

Package ‘PF’

April 17, 2024

Type Package

Title Prevented fraction

Version 9.6.6

Date 2024-04-17

Description Functions related to PF (prevented fraction). Calculate incidence density ratio, confidence interval, and Rao-Scott weights of PF by multiple methods. See <http://goo.gl/eJ6Rxi> for definition of PF.
No endorsement, claim, or warranty is implied for this package. It is made available for investigational or pedagogical use only.

License file LICENSE

URL https://www.aphis.usda.gov/animal_health/vet_biologics/publications/STATWI0007.pdf, <https://github.com/ABS-dev/PF/blob/master/doc/PFPackageManual.pdf>

BugReports <https://github.com/ABS-dev/PF/issues>

LazyLoad true

LazyData true

Depends R (>= 4.0)

Imports methods,
plyr,
dplyr,

Suggests tidyr,
magrittr,
testthat,
knitr,
R.rsp,
rmarkdown

Collate 'aaa.r'
'aab.r'
'aac.r'
'class.r'
'generics.r'
'PF.r'
'PF-package.r'
'IDRlsi.r'
'IDRsc.r'

'phiWt.r'
 'RRlsi.r'
 'RRmh.r'
 'RRmpWald.r'
 'RRor.r'
 'RRotsst.r'
 'RRsc.r'
 'RRstr.r'
 'RRtosst.r'
 'rsbWt.r'
 'tauWt.r'

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Config/build/clean-inst-doc FALSE

VignetteBuilder R.rsp

R topics documented:

.rr.score.asymp	3
bird	3
birdm	4
IDRlsi	4
IDRsc	7
New	9
pf-class	10
phiWt	10
print.rr1	11
rat	12
rr1-class	12
RRlsi	13
RRmh	15
rrmp-class	17
RRmpWald	18
RRor	20
rror-class	22
RRotsst	23
RRsc	25
rrsc-class	27
rrsi-class	28
RRstr	29
rrstr-class	31
RRtosst	32
rsb	34
rsbWt	35
Set1	36
set1	37
Table6	37
table6	37
tauWt	38

<code>.rr.score.asymp</code>	3
------------------------------	---

Index	40
--------------	-----------

<code>.rr.score.asymp</code>	<i>Internal function.</i>
------------------------------	---------------------------

Description

Internal function.

Usage

```
.rr.score.asymp(y, alpha = 0.05, iter.max = 18, converge = 1e-04, mn = FALSE)
```

Arguments

<code>y</code>	data
<code>alpha</code>	alpha
<code>iter.max</code>	maximum number of iterations
<code>converge</code>	convergence criterion
<code>mn</code>	boolean whether to calculate MN or use default value of 1.0

Examples

```
# none
```

<code>bird</code>	<i>bird dataset</i>
-------------------	---------------------

Description

bird dataset

Format

- a data.frame with 6 observations of the following 4 variables, no NAs
- y** number positive
- n** total number in group tx x all
- tx** treatment 'vac' or 'con'
- all** all?

References

we need some references

birdm	<i>birdm dataset</i>
-------	----------------------

Description

birdm dataset

Format

a data.frame with 6 observations of the following 4 variables, no NAs

y number positive

n total number in group tx x all

tx treatment 'vac' or 'con'

all all?

References

we need some references

IDRlsi	<i>IDR likelihood support interval.</i>
--------	---

Description

Estimates likelihood support interval for the incidence density ratio or prevented fraction based on it.

Usage

```
IDRlsi(
  y = NULL,
  formula = NULL,
  data = NULL,
  alpha = 0.05,
  k = 8,
  use.alpha = FALSE,
  pf = TRUE,
  converge = 1e-08,
  rnd = 3,
  start = NULL,
  trace.it = FALSE,
  iter.max = 24,
  compare = c("con", "vac")
)
```

Arguments

y	Data vector $c(y1, n1, y2, n2)$ where y are the positives, n are the total, and group 1 is compared to group 2 (control or reference).
formula	Formula of the form $cbind(y, n) \sim x$, where y is the number positive, n is the group size, x is a factor with two levels of treatment.#'
data	data.frame containing variables of the formula.
alpha	Complement of the confidence level.
k	Likelihood ratio criterion.
use.alpha	Base choice of k on its relationship to alpha?
pf	Estimate <i>IDR</i> or its complement <i>PF</i> ?
converge	Convergence criterion
rnd	Number of digits for rounding. Affects display only, not estimates.
start	describe here.
trace.it	Verbose tracking of the iterations?
iter.max	Maximum number of iterations
compare	Text vector stating the factor levels: compare[1] is the vaccinate group to which compare[2] (control or reference) is compared.

Details

Estimates likelihood support interval for the incidence density ratio based on orthogonal factoring of reparameterized likelihood. The incidence density is the number of cases per subject-time; its distribution is assumed Poisson.

Likelihood support intervals are usually formed based on the desired likelihood ratio, often 1/8 or 1/32. Under some conditions the log likelihood ratio may follow the chi square distribution. If so, then $\alpha = 1 - F(2\log(k), 1)$, where F is a chi-square CDF. `RRsc()` will make the conversion from α to k if `use.alpha = TRUE`.

The data may also be a matrix. In that case y would be entered as `matrix(c(y1, n1 - y1, y2, n2 - y2), 2, 2, byrow = TRUE)`.

Value

A `rrsi` object with the following elements.

estimate	vector with point and interval estimate
estimator	either <i>PF</i> or <i>IDR</i>
y	data.frame with "y1", "n1", "y2", "n2" values.
k	Likelihood ratio criterion
rnd	how many digits to round the display
alpha	complement of confidence level

Author(s)

[PF-package](#)

References

Royall R. *Statistical Evidence: A Likelihood Paradigm*. Chapman & Hall, Boca Raton, 1997. Section 7.2.

See Also

[IDRsc](#)

Examples

```
# Both examples represent the same observation, with data entry by vector
# and matrix notation.
```

```
y_vector <- c(26, 204, 10, 205)
IDRlsi(y_vector, pf = FALSE)
```

```
# 1/8 likelihood support interval for IDR
```

```
# corresponds to 95.858% confidence
# (under certain assumptions)
```

```
# IDR
# IDR  LL  UL
# 2.61 1.26 5.88
```

```
y_matrix <- matrix(c(26, 178, 10, 195), 2, 2, byrow = TRUE)
y_matrix
#      [, 1] [, 2]
# [1, ]  26  178
# [2, ]  10  195
```

```
IDRlsi(y_matrix, pf = FALSE)
```

```
# 1/8 likelihood support interval for IDR
```

```
# corresponds to 95.858% confidence
# (under certain assumptions)
```

```
# IDR
# IDR  LL  UL
# 2.61 1.26 5.88
```

```
data1 <- data.frame(group = rep(c("treated", "control"), each = 5),
                    n = c(rep(41, 4), 40, rep(41, 5)),
                    y = c(4, 5, 7, 6, 4, 1, 3, 3, 2, 1),
                    cage = rep(paste('cage', 1:5), 2))
IDRlsi(data = data1, formula = cbind(y, n) ~ group,
       compare = c("treated", "control"), pf = FALSE)
```

```
# 1/8 likelihood support interval for IDR
```

```
# corresponds to 95.858% confidence
# (under certain assumptions)
```

```
# IDR
# IDR  LL  UL
```

```
# 2.61 1.26 5.88

require(dplyr)
data2 <- data1 %>%
  group_by(group) %>%
  summarize(sum_y = sum(y),
            sum_n = sum(n))

IDRlsi(data = data2, formula = cbind(sum_y, sum_n) ~ group,
        compare = c("treated", "control"), pf = FALSE)

# 1/8 likelihood support interval for IDR

# corresponds to 95.858% confidence
# (under certain assumptions)

# IDR
# IDR  LL  UL
# 2.61 1.26 5.88
```

IDRsc

IDR confidence interval.

Description

Estimates confidence interval for the incidence density ratio or prevented fraction based on it.

Usage

```
IDRsc(
  y = NULL,
  data = NULL,
  formula = NULL,
  compare = c("con", "vac"),
  alpha = 0.05,
  pf = TRUE,
  rnd = 3
)
```

Arguments

<code>y</code>	Data vector <code>c(y1, n1, y2, n2)</code> where <code>y</code> are the positives, <code>n</code> are the total, and group 1 is compared to group 2 (control or reference).
<code>data</code>	<code>data.frame</code> containing variables of formula.
<code>formula</code>	Formula of the form <code>cbind(y, n) ~ x</code> , where <code>y</code> is the number positive, <code>n</code> is the group size, <code>x</code> is a factor with two levels of treatment.
<code>compare</code>	Text vector stating the factor levels: <code>compare[1]</code> is the vaccinate group to which <code>compare[2]</code> (control or reference) is compared.
<code>alpha</code>	Complement of the confidence level.
<code>pf</code>	Estimate <i>IDR</i> , or its complement <i>PF</i> ?
<code>rnd</code>	Number of digits for rounding. Affects display only, not estimates.

Details

The incidence density is the number of cases per subject-time; its distribution is assumed Poisson. IDRsc estimates a confidence interval for the incidence density ratio using Siev's formula based on the Poisson score statistic. $IDR = \widehat{IDR} \left\{ 1 + \left(\frac{1}{y_1} + \frac{1}{y_2} \right) \frac{z_{\alpha/2}^2}{2} \pm \frac{z_{\alpha/2}^2}{2y_1 y_2} \sqrt{y_{\bullet} \left(y_{\bullet} z_{\alpha/2}^2 + 4y_1 y_2 \right)} \right\}$

The data may also be a matrix. In that case y would be entered as `matrix(c(y1, n1 - y1, y2, n2 - y2), 2, 2, byrow = TRUE)`.

Value

A [rr1](#) object with the following elements.

estimate	vector with point and interval estimate
estimator	either <i>PF</i> or <i>IDR</i>
y	data vector
rnd	how many digits to round the display
alpha	complement of confidence level

Author(s)

[PF-package](#)

References

Siev D, 1994. Estimating vaccine efficacy in prospective studies. *Preventive Veterinary Medicine* 20:279-296, Appendix 1.
 Graham PL, Mengersen K, Morton AP, 2003. Confidence limits for the ratio of two rates based on likelihood scores:non-iterative method *Statistics in Medicine* 22:2071-2083.
 Siev D, 2004. Letter to the editor. *Statistics in Medicine* 23:693. (Typographical error in formula: replace the two final minus signs with subscript dots.)

See Also

[IDR1si](#)

Examples

```
# All examples represent the same observation, with data entry by vector,
# matrix, and formula+data notation.

y_vector <- c(26, 204, 10, 205)
IDRsc(y_vector, pf = FALSE)

# IDR
# 95% interval estimates

# IDR   LL   UL
# 2.61  1.28  5.34

y_matrix <- matrix(c(26, 178, 10, 195), 2, 2, byrow = TRUE)
y_matrix
#      [, 1] [, 2]
# [1, ]   26  178
```



```

# [2, ]    10  195

IDRsc(y_matrix, pf = FALSE)

# IDR
# 95% interval estimates

# IDR  LL  UL
# 2.61 1.28 5.34

require(dplyr)
data1 <- data.frame(group = rep(c("treated", "control"), each = 5),
  n = c(rep(41, 4), 40, rep(41, 5)),
  y = c(4, 5, 7, 6, 4, 1, 3, 3, 2, 1),
  cage = rep(paste('cage', 1:5), 2))
data2 <- data1 %>%
  group_by(group) %>%
  summarize(sum_y = sum(y),
    sum_n = sum(n))
IDRsc(data = data2, formula = cbind(sum_y, sum_n) ~ group,
  compare = c("treated", "control"), pf = FALSE)

# IDR
# 95% interval estimates

# IDR  LL  UL
# 2.61 1.28 5.34

```

New

New dataset

Description

New dataset

Format

a data frame with 52 observations of the following 3 variables, no NAs

cage cage ID. 1 - 26

tx treatment. one of 'con' or 'vac'

pos numeric indicator of positive response. 0 = FALSE or 1 = TRUE

References

We need some references

pf-class

Data class pf

Description

data class pf

Fields

- [estimator] either "PF" or "IDR"
- [rnd] how many digits to round display
- [alpha] complement of c.i.

Author(s)

[PF-package](#)

See Also

[rrl](#), [rrsi](#), [rrsc](#), [rrstr](#)

phiWt

Binomial dispersion parameter.

Description

MME estimate of dispersion parameter ϕ .

Usage

```
phiWt(fit, subset.factor = NULL, fit.only = TRUE, show.warns = FALSE)
```

Arguments

fit	A glm object.
subset.factor	Factor for estimating phi by subset.
fit.only	Return only the new fit? If FALSE, also returns the weights and phi estimates.
show.warns	Show warnings

Details

Estimates binomial dispersion parameter ϕ by the method of moments. Refits the model, weighting the observations by $1/\phi$. Uses quasibinomial family in `glm()`.

Value

A list with the following elements.

fit	the new model fit, updated by the estimated weights
weights	vector of weights
phi	vector of phi estimates

Author(s)[PF-package](#)**References**

Wedderburn RWM, 1974. Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika* 61:439-447.

See Also[tauWt](#), [RRor](#).**Examples**

```
birdm.fit <- glm(cbind(y, n - y)~tx-1, binomial, birdm)
RRor(phiWt(birdm.fit))
#
# 95% t intervals on 4 df
#
# PF
#      PF      LL      UL
# 0.479 -0.537 0.823
#
#      mu.hat  LL      UL
# txcon 0.768 0.95 0.367
# txvac 0.400 0.78 0.111
#
```

print.rr1

*Print values for PF data obhects.***Description**

Print values for PF data obhects.

Usage

```
## S3 method for class 'rr1'
print(x, ...)

## S3 method for class 'rror'
print(x, ...)

## S3 method for class 'rrsi'
print(x, ...)

## S3 method for class 'rrmp'
print(x, ...)

## S3 method for class 'rrsc'
print(x, ...)

## S3 method for class 'rrstr'
print(x, ...)
```

Arguments

x object of class rr1, rror, rrsi, rrmf, rrstr, rrsc
... other arguments not used by this method

rat	<i>rat dataset</i>
-----	--------------------

Description

rat dataset

Format

a data.frame with 32 observations of the following 3 variables, no NAs

y number positive
n total number
group treatment group: 'control' or 'treated'

References

Weil's rat data (Table 1 of Rao and Scott)

rr1-class	<i>Data class rr1</i>
-----------	-----------------------

Description

Data class rr1

Fields

- [estimate] vector with point and interval estimate
- [estimator] either "PF" or "IDR"
- [y]data.frame with restructured input
- [rnd]how many digits to round display
- [alpha]complement of c.i.

Author(s)

[PF-package](#)

See Also

[IDRsc](#), [RRotsst](#), [RRtosst](#)

RRlsi

*RR likelihood support interval.***Description**

likelihood support interval for the risk ratio or prevented fraction by the likelihood profile.

Usage

```
RRlsi(
  y = NULL,
  formula = NULL,
  data = NULL,
  compare = c("vac", "con"),
  alpha = 0.05,
  k = 8,
  use.alpha = FALSE,
  pf = TRUE,
  iter.max = 50,
  converge = 1e-06,
  rnd = 3,
  start = NULL,
  track = FALSE,
  full.track = FALSE
)
```

Arguments

y	Data vector c(y1, n1, y2, n2) where y are the positives, n are the total, and group 1 is compared to group 2 (control or reference group).
formula	Formula of the form cbind(y, n) ~ x, where y is the number positive, n is the group size, x is a factor with two levels of treatment.
data	data.frame containing variables of formula.
compare	Text vector stating the factor levels: compare[1] is the vaccinate group to which compare[2] (control or reference) is compared.
alpha	Complement of the confidence level (see details).
k	Likelihood ratio criterion.
use.alpha	Base choice of k on its relationship to alpha?
pf	Estimate <i>RR</i> or its complement <i>PF</i> ?
iter.max	Maximum number of iterations
converge	Convergence criterion
rnd	Number of digits for rounding. Affects display only <i>RR</i> , not estimates.
start	Optional starting value.
track	Verbose tracking of the iterations?
full.track	Verbose tracking of the iterations?

Details

Estimates a likelihood support interval for *RR* or *PF* by the profile likelihood method using the DUD algorithm.

Likelihood support intervals are usually formed based on the desired likelihood ratio, often 1/8 or 1/32. Under some conditions the log likelihood ratio may follow the chi square distribution. If so, then $\alpha = 1 - F(2\log(k), 1)$, where F is a chi-square CDF. `RRlsi()` will make the conversion from α to k if `use.alpha = TRUE`.

The data may also be a matrix. In that case `y` would be entered as `matrix(c(y1, n1-y1, y2, n2-y2), 2, 2, byrow = TRUE)`.

Value

An object of class `rrsi` with the following fields:

<code>estimate</code>	matrix of point and interval estimates - see details
<code>estimator</code>	either "PF" or "RR"
<code>y</code>	data.frame with "y1", "n1", "y2", "n2" values.
<code>rnd</code>	how many digits to round the display
<code>k</code>	likelihood ratio criterion
<code>alpha</code>	complement of confidence level

Author(s)

[PF-package](#)

References

Royall R. *Statistical Evidence: A Likelihood Paradigm*. Chapman & Hall, Boca Raton, 1997. Section 7.6
 Ralston ML, Jennrich RI, 1978. DUD, A Derivative-Free Algorithm for Nonlinear Least Squares. *Technometrics* 20:7-14.

Examples

```
# All examples represent the same observation, with data entry by vector,
# matrix, and formula+data notation.

y_vector <- c(4, 24, 12, 28)
RRlsi(y_vector)

# 1/8 likelihood support interval for PF

# corresponds to 95.858% confidence
# (under certain assumptions)

# PF
#   PF   LL   UL
# 0.6111 0.0168 0.8859

y_matrix <- matrix(c(4, 20, 12, 16), 2, 2, byrow = TRUE)
```

```

y_matrix
#      [, 1] [, 2]
# [1, ]    4   20
# [2, ]   12   16

RRlsi(y_matrix)

# 1/8 likelihood support interval for PF

# corresponds to 95.858% confidence
# (under certain assumptions)

# PF
#   PF    LL    UL
# 0.6111 0.0168 0.8859

require(dplyr)
data1 <- data.frame(group = rep(c("treated", "control"), each = 2),
  y = c(1, 3, 7, 5),
  n = c(12, 12, 14, 14),
  cage = rep(paste('cage', 1:2), 2))

data2 <- data1 %>%
  group_by(group) %>%
  summarize(sum_y = sum(y),
    sum_n = sum(n))
RRlsi(data = data2, formula = cbind(sum_y, sum_n) ~ group,
  compare = c("treated", "control"))

# 1/8 likelihood support interval for PF
#
# corresponds to 95.858% confidence
# (under certain assumptions)
#
# PF
#   PF    LL    UL
# 0.6111 0.0168 0.8859

```

RRmh

Mantel-Haenszel method, CI for common RR over strata or clusters with sparse data.

Description

Estimates confidence intervals for the risk ratio or prevented fraction from clustered or stratified data, using a Mantel-Haenszel estimator for sparse data.

Usage

```

RRmh(
  formula = NULL,
  data = NULL,
  compare = c("vac", "con"),
  Y,

```

```

    alpha = 0.05,
    pf = TRUE,
    rnd = 3
  )

```

Arguments

formula	Formula of the form <code>cbind(y, n) ~ x + cluster(w)</code> , where <code>y</code> is the number positive, <code>n</code> is the group size, <code>x</code> is a factor with two levels of treatment, and <code>w</code> is a factor indicating the clusters.
data	<code>data.frame</code> containing variables for formula
compare	Text vector stating the factor levels: <code>compare[1]</code> is the vaccinate group to which <code>compare[2]</code> (control or reference) is compared.
Y	Matrix of data, $K \times 4$. Each row is a stratum or cluster. The columns are y_1, n_1, y_2, n_2 , where the y 's are the number of positive in each group, and the n is the total in each group. Group 1 corresponds to vaccinates and group 2 are controls or reference. If data entered by formula and dataframe, <code>Y</code> is generated automatically.
alpha	Complement of the confidence level.
pf	Estimate <i>RR</i> or its complement <i>PF</i> ?
rnd	Number of digits for rounding. Affects display only, not estimates.

Details

Based on the Mantel-Haenszel (1959) procedure for sparse data developed by Greenland and Robins (1985). The confidence limits are based on asymptotic normality of the log(risk ratio). Agresti and Hartzel (2000) favor this procedure for small, sparse data sets, but they warn that it is less efficient than maximum likelihood for large data sets.

Value

An object of class `rr1` with the following fields.

estimate	vector of point and interval estimates: point estimate, lower confidence limit, upper confidence limit
estimator	either "PF" or "RR"
y	<code>data.frame</code> of restructured input
rnd	how many digits to round the display
alpha	complement of confidence level

Note

If either all y_1 's or all y_2 's are zero, a division by zero may occur, and a NaN returned for some values.

Vignette *Examples for Stratified Designs* forthcoming with more examples.

Call to this function may be one of two formats: (1) specify data and formula or (2) as a matrix `Y`

```
RRmh(formula, data, compare = c('b', 'a'), pf = TRUE, alpha = 0.05, rnd = 3)
```

```
RRmh(Y, pf = TRUE, alpha = 0.05, rnd = 3)
```


Author(s)[PF-package](#)**References**

Mantel N, Haenszel W, 1959. Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute* 22: 719-748.

Greenland S, Robins JM, 1985. Estimation of a common effect parameter from sparse follow-up data. *Biometrics* 41: 55-68. Errata, 45: 1323-1324.

Agresti A, Hartzel J, 2000. Strategies for comparing treatments on a binary response with multi-centre data. *Statistics in Medicine* 19: 1115-1139.

Lachin JM, 2000. *Biostatistical Methods: The Assessment of Relative Risks* (Wiley, New York), Sec. 4.3.1.

See Also[rr1](#)**Examples**

```
## Table 1 from Gart (1985)
## as data frame

# tx group "b" is control
RRmh(cbind(y, n) ~ tx + cluster(clus),
      Table6,
      compare = c('a', 'b'), pf = FALSE)

# RR
# 95% interval estimates
#
#   RR   LL   UL
# 2.67 1.37 5.23
#

## or as matrix
RRmh(Y = table6, pf = FALSE)

# RR
# 95% interval estimates
#
#   RR   LL   UL
# 2.67 1.37 5.23
```

rrmp-class

*Data class rrmp***Description**

data class rrmp

Fields

- [estimate] vector with point and interval estimate
- [estimator] either "PF" or "IDR"
- [y] data vector
- [rnd] how many digits to round display
- [alpha] complement of c.i.
- [compare] text vector, same as input
- [multvec] data.frame showing the multinomial representation of the data

Author(s)

[PF-package](#)

See Also

[RRmpWald](#)

RRmpWald

Wald confidence intervals for RR from matched pairs

Description

Estimates confidence intervals for the risk ratio or prevented fraction from matched pairs.

Usage

```
RRmpWald(
  formula = NULL,
  data = NULL,
  compare = c("vac", "con"),
  affected = 1,
  x,
  alpha = 0.05,
  pf = TRUE,
  tdist = TRUE,
  df = NULL,
  rnd = 3
)
```

Arguments

formula	Formula of the form $y \sim x + \text{cluster}(w)$, where y is the indicator for an individual's positive response, x is a factor with two levels of treatment, and w identifies the pairs.
data	data.frame containing variables in formula
compare	Text vector stating the factor levels: compare[1] is the vaccinate group to which compare[2] (control or reference) is compared.
affected	Indicator for positive response

x	Alternative data input. Instead of formula and data frame, data may be input as frequency vector. See example for how to order this vector.
alpha	Complement of the confidence level
pf	Estimate <i>RR</i> or its complement <i>PF</i> ?
tdist	Use t distribution?
df	Degrees of freedom. When NULL, the function will default to $df = N - 2$, where N is the total number of pairs.
rnd	Number of digits for rounding. Affects display only, not estimates.

Details

Estimates confidence intervals for the risk ratio or prevented fraction from matched pairs. The response is the tetranomial vector $c(11, 12, 21, 22)$, where the first index is the row and the second index is the column when displayed as a 2x2 table. Wald type confidence intervals are found by applying the delta method to the multinomial variance. This method fails when there are no responders in one of the treatment groups.

Alternative forms of data entry are illustrated by the output, say Y , where $c(Y\$xtable) = Y\$freqvec = Y\$multvec\$Freq$.

If $RR = 0$ ($PF = 1$), the function will return degenerate interval.

Value

A `rrmp` object with the following fields:

estimate	vector of point and interval estimates - see details
estimator	either "PF" or "RR"
compare	text vector, same as input
alpha	complement of confidence level
rnd	how many digits to round the display
multvec	data frame showing the multinomial representation of the data

Note

Experimental functions for estimating profile likelihood intervals are in the CVBmisc package.

Call to this function may be one of two formats: (1) specify data and formula or (2) as a vector x

```
RRmpWald(formula, data, compare = c('vac', 'con'), affected = 1, alpha = 0.05,
pf = TRUE, tdist = TRUE, df = NULL, rnd = 3)
```

```
RRmpWald(x, compare = c('vac', 'con'), affected = 1, alpha = 0.05,
pf = TRUE, tdist = TRUE, df = NULL, rnd = 3)
```

Author(s)

[PF-package](#)

Examples

```

RRmpWald(pos ~ tx + cluster(cage), New, compare = c('vac', 'con'))

# PF
# 95% interval estimates
#
#   PF   LL   UL
# 0.550 0.183 0.752

require(magrittr)
thistable <- New %>%
  tidyr::spread(tx, pos) %>%
  dplyr::mutate(vac = factor(vac, levels = 1:0),
    con = factor(con, levels = 1:0)) %>%
  with(., table(vac, con))
thistable
#   con
# vac  1  0
#   1  7  2
#   0 13  4
as.vector(thistable)
# [1]  7 13  2  4

RRmpWald(x = as.vector(thistable))

# PF
# 95% interval estimates
#
#   PF   LL   UL
# 0.550 0.183 0.752

```

RRor

RR estimate from logistic regression.

Description

Model based interval estimate of the risk ratio or prevented fraction from a logistic regression model.

Usage

```

RRor(
  fit = NULL,
  beta.hat = NULL,
  var.beta.hat = NULL,
  degf = NULL,
  which = c(1, 2),
  pf = TRUE,
  norm = FALSE,
  alpha = 0.05,
  rnd = 3
)

```

Arguments

<code>fit</code>	A glm object.
<code>beta.hat</code>	Parameters estimates from a logistic regression with no intercept.
<code>var.beta.hat</code>	Variance-covariance matrix from a logistic regression with no intercept.
<code>degf</code>	Degrees of freedom.
<code>which</code>	Numeric vector indicating which parameters to compare, so that $RR = \text{compare}[2]/\text{compare}[1]$
<code>pf</code>	Estimate RR or its complement PF ?
<code>norm</code>	Estimate confidence interval using quantiles of Gaussian rather than t distribution quantiles?
<code>alpha</code>	Complement of the confidence level.
<code>rnd</code>	Number of digits for rounding. Affects display only, not estimates.

Details

Estimates confidence intervals using the delta method on parameters from a generalized linear model with logit link.

$RR = \mu_2/\mu_1$, where μ_i are the estimated probabilities from the model.

Value

A [rror](#) object with the following fields.

<code>estimate</code>	vector with point and interval estimate
<code>estimator</code>	either PF or RR
<code>mu</code>	matrix with rows giving probability estimates for each of the groups
<code>rnd</code>	how many digits to round the display
<code>alpha</code>	complement of confidence level
<code>norm</code>	logical indicating Gaussian or t interval
<code>degf</code>	degrees of freedom

Note

Call to this function may be one of two formats: (1) specify `fit` or (2) `beta.hat`, `var.beta.hat`, `degf` `RRor(fit, degf = NULL, pf = TRUE, alpha = 0.05, which = c(1, 2), norm = TRUE, rnd = 3)`

`RRor(beta.hat, var.beta.hat, degf, pf = TRUE, alpha = 0.05, which = c(1, 2), norm = TRUE, rnd = 3)`

Author(s)

[PF-package](#)

See Also

[rror](#), [phiWt](#), [tauWt](#) [StatWI007](#) for more examples

Examples

```

bird.fit <- glm(cbind(y, n - y) ~ tx - 1, binomial, bird)
RRor(tauWt(bird.fit))

# 95% t intervals on 4 df
#
# PF
#   PF      LL      UL
# 0.500 -0.583 0.842
#
#      mu.hat      LL      UL
# txcon 0.733 0.943 0.3121
# txvac 0.367 0.752 0.0997

RRor(phiWt(bird.fit))
# 95% t intervals on 4 df
#
# PF
#   PF      LL      UL
# 0.500 -0.583 0.842
#
#      mu.hat      LL      UL
# txcon 0.733 0.943 0.3121
# txvac 0.367 0.752 0.0997

```

rror-class

*Data class rror***Description**

data class rror

Fields

- [estimate] vector with point and interval estimate
- [estimator] either "PF" or "IDR"
- [y] data vector
- [rnd] how many digits to round display
- [alpha] complement of c.i.
- [norm] logical indicating Gaussian or t interval
- [degf] degrees of freedom
- [mu] matrix with rows giving probability estimates for each of the groups

Author(s)[PF-package](#)**See Also**[RRor](#)

RRotsst

*RR exact CI, OTSST method.***Description**

Estimates confidence interval for the risk ratio or prevented fraction, exact method based on the score statistic (inverts one two-sided test).

Usage

```
RRotsst(
  y = NULL,
  data = NULL,
  formula = NULL,
  compare = c("vac", "con"),
  alpha = 0.05,
  pf = TRUE,
  stepstart = 0.1,
  iter.max = 36,
  converge = 1e-06,
  rnd = 3,
  trace.it = FALSE,
  nuisance.points = 120,
  gamma = 1e-06
)
```

Arguments

y	Data vector c(y1, n1, y2, n2) where y are the positives, n are the total, and group 1 is compared to group 2 (control or reference).
data	data.frame containing variables of the formula.
formula	Formula of the form cbind(y, n) ~ x, where y is the number positive, n is the group size, x is a factor with two levels of treatment.
compare	Text vector stating the factor levels: compare[1] is the vaccinate group to which compare[2] (control or reference) is compared.
alpha	Complement of the confidence level.
pf	Estimate <i>RR</i> or its complement <i>PF</i> ?
stepstart	starting interval for step search
iter.max	Maximum number of iterations
converge	Convergence criterion
rnd	Number of digits for rounding. Affects display only, not estimates.
trace.it	Verbose tracking of the iterations?
nuisance.points	number of points over which to evaluate nuisance parameter
gamma	parameter for Berger-Boos correction (restricts range of nuisance parameter evaluation)

Details

Estimates confidence intervals based on the score statistic that are 'exact' in the sense of accounting for discreteness. The score statistic is used to select tail area tables, and the binomial probability is estimated over the tail area by taking the maximum over the nuisance parameter. Algorithm is a simple step search.

The data may also be a matrix. In that case y would be entered as `matrix(c(y1, n1 - y1, y2, n2 - y2), 2, 2, byrow = TRUE)`.

Value

An object of class `rr1` with the following fields:

<code>estimate</code>	vector with point and interval estimate
<code>estimator</code>	either "PF" or "RR"
<code>y</code>	data.frame with "y1", "n1", "y2", "n2" values.
<code>rnd</code>	how many digits to round the display
<code>alpha</code>	complement of confidence level

Author(s)

[PF-package](#)

References

Koopman PAR, 1984. Confidence intervals for the ratio of two binomial proportions. *Biometrics* 40:513-517.
 Agresti A, Min Y, 2001. On small-sample confidence intervals for parameters in discrete distribution. *Biometrics* 57: 963-971.
 Berger RL, Boos DD, 1994. P values maximized over a confidence set for the nuisance parameter. *Journal of the American Statistical Association* 89:214-220.

See Also

[RRotsst](#), [rr1](#).

Examples

```
# All examples represent the same observation, with data entry by multiple
# options.

y_vector <- c(4, 24, 12, 28)
RRotsst(y_vector, rnd = 3)

# PF
# 95% interval estimates

#   PF    LL    UL
# 0.6111 0.0148 0.8519

y_matrix <- matrix(c(4, 20, 12, 16), 2, 2, byrow = TRUE)
RRotsst(y_matrix, rnd = 3)
```



```

# PF
# 95% interval estimates

#   PF      LL      UL
# 0.6111 0.0148 0.8519

require(dplyr)
data1 <- data.frame(group = rep(c("treated", "control"), each = 2),
  y = c(1, 3, 7, 5),
  n = c(12, 12, 14, 14),
  cage = rep(paste('cage', 1:2), 2))

data2 <- data1 %>%
  group_by(group) %>%
  summarize(sum_y = sum(y),
    sum_n = sum(n))
RRotsst(data = data2, formula = cbind(sum_y, sum_n) ~ group,
  compare = c("treated", "control"))

# PF
# 95% interval estimates
#
# PF      LL      UL
# 0.6111 0.0148 0.8519

```

RRsc

RR score based asymptotic CI.

Description

Estimates confidence intervals for the risk ratio or prevented fraction based on the score statistic.

Usage

```

RRsc(
  y = NULL,
  data = NULL,
  formula = NULL,
  compare = c("vac", "con"),
  alpha = 0.05,
  pf = TRUE,
  trace.it = FALSE,
  iter.max = 18,
  converge = 1e-06,
  rnd = 3
)

```

Arguments

y	Data vector c(y1, n1, y2, n2) where y are the positives, n are the total, and group 1 is compared to group 2 (control or reference group).
data	data.frame containing variables of formula.

formula	Formula of the form <code>cbind(y, n) ~ x</code> , where <code>y</code> is the number positive, <code>n</code> is the group size, <code>x</code> is a factor with two levels of treatment.
compare	Text vector stating the factor levels: <code>compare[1]</code> is the vaccinate group to which <code>compare[2]</code> (control or reference) is compared.
alpha	Complement of the confidence level.
pf	Estimate <i>RR</i> or its complement <i>PF</i> ?
trace.it	Verbose tracking of the iterations?
iter.max	Maximum number of iterations
converge	Convergence criterion
rnd	Number of digits for rounding. Affects display only, not estimates.

Details

Estimates are returned for three estimators based on the score statistic. The score method was introduced by Koopman (1984). Gart and Nam's modification (1988) includes a skewness correction. The method of Miettinen and Nurminen (1985) is a version made slightly more conservative than Koopman's by including a factor of $(N-1)/N$. The starting estimate for the DUD algorithm is obtained by the modified Katz method (log method with 0.5 added to each cell). Both forms of the Katz estimate may be retrieved from the returned object using `RRsc()$estimate`.

The data may also be a matrix. In that case `y` would be entered as `matrix(c(y1, n1-y1, y2, n2-y2), 2, 2, byrow = TRUE)`.

Value

A `rrsc` object with the following fields.

estimate	matrix of point and interval estimates - see details
estimator	either "PF" or "RR"
y	data.frame with "y1", "n1", "y2", "n2" values.
rnd	how many digits to round the display
alpha	complement of confidence level

Author(s)

[PF-package](#)

References

- Gart JJ, Nam J, 1988. Approximate interval estimation of the ratio of binomial parameters: a review and corrections for skewness. *Biometrics* 44:323-338.
- Koopman PAR, 1984. Confidence intervals for the ratio of two binomial proportions. *Biometrics* 40:513-517.
- Miettinen O, Nurminen M, 1985. Comparative analysis of two rates. *Statistics in Medicine* 4:213-226.
- Ralston ML, Jennrich RI, 1978. DUD, A Derivative-Free Algorithm for Nonlinear Least Squares. *Technometrics* 20:7-14.

See Also

[rrsc](#)

Examples

```
# All examples represent the same observation, with data entry by using
# multiple notation options.
```

```
y_vector <- c(4, 24, 12, 28)
RRsc(y_vector)
```

```
# PF
# 95% interval estimates
```

```
# PF      LL      UL
# MN method 0.611 0.0251 0.857
# score method 0.611 0.0328 0.855
# skew corr  0.611 0.0380 0.876
```

```
y_matrix <- matrix(c(4, 20, 12, 16), 2, 2, byrow = TRUE)
#      [, 1] [, 2]
# [1, ]    4   20
# [2, ]   12   16
```

```
RRsc(y_matrix)
```

```
# PF
# 95% interval estimates
```

```
# PF      LL      UL
# MN method 0.611 0.0251 0.857
# score method 0.611 0.0328 0.855
# skew corr  0.611 0.0380 0.876
```

```
require(dplyr)
data1 <- data.frame(group = rep(c("treated", "control"), each = 2),
  y = c(1, 3, 7, 5),
  n = c(12, 12, 14, 14),
  cage = rep(paste('cage', 1:2), 2))
```

```
data2 <- data1 %>%
  group_by(group) %>%
  summarize(sum_y = sum(y),
    sum_n = sum(n))
RRsc(data = data2, formula = cbind(sum_y, sum_n) ~ group,
  compare = c("treated", "control"))
```

```
# PF
# 95% interval estimates
```

```
# PF      LL      UL
# MN method 0.611 0.0251 0.857
# score method 0.611 0.0328 0.855
# skew corr  0.611 0.0380 0.876
```

Description

data class rrsc

Fields

- [estimate] vector with point and interval estimate
- [rnd] how many digits to round display
- [alpha] complement of c.i.
- [estimator] either "PF" or "RR"
- [y] data.frame with restructured input

Author(s)

[PF-package](#)

See Also

[RRsc](#)

rrsi-class

Data class rrsi

Description

data class rrsi

Fields

- [y] data.frame with restructured input
- [k] likelihood ratio criterion
- [rnd] digits to round display
- [alpha] complement of c.i.
- [estimate] vector with point and interval estimate
- [estimator] either "PF" or "IDR"

Author(s)

[PF-package](#)

See Also

[IDRlsi](#), [RRlsi](#)

RRstr

*Gart-Nam method, CI for common RR over strata or clusters.***Description**

Estimates confidence intervals for the risk ratio or prevented fraction from clustered or stratified data.

Usage

```
RRstr(
  formula = NULL,
  data = NULL,
  compare = c("vac", "con"),
  Y,
  alpha = 0.05,
  pf = TRUE,
  trace.it = FALSE,
  iter.max = 24,
  converge = 1e-06,
  rnd = 3,
  multiplier = 0.7,
  divider = 1.1
)
```

Arguments

formula	Formula of the form <code>cbind(y, n) ~ x + cluster(w)</code> , where <code>y</code> is the number positive, <code>n</code> is the group size, <code>x</code> is a factor with two levels of treatment, and <code>w</code> is a factor indicating the clusters.
data	data.frame containing variables of formula
compare	Text vector stating the factor levels: <code>compare[1]</code> is the control or reference group to which <code>compare[2]</code> is compared
Y	Matrix of data. Each row is a stratum or cluster. The columns are <code>y2</code> , <code>n2</code> , <code>y1</code> , <code>n1</code> . If data entered by formula and dataframe, <code>Y</code> is generated automatically.
alpha	Size of the homogeneity test and complement of the confidence level.
pf	Estimate <i>RR</i> or its complement <i>PF</i> ?
trace.it	verbose tracking of the iterations?
iter.max	Maximum number of iterations
converge	Convergence criterion
rnd	Number of digits for rounding. Affects display only, not estimates.
multiplier	internal control parameter for algorithm
divider	internal control parameter for algorithm

Details

Uses the DUD algorithm to estimate confidence intervals by the method of Gart.

Value

A `rrstr` object with the following fields:

<code>estimate</code>	matrix of point and interval estimates - starting value, MLE, and skewness corrected
<code>hom</code>	list of homogeneity statistic, p-value, and degrees of freedom, or error message if appropriate.
<code>estimator</code>	either "PF" or "RR"
<code>y</code>	data.frame of restructured input
<code>compare</code>	groups compared
<code>rnd</code>	how many digits to round the display
<code>alpha</code>	size of test; complement of confidence level

Note

Vignette *Examples for Stratified Designs* forthcoming with more examples.

Call to this function may be one of two formats: (1) specify data and formula or (2) as a matrix `Y`

```
RRstr(formula, data, compare = c('b', 'a'), pf = TRUE, alpha = 0.05, trace.it = FALSE,
iter.max = 24, converge = 1e-6, rnd = 3, multiplier = 0.7, divider = 1.1)
```

```
RRstr(Y, compare = c('b', 'a'), pf = TRUE, alpha = 0.05, trace.it = FALSE, iter.max = 24,
converge = 1e-6, rnd = 3, multiplier = 0.7, divider = 1.1)
```

Author(s)

[PF-package](#)

References

- Gart JJ, 1985. Approximate tests and interval estimation of the common relative risk in the combination of 2×2 tables. *Biometrika* 72:673-677.
- Gart JJ, Nam J, 1988. Approximate interval estimation of the ratio of binomial parameters: a review and corrections for skewness. *Biometrics* 44:323-338.
- Ralston ML, Jennrich RI, 1978. DUD, A Derivative-Free Algorithm for Nonlinear Least Squares. *Technometrics* 20:7-14.

See Also

[rrstr](#)

Examples

```
## Table 1 from Gart (1985)
## as data frame
## "b" is control group
RRstr(cbind(y, n) ~ tx + cluster(clus),
      Table6,
      compare = c('a', 'b'), pf = FALSE)

# Test of homogeneity across clusters
```

```

# stat      0.954
# df        3
# p         0.812

# RR estimates

#          RR   LL   UL
# starting  2.66 1.37 5.18
# mle       2.65 1.39 5.03
# skew corr 2.65 1.31 5.08

## or as matrix
RRstr(Y = table6, pf = FALSE)

tst <- data.frame(y = c(0, 2, 0, 4, 0, 3, 0, 7),
  n = rep(10, 8),
  tx = rep(c('a', 'b'), 4),
  clus = rep(paste('Row', 1:4, sep = ''), each = 2))

```

rrstr-class

*Data class rrstr***Description**

data class rrstr

Fields

- [estimate] vector with point and interval estimate
- [rnd] how many digits to round display
- [alpha] complement of c.i.
- [estimator] either "PF" or "RR"
- [hom]list of homogeneity statistic, p-value, and degrees of freedom. If $\Phi == 0$ | $\Phi == 1$, homogeneity test is not possible and error message displays
- [y]data.frame of restructured input
- [compare] groups compared

Author(s)[PF-package](#)**See Also**[RRstr](#)

RRtosst

*RR exact CI, TOSST method.***Description**

Estimates confidence interval for the risk ratio or prevented fraction; exact method based on the score statistic (inverts two one-sided tests).

Usage

```
RRtosst(
  y = NULL,
  formula = NULL,
  data = NULL,
  compare = c("vac", "con"),
  alpha = 0.05,
  pf = TRUE,
  stepstart = 0.1,
  iter.max = 36,
  converge = 1e-06,
  rnd = 3,
  trace.it = FALSE,
  nuisance.points = 120,
  gamma = 1e-06
)
```

Arguments

y	Data vector c(y1, n1, y2, n2) where y are the positives, n are the total, and group 1 is compared to group 2 (control or reference group).
formula	Formula of the form cbind(y, n) ~ x, where y is the number positive, n is the group size, x is a factor with two levels of treatment.
data	data.frame containing variables of formula.
compare	Text vector stating the factor levels: compare[1] is the vaccinate group to which compare[2] (control or reference) is compared.
alpha	Complement of the confidence level.
pf	Estimate <i>RR</i> or its complement <i>PF</i> ?
stepstart	starting interval for step search
iter.max	Maximum number of iterations
converge	Convergence criterion
rnd	Number of digits for rounding. Affects display only, not estimates.
trace.it	Verbose tracking of the iterations?
nuisance.points	number of points over which to evaluate nuisance parameter
gamma	parameter for Berger-Boos correction (restricts range of nuisance parameter evaluation)

Details

Estimates confidence intervals based on the score statistic that are 'exact' in the sense of accounting for discreteness. Inverts two one-sided score tests. The score statistic is used to select tail area tables, and the binomial probability is estimated over the tail area by taking the maximum over the nuisance parameter. Algorithm is a simple step search.

The data may also be a matrix. In that case y would be entered as `matrix(c(y1, n1-y1, y2, n2-y2), 2, 2, byrow = TRUE)`.

Value

A `rr1` object with the following fields.

<code>estimate</code>	vector with point and interval estimate
<code>estimator</code>	either "PF" or "RR"
<code>y</code>	data.frame with "y1", "n1", "y2", "n2" values.
<code>rnd</code>	how many digits to round the display
<code>alpha</code>	complement of confidence level

Author(s)

[PF-package](#)

References

Koopman PAR, 1984. Confidence intervals for the ratio of two binomial proportions. *Biometrics* 40:513-517.
 Agresti A, Min Y, 2001. On small-sample confidence intervals for parameters in discrete distribution. *Biometrics* 57: 963-971.
 Berger RL, Boos DD, 1994. P values maximized over a confidence set for the nuisance parameter. *Journal of the American Statistical Association* 89:214-220.

See Also

[RRotsst](#), [rr1](#)

Examples

```
# Both examples represent the same observation, with data entry by vector
# and matrix notation.

y_vector <- c(4, 24, 12, 28)
RRtosst(y_vector)

# PF
# 95% interval estimates

#   PF   LL   UL
# 0.611 0.012 0.902

y_matrix <- matrix(c(4, 20, 12, 16), 2, 2, byrow = TRUE)
#      [, 1] [, 2]
# [1, ]    4    20
```

```

# [2, ]    12    16

RRtosst(y_matrix)

# PF
# 95% interval estimates

#    PF    LL    UL
# 0.611 0.012 0.902

require(dplyr)
data1 <- data.frame(group = rep(c("treated", "control"), each = 2),
  y = c(1, 3, 7, 5),
  n = c(12, 12, 14, 14),
  cage = rep(paste('cage', 1:2), 2))
data2 <- data1 %>%
  group_by(group) %>%
  summarize(sum_y = sum(y),
    sum_n = sum(n))
RRtosst(data = data2, formula = cbind(sum_y, sum_n) ~ group,
  compare = c("treated", "control"))

# PF
# 95% interval estimates

#    PF    LL    UL
# 0.611 0.012 0.902

```

rsb

*Rao-Scott weights.***Description**

Rao-Scott weights.

Usage

```
rsb(y = NULL, n = NULL, formula = NULL, data = NULL, id = NULL)
```

Arguments

y	vector of number positive.
n	vector of total number.
formula	Formula of the form <code>cbind(y, n) ~ id</code> , where y is the number positive, n is the total number, id is a factor for estimating the weights by subset.
data	data.frame containing variables of formula.
id	vector of factor for estimating the weights by subset.

Details

Estimates the cluster design effect d_i as the variance inflation due to clustering by the method of Rao and Scott. `rsb` estimates the d_i for use by `rsbwt` or other functions.

Value

A list with the following elements.

w	vector of weights
d	vector of d_i estimates

Author(s)

[PF-package](#)

References

Rao JNK, Scott AJ, 1992. A simple method for the analysis of clustered binary data. *Biometrics* 48:577-585.

See Also

[rsbWt.](#)

Examples

```
# Weil's rat data (Table 1 of Rao and Scott)
rsb(rat$y, rat$n, id = rat$group)$d
# control treated
# 1.232495 3.952861
rsb(data = rat, formula = cbind(y, n) ~ group)$d
# control treated
# 1.232495 3.952861
```

rsbWt	<i>Rao-Scott weighting.</i>
-------	-----------------------------

Description

Rao-Scott weighting of clustered binomial observations.

Usage

```
rsbWt(fit = NULL, subset.factor = NULL, fit.only = TRUE)
```

Arguments

fit	A glm object.
subset.factor	Factor for estimating the weights by subset.
fit.only	Return only the new fit? If FALSE, also returns the weights and phi estimates.

Details

Estimates the cluster design effect d_i as the variance inflation due to clustering by the method of Rao and Scott. Observations are then weighted by the inverse of the d_i .

Value

A list with the following elements.

<code>fit</code>	the new model fit, updated by the estimated weights
<code>weights</code>	vector of weights
<code>d</code>	vector of d_i estimates

Author(s)

[PF-package](#)

References

Rao JNK, Scott AJ, 1992. A simple method for the analysis of clustered binary data. *Biometrics* 48:577-585.

See Also

[RRor](#), [rsb](#).

Examples

```
birdm.fit <- glm(cbind(y,n-y)~tx-1,binomial,birdm)
RRor(rsbWt(birdm.fit))
#
# 95% t intervals on 4 df
#
# PF
#      PF      LL      UL
# 0.479 -1.061 0.868
#
#      mu.hat      LL      UL
# txcon 0.768 0.968 0.2659
# txvac 0.400 0.848 0.0737
#
```

Set1

Set1 dataset

Description

Set1 dataset

Format

a data.frame with 6 observation of the following 4 variables, no NAs

y number positive

n total number in group tx x clus

tx treatment 'vac' or 'con'

clus cluster ID

References

We need some references

set1	<i>set1 dataset</i>
------	---------------------

Description

set1 dataset

Format

a 3 x 4 matrix of data in [Set1](#)

References

we need some references!

Table6	<i>Table6 dataset</i>
--------	-----------------------

Description

Table6 dataset

Format

a data.frame with 8 observations of the following 4 variables, no NAs

y number positive

n total number in group tx x clus

tx treatment 'a' or 'b'

clus cluster ID

References

Table 1 from Gart (1985)

table6	<i>table6 dataset</i>
--------	-----------------------

Description

table6 dataset

Format

matrix for of data in [Table6](#)

tauWt

*Binomial dispersion: intra-cluster correlation parameter.***Description**

MME estimates of binomial dispersion parameter τ (intra-cluster correlation).

Usage

```
tauWt(
  fit,
  subset.factor = NULL,
  fit.only = TRUE,
  iter.max = 12,
  converge = 1e-06,
  trace.it = FALSE
)
```

Arguments

fit	A glm object.
subset.factor	Factor for estimating tau by subset.
fit.only	Return only the final fit? If FALSE, also returns the weights and tau estimates.
iter.max	Maximum number of iterations.
converge	Convergence criterion: difference between model degrees of freedom and Pearson's chi-square. Default 1e-6.
trace.it	Display print statments indicating progress

Details

Estimates binomial dispersion parameter τ by the method of moments. Iteratively refits the model by the Williams procedure, weighting the observations by $1/\phi_{ij}$, where $\phi_{ij} = 1 + \tau_j(n_{ij} - 1)$, j indexes the subsets, and i indexes the observations.

Value

A list with the following elements.

fit	the new model fit, updated by the estimated weights
weights	vector of weights
phi	vector of phi estimates

Author(s)

[PF-package](#)

References

Williams DA, 1982. Extra-binomial variation in logistic linear models. *Applied Statistics* 31:144-148.

Wedderburn RWM, 1974. Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika* 61:439-447.

See Also[phiWt](#), [RRor](#).**Examples**

```
birdm.fit <- glm(cbind(y,n-y)~tx-1, binomial, birdm)
RRor(tauWt(birdm.fit))

# 95% t intervals on 4 df
#
# PF
#      PF      LL      UL
# 0.489 -0.578  0.835
#
#      mu.hat      LL      UL
# txcon  0.737 0.944 0.320
# txvac  0.376 0.758 0.104
#
```

Index

* datasets

bird, [3](#)
birdm, [4](#)
New, [9](#)
rat, [12](#)
Set1, [36](#)
set1, [37](#)
Table6, [37](#)
table6, [37](#)
.rr.score.asymp, [3](#)

bird, [3](#)
birdm, [4](#)

glm, [10](#), [21](#), [35](#), [38](#)

IDRlsi, [4](#), [8](#), [28](#)
IDRsc, [6](#), [7](#), [12](#)

New, [9](#)

pf (pf-class), [10](#)
pf-class, [10](#)
PF-package, [5](#), [8](#), [10–12](#), [14](#), [17–19](#), [21](#), [22](#),
[24](#), [26](#), [28](#), [30](#), [31](#), [33](#), [35](#), [36](#), [38](#)
phiWt, [10](#), [21](#), [39](#)
print.rr1, [11](#)
print.rrmp (print.rr1), [11](#)
print.rror (print.rr1), [11](#)
print.rrsc (print.rr1), [11](#)
print.rrsi (print.rr1), [11](#)
print.rrstr (print.rr1), [11](#)

rat, [12](#)
rr1, [8](#), [10](#), [16](#), [17](#), [24](#), [33](#)
rr1 (rr1-class), [12](#)
rr1-class, [12](#)
RRlsi, [13](#), [28](#)
RRmh, [15](#)
rrmp, [19](#)
rrmp (rrmp-class), [17](#)
rrmp-class, [17](#)
RRmpWald, [18](#), [18](#)
RRor, [11](#), [20](#), [22](#), [36](#), [39](#)
rror, [21](#)
rror (rror-class), [22](#)
rror-class, [22](#)
RRotsst, [12](#), [23](#), [33](#)
RRsc, [25](#), [28](#)
rrsc, [10](#), [26](#)
rrsc (rrsc-class), [27](#)
rrsc-class, [27](#)
rrsi, [5](#), [10](#), [14](#)
rrsi (rrsi-class), [28](#)
rrsi-class, [28](#)
RRstr, [29](#), [31](#)
rrstr, [10](#), [30](#)
rrstr (rrstr-class), [31](#)
rrstr-class, [31](#)
RRtosst, [12](#), [24](#), [32](#)
rsb, [34](#), [36](#)
rsbWt, [35](#), [35](#)

Set1, [36](#), [37](#)
set1, [37](#)

Table6, [37](#), [37](#)
table6, [37](#)
tauWt, [11](#), [21](#), [38](#)