Package 'sasLM'

February 19, 2022

Version 0.6.6

Title 'SAS' Linear Model

Description This is a core implementation of 'SAS' procedures for linear models - GLM, REG, and ANOVA. Some R packages provide type II and type III SS. However, the results of nested and complex designs are often different from those of 'SAS.' Different results does not necessarily mean incorrectness. However, many wants the same results to SAS. This package aims to achieve that.
Reference: Littell RC, Stroup WW, Freund RJ (2002, ISBN:0-471-22174-0).

Depends R (>= 3.0.0), mvtnorm

Imports methods

Author Kyun-Seop Bae [aut]

Maintainer Kyun-Seop Bae <k@acr.kr>

Repository CRAN

URL https://cran.r-project.org/package=sasLM

R topics documented:

sasLM-package *'SAS' Linear Model*

Description

This is a core implementation of 'SAS' procedures for linear models - GLM, REG, and ANOVA. Some packages provide type II and type III SS. However, the results of nested and complex designs are often different from those of 'SAS'. Different results does not necessarily mean incorrectness. However, many wants the same results to 'SAS'. This package aims to achieve that. Reference: Littell RC, Stroup WW, Freund RJ (2002, ISBN:0-471-22174-0).

Details

This will serve those who want SAS PROC GLM, REG, and ANOVA in R.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
## SAS PROC GLM Script for Typical Bioequivalence Data
# PROC GLM DATA=BEdata;
#   CLASS SEQ SUBJ PRD TRT;
#   MODEL LNCMAX = SEQ SUBJ(SEQ) PRD TRT;
#   RANDOM SUBJ(SEQ)/TEST;
#   LSMEANS TRT / DIFF=CONTROL("R") CL ALPHA=0.1;
#   ODS OUTPUT LSMeanDiffCL=LSMD;

# DATA LSMD;  SET LSMD;
#   PE = EXP(DIFFERENCE);
#   LL = EXP(LowerCL);
#   UL = EXP(UpperCL);
# PROC PRINT DATA=LSMD; RUN;
##

## SAS PROC GLM equivalent
BEdata = af(BEdata, c("SEQ", "SUBJ", "PRD", "TRT")) # Columns as factor
formula1 = log(CMAX) ~ SEQ/SUBJ + PRD + TRT # Model
GLM(formula1, BEdata) # ANOVA tables of Type I, II, III SS
EMS(formula1, BEdata) # EMS table
T3test(formula1, BEdata, Error="SEQ:SUBJ") # Hypothesis test
ci0 = CIest(formula1, BEdata, "TRT", c(-1, 1), 0.90) # 90$ CI
exp(ci0[, c("Estimate", "Lower CL", "Upper CL")]) # 90% CI of GMR

## 'nlme' or SAS PROC MIXED is preferred for an unbalanced case
## SAS PROC MIXED equivalent
# require(nlme)
# Result = lme(log(CMAX) ~ SEQ + PRD + TRT, random=~1|SUBJ, data=BEdata)
# summary(Result)
# VarCorr(Result)
# ci = intervals(Result, 0.90) ; ci
# exp(ci$fixed["TRTT",])
##
```

af *Convert some columns of a data.frame to factors*

Description

Conveniently convert some columns of data.frame into factors.

Usage

```
af(DataFrame, Cols)
```

Arguments

DataFrame	a `data.frame`
Cols	column names or indices to be converted

Details

It performs conversion of some columns in a `data.frame` into factors conveniently.

Value

Returns a `data.frame` with converted columns.

Author(s)

Kyun-Seop Bae k@acr.kr

ANOVA *Analysis of Variance similar to SAS PROC ANOVA*

Description

Analysis of variance with type I, II, and III sum of squares.

Usage

```
ANOVA(Formula, Data, eps=1e-8)
```

Arguments

Formula	a conventional formula for a linear model.
Data	a `data.frame` to be analyzed
eps	Less than this value is considered as zero.

Details

It performs the core function of SAS PROC ANOVA.

Value

The result is comparable to that of SAS PROC ANOVA.

`ANOVA`	ANOVA table for the model
`Type I`	Type I sum of square table
`Type II`	Type II sum of square table
`Type III`	Type III sum of square table

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
ANOVA(uptake ~ Plant + Type + Treatment + conc, CO2)
```

---

aov1 — *ANOVA with Type I SS*

Description

ANOVA with Type I SS.

Usage

```
aov1(Formula, Data, eps=1e-8)
```

Arguments

`Formula`	a conventional formula for a linear model.
`Data`	a `data.frame` to be analyzed
`eps`	Less than this value is considered as zero.

Details

It performs the core function of SAS PROC ANOVA.

Value

The result table is comparable to that of SAS PROC ANOVA.

`Df`	degree of freedom
`Sum Sq`	sum of square for the set of contrasts
`Mean Sq`	mean square
`F value`	F value for the F distribution
`Pr(>F)`	proability of larger than F value

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
aov1(uptake ~ Plant + Type + Treatment + conc, CO2)
```

---

aov2 *ANOVA with Type II SS*

Description

ANOVA with Type II SS.

Usage

```
aov2(Formula, Data, eps=1e-8)
```

Arguments

`Formula`	a conventional formula for a linear model.
`Data`	a `data.frame` to be analyzed
`eps`	Less than this value is considered as zero.

Details

It performs the core function of SAS PROC ANOVA.

Value

The result table is comparable to that of SAS PROC ANOVA.

`Df`	degree of freedom
`Sum Sq`	sum of square for the set of contrasts
`Mean Sq`	mean square
`F value`	F value for the F distribution
`Pr(>F)`	proability of larger than F value

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
aov2(uptake ~ Plant + Type + Treatment + conc, CO2)
aov2(uptake ~ Type, CO2)
aov2(uptake ~ Type - 1, CO2)
```

aov3 *ANOVA with Type III SS*

Description

ANOVA with Type III SS.

Usage

```
aov3(Formula, Data, eps=1e-8)
```

Arguments

`Formula`	a conventional formula for a linear model.
`Data`	a `data.frame` to be analyzed
`eps`	Less than this value is considered as zero.

Details

It performs the core function of SAS PROC ANOVA.

Value

The result table is comparable to that of SAS PROC ANOVA.

`Df`	degree of freedom
`Sum Sq`	sum of square for the set of contrasts
`Mean Sq`	mean square
`F value`	F value for the F distribution
`Pr(>F)`	proability of larger than F value

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
aov3(uptake ~ Plant + Type + Treatment + conc, CO2)
```

BEdata *An Example Data of Bioequivalence Study*

Description

Contains Cmax data from a real bioequivalence study.

Usage

```
BEdata
```

Format

A data frame with 91 observations on the following 6 variables.

- `ADM` Admission or Hospitalization Group Code: 1, 2, or 3
- `SEQ` Group or Sequence character code: 'RT' or 'TR"
- `PRD` Period numeric value: 1 or 2
- `TRT` Treatment or Drug code: 'R' or 'T'
- `SUBJ` Subject ID
- `CMAX` Cmax values

Details

This contains a real data of 2x2 bioequivalence study, which have three different hospitalization groups. See Bae KS, Kang SH. Bioequivalence data analysis for the case of separate hospitalization. Transl Clin Pharmacol. 2017;25(2):93-100. doi.org/10.12793/tcp.2017.25.2.93

---

bk *Beautify the output of knitr::kable*

Description

Trailing zeros after integer is somwwhat annoying. This removes those in the vector of strings.

Usage

```
bk(ktab, rpltag=c("n", "N"), dig=10)
```

Arguments

`ktab`	an output of `knitr::kable`
`rpltag`	tag string of replacement rows. This is usually "n" which means the sample count.
`dig`	maximum digits of decimals in the `kable` output

Details

This is convenient if used with tsum0, tsum1, tsum2, tsum3, This requires `knitr::kable`.

Value

A new processed vector of strings. The class is still `knitr_kable`.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`tsum0`, `tsum1`, `tsum2`, `tsum3`

Examples

```
## OUTPUT example
# t0 = tsum0(CO2, "uptake", c("mean", "median", "sd", "length", "min", "max"))
# bk(kable(t0)) # requires knitr package
#
# |       |        x|
# |:------|--------:|
# |mean   | 27.21310|
# |median | 28.30000|
# |sd     | 10.81441|
# |n      | 84      |
# |min    |  7.70000|
# |max    | 45.50000|

# t1 = tsum(uptake ~ Treatment, CO2,
#           e=c("mean", "median", "sd", "min", "max", "length"),
#           ou=c("chilled", "nonchilled"),
#           repl=list(c("median", "length"), c("med", "N")))
#
# bk(kable(t1, digits=3)) # requires knitr package
#
# |     | chilled| nonchilled| Combined|
# |:----|-------:|----------:|--------:|
# |mean |  23.783|     30.643|   27.213|
# |med  |  19.700|     31.300|   28.300|
# |sd   |  10.884|      9.705|   10.814|
# |min  |   7.700|     10.600|    7.700|
# |max  |  42.400|     45.500|   45.500|
# |N    |  42    |     42    |   84    |
```

`BY` *Analysis BY variable*

Description

GLM, REG, aov1 etc. functions can be run by levels of a variable.

Usage

```
BY(FUN, Formula, Data, By, ...)
```

Arguments

FUN	Function name to be called such as GLM, REG
Formula	a conventional formula for a linear model.
Data	a data.frame to be analyzed
By	a variable name in the Data
...	arguments to be passed to FUN function

Details

This mimics SAS procedues' BY clause.

Value

a list of FUN function outputs. The names are after each level.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
BY(GLM, uptake ~ Treatment + as.factor(conc), CO2, By="Type")
BY(REG, uptake ~ conc, CO2, By="Type")
```

CIest *Confidence Interval Estimation*

Description

Get point estimate and its confidence interval with given contrast and alpha value using t distribution.

Usage

```
CIest(Formula, Data, Term, Contrast, conf.level=0.95)
```

Arguments

Formula	a conventional formula for a linear model
Data	a data.frame to be analyzed
Term	a factor name to be estimated
Contrast	a level vector. Level is alphabetically ordered by default.
conf.level	confidence level of confidence interval

Details

Get point estimate and its confidence interval with given contrast and alpha value using t distribution.

Value

`Estimate`	point estimate of the input linear constrast
`Lower CL`	lower confidence limit
`Upper CL`	upper confidence limit
`Std. Error`	standard error of the point estimate
`t value`	value for t distribution
`Df`	degree of freedom
`Pr(>|t|)`	probability of larger than absolute t value from t distribution with residual's degree of freedom

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
CIest(log(CMAX) ~ SEQ/SUBJ + PRD + TRT, BEdata, "TRT", c(-1, 1), 0.90) # 90% CI
```

`Coll` *Collinearity Diagnostics*

Description

Collearity digsnotics with tolerance, VIF, eigenvalue, condition index, variance proportions

Usage

```
Coll(Formula, Data)
```

Arguments

`Formula`	fomula of the model
`Data`	input data as a matrix or data.frame

Details

Sometimes collinearity diagnostics after multiple linear regression are necessary.

Value

`Tol`	tolerance of independent variables
`VIF`	variance inflation factor of independent variables
`Eigenvalue`	eigenvalue of Z'Z (crossproduct) of standardized independent variables
`Cond. Index`	condition index
`under the names of coefficients`	proportions of variances

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
Coll(mpg ~ disp + hp + drat + wt + qsec, mtcars)
```

CONTR — *F Test with a Set of Contrasts*

Description

Do F test with a given set of contrasts.

Usage

```
CONTR(L, Formula, Data, mu=0)
```

Arguments

`L`	contrast matrix. Each row is a contrast.
`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`mu`	a vector of mu for the hypothesis L. The length should be equal to the row count of L.

Details

It performs F test with a given set of contrasts (a matrix). It is similar to the CONTRAST clause of SAS PROC GLM. This can test the hypotheis that the linear combination (function)'s mean vector is mu.

Value

Returns sum of square and its F value and p-value.

`Df`	degree of freedom
`Sum Sq`	sum of square for the set of contrasts
`Mean Sq`	mean square
`F value`	F value for the F distribution
`Pr(>F)`	proability of larger than F value

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`cSS`

Examples

```
CONTR(t(c(0, -1, 1)), uptake ~ Type, CO2) # sum of square
ANOVA(uptake ~ Type, CO2) # compare with the above
```

`Cor.test` *Correlation test of multiple numeric columns*

Description

Testing correlation between numerics columns of data with Pearson method.

Usage

```
Cor.test(Data, conf.level=0.95)
```

Arguments

`Data`	a matrix or a data.frame
`conf.level`	confidence level

Details

It uses all numeric columns of input data. It uses "pairwise.complete.obs" rows.

Value

Row names show which columns are used for the test

`Estimate`	point estimate of correlation
`Lower CL`	upper confidence limit
`Upper CL`	lower condidence limit
`t value`	t value of the t distribution
`Df`	degree of freedom
`Pr(>|t|)`	probability with the t distribution

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
Cor.test(mtcars)
```

cSS *Sum of Square with a Given Contrast Set*

Description

Calculates sum of squares of a contrast from a `lfit` result.

Usage

```
cSS(K, rx, mu=0, eps=1e-8)
```

Arguments

`K`	contrast matrix. Each row is a contrast.
`rx`	a result of `lfit` function
`mu`	a vector of mu for the hypothesis K. The length should be equal to the row count of K.
`eps`	Less than this value is considered as zero.

Details

It calculates sum of squares with given a contrast matrix and a `lfit` result. It corresponds to SAS PROC GLM CONTRAST. This can test the hypotheis that the linear combination (function)'s mean vector is mu.

Value

Returns sum of square and its F value and p-value.

`Df`	degree of freedom
`Sum Sq`	sum of square for the set of contrasts
`Mean Sq`	mean square
`F value`	F value for the F distribution
`Pr(>F)`	proability of larger than F value

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`CONTR`

Examples

```
rx = REG(uptake ~ Type, CO2, summarize=FALSE)
cSS(t(c(0, -1, 1)), rx) # sum of square
ANOVA(uptake ~ Type, CO2) # compare with the above
```

`CV` *Coefficient of Variation in percentage*

Description

Coefficient of variation in percentage.

Usage

```
CV(x)
```

Arguments

`x`	a numeric vector

Details

It removes `NA`.

Value

Coefficient of variation in percentage.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
CV(mtcars$mpg)
```

`Diffogram` *Plot Pairwise Differences*

Description

Plot pairwise differences by a common.

Usage

```
Diffogram(Formula, Data, Term, conf.level=0.95, adj="lsd", ...)
```

Arguments

`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`Term`	a factor name to be estimated
`conf.level`	confidence level of confidence interval
`adj`	"lsd", "tukey", "scheffe", "bon", or "duncan" to adjust p-value and confidence limit
`...`	arguments to be passed to `plot`

Details

This usually shows the shortest interval. It corresponds to SAS PROC GLM PDIFF. For adjust-method method "dunnett", see PDIFF function.

Value

no return value, but a plot on the current device

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

LSM, PDIFF

Examples

```
Diffogram(uptake ~ Type*Treatment + as.factor(conc), CO2, "as.factor(conc)")
```

---

e1 — *Get a Contrast Matrix for Type I SS*

Description

Makes a contrast matrix for type I SS using forward Doolittle method.

Usage

```
e1(Formula, Data, eps=1e-8)
```

Arguments

Formula	a conventional formula for a linear model
Data	a data.frame to be analyzed
eps	Less than this value is considered as zero.

Details

It makes a contrast matrix for type I SS.

Value

A contrast matrix for type I SS.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
round(e1(uptake ~ Plant + Type + Treatment + conc, CO2), 12)
```

e2 — *Get a Contrast Matrix for Type II SS*

Description

Makes a contrast matrix for type II SS.

Usage

```
e2(Formula, Data, eps=1e-8)
```

Arguments

`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`eps`	Less than this value is considered as zero.

Details

It makes a contrast matrix for type II SS.

Value

Returns a contrast matrix for type II SS.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
round(e2(uptake ~ Plant + Type + Treatment + conc, CO2), 12)
round(e2(uptake ~ Type, CO2), 12)
round(e2(uptake ~ Type - 1, CO2), 12)
```

e3 — *Get a Contrast Matrix for Type III SS*

Description

Makes a contrast matrix for type III SS.

Usage

```
e3(Formula, Data, eps=1e-8)
```

Arguments

`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`eps`	Less than this value is considered as zero.

Details

It makes a contrast matrix for type III SS.

Value

Returns a contrast matrix for type III SS.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
round(e3(uptake ~ Plant + Type + Treatment + conc, CO2), 12)
```

EMS *Expected Mean Square Formula*

Description

Calculates a formula table for expected mean square of the given contrast. The default is for Type III SS.

Usage

```
EMS(Formula, Data, Type=3, eps=1e-8)
```

Arguments

Formula	a conventional formula for a linear model
Data	a `data.frame` to be analyzed
Type	type of sum of squares. The default is 3. Type 4 is not supported yet.
eps	Less than this value is considered as zero.

Details

This is necessary for further hypothesis test of nesting factors.

Value

A coefficient matrix for Type III expected mean square

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
f1 = log(CMAX) ~ SEQ/SUBJ + PRD + TRT
EMS(f1, BEdata)
EMS(f1, BEdata, Type=1)
EMS(f1, BEdata, Type=2)
```

est *Estimate Linear Functions*

Description

Estimates Linear Functions with a given GLM result.

Usage

```
est(L, X, rx, conf.level=0.95, adj="lsd", paired=FALSE)
```

Arguments

`L`	a matrix of linear contrast rows to be tested
`X`	a model (design) matrix from `ModelMatrix`
`rx`	a result of `lfit` function
`conf.level`	confidence level of confidence limit
`adj`	adjustment method for grouping. This supports "tukey", "bon", "scheffe", "duncan", and "dunnett". This only affects grouping, not the confidence interval.
`paired`	If this is TRUE, L matrix is for the pairwise comparison such as PDIFF function.

Details

It tests rows of linear function. Linear function means linear combination of estimated coefficients. It corresponds to SAS PROC GLM ESTIMATE. Same sample size per group is assumed for the Tukey adjustment.

Value

`Estimate`	point estimate of the input linear constrast
`Lower CL`	lower confidence limit by "lsd" method
`Upper CL`	upper confidence limit by "lsd" method
`Std. Error`	standard error of the point estimate
`t value`	value for t distribution for other than "scheffe" method
`F value`	value for F distribution for "scheffe" method only
`Df`	degree of freedom of residuals
`Pr(>|t|)`	probability of larger than absolute t value from t distribution with residual's degree of freedom, for other than "scheffe" method
`Pr(>F)`	probability of larger than F value from F distribution with residual's degree of freedom, for "scheffe" method only

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`ESTM`, `PDIFF`

Examples

```
x = ModelMatrix(uptake ~ Type, CO2)
rx = REG(uptake ~ Type, CO2, summarize=FALSE)
est(t(c(0, -1, 1)), x$X, rx) # Quevec - Mississippi
t.test(uptake ~ Type, CO2) # compare with the above
```

ESTM *Estimate Linear Function*

Description

Estimates Linear Function with a formula and a dataset.

Usage

```
ESTM(L, Formula, Data, conf.level=0.95)
```

Arguments

`L`	a matrix of linear functions rows to be tested
`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`conf.level`	confidence level of confidence limit

Details

It tests rows of linear functions. Linear function means linear combination of estimated coefficients. It is similar to SAS PROC GLM ESTIMATE. This is a little convenient version of `est` function.

Value

`Estimate`	point estimate of the input linear constrast
`Lower CL`	lower confidence limit
`Upper CL`	upper confidence limit
`Std. Error`	standard error of the point estimate
`t value`	value for t distribution
`Df`	degree of freedom
`Pr(>|t|)`	probability of larger than absolute t value from t distribution with residual's degree of freedom

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`est`

Examples

```
ESTM(t(c(0, -1, 1)), uptake ~ Type, CO2) # Quevec - Mississippi
```

estmb — *Estimability Check*

Description

Check the estimability of row vectors of coefficients.

Usage

```
estmb(L, X, g2, eps=1e-8)
```

Arguments

`L`	row vectors of coefficients
`X`	a model (design) matrix from `ModelMatrix`
`g2`	g2 generalized inverse of `crossprod(X)`
`eps`	absolute value less than this is considered to be zero.

Details

It checks estimability of L, row vectors of coefficients. This corresponds to SAS PROC GLM ESTIMATE. See <Kennedy Jr. WJ, Gentle JE. Statistical Computing. 1980> p361 or <Golub GH, Styan GP. Numerical Computations for Univariate Linear Models. 1971>.

Value

a vector of logical values indicating which row is estimable (as TRUE)

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

G2SWEEP

G2SWEEP — *Generalized inverse matrix of type 2, g2 inverse*

Description

Generalized inserve is usually not unique. Some programs use this algorithm to get a uniuqe generalized inverse matrix.

Usage

```
G2SWEEP(A, Augmented=FALSE, eps=1e-08)
```

Arguments

- `A` a matrix to be inverted
- `Augmented` If this is `TRUE` and `A` is a model(design) matrix X, the last column should be X'y, the last row y'X, and the last cell y'y. See the reference and example for the detail.
- `eps` Less than this value is considered as zero.

Details

Generalized inverse of g2-type is used by some softwares to do linear regression. See 'SAS Techinical Report R106, The Sweep Operator: Its importance in Statistical Computing' by J. H. Goodnight for the detail.

Value

when `Augmented=FALSE`
: ordinary g2 inverse

when `Augmented=TRUE`
: g2 inverse and beta hats in the last column and the last row, and sum of square error (SSE) in the last cell

`attribute "rank"`
: the rank of input matrix

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`lfit`, `ModelMatrix`

Examples

```
f1 =   uptake ~ Type + Treatment # formula
x = ModelMatrix(f1, CO2)  # Model matrix and relevant information
y = model.frame(f1, CO2)[,1] # observation vector
nc = ncol(x$X) # number of columns of model matrix
XpY = crossprod(x$X, y)
aXpX = rbind(cbind(crossprod(x$X), XpY), cbind(t(XpY), crossprod(y)))
ag2 = G2SWEEP(aXpX, Augmented=TRUE)
b = ag2[1:nc, (nc + 1)] ; b # Beta hat
iXpX = ag2[1:nc, 1:nc] ; iXpX # g2 inverse of X'X
SSE = ag2[(nc + 1), (nc + 1)] ; SSE # Sum of Square Error
DFr = nrow(x$X) - attr(ag2, "rank") ; DFr # Degree of freedom for the residual

# Compare the below with the above
REG(f1, CO2)
aov1(f1, CO2)
```

geoCV *Geometric Coefficient of Variation in percentage*

Description

Geometric coefficient of variation in percentage.

Usage

```
geoCV(x)
```

Arguments

x	a numeric vector

Details

It removes NA. This is sqrt(exp(var(log(x))) - 1)*100.

Value

Geometric coefficient of variation in percentage.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
CV(mtcars$mpg)
```

geoMean *Geometric Mean without NA*

Description

mean without NA values.

Usage

```
geoMean(x)
```

Arguments

x	a vector of numerics

Details

It removes NA in the input vector.

Value

geometric mean value

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

geoCV

GLM *General Linear Model similar to SAS PROC GLM*

Description

GLM is the main function of this package.

Usage

```
GLM(Formula, Data, lsm=FALSE, conf.level=0.95, eps=1e-8)
```

Arguments

`Formula`	a conventional formula for a linear model.
`Data`	a `data.frame` to be analyzed
`lsm`	if `TRUE`, least square mean will be in the output
`conf.level`	confidence level for the confidence limit of the least square mean
`eps`	Less than this value is considered as zero.

Details

It performs the core function of SAS PROC GLM. Least square means for the tnteraction term of three variables is not supported yet.

Value

The result is comparable to that of SAS PROC GLM.

`ANOVA`	ANOVA table for the model
`Type I`	Type I sum of square table
`Type II`	Type II sum of square table
`Type III`	Type III sum of square table
`Parameter`	Parameter table with standard error, t value, p value. `TRUE` is 1, and `FALSE` is 0 in the `Estimable` column.
`Least Square Mean`	Least square mean table with confindence limit

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
GLM(uptake ~ Type*Treatment + conc, CO2[-1,]) # Making data unbalanced
GLM(uptake ~ Type*Treatment + conc, CO2[-1,], lsm=TRUE)
```

---

is.cor   *Is it a corrleation matrix?*

Description

Testing if the input matrix is a correlation matrix or not

Usage

```
is.cor(m, eps=1e-16)
```

Arguments

m	a presumed correlation matrix
eps	epsilon value. Absolute value less than this is considered as zero.

Details

Diagonal component should not be necessarily 1. But it should be close to 1.

Value

TRUE or FALSE

Author(s)

Kyun-Seop Bae k@acr.kr

Kurtosis   *Kurtosis*

Description

Kurtosis with a conventional formula.

Usage

```
Kurtosis(x)
```

Arguments

x	a vector of numerics

Details

It removes NA in the input vector.

Value

Estimate of kurtosis

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

KurtosisSE

KurtosisSE *Standard Error of Kurtosis*

Description

Standard error of the estimated kurtosis with a conventional formula.

Usage

```
KurtosisSE(x)
```

Arguments

x	a vector of numerics

Details

It removes NA in the input vector.

Value

Standard error of the estimated kurtosis

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

Kurtosis

LCL *Lower Confidence Limit*

Description

The estimate of the lower bound of confidence limit using t-distribution

Usage

```
LCL(x, conf.level=0.95)
```

Arguments

x	a vector of numerics
conf.level	confidence level

Details

It removes NA in the input vector.

Value

The estimate of the lower bound of confidence limit using t-distribution

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

UCL

lfit *Linear Fit*

Description

Fits a least square linear model.

Usage

```
lfit(x, y, eps=1e-8)
```

Arguments

x	a result of ModelMatrix
y	a column vector of response, dependent variable
eps	Less than this value is considered as zero.

Details

Minimum version of least square fit of a linear model

Value

coeffcients	beta coefficients
g2	g2 inverse
rank	rank of the model matrix
DFr	degree of freedom for the residual
SSE	sum of squares error
SST	sum of squares total
R2	R-squared
n	count of observations
R2ADJ	Adjusted R-squared

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

ModelMatrix

Examples

```
f1 = uptake ~ Type*Treatment + conc
x = ModelMatrix(f1, CO2)
y = model.frame(f1, CO2)[,1]
lfit(x, y)
```

---

lr — *Linear Regression with g2 inverse*

Description

Coefficients calculated with g2 inverse. Output is similar to `summary(lm())`.

Usage

```
lr(Formula, Data, eps=1e-8)
```

Arguments

Formula	a conventional formula for a linear model
Data	a `data.frame` to be analyzed
eps	Less than this value is considered as zero.

Details

It uses G2SWEEP to get g2 inverse. The result is similar to `summary(lm())` without options.

Value

The result is comparable to that of SAS PROC REG.

`Estimate`	point estimate of parameters, coefficients
`Std. Error`	standard error of the point estimate
`t value`	value for t distribution
`Pr(>|t|)`	probability of larger than absolute t value from t distribution with residual's degree of freedom

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
lr(uptake ~ Plant + Type + Treatment + conc, CO2)
lr(uptake ~ Plant + Type + Treatment + conc - 1, CO2)
lr(uptake ~ Type, CO2)
lr(uptake ~ Type - 1, CO2)
```

`lr0` *Simple Linear Regressions with Each Independent Variable*

Description

Usually the first step to mulitple linear regression is the simple linear regressions with single independent variable.

Usage

```
lr0(Formula, Data)
```

Arguments

`Formula`	a conventional formula for a linear model. Intercept will be added always.
`Data`	a `data.frame` to be analyzed

Details

It performs .

Value

Each row means one simple linear regression with that row name as the only independent variable.

`Intercept`	estimate of the intecept
`SE(Intercept)`	standard error of the intercept
`Slope`	estimate of the slope
`SE(Slope)`	standard error of the slope
`Rsq`	R-squared for the simple linear model
`Pr(>F)`	p-value of slope or the model

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
lr0(uptake ~ Plant + Type + Treatment + conc, CO2)
lr0(mpg ~ ., mtcars)
```

LSM *Least Square Means*

Description

Estimates least square means using g2 inverse.

Usage

```
LSM(Formula, Data, Term, conf.level=0.95, adj="lsd", hideNonEst=TRUE,
    PLOT=FALSE, ...)
```

Arguments

`Formula`	a conventional formula of model
`Data`	data.frame
`Term`	term name to be returned. If there is only one independent variable, this can be omitted.
`conf.level`	confidence level for the confidence limit
`adj`	adjustment method for grouping, "lsd"(default), "tukey", "bon", "duncan", "scheffe" are available. This does not affects SE, Lower CL, Upper CL of the output table.
`hideNonEst`	hide non-estimables
`PLOT`	whether to plot LSMs and their confidence intervals
`...`	arguments to be passed to `plot`

Details

It corresponds to SAS PROC GLM LSMEANS. The result of the second example below may be different from `emmeans`. This is because SAS or this function calculates mean of the transformed continuous variable. However, `emmeans` calculates the average before the transformation. Interaction of three variables is not supported yet. For adjustmethod method "dunnett", see `PDIFF` function.

Value

Returns a table of expectations, t values and p-values.

`Group`	group character. This appears with one-way ANOVA or `Term` or `adj` argument is provided.
`LSmean`	point estimate of least square mean
`LowerCL`	lower confidence limit with the given confidence level by "lsd" method
`UpperCL`	upper confidence limit with the given confidence level by "lsd" method
`SE`	standard error of the point estimate
`Df`	degree of freedom of point estimate

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

PDIFF, Diffogram

Examples

```
LSM(uptake ~ Type, CO2[-1,])
LSM(uptake ~ Type - 1, CO2[-1,])
LSM(uptake ~ Type*Treatment + conc, CO2[-1,])
LSM(uptake ~ Type*Treatment + conc - 1, CO2[-1,])
LSM(log(uptake) ~ Type*Treatment + log(conc), CO2[-1,])
LSM(log(uptake) ~ Type*Treatment + log(conc) - 1, CO2[-1,])
LSM(log(uptake) ~ Type*Treatment + as.factor(conc), CO2[-1,])
LSM(log(uptake) ~ Type*Treatment + as.factor(conc) - 1, CO2[-1,])
LSM(log(CMAX) ~ SEQ/SUBJ + PRD + TRT, BEdata)
LSM(log(CMAX) ~ SEQ/SUBJ + PRD + TRT - 1, BEdata)
```

Max *Max without NA*

Description

maximum without NA values.

Usage

```
Max(x)
```

Arguments

x	a vector of numerics

Details

It removes NA in the input vector.

Value

maximum value

Author(s)

Kyun-Seop Bae k@acr.kr

`Mean` *Mean without NA*

Description

mean without `NA` values.

Usage

```
Mean(x)
```

Arguments

`x`	a vector of numerics

Details

It removes `NA` in the input vector.

Value

mean value

Author(s)

Kyun-Seop Bae k@acr.kr

`Median` *Median without NA*

Description

median without `NA` values.

Usage

```
Median(x)
```

Arguments

`x`	a vector of numerics

Details

It removes `NA` in the input vector.

Value

median value

Author(s)

Kyun-Seop Bae k@acr.kr

`Min` *Min without NA*

Description

minimum without `NA` values.

Usage

```
Min(x)
```

Arguments

`x`	a vector of numerics

Details

It removes `NA` in the input vector.

Value

minimum value

Author(s)

Kyun-Seop Bae k@acr.kr

`ModelMatrix` *Model Matrix*

Description

This model matrix is similar to `model.matrix`. But it does not omit unnecessary columns.

Usage

```
ModelMatrix(Formula, Data, KeepOrder=FALSE)
```

Arguments

`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`KeepOrder`	If `KeepOrder` is `TRUE`, terms in `Formula` will be kept. This is for Type I SS.

Details

It makes the model(design) matrix for `GLM`.

Value

Model matrix and attributes similar to the output of `model.matrix`.

`X`	design matrix, i.e. model matrix
`terms`	detailed information about terms such as formula and labels
`termsIndices`	term indices
`assign`	assignemnt of columns for each terms in order, different way of expressing term indices

Author(s)

Kyun-Seop Bae k@acr.kr

`N` *Number of observations*

Description

Number of observations excluding `NA` values

Usage

```
N(x)
```

Arguments

`x`	a vector of numerics

Details

It removes `NA` in the input vector.

Value

Count of the observation

Author(s)

Kyun-Seop Bae k@acr.kr

pB *Plot Confidence and Prediction Bands for Simple Linear Regression*

Description

It plots bands of confidence interval and prediction interval for simple linear regression.

Usage

```
pB(Formula, Data, Resol=300, conf.level=0.95, lx, ly, ...)
```

Arguments

`Formula`	a formula
`Data`	a data.frame
`Resol`	resolution for the output
`conf.level`	confidence level
`lx`	x position of legend
`ly`	y position of legend
`...`	arguments to be passed to `plot`

Details

It plots. Discard return values. If `lx` or `ly` is missing, legend position is calculated automatically.

Value

Ignore return values.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
pB(hp ~ disp, mtcars)
pB(mpg ~ disp, mtcars)
```

`Pcor.test` *Partial Correlation test of multiple columns*

Description

Testing partial correlation between many columns of data with Pearson method.

Usage

```
Pcor.test(Data, x, y)
```

Arguments

`Data`	a numeric matrix or data.frame
`x`	names of to be tested columns
`y`	names of control columns

Details

It performs multiple partial correlation test. It uses "complete.obs" rows of x and y columns.

Value

Row names show which columns are used for the test

`Estimate`	point estimate of correlation
`Df`	degree of freedom
`t value`	t value of the t distribution
`Pr(>|t|)`	probability with the t distribution

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
Pcor.test(mtcars, c("mpg", "hp", "qsec"), c("drat", "wt"))
```

pD *Diagnostic Plot for Regression*

Description

Four standard diagnostic plots for regression.

Usage

```
pD(rx, Title=NULL)
```

Arguments

`rx`	a result of lm, which can give `fitted`, `residuals`, and `rstandard`.
`Title`	title to be printed on the plot

Details

Most frequently used diagnostic plots are 'observed vs. fitted', 'standarized residual vs. fitted', 'distribution plot of standard residuals', and 'Q-Q plot of standardized residuals'.

Value

Four diagnostic plots in a page.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
pD(lm(uptake ~ Plant + Type + Treatment + conc, CO2), "Diagnostic Plot")
```

PDIFF *Pairwise Difference*

Description

Estimates pairwise difference by a common method.

Usage

```
PDIFF(Formula, Data, Term, conf.level=0.95, adj="lsd", ref, PLOT=FALSE,
      reverse=FALSE, ...)
```

Arguments

Formula	a conventional formula for a linear model
Data	a data.frame to be analyzed
Term	a factor name to be estimated
conf.level	confidence level of confidence interval
adj	"lsd", "tukey", "scheffe", "bon", "duncan", or "dunnett" to adjust p-value and confidence limit
ref	reference or control level for Dunnett test
PLOT	whether to plot or not the diffogram
reverse	reverse A - B to B - A
...	arguments to be passed to plot

Details

It corresponds to PDIFF option of SAS PROC GLM.

Value

Returns a table of expectations, t values and p-values. Outpuc columns may vary according to the adjustment option.

Estimate	point estimate of the input linear constrast
Lower CL	lower confidence limit
Upper CL	upper confidence limit
Std. Error	standard error of the point estimate
t value	value for t distribution
Df	degree of freedom
Pr(>|t|)	probability of larger than absolute t value from t distribution with residual's degree of freedom

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

LSM, Diffogram

Examples

```
PDIFF(uptake ~ Type*Treatment + as.factor(conc), CO2, "as.factor(conc)")
PDIFF(uptake ~ Type*Treatment + as.factor(conc), CO2, "as.factor(conc)", adj="tukey")
```

`QuartileRange` *Inter-Quartile Range*

Description

Interquartile range (Q3 - Q1) with a conventional formula.

Usage

```
QuartileRange(x, Type=6)
```

Arguments

`x`	a vector of numerics
`Type`	a type specifier to be passed to `IQR` function

Details

It removes `NA` in the input vector.

Value

The value of interquartile range

Author(s)

Kyun-Seop Bae k@acr.kr

`Range` *Range*

Description

The range, maximum - minimum, as a scalar value.

Usage

```
Range(x)
```

Arguments

`x`	a vector of numerics

Details

It removes `NA` in the input vector.

Value

A scalar value of range

Author(s)

Kyun-Seop Bae k@acr.kr

`REG` *Regression of Linear Least Square, similar to SAS PROC REG*

Description

REG is similar to SAS PROC REG.

Usage

```
REG(Formula, Data, eps=1e-8, summarize=TRUE)
```

Arguments

`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`eps`	Less than this value is considered as zero.
`summarize`	If this is `FALSE`, `REG` returns just `lfit` result.

Details

It performs the core function of SAS PROC REG.

Value

The result is comparable to that of SAS PROC REG.

`Estimate`	point estimate of parameters, coefficients
`Estimable`	estimability: 1=TRUE, 0=FALSE. This appears only when at least one inestimability occurs.
`Std. Error`	standard error of the point estimate
`t value`	value for t distribution
`Pr(>|t|)`	probability of larger than absolute t value from t distribution with residual's degree of freedom

If `summarize=FALSE`, `REG` returns;

`coeffcients`	beta coefficients
`g2`	g2 inverse
`rank`	rank of the model matrix
`DFr`	degree of freedom for the residual
`SSE`	sum of square error

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

lr

Examples

```
REG(uptake ~ Plant + Type + Treatment + conc, CO2)
REG(uptake ~ conc, CO2, summarize=FALSE)
```

---

regD — *Regression of Conventional Way with Rich Diagnostics*

Description

regD provides rich diagnostics such as student residual, leverage(hat), Cook's D, studentized deleted residual, DFFITS, and DFBETAS.

Usage

```
regD(Formula, Data)
```

Arguments

Formula	a conventional formula for a linear model
Data	a data.frame to be analyzed

Details

It performs the conventional regression analysis. This does not use g2 inverse, therefore it cannot handle singular matrix. If the model(design) matrix is not full rank, use REG or less parameters.

Value

Coefficients	conventional coefficients summary with Wald statistics
Diagnostics	Diagnostics table for detecting outlier or influential/leverage points. This includes fitted (Predicted), residual (Residual), standard error of residual(se_resid), studentized residual(RStudent), hat(Leverage), Cook's D, studentized deleted residual(sdResid), DIFFITS, and COVRATIO.
DFBETAS	Column names are the names of coefficients. Each row shows how much each coefficient is affected by deleting the coressponding row of observation.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
regD(uptake ~ conc, CO2)
```

satt — *Satterthwaite Approximation of Pooled Variance and Degree of Freedom*

Description

Calculates pooled variance and degree of freedom using Satterthwaite equation.

Usage

```
satt(vars, dfs, ws=c(1, 1))
```

Arguments

`vars`	a vector of variances
`dfs`	a vector of degree of freedoms
`ws`	a vector of weights

Details

The input can be more than two variances.

Value

`Variance`	pooled variance
`Df`	degree of freedom

Author(s)

Kyun-Seop Bae k@acr.kr

SD — *Standard Deviation*

Description

Standard deviation of sample.

Usage

```
SD(x)
```

Arguments

`x`	a vector of numerics

Details

It removes `NA` in the input vector. The length of the vector should be larger than 1.

Value

Sample standard deviation

Author(s)

Kyun-Seop Bae k@acr.kr

`SEM` *Standard Error of the Sample Mean*

Description

The estimate of the standard error of the sample mean

Usage

```
SEM(x)
```

Arguments

`x`	a vector of numerics

Details

It removes `NA` in the input vector.

Value

The estimate of the standard error of the sample mean

Author(s)

Kyun-Seop Bae k@acr.kr

`Skewness` *Skewness*

Description

Skewness with a conventional formula.

Usage

```
Skewness(x)
```

Arguments

`x`	a vector of numerics

Details

It removes NA in the input vector.

Value

Estimate of skewness

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

SkewnessSE

SkewnessSE *Standard Error of Skewness*

Description

Standard errof of the skewness with a conventional formula.

Usage

```
SkewnessSE(x)
```

Arguments

x	a vector of numerics

Details

It removes NA in the input vector.

Value

Standard error of the estimated skewness

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

Skewness

`SLICE` *F Test with Slice*

Description

Do F test with a given slice term.

Usage

```
SLICE(Formula, Data, mTerm, sTerm)
```

Arguments

`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`mTerm`	a factor name (not interaction) to calculate sum of square and do F test with least square means
`sTerm`	a factor name to be used for slice

Details

It performs F test with a given slice term. It is similar to the SLICE option SAS PROC GLM.

Value

Returns sum of square and its F value and p-value.

`Df`	degree of freedom
`Sum Sq`	sum of square for the set of contrasts
`Mean Sq`	mean square
`F value`	F value for the F distribution
`Pr(>F)`	proability of larger than F value

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
SLICE(uptake ~ Type*Treatment, CO2, "Type", "Treatment")
SLICE(uptake ~ Type*Treatment, CO2, "Treatment", "Type")
```

`SS` *Sum of Square*

Description

Sum of squares with ANOVA.

Usage

```
SS(x, rx, L, eps=1e-8)
```

Arguments

`x`	a result of `ModelMatrix` containing design information
`rx`	a result of `lfit`
`L`	linear hypothesis, a full matrix matching the information in `x`
`eps`	Less than this value is considered as zero.

Details

It calculates sum of squares and completes the ANOVA table.

Value

`ANOVA table`	a classical ANOVA table without the residual(Error) part.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`ModelMatrix`, `lfit`

`T3MS` *Type III Expected Mean Square Formula*

Description

Calculates a formula table for expected mean square of Type III SS.

Usage

```
T3MS(Formula, Data, L0, eps=1e-8)
```

Arguments

`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`L0`	a matrix of row linear contrasts, if missed, `e3` is used
`eps`	Less than this value is considered as zero.

Details

This is necessary for further hypothesis test of nesting factors.

Value

A coefficient matrix for Type III expected mean square

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
T3MS(log(CMAX) ~ SEQ/SUBJ + PRD + TRT, BEdata)
```

T3test — *Test Type III SS using error term other than MSE*

Description

Hypothesis test of Type III SS using an error term other than MSE. This corresponds to SAS PROC GLM's RANDOM /TEST clause.

Usage

```
T3test(Formula, Data, Error="", eps=1e-8)
```

Arguments

`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`Error`	an error term. Term name should be exactly same one listed the ANOVA output.
`eps`	Less than this value is considered as zero.

Details

It tests a factor of type III SS using some other term as an error term. Here the error term should not be MSE.

Value

Returns one or more ANOVA table(s) of type III SS.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
T3test(log(CMAX) ~ SEQ/SUBJ + PRD + TRT, BEdata, "SEQ:SUBJ")
```

---

trimmedMean *Trimmed Mean*

Description

Trimmed mean wrapping mean function .

Usage

```
trimmedMean(x, Trim=0.05)
```

Arguments

x	a vector of numerics
Trim	trimming proportion. Default is 0.05

Details

It removes NA in the input vector.

Value

The value of trimmed mean

Author(s)

Kyun-Seop Bae k@acr.kr

---

tsum *Table Summary*

Description

Summarize a continuous dependent variable with or without independent variables.

Usage

```
tsum(Formula=NULL, Data=NULL, ColNames=NULL, MaxLevel=30, ...)
```

Arguments

Formula	a conventional formula
Data	a data.frame or a matrix
ColNames	If there is no Formula, this will be used.
MaxLevel	More than this will not be handled.
...	arguments to be passed to `tsum0`, `tsum1`, `tsum2`, or `tsum3`

Details

A convenient summarization function for a continuous variable. This is a wrapper function to `tsum0`, `tsum1`, `tsum2`, or `tsum3`.

Value

A data.frame of descriptive summarization values.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`tsum0`, `tsum1`, `tsum2`, `tsum3`

Examples

```
tsum(lh)
t(tsum(CO2))
t(tsum(uptake ~ Treatment, CO2))
tsum(uptake ~ Type + Treatment, CO2)
print(tsum(uptake ~ conc + Type + Treatment, CO2), digits=3)
```

`tsum0` *Table Summary 0 independent(x) variable*

Description

Summarize a continuous dependent(y) variable without any independent(x) variable.

Usage

```
tsum0(d, y, e=c("Mean", "SD", "N"), repl=list(c("length"), c("n")))
```

Arguments

d	a data.frame or matrix with colnames
y	y variable name, a continuous variable
e	a vector of summarize function names
repl	list of strings to replace after summarize. Length of list should be 2, and both should have the same length.

Details

A convenient summarization function for a continuous variable.

Value

A vector of summarized values

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

tsum, tsum1, tsum2, tsum3

Examples

```
tsum0(CO2, "uptake")
tsum0(CO2, "uptake", repl=list(c("mean", "length"), c("Mean", "n")))
```

---

tsum1 *Table Summary 1 independent(x) variable*

Description

Summarize a continuous dependent(y) variable with one independent(x) variable.

Usage

```
tsum1(d, y, u, e=c("Mean", "SD", "N"), ou="", repl=list(c("length"), ("n")))
```

Arguments

d	a data.frame or matrix with colnames
y	y variable name. a continuous variable
u	x variable name, upper side variable
e	a vector of summarize function names
ou	order of levels of upper side x variable
repl	list of strings to replace after summarize. Length of list should be 2, and both should have the same length.

Details

A convenient summarization function for a continuous variable with one x varaible.

Value

A data.frame of summarized values. Row names are from e names. Column names are from the levels of x variable.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

tsum, tsum0, tsum2, tsum3

Examples

```
tsum1(CO2, "uptake", "Treatment")
tsum1(CO2, "uptake", "Treatment",
      e=c("mean", "median", "sd", "min", "max", "length"),
      ou=c("chilled", "nonchilled"),
      repl=list(c("median", "length"), c("med", "n")))
```

---

tsum2 *Table Summary 2 independent(x) variables*

Description

Summarize a continuous dependent(y) variable with two independent(x) variables.

Usage

```
tsum2(d, y, l, u, e=c("Mean", "SD", "N"), h=NULL, ol="", ou="", rm.dup=TRUE,
      repl=list(c("length"), c("n")))
```

Arguments

d	a data.frame or matrix with colnames
y	y variable name. a continuous variable
l	x variable name to be shown on the left side
u	x variable name to be shown on the upper side
e	a vector of summarize function names
h	a vector of summarize function names for the horizontal subgroup. If NULL, it becomes same to e argument.
ol	order of levels of left side x variable
ou	order of levels of upper side x variable
rm.dup	if TRUE, duplicated name of levels are specified on the first occurrence only.
repl	list of strings to replace after summarize. Length of list should be 2, and both should have the same length.

Details

A convenient summarization function for a continuous variable with two x varaibles; one on the left side, the other on the upper side.

Value

A data.frame of summarized values. Column names are from the levels of u. Row names are basically from the levels of l.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

tsum, tsum0, tsum1, tsum3

Examples

```
tsum2(CO2, "uptake", "Type", "Treatment")
tsum2(CO2, "uptake", "Type", "conc")
tsum2(CO2, "uptake", "Type", "Treatment",
      e=c("mean", "median", "sd", "min", "max", "length"),
      ou=c("chilled", "nonchilled"),
      repl=list(c("median", "length"), c("med", "n")))
```

---

tsum3 *Table Summary 3 independent(x) variables*

Description

Summarize a continuous dependent(y) variable with three independent(x) variables.

Usage

```
tsum3(d, y, l, u, e=c("Mean", "SD", "N"), h=NULL, ol1="", ol2="", ou="",
      rm.dup=TRUE, repl=list(c("length"), c("n")))
```

Arguments

d	a data.frame or matrix with colnames
y	y variable name. a continuous variable
l	a vector of two x variable name to be shown on the left side. The length should be 2.
u	x variable name to be shown on the upper side
e	a vector of summarize function names
h	a list of two vectors of summarize function names for the first and second horizontal subgroups. If NULL, it becomes same to e argument.
ol1	order of levels of 1st left side x variable
ol2	order of levels of 2nd left side x variable
ou	order of levels of upper side x variable
rm.dup	if TRUE, duplicated name of levels are specified on the first occurrence only.
repl	list of strings to replace after summarize. Length of list should be 2, and both should have the same length.

Details

A convenient summarization function for a continuous variable with three x varaibles; two on the left side, the other on the upper side.

Value

A data.frame of summarized values. Column names are from the levels of `u`. Row names are basically from the levels of `l`.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`tsum`, `tsum0`, `tsum1`, `tsum2`

Examples

```
tsum3(CO2, "uptake", c("Type", "Treatment"), "conc")
tsum3(CO2, "uptake", c("Type", "Treatment"), "conc",
      e=c("mean", "median", "sd", "min", "max", "length"),
      h=list(c("mean", "sd", "length"), c("mean", "length")),
      ol2=c("chilled", "nonchilled"),
      repl=list(c("median", "length"), c("med", "n")))
```

UCL *Upper Confidence Limit*

Description

The estimate of the upper bound of confidence limit using t-distribution

Usage

```
UCL(x, conf.level=0.95)
```

Arguments

`x`	a vector of numerics
`conf.level`	confidence level

Details

It removes `NA` in the input vector.

Value

The estimate of the upper bound of confidence limit using t-distribution

Author(s)

Kyun-Seop Bae k@acr.kr

UNIV *Univariate Descriptive Statistics*

Description

Returns descriptive statistics of a numeric vector.

Usage

```
UNIV(x, conf.level = 0.95)
```

Arguments

`x`	a numeric vector
`conf.level`	confidence level for confidence limit

Details

A convenient and comprehensive descriptive statistics. NA is removed during the calculation. This is similar to SAS PROC UNIVARIATE.

Value

`nAll`	count of all element in the input vector
`nNA`	count of NA element
`nFinite`	count of finite numbers
`Mean`	mean excluding NA
`SD`	standard deviation excluding NA
`CV`	coefficient of variation in percent
`SEM`	standard error of the sample mean, sample mean divided by nFinite
`LowerConfLimit`	lower confidence limit of given confidence interval
`UpperConfLimit`	upper confidence limit of given confidence interval
`TrimmedMean`	trimmed mean with trimming 1 - confidence level
`Min`	minimum value
`Q1`	first quartile value
`Median`	median value
`Q3`	third quartile value
`Max`	maximum value
`Range`	range of finite numbers. maximum - minimum
`Skewness`	skewness
`SkewnessSE`	standard error of skewness
`Kurtosis`	kurtosis
`KurtosisSE`	kurtosis
`GeometricMean`	geometric mean, calculated only when all given values are positive.
`GeometricCV`	geometric coefficient of variation in percent, calculated only when all given values are positive.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
UNIV(lh)
```

Index