as computed from the chi2 test.
- **df** The number of free parameters in the chi2 test.
- **p.val** The p-value from the chi2 test.
- **mnij** The minimum coun in the cells of the table.
- **p.val.e** The p-value from the exact Fisher test.
- **pow** The power (if available) from the chi2 test.
- **nb** The total number of counts in the table.

pairs_no_pchi2 A data.frame with by rows the pairs of variables with no chi2 test available because of their corresponding contingency table.

pairs_large_pchi2 A data.frame with by rows the pairs of variables with no chi2 test available because their p-value is larger than the threshold.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
vars_disc <- c("health","gender","region")
resu_    <- tab_chi2all(A,vars_disc,0.05)
print(head(resu_$tabchi2))
```

tab_chi2oneall *A function for computing the chi2 test from one against several variables*

Description

This function take as input a data.frame and the names of the categorical variables plus one additional variable in order to compute the tests and aggregates them in a data.frame, without filtering. The tests are computed with a loop on the whole set in vars_disc.

Usage

```
tab_chi2oneall(A, vars_disc, var_y)
```

Arguments

`A`	The data.frame with the variables to test.
`vars_disc`	The vector with the string names of the variables to test.
`var_y`	The string name of one variable.

Value

A data.frame with with the chi2 tests by pairs of variables with the following columns

row The variables from the input vector of variable names vars_disc.

col The variable with its name in the input variable var_y.

nbr The number of modalities of the first variable.

nbc The number of modalities of the second variable.

chi2 The statistics as computed from the chi2 test.

df The number of free parameters in the chi2 test.

p.val The p-value from the chi2 test.

mnij The minimum coun in the cells of the table.

p.val.e The p-value from the exact Fisher test.

pow The power (if available) from the chi2 test.

nb The total number of counts in the table.

Examples

```
data(DebTrivedi)
A          <- DebTrivedi
vars_disc <- c("health","gender","region")
var_y     <- "hosp"
resu_    <- tab_chi2oneall(A,vars_disc,var_y)
print(head(resu_))
print(head(resu_[resu_$p.val<=0.05,]))
```

---

tab_contents — *A function for generating a very simple description of the variables in a data.frame*

Description

This function show the list of variables with the name of their classes of variable (numerical,integer,...) in r and the number of unique values for each variable.

Usage

```
tab_contents(D)
```

Arguments

D	The data.frame with the variables to describe.

Value

A data.frame with each row for a variable from the input data.frame and with the following columns.

variable The variable name from the column names of D.

r_class The type of the variable from the R langage.

nblevels The total number of unique observations.

nbobs The total number of non missing observations.

Warning: with class factor, it may exist empty levels not counted. The function may be considered only after the function data_prepare().

Examples

```
data(DebTrivedi)
A    <- DebTrivedi
A$id <- 1:nrow(A)
resu_<-tab_contents(A)
print(head(resu_))
```

`tab_desc2class_cont` *A function for generating a table with statistics of continuous variables*

Description

This function returns the whole table with the statistics number of observations, mean, standard-error, median, min, max from a set of continuous variables versus one categorical variable with two (or eventually more modalities).

Usage

```
tab_desc2class_cont(A, vars_cont, var_y, nbdigits = 2)
```

Arguments

`A` The data.frame with the data.

`vars_cont` The names of the continuous variables.

`var_y` The name of the categorical variable.

`nbdigits` The number of decimals to keep.

Value

A data.frame with each row for a continuous variable and with the following columns, where ¡modality¿ is one of the modalities of the variable whose name in written in var_y.

MEAN_¡modality¿ The mean of the continuous variable for the modality.

STD_¡modality¿ The standard-deviation of the variable for the modality.

MD_¡modality¿ The median of the continuous variable for the modality.

MIN_¡modality¿ The minimum of the continuous variable for the modality.

MAX_¡modality¿ The maximum of the continuous variable for the modality.

Nnotna__¡modality¿ The number of non missing observation for the modality.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
vars_cont <- c("age","ofp")
resu_    <- tab_desc2class_cont(A,vars_cont,"gender")
print(resu_)
```

`tab_desc_cont` *A function for describing the data.frame for the continuous variables*

Description

This function take as input a data.frame and the names of the continuous variables in order to print in a table the names, plus the median, mean, standard-deviation, minimum, maximum, number of NA, and the number of not NA.

Usage

```
tab_desc_cont(A, vars_cont, nbdigits = 2)
```

Arguments

`A` The data.frame with the variables to test.

`vars_cont` The names of the variables with continuous values.

`nbdigits` The number of decimals to keep.

Value

A data.frame with each row for a continuous variable and with the following columns.

var The name of a variable from the vector of names vars_cont.

median The median of the variable.

mean The mean of the variable.

sd The standard-deviation of the variable.

min The minimum of the variable.

max The maximum of the variable.

nb_na The number of missing values of the variable.

nb The number of non missing values of the variable.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
vars_cont <- c("age","ofp")
resu_    <- tab_desc_cont(A,vars_cont)
print(resu_)
```

`tab_desc_disc` *A function for describing the data.frame for the categorical variables*

Description

This function take as input a data.frame and the names of the categorical variables in order to print in a table the names, frequencies per modalities, percentable per modalities, and the number of levels and number of NA values.

Usage

```
tab_desc_disc(A, vars_disc, nbdigits = 2)
```

Arguments

`A`	The data.frame with the variables to test.
`vars_disc`	The names of the variables with categorical values.
`nbdigits`	The number of decimals to keep.

Value

A data.frame with each row for a categorical variable and with the following columns.

var The name of a variable from the vector of names vars_disc.

nb_na The number of missing values.

nblevel The total number of unique observations.

nbperlevel The numbers of observations by modality.

propperlevel The proportions of observations by modality.

namelevel The corresponding modality names.

Examples

```
data(DebTrivedi)
A          <- DebTrivedi
vars_disc <- c("health","gender","region")
resu_     <- tab_desc_disc(A,vars_disc)
print(resu_)
```

`tab_glmorr` *A function for computing the OR from the logistic regression*

Description

This function take the data.frame with all the variable and construct a new matrix with the odds ratio from a logistic regression, from the full model and from the reduced model after a selection by AIC.

Usage

```
tab_glmorr(A, vars_x, var_y)
```

Arguments

`A`	The data.frame for the analysis.
`vars_x`	The variables for the table with glm for or and rr.
`var_y`	The variable for the disease yes/not.

Value

A list with following entries.

frm The formula for the logistic regression for computing the odds ratio on a full model before selecting the variables.

tabl_coeff_full The resulting table of coefficients regression with all the variables.

tabl_coeff_small The resulting table of coefficients regression with the variable kept after a selection by AIC.

allcoeffs The two tables of coefficients side to side for comparison purpose.

allors The odds ratios from the two tables of coefficients side to side in allcoeffs.

fit_full The r object from the glm regression with all variables.

fit_small The r object from the glm regression with selected variables.

yX The dataframe restricted to the variable in vars_x and var_y.

Examples

```
## Not run:
data(DebTrivedi)
A=DebTrivedi
A$hospbin = as.integer(A$hosp>0)
vars_x = c("health","region","gender","married","employed")
var_y  = "hospbin"
gg     = dataepi::tab_glmorr(A,vars_x,var_y)
print(gg$allors)

## End(Not run)
```

`tab_tt2classes_cont` *A function for generating a table with the t-test of diverse variables*

Description

This function returns the whole table with the t-tests from a set of continuous variables versus one categorical variable with two (or eventually more modalities).

Usage

```
tab_tt2classes_cont(A, vars_cont, var_y)
```

Arguments

`A` The data.frame with the data.

`vars_cont` The names of the continuous variables.

`var_y` The name of the categorical variable.

Value

A data.frame with each row for a continuous variable and with the following columns.

var1 The name of the continuous variable from group 1.

median1 The median of the continuous variable from group 1.

mean1 The mean of the continuous variable from group 1.

sd1 The standard-deviation of the continuous variable from group 1.

nb1 The sample size of group 1.

var2 The name of the continuous variable from group 2.

median2 The median of the continuous variable from group 2.

mean2 The mean of the continuous variable from group 2.

sd2 The standard-deviation of the continuous variable from group 2.

nb2 The sample size of group 2.

T_t.test (2cl) The statistics computed for the t-Student test.

P_t.test (2cl) The p-value computed for the t-Student test.

P¡0.05 Power_t.t The power computed for the t-Student test.

Examples

```
data(DebTrivedi)
A          <- DebTrivedi
vars_cont <- c("age","ofp")
resu_    <- tab_tt2classes_cont(A,vars_cont,"gender")
print(resu_)
```

`tab_ttanova_cont` *A function for generating a table with the anova of diverse variables*

Description

This function returns the whole table with the anova tests from a set of continuous variables and categorical variables with different number of modalities, plus other related tests.

Usage

```
tab_ttanova_cont(A, vars_cont, vars_disc)
```

Arguments

`A`	The data.frame with the data.
`vars_cont`	The names of the continuous variables.
`vars_disc`	The names of the categorical variables.

Value

A list with the following entries.

tabstat A data.frame with each row for a continuous variable and a categorical variable (binary or poly) with the following columns.

var_cont The name of the continuous variable.

var_disc The name of the discrete variable.

n1 The size of groupe 1.

n2 The size of groupe 2.

... ...

ng The size of groupe g (if it exists for the variable in var_disc).

norm1 The p-value of the normality test from group 1.

norm2 The p-value of the normality test from group 2.

... ...

normg The p-value of the normality test from group g (if exists).

test12_vr The p-value of the equality test of the variances for two groups.

test12_eq The p-value of the equality t-test of the means for two groups.

test12_gt The p-value of the equality t-test of the means for two groups with option "greater".

test12_ls The p-value of the equality t-test of the means for two groups with option "less".

test12_wx_eq The p-value of the wilkox rank sum (not paired) equality test of the means for two groups.

test12_wx_gq The p-value of the wilkox rank sum (not paired) equality test of the means for two groups with option "greater".

test12_wx_ls The p-value of the wilkox rank sum (not paired) equality test of the means for two groups with option "less".

test_vr The p-value of the equality test of the variances for g¿2 groups.

test_aov The p-value of the equality test of the means for g¿2 groups.

test_aov_check The p-value from the normality test of the residual from the anova test for g¿2 groups.

test_welch The p-value from the welch test for g¿=2 groups and variances not equal, under normality and variances supposed not equal.

test_krusk The p-value from the kruskal-wallis rank sum test for g¿=2 groups to check the equality in distribution of the means.

pairs_no_panova A list of pairs of variables with non available anova test.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
vars_cont <- c("age","ofp")
vars_disc <- c("gender","region","health")
resu_    <- tab_ttanova_cont(A,vars_cont,vars_disc)
print(resu_)
```

viz_all2x2 *A function for showing the modalities in a 2d view plus a related indicator*

Description

This function take the set of 2x2 table for the odds ratio in order to construct a new matrix of the whole set of modalities and performs pca. The obtained projection is visualized and an indicator is also obtained.

Usage

```
viz_all2x2(TABSTAT, g_ = 3, graph_ = "PCA")
```

Arguments

TABSTAT	The table from the list of 2x2 tables from the data.
g_	The number of groups for the clustering.
graph_	The characters string with null value for no projection, with the value to show the "PCA", the value "COSTAT" to show the percentages for Sick=1, and the value "CASTAT" to show the percentages for the Sick=0.

Value

A list with the following entries.

tabstat A data.frame with in the first left columns the matrix from the tables 2x2 aggregated from the function tab_all2x2() after normalization of its pairs of columns, followed by two columns, one for the variable name and one for the modality name.

pca The r object from a pca of the normalized matrix of counts.

kmeans The r object from a kmeans of the normalized matrix of counts.

ORpca An alternative indicator for odds ratios.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
A$hospbin <- as.integer(A$hosp>0)
vars_disc <- c("health","gender","region")
var_y     <- "hospbin"
resu_or   <- dataepi::tab_all2x2(A,vars_disc,var_y,dataepi::stat_oddsratio)
resu_fv   <- viz_all2x2(resu_or$tabstat,g_=7,graph_="PCA")
print(resu_fv$tabstat)
```

Index