

# Package ‘sasLM’

January 7, 2023

**Version** 0.9.4

**Title** 'SAS' Linear Model

**Description** This is a core implementation of 'SAS' procedures for linear models - GLM, REG, ANOVA, FREQ, and UNIVARIATE. 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 want 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

**Suggests** MASS

**Author** Kyun-Seop Bae [aut]

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

**Copyright** 2020-, Kyun-Seop Bae

**License** GPL-3

**Repository** CRAN

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

**NeedsCompilation** no

## R topics documented:

sasLM-package	3
af	4
aov1	5
aov2	6
aov3	7
BEdata	8
bk	8
BY	9
CIest	10
Coll	11
CONTR	12
Cor.test	13
corFisher	14
cSS	15

CV . . . . .	16
Diffogram . . . . .	16
e1 . . . . .	17
e2 . . . . .	18
e3 . . . . .	18
EMS . . . . .	19
est . . . . .	20
ESTM . . . . .	21
estmb . . . . .	22
G2SWEEP . . . . .	22
geoCV . . . . .	24
geoMean . . . . .	24
GLM . . . . .	25
is.cor . . . . .	26
Kurtosis . . . . .	27
KurtosisSE . . . . .	27
LCL . . . . .	28
lfit . . . . .	29
lr . . . . .	30
lr0 . . . . .	31
LSM . . . . .	32
Max . . . . .	33
Mean . . . . .	34
Median . . . . .	34
Min . . . . .	35
ModelMatrix . . . . .	35
N . . . . .	36
OR . . . . .	37
ORcmh . . . . .	38
ORinv . . . . .	39
ORmn . . . . .	40
ORmn1 . . . . .	41
pB . . . . .	42
Pcor.test . . . . .	43
pD . . . . .	44
PDIFF . . . . .	44
pResD . . . . .	46
QuartileRange . . . . .	46
Range . . . . .	47
RD . . . . .	47
RDinv . . . . .	48
RDmn . . . . .	50
RDmn1 . . . . .	51
REG . . . . .	52
regD . . . . .	53
RR . . . . .	54
RRinv . . . . .	55
RRmn . . . . .	56
RRmn1 . . . . .	57
satt . . . . .	58
ScoreCI . . . . .	59
SD . . . . .	60

SEM . . . . .	60
Skewness . . . . .	61
SkewnessSE . . . . .	61
SLICE . . . . .	62
SS . . . . .	63
T3MS . . . . .	63
T3test . . . . .	64
trimmedMean . . . . .	65
tsum . . . . .	65
tsum0 . . . . .	66
tsum1 . . . . .	67
tsum2 . . . . .	68
tsum3 . . . . .	69
UCL . . . . .	70
UNIV . . . . .	71
WhiteTest . . . . .	72
<b>Index</b>	<b>73</b>

---

sasLM-package	<i>'SAS' Linear Model</i>
---------------	---------------------------

---

## 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

---

aov1	<i>ANOVA with Type I SS</i>
------	-----------------------------

---

**Description**

ANOVA with Type I SS.

**Usage**

```
aov1(Formula, Data)
```

**Arguments**

Formula	a conventional formula for a linear model.
Data	a <code>data.frame</code> to be analyzed

**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)	probability of larger than F value

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**Examples**

```
aov1(uptake ~ Plant + Type + Treatment + conc, C02)
```

---

aov2	<i>ANOVA with Type II SS</i>
------	------------------------------

---

**Description**

ANOVA with Type II SS.

**Usage**

```
aov2(Formula, Data)
```

**Arguments**

Formula	a conventional formula for a linear model.
Data	a <code>data.frame</code> to be analyzed

**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)	probability of larger than F value

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**Examples**

```
aov2(uptake ~ Plant + Type + Treatment + conc, C02)
aov2(uptake ~ Type, C02)
aov2(uptake ~ Type - 1, C02)
```

---

aov3*ANOVA with Type III SS*

---

**Description**

ANOVA with Type III SS.

**Usage**

```
aov3(Formula, Data)
```

**Arguments**

Formula	a conventional formula for a linear model.
Data	a <code>data.frame</code> to be analyzed

**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)	probability of larger than F value

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**Examples**

```
aov3(uptake ~ Plant + Type + Treatment + conc, C02)
```

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 somewhat 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](mailto: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**

<code>FUN</code>	Function name to be called such as GLM, REG
<code>Formula</code>	a conventional formula for a linear model.
<code>Data</code>	a <code>data.frame</code> to be analyzed
<code>By</code>	a variable name in the Data
<code>...</code>	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), C02, By="Type")
BY(REG, uptake ~ conc, C02, By="Type")
```

---

CIest	<i>Confidence Interval Estimation</i>
-------	---------------------------------------

---

**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**

<code>Formula</code>	a conventional formula for a linear model
<code>Data</code>	a <code>data.frame</code> to be analyzed
<code>Term</code>	a factor name to be estimated
<code>Contrast</code>	a level vector. Level is alphabetically ordered by default.
<code>conf.level</code>	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 contrast
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	<i>Collinearity Diagnostics</i>
------	---------------------------------

---

**Description**

Collinearity diagnostics with tolerance, VIF, eigenvalue, condition index, variance proportions

**Usage**

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

**Arguments**

Formula	formula 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
Proportions of variances	under the names of coefficients

**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 hypothesis 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)	probability 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
GLM(uptake ~ Type, CO2) # compare with the above
```

---

Cor.test	<i>Correlation test of multiple numeric columns</i>
----------	---

---

**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 confidence 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)
```

---

`corFisher`*Correlation test by Fisher's Z transformation*

---

**Description**

Testing correlation between two numeric vectors by Fisher's Z transformation

**Usage**

```
corFisher(x, y, conf.level=0.95, rho=0)
```

**Arguments**

<code>x</code>	the first input numeric vector
<code>y</code>	the second input numeric vector
<code>conf.level</code>	confidence level
<code>rho</code>	population correlation rho under null hypothesis

**Details**

This only accept two numeric vectors.

**Value**

<code>N</code>	sample size, length of input vectors
<code>r</code>	sample correlation
<code>Fisher.z</code>	Fisher's z
<code>bias</code>	bias to correct
<code>rho.hat</code>	point estimate of population rho
<code>conf.level</code>	confidence level for the confidence interval
<code>lower</code>	lower limit of condidence interval
<code>upper</code>	upper limit of condidence interval
<code>rho0</code>	population correlation rho under null hypothesis
<code>p.value</code>	p value under the null hypothesis

**Author(s)**

Kyun-Seop Bae `k@acr.kr`

**References**

Fisher RA. Statistical Methods for Research Workers. 14e. 1973

**Examples**

```
corFisher(mtcars$disp, mtcars$hp, rho=0.6)
```

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 <code>lfit</code> 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 hypothesis 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)	probability of larger than F value

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**See Also**

[CONTR](#)

**Examples**

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

---

CV	<i>Coefficient of Variation in percentage</i>
----	---

---

**Description**

Coefficient of variation in percentage.

**Usage**

```
CV(y)
```

**Arguments**

y                      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	<i>Plot Pairwise Differences</i>
-----------	----------------------------------

---

**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 adjustment 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), C02, "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(XpX, eps=1e-8)
```

**Arguments**

XpX	crossproduct of a design or model matrix. This should have appropriate column names.
eps	Less than this value is considered as zero.

**Details**

It makes a contrast matrix for type I SS. If zapsmall is used, the result becomes more inaccurate.

**Value**

A contrast matrix for type I SS.

**Author(s)**

Kyun-Seop Bae k@acr.kr

**Examples**

```
x = ModelMatrix(uptake ~ Plant + Type + Treatment + conc, C02)
round(e1(crossprod(x$x)), 12)
```

e2

*Get a Contrast Matrix for Type II SS***Description**

Makes a contrast matrix for type II SS.

**Usage**

```
e2(x, eps=1e-8)
```

**Arguments**

x	an output of ModelMatrix
eps	Less than this value is considered as zero.

**Details**

It makes a contrast matrix for type II SS. If zapsmall is used, the result becomes more inaccurate.

**Value**

A contrast matrix for type II SS.

**Author(s)**

Kyun-Seop Bae k@acr.kr

**Examples**

```
round(e2(ModelMatrix(uptake ~ Plant + Type + Treatment + conc, C02)), 12)
round(e2(ModelMatrix(uptake ~ Type, C02)), 12)
round(e2(ModelMatrix(uptake ~ Type - 1, C02)), 12)
```

e3

*Get a Contrast Matrix for Type III SS***Description**

Makes a contrast matrix for type III SS.

**Usage**

```
e3(x, eps=1e-8)
```

**Arguments**

x	an output of ModelMatrix
eps	Less than this value is considered as zero.

**Details**

It makes a contrast matrix for type III SS. If `zapsmall` is used, the result becomes more inaccurate.

**Value**

A contrast matrix for type III SS.

**Author(s)**

Kyun-Seop Bae `k@acr.kr`

**Examples**

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

---

EMS	<i>Expected Mean Square Formula</i>
-----	-------------------------------------

---

**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 <code>data.frame</code> 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	<i>Estimate Linear Functions</i>
-----	----------------------------------

---

**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 <code>ModelMatrix</code>
rx	a result of <code>lfit</code> 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 <code>PDIFF</code> 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 contrast
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](mailto:k@acr.kr)

**See Also**

[ESTM](#), [PDIFF](#)

**Examples**

```
x = ModelMatrix(uptake ~ Type, C02)
rx = REG(uptake ~ Type, C02, summarize=FALSE)
est(t(c(0, -1, 1)), x$X, rx) # Quevec - Mississippi
t.test(uptake ~ Type, C02) # 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 contrast
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, C02) # Quevec - Mississippi
```

---

estmb	<i>Estimability Check</i>
-------	---------------------------

---

**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 <code>ModelMatrix</code>
g2	g2 generalized inverse of <code>crossprod(X)</code>
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](mailto:k@acr.kr)

**See Also**

[G2SWEEP](#)

---

G2SWEEP	<i>Generalized inverse matrix of type 2, g2 inverse</i>
---------	---

---

**Description**

Generalized inverse is usually not unique. Some programs use this algorithm to get a unique 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 Technical 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, C02) # Model matrix and relevant information
y = model.frame(f1, C02)[, 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, C02)
aov1(f1, C02)
```

---

 geoCV

*Geometric Coefficient of Variation in percentage*


---

**Description**

Geometric coefficient of variation in percentage.

**Usage**

```
geoCV(y)
```

**Arguments**

`y` a numeric vector

**Details**

It removes NA. This is  $\sqrt{\exp(\text{var}(\log(x))) - 1} \times 100$ .

**Value**

Geometric coefficient of variation in percentage.

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**See Also**

[geoMean](#)

**Examples**

```
geoCV(mtcars$mpg)
```

---

 geoMean

*Geometric Mean without NA*


---

**Description**

mean without NA values.

**Usage**

```
geoMean(y)
```

**Arguments**

`y` a vector of numerics



**Details**

It removes NA in the input vector.

**Value**

geometric mean value

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**See Also**

[geoCV](#)

**Examples**

```
geoMean(mtcars$mpg)
```

---

GLM

*General Linear Model similar to SAS PROC GLM*


---

**Description**

GLM is the main function of this package.

**Usage**

```
GLM(Formula, Data, BETA=FALSE, EMEAN=FALSE, conf.level=0.95, Weights=1)
```

**Arguments**

Formula	a conventional formula for a linear model.
Data	a <code>data.frame</code> to be analyzed
BETA	if TRUE, coefficients (parameters) of REG will be returned. This is equivalent to SOLUTION option of SAS PROC GLM
EMEAN	if TRUE, least square means (or expected means) will be returned. This is equivalent to LSMEANS clause of SAS PROC GLM
conf.level	confidence level for the confidence limit of the least square mean
Weights	weights for the weighted least square

**Details**

It performs the core function of SAS PROC GLM. Least square means for the interaction 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
Fitness	Some measures of goodness of fit such as R-square and CV
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. This is returned only with BETA=TRUE option.
Expected Mean	Least square (or expected) mean table with confidence limit. This is returned only with EMEAN=TRUE option.

**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,], BETA=TRUE)
GLM(uptake ~ Type*Treatment + conc, CO2[-1,], EMEAN=TRUE)
GLM(uptake ~ Type*Treatment + conc, CO2[-1,], BETA=TRUE, EMEAN=TRUE)
```

---

is.cor

---

*Is it a correlation 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	<i>Kurtosis</i>
----------	-----------------

---

**Description**

Kurtosis with a conventional formula.

**Usage**

```
Kurtosis(y)
```

**Arguments**

y                      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	<i>Standard Error of Kurtosis</i>
------------	-----------------------------------

---

**Description**

Standard error of the estimated kurtosis with a conventional formula.

**Usage**

```
KurtosisSE(y)
```

**Arguments**

y                      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](mailto: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(y, conf.level=0.95)
```

**Arguments**

<code>y</code>	a vector of numerics
<code>conf.level</code>	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](mailto: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**

coefficients	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
DFr2	degree of freedom of the residual for beta coefficient

**Author(s)**

Kyun-Seop Bae k@acr.kr

**See Also**

[ModelMatrix](#)

**Examples**

```
f1 = uptake ~ Type*Treatment + conc
x = ModelMatrix(f1, C02)
y = model.frame(f1, C02)[,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 <code>data.frame</code> 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](mailto:k@acr.kr)

**Examples**

```
lr(uptake ~ Plant + Type + Treatment + conc, C02)
lr(uptake ~ Plant + Type + Treatment + conc - 1, C02)
lr(uptake ~ Type, C02)
lr(uptake ~ Type - 1, C02)
```

---

lr0

---

Simple Linear Regressions with Each Independent Variable

---

## Description

Usually the first step to multiple 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 <code>data.frame</code> 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 intercept
SE(Intercept)	standard error of the intercept
Slope	estimate of the slope
SE(Slope)	standard error of the slope
Rsquared	R-squared for the simple linear model
Pr(>F)	p-value of slope or the model

## Author(s)

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

## Examples

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

---

LSM	<i>Least Square Means</i>
-----	---------------------------

---

**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, C02[-1,])
LSM(uptake ~ Type - 1, C02[-1,])
LSM(uptake ~ Type*Treatment + conc, C02[-1,])
LSM(uptake ~ Type*Treatment + conc - 1, C02[-1,])
LSM(log(uptake) ~ Type*Treatment + log(conc), C02[-1,])
LSM(log(uptake) ~ Type*Treatment + log(conc) - 1, C02[-1,])
LSM(log(uptake) ~ Type*Treatment + as.factor(conc), C02[-1,])
LSM(log(uptake) ~ Type*Treatment + as.factor(conc) - 1, C02[-1,])
LSM(log(CMAX) ~ SEQ/SUBJ + PRD + TRT, BEdata)
LSM(log(CMAX) ~ SEQ/SUBJ + PRD + TRT - 1, BEdata)
```

---

Max	<i>Max without NA</i>
-----	-----------------------

---

Description

maximum without NA values.

Usage

```
Max(y)
```

Arguments

y                      a vector of numerics

Details

It removes NA in the input vector.

Value

maximum value

Author(s)

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

---

Mean	<i>Mean without NA</i>
------	------------------------

---

**Description**

mean without NA values.

**Usage**

```
Mean(y)
```

**Arguments**

y                      a vector of numerics

**Details**

It removes NA in the input vector.

**Value**

mean value

**Author(s)**

Kyun-Seop Bae k@acr.kr

---

Median	<i>Median without NA</i>
--------	--------------------------

---

**Description**

median without NA values.

**Usage**

```
Median(y)
```

**Arguments**

y                      a vector of numerics

**Details**

It removes NA in the input vector.

**Value**

median value

**Author(s)**

Kyun-Seop Bae k@acr.kr

---

Min	<i>Min without NA</i>
-----	-----------------------

---

**Description**

minimum without NA values.

**Usage**

```
Min(y)
```

**Arguments**

`y` a vector of numerics

**Details**

It removes NA in the input vector.

**Value**

minimum value

**Author(s)**

Kyun-Seop Bae k@acr.kr

---

ModelMatrix	<i>Model Matrix</i>
-------------	---------------------

---

**Description**

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

**Usage**

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

**Arguments**

Formula	a conventional formula for a linear model
Data	a <code>data.frame</code> to be analyzed
KeepOrder	If <code>KeepOrder</code> is TRUE, terms in <code>Formula</code> will be kept. This is for Type I SS.
XpX	If <code>XpX</code> is TRUE, the cross-product of the design matrix ( $XpX$ , $X'X$ ) will be returned instead of the design matrix ( $X$ ).

**Details**

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

**Value**

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

<code>X</code>	design matrix, i.e. model matrix
<code>XpX</code>	cross-product of the design matrix, $X'X$
<code>terms</code>	detailed information about terms such as formula and labels
<code>termsIndices</code>	term indices
<code>assign</code>	assignment of columns for each terms in order, different way of expressing term indices

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

---

N	<i>Number of observations</i>
---	-------------------------------

---

**Description**

Number of observations excluding NA values

**Usage**

`N(y)`

**Arguments**

`y` 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](mailto:k@acr.kr)

---

OR	<i>Odds Ratio of two groups</i>
----	---------------------------------

---

**Description**

Odds Ratio between two groups

**Usage**

```
OR(y1, n1, y2, n2, conf.level=0.95)
```

**Arguments**

y1	positive event count of test (the first) group
n1	total count of the test (the first) group
y2	positive event count of control (the second) group
n2	total count of control (the second) group
conf.level	confidence level

**Details**

It calculates odds ratio of two groups. No continuity correction here. If you need percent scale, multiply the output by 100.

**Value**

The result is a data.frame.

odd1	proportion from the first group
odd2	proportion from the second group
OR	odds ratio, odd1/odd2
SElog	standard error of log(OR)
lower	lower confidence limit of OR
upper	upper confidence limit of OR

**Author(s)**

Kyun-Seop Bae k@acr.kr

**See Also**

[RD](#), [RR](#), [RDmn1](#), [RRmn1](#), [ORmn1](#), [RDmn](#), [RRmn](#), [ORmn](#)

**Examples**

```
OR(104, 11037, 189, 11034) # no continuity correction
```

ORcmh

*Odds Ratio of two groups with strata by CMH method***Description**

Odds ratio and its score confidence interval of two groups with stratification by Cochran-Mantel-Haenszel method

**Usage**

```
ORcmh(d0, conf.level=0.95)
```

**Arguments**

<code>d0</code>	A data.frame or matrix, of which each row means a strata. This should have four columns named y1, n1, y2, and n2; y1 and y2 for events of each group, n1 and n2 for sample size of each strata. The second group is usually the control group.
<code>conf.level</code>	confidence level

**Details**

It calculates odds ratio and its score confidence interval of two groups. This can be also used for meta-analysis.

**Value**

The following output will be returned for each strata and common value. There is no standard error.

<code>odd1</code>	odd from the first group, $y1/(n1 - y1)$
<code>odd2</code>	odd from the second group, $y2/(n2 - y2)$
<code>OR</code>	odds ratio, $odd1/odd2$ . Point estimate of common OR is calculated with MH weight.
<code>lower</code>	lower confidence limit of OR
<code>upper</code>	upper confidence limit of OR

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**See Also**

[RDmn1](#), [RRmn1](#), [ORmn1](#), [RDmn](#), [RRmn](#), [ORmn](#), [RDinv](#), [RRinv](#), [ORinv](#)

**Examples**

```
d1 = matrix(c(25, 339, 28, 335, 23, 370, 40, 364), nrow=2, byrow=TRUE)
colnames(d1) = c("y1", "n1", "y2", "n2")
ORcmh(d1)
```

---

ORinv	<i>Odds Ratio of two groups with strata by inverse variance method</i>
-------	--

---

**Description**

Odds ratio and its score confidence interval of two groups with stratification by inverse variance method

**Usage**

```
ORinv(d0, conf.level=0.95)
```

**Arguments**

<code>d0</code>	A data.frame or matrix, of which each row means a strata. This should have four columns named <code>y1</code> , <code>n1</code> , <code>y2</code> , and <code>n2</code> ; <code>y1</code> and <code>y2</code> for events of each group, <code>n1</code> and <code>n2</code> for sample size of each strata. The second group is usually the control group.
<code>conf.level</code>	confidence level

**Details**

It calculates odds ratio and its score confidence interval of two groups. The confidence interval is asymmetric and there is no standard error in the output. This supports stratification. This implementation uses `nlminb` minimization function which usually gives about 7 significant digits. `PropCIs::orscoreci` function uses incremental or decremental search by the factor of 1.001 which gives only about 3 significant digits.

**Value**

The following output will be returned for each strata and common value. There is no standard error.

<code>odd1</code>	odd from the first group, $y1/(n1 - y1)$
<code>odd2</code>	odd from the second group, $y2/(n2 - y2)$
<code>OR</code>	odds ratio, <code>odd1/odd2</code> . Point estimate of common OR is calculated with MH weight.
<code>lower</code>	lower confidence limit of OR
<code>upper</code>	upper confidence limit of OR

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**References**

Miettinen O, Nurminen M. Comparative analysis of two rates. Stat Med 1985;4:213-26

**See Also**

[RDMn1](#), [RRmn1](#), [ORMn1](#), [RDmn](#), [RRmn](#), [ORMn](#), [RDinv](#), [RRinv](#), [ORcmh](#)

**Examples**

```
d1 = matrix(c(25, 339, 28, 335, 23, 370, 40, 364), nrow=2, byrow=TRUE)
colnames(d1) = c("y1", "n1", "y2", "n2")
ORinv(d1)
```

ORmn

*Odds Ratio and Score CI of two groups with strata by MN method***Description**

Odds ratio and its score confidence interval of two groups with stratification by Miettinen and Nurminen method

**Usage**

```
ORmn(d0, conf.level=0.95, eps=1e-8)
```

**Arguments**

d0	A data.frame or matrix, of which each row means a strata. This should have four columns named y1, n1, y2, and n2; y1 and y2 for events of each group, n1 and n2 for sample size of each strata. The second group is usually the control group.
conf.level	confidence level
eps	absolute value less than eps is regarded as negligible

**Details**

It calculates odds ratio and its score confidence interval of two groups. The confidence interval is asymmetric and there is no standard error in the output. This supports stratification. This implementation uses nlminb minimization function which usually gives about 7 significant digits. PropCIs::orscoreci function uses incremental or decremental search by the factor of 1.001 which gives only about 3 significant digits.

**Value**

The following output will be returned for each strata and common value. There is no standard error.

odd1	odd from the first group, $y1/(n1 - y1)$
odd2	odd from the second group, $y2/(n2 - y2)$
OR	odds ratio, odd1/odd2. Point estimate of common OR is calculated with MN weight.
lower	lower confidence limit of OR
upper	upper confidence limit of OR

**Author(s)**

Kyun-Seop Bae k@acr.kr

**References**

Miettinen O, Nurminen M. Comparative analysis of two rates. Stat Med 1985;4:213-26



**See Also**

[RDMn1](#), [RRmn1](#), [ORmn1](#), [RDmn](#), [RRmn](#), [RDinv](#), [RRinv](#), [ORinv](#), [ORcmh](#)

**Examples**

```
d1 = matrix(c(25, 339, 28, 335, 23, 370, 40, 364), nrow=2, byrow=TRUE)
colnames(d1) = c("y1", "n1", "y2", "n2")
ORmn(d1)
```

---

ORmn1

*Odds Ratio and Score CI of two groups without strata by MN method*


---

**Description**

Odds ratio and its score confidence interval of two groups without stratification

**Usage**

```
ORmn1(y1, n1, y2, n2, conf.level=0.95, eps=1e-8)
```

**Arguments**

y1	positive event count of test (the first) group
n1	total count of the test (the first) group
y2	positive event count of control (the second) group
n2	total count of control (the second) group
conf.level	confidence level
eps	absolute value less than eps is regarded as negligible

**Details**

It calculates odds ratio and its score confidence interval of two groups. The confidence interval is asymmetric and there is no standard error in the output. This does not support stratification. This implementation uses nlminb minimization function which usually gives about 7 significant digits. PropCIs::orscoreci function uses incremental or decremental search by the factor of 1.001 which gives only about 3 significant digits.

**Value**

The result is a data.frame. There is no standard error.

odd1	odd from the first group, $y1/(n1 - y1)$
odd2	odd from the second group, $y2/(n2 - y2)$
OR	odds ratio, $odd1/odd2$
lower	lower confidence limit of OR
upper	upper confidence limit of OR

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

## References

Miettinen O, Nurminen M. Comparative analysis of two rates. Stat Med 1985;4:213-26

## See Also

[RDmn1](#), [RRmn1](#), [RDmn](#), [RRmn](#), [ORmn](#)

## Examples

```
ORmn1(104, 11037, 189, 11034)
```

---

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](mailto: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**

<code>Data</code>	a numeric matrix or data.frame
<code>x</code>	names of columns to be tested
<code>y</code>	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

<code>Estimate</code>	point estimate of correlation
<code>Df</code>	degree of freedom
<code>t value</code>	t value of the t distribution
<code>Pr(&gt; t )</code>	probability with the t distribution

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**Examples**

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

---

pD	<i>Diagnostic Plot for Regression</i>
----	---------------------------------------

---

**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, C02), "Diagnostic Plot")
```

---

PDIFF	<i>Pairwise Difference</i>
-------	----------------------------

---

**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 <code>data.frame</code> 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), C02, "as.factor(conc)")
PDIFF(uptake ~ Type*Treatment + as.factor(conc), C02, "as.factor(conc)", adj="tukey")
```

---

pResD	<i>Residual Diagnostic Plot for Regression</i>
-------	--

---

**Description**

Nine residual diagnostics plot.

**Usage**

```
pResD(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**

SAS style residual diagnostic plots.

**Value**

Nine residual diagnostic plots in a page.

**Author(s)**

Kyun-Seop Bae k@acr.kr

**Examples**

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

---

QuartileRange	<i>Inter-Quartile Range</i>
---------------	-----------------------------

---

**Description**

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

**Usage**

```
QuartileRange(y, Type=2)
```

**Arguments**

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

**Details**

It removes NA in the input vector. Type 2 is SAS default, while Type 6 is SPSS default.

**Value**

The value of interquartile range

**Author(s)**

Kyun-Seop Bae k@acr.kr

---

Range	<i>Range</i>
-------	--------------

---

**Description**

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

**Usage**

```
Range(y)
```

**Arguments**

y                      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

---

RD	<i>Risk Difference between two groups</i>
----	---

---

**Description**

Risk (proportion) difference between two groups

**Usage**

```
RD(y1, n1, y2, n2, conf.level=0.95)
```

**Arguments**

y1	positive event count of test (the first) group
n1	total count of the test (the first) group
y2	positive event count of control (the second) group
n2	total count of control (the second) group
conf.level	confidence level

**Details**

It calculates risk difference between two groups. No continuity correction here. If you need percent scale, multiply the output by 100.

**Value**

The result is a data.frame.

p1	proportion from the first group
p2	proportion from the second group
RD	risk difference, $p1 - p2$
SE	standard error of RD
lower	lower confidence limit of RD
upper	upper confidence limit of RD

**Author(s)**

Kyun-Seop Bae k@acr.kr

**See Also**

[RR](#), [OR](#), [RDmn1](#), [RRmn1](#), [ORmn1](#), [RDmn](#), [RRmn](#), [ORmn](#)

**Examples**

```
RD(104, 11037, 189, 11034) # no continuity correction
```

---

RDinv

---

*Risk Difference between two groups with strata by inverse variance method*


---

**Description**

Risk difference and its score confidence interval between two groups with stratification by inverse variance method

**Usage**

```
RDinv(d0, conf.level=0.95)
```



**Arguments**

<code>d0</code>	A data.frame or matrix, of which each row means a strata. This should have four columns named <code>y1</code> , <code>n1</code> , <code>y2</code> , and <code>n2</code> ; <code>y1</code> and <code>y2</code> for events of each group, <code>n1</code> and <code>n2</code> for sample size of each strata. The second group is usually the control group.
<code>conf.level</code>	confidence level

**Details**

It calculates risk difference and its score confidence interval between two groups. The confidence interval is asymmetric and there is no standard error in the output. If you need percent scale, multiply the output by 100. This supports stratification. This implementation uses `nlminb` minimization function.

**Value**

The following output will be returned for each strata and common value. There is no standard error.

<code>p1</code>	proportion from the first group, $y1/n1$
<code>p2</code>	proportion from the second group, $y2/n2$
<code>RD</code>	risk difference, $p1 - p2$ . Point estimate of common RD is calculated with MH weight.
<code>lower</code>	lower confidence limit of RD
<code>upper</code>	upper confidence limit of RD

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**References**

Miettinen O, Nurminen M. Comparative analysis of two rates. *Stat Med* 1985;4:213-26

**See Also**

[RDmn1](#), [RRmn1](#), [ORmn1](#), [RDmn](#), [RRmn](#), [ORmn](#), [RRinv](#), [ORinv](#), [ORcmh](#)

**Examples**

```
d1 = matrix(c(25, 339, 28, 335, 23, 370, 40, 364), nrow=2, byrow=TRUE)
colnames(d1) = c("y1", "n1", "y2", "n2")
RDinv(d1)
```

---

RDmn	<i>Risk Difference and Score CI between two groups with strata by MN method</i>
------	---

---

### Description

Risk difference and its score confidence interval between two groups with stratification by Miettinen and Nurminen method

### Usage

```
RDmn(d0, conf.level=0.95, eps=1e-8)
```

### Arguments

d0	A data.frame or matrix, of which each row means a strata. This should have four columns named y1, n1, y2, and n2; y1 and y2 for events of each group, n1 and n2 for sample size of each strata. The second group is usually the control group.
conf.level	confidence level
eps	absolute value less than eps is regarded as negligible

### Details

It calculates risk difference and its score confidence interval between two groups. The confidence interval is asymmetric and there is no standard error in the output. If you need percent scale, multiply the output by 100. This supports stratification. This implementation uses nlminb minimization function.

### Value

The following output will be returned for each strata and common value. There is no standard error.

p1	proportion from the first group, y1/n1
p2	proportion from the second group, y2/n2
RD	risk difference, p1 - p2. Point estimate of common RD is calculated with MN weight.
lower	lower confidence limit of RD
upper	upper confidence limit of RD

### Author(s)

Kyun-Seop Bae k@acr.kr

### References

Miettinen O, Nurminen M. Comparative analysis of two rates. Stat Med 1985;4:213-26

### See Also

[RDmn1](#), [RRmn1](#), [ORMn1](#), [RRmn](#), [ORMn](#), [RDinv](#), [RRinv](#), [ORinv](#), [ORcmh](#)

**Examples**

```
d1 = matrix(c(25, 339, 28, 335, 23, 370, 40, 364), nrow=2, byrow=TRUE)
colnames(d1) = c("y1", "n1", "y2", "n2")
RDmn(d1)
```

---

RDmn1	<i>Risk Difference and Score CI between two groups without strata by MN method</i>
-------	--

---

**Description**

Risk difference and its score confidence interval between two groups without stratification

**Usage**

```
RDmn1(y1, n1, y2, n2, conf.level=0.95, eps=1e-8)
```

**Arguments**

y1	positive event count of test (the first) group
n1	total count of the test (the first) group
y2	positive event count of control (the second) group
n2	total count of control (the second) group
conf.level	confidence level
eps	absolute value less than eps is regarded as negligible

**Details**

It calculates risk difference and its score confidence interval between two groups. The confidence interval is asymmetric and there is no standard error in the output. If you need percent scale, multiply the output by 100. This does not support stratification. This implementation uses nlminb minimization function.

**Value**

The result is a data.frame. There is no standard error.

p1	proportion from the first group, y1/n1
p2	proportion from the second group, y2/n2
RD	risk difference, p1 - p2
lower	lower confidence limit of RD
upper	upper confidence limit of RD

**Author(s)**

Kyun-Seop Bae k@acr.kr

**References**

Miettinen O, Nurminen M. Comparative analysis of two rates. Stat Med 1985;4:213-26

See Also

[RRmn1](#), [ORMn1](#), [RDmn](#), [RRmn](#), [ORMn](#)

Examples

```
RDmn1(104, 11037, 189, 11034)
```

---

REG	<i>Regression of Linear Least Square, similar to SAS PROC REG</i>
-----	---

---

Description

REG is similar to SAS PROC REG.

Usage

```
REG(Formula, Data, conf.level=0.95, HC=FALSE, Weights=1, summarize=TRUE)
```

Arguments

Formula	a conventional formula for a linear model
Data	a data.frame to be analyzed
conf.level	confidence level for the confidence limit
HC	heteroscedasticity related output is required such as HC0, HC3, White’s first and second moment specification test
Weights	weights for the each observation or residual square. This is usually the inverse of each variance.
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.

The first part is ANOVA table.

The second part is measures about fitness.

The third part is the estimates of coefficients.

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
Lower CL	lower confidence limit with conf.level
Upper CL	lower confidence limit with conf.level
Df	degree of freedom

t value	value for t distribution
Pr(> t )	probability of larger than absolute t value from t distribution with residual's degree of freedom

The above result is repeated using HC0 and HC3, with following White's first and second moment specification test, if HC option is specified. The statistic t values and their p values with HC1 and HC2 are between those of HC0 and H3.

If summarize=FALSE, REG returns;

coefficients	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, C02)
REG(uptake ~ conc, C02, 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), DFFITS, and COVRATIO.
DFBETAS	Column names are the names of coefficients. Each row shows how much each coefficient is affected by deleting the corresponding row of observation.

**Author(s)**

Kyun-Seop Bae k@acr.kr

**Examples**

```
regD(uptake ~ conc, C02)
```

---

RR	<i>Relative Risk of two groups</i>
----	------------------------------------

---

**Description**

Relative Risk between two groups

**Usage**

```
RR(y1, n1, y2, n2, conf.level=0.95)
```

**Arguments**

y1	positive event count of test (the first) group
n1	total count of the test (the first) group
y2	positive event count of control (the second) group
n2	total count of control (the second) group
conf.level	confidence level

**Details**

It calculates relative risk of two groups. No continuity correction here. If you need percent scale, multiply the output by 100.

**Value**

The result is a data.frame.

p1	proportion from the first group
p2	proportion from the second group
RR	relative risk, p1/p2
SElog	standard error of log(RR)
lower	lower confidence limit of RR
upper	upper confidence limit of RR

**Author(s)**

Kyun-Seop Bae k@acr.kr

**See Also**[RD](#), [OR](#), [RDmn1](#), [RRmn1](#), [ORmn1](#), [RDmn](#), [RRmn](#), [ORmn](#)**Examples**

```
RR(104, 11037, 189, 11034) # no continuity correction
```

---

RRinv	<i>Relative Risk of two groups with strata by inverse variance method</i>
-------	---

---

**Description**

Relative risk and its score confidence interval of two groups with stratification by inverse variance method

**Usage**

```
RRinv(d0, conf.level=0.95)
```

**Arguments**

d0	A data.frame or matrix, of which each row means a strata. This should have four columns named y1, n1, y2, and n2; y1 and y2 for events of each group, n1 and n2 for sample size of each strata. The second group is usually the control group.
conf.level	confidence level

**Details**

It calculates relative risk and its score confidence interval of two groups. The confidence interval is asymmetric and there is no standard error in the output. This supports stratification. This implementation uses nlminb minimization function which usually gives about 7 significant digits. PropCIs::riskscoreci function uses cubic equation approximation which gives only about 2 significant digits.

**Value**

The following output will be returned for each strata and common value. There is no standard error.

p1	proportion from the first group, y1/n1
p2	proportion from the second group, y2/n2
RR	relative risk, p1/p2. Point estimate of common RR is calculated with MH weight.
lower	lower confidence limit of RR
upper	upper confidence limit of RR

**Author(s)**

Kyun-Seop Bae k@acr.kr

## References

Miettinen O, Nurminen M. Comparative analysis of two rates. Stat Med 1985;4:213-26

## See Also

[RDmn1](#), [RRmn1](#), [ORmn1](#), [RDmn](#), [RRmn](#), [ORmn](#), [RDinv](#), [ORinv](#), [ORcmh](#)

## Examples

```
d1 = matrix(c(25, 339, 28, 335, 23, 370, 40, 364), nrow=2, byrow=TRUE)
colnames(d1) = c("y1", "n1", "y2", "n2")
RRinv(d1)
```

---

RRmn	<i>Relative Risk and Score CI of two groups with strata by MN method</i>
------	--

---

## Description

Relative risk and its score confidence interval of two groups with stratification by Miettinen and Nurminen method

## Usage

```
RRmn(d0, conf.level=0.95, eps=1e-8)
```

## Arguments

d0	A data.frame or matrix, of which each row means a strata. This should have four columns named y1, n1, y2, and n2; y1 and y2 for events of each group, n1 and n2 for sample size of each strata. The second group is usually the control group.
conf.level	confidence level
eps	absolute value less than eps is regarded as negligible

## Details

It calculates relative risk and its score confidence interval of two groups. The confidence interval is asymmetric and there is no standard error in the output. This supports stratification. This implementation uses nlminb minimization function which usually gives about 7 significant digits. PropCIs::riskscoreci function uses cubic equation approximation which gives only about 2 significant digits.

## Value

The following output will be returned for each strata and common value. There is no standard error.

p1	proportion from the first group, y1/n1
p2	proportion from the second group, y2/n2
RR	relative risk, p1/p2. Point estimate of common RR is calculated with MN weight.
lower	lower confidence limit of RR
upper	upper confidence limit of RR



**Author(s)**

Kyun-Seop Bae k@acr.kr

**References**

Miettinen O, Nurminen M. Comparative analysis of two rates. Stat Med 1985;4:213-26

**See Also**

[RDmn1](#), [RRmn1](#), [ORmn1](#), [RDmn](#), [ORmn](#), [RDinv](#), [RRinv](#), [ORinv](#), [ORcmh](#)

**Examples**

```
d1 = matrix(c(25, 339, 28, 335, 23, 370, 40, 364), nrow=2, byrow=TRUE)
colnames(d1) = c("y1", "n1", "y2", "n2")
RRmn(d1)
```

---

RRmn1	<i>Relative Risk and Score CI of two groups without strata by MN method</i>
-------	---

---

**Description**

Relative risk and its score confidence interval of two groups without stratification

**Usage**

```
RRmn1(y1, n1, y2, n2, conf.level=0.95, eps=1e-8)
```

**Arguments**

y1	positive event count of test (the first) group
n1	total count of the test (the first) group
y2	positive event count of control (the second) group
n2	total count of control (the second) group
conf.level	confidence level
eps	absolute value less than eps is regarded as negligible

**Details**

It calculates relative risk and its score confidence interval of two groups. The confidence interval is asymmetric and there is no standard error in the output. This does not support stratification. This implementation uses nlminb minimization function which usually gives about 7 significant digits. PropCIs::riskscoreci function uses cubic equation approximation which gives only about 2 significant digits.

**Value**

The result is a data.frame. There is no standard error.

p1	proportion from the first group, $y_1/n_1$
p2	proportion from the second group, $y_2/n_2$
RR	relative risk, $p_1/p_2$
lower	lower confidence limit of RR
upper	upper confidence limit of RR

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**References**

Miettinen O, Nurminen M. Comparative analysis of two rates. Stat Med 1985;4:213-26

**See Also**

[RDmn1](#), [ORmn1](#), [RDmn](#), [RRmn](#), [ORmn](#)

**Examples**

```
RRmn1(104, 11037, 189, 11034)
```

---

satt

*Satterthwaite Approximation of 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	approximated variance
Df	degree of freedom

**Author(s)**

Kyun-Seop Bae k@acr.kr

---

ScoreCI

*Score CI of a proportion*

---

**Description**

Score confidence of a proportion in one group

**Usage**

```
ScoreCI(y, n, conf.level=0.95)
```

**Arguments**

y	positive event count of a group
n	total count of a group
conf.level	confidence level

**Details**

It calculates score confidence interval of a proportion in one group. The confidence interval is asymmetric and there is no standard error in the output. If you need percent scale, multiply the output by 100.

**Value**

The result is a data.frame. There is no standard error.

Prop	proportion from the first group
lower	lower confidence limit of Prop
upper	upper confidence limit of Prop

**Author(s)**

Kyun-Seop Bae k@acr.kr

**See Also**

[binom.test](#)

**Examples**

```
ScoreCI(104, 11037)
```

---

SD	<i>Standard Deviation</i>
----	---------------------------

---

**Description**

Standard deviation of sample.

**Usage**

$SD(y)$

**Arguments**

$y$  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	<i>Standard Error of the Sample Mean</i>
-----	--

---

**Description**

The estimate of the standard error of the sample mean

**Usage**

$SEM(y)$

**Arguments**

$y$  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	<i>Skewness</i>
----------	-----------------

---

**Description**

Skewness with a conventional formula.

**Usage**

`Skewness(y)`

**Arguments**

`y` a vector of numerics

**Details**

It removes NA in the input vector.

**Value**

Estimate of skewness

**Author(s)**

Kyun-Seop Bae [k@acr.kr](mailto:k@acr.kr)

**See Also**

[SkewnessSE](#)

---

SkewnessSE	<i>Standard Error of Skewness</i>
------------	-----------------------------------

---

**Description**

Standard error of the skewness with a conventional formula.

**Usage**

`SkewnessSE(y)`

**Arguments**

`y` 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	<i>F Test with Slice</i>
-------	--------------------------

---

**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. Row names are the levels of the slice term.

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)	probability of larger than F value

**Author(s)**

Kyun-Seop Bae k@acr.kr

**Examples**

```
SLICE(uptake ~ Type*Treatment, C02, "Type", "Treatment")
SLICE(uptake ~ Type*Treatment, C02, "Treatment", "Type")
```

---

SS	<i>Sum of Square</i>
----	----------------------

---

**Description**

Sum of squares with ANOVA.

**Usage**

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

**Arguments**

x	a result of <code>ModelMatrix</code> containing design information
rx	a result of <code>lfit</code>
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](mailto:k@acr.kr)

**See Also**

[ModelMatrix](#), [lfit](#)

---

T3MS	<i>Type III Expected Mean Square Formula</i>
------	--

---

**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	<i>Test Type III SS using error term other than MSE</i>
--------	---

---

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	<i>Trimmed Mean</i>
-------------	---------------------

---

**Description**

Trimmed mean wrapping mean function .

**Usage**

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

**Arguments**

y	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	<i>Table Summary</i>
------	----------------------

---

**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	<i>Table Summary 0 independent(x) variable</i>
-------	--

---

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	<i>Table Summary 1 independent(x) variable</i>
-------	--

---

**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 variable.

**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](mailto: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	<i>Table Summary 2 independent(x) variables</i>
-------	---

---

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 variables; 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 variables; 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(y, conf.level=0.95)
```

**Arguments**

y	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(y, conf.level = 0.95)
```

**Arguments**

y	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
IQR	inter-quartile range type 2 which is SAS default
MAD	mean absolute deviation
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(1h)
```

---

WhiteTest

*White's Model Specification Test*

---

**Description**

This is shown in SAS PROC REG as the Test of First and Second Moment Specification.

**Usage**

```
WhiteTest(rx)
```

**Arguments**

rx                      a result of lm

**Details**

This is also called as White's general test for heteroskedasticity.

**Value**

Returns a direct test results by more coomplex theorem 2 , not by simpler corollary 1.

**Author(s)**

Kyun-Seop Bae k@acr.kr

**References**

White H. A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. *Econometrica* 1980;48(4):817-838.

**Examples**

```
WhiteTest(lm(mpg ~ disp, mtcars))
```



# Index

## \* Descriptive Statistics

UNIV, [71](#)

## \* Table Summary

tsum, [65](#)

tsum0, [66](#)

tsum1, [67](#)

tsum2, [68](#)

tsum3, [69](#)

## \* datasets

BEdata, [8](#)

af, [4](#)

aov1, [5](#)

aov2, [6](#)

aov3, [7](#)

BEdata, [8](#)

binom.test, [59](#)

bk, [8](#)

BY, [9](#)

CIest, [10](#)

Coll, [11](#)

CONTR, [12](#), [15](#)

Cor.test, [13](#)

corFisher, [14](#)

cSS, [12](#), [15](#)

CV, [16](#)

Diffogram, [16](#), [33](#), [45](#)

e1, [17](#)

e2, [18](#)

e3, [18](#)

EMS, [19](#)

est, [20](#), [21](#)

ESTM, [20](#), [21](#)

estmb, [22](#)

ESTMIMATE (ESTM), [21](#)

G2SWEEP, [22](#), [22](#)

geoCV, [24](#), [25](#)

geoMean, [24](#), [24](#)

GLM, [25](#)

is.cor, [26](#)

Kurtosis, [27](#), [28](#)

KurtosisSE, [27](#), [27](#)

LCL, [28](#)

lfit, [23](#), [29](#), [63](#)

lr, [30](#), [53](#)

lr0, [31](#)

LSM, [17](#), [32](#), [45](#)

Max, [33](#)

Mean, [34](#)

Median, [34](#)

Min, [35](#)

ModelMatrix, [23](#), [29](#), [35](#), [63](#)

N, [36](#)

OR, [37](#), [48](#), [55](#)

ORcmh, [38](#), [39](#), [41](#), [49](#), [50](#), [56](#), [57](#)

ORinv, [38](#), [39](#), [41](#), [49](#), [50](#), [56](#), [57](#)

ORMn, [37–39](#), [40](#), [42](#), [48–50](#), [52](#), [55–58](#)

ORMn1, [37–39](#), [41](#), [41](#), [48–50](#), [52](#), [55–58](#)

pB, [42](#)

Pcor.test, [43](#)

pD, [44](#)

PDIFF, [17](#), [20](#), [33](#), [44](#)

pResD, [46](#)

QuartileRange, [46](#)

Range, [47](#)

RD, [37](#), [47](#), [55](#)

RDinv, [38](#), [39](#), [41](#), [48](#), [50](#), [56](#), [57](#)

RDmn, [37–39](#), [41](#), [42](#), [48](#), [49](#), [50](#), [52](#), [55–58](#)

RDmn1, [37–39](#), [41](#), [42](#), [48–50](#), [51](#), [55–58](#)

REG, [52](#)

regD, [53](#)

RR, [37](#), [48](#), [54](#)

RRinv, [38](#), [39](#), [41](#), [49](#), [50](#), [55](#), [57](#)

RRmn, [37–39](#), [41](#), [42](#), [48–50](#), [52](#), [55](#), [56](#), [56](#), [58](#)

RRmn1, [37–39](#), [41](#), [42](#), [48–50](#), [52](#), [55–57](#), [57](#)

sasLM (sasLM-package), [3](#)

sasLM-package, 3  
satt, 58  
ScoreCI, 59  
SD, 60  
SEM, 60  
Skewness, 61, 62  
SkewnessSE, 61, 61  
SLICE, 62  
SS, 63  
  
T3MS, 63  
T3test, 64  
trimmedMean, 65  
tsum, 65, 67–70  
tsum0, 9, 66, 66, 68–70  
tsum1, 9, 66, 67, 67, 69, 70  
tsum2, 9, 66–68, 68, 70  
tsum3, 9, 66–69, 69  
  
UCL, 28, 70  
UNIV, 71  
  
WhiteTest, 72