Package 'mStats'

March 31, 2020

Type Package

Title Epidemiological Data Analysis

Version 3.2.2

Maintainer Myo Minn Oo <dr.myominnoo@gmail.com>

Description This is a tool for epidemiologist, medical data analyst, medical or public health professionals. It contains three domains of functions: functions for 1) data management, 2) statistical analysis and 3) calculating epidemiological measures. The calculations are mainly based on three books, 1) Betty R. K, (2006, ISBN:978–0–86542–871–3), 2) B. Burt Gerstman (2013, ISBN:978-1-4443-3608-5) and 3) Douglas G Altman (2005, ISBN:0 7279 1375 1).

License GPL-2 | GPL-3

URL https://myominnoo.github.io/

BugReports https://github.com/myominnoo/mStats/issues

Depends R (>= 3.0.0)

Imports stats, utils

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Myo Minn Oo [aut, cre] (<https://orcid.org/0000-0003-4089-016X>)

Repository CRAN

Date/Publication 2020-03-31 10:00:09 UTC

R topics documented:

`addDashLines` *Add dash lines to* `data.frame`

Description

This adds dash lines to the dataframe in the first and last rows.

Usage

```
addDashLines(.df, .vLine = 0)
```

Arguments

`.df`	Dataset
`.vLine`	positional index of Vertical Line `|` in dataset

Value

modifided `data.frame`

---

append *Append multiple datasets*

Description

`append()` joins multiple datasets of the same column names.

Usage

```
append(data, ...)
```

Arguments

`data`	Base dataset
`...`	Datasets to be appended

Details

Multiple datasets can be appended only if they have the same variables in the same order.

```
append(data, data1, data2, data3, etc)
```

Value

Modified Dataset

Author(s)

For any feedback, please contact `Myo Minn Oo` via:

Email: `<dr.myominnoo@gmail.com>`

Website: https://myominnoo.github.io/

Examples

```
## use infert dataset
data(infert)
codebook(infert)

infert.new <- append(infert, infert)
codebook(infert.new)
```

codebook *Codebook: Detailed information about variables*

Description

`codebook()` provides detailed information about each variable within the dataset.

Usage

```
codebook(data)
```

Arguments

`data`	dataframe

Details

`codebook` generates the report of data structure with names, data lables, types, number of observations, number of observations with missing values and percentage of observations with missing values.

ANNOTATIONS:

Variable - Names of variables

Label - Labels of variables

Type - Types of variables

_Obs - Counts of valid observations

_NA - Counts of observations with missing value

_NA(%) - Percentage of observations with missing value

Value

Codebook in data.frame format

Author(s)

For any feedback, please contact `Myo Minn Oo` via:

Email: `<dr.myominnoo@gmail.com>`

Website: https://myominnoo.github.io/

Examples

```
## use infert data
data(infert)
codebook(infert)

## add labels
infert.new <- labelVar(infert,
                       c(education, age, parity, induced, case, spontaneous,
                         stratum, pooled.stratum),
                       c("Education", "Age in years of case", "Count",
                         "# of prior induced abortions", "case status",
                         "# of prior spon. abortions",
                         "Matched set number", "Stratum Number"))
infert.new <- labelData(infert.new,
                        "Infertility after Spontaneous and Induced Abortion")
codebook(infert.new)
```

`diagTest` *Statistics of Diagnostic Tests*

Description

`diagTest` reports statistics of diagnostic tests

Usage

```
diagTest(x, y = NULL, p = NULL, rnd = 2, print.table = TRUE)

## Default S3 method:
diagTest(...)

## S3 method for class 'table'
diagTest(x, y = NULL, p = NULL, rnd = 2, print.table = TRUE)

## S3 method for class 'numeric'
diagTest(x, y = NULL, p = NULL, rnd = 2, print.table = TRUE)
```

Arguments

`x`	a factor object or a table
`y`	an optional factor object
`p`	disease prevalence
`rnd`	specify rounding of numbers. See `round`.
`print.table`	logical value to display formatted outputs
`...`	optional arguments

Details

The screening tests are based on Bayes' Theorem. These tests help clinicians to correctly predict the presence or absence of a particular disease from the knowlege of test results (positive or negative) and/or the status of presenting symptoms (present or absent) or information regarding the likelihood of positive and negative test results and the likelihood of the presence or absence of a particular symptom in patients with and without a particular disease.

Statistics of diagnostic tests:

Sensitivity: True Positive (TP) rate among diseased (D+) = TP / D+

Specificity: True Negative (TN) rate among non-diseased (D-) = TN / D-

Positive predictive value (PPV): probability of D+ when all positives (AP) = D+ / AP

Negative predictive value (NPV): probability of D- when all negatives (AN) = D+ / AN

Likelihood ratio (LR) = To summarize information about a diagnostic test LR of positive result (LR+) = (Sensitivity) / (1 - Specificity) LR of negative result (LR-) = (1 - Sensitivity) / (Specificity)

Calculating confidence intervals:

95% Confidence intervals were calculated by using Wilson Score Interval as well as its continuity correction. The method was developed by Edwin Bidwell Wilson in 1927. This interval has good properties even for a small sample. Just lik Pearson's chi-squared test and Yates' continuity correction.

Several other methods have been developed such as Clopper–Pearson interval and Agresti–Coull interval. But Wilson score interval methods (with or without continuity correction) have been shown to be the most accurate and the most robust. For further details, see references.

Reference:

1. Biostatistics A Foundation for Analysis in the Health Sciences (10th Edition). Chapter 3.5 BAYES' THEOREM, SCREENING TESTS, SENSITIVITY, SPECIFICITY
2. Avijit Hazra, Nithya Gogtay. Biostatistics Series Module 7: The Statistics of Diagnostic Tests. Indian J Dermatol. 2017 Jan-Feb; 62(1): 18-24. doi: 10.4103/0019-5154.198047
3. Brown, Lawrence D.; Cai, T. Tony; DasGupta, Anirban (2001). "Interval Estimation for a Binomial Proportion". Statistical Science. 16 (2): 101–133.
4. Wallis, Sean A. (2013). "Binomial confidence intervals and contingency tests: mathematical fundamentals and the evaluation of alternative methods" .Journal of Quantitative Linguistics. 20 (3): 178–208.
5. Newcombe, R. G. (1998). "Two-sided confidence intervals for the single proportion: comparison of seven methods". Statistics in Medicine. 17 (8): 857–872. doi:10.1002/(SICI)1097-0258(19980430)17:8<857::AID-SIM777>3.0.CO;2-E. PMID 9595616.

Author(s)

For any feedback, please contact `Myo Minn Oo` via:

Email: `<dr.myominnoo@gmail.com>`

Website: https://myominnoo.github.io/

Examples

```
#### Biostatistics A Foundation for Analysis in the Health Sciences (10th Edition)
# numbers taken from Example 3.5.1

diagTest(as.table(matrix(c(436, 5, 14, 495), ncol = 2, byrow = TRUE)))
diagTest(as.table(matrix(c(436, 5, 14, 495), ncol = 2, byrow = TRUE)),
         p = .113)

#### Avijit Hazra, Nithya Gogtay. Biostatistics Series Module 7: The
#### Statistics of Diagnostic Tests. Indian J Dermatol. 2017 Jan-Feb; 62(1):
#### 18-24. doi: 10.4103/0019-5154.198047

diagTest(as.table(matrix(c(2, 18, 1, 182), ncol = 2, byrow = TRUE)))
diagTest(as.table(matrix(c(2, 18, 1, 182), ncol = 2, byrow = TRUE)),
         p = .10)
diagTest(as.table(matrix(c(30, 35, 23, 12), ncol = 2, byrow = TRUE)))
diagTest(as.table(matrix(c(30, 35, 23, 12), ncol = 2, byrow = TRUE)),
         p = .10)

#### Just an example to demonstrate
# diagTest(infert$case, infert$spontaneous)
```

---

duplicates *Report and tag duplicated observations*

Description

`duplicates()` reports duplications and creates indexes.

Usage

```
duplicates(data, ..., drop = FALSE)
```

Arguments

`data`	Dataset
`...`	Variables to find duplications.
`drop`	if `TRUE`, delete all duplicated records, keeping all unique. If not specified, all variables are used.

Details

Specified variables are used to search for duplications. If not specified, all variables are used.

Then they are pasted as a character vector for speedy operation, extract duplication data and make a report.

The return dataset is added a new variable called dup for further use.

ANNOTATIONS:

`Copies` - nth Copies

`Observations` - Number of corresponding observations

`Surplus` - Number of surplus observations

`dup` - indicates copies within the dataset:

0 = unique observations

2 = duplicated two times

3 = duplicated three times and so on ...

Value

Modified dataset with additional variable `dup`

Author(s)

For any feedback, please contact `Myo Minn Oo` via:

Email: `<dr.myominnoo@gmail.com>`

Website: https://myominnoo.github.io/

Examples

```
## use infert data
data(infert)
codebook(infert)

## find duplicates by pooled.stratum
duplicates(infert, pooled.stratum)

## find duplicates by stratum and pooled.stratum
duplicates(infert, stratum, pooled.stratum)

## find and remove duplicates by pooled.stratum
duplicates(infert, pooled.stratum, drop = TRUE)
```

`egen` *An extension to* `generate`

Description

`egen()` transforms a numeric vector to a factor vector.

Usage

```
egen(data, var, cut = NULL, new_var = NULL, lbl = NULL)
```

Arguments

data	Dataset
var	Exisitng variable
cut	either a single number or a numeric vector.
new_var	name of new variable to be generated
lbl	labels to specify

Details

egen allows easy conversion of a numerical variable to a categorical variable.

Cut-off Intervals

If the interval is not specified, it is cut at an interval of 10, starting from the minimum value. Otherwise, it is divided into corresponding intervals by specified cut-off points.

Automatic Labelling

If lbl is not specified, labels are constructed in this format: lbl[##-##]

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## use infert data

data(infert)
codebook(infert)
summ(infert, age)


## transform continuous to category
infert.new <- egen(infert, age, c(20, 30, 40, 50))
tab(infert.new, age.cat)

## specifiy labels and name
infert.new <- egen(infert, age, c(20, 30, 40), "age_cat", c("<30", "30-39", "40+"))
tab(infert.new, age_cat)
```

enquos *Get names within three dots in a function*

Description

enquos retrieves names of arguments from as.list(match.call()) and returns names within three dots ....

checkEnquos() checks colon separators as well as variables whether they are specified or not. if not, it gets all variables within the dataset.

Usage

```
enquos(.args, .argsName)

checkEnquos(.data, .vars, .types = "tab")
```

Arguments

.args	argument lists from as.list(match.call())
.argsName	Names of arguments to be removed.
.data	Dataset
.vars	.vars obtained with as.character(enquos())
.types	"tab" to get categorical data; otherwise, numerical data

Value

A character vector

Functions

- checkEnquos:

esttab *Tabulate estimates of regression models for comparison*

Description

esttab() produces publication quality tables which can be exported to .csv format.

Usage

```
esttab(...)
```

Arguments

...	regression models of mStats's functions

Details

It is an extention to functions like regress and logistic. It processes their outputs and generates a publication quality tabulation for model comparisons.

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## use infert data
data(infert)
codebook(infert)

## running different logistic regression models
m1 <- logistic(infert, case, education, odds_ratio = FALSE)
m2 <- logistic(infert, case, education, age, odds_ratio = FALSE)
m3 <- logistic(infert, case, education, age, parity, odds_ratio = FALSE)
m4 <- logistic(infert, case, education, age, parity, induced, odds_ratio = FALSE)
m5 <- logistic(infert, case, education, age, parity, induced, spontaneous,
               odds_ratio = FALSE)
e <- esttab(m1, m2, m3, m4, m5)
```

---

expand2 *Duplicate observations within a dataframe*

Description

expand2 generates duplicated observations within a dataframe.

Usage

```
expand2(data, n_n = NULL, copies = 2, original = TRUE)
```

Arguments

data	a data frame object
n_n	index or indexes specifying row numbers
copies	desired number of copies
original	a logical indicating whether to keep the original dataframe

Details

expand2 appends observations from the dataframe with n copies of the observations with specified indexes of observations or all data.

Value

data.frame

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## use infert data
data(infert)
codebook(infert)

## create duplicates
infert.new <- expand2(infert, 1:5, copies = 2)
codebook(infert.new)


## check duplicates report and rmeove dup
infert.dupremove <- duplicates(infert.new, drop = TRUE)
codebook(infert.dupremove)

## remove only 3 copies
infert.3copies <- duplicates(infert.new)
tab(infert.3copies, dup)

infert.3copies <- filter(infert.3copies, dup < 2)
codebook(infert.3copies)
```

---

expandTables *Convert* 2x2 tables *to* data.frame

Description

expandTables() generates a data frame and supports two levels.

Usage

```
expandTables(
  ...,
  exp_name = "exposure",
  exp_lvl = c("exposed", "unexposed"),
  case_name = "outcome",
  case_lvl = c("disease", "healthy"),
  strata_name = "strata",
  stringsAsFactors = FALSE
)
```

Arguments

`...`	vectors of 2x2 tables
`exp_name`	Name of `Exposure` Variable
`exp_lvl`	Names of two categories in the order of Exposed and non-exposed
`case_name`	Name of `Case` variable
`case_lvl`	names of two categories in the order of
`strata_name`	Name of stratified variable
`stringsAsFactors`	`TRUE` or `FALSE` If `TRUE`, character vector is converted to a factor when `data.frame` is constructed.

Details

`expandTables` uses the vectors of `2x2` tables and generates a data frame of at least two columns: Exposure and Outcome.

```
expandTables(c(100, 200, 100, 200))
```

Strata

Strata can be included using multiple named vectors. The generated tables can be used for further calculation such as Mantel Haenszel methods.

```
expandTables(
             strata1 = c(100, 200, 100, 200),
             strata2 = c(100, 200, 100, 200),
             strata3 = c(100, 200, 100, 200),
             exp_name = "Exposure",
             exp_lvl = c("exposed", "unexposed"),
             case_name = "Outcome",
             case_lvl = c("case", "control"),
             strata_name = "Strata"
)
```

Labels

If variables' names or levels are not specified, the followings are applied.

1. Exposure Name: `exposure`
2. Exposure levels: `exposed` and `unexposed`
3. Outcome Name: `outcome`
4. Outcome levels: `disease` and `healthy`
5. Strata Name: `strata`
6. Note: Strata levels are not considered as vectors must be named.

Value

`data.frame`

Author(s)

For any feedback, please contact `Myo Minn Oo` via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## Asthma Example from Essential Medical Statistics
## page 160

asthma <- expandTables(c(81, 995, 57, 867),
              exp_name = "sex",
              exp_lvl = c("woman", "man"),
              case_name = "asthma",
              case_lvl = c("yes", "no"))

## label variable and dataset
asthma <- labelData(asthma, "Hypothetical Data of Asthma Prevalence")
asthma <- labelVar(asthma, c(sex, asthma), c("Man or Woman", "Asthma or No Asthma"))

## Checking codebook
codebook(asthma)


## simple tabulation
tab(asthma)

## cross-tabulation
tab(asthma, sex, by = asthma)
```

export *Export outputs*

Description

`export()` writes to `.csv` file

Usage

```
export(.data, file = "", append = FALSE)
```

Arguments

`.data`	Outputs as `data.frame` or `list`
`file`	file name
`append`	logical. If `TRUE`, the output is appended to the file. If `FALSE`, any existing file of the name is destroyed.

Details

Outputs from the following functions can be exported.

`tab`

`summ`

`tabOdds`

`tabRisks`

`mhor`

`mhrr`

`logistic`

`regress`

`strate`

Author(s)

For any feedback, please contact `Myo Minn Oo` via:

Email: `<dr.myominnoo@gmail.com>`

Website: https://myominnoo.github.io/

formatDate *Format Dates*

Description

formatDate converts characters or numbers to dates. is.Date indicates which elements are Dates.

Usage

```
formatDate(x, format = "dmY", sep = "/", century = NULL)

is.Date(x)

year(x)

month(x)

day(x)
```

Arguments

x	a character or numeric object
format	only for character vectors:
sep	separator character for date components
century	specify either 2000 or 1900 for two-digit years

Details

dmy represents dd mm YYYY format. In combination with separators from sep, this can change to several date formats. For example, dmy + - convert to dd-mm-yyyy format.

Possible conversions

1. dmy + - »> dd-mm-yyyy
2. mdy + - »> mm-dd-yyyy
3. ymd + - »> yyyy-mm-dd
4. dmy + / »> dd/mm/yyyy
5. mdy + / »> mm/dd/yyyy
6. ymd + / »> yyyy/mm/dd

Numeric conversions

Origin is set at 1899-12-30.

See as.Date for more date formats.

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## convert strings to dates
x <- c("2019-01-15", "2019-01-20", "2019-01-21", "2019-01-22")

# check if it is a Date format
is.Date(x)

y <- formatDate(x, "Ymd", "-")

# check if it is a Date format
is.Date(y)
y



## convert numbers to dates
x <- 42705:42710
y <- formatDate(x)
is.Date(y)
y


## get day, month or year
day(y)
month(y)
year(y)
```

---

generate *Create a new variable*

Description

`generate()` creates a new variable within the data frame

Usage

```
generate(data, var, .expr = NULL)
```

Arguments

data	Dataset
var	New variable
.expr	value or Expression for simple arithmetic or logical operations: See examples.

Details

If the data is specified, it returns modified dataset with newly created variable. The value of the variable are specified by expr argument.

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## use infert data
data(infert)
codebook(infert)

## create a new variable induced2 with 0 and 1+2
infert.new <- generate(infert, induced2, as.numeric(induced == 0))
tab(infert.new, induced2)

## create a new variable with values the same as education
infert.new <- generate(infert, edu, education)
tab(infert.new, edu)
```

---

keep Keep, drop *and* filter *variables;* arrange *observations*

Description

keep() or drop() variables within dataframe

arrange() sorts variables or columns

filter() keeps observations that satisfy specified conditions

Usage

```
keep(data, ...)

drop(data, ...)

arrange(data, ...)

filter(data, ...)
```

Arguments

data	Dataset
...	Variables or conditions in filter()

Value

Modified Dataset

Select Variables

keep() removes unspecified variables from the dataset. Variables are also rearranged based on the order they are mentioned.

```
keep(data, var1, var2, var3, etc)
```

drop() works the opposite of keep() removed specified variables from the dataset.

```
drop(data, var5, var6, var7, etc)
```

Sort

arrange() sorts the dataset by specifying variables. The minus symbol - indicates descending order.

```
arrange(data, var1, var2, -var3, var4)
```

Filter

filter keeps observations that meet the specified conditions. If conditions are specified by comma, they are joined by AND operators

```
filter(data, var1 == var2 | var3 > var4)

filter(data, var1 == var2, var3 > var4)
```

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## use infert data
data(infert)
codebook(infert)

## DEMONSTRATION: KEEP
## suppose we want to keep education, age, induced and case
infert.new <- keep(infert, education, age, induced, case)
codebook(infert.new)

## use colon separator :
infert.new <- keep(infert, education:case)
codebook(infert.new)

## change the order of variables
infert.new <- keep(infert, stratum, pooled.stratum, education:spontaneous)
codebook(infert.new)
```

`labelVar` *Add or replace labels of variables and dataset*

Description

`labelVar` labels `variables`.

`labelData` labels `Dataset`.

Usage

```
labelVar(data, var, lbl = NULL)

labelData(data, lbl = NULL)
```

Arguments

`data`	dataset
`var`	one variable or variables
`lbl`	to specify labels. If not specified, label is removed

Details

Labels are useful to provide more detailed information about variables or dataset. Many functions in `mStats` package extract label information and display them as footnote.

Label single or multiple variables

Single or multiple variables can be labelled.

Example: single variable

```
labelVar(data, var, lbl)
```

Example: multiple variable

```
labelVar(data,
      c(var1, var2, var3),
      c(lbl1, lbl2, lbl3))
```

Label Dataset

Dataset can also be labelled and displayed in codebook.

```
labelData(data, lbl)
```

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## use infert data
data(infert)

## label education
infert.new <- labelVar(infert, education, "patient's education")
codebook(infert.new)

## label multiple variables
infert.new <- labelVar(infert, c(education, age, case),
                       c("patient's education", "age in years", "case status"))
codebook(infert.new)


## label dataset
infert.new <- labelData(infert.new,
                        "Infertility after Spontaneous and Induced Abortion")
codebook(infert.new)
```

lagRows *Create lagged variables*

Description

lagRows() creates lagged version of existing variables

Usage

```
lagRows(data, var, by = NULL, lag_var = NULL)
```

Arguments

data	Dataset
var	Existing variable
by	If specified, lagged variable is created grouped by this variable. If not specified, it is created taking the whole variable.
lag_var	name for lagged variable

Details

This is often encountered in time-related analysis. In a lagged variable, values from earlier points in time are placed in later rows of dataset.

Value

Modified Dataset

Note

Before using lagRows, the dataset needs to be sorted by a id variable or similar variable.

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## create a dataset with dates
df <- data.frame(
    hospid = 1:100,
    docid = round(runif(100, 1, 10)),
    dis_date = formatDate(runif(100, 42700, 42800))
```

```
)

## lagged dis_date, not specifed "by"
lagRows(df, dis_date)

## lagged dis_date by docid
## first we need to sort
df1 <- arrange(df, docid)
df1
lagRows(df1, dis_date, by = docid, lag_var = "lag_date")
```

`leftJoin` *Join two or more datasets*

Description

`leftJoin()` merges two or multiple datasets sharing common variables and keeping all rows from x or master.

Usage

```
leftJoin(data, ..., by)
```

Arguments

`data`	master dataset
`...`	mergers or datasets to merge into master dataset
`by`	common variables

Details

The join keeps all rows or observations in master dataset with matched observations from mergers. It adds one more variable `merge_` to the resulting dataset. The value `1` of `merge_` indicates the rows are from master dataset and for 2, rows from merger dataset and 3 is matched observations. In leftJoin, there can be both 1 or 3 in the return dataset.

Value

Modified dataset in `data.frame`

Displaying notes

The notes are displayed in a fashion to inform the user what has been joined or not joined. This provides useful insights into one's own data. This is inspired by `STATA`.

Note

For tibble data format, the return dataset from Join operation results in data.framesince the function is based on merge.

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## set seed
set.seed(123)
## first, create a patient dataset
patient <- data.frame(
    hospid = 1:100,
    docid = round(runif(100, 1, 15)),
    sex = runif(100, 1, 2),
    age = runif(100, 30, 60)
)

## now create a doctor dataset
doc <- data.frame(
   docid = c(1:10, 21:25),
   rating = round(runif(15, 1, 5))
)

## left join the two dataset
leftJoin(patient, doc, by = "docid")

## there are 36 records not matched, 31 not matched from master dataset,
## 5 not matched from merger dataset. 69 Final matched records
```

---

logistic *Logistic Regression Models*

Description

logistic() produces regression outputs which mirror outputs from STATA.

Usage

```
logistic(data, y, ..., odds_ratio = TRUE, rnd = 3)
```

Arguments

data	Dataset
y	Dependent variable
...	Independent variable or multiple variables
odds_ratio	if TRUE, odds ratios, exponentiated coefficients are calculated. Otherwise, coefficients are estimated.
rnd	specify rounding of numbers. See round.

Details

logistic is based on glm with binomial family. All statistics presented in the function's output are derivates of glm, except AIC value which is obtained from AIC.

Outputs

Outputs can be divided into three parts.

1. Information about the model

Here provides number of observations (Obs.), chi value from Likelihood Ratio test (LR chi2) and its degree of freedom, p-value from LR test, Pseudo R Squared, log likelihood and AIC values.

1. Regression Output

Coefficients from summary of model are tabulated here along with 95\ confidence interval.

Value

A list of two data.frame and model

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## use infert data
data(infert)

## run logistic regression
logistic(infert, case, induced, spontaneous)

## get coefficient instead of odds ratio
logistic(infert, case, induced, spontaneous, odds_ratio = FALSE)
```

mhor *Calculating Odds Ratios*

Description

`mhor()` calculates odds ratios, Mantel Haenszel pooled estimates and 95% CI.

Usage

```
mhor(
  data,
  ...,
  by,
  strata = NULL,
  exp_value = NULL,
  case_value = NULL,
  plot = TRUE,
  na.rm = FALSE,
  rnd = 3
)
```

Arguments

`data`	Dataset
`...`	Variable or multiple variables Colon separator : can be used to specify multiple variables.
`by`	Varaiable for cross-tabulation
`strata`	Variable for stratification
`exp_value`	value for exposure as reference
`case_value`	value for outcome as reference
`plot`	logical value to display plots of odds ratios including MH estimates across a categorical variable
`na.rm`	A logical value to specify missing values, `NA` in the table
`rnd`	specify rounding of numbers. See `round`.

Details

Value can be set as baseline by specifying `exp_value`. This is used when the exposed and case values are not in the right place.

It produces a table with Odds Ratio,95% CI as well as p-value. If `strata` is specified, `Mantel-Haenzsel` Pooled estimates of Odds Ratio is generated along with Chi-sqaured test for heterogeneity.

Odds Ratio, OR

$$OR = (D1xH0)/(D0xH1)$$

Error Factor, EF using Woolf's formula

$$95\%CI = OR/EForORxEF$$

$$EF = exp(1.96xSE(log(OR)))$$

$$SE(log(OR)) = \sqrt{1/D1 + 1/H1 + 1/D0 + 1/H0}$$

Calculating p-value from Wald's z test

$$z = logOR/SE(logOR)$$

Mantel-Haenszel's OR

$$ORMH = Q/R$$

$$Q = \sum (D1ixH0i)/ni$$

$$R = \sum (D0ixH1i)/ni$$

Calculating CI for MH-OR

$$95\%CI = OR/EForORxEF$$

$$SE(ORMH) = \sqrt{V/(QxR)}$$

$$V = \sum (DixHixn0ixn1i)/((ni)^2x(ni-1))$$

Chi-square test for MHOR, df = 1

$$X^2(MH), Chi - squarevalue = U^2/V$$

$$U = O - E$$

$$O = \sum D1i$$

$$E = \sum Dixn1i/ni$$

Chi-square test for Heterogeneity

$$X^2 = \sum (D1ixH0i - ORMHxD0ixH1i)^2/ORMHxVixni^2$$

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

References

1. Betty R. Kirkwood, Jonathan A.C. Sterne (2006, ISBN:978–0–86542–871–3)
2. B. Burt Gerstman (2013, ISBN:978-1-4443-3608-5)
3. Douglas G Altman (2005, ISBN:0 7279 1375 1)

Examples

```
### Example from Essential Medical Statistics
# Page 178, Chapter 18: Controlling for confounding: Stratification
lepto <- expandTables(
    male = c(36, 14, 50, 50), female = c(24, 126, 10, 90),
    exp_name = "area", exp_lvl = c("Rural", "Urban"),
    case_name = "ab", case_lvl = c("Yes", "No"),
    strata_name = "gender"
)

## label variables and data
lepto <- labelData(lepto, "Prevalence survey of leptospirosis in West Indies")
lepto <- labelVar(lepto,
              c(area, ab, gender),
              c("Type of area", "Leptospirosis Antibodies",
                "Gender: Male or female"))

## check dataset
codebook(lepto)

## Calculate OR
mhor(lepto, area, by = ab, case_value = "Yes")
```

---

mhrr *Calculating measures of Risk including Risk Ratio*

Description

`mhrr()` calculates different measures of risk including risk ratios (RR) as well as Mantel-Haenszel pooled estimates.

Usage

```
mhrr(
  data,
  ...,
  by,
  strata = NULL,
  exp_value = NULL,
  case_value = NULL,
  measure = "ratio",
  plot = TRUE,
  na.rm = FALSE,
  rnd = 3
)
```

Arguments

data	Dataset
...	Variable or multiple variables Colon separator : can be used to specify multiple variables.
by	Varaiable for cross-tabulation
strata	Variable for stratification
exp_value	value for exposure as reference
case_value	value for outcome as reference
measure	choose different measures related to risks. This is not case-sensitive. By default, measure is RATIO and calculates rate ratio. For Risk Ratio; ratio = Risk Ratio diff = Risk Diff. (Risk Difference) ar = Attributable Risk, specifically Attributable Risk Percent or Proportional Attributable Risk (This is not converted to Percent scale) par = Pop. AR (Population Attributable Risk) efficacy = Efficacy of a certain treatment or exposure. For Mantel-Haenszel estimates, ratio = Risk Ratio diff = Risk Diff. (Risk Difference)
plot	logical value to display plots of risk ratios or differences including MH estimates across a categorical variable
na.rm	A logical value to specify missing values, NA in the table
rnd	specify rounding of numbers. See round.

Details

Value can be set as baseline by specifying exp_value. This is used when the exposed and case values are not in the right place.

It produces a table with Odds Ratio,95% CI as well as p-value. If `strata` is specified, `Mantel-Haenzsel` Pooled estimates of Odds Ratio is generated along with Chi-sqaured test for heterogeneity.

The following entails formulas used for calculating measures.

$$Riskamongexposed, R1 = A/(A + C)$$

$$Riskamongunexposed, R0 = B/(B + D)$$

$$Numberamongexposed, N1 = A + B$$

$$Numberamongunexposed, N0 = C + D$$

$$Numberamongdiseased, D1 = A + C$$

$$Numberamonghealth, D0 = B + D$$

$$Samplesize, N = N1 + N0$$

Risk Ratio, RR

Risk ratio is sometimes called as relative risk (RR).

$$RiskRatio = R1/R0$$

using delta method, See page 155 of Essential Medical Statistics:

$$SE(logRR) = sqrt(1/A - 1/N1 + 1/C - 1/N0)$$

$$LowerLimitofCI = exp(logRR - 1.96xSE(logRR))$$

$$UpperLimitofCI = exp(logRR + 1.96xSE(logRR))$$

Test of null hypothesis for Risk Ratio:

$$z = logRR/SE(logRR)$$

Mantel-Haenszel Method for RR:

k = Strata

$$RR_MH = \sum AkxN0k/Nk / \sum CkxN1k/Nk$$

$$SE(logRR_MH) = \sqrt{\sum D1kxN1kxN0k/Nk^2 - AkxCk/Nk / \sum AkxN0k/Nkx \sum CkxN1k/Nk}$$

$$LowerLimitofCI = exp(logRR_MH - 1.96xSE(logRR_MH))$$

$$UpperLimitofCI = exp(logRR_MH + 1.96xSE(logRR_MH))$$

Mantel–Haenszel test statistic:

A test of association (H0: RR_MH = 1 cohort studies; H0: RR_MH = 1 case–control studies) is carried out with the Mantel–Haenszel test statistic using chi-squared distribution:

$$x2_MH = (\sum Ak - \sum N1kxM1k/Nk)^2 / \sum N1kxN0kxM1kxM0k/Nk^2x(Nk-1)$$

Degree of freedom is 1.

Risk Difference, RD

$$RD = R1 - R0$$

$$SE(RD) = \sqrt{(R1(1-R1)/N1) + R0(1-R0)/N0}$$

$$LowerLimitofCI = RD - (1.96xSE(RD))$$

$$UpperLimitofCI = RD + (1.96xSE(RD))$$

Test that the difference between two proportions is zero:

$$z = RD/SE(RD)$$

Mantel-Haenszel Method for Risk Difference:

This method was proposed by Cochran and by Mantel and Haenszel. Cochran proposed using the weights nkmk/Nk, which he showed empirically to be optimal in testing a hypothesis of zero risk difference if the risk differences were constant on a logit scale. These weights will be called the Cochran-Mantel-Haenszel (CMH) weights and the estimator based on these weights will be called the CMH estimator. See details at Thomas W. O'Gorman (1994) doi.org/10.1016/0197-2456(94)90017-5

k = strata

$$WeightofCMHestimator, W = N1kxN2k/Nk$$

$$RD_CMH = \sum WkxRDk/\sum Wk$$

$$VarianceofRD_CMH, Lk = (AkxBkxN0^3 + CkxDkxN1^3)/N1kxN0kxNk^2$$

$$LowerlimitofCI = RD_CMH - (1.96x\sum Lk^1/2/\sum(Wk))$$

$$LowerlimitofCI = RD_CMH + (1.96x\sum Lk^1/2/\sum(Wk))$$

Mantel–Haenszel test statistic: Same as Risk Ratio:
A test of association (H0: RD_CMH = 1 cohort studies; H0: RD_CMH = 1 case–control studies) is carried out with the Mantel–Haenszel test statistic using chi-squared distribution:

$$x2_MH = (\sum Ak - \sum N1kxM1k/Nk)^2/\sum N1kxN0kxM1kxM0k/Nk^2x(Nk-1)$$

Degree of freedom is 1.

Population Attributable Risk, PAR
A measure of the proportion of individuals in the total population with the disease attributed to exposure to the risk factor is given by the attributable risk (AR). P is the prevalence of the risk factor in the population and RR is the relative risk for disease associated with the risk factor.

$$PAR = P(RR-1)/(1+P(RR-1))$$

$$LowerlimitofCI = P(RR_LL-1)/(1+P(RR_LL-1))$$

$$UpperlimitofCI = P(RR_UL-1)/(1+P(RR_UL-1))$$

RR_LL = Lower limit of CI of RR
RR_UL = Upper limit of CI of RR

Attributable Risk, AR
The function produces Attributable Risk percent (Raw form = not converted to 100%).

$$AR = RR - 1/RR$$

$$SE(AR) = \sqrt{D1/N(1-D1/N)(1/N1+1/N0)}$$

Efficacy
The efficacy of a treatment or intervention is measured by the proportion of cases that it prevents. Efficacy is directly calculated from the risk ratio comparing disease outcome in the treated versus control group. For a successful treatment (or intervention) this ratio will be less than 1.

$$Efficacy = 1 - RR$$

$$LowerlimitofCI = 1 - RRxexp(1.96xSE(logRR))$$

$$LowerlimitofCI = 1 - RR/exp(1.96xSE(logRR))$$

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

References

1. Betty R. Kirkwood, Jonathan A.C. Sterne (2006, ISBN:978–0–86542–871–3)
2. B. Burt Gerstman (2013, ISBN:978-1-4443-3608-5)
3. Douglas G Altman (2005, ISBN:0 7279 1375 1)

Examples

```
### Demonstration: Calculating Risk Ratios


## Essential Medical Statistics, Betty R. Kirkwood, Second Edition
## Chapter 16, Table 16.4, Page 154
## For Risk Ratio
lung <- expandTables(
    c(39, 29961, 6, 59994),
    exp_name = "smoking",
    exp_lvl = c("Smokers", "Non-smokers"),
    case_name = "cancer",
    case_lvl = c("Yes", "No")
)

## label variable and dataset
lung <- labelVar(lung,
                 c(smoking, cancer),
                 c("Smoking versus non-smoking", "Lung cancer: Yes or No"))
lung <- labelData(lung,
            "Association between smoking and lung cancer (Follow up one year)")

## check dataset
codebook(lung)

## calculate RR
mhrr(lung, smoking, by = cancer, exp_value = "Smokers", case_value = "Yes")

## calculate risk difference
mhrr(lung, smoking, by = cancer, exp_value = "Smokers", case_value = "Yes",
     measure = "diff", rnd = 5)
## Risk difference = 0.0012 = 0.12 %
```

plotRisks *Plot estimates with 95% Confidence Interval*

Description

It creates a plot for estimates and their 95\

Usage

```
plotRisks(.p, .ll, .ul, .x.name, .by.name, .ylab = "Risks")
```

Arguments

.p	prevalence
.ll	lower limit of CI
.ul	upper limit of CI
.x.name	xlab or x variable
.by.name	ylab or y variable
.ylab	ylab

printText *Display functions for* mStats

Description

Printing Functions to format and display outputs from mStats package

printText, an old version of printText2 which produces a better output.

printText2() can print data.frame in a well-formatted style. This added printLines for better visualization.

printLines() produces lines as strings for specified length.

printMsg() produces any text withint two round brackets.

getnPrintLabel() extract labels and printMsg().

wrapText() add next line to strings. This is used with cat().

Usage

```
printText(.x, .txt, .split = NULL, .printDF = FALSE)

printText2(.x, .txt, .split = NULL, .printDF = FALSE)

printLines(.x = "=", .width = 80)

printMsg(.txt = NULL)

getnPrintLabel(.data, .var.name)

wrapText(.txt, .width = 70, .sep = "\n")
```

Arguments

.x	vector, matrix, dataframe or separator (in case of printLines)
.txt	texts
.split	separator for printText
.printDF	If yes, print as Data.frame
.width	desired character length to display
.data	Dataset
.var.name	Variable name to retrieve label
.sep	separator for line break

Functions

- printText2: example
- printLines: example
- printMsg: example
- getnPrintLabel: example
- wrapText: example

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

recode *Recode contents of a variable*

Description

recode easily manipulates contents of a new variable of a data.frame

Usage

```
recode(data, var, old_values, new_values)
```

Arguments

data	dataset
var	name of a new variable
old_values	vector
new_values	vector (length of 1 or same with values_old)

Details

recode changes the values of variables including categorical variables according to the rules specified below.

In case of factor, recode first converts the vector into character, recodes and then revert back to factor.

If data is specified, it returns the whole dataframe with recoded variables.

Sample Inputs for conversion:

Old.value to New.value

»»

c(#, #) »» c(#, #)

c(#, #) »» #

#:# »» #

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## use infert data
data(infert)

## tabulate induced to check values
tab(infert, induced)

## recode induced: 1 and 2 into 1
infert.new <- recode(infert, induced, c(1, 2), 1)

## tabulate to check
tab(infert.new, induced)
```

---

regress *Linear Regression Model*

Description

`regress()` produces regression outputs which mirror outputs from `STATA`.

Usage

```
regress(data, y, ..., robust = FALSE, plot = FALSE, rnd = 3)
```

Arguments

`data`	Dataset
`y`	Dependent variable
`...`	Independent variable or multiple variables
`robust`	if `TRUE`, robust standard errors are calculated. It is used when heteroskedasticity is detected in data. Otherwise, OLS standard errors are estimated.
`plot`	logical: produces plots for model assumption
`rnd`	specify rounding of numbers. See `round`.

Details

`regress` is based on `lm`. All statistics presented in the function's output are derivates of `lm`, except AIC value which is obtained from `AIC`.

Outputs

Outputs can be divided into three parts.

1. Information about the model

Here provides number of observations (Obs.), F value, p-value from F test, R Squared value, Adjusted R Squared value, square root of mean square error (Root MSE) and AIC value.

1. Errors

Outputs from `anova(model)` is tabulated here. SS, DF and MS indicate sum of square of errors, degree of freedom and mean of square of errors.

1. Regression Output

Coefficients from summary of model are tabulated here along with 95\ confidence interval.

using Robust Standard Errors

if heteroskedasticity is present in our data sample, the ordinary least square (OLS) estimator will remain unbiased and consistent, but not efficient. The estimated OLS standard errors will be biased and cannot be solved with a larger sample size. To remedy this, robust standard erros can be used to adjusted standard errors.

$$VarianceofRobust = (N/N-K)(X'X)^(-1) \sum XiX'iei^2(X'X)^(-1)$$

where N = number of observations, and K = the number of regressors (including the intercept). This returns a Variance-covariance (VCV) matrix where the diagonal elements are the estimated heteroskedasticity-robust coefficient variances — the ones of interest. Estimated coefficient standard errors are the square root of these diagonal elements.

Note: Credits to Kevin Goulding, The Tarzan Blog.

Value

A list of three `data.frame` and `model`

Author(s)

For any feedback, please contact `Myo Minn Oo` via:

Email: `<dr.myominnoo@gmail.com>`

Website: https://myominnoo.github.io/

Examples

```
## use airquality dataset
data(airquality)
codebook(airquality)

summ(airquality)


## linear model for Ozone
regress(airquality, Ozone, Wind)

## run again with robust standard errors and with plots to check assumption
regress(airquality, Ozone, Wind, robust = TRUE, plot = TRUE)

## linear model with multiple predictors
```

```
regress(airquality, Ozone, Wind, Solar.R, Temp, Month, Day,
        robust = TRUE, plot = TRUE)
```

---

rename *Rename variable*

Description

rename() changes names of variables.

Usage

```
rename(data, old_var, new_var)
```

Arguments

data	dataset
old_var	name of existing variable
new_var	new name to be changed

Details

rename() changes the name of an existing variable, var_old to a new name, var_new; the contents of the variable are unchanged.

A group of variables can be also renamed by specifying the same number of variables in both var_old and var_new

Value

data.frame

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## using infert dataset
data(infert)

# renaming one variable
infert.new <- rename(infert, age, AGE)
codebook(infert.new)

# renaming a group of variables
infert.new <- rename(infert,
                     c(age, parity, induced, case),
                     c(AGE, PARITY, INDUCED, CASE))
codebook(infert.new)
```

---

replace *Change contents of an existing variable*

Description

`replace()` alters the values of a variable when specified conditions are met.

```
replace(data, var, value,
     var < somevalue | var > somevalue, is.na(var))
```

If conditions are not specified, `replac()` changes the whole variable with specified value.

```
replace(data, var, value)
```

Usage

```
replace(data, var, value, ...)
```

Arguments

`data`	Dataset
`var`	Variable
`value`	Replacement value
`...`	`if` conditions.

Details

It is used when multiple conditions have to be met to change a value. The function first checks whether specified value is a variable of the dataset. If yes, then the values are replaced with those of that variables with the conditions.

Value

Modified Dataset

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

Examples

```
## use infert dataset
data(infert)

## replace parity == NA if parity > 4
tab(infert, parity)

infert.new <- replace(infert, parity, NA, parity > 4)
tab(infert.new, parity)

## replace education as character
infert.new <- replace(infert, education, as.character(education))
codebook(infert.new)
tab(infert.new, education)
```

---

rowColOrder *Change reference level by row or column*

Description

`tblRowColOrder()` changes the order of levels in rows or columns by specifying names of levels.

Usage

```
rowColOrder(.tbl, .exp.value = NULL, .case.value = NULL)
```

Arguments

`.tbl`	table
`.exp.value`	reference row
`.case.value`	reference column

Details

This function is used in calculating risks, odds and associated measures.

scoreCI *Calculate confidence intervals by the Wilson Score method*

Description

`scoreCI()` generates confidence intervals by the Wilson Score method

`ciCollapse` formats two values in this format (`##.# -##.#`).

Usage

```
scoreCI(p, n, z = 1.96, correct = FALSE)

ciCollapse(ci, sep = " - ", rnd = 2, bracket = TRUE)
```

Arguments

`p`	proportion
`n`	sample size
`z`	confidence level
`correct`	a logical indicating whether to apply continuity correction
`ci`	a vector of two values (lower and upper CI values)
`sep`	separator for line break
`rnd`	specify rounding of numbers. See `round`.
`bracket`	a logical indicating whether to paste bracket to ci values

Details

`scoreCI`

The Wilson score interval is an improvement over the normal approximation interval in that the actual coverage probability is closer to the nominal value. It was developed by Edwin Bidwell Wilson (1927). (Wikipedia)

Reference:

1. Brown, Lawrence D.; Cai, T. Tony; DasGupta, Anirban (2001). "Interval Estimation for a Binomial Proportion". Statistical Science. 16 (2): 101–133.
2. Wallis, Sean A. (2013). "Binomial confidence intervals and contingency tests: mathematical fundamentals and the evaluation of alternative methods" .Journal of Quantitative Linguistics. 20 (3): 178–208.
3. Newcombe, R. G. (1998). "Two-sided confidence intervals for the single proportion: comparison of seven methods". Statistics in Medicine. 17 (8): 857–872. doi:10.1002/(SICI)1097-0258(19980430)17:8<857::AID-SIM777>3.0.CO;2-E. PMID 9595616.

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

See Also

diagTest

Examples

```
scoreCI(.20, 200)
scoreCI(.20, 200, correct = TRUE)
```

---

splitByColon *Split variable names separated by colon*

Description

splitByColon() split arguments by colon separator ":"

Usage

```
splitByColon(.data, .vars.names, .colon)
```

Arguments

.data	Dataset
.vars.names	Variables
.colon	Logical. TRUE indicates containing colon.

Value

character vector

splitTables *Split tables to make* 2x2 *tables*

Description

`splitTables()` separates table with rows more than 2 into several tables using a reference row.

Usage

```
splitTables(.tbl, .exp.value = NULL)
```

Arguments

`.tbl`	table
`.exp.value`	reference row

Details

This function is used in calculating risks, odds and associated measures.

strate *Tabulate Incidence Rates from time-to-event data*

Description

`strate()` calculates incidence rates and Corresponding 95\

Usage

```
strate(data, time, status, ..., fail = NULL, per = 1, plot = TRUE, rnd = 3)
```

Arguments

`data`	Dataset
`time`	person-time variable
`status`	outcome variable: preferrably 1 for event, 0 for censored
`...`	Variable or multiple variables Colon separator : can be used to specify multiple variables.
`fail`	Specify failure event
`per`	units to be used in reported rates
`plot`	logical value to display plots of rates across a categorical variable
`rnd`	Rounding of numbers

Details

Rates of event occurrences, known as incidence rates are outcome measures in longitudinal studies. In most longitudinal studies, follow-up times vary due to logistic resasons, different periods of recruitment, delay enrolment into the study, lost-to-follow-up, immigration or emigration and death.

Follow-up time in longitudinal studies

Period of observation (called as follow-up time) starts when individuals join the study and ends when they either have an outcome of interest, are lost-to- follow-up or the follow-up period ends, whichever happens first. This period is called **person-year-at-risk**. This is denoted by *PY* in `strate` function's output and numer of event by *D*.

Rate

is calcluated using the following formula:

$$\lambda = D/PY$$

Confidence interval of rate

is derived using the following formula:

$$95\%CI(rate) = ratexErrorFactor$$

$$ErrorFactor(rate) = exp(1.96/\sqrt{D})$$

`plot`, if `TRUE`, produces a graph of the rates against the numerical code used for categories of `by`.

Author(s)

For any feedback, please contact `Myo Minn Oo` via:

Email: `<dr.myominnoo@gmail.com>`

Website: https://myominnoo.github.io/

References

Betty R. Kirkwood, Jonathan A.C. Sterne (2006, ISBN:978–0–86542–871–3)

`summ` *Number Summary for numerical data*

Description

`summ()` generates summary statistics for numerical data as well as grouped summary measures.

Usage

```
summ(data, ..., by = NULL, na.rm = FALSE, rnd = 1)
```

Arguments

data	Dataset
...	Variable or multiple variables Colon separator : can be used to specify multiple variables.
by	Varaiable for cross-tabulation
na.rm	A logical value to specify missing values,
rnd	specify rounding of numbers. See round.

Details

summ() reports seven number summary statistics, normality and other additional metadata.

```
summ(data, var1)

summ(data, var1, var2, var3:var5, var10)

summ(data)
```

Normality test is perfomed by Shapiro-Wilk Normality Test. See more at shapiro.test.

' **ANNOTATIONS**

Obs. = observation

NA. = missing data

Mean = Mean value

Std.Dev = Standard deviation

Median = Median value

Q1 = First quartile or percentile

Q3 = Third quartile or percentile

Min = Minimum value

Max = Maximum value

Normality = P-value from Shapiro-Wilk Normality Test
Grouped Summary Measures

If by is specified, grouped summary measures are calculated and produced five number summary, excluding minimum and maximum. In addition, if levels of by are more than 2, p-values from ANOVA and Kruskal Wallis tests are displayed. Otherwise, Student's t-test and Wilcoxon signed rank test are measured and their respective p-values are tabulated.

There are two parts of the final table. The first part tabulates grouped summary measures and second part tabulates one-variable summary measures for corresponding variables.

```
summ(data, var1, var2, by = var3)

summ(data, var1, var2, var3:var5, var10, by = var11)
```

```
summ(data, by = var11)
```

Using colon : spearator

Colon separator : can be used to indicate sequence of variables.

```
summ(data, var1, var2, var3:var5, var10)
```

Value

summary measures as `data.frame`

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

References

Betty R. Kirkwood, Jonathan A.C. Sterne (2006, ISBN:978–0–86542–871–3)

Examples

```
## use iris dataset
data(iris)

summ(iris, Sepal.Length)
summ(iris, Sepal.Length:Petal.Width)

summ(iris)
```

---

tab *Tabulation*

Description

`tab()` generates one-way or two-way tabulation of variables.

Usage

```
tab(data, ..., by = NULL, row.pct = TRUE, na.rm = FALSE, rnd = 1)
```

Arguments

data	Dataset
...	Variable or multiple variables Colon separator : can be used to specify multiple variables.
by	Varaiable for cross-tabulation
row.pct	TRUE, FALSE or NULL: TRUE shows row percentages. FALSE shows column percentages. NULL shows no percentages.
na.rm	A logical value to specify missing values,
rnd	specify rounding of numbers. See round.

Details

One-way tabulation

If by is not specified, tab generates one-way tabulation of a variable or multiple variables. ... accepts multiple variables and produces corresponding tabulations.

Tabulation is displayed in Freq. (frequency), Percent. (Relative Frequency) and Cum.Percent. (Cumulative Relative frequency).

```
tab(data, var1)
```

```
tab(data, var1, var2, var3:var5, var10)
```

Two-way tabulation

Specifying by produces two-way tables. P-values from Chi-squared and Fisher's Exact tests are also shown.

Data type

Tabulation of the whole dataset requires variables to be in either of these data types: character, factor, order factor, logical.

if ... is not specified, tabulation of the whole dataset is produced.:

```
tab(data)
```

Using colon : spearator

Colon separator : can be used to indicate sequence of variables.

```
tab(data, var1, var2, var3:var5, var10)
```

Value

tabulation as list

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

References

Betty R. Kirkwood, Jonathan A.C. Sterne (2006, ISBN:978–0–86542–871–3)

Examples

```
## use infert data
data(infert)

## single variable
tab(infert, parity)

## multiple variables
tab(infert, parity, induced, case:pooled.stratum)

## tabulate the whole dataset
tab(infert)



## cross-tabulation
tab(infert, parity, by = case)
tab(infert, parity, by = case, row.pct = FALSE)
tab(infert, parity, by = case, row.pct = NULL)


## multiple variable
tab(infert, age, parity:spontaneous, education, by = case)
```

---

tabOdds *Calculating Odds*

Description

tabOdds generates cross-tabulation between two variables and display odds of failure var_case among exposure variable var_exp. It is used in case-control studies.

Usage

```
tabOdds(
  data,
  ...,
  by,
  exp_value = NULL,
  case_value = NULL,
  plot = TRUE,
  na.rm = FALSE,
  rnd = 3
)
```

Arguments

data	Dataset
...	Variable or multiple variables Colon separator : can be used to specify multiple variables.
by	Varaiable for cross-tabulation
exp_value	value for exposure as reference
case_value	value for outcome as reference
plot	logical value to display plots of rates across a categorical variable
na.rm	A logical value to specify missing values, NA in the table
rnd	specify rounding of numbers. See round.

Details

A table tabulating odds and corresponding 95\ is generated.

Formula for calculating Odds

$$OR = d1xh0/d0xh1$$

Error Factor (EF)

$$EF = exp(1.96xSE(logodds))$$

$$SE(logodds) = \sqrt{1/d + 1/h}$$

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

References

1. Betty R. Kirkwood, Jonathan A.C. Sterne (2006, ISBN:978–0–86542–871–3)
2. B. Burt Gerstman (2013, ISBN:978-1-4443-3608-5)
3. Douglas G Altman (2005, ISBN:0 7279 1375 1)

Examples

```
## use infert data
data(infert)

tabOdds(infert, education, by = case, plot = FALSE)
```

`tabRisks` *Calculating Risks and Relative Risks*

Description

`tabRisks()` cross-tabulates two variables and reports risks of failure by among exposed and unexposed levels of explanatory variable .... It is used in cross-sectional studies.

Usage

```
tabRisks(
  data,
  ...,
  by,
  exp_value = NULL,
  case_value = NULL,
  plot = TRUE,
  na.rm = FALSE,
  rnd = 3
)
```

Arguments

`data`	Dataset
`...`	Variable or multiple variables Colon separator : can be used to specify multiple variables.
`by`	Varaiable for cross-tabulation
`exp_value`	value for exposure as reference
`case_value`	value for outcome as reference
`plot`	logical value to display plots of rates across a categorical variable
`na.rm`	A logical value to specify missing values, `NA` in the table
`rnd`	specify rounding of numbers. See `round`.

Details

Risks are sometimes called proportions, incidence proportions or prevalence.

Calculating Risks

$$Risks = Outcome of Interest(A)/SampleSize(n)$$

$$StandardError, SE = \sqrt{(}px(1-p)/n)$$

$$95\%CI = Risks + / - (1.96xSE)$$

Note

This method should be avoided in small samples. Quadratic or exact binomial methods are preferred in this regard.

Author(s)

For any feedback, please contact Myo Minn Oo via:

Email: <dr.myominnoo@gmail.com>

Website: https://myominnoo.github.io/

References

1. Betty R. Kirkwood, Jonathan A.C. Sterne (2006, ISBN:978–0–86542–871–3)
2. B. Burt Gerstman (2013, ISBN:978-1-4443-3608-5)
3. Douglas G Altman (2005, ISBN:0 7279 1375 1)

Examples

```
## use infert data
data(infert)

tabRisks(infert, education, by = case, case_value = 1, plot = FALSE)
tabRisks(infert, induced, by = case, plot = FALSE)

tabRisks(infert, education, induced, by = case, plot = FALSE)
```

Index