Package 'sasLM'

June 15, 2026

Version 1.0.0

Title 'SAS' Linear Model

Description This is a core implementation of 'SAS' procedures for linear models - GLM, REG, ANOVA, TTEST, 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 do not necessarily mean incorrectness. However, many want the same results as 'SAS'. This package aims to achieve that. Reference: Littell RC, Stroup WW, Freund RJ (2002, ISBN:0-471-22174-0).

Depends R (>= 3.5.0), mvtnorm

Imports methods

Suggests MASS, knitr, rmarkdown

Author Kyun-Seop Bae [aut, cre]

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

License GPL-3

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

NeedsCompilation no

Repository CRAN

Date/Publication 2026-06-15 05:10:03 UTC

Contents

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'. A different result does not necessarily mean incorrectness. However, many want the same results as '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
RanTest(formula1, BEdata, Random="SUBJ") # Hypothesis test with SUBJ as random
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 a `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 *ANOVA with Type I SS*

Description

ANOVA with Type I SS.

Usage

```
aov1(Formula, Data, BETA=FALSE, Resid=FALSE)
```

Arguments

Formula	a conventional formula for a linear model.
Data	a data.frame to be analyzed
BETA	if TRUE, coefficients (parameters) of REG will be returned. This is equivalent to the SOLUTION option of SAS PROC GLM.
Resid	if TRUE, fitted values (y hat) and residuals will be returned

Details

It performs the core function of SAS PROC GLM, and returns Type I SS. This also accepts continuous independent variables.

Value

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

Df	degrees of freedom
Sum Sq	sum of squares for the set of contrasts
Mean Sq	mean square
F value	F value for the F distribution
Pr(>F)	probability of a larger F value

The next returns are optional.

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 the BETA=TRUE option.
Fitted	Fitted values or y hat. This is returned only with the Resid=TRUE option.
Residual	Weighted residuals. This is returned only with the Resid=TRUE option.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
aov1(uptake ~ Plant + Type + Treatment + conc, CO2)
aov1(uptake ~ Plant + Type + Treatment + conc, CO2, BETA=TRUE)
aov1(uptake ~ Plant + Type + Treatment + conc, CO2, Resid=TRUE)
aov1(uptake ~ Plant + Type + Treatment + conc, CO2, BETA=TRUE, Resid=TRUE)
```

aov2 *ANOVA with Type II SS*

Description

ANOVA with Type II SS.

Usage

```
aov2(Formula, Data, BETA=FALSE, Resid=FALSE)
```

Arguments

Formula	a conventional formula for a linear model.
Data	a data.frame to be analyzed
BETA	if TRUE, coefficients (parameters) of REG will be returned. This is equivalent to the SOLUTION option of SAS PROC GLM.
Resid	if TRUE, fitted values (y hat) and residuals will be returned

Details

It performs the core function of SAS PROC GLM, and returns Type II SS. This also accepts continuous independent variables.

Value

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

Df	degrees of freedom
Sum Sq	sum of squares for the set of contrasts
Mean Sq	mean square
F value	F value for the F distribution
Pr(>F)	probability of a larger F value

The next returns are optional.

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 the BETA=TRUE option.
Fitted	Fitted values or y hat. This is returned only with the Resid=TRUE option.
Residual	Weighted residuals. This is returned only with the Resid=TRUE option.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
aov2(uptake ~ Plant + Type + Treatment + conc, CO2)
aov2(uptake ~ Plant + Type + Treatment + conc, CO2, BETA=TRUE)
aov2(uptake ~ Plant + Type + Treatment + conc, CO2, Resid=TRUE)
aov2(uptake ~ Plant + Type + Treatment + conc, CO2, BETA=TRUE, Resid=TRUE)
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, BETA=FALSE, Resid=FALSE)
```

Arguments

Formula	a conventional formula for a linear model.
Data	a data.frame to be analyzed
BETA	if TRUE, coefficients (parameters) of REG will be returned. This is equivalent to the SOLUTION option of SAS PROC GLM.
Resid	if TRUE, fitted values (y hat) and residuals will be returned

Details

It performs the core function of SAS PROC GLM, and returns Type III SS. This also accepts continuous independent variables.

Value

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

Df	degrees of freedom
Sum Sq	sum of squares for the set of contrasts
Mean Sq	mean square
F value	F value for the F distribution
Pr(>F)	probability of a larger F value

The next returns are optional.

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 the BETA=TRUE option.
Fitted	Fitted values or y hat. This is returned only with the Resid=TRUE option.
Residual	Weighted residuals. This is returned only with the Resid=TRUE option.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
aov3(uptake ~ Plant + Type + Treatment + conc, CO2)
aov3(uptake ~ Plant + Type + Treatment + conc, CO2, BETA=TRUE)
aov3(uptake ~ Plant + Type + Treatment + conc, CO2, Resid=TRUE)
aov3(uptake ~ Plant + Type + Treatment + conc, CO2, BETA=TRUE, Resid=TRUE)
```

---

aspirinCHD *An example dataset for meta-analysis - aspirin in coronary heart disease*

Description

The data is from 'Canner PL. An overview of six clinical trials of aspirin in coronary heart disease. Stat Med. 1987'

Usage

```
aspirinCHD
```

Format

A data frame with 6 rows.

`y1` death event count of aspirin group

`n1` total subjects of the aspirin group

`y2` death event count of placebo group

`n2` total subjects of the placebo group

Details

This data is for educational purposes.

References

Canner PL. An overview of six clinical trials of aspirin in coronary heart disease. Stat Med. 1987;6:255-263.

BEdata *An Example Dataset of a 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 real data from a 2x2 bioequivalence study, which has 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 integers are somewhat annoying. This removes them from 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`, or `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

Functions such as GLM, REG, and aov1 can be run by levels of a variable.

Usage

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

Arguments

FUN	Function name to be called, such as GLM or 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 the BY clause of SAS procedures.

Value

a list of FUN function outputs. The names of the list are the levels of the By variable.

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 the point estimate and its confidence interval with a given contrast and alpha value using the 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. Levels are alphabetically ordered by default.
conf.level	confidence level of the confidence interval

Details

Get the point estimate and its confidence interval with a given contrast and alpha value using the 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 the t distribution
Df	degrees of freedom
Pr(>|t|)	probability of a larger absolute t value from the t distribution with the residual degrees 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

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

Usage

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

Arguments

`Formula`	formula of the model
`Data`	input data as a matrix or a `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 an 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 an 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 the sum of squares and its F value and p-value.

Df	degrees of freedom
Sum Sq	sum of squares for the set of contrasts
Mean Sq	mean square
F value	F value for the F distribution
Pr(>F)	probability of a larger 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` *Correlation test of multiple numeric columns*

Description

Testing correlation between numeric columns of data with the 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 the 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`	lower confidence limit
`Upper CL`	upper confidence limit
`t value`	t value of the t distribution
`Df`	degrees 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

`x`	the first input numeric vector
`y`	the second input numeric vector
`conf.level`	confidence level
`rho`	population correlation rho under the null hypothesis

Details

This accepts only two numeric vectors.

Value

`N`	sample size, length of input vectors
`r`	sample correlation
`Fisher.z`	Fisher's z
`bias`	bias to correct
`rho.hat`	point estimate of population rho
`conf.level`	confidence level for the confidence interval
`lower`	lower limit of confidence interval
`upper`	upper limit of confidence interval
`rho0`	population correlation rho under the null hypothesis
`p.value`	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 the sum of squares of a contrast from an `lfit` result.

Usage

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

Arguments

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

Details

It calculates the sum of squares with a given contrast matrix and an `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 the sum of squares and its F value and p-value.

`Df`	degrees of freedom
`Sum Sq`	sum of squares for the set of contrasts
`Mean Sq`	mean square
`F value`	F value for the F distribution
`Pr(>F)`	probability of a larger 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
GLM(uptake ~ Type, CO2) # compare with the above
```

CumAlpha *Cumulative Alpha with various z's and ti*

Description

Cumulative alpha values for repeated hypothesis tests with changing bound z-values and times of test (ti).

Usage

```
CumAlpha(z, side=2, ti=NULL, c0=NULL, Seed=5)
```

Arguments

z	vector of upper z-value bounds for the repeated hypothesis test
side	1=one-sided test, 2=two-sided test
ti	vector of times (or information amount) of test. All values should be in [0, 1] and sorted. If not specified, equal intervals are assumed.
c0	correlation matrix. If not specified, Brownian motion is assumed.
Seed	seed value for the `mvtnorm::pmvnorm` function

Details

It calculates cumulative alpha-values for the repeated hypothesis test with a vector of upper bound z-values. If the times of test are not specified, linear (proportional) increase of information amount and Brownian motion of z-values are assumed, i.e. the correlation is sqrt(t_i/t_j).

Value

The result is a matrix.

ti	time of test
cum.alpha	cumulative alpha values

Author(s)

Kyun-Seop Bae k@acr.kr

References

Reboussin DM, DeMets DL, Kim K, Lan KKG. Computations for group sequential boundaries using the Lan-DeMets function method. Controlled Clinical Trials. 2000;21:190-207.

Examples

```
CumAlpha(z=rep(qnorm(1 - 0.05/2), 10)) # two-side Z-test with alpha=0.05 for ten times
```

CV *Coefficient of Variation in percentage*

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 *Plot Pairwise Differences*

Description

Plot pairwise differences by a common method.

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 the 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 the PDIFF option of SAS PROC GLM. For the adjustment method "dunnett", see the 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)")
```

Drift *Drift defined by Lan and DeMets for Group Sequential Design*

Description

Calculate the drift value with given upper bounds (z-values), times of test, and power.

Usage

```
Drift(bi, ti=NULL, Power=0.9)
```

Arguments

bi	upper bound z-values
ti	times of test. These should be in the range of [0, 1]. If omitted, equal intervals are assumed.
Power	target power at the final test

Details

It calculates the drift value with given upper bound z-values, times of test, and power. If the times of test are not given, equal intervals are assumed. `mvtnorm::pmvt` (with noncentrality) is better than `pmvnorm` in calculating power and sample size. However, Lan-DeMets used the multivariate normal rather than the multivariate noncentral t distribution. This function follows Lan-DeMets for consistency with previous results.

Value

Drift value for the given condition

Author(s)

Kyun-Seop Bae k@acr.kr

References

Reboussin DM, DeMets DL, Kim K, Lan KKG. Computations for group sequential boundaries using the Lan-DeMets function method. Controlled Clinical Trials. 2000;21:190-207.

Examples

```
Drift(seqBound(ti=(1:5)/5)[, "up.bound"])
```

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

Description

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

Usage

```
e1(XpX, eps=1e-8)
```

Arguments

`XpX`	the crossproduct of a design or model matrix. This should have appropriate column names.
`eps`	A value less than this is considered zero.

Details

It makes a contrast matrix for Type I SS. If `zapsmall` is used, the result becomes less accurate.

Value

A contrast matrix for Type I SS.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
x = ModelMatrix(uptake ~ Plant + Type + Treatment + conc, CO2)
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	A value less than this is considered zero.

Details

It makes a contrast matrix for Type II SS. If zapsmall is used, the result becomes less accurate.

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, CO2)), 12)
round(e2(ModelMatrix(uptake ~ Type, CO2)), 12)
round(e2(ModelMatrix(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(x, eps=1e-8)
```

Arguments

x	an output of ModelMatrix
eps	A value less than this is considered zero.

Details

It makes a contrast matrix for Type III SS. If zapsmall is used, the result becomes less accurate.

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 *Expected Mean Square Formula*

Description

Calculates a formula table for the 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	A value less than this is considered zero.

Details

This is necessary for further hypothesis tests 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 the lfit function
conf.level	confidence level of the 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, the L matrix is for the pairwise comparison such as that of the PDIFF function.

Details

It tests rows of linear functions. A linear function means a linear combination of estimated coefficients. It corresponds to the ESTIMATE statement of SAS PROC GLM. The 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 the "lsd" method
`Upper CL`	upper confidence limit by the "lsd" method
`Std. Error`	standard error of the point estimate
`t value`	value for the t distribution, for methods other than "scheffe"
`F value`	value for the F distribution, for the "scheffe" method only
`Df`	degrees of freedom of the residuals
`Pr(>|t|)`	probability of a larger absolute t value from the t distribution with the residual degrees of freedom, for methods other than "scheffe"
`Pr(>F)`	probability of a larger F value from the F distribution with the residual degrees of freedom, for the "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) # Quebec - 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 function rows to be tested
Formula	a conventional formula for a linear model
Data	a data.frame to be analyzed
conf.level	confidence level of the confidence limit

Details

It tests rows of linear functions. A linear function means a linear combination of estimated coefficients. It is similar to the ESTIMATE statement of SAS PROC GLM. This is a convenient version of the 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 the t distribution
Df	degrees of freedom
Pr(>|t|)	probability of a larger absolute t value from the t distribution with the residual degrees 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

Checks 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`	An absolute value less than this is considered to be zero.

Details

It checks the estimability of L, row vectors of coefficients. This corresponds to the ESTIMATE statement of SAS PROC GLM. 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 rows are estimable (as TRUE)

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`G2SWEEP`

ExitP *Exit Probability with Cumulative Z-test in Group Sequential Design*

Description

Exit probabilities with the given drift, upper bounds, and times of test.

Usage

```
ExitP(Theta, bi, ti=NULL)
```

Arguments

Theta	drift value defined by Lan-DeMets. See the reference.
bi	upper bound z-values
ti	times of test. These should be in the range of [0, 1]. If omitted, even intervals are assumed.

Details

It calculates exit probabilities and cumulative exit probabilities with the given drift, upper z-bounds, and times of test. If the times of test are not given, even intervals are assumed. `mvtnorm::pmvt` (with noncentrality) is better than `pmvnorm` in calculating power and sample size. However, Lan-DeMets used the multivariate normal rather than the multivariate noncentral t distribution. This function follows Lan-DeMets for consistency with previous results.

Value

The result is a matrix.

ti	time of test
bi	upper z-bound
cum.alpha	cumulative alpha-value

Author(s)

Kyun-Seop Bae k@acr.kr

References

Reboussin DM, DeMets DL, Kim K, Lan KKG. Computations for group sequential boundaries using the Lan-DeMets function method. Controlled Clinical Trials. 2000;21:190-207.

Examples

```
b0 = seqBound(ti=(1:5)/5)[, "up.bound"]
ExitP(Theta = Drift(b0), bi = b0)
```

g2inv *Generalized type 2 inverse matrix, g2 inverse*

Description

A generalized inverse is usually not unique. Some programs use this algorithm to get a unique generalized inverse matrix. This uses the SWEEP operator and works for non-square matrices also.

Usage

```
g2inv(A, eps=1e-08)
```

Arguments

A	a matrix to be inverted
eps	A value less than this is considered zero.

Details

See 'SAS Technical Report R106, The Sweep Operator: Its Importance in Statistical Computing' by J. H. Goodnight for details.

Value

g2 inverse

Author(s)

Kyun-Seop Bae k@acr.kr

References

Searle SR, Khuri AI. Matrix Algebra Useful for Statistics. 2e. John Wiley and Sons Inc. 2017.

See Also

G2SWEEP

Examples

```
A = matrix(c(1, 2, 4, 3, 3, -1, 2, -2, 5, -4, 0, -7), byrow=TRUE, ncol=4) ; A
g2inv(A)
```

G2SWEEP *Generalized inverse matrix of type 2 for linear regression*

Description

A 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. If A is not a square matrix, G2SWEEP calls the g2inv function.
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 details. If the input matrix A is not a square matrix, the Augmented option cannot be TRUE.
eps	A value less than this is considered zero.

Details

The generalized inverse of g2-type is used by some software to do linear regression. See 'SAS Technical Report R106, The Sweep Operator: Its Importance in Statistical Computing' by J. H. Goodnight for details.

Value

when Augmented=FALSE
: ordinary g2 inverse

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

attribute "rank"
: the rank of the 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(y)
```

Arguments

y	a numeric vector

Details

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

Value

Geometric coefficient of variation in percentage.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

geoMean

Examples

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

---

geoMean *Geometric Mean without NA*

Description

Geometric 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

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, Resid=FALSE, conf.level=0.95,
    Weights=1)
```

Arguments

Formula	a conventional formula for a linear model.
Data	a data.frame to be analyzed
BETA	if TRUE, coefficients (parameters) of REG will be returned. This is equivalent to the SOLUTION option of SAS PROC GLM
EMEAN	if TRUE, least square means (or expected means) will be returned. This is equivalent to the LSMEANS statement of SAS PROC GLM
Resid	if TRUE, fitted values (y hat) and residuals will be returned
conf.level	confidence level for the confidence limit of the least square mean
Weights	weights for the weighted least squares. This should be a scalar or a vector of the same length as the number of rows of Data. Observations with nonpositive weights are excluded from the analysis as SAS does.

Details

It performs the core function of SAS PROC GLM. Least square means for the interaction term of three variables are 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 squares table
Type II	Type II sum of squares table
Type III	Type III sum of squares 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 the BETA=TRUE option.
Expected Mean	Least square (or expected) mean table with confidence limits. This is returned only with the EMEAN=TRUE option.

Fitted	Fitted values or y hat in the original scale, as SAS OUTPUT P= does, even with Weights. This is returned only with the Resid=TRUE option.
Residual	Residuals in the original scale, as SAS OUTPUT R= does, even with Weights. This is returned only with the Resid=TRUE option.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

REG, aov1, aov2, aov3, LSM, PDIFF

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

---

is.cor *Is it a correlation matrix?*

Description

Tests 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. An absolute value less than this is considered zero.

Details

A diagonal component does not need to be exactly 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(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 *Standard Error of Kurtosis*

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

See Also

Kurtosis

LCL *Lower Confidence Limit*

Description

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

Usage

```
LCL(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 lower bound of the confidence limit using the t-distribution

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

UCL

`lfit` *Linear Fit*

Description

Fits a least squares linear model.

Usage

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

Arguments

`x`	a result of `ModelMatrix`
`y`	a column vector of the response (dependent) variable
`eps`	A value less than this is considered zero.

Details

A minimal version of the least squares fit of a linear model

Value

`coefficients`	beta coefficients
`g2`	g2 inverse
`rank`	rank of the model matrix
`DFr`	degrees of freedom for the residual
`SSE`	sum of squares error
`SST`	sum of squares total
`DFr2`	degrees of freedom of the residual for the beta coefficient

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 are calculated with the g2 inverse. The 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`	A value less than this is considered zero.

Details

It uses `G2SWEEP` to get the 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 the t distribution
`Pr(>|t|)`	probability of a larger absolute t value from the t distribution with the residual degrees 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 in multiple linear regression is to perform simple linear regressions with each single independent variable.

Usage

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

Arguments

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

Details

It performs simple linear regression for each independent variable.

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
`Rsq`	R-squared for the simple linear model
`Pr(>F)`	p-value of the 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 Squares Means*

Description

Estimates least squares means using the g2 inverse.

Usage

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

Arguments

Formula	a conventional formula for a linear model
Data	a data.frame to be analyzed
Term	a 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", and "scheffe" are available. This does not affect the SE, Lower CL, and Upper CL of the output table.
hideNonEst	logical. whether to hide non-estimable values
PLOT	logical. whether to plot LSMs and their confidence intervals
descend	logical. This specifies whether the plotting order is ascending or descending.
...	arguments to be passed to plot

Details

It is equivalent to the LSMEANS statement of SAS PROC GLM. The result of the second example below may be different from emmeans. This is because SAS and this function calculate the mean of the transformed continuous variable. However, emmeans calculates the average before the transformation. An interaction of three variables is not supported yet. For the "dunnett" adjustment method, see the PDIFF function.

Value

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

Group	group character. This appears when the model is one-way ANOVA or when the Term or adj argument is provided.
LSmean	point estimate of the least squares mean
LowerCL	lower confidence limit at the given confidence level by the "lsd" method
UpperCL	upper confidence limit at the given confidence level by the "lsd" method
SE	standard error of the point estimate
Df	degrees of freedom of the 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(y)
```

Arguments

y	a vector of numerics

Details

It removes NA values from 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(y)
```

Arguments

y	a vector of numerics

Details

It removes NA values from 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(y)
```

Arguments

y	a vector of numerics

Details

It removes NA values from 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(y)
```

Arguments

`y`	a vector of numerics

Details

It removes `NA` values from 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, XpX=FALSE)
```

Arguments

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

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

Author(s)

Kyun-Seop Bae k@acr.kr

`mtest` *Independent two groups t-test similar to PROC TTEST with summarized input*

Description

This is comparable to SAS PROC TTEST, except that it uses summarized input (sufficient statistics).

Usage

```
mtest(m1, s1, n1, m0, s0, n0, conf.level=0.95)
```

Arguments

`m1`	mean of the first (test, active, experimental) group
`s1`	sample standard deviation of the first group
`n1`	sample size of the first group
`m0`	mean of the second (reference, control, placebo) group
`s0`	sample standard deviation of the second group
`n0`	sample size of the second group
`conf.level`	confidence level

Details

This uses summarized input. This also produces confidence intervals of means and variances by group.

Value

The output format is comparable to that of SAS PROC TTEST.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`TTEST`, `tmtest`, `ztest`

Examples

```
mtest(5.4, 10.5, 3529, 5.1, 8.9, 5190) # NEJM 388;15 p1386
```

N *Number of observations*

Description

Number of observations excluding NA values.

Usage

```
N(y)
```

Arguments

y	a vector of numerics

Details

It removes NA values from the input vector.

Value

Count of the observations

Author(s)

Kyun-Seop Bae k@acr.kr

OBFBound *O'Brien-Fleming bounds for cumulative Z-test in Group Sequential Design*

Description

Sequential O'Brien-Fleming upper bounds for the cumulative Z-test on accumulating data. Z values are correlated. This is usually used for group sequential design.

Usage

```
OBFBound(K, alpha=0.05, side=2, ti=NULL, c0=NULL)
```

Arguments

K	count of tests, including the final one
alpha	goal alpha value for the last test at time 0.
side	1=one-sided test, 2=two-sided test
ti	times for test. These should be in [0, 1]. If not specified, equal intervals are assumed.
c0	correlation matrix. If not specified, Brownian motion is assumed.

Details

It calculates O'Brien-Fleming upper z-bounds and cumulative alpha-values for the repeated test in group sequential design.

Value

The result is a matrix.

`ti`	time of test
`z`	O'Brien-Fleming upper z-bound
`cum.alpha`	cumulative alpha-value

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

seqBound, PocockBound

Examples

```
OBFBound(K=2)
OBFBound(K=3)
OBFBound(K=4)
OBFBound(K=5)
```

OR *Odds Ratio of two groups*

Description

Odds ratio between two groups

Usage

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

Arguments

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

Details

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

Value

The result is a `data.frame`.

`odd1`	odds from the first group, y1/(n1 - y1)
`odd2`	odds from the second group, y2/(n2 - y2)
`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 the CMH method*

Description

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

Usage

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

Arguments

`d0`	A `data.frame` or matrix, of which each row means a stratum. 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 stratum. The second group is usually the control group.
`conf.level`	confidence level

Details

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

Value

The following output will be returned for each stratum and the common value.

odd1	odds from the first group, y1/(n1 - y1)
odd2	odds from the second group, y2/(n2 - y2)
OR	odds ratio, odd1/odd2. The point estimate of the common OR is calculated with the MH weights.
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

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 *Odds Ratio of two groups with strata by the inverse variance method*

Description

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

Usage

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

Arguments

d0	A data.frame or matrix, of which each row means a stratum. 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 stratum. The second group is usually the control group.
conf.level	confidence level

Details

It calculates the odds ratio and its confidence interval of two groups by the inverse variance method. This supports stratification. This can be used for meta-analysis also.

Value

The following output will be returned for each stratum and the common value.

odd1	odds from the first group, y1/(n1 - y1)
odd2	odds from the second group, y2/(n2 - y2)
OR	odds ratio, odd1/odd2. The point estimate of the common OR is calculated with the inverse variance weights.
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

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 the MN method*

Description

Odds ratio and its score confidence interval of two groups with stratification by the 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 stratum. 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 stratum. 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 the common odds ratio and its score confidence interval of two groups with stratification. The confidence interval is asymmetric, and there is no standard error in the output. For the stratified case, the inverse variance weighted score statistic with the bias correction is used, following Laud. The common odds ratio point estimate is the zero of the weighted score statistic, and the confidence bounds are found with the uniroot function. The result agrees with ratesci::scoreci(contrast="OR", stratified=TRUE, skew=FALSE) to at least 7 significant digits. For a single stratum, it returns the classical Miettinen-Nurminen interval of ORmn1. This can be used for meta-analysis also.

Value

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

odd1	odds from the first group, y1/(n1 - y1). For the common value, it is calculated with the weights at the point estimate.
odd2	odds from the second group, y2/(n2 - y2). For the common value, it is calculated with the weights at the point estimate.
OR	odds ratio of the stratum. The common OR is the zero of the inverse variance weighted score statistic.
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

Laud PJ. Equal-tailed confidence intervals for comparison of rates. Pharmaceutical Statistics 2017;16:334-348

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)
d2 = data.frame(y1=c(4, 2, 10), n1=c(20, 20, 20), y2=c(8, 11, 2), n2=c(20, 20, 20))
ORmn(d2)
```

ORmn1 *Odds Ratio and Score CI of two groups without strata by the 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 the test (the first) group
n1	total count of the test (the first) group
y2	positive event count of the control (the second) group
n2	total count of the control (the second) group
conf.level	confidence level
eps	absolute value less than eps is regarded as negligible

Details

It calculates the odds ratio and its score confidence interval of the two groups. The confidence interval is asymmetric, and there is no standard error in the output. This does not support stratification. This implementation uses the `uniroot` function, which usually gives at least 5 significant digits. In contrast, the `PropCIs::orscoreci` function uses an incremental or decremental search by a factor of 1.001, which gives less than 3 significant digits.

Value

There is no standard error.

`odd1`	odds from the first group, y1/(n1 - y1)
`odd2`	odds 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

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 the 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 the legend
ly	y position of the legend
...	arguments to be passed to plot

Details

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

Value

Ignore the 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 the Pearson method.

Usage

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

Arguments

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

Details

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

Value

Row names show which columns are used for the test.

`Estimate`	point estimate of correlation
`Df`	degrees 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

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

Value

Four diagnostic plots on 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 differences 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 the confidence interval
adj	"lsd", "tukey", "scheffe", "bon", "duncan", or "dunnett" to adjust the p-value and confidence limit
ref	reference or control level for the Dunnett test. If missing, the first level is used, as SAS does.
PLOT	whether to plot the diffogram
reverse	reverse A - B to B - A
...	arguments to be passed to plot

Details

It corresponds to the PDIFF option of SAS PROC GLM.

Value

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

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 the t distribution
Df	degrees of freedom
Pr(>|t|)	probability of a larger absolute t value from the t distribution with the residual's degrees 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")
```

PocockBound — *Pocock (fixed) Bound for the cumulative Z-test with a final target alpha-value*

Description

Cumulative alpha values for the cumulative hypothesis test with a fixed upper bound z-value in group sequential design.

Usage

```
PocockBound(K=2, alpha=0.05, side=2)
```

Arguments

K	total number of tests
alpha	alpha value at the final test
side	1=one-sided test, 2=two-sided test

Details

Pocock suggested a fixed upper bound z-value for the cumulative hypothesis test in group sequential designs.

Value

a fixed upper bound z-value for the K times repeated hypothesis test with a final alpha-value. Attributes are:

ti	time of test. Equal intervals are assumed.
cum.alpha	cumulative alpha value

Author(s)

Kyun-Seop Bae k@acr.kr

References

Reboussin DM, DeMets DL, Kim K, Lan KKG. Computations for group sequential boundaries using the Lan-DeMets function method. Controlled Clinical Trials. 2000;21:190-207.

See Also

seqBound, OBFBound

Examples

```
PocockBound(K=2) # Z-value of upper bound for the two-stage design
PocockBound(K=3) # Z-value of upper bound for the two-stage design
PocockBound(K=4) # Z-value of upper bound for the two-stage design
PocockBound(K=5) # Z-value of upper bound for the two-stage design
```

---

pResD *Residual Diagnostic Plot for Regression*

Description

Nine residual diagnostic plots.

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 on a page.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

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

QuartileRange *Interquartile Range*

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 the IQR function

Details

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

Value

The value of the interquartile range

Author(s)

Kyun-Seop Bae k@acr.kr

Range *Range*

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 the range

Author(s)

Kyun-Seop Bae k@acr.kr

`RanTest` *Test with Random Effects*

Description

Hypothesis test with a specified type of SS using random effects as error terms. This corresponds to SAS PROC GLM's RANDOM /TEST statement.

Usage

```
RanTest(Formula, Data, Random="", Type=3, eps=1e-8)
```

Arguments

`Formula`	a conventional formula for a linear model
`Data`	a `data.frame` to be analyzed
`Random`	a vector of random effects. All should be specified as primary terms, not as interaction terms. All interaction terms with a random factor are regarded as random effects.
`Type`	Sum of squares type to be used as contrast
`eps`	A value less than this is considered as zero.

Details

Type can be from 1 to 3. All interaction terms with a random factor are regarded as random effects. Here the error term should not be the MSE.

Value

Returns ANOVA and E(MS) tables with the specified type of SS.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
RanTest(log(CMAX) ~ SEQ/SUBJ + PRD + TRT, BEdata, Random="SUBJ")
fBE = log(CMAX) ~ ADM/SEQ/SUBJ + PRD + TRT
RanTest(fBE, BEdata, Random=c("ADM", "SUBJ"))
RanTest(fBE, BEdata, Random=c("ADM", "SUBJ"), Type=2)
RanTest(fBE, BEdata, Random=c("ADM", "SUBJ"), Type=1)
```

RD *Risk Difference between two groups*

Description

Risk (proportion) difference between two groups

Usage

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

Arguments

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

Details

It calculates the risk difference between the two groups. No continuity correction here. If you need the 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 confidence interval between two groups with stratification by the inverse variance method

Usage

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

Arguments

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

Details

It calculates the risk difference and its confidence interval between two groups by the inverse variance method. The common risk difference is given by both the fixed effect model and the DerSimonian-Laird random effects model, with Cochran's Q test for heterogeneity. If you need the percent scale, multiply the output by 100. This can be used for meta-analysis also.

Value

RDs	risk difference and its confidence interval of each stratum
Heterogeneity	Cochran's Q statistic for heterogeneity across the strata and its p-value
tau2	between-strata variance estimated by the method of moments
Fixed	common risk difference, its standard error, and its confidence interval by the fixed effect model with the inverse variance weights
Random	common risk difference, its standard error, and its confidence interval by the DerSimonian-Laird random effects model

Author(s)

Kyun-Seop Bae k@acr.kr

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 *Risk Difference and Score CI between two groups with strata by the MN method*

Description

Risk difference and its score confidence interval between two groups with stratification by the 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 represents a stratum. This should have four columns named y1, n1, y2, and n2; y1 and y2 for events of each group, n1 and n2 for the sample size of each stratum. The second group is usually the control group. The maximum allowable value for n1 and n2 is 1e8.
conf.level	confidence level
eps	an absolute value less than eps is regarded as negligible

Details

It calculates the risk difference and its score confidence interval between the two groups. The confidence interval is asymmetric, and there is no standard error in the output. If you need the percent scale, multiply the output by 100. This supports stratification. This implementation uses the uniroot function, which usually gives at least 5 significant digits. This can be used for meta-analysis also.

Value

The following output will be returned for each stratum and the 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. The point estimate of the common RD is calculated with the 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)
d2 = data.frame(y1=c(4, 2, 10), n1=c(20, 20, 20), y2=c(8, 11, 2), n2=c(20, 20, 20))
RDmn(d2)
```

RDmn1 *Risk Difference and Score CI between two groups without strata by the MN method*

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 the test (the first) group
n1	total count of the test (the first) group. The maximum allowable value is 1e8.
y2	positive event count of the control (the second) group
n2	total count of the control (the second) group. The maximum allowable value is 1e8.
conf.level	confidence level
eps	an absolute value less than eps is regarded as negligible

Details

It calculates the risk difference and its score confidence interval between the two groups. The confidence interval is asymmetric, and there is no standard error in the output. If you need the percent scale, multiply the output by 100. This does not support stratification. This implementation uses the uniroot function, which usually gives at least 5 significant digits.

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

Description

REG is similar to SAS PROC REG.

Usage

```
REG(Formula, Data, conf.level=0.95, HC=FALSE, Resid=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, and White's first and second moment specification test
Resid	if TRUE, fitted values (y hat) and residuals will be returned

Weights	weights for each observation, usually the inverse of each variance. This should be a scalar or a vector of the same length as the number of rows of Data. Observations with nonpositive weights are excluded from the analysis, as SAS does.
summarize	If this is FALSE, REG returns just the 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 the ANOVA table.

The second part is measures of 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	upper confidence limit with conf.level
Df	degrees of freedom
t value	value for the t distribution
Pr(>|t|)	probability of a larger absolute t value from the t distribution with the residual's degrees of freedom

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

Fitted	Fitted value or y hat in the original scale as SAS OUTPUT P= does, even with Weights. This is returned only with the Resid=TRUE option.
Residual	Residuals in the original scale as SAS OUTPUT R= does, even with Weights. This is returned only with the Resid=TRUE option.

If summarize=FALSE, REG returns;

coefficients	beta coefficients
g2	g2 inverse
rank	rank of the model matrix
DFr	degrees of freedom for the residual
SSE	sum of squared errors

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

lr

Examples

```
REG(uptake ~ Plant + Type + Treatment + conc, CO2)
REG(uptake ~ conc, CO2, HC=TRUE)
REG(uptake ~ conc, CO2, Resid=TRUE)
REG(uptake ~ conc, CO2, HC=TRUE, Resid=TRUE)
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 the g2 inverse; therefore, it cannot handle a singular matrix. If the model (design) matrix is not of full rank, use REG or fewer parameters.

Value

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

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

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

RR *Relative Risk of the two groups*

Description

Relative Risk between the two groups

Usage

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

Arguments

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

Details

It calculates the relative risk of the two groups. No continuity correction here. If you need the 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 *Relative Risk of two groups with strata by inverse variance method*

Description

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

Usage

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

Arguments

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

Details

It calculates the relative risk and its confidence interval of two groups by the inverse variance method on the log scale. The common relative risk is given by both the fixed effect model and the DerSimonian-Laird random effects model, with Cochran's Q test for heterogeneity. This can be used for meta-analysis also.

Value

RRs	relative risk, its confidence interval, and the percent weights (pwi for the fixed effect model, pwsi for the random effects model) of each stratum
Heterogeneity	Cochran's Q statistic for heterogeneity across the strata and its p-value
tau2	between-strata variance estimated by the method of moments
Fixed	common relative risk and its confidence interval by the fixed effect model with the inverse variance weights
Random	common relative risk and its confidence interval by the DerSimonian-Laird random effects model

Author(s)

Kyun-Seop Bae k@acr.kr

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 *Relative Risk and Score CI of two groups with strata by the MN method*

Description

Relative risk and its score confidence interval of two groups with stratification by the 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 represents a stratum. This should have four columns named y1, n1, y2, and n2; y1 and y2 for events of each group, n1 and n2 for the sample size of each stratum. The second group is usually the control group.
conf.level	confidence level
eps	an absolute value less than eps is regarded as negligible

Details

It calculates the relative risk and its score confidence interval of the two groups. The confidence interval is asymmetric, and there is no standard error in the output. This supports stratification. This implementation uses the uniroot function, which usually gives at least 5 significant digits, whereas the PropCIs::riskscoreci function uses a cubic equation approximation which gives only about 2 significant digits. This can be used for meta-analysis also.

Value

The following output will be returned for each stratum and the 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. The point estimate of the common RR is calculated with the 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)
d2 = data.frame(y1=c(4, 2, 10), n1=c(20, 20, 20), y2=c(8, 11, 2), n2=c(20, 20, 20))
RRmn(d2)
```

RRmn1 *Relative Risk and Score CI of two groups without strata by the MN method*

Description

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

Usage

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

Arguments

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

Details

It calculates the relative risk and its score confidence interval of the two groups. The confidence interval is asymmetric, and there is no standard error in the output. This does not support stratification. This implementation uses the `uniroot` function, which usually gives at least 5 significant digits, whereas the `PropCIs::riskscoreci` function uses a cubic equation approximation which gives only about 2 significant digits.

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
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, ORmn1, RDmn, RRmn, ORmn

Examples

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

satt *Satterthwaite Approximation of Variance and Degrees of Freedom*

Description

Calculates the pooled variance and degrees of freedom using the Satterthwaite equation.

Usage

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

Arguments

vars	a vector of variances
dfs	a vector of degrees of freedom
ws	a vector of weights

Details

The input can contain more than two variances.

Value

Variance	approximated variance
Df	degrees of freedom

Author(s)

Kyun-Seop Bae k@acr.kr

ScoreCI *Score Confidence Interval for a Proportion or a Binomial Distribution*

Description

Score confidence interval 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 the 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 the percent scale, multiply the output by 100.

Value

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

`PE`	point estimate of the proportion
`Lower`	lower confidence limit of the proportion
`Upper`	upper confidence limit of the proportion

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`binom.test`, `prop.test`

Examples

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

SD *Standard Deviation*

Description

Standard deviation of a 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 *Standard Error of the Sample Mean*

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

---

seqBound *Sequential bounds for cumulative Z-test in Group Sequential Design*

Description

Sequential upper bounds for cumulative Z-test on accumulating data. Z values are correlated. This is usually used for a group sequential design.

Usage

```
seqBound(ti, alpha = 0.05, side = 2, t2 = NULL, asf = 1)
```

Arguments

ti	times for the tests. These should be in [0, 1].
alpha	goal alpha value for the last test at time 1.
side	1=one-sided test, 2=two-sided test
t2	fractions of the information amount. These should be in [0, 1]. If not available, ti will be used instead.
asf	alpha spending function. 1=O'Brien-Fleming type (approximate, not exact), 2=Pocock type (approximate, not exact), 3=alpha*ti, 4=alpha*ti^1.5, 5=alpha*ti^2

Details

It calculates upper z-bounds and cumulative alpha-values for the repeated tests in a group sequential design. The correlation is assumed to be sqrt(t_i/t_j). Use PocockBound and OBFBound for more exact bounds.

Value

The result is a matrix.

time	time of test
up.bound	upper z-bound
cum.alpha	cumulative alpha-value

Author(s)

Kyun-Seop Bae k@acr.kr

References

Reboussin DM, DeMets DL, Kim K, Lan KKG. Computations for group sequential boundaries using the Lan-DeMets function method. Controlled Clinical Trials. 2000;21:190-207.

See Also

PocockBound, OBFBound

Examples

```
seqBound(ti=(1:5)/5)
seqBound(ti=(1:5)/5, asf=2)
```

seqCI *Confidence interval with the last Z-value for a group sequential design*

Description

Confidence interval with given upper bounds, times of tests, the last Z-value, and confidence level.

Usage

```
seqCI(bi, ti, Zval, conf.level=0.95)
```

Arguments

bi	upper bound z-values
ti	times for the tests. These should be in [0, 1].
Zval	the last z-value from the observed data. This is not necessarily the planned final Z-value.
conf.level	confidence level

Details

It calculates the confidence interval with given upper bounds, times of tests, the last Z-value, and confidence level. It assumes a two-sided test. `mvtnorm::pmvt` (with noncentrality) is better than `pmvnorm` in calculating power, sample size, and confidence interval. However, Lan-DeMets used the multivariate normal rather than the multivariate noncentral t distribution. This function follows Lan-DeMets for consistency with previous results. For the theoretical background, see the reference.

Value

confidence interval of the Z-value for the given confidence level.

Author(s)

Kyun-Seop Bae k@acr.kr

References

Reboussin DM, DeMets DL, Kim K, Lan KKG. Computations for group sequential boundaries using the Lan-DeMets function method. Controlled Clinical Trials. 2000;21:190-207.

Examples

```
seqCI(bi = c(2.53, 2.61, 2.57, 2.47, 2.43, 2.38),
      ti = c(.2292, .3333, .4375, .5833, .7083, .8333), Zval=2.82)
```

`Skewness` *Skewness*

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

See Also

`SkewnessSE`

`SkewnessSE` *Standard Error of Skewness*

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 *F Test with Slice*

Description

Performs an F test with a given slice term.

Usage

```
SLICE(Formula, Data, Term, By)
```

Arguments

Formula	a conventional formula for a linear model
Data	a data.frame to be analyzed
Term	a factor name (not an interaction) to calculate the sum of squares and do an F test with least square means
By	a factor name to be used for slicing

Details

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

Value

Returns the sum of squares and its F value and p-value. Row names are the levels of the slice term.

Df	degrees of freedom
Sum Sq	sum of squares for the set of contrasts
Mean Sq	mean square
F value	F value for the F distribution
Pr(>F)	probability of a larger 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	Values less than this are considered as zero.

Details

It calculates the 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 the 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 missing, e3 is used
eps	Values less than this are considered as zero.

Details

This is necessary for further hypothesis tests 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 an 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, H="", E="", eps=1e-8)
```

Arguments

Formula	a conventional formula for a linear model
Data	a data.frame to be analyzed
H	Hypothesis term
E	Error term
eps	Values less than this are 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, E=c("SEQ:SUBJ"))
T3test(log(CMAX) ~ SEQ/SUBJ + PRD + TRT, BEdata, H="SEQ", E=c("SEQ:SUBJ"))
```

---

tmtest *Independent two means test similar to t.test with summarized input*

Description

This produces essentially the same result as t.test, except using summarized input (sufficient statistics).

Usage

```
tmtest(m1, s1, n1, m0, s0, n0, conf.level=0.95, nullHypo=0, var.equal=FALSE)
```

Arguments

m1	mean of the first (test, active, experimental) group
s1	sample standard deviation of the first group
n1	sample size of the first group
m0	mean of the second (reference, control, placebo) group
s0	sample standard deviation of the second group
n0	sample size of the second group

conf.level	confidence level
nullHypo	value for the difference of means under the null hypothesis
var.equal	assumption on the variance equality

Details

The default is the Welch t-test with the Satterthwaite approximation.

Value

The output format is very similar to that of t.test.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

mtest, TTEST, ztest

Examples

```
tmtest(5.4, 10.5, 3529, 5.1, 8.9, 5190) # NEJM 388;15 p1386
tmtest(5.4, 10.5, 3529, 5.1, 8.9, 5190, var.equal=TRUE)
```

---

trimmedMean *Trimmed Mean*

Description

Trimmed mean wrapping the 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 the 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`	An independent variable with more levels 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 for `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 with 0 independent (x) variables*

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 column names
`y`	y variable name, a continuous variable
`e`	a vector of summary function names
`repl`	a list of strings to replace after summarization. The length of the list should be 2, and both elements 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 with 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 column names
y	y variable name, a continuous variable
u	x variable name, upper side variable
e	a vector of summary function names
ou	order of levels of the upper side x variable
repl	a list of strings to replace after summarization. The length of the list should be 2, and both elements 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 the 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 with 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 column names
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 summary function names
h	a vector of summary function names for the horizontal subgroup. If NULL, it becomes the same as the e argument.
ol	order of levels of the left side x variable
ou	order of levels of the upper side x variable
rm.dup	if TRUE, duplicated names of levels are specified on the first occurrence only.
repl	a list of strings to replace after summarization. The length of the list should be 2, and both elements 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 with 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 column names
y	y variable name, a continuous variable
l	a vector of two x variable names 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 summary function names
h	a list of two vectors of summary function names for the first and second horizontal subgroups. If NULL, it becomes the same as the e argument.
ol1	order of levels of the 1st left side x variable
ol2	order of levels of the 2nd left side x variable
ou	order of levels of the upper side x variable
rm.dup	if TRUE, duplicated names of levels are specified on the first occurrence only.
repl	a list of strings to replace after summarization. The length of the list should be 2, and both elements 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")))
```

TTEST *Independent two groups t-test comparable to PROC TTEST*

Description

This is comparable to SAS PROC TTEST.

Usage

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

Arguments

x	a vector of data from the first (test, active, experimental) group
y	a vector of data from the second (reference, control, placebo) group
conf.level	confidence level

Details

Be cautious when choosing the row to use in the output.

Value

The output format is comparable to that of SAS PROC TTEST.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

mtest, tmtest, ztest

Examples

```
TTEST(mtcars[mtcars$am==1, "mpg"], mtcars[mtcars$am==0, "mpg"])
```

UCL *Upper Confidence Limit*

Description

The estimate of the upper bound of the confidence limit using the 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 the confidence limit using the 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 function for descriptive statistics. NA is removed during the calculation. This is similar to SAS PROC UNIVARIATE.

Value

nAll	count of all elements in the input vector
nNA	count of NA elements
nFinite	count of finite numbers
Mean	mean excluding NA
Variance	variance excluding NA
SD	standard deviation excluding NA
CV	coefficient of variation in percent
SEM	standard error of the sample mean, the sample standard deviation divided by the square root of nFinite
LowerCL	lower confidence limit of mean
UpperCL	upper confidence limit of mean
TrimmedMean	trimmed mean with a trimming proportion of 1 - confidence level
Min	minimum value
Q1	first quartile value with quantile type 2, the SAS default
Median	median value
Q3	third quartile value with quantile type 2, the SAS default
Max	maximum value
Range	range of finite numbers, maximum - minimum
IQR	interquartile range with quantile type 2, the SAS default
MAD	median absolute deviation
VarLL	lower confidence limit of variance
VarUL	upper confidence limit of variance
Skewness	skewness
SkewnessSE	standard error of skewness
Kurtosis	kurtosis
KurtosisSE	standard error of 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)
```

vtest *F-Test for the ratio of two groups' variances*

Description

F-test for the ratio of two groups' variances. This is similar to `var.test` except using the summarized input.

Usage

```
vtest(v1, n1, v0, n0, ratio=1, conf.level=0.95)
```

Arguments

`v1`	sample variance of the first (test, active, experimental) group
`n1`	sample size of the first group
`v0`	sample variance of the second (reference, control, placebo) group
`n0`	sample size of the second group
`ratio`	value for the ratio of variances under the null hypothesis
`conf.level`	confidence level

Details

For the confidence interval of one group, use the `UNIV` function.

Value

The output format is very similar to that of `var.test`.

Author(s)

Kyun-Seop Bae k@acr.kr

Examples

```
vtest(10.5^2, 3529, 8.9^2, 5190) # NEJM 388;15 p1386
vtest(2.3^2, 13, 1.5^2, 11, conf.level=0.9) # Red book p240
```

`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 White's general test for heteroskedasticity.

Value

Returns a direct test result by the more complex Theorem 2, not by the 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))
```

`ztest` *Test for the difference of two groups' means*

Description

This is similar to the two groups t-test, but using the standard normal (Z) distribution.

Usage

```
ztest(m1, s1, n1, m0, s0, n0, conf.level=0.95, nullHypo=0)
```

Arguments

`m1`	mean of the first (test, active, experimental) group
`s1`	known standard deviation of the first group
`n1`	sample size of the first group
`m0`	mean of the second (reference, control, placebo) group
`s0`	known standard deviation of the second group
`n0`	sample size of the second group
`conf.level`	confidence level
`nullHypo`	value for the difference of means under the null hypothesis

Details

Use this only for known standard deviations (or variances) or very large sample sizes per group.

Value

The output format is very similar to that of `t.test`.

Author(s)

Kyun-Seop Bae k@acr.kr

See Also

`mtest`, `tmtest`, `TTEST`

Examples

```
ztest(5.4, 10.5, 3529, 5.1, 8.9, 5190) # NEJM 388;15 p1386
```

Index