Package 'aods3'

July 8, 2022

Version 0.4-1.2

Title Analysis of Overdispersed Data using S3 Methods

Author Matthieu Lesnoff <matthieu.lesnoff@cirad.fr> and Renaud
    Lancelot <renaud.lancelot@cirad.fr>

Maintainer Renaud Lancelot <renaud.lancelot@cirad.fr>

Depends R (>= 3.0.0), stats

Suggests boot, lme4

Description Provides functions to analyse overdispersed
    counts or proportions. These functions should be considered as
    complements to more sophisticated methods such as generalized
    estimating equations (GEE) or generalized linear mixed effect
    models (GLMM). aods3 is an S3 re-implementation of the
    deprecated S4 package aod.

License GPL (>= 2)

LazyData yes

NeedsCompilation no

Repository CRAN

Date/Publication 2022-07-08 09:25:25 UTC

R topics documented:

---

antibio	*Antibiotics against Shipping Fever in Calves*

Description

Hypothetical drug trial to compare the effect of four antibiotics against Shipping fever in calves (Shoukri and Pause, 1999, Table 3.11).

Usage

```
data(antibio)
```

Format

A data frame with 24 observations on the following 3 variables.

treatment A factor with levels 1, 2, 3 and 4

n A numeric vector: the number of treated animals within a two-week period.

m A numeric vector: the number of deaths at the end of the two weeks.

References

Shoukri, M.M., Pause, C.A., 1999, 2nd ed. *Statistical methods for health sciences*. CRC Press, London.

---

aodml	*ML Estimation of Generalized Linear Models for Overdispersed Count Data*

Description

The function fits a beta-binomial (BB) or a negative binomial (NB) generalized linear model from clustered data.

For the BB model, data have the form $\{(n_1, m_1), (n_2, m_2), ..., (n_N, m_N)\}$, where n_i is the size of cluster i, m_i the number of "successes", and N the number of clusters. The response is the proportion $y = m/n$.

For the NB model, data can be of two types. When modeling simple counts, data have the form $\{m_1, m_2, ..., m_N\}$, where m_i is the number of occurences of the event under study. When modeling rates (e.g. hazard rates), data have the same form as for the BB model, where n_i is the denominator of the rate for cluster i (considered as an "offset", i.e. a constant known value) and m_i the number of occurences of the event. For both types of data, the response is the count $y = m$.

Usage

```
aodml(formula,
  data,
  family = c("bb", "nb"),
  link = c("logit", "cloglog", "probit"),
phi.formula = ~ 1,
  phi.scale = c("identity", "log", "inverse"),
  phi.start = NULL,
  fixpar = list(),
  hessian = TRUE,
  method = c("BFGS", "Nelder-Mead", "CG", "SANN"),
  control = list(maxit = 3000, trace = 0), ...)

  ## S3 method for class 'aodml'
print(x, ...)
  ## S3 method for class 'aodml'
summary(object, ...)
  ## S3 method for class 'aodml'
anova(object, ...)
  ## S3 method for class 'anova.aodml'
print(x, digits, ...)
  ## S3 method for class 'aodml'
fitted(object, ..., what = c("mu", "nu", "eta", "phi"))
  ## S3 method for class 'aodml'
residuals(object, ..., type = c("deviance", "pearson", "response"))
  ## S3 method for class 'aodml'
coef(object, ...)
```

```
  ## S3 method for class 'aodml'
df.residual(object, ...)
  ## S3 method for class 'aodml'
logLik(object, ...)
  ## S3 method for class 'aodml'
deviance(object, ...)
  ## S3 method for class 'aodml'
AIC(object, ..., k = 2)
  ## S3 method for class 'aodml'
vcov(object, ...)
  ## S3 method for class 'aodml'
predict(object, ..., type = c("link", "response"), se.fit = FALSE, newdata = NULL)
```

Arguments

formula	A formula for the mean μ, defining the parameter vector b (see details).
	For the BB model, the left-hand side of the formula must be of the form
	`cbind(m, n - m) ~ ...`
	where the fitted proportion is `m/n`.
	For the NB model, the left-hand side of the formula must be of the form
	`m ~ ...`
	where the fitted count is `m`. To fit a rate, argument `offset` must be used in the right-hand side of the formula (see examples).
data	A data frame containing the response (`m` and, optionnally, `n`) and the explanatory variable(s).
family	Define the model which is fitted: "bb" for the BB model and "nb" for the NB model.
link	For the BB model only. Define the link function g for the mean μ: "cloglog", "logit" (default) or "probit". For the NB model, `link` is automatically set to "log".
phi.formula	A right-hand side formula to model optional heterogeneity for the over-dispersion parameter Φ (see details). Only one single factor is allowed.
	Default to `formula(~ 1)` (i.e. no heterogeneity).
phi.scale	Scale on which Φ is estimated (see details): "identity" (default), "log" or "inverse".
phi.start	Optional starting values for Φ. Default to 0.1.
fixpar	An optional list of 2 vectors of same length ($>= 1$) to set some parameters as constant in the model.
	The first vector indicates which parameters are set as constant (i.e., not optimized) in the global parameter vector (b, Φ).
	The second vector indicates the corresponding values. For instance,
	`fixpar = list(c(4, 5), c(0, 0.1))`
	means that the 4th and 5th components of vector (b, Φ) are set to 0 and 0.1. Argument `fixpar` can be usefull, for instance, to calculate profiled log-likehoods.

hessian	A logical (default to TRUE). If FALSE, the hessian and the variances-covariances matrices of the parameters are not calculated.
method	Define the method used for the optimization (see optim).
control	A list to control the optimization parameters. See optim. By default, the maximum number of iterations is set to 3000, and trace is set to 0 to avoid spurious warnings.
object	An object of class aodml
x	An object of class aodml
digits	Number of digits to print in print.summary.aodml and print.anova.aodml. Default to max(3, getOption("digits") - 3)
...	Further arguments passed to optim (e.g. argument method if using function aodml), or further objects of class aodml (function anova.aodml), or further arguments passed to print.aodml and print.anova.aodml.
what	For function fitted, a character string indicating the type of fitted values to be returned: legal values are "mu" for the fitted response; "nu" for the fitted linear predictor without offset (link scale); "eta" for the fitted linear predictor with offset (link scale); "phi" for the fitted overdispersion coefficient.
type	For function residuals, a character string indicating the type of residuals to be computed; legal values are "deviance" for the deviance's residuals, "pearson" for the Pearson's residuals, and "response" for the response. For function predict, a character string indicating the type of prediction to be computed; legal values are "link" and "response".
k	Numeric scalar for the penalty parameter used to compute the information criterion. The default value ($k = 2$) is the regular AIC = -2 * logLik + 2 * p, where p is the number of model coefficients. NB: for AICc, k is set to 2, and AICc = AIC + 2 * p * $(p + 1)$ / $(n - p - 1)$, with n the number of observations.
se.fit	Logical scalar indicating whether standard errors should be computed for the predicted values. Default to FALSE.
newdata	A data.frame containing the explanatory variables - and possibly the offset - for the values of which predictions are to be made.

Details

Beta-binomial model (BB)

For a given cluster (n, m), the model is

$$m|\lambda, n \ Binomial(n, \lambda)$$

where λ follows a Beta distribution $Beta(a_1, a_2)$. Noting B the beta function, the marginal (beta-binomial) distribution of m is

$$P(m|n) = C(n, m) * B(a_1 + m, a_2 + n - m)/B(a_1, a_2)$$

Using the re-parameterization $\mu = a_1/(a_1 + a_2)$ and $\Phi = 1/(a_1 + a_2 + 1)$, the marginal mean and variance are

$$E[m] = n * \mu$$

$$Var[m] = n * \mu * (1 - \mu) * (1 + (n - 1) * \Phi)$$

The response in aodml is $y = m/n$. The mean is $E[y] = \mu$, defined such as $\mu = g^{-1}(X * b) = g^{-1}(\nu)$, where g is the link function, X is a design-matrix, b a vector of fixed effects and $\nu = X * b$ is the corresponding linear predictor. The variance is $Var[y] = (1/n) * \mu * (1 - \mu) * (1 + (n - 1) * \Phi)$.

Negative binomial model (NB)

—— Simple counts (model with no offset)

For a given cluster (m), the model is

$$y|\lambda \; Poisson(\lambda)$$

where λ follows a Gamma distribution of mean μ and shape k ($Var[\lambda] = \mu^2/k$). Noting G the gamma function, the marginal (negative binomial) distribution of m is

$$P(m) = G(m + k)/(m! * G(k)) * (k/(k + \mu))^k * (\mu/(k + \mu))^m$$

Using the re-parameterization $\Phi = 1/k$, the marginal mean and variance are

$$E[m] = \mu$$

$$Var[m] = \mu + \Phi * \mu^2$$

The response in aodml is $y = m$. The mean is $E[y] = \mu$, defined such as $\mu = exp(X * b) = exp(\nu)$. The variance is $Var[y] = \mu + \Phi * \mu^2$.

—— Rates (model with offset)

For a given cluster (n, m), the model is

$$m|\lambda, n \; Poisson(\lambda)$$

The marginal (negative binomial) distribution $P(m|n)$ is the same as for the case with no offset (= $P(m)$). The response in aodml is $y = m$. The mean is $E[y] = \mu$, defined such as $\mu = exp(X * b + log(n)) = exp(\nu + log(n)) = exp(\eta)$, where $log(n)$ is the offset. The variance

is $Var[y] = \mu + \Phi * \mu^2$.

Other details

Argument `phi.scale` of function `aodml` enables to estimate the over-dispersion parameter under different scales.

If `phi.scale = "identity"` (Default), the function estimates Φ.

If `phi.scale = "log"`, the function estimates $log(\Phi)$.

If `phi.scale = "inverse"`, the function estimates $1/\Phi$.

The full parameter vector returned by `aodml`, say `param`, is equal to (b, Φ). This vector is estimated by maximizing the log-likelihood of the marginal model using function `optim`. The estimated variances-covariances matrix of `param` is calculated by the inverse of the observed hessian matrix returned by `optim`, and is referred to as `varparam`.

Value

An object of class `aodml`, printed and summarized by various functions. Function `deviance.aodml` returns the value `-2 * (logL - logL_max)`. The "deviance" used in function `AIC.aodml` to calculate AIC and AICc is `-2 * logL`.

References

Crowder, M.J., 1978. *Beta-binomial anova for proportions*. Appl. Statist. 27, 34-37.

Griffiths, D.A., 1973. *Maximum likelihood estimation for the beta-binomial distribution and an application to the household distribution of the total number of cases of disease*. Biometrics 29, 637-648.

Lawless, J.F., 1987. *Negative binomial and mixed Poisson regression*. The Canadian Journal of Statistics, 15(3): 209-225.

McCullagh, P., Nelder, J. A., 1989, 2nd ed. *Generalized linear models*. New York, USA: Chapman and Hall.

Prentice, R.L., 1986. *Binary regression using an extended beta-binomial distribution, with discussion of correlation induced by covariate measurement errors*. J.A.S.A. 81, 321-327.

Williams, D.A., 1975. *The analysis of binary responses from toxicological experiments involving reproduction and teratogenicity*. Biometrics 31, 949-952.

See Also

`glm` and `optim`

Examples

```
#------ Beta-binomial model

data(orob2)
fm1 <- aodml(cbind(m, n - m) ~ seed, data = orob2, family = "bb")
```

```
# summaries
fm1
summary(fm1)
coef(fm1)
vcov(fm1)
logLik(fm1)
deviance(fm1)
AIC(fm1)
gof(fm1)

# predictions
cbind(
  fitted(fm1),
  fitted(fm1, what = "nu"),
  fitted(fm1, what = "eta"),
  fitted(fm1, what = "phi")
)
predict(fm1, type = "response", se.fit = TRUE)
newdat <- data.frame(seed = c("O73", "O75"))
predict(fm1, type = "response", se.fit = TRUE, newdata = newdat)

# model with heterogeneity in phi
fm <- aodml(cbind(m, n - m) ~ seed, data = orob2,
  family = "bb", phi.formula = ~ seed)
summary(fm)
AIC(fm1, fm)

# various phi scales
fm <- aodml(cbind(m, n - m) ~ seed, data = orob2, family = "bb")
fm$phi
fm$phi.scale
fm <- aodml(cbind(m, n - m) ~ seed, data = orob2, family = "bb",
            phi.scale = "log")
fm$phi
fm$phi.scale
fm <- aodml(cbind(m, n - m) ~ seed, data = orob2, family = "bb",
            phi.scale = "inverse")
fm$phi
fm$phi.scale

### Models with coefficient(s) set as constant

# model with 1 phi coefficient, set as constant "0.02"
fm <- aodml(formula = cbind(m, n - m) ~ seed * root, data = orob2,
  family = "bb", fixpar = list(5, 0.02))
fm$param
fm$varparam

# model with 2 phi coefficients, with the first set as constant ~ "0"
fm <- aodml(formula = cbind(m, n - m) ~ seed * root, data = orob2,
  family = "bb", phi.formula = ~ seed, fixpar = list(5, 1e-15))
fm$param
fm$varparam
```

```
# model with 2 phi coefficients, with the first set as constant ~ "0",
# and the mu intercept (1rst coef of vector b) set as as constant "-0.5"
fm <- aodml(formula = cbind(m, n - m) ~ seed * root, data = orob2,
             family = "bb", phi.formula = ~ seed,
             fixpar = list(c(1, 5), c(-0.5, 1e-15)))
fm$param
fm$varparam

### Model tests

# LR tests - non-constant phi
fm0 <- aodml(cbind(m, n - m) ~ 1, data = orob2, family = "bb")
fm2 <- aodml(cbind(m, n - m) ~ seed + root, data = orob2, family = "bb")
fm3 <- aodml(cbind(m, n - m) ~ seed * root, data = orob2, family = "bb")
anova(fm0, fm1, fm2, fm3)

# LR tests - constant phi
# phi is assumed to be estimated from fm3
fm2.bis <- aodml(cbind(m, n - m) ~ seed  + root, data = orob2,
                   family = "bb", fixpar = list(4, fm3$phi))
LRstat <- 2 * (logLik(fm3) - logLik(fm2.bis))
pchisq(LRstat, df = 1, lower.tail = FALSE)

# Wald test of the seed factor in fm1
wald.test(b = coef(fm3), varb = vcov(fm3), Terms = 4)

#------ Negative binomial model

### Modelling counts

data(salmonella)
fm1 <- aodml(m ~ log(dose + 10) + dose, data = salmonella, family = "nb")
## fm1 <- aodml(m ~ log(dose + 10) + dose, data = salmonella, family = "nb",
##               method = "Nelder-Mead")

# summaries
fm1
summary(fm1)
coef(fm1)
vcov(fm1)
logLik(fm1)
deviance(fm1)
AIC(fm1)
gof(fm1)

# predictions
cbind(
  fitted(fm1),
  fitted(fm1, what = "nu"),
  fitted(fm1, what = "eta"),
  fitted(fm1, what = "phi")
)
```

```
predict(fm1, type = "response", se.fit = TRUE)
newdat <- data.frame(dose = c(20, 40))
predict(fm1, type = "response", se.fit = TRUE, newdata = newdat)

# various phi scales
fm <- aodml(m ~ log(dose + 10) + dose, data = salmonella, family = "nb")
fm$phi
fm$phi.scale
fm <- aodml(m ~ log(dose + 10) + dose, data = salmonella,
            family = "nb", phi.scale = "log")
fm$phi
fm$phi.scale
fm <- aodml(m ~ log(dose + 10) + dose, data = salmonella,
            family = "nb", phi.scale = "inverse")
fm$phi
fm$phi.scale

# LR and Wald tests of the "log(dose + 10) + dose" factors
fm0 <- aodml(m ~ 1, data = salmonella, family = "nb")
anova(fm0, fm1)
fm0.bis <- aodml(m ~ 1, data = salmonella, family = "nb",
                 fixpar = list(2, fm1$phi))
LRstat <- 2 * (logLik(fm1) - logLik(fm0.bis))
pchisq(LRstat, df = 2, lower.tail = FALSE)
wald.test(b = coef(fm1), varb = vcov(fm1), Terms = 2:3)

### Modelling a rate

data(dja)
# rate "m / trisk"
fm <- aodml(formula = m ~ group + offset(log(trisk)),
            data = dja, family = "nb")
summary(fm)

fm <- aodml(formula = m ~ group + offset(log(trisk)),
            phi.formula = ~ group, data = dja, family = "nb",
            phi.scale = "log")
summary(fm)

# model with 1 phi coefficient, set as constant "0.8"
fm <- aodml(formula = m ~ group + offset(log(trisk)), data = dja,
            family = "nb", phi.formula = ~1, fixpar = list(3, 0.8))
fm$param
fm$varparam

# model with 2 phi coefficients, with the first set as constant ~ "0" in the identity scale
fm <- aodml(formula = m ~ group + offset(log(trisk)), data = dja,
            family = "nb", phi.formula = ~ group, phi.scale = "log",
            fixpar = list(4, -15))
fm$param
fm$varparam

# model with 2 phi coefficients, with the first set as constant "0" in the identity scale,
```

```
# and the mu intercept coefficient (1rst coef of vector b) set as as constant "-0.5"
fm <- aodml(formula = m ~ group + offset(log(trisk)), data = dja,
  family = "nb", phi.formula = ~ group, phi.scale = "log",
  fixpar = list(c(1, 4), c(-0.5, -15)))
fm$param
fm$varparam
```

---

aodql	*QL/MM Estimation of Generalized Linear Models for Overdispersed Count Data*

Description

From clustered data, the function fits generalized linear models containing an over-dispersion parameter Φ using quasi-likelihood estimating equations for the mean μ and a moment estimator for Φ.

For binomial-type models, data have the form $\{(n_1, m_1), (n_2, m_2), ..., (n_N, m_N)\}$, where n_i is the size of cluster i, m_i the number of "successes", and N the number of clusters. The response is the proportion $y = m/n$.

For Poisson-type models, data can be of two forms. When modeling "simple counts", data have the form $\{m_1, m_2, ..., m_N\}$, where m_i is the number of occurences of the event under study. When modeling rates (e.g. hazard rates), data have the same form as for the BB model, where n_i is the denominator of the rate for cluster i (considered as an "offset", i.e. a constant known value) and m_i the number of occurences of the event. For both forms of data, the response is the count $y = m$.

Usage

```
aodql(formula,
      data,
      family = c("qbin", "qpois"),
      link = c("logit", "cloglog", "probit"),
      method = c("chisq", "dev"),
      phi = NULL,
      tol = 1e-5, ...)

  ## S3 method for class 'aodql'
anova(object, ...)
  ## S3 method for class 'aodql'
coef(object, ...)
  ## S3 method for class 'aodql'
deviance(object, ...)
  ## S3 method for class 'aodql'
df.residual(object, ...)
  ## S3 method for class 'aodql'
fitted(object, ...)
  ## S3 method for class 'aodql'
```

```
logLik(object, ...)
  ## S3 method for class 'aodql'
predict(object, ...)
  ## S3 method for class 'aodql'
print(x, ...)
  ## S3 method for class 'aodql'
residuals(object, ...)
  ## S3 method for class 'aodql'
summary(object, ...)
  ## S3 method for class 'aodql'
vcov(object, ...)
```

Arguments

formula
A formula for the mean μ, defining the parameter vector b (see details). For binomial-type models, the left-hand side of the formula must be of the form `cbind(m, n - m) ~ ...` where the fitted proportion is `m/n`. For Poisson-type models, the left-hand side of the formula must be of the form `m ~ ...` where the fitted count is `m`. To fit a rate, argument `offset` must be used in the right-hand side of the formula (see examples).

data
A data frame containing the response (m and, optionnally, n) and the explanatory variable(s).

family
Define the model which is fitted: "qbin" for binomial-type models and "qpois" for Poisson-type models.

link
For binomial-type models only. Define the link function g for the mean μ: "cloglog", "logit" (default) or "probit". For Poisson-type models, `link` is automatically set to "log".

method
For function `aodql`, define the statistics used for the moment estimation of phi; legal values are "chisq" (default) for the chi-squared statistics or "dev" for the deviance statistics.

phi
An optional value defining the over-dispersion parameter Φ if it is set as constant. Default to NULL (in that case, Φ is estimated).

tol
A positive scalar (default to 0.001). The algorithm stops at iteration $r + 1$ when the condition $\chi^2[r + 1] - \chi^2[r] <= tol$ is met by the χ^2 statistics .

...
Further arguments to passed to the appropriate functions.

object
An object of class "aodql".

x
An object of class "aodql".

Details

Binomial-type models

For a given cluster (n, m), the model is

$$m|\lambda, n \sim Binomial(n, \lambda)$$

where λ follows a random variable of mean $E[\lambda] = \mu$ and variance $Var[\lambda] = \Phi * \mu * (1 - \mu)$. The marginal mean and variance of m are

$$E[m] = n * \mu$$

$$Var[m] = n * \mu * (1 - \mu) * (1 + (n - 1) * \Phi)$$

The response in `aodql` is $y = m/n$. The mean is $E[y] = \mu$, defined such as $\mu = g^{-1}(X * b) = g^{-1}(\nu)$, where g is the link function, X is a design-matrix, b a vector of fixed effects and $\nu = X * b$ is the corresponding linear predictor. The variance is $Var[y] = (1/n) * \mu * (1 - \mu) * (1 + (n - 1) * \Phi)$.

Poisson-type models

—— *Simple counts (model with no offset)*

For a given cluster (m), the model is

$$y|\lambda \sim Poisson(\lambda)$$

where λ follows a random distribution of mean μ and variance $\Phi * \mu^2$. The mean and variance of the marginal distribution of m are

$$E[m] = \mu$$

$$Var[m] = \mu + \Phi * \mu^2$$

The response in `aodql` is $y = m$. The mean is $E[y] = \mu$, defined such as $\mu = exp(X * b) = exp(\nu)$. The variance is $Var[y] = \mu + \Phi * \mu^2$.

—— *Rates (model with offset)*

For a given cluster (n, m), the model is

$$m|\lambda, n \sim Poisson(\lambda)$$

where λ follows the same random distribution as for the case with no offset. The marginal mean and variance are

$$E[m|n] = \mu$$

$$Var[m|n] = \mu + \Phi * \mu^2$$

The response in aodql is $y = m$. The mean is $E[y] = \mu$, defined such as $\mu = exp(X*b+log(n)) = exp(\nu + log(n)) = exp(\eta)$, where $log(n)$ is the offset. The variance is $Var[y] = \mu + \Phi * \mu^2$.

Other details

Vector b and parameter Φ are estimated iteratively, using procedures referred to as "Model I" in Williams (1982) for binomial-type models, and "Procedure II" in Breslow (1984) for Poisson-type models.

Iterations are as follows. Quasi-likelihood estimating equations (McCullagh & Nelder, 1989) are used to estimate b (using function glm and its weights argument), Φ being set to a constant value. Then, Φ is calculated by the moment estimator, obtained by equalizing the goodness-of-fit statistic (chi-squared X2 or deviance D) of the model to its degrees of freedom.
Parameter Φ can be set as constant, using argument phi. In that case, only b is estimated.

Value

An object of class aodql, printed and summarized by various functions.

References

Breslow, N.E., 1984. *Extra-Poisson variation in log-linear models*. Appl. Statist. 33, 38-44.
Moore, D.F., 1987, *Modelling the extraneous variance in the presence of extra-binomial variation*. Appl. Statist. 36, 8-14.
Moore, D.F., Tsiatis, A., 1991. *Robust estimation of the variance in moment methods for extra-binomial and extra-poisson variation*. Biometrics 47, 383-401. McCullagh, P., Nelder, J. A., 1989, 2nd ed. *Generalized linear models*. New York, USA: Chapman and Hall.
Williams, D.A., 1982, *Extra-binomial variation in logistic linear models*. Appl. Statist. 31, 144-148.

See Also

glm

Examples

```
#------ Binomial-type models

data(orob2)
fm <- aodql(cbind(m, n - m) ~ seed, data = orob2, family = "qbin")
coef(fm)
vcov(fm)
summary(fm)
# chi2 tests of the seed factor in fm
wald.test(b = coef(fm), varb = vcov(fm), Terms = 2)

# chi-2 vs. deviance statistic to estimate phi
fm1 <- aodql(cbind(m, n - m) ~ seed + root, data = orob2, family = "qbin")
fm2 <- aodql(cbind(m, n - m) ~ seed + root, data = orob2, family = "qbin", method = "dev")
```

```
coef(fm1)
coef(fm2)
fm1$phi
fm2$phi
vcov(fm1)
vcov(fm2)
gof(fm1)
gof(fm2)

# estimate with fixed phi
fm <- aodql(cbind(m, n - m) ~ seed, data = orob2, family = "qbin", phi = 0.05)
coef(fm)
vcov(fm)
summary(fm)

#------ Poisson-type models

data(salmonella)
fm <- aodql(m ~ log(dose + 10) + dose, data = salmonella, family = "qpois")
coef(fm)
vcov(fm)
summary(fm)
# chi2 tests of the "log(dose + 10) + dose" factors
wald.test(b = coef(fm), varb = vcov(fm), Terms = 2:3)

# chi-2 vs. deviance statistic to estimate phi
fm1 <- aodql(m ~ log(dose + 10) + dose, data = salmonella, family = "qpois")
fm2 <- aodql(m ~ log(dose + 10) + dose, data = salmonella, family = "qpois", method = "dev")
coef(fm1)
coef(fm2)
fm1$phi
fm2$phi
vcov(fm1)
vcov(fm2)
gof(fm1)
gof(fm2)

# estimate with fixed phi
fm <- aodql(m ~ log(dose + 10) + dose, data = salmonella, family = "qpois", phi = 0.05)
coef(fm)
vcov(fm)
summary(fm)

# modelling a rate
data(dja)
# rate "m / trisk"
fm <- aodql(formula = m ~ group + offset(log(trisk)), data = dja, family = "qpois")
summary(fm)
```

aods3-pkg   *Analysis of Overdispersed Data*

Description

This package provides functions to analyse overdispersed counts or proportions. The functions should be considered as complements to more sophisticated methods such as generalized estimating equations (GEE) or generalized linear mixed effect models (GLMM).

Details

Functions Index :

aodml	ML Estimation of Generalized Linear Models for Overdispersed Count Data
aodql	QL/MM Estimation of Generalized Linear Models for Overdispersed Count Data
drs	Test of Proportion Homogeneity between Groups using Donner's and Rao-Scott's Adjustments
gof	Test of Goodness-of-Fit of Models for Count data
iccbin	Intra-Cluster Correlation for Clustered Binomial data
invlink	Transformation from the Link Scale to the Observation Scale
link	Transformation from the Observation Scale to the Link Scale
splitbin	Split Grouped Data Into Individual Data
varbin	Estimate of a Probability from Clustered Binomial Data
wald.test	Wald Test for Model Coefficients

Data sets Index :

antibio	Antibiotics against Shipping Fever in Calves
cohorts	Data set: Age, Period and Cohort Effects for Vital Rates
dja	Mortality of Djallonke Lambs in Senegal
lizards	A Comparison of Site Preferences of Two Species of Lizard
mice	Pregnant Female Mice Experiment
orob1	Germination Data
orob2	Germination Data
rabbits	Rabbits Foetuses Survival Experiment
rats	Rats Diet Experiment
salmonella	Salmonella Reverse Mutagenicity Assay

Author(s)

Matthieu Lesnoff <matthieu.lesnoff@cirad.fr> and Renaud Lancelot <renaud.lancelot@cirad.fr>
Maintainer: Renaud Lancelot <renaud.lancelot@cirad.fr>

---

cohorts   *Age, Period and Cohort Effects for Vital Rates*

Description

Number of prostate cancer deaths and midperiod population for nonwhites in the USA by age and period. The cohort index k is related to age and period indices (i and j, respectively) by $k = j + I - i$, where $I = max(i)$ (Holford, 1983, Table 2).

Usage

```
data(cohorts)
```

Format

A data frame with 49 observations on the following 4 variables.

period A factor with levels 1935-, 1940-, ..., 1965-.

age A factor with levels 50-, 55-, ..., 80-.

m Numeric: the number of prostate cancer deaths.

n Numeric: the midperiod population size.

References

Holford, T.R., 1983. *The estimation of age, period and cohort effects for vital rates*. Biometrics 39, 311-324.

---

dja   *Mortality of Djallonke Lambs in Senegal*

Description

Field trial to assess the effect of ewes deworming (prevention of gastro-intestinal parasitism) on the mortality of their offspring (age < 1 year). This data set is extracted from a large database on small ruminants production and health in Senegal (Lancelot et al., 1998). Data were collected in a sample of herds in Kolda (Upper Casamance, Senegal) during a multi-site survey (Faugère et al., 1992). See also the references below for a presentation of the follow-up survey (Faugère and Faugère, 1986) and a description of the farming systems (Faugère et al., 1990).

Usage

```
data(dja)
```

Format

A data frame with 21 observations on the following 4 variables.

group a factor with 2 levels: CTRL and TREAT, indicating the treatment.

village a factor indicating the village of the herd.

herd a factor indicating the herd.

n a numeric vector: the number of animals exposed to mortality.

trisk a numeric vector: the exposition time to mortality (in year).

m a numeric vector: the number of deaths.

References

Faugère, O., Faugère, B., 1986. *Suivi de troupeaux et contrôle des performances individuelles des petits ruminants en milieu traditionnel africain. Aspects méthodologiques.* Rev. Elev. Méd. vét. Pays trop., 39 (1): 29-40.

Faugère, O., Dockès, A.-C., Perrot, C., Faugère, B., 1990. *L'élevage traditionnel des petits ruminants au Sénégal. I. Pratiques de conduite et d'exploitation des animaux chez les éleveurs de la région de Kolda.* Revue Elev. Méd. vét. Pays trop. 43: 249-259.

Faugère, O., Tillard, E., Faugère, B., 1992. *Prophylaxie chez les petits ruminants au Sénégal : régionalisation d'une politique nationale de protection sanitaire.* In: B. Rey, S. H. B. Lebbie, L. Reynolds (Ed.), First biennial conference of the African Small Ruminant Research Network, ILCA, 1990, ILRAD, Nairobi, pp. 307-314.

Lancelot, R., Faye, B., Juanès, X., Ndiaye, M., Pérochon, L., Tillard, E., 1998. *La base de données BAOBAB: un outil pour modéliser la production et la santé des petits ruminants dans les systèmes d'élevage traditionnels au Sénégal.* Revue Elev. Méd. vét. Pays trop., 51 (2): 135-146.

---

drs	*Test of Proportion Homogeneity between Groups using Donner's and Rao-Scott's Adjustments*

Description

The function tests the homogeneity of probabilities between J groups (H_0: $\mu_1 = \mu_2 = ... = \mu_J$) from clustered binomial data $\{(n_1, m_1), (n_2, m_2), ..., (n_N, m_N)\}$, where n_i is the size of cluster i, m_i the number of "successes" (proportions are $y = m/n$), and N the number of clusters. The function uses adjusted chi-squared statistics, with either the correction proposed by proposed by Donner (1989) or the correction proposed by Rao and Scott (1993).

Usage

```
drs(formula, data, type = c("d", "rs"), C = NULL, pooled = FALSE)

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

Arguments

formula	An formula where the left-hand side is a matrix of the form cbind(m, n-m) (the modelled proportion is m/n). The right-hand side must specify a single grouping variable.
type	A character string: either "d" for the Donner's test and "rs" for the Rao and Scott's test.
data	A data frame containing n, m) and the grouping variable.

C	An optional vector of a priori J cluster correction factors used for the Donner's test or design effects factors used for the Rao-Scott's test. If C is set no NULL (default), it is calculated internally (see details).
pooled	Logical indicating if a pooled design effect is estimated over the J groups for the Rao-Scott's test (see details). Default to FALSE.
x	An object of class "drf".
...	Further arguments to be passed to `print`.

Details

Donner's test

The chi-squared statistic is adjusted with the correction factor C_j computed in each group j. The test statistic is given by:

$$X^2 = \sum_j ((m_j - n_j * \mu)^2 / (C_j * n_j * \mu * (1 - \mu)))$$

where $\mu = \sum_j (m_j) / \sum_j (n_j)$ and $C_j = 1 + (n_{A,j} - 1) * \rho$. $n_{A,j}$ is a scalar depending on the cluster sizes, and ρ is the ANOVA estimate of the intra-cluster correlation assumed common across groups (see Donner, 1989 or Donner et al., 1994). The statistic is compared to a chi-squared distribution with $J - 1$ degrees of freedom. Fixed correction factors C_j can be specified with the argument C.

Rao ans Scott's test

The method uses design effects and "effective" sample sizes. The design effect C_j in each group j is estimated by $C_j = v_{ratio,j} / v_{bin,j}$, where $v_{ratio,j}$ is the variance of the ratio estimate of the probability in group i (Cochran, 1999, p. 32 and p. 66) and $v_{bin,j}$ is the standard binomial variance. The C_j are used to compute the effective sample sizes $n_{adj,j} = n_j / C_j$, the effective numbers of successes $m_{adj,j} = m_j / C_j$ in each group j, and the overall effective proportion $mu_a dj = \sum_j m_{adj,j} / \sum_j C_j$. The test statistic is obtained by substituting these quantities in the usual chi-squared statistic, yielding:

$$X^2 = \sum_j ((m_{adj,j} - n_{adj,j} * muadj)^2 / (n_{adj,j} * muadj * (1 - muadj)))$$

which is compared to a chi-squared distribution with $J - 1$ degrees of freedom.
A pooled design effect over the J groups is estimated if argument `pooled = TRUE` (see Rao and Scott, 1993, Eq. 6). Fixed design effects C_j can be specified with the argument C.

Value

An object of class `drs`, printed with `print.drs`.

References

Donner, A., 1989. *Statistical methods in ophthalmology: an adjusted chi-squared approach*. Biometrics 45, 605-611.

Donner, A., 1993. *The comparison of proportions in the presence of litter effects*. Prev. Vet. Med. 18, 17-26.

Donner, A., Eliasziw, M., Klar, N., 1994. *A comparison of methods for testing homogeneity of proportions in teratologic studies*. Stat. Med. 13, 1253-1264.

See Also

chisq.test

Examples

```
data(dja)
# Donner
drs(formula = cbind(m, n - m) ~ group, data = dja, type = "d")
# Rao and Scott
drs(formula = cbind(m, n - m) ~ group, type = "rs", data = dja)
drs(formula = cbind(m, n - m) ~ group, type = "rs", data = dja, pooled = TRUE)
# standard chi2 test
drs(formula = cbind(m, n - m) ~ group, data = dja, type = "d", C = c(1:1))
drs(formula = cbind(m, n - m) ~ group, data = dja, type = "rs", C = c(1:1))
```

---

gof   *Test of Goodness-of-Fit of Models for Count data*

Description

The function returns a chi-squared test of goodness of fit for models of class glm, aodml or aodql.

Usage

```
gof(object)
gof.default(object)

## S3 method for class 'gof'
print(x, ..., digits =  max(3, getOption("digits") - 3))
```

Arguments

object	An object of class glm, aodml or aodquasi.
x	An object of class gof.
digits	A numerical scalar indicating the number of digits to be printed after the decimal place.
...	Further arguments passed to print.

Details

Function gof calculates the deviance D and the Pearson chi-squared X^2 statistics for the model under consideration. Let y be the observed response, and $E[y] = \mu$ and $Var[y]$ its mean and variance estimated from the model, statistic X^2 is calculated by:

$$X^2 = \sum_i ((y_i - \mu)^2 / Var[y_i])$$

Assuming that the data length is N and the number of the parameters in the model is p, eqnD and eqnX^2 are compared to a chi-squared distribution with $N - p$ degrees of freedom.

Value

An object of class gof, printed with print.gof.

References

Agresti, A. *Categorical data analysis*. Wiley, 1990.

See Also

residuals, chisq.test

Examples

```
data(orob2)
fm1 <- glm(cbind(m, n - m) ~ seed, data = orob2, family = binomial)
fm2 <- aodml(cbind(m, n - m) ~ seed, data = orob2, family = "bb")
gof(fm1)
gof(fm2)
```

---

iccbin   *Intra-Cluster Correlation for Clustered Binomial data*

Description

The function estimates the intraclass correlation ρ from clustered binomial data:

$$\{(n_1, m_1), (n_2, m_2), ..., (n_N, m_N)\},$$

where n_i is the size of cluster i, m_i the number of "successes" (proportions are $y = m/n$), and N the number of clusters. The function uses a one-way random effect model. Three estimates, corresponding to methods referred to as "A", "B" and "C" in Goldstein et al. (2002), can be returned.

Usage

```
   iccbin(n, m, method = c("A", "B", "C"), nAGQ = 1, M = 1000)

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

Arguments

n	A vector of the sizes of the clusters.
m	A vector of the numbers of successes (proportions are eqny = m / n).
method	A character ("A", "B" or "C") defining the calculation method. See Details.
nAGQ	Same as in function glmer of package **lme4**. Only for methods "A" and "B". Default to 1.
M	Number of Monte Carlo (MC) replicates used in method "B". Default to 1000.
x	An object of class "iccbin".
...	Further arguments to ba passed to "print".

Details

Before computations, the clustered data are split to binary "0/1" observations y_{ij} (observation j in cluster i). The methods of calculation are described in Goldstein et al. (2002).

Methods "A" and "B" use the 1-way logistic binomial-Gaussian model

$$y_{ij}|\mu_{ij} \sim Bernoulli(\mu_{ij})$$

$$logit(\mu_{ij}) = b_0 + u_i,$$

where b_0 is a constant and u_i a cluster random effect with $u_i \sim N(0, s_u^2)$. The ML estimate of the variance component s_u^2 is calculated with the function glmer of package **lme4**. The intra-class correlation $\rho = Corr[y_{ij}, y_{ik}]$ is then calculated from a first-order model linearization around $E[u_i] = 0$ in method "A", and with Monte Carlo simulations in method "B".

Method "C" provides the common ANOVA (moment) estimate of ρ. For details, see for instance Donner (1986), Searle et al. (1992) or Ukoumunne (2002).

Value

An object of class iccbin, printed with print.iccbin.

References

Donner A., 1986, *A review of inference procedures for the intraclass correlation coefficient in the one-way random effects model*. International Statistical Review 54, 67-82.
Searle, S.R., Casella, G., McCulloch, C.E., 1992. *Variance components*. Wiley, New York.
Ukoumunne, O. C., 2002. *A comparison of confidence interval methods for the intraclass correlation coefficient in cluster randomized trials*. Statistics in Medicine 21, 3757-3774.
Golstein, H., Browne, H., Rasbash, J., 2002. *Partitioning variation in multilevel models*. Understanding Statistics 1(4), 223-231.

See Also

glmer

Examples

```
data(rats)
z <- rats[rats$group == "TREAT", ]
# A: glmm (model linearization)
iccbin(z$n, z$m, method = "A")
iccbin(z$n, z$m, method = "A", nAGQ = 10)
# B: glmm (Monte Carlo)
```

```
iccbin(z$n, z$m, method = "B")
iccbin(z$n, z$m, method = "B", nAGQ = 10, M = 1500)
# C: lmm (ANOVA moments)
iccbin(z$n, z$m, method = "C")

## Not run:
  # Example of CI calculation with nonparametric bootstrap
  require(boot)
  foo <- function(X, ind) {
    n <- X$n[ind]
    m <- X$m[ind]
    iccbin(n = n, m = m, method = "C")$rho
    }
 res <- boot(data = z[, c("n", "m")], statistic = foo, R = 500, sim = "ordinary", stype = "i")
  res
  boot.ci(res, conf = 0.95, type = "basic")

## End(Not run)
```

---

invlink   *Transformation from the Link Scale to the Observation Scale*

Description

The function transforms a variable from the link scale to the observation scale (probability or count).

Usage

```
invlink(x, type = c("cloglog", "log", "logit", "probit"))
```

Arguments

x   A vector of real numbers.

type   A character string: "cloglog", "log", "logit" or "probit".

Value

clog-log: $1 - exp(-exp(x))$
log: $exp(x)$
logit: $exp(x)/(1 + exp(x))$
probit: $pnorm(x)$

See Also

link

Examples

```
x <- seq(-5, 5, length = 100)
plot(x, invlink(x, type = "logit"), type = "l", lwd = 2, ylab = "Probability")
lines(x, invlink(x, type = "cloglog"), lty = 2, lwd = 2)
grid(col = "black")
legend(-5, 1, legend = c("alogit(x)", "acloglog(x)"), lty = c(1, 2), bg = "white")
```

---

link   *Transformation from the Observation Scale to the Link Scale*

Description

The function transforms a variable from the observation scale (probability or count) to the link scale.

Usage

```
link(x, type = c("cloglog", "log", "logit", "probit"))
```

Arguments

x   A vector of real numbers.

type   A character string: "cloglog", "log", "logit" or "probit".

Value

clog-log: $log(-log(1 - x))$
log: $log(x)$
logit: $log(x/(1 - x))$
probit: $qnorm(x)$

See Also

[invlink](#)

Examples

```
x <- seq(.001, .999, length = 100)
plot(x, link(x, type = "logit"), type = "l", lwd = 2, ylab = "link(proba.)")
lines(x, link(x, type = "cloglog"), lty = 2, lwd = 2)
grid(col = "black")
legend(0, 6, legend = c("logit(x)", "cloglog(x)"), lty = c(1, 2), bg = "white")
```

lizards	*A Comparison of Site Preferences of Two Species of Lizard*

Description

"These data describe the daytime habits of two species of lizards, *grahami* and *opalinus*. They were collected by observing occupied sites or perches and recording the appropriate description, namely species involved, time of the day, height and diameter of the perch and whether the site was sunny or shaded. Time of the day is recorded as early, mid-day or late." (McCullagh and Nelder, 1989, p.129).

Usage

```
data(lizards)
```

Format

A data frame with 24 observations on the following 6 variables.

Site A factor with levels Sun and Shade.

Diameter A factor with levels D <= 2 and D > 2 (inches).

Height A factor with levels H < 5 and H >= 5 (feet).

Time A factor with levels Early, Mid-day and Late.

grahami A numeric vector giving the observed sample size for *grahami* lizards.

opalinus A numeric vector giving the observed sample size for *opalinus* lizards.

Details

The data were originally published in Fienberg (1970).

Source

McCullagh, P., Nelder, J. A., 1989, 2nd ed. *Generalized linear models*. New York, USA: Chapman and Hall.

References

Fienberg, S.E., 1970. *The analysis of multidimensional contingency tables*. Ecology 51: 419-433.

---

mice   *Pregnant Female Mice Experiment*

Description

Unpublished laboratory data on the proportion of affected foetuses in two groups (control and treatment) of 10 pregnant female mice (Kupper and Haseman, 1978, p. 75).

Usage

```
data(mice)
```

Format

A data frame with 20 observations on the following 3 variables.

group a factor with levels CTRL and TREAT

n a numeric vector: the total number of foetuses.

m a numeric vector: the number of affected foetuses.

References

Kupper, L.L., Haseman, J.K., 1978. *The use of a correlated binomial model for the analysis of a certain toxicological experiments.* Biometrics 34, 69-76.

---

orob1   *Germination Data*

Description

Data describing the germination for seed *Orobanche cernua* cultivated in three dilutions of a bean root extract (Crowder, 1978, Table 1). The mean proportions of the three groups are 0.142, 0.872 and 0.842, and the overall mean is 0.614.

Usage

```
data(orob1)
```

Format

A data frame with 16 observations on the following 3 variables.

dilution a factor with 3 levels: 1/1, 1/25 and 1/625.

n a numeric vector: the number of seeds exposed to germination.

m a numeric vector: the number of seeds which actually germinated.

References

Crowder, M.J., 1978. *Beta-binomial anova for proportions*. Appl. Statist. 27, 34-37.

---

orob2	*Germination Data*

Description

A 2 x 2 factorial experiment comparing 2 types of seed and 2 root extracts (Crowder, 1978, Table 3). There are 5 or 6 replicates in each of the 4 treatment groups, and each replicate comprises a number of seeds varying between 4 and 81. The response variable is the proportion of seeds germinating in each replicate.

Usage

```
data(orob2)
```

Format

A data frame with 21 observations on the following 4 variables.

seed a factor with 2 levels: O73 and O75.

root a factor with 2 levels BEAN and CUCUMBER.

n a numeric vector: the number of seeds exposed to germination.

m a numeric vector: the number of seeds which actually germinated.

References

Crowder, M.J., 1978. *Beta-binomial anova for proportions*. Appl. Statist. 27, 34-37.

---

rabbits	*Rabbits Foetuses Survival Experiment*

Description

Experimental data for analyzing the effect of an increasing dose of a compound on the proportion of live foetuses affected (Paul, 1982, Table 1). Four treatment-groups were considered: control "C", low dose "L", medium dose "M" and high dose "H". The animal species used in the experiment was banded Dutch rabbit.

Usage

```
data(rabbits)
```

Format

A data frame with 84 observations on the following 3 variables.

group a factor with levels `C`, `H`, `L` and `M`

n a numeric vector: the total number of foetuses.

m a numeric vector: the number of affected foetuses.

References

Paul, S.R., 1982. *Analysis of proportions of affected foetuses in teratological experiments.* Biometrics 38, 361-370.

---

rats   *Rats Diet Experiment*

Description

"Weil (1970) in Table 1 gives the results from an experiment comprising two treatments. One group of 16 pregnant female rats was fed a control diet during pregnancy and lactation, the diet of a second group of 16 pregnant females was treated with a chemical. For each litter the number n of pups alive at 4 days and the number x of pups that survived the 21 day lactation period were recorded." (Williams, 1975, p. 951).

Usage

```
data(rats)
```

Format

A data frame with 32 observations on the following 3 variables.

group A factor with levels `CTRL` and `TREAT`

n A numeric vector: the number of pups alive at 4 days.

m A numeric vector: the number of pups that survived the 21 day lactation.

Source

Williams, D.A., 1975. *The analysis of binary responses from toxicological experiments involving reproduction and teratogenicity.* Biometrics 31, 949-952.

References

Weil, C.S., 1970. *Selection of the valid number of sampling units and a consideration of their combination in toxicological studies involving reproduction, teratogenesis or carcinogenesis.* Fd. Cosmet. Toxicol. 8, 177-182.

---

salmonella   *Salmonella Reverse Mutagenicity Assay*

Description

"Data for our third example were compiled by Margolin et al. (1981) from an Ames *Salmonella* reverse mutagenicity assay. Table 1 shows the number of revertant colonies observed on each of 3 replicate plates tested at each of 6 dose levels of quinoline." (Breslow, 1984, Table 1).

Usage

```
data(salmonella)
```

Format

A data frame with 18 observations on the following 2 variables.

dose a numeric vector: the dose level of quinoline (microgram per plate).

m a numeric vector: the number of revertant colonies of TA98 *Salmonella*.

Source

Breslow, N.E., 1984. *Extra-Poisson variation in log-linear models*. Applied Statistics 33(1), 38-44.

References

Margolin, B.H., Kaplan, N., Zeiger, E., 1981. *Statistical analysis of the Ames **Salmonella** / microsome test*. Proc. Natl Acad. Sci. USA 76, 3779-3783.

---

splitbin   *Split Grouped Data Into Individual Data*

Description

The function splits grouped binomial data and optional covariates to individual binary data. Two types of grouped data are managed by `splitbin`:

- Grouped data with weights;
- Grouped data of form `cbind(success, failure)`.

When weights, successes or failures involve non-integer numbers, these numbers are rounded (using `round()`) before splitting.

Usage

```
splitbin(formula, data, id = "idbin")
```

Arguments

`formula`	A formula. The left-hand side describes the grouped data. The right-hand side describes the covariates. See examples for syntax.
`data`	A data frame where all the variables described in `formula` are found.
`id`	An optional character string naming the identifier (= grouping factor). Default to "idbin".

Value

A data frame built according to the formula and function used in the call.

Examples

```
# grouped data with weights
z <- data.frame(
    m = c(0, 1, 0, 1),
    f1 = c("A", "A", "B", "B"),
    f2 = c("C", "D", "C", "D"),
    n = c(4, 2, 1, 3)
    )
z
splitbin(formula = n ~ f1, data = z)$tab
splitbin(formula = n ~ f1 + f2 + m , data = z)$tab

# grouped data of form "cbind(success, failure)"
z <- data.frame(
    m = c(4, 1),
    n = c(5, 3),
    f1 = c("A", "B"),
    f2 = c("C", "D")
    )
z
splitbin(formula = cbind(m, n - m) ~ 1, data = z)$tab
splitbin(formula = cbind(m, n - m) ~ f1 + f2, data = z)$tab
splitbin(formula = cbind(m, n - m) ~ f1 + f2, data = z)$tab
```

---

varbin	*Estimate of a Probability from Clustered Binomial Data*

Description

The function estimates a probability and its variance from clustered binomial data

$$\{(n_1, m_1), (n_2, m_2), ..., (n_N, m_N)\},$$

where n_i is the size of cluster i, m_i the number of "successes" (proportions are $y = m/n$), and N the number of clusters. Confidence intervals are calculated using a normal approximation, which might be inappropriate when the probability is close to 0 or 1.

Usage

```
    varbin(n, m, alpha = 0.05, R = 5000)

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

Arguments

n	A vector of the sizes of the clusters.
m	A vector of the numbers of successes (proportions are eqny = m / n).
alpha	The significance level for the confidence intervals. Default to 0.05, providing 95% CI's.
R	The number of bootstrap replicates to compute bootstrap mean and variance. Default to 5000.
x	An object of class "varbin".
...	Further arguments to be passed to "print".

Details

Five methods are used for the estimations. Let us consider N clusters of sizes $n_1, \ldots, n_N$ with observed count responses $m_1, \ldots, m_N$. We note $y_i = m_i/n_i (i = 1, \ldots, N)$ the observed proportions. The underlying assumption is that the probability, say mu, is homogeneous across the clusters.

Binomial method: the probability estimate and its variance are calculated by

$\mu = (sum_i(m_i))/(sum_i(n_i))$ (ratio estimate) and

$\mu * (1 - \mu)/(sum_i(n_i) - 1)$, respectively.

Ratio method: the probability μ is estimated as for the binomial method (ratio estimate). The one-stage cluster sampling formula is used to calculate the variance of μ (see Cochran, 1999, p. 32 and p. 66).

Arithmetic method: the probability is estimated by $\mu = sum_i(y_i)/N$. The variance of μ is estimated by $sum_i(y_i - \mu)^2/(N * (N - 1))$.

Jackknife method: the probability is estimated by μ defined by the arithmetic mean of the pseudovalues $y_{v,i}$. The variance is estimated by $sum_i(y_{v,i} - \mu)^2/(N * (N - 1))$ (Gladen, 1977, Paul, 1982).

Bootstrap method: R samples of clusters of size N are drawn with equal probability from the initial sample $(y_1, \ldots, y_N)$ (Efron and Tibshirani, 1993). The bootstrap estimate μ and its estimated variance are the arithmetic mean and the empirical variance (computed with denominator $R - 1$) of the R binomial ratio estimates, respectively.

Value

An object of class varbin, printed with print.varbin.

References

Cochran, W.G., 1999, 3th ed. *Sampling techniques*. Wiley, New York.

Efron, B., Tibshirani, R., 1993. *An introduction to the bootstrap*. Chapman and Hall, London.

Gladen, B., 1977. *The use of the jackknife to estimate proportions from toxicological data in the presence of litter effects*. JASA 74(366), 278-283.

Paul, S.R., 1982. *Analysis of proportions of affected foetuses in teratological experiments*. Biometrics 38, 361-370.

See Also

[boot](#)

Examples

```
data(rabbits)
z <- rabbits[rabbits$group == "M", ]
varbin(z$n, z$m)
by(rabbits,
list(group = rabbits$group),
  function(x) varbin(n = x$n, m = x$m, R = 1000))
```

---

wald.test   *Wald Test for Model Coefficients*

Description

The function returns a Wald chi-squared test or a F test for a vector of model coefficients (possibly of length one), given its variance-covariance matrix.

Usage

```
wald.test(b, varb, Terms = NULL, L = NULL, H0 = NULL,  df = NULL, verbose = FALSE, ...)

  ## S3 method for class 'wald.test'
print(x, ..., digits = max(3, getOption("digits") - 3))
```

Arguments

b	A vector of coefficients with their var-cov matrix varb. Coefficients b and var-cov matrix are usually extracted using appropriate coef and vcov functions.
varb	A var-cov matrix of coefficients b (see above).
Terms	An optional integer vector specifying which coefficients should be *jointly* tested, using a Wald chi-squared test or a F test. The elements of varb correspond to the columns or rows of the var-cov matrix given in varb. Default is NULL.
L	An optional matrix conformable to b, such as its product with b i.e., L %*% b gives the linear combinations of the coefficients to be tested. Default is NULL.

H0	A numeric vector giving the null hypothesis H_0 for the test. It must be as long as Terms or must have the same number of columns as L. Default to 0 for all the coefficients to be tested.
df	A numeric vector giving the degrees of freedom to be used in an F test, i.e. the degrees of freedom of the residuals of the model from which b and varb were fitted. Default to NULL, for no F test. See the section **Details** for more information.
verbose	A logical scalar controlling the amount of output information. The default is FALSE, providing minimum output.
x	An object of class "wald.test"
digits	A numeric scalar indicating the number of digits to be kept after the decimal place.
...	Additional arguments to print.

Details

The assumption is that the coefficients follow asymptotically a multivariate normal distribution with mean equal to the model coefficients b and variance equal to their var-cov matrix varb.

One (and only one) of Terms or L must be given. When L is given, it must have the same number of columns as the length of b, and the same number of rows as the number of linear combinations of coefficients.

When df is given, the chi-squared Wald statistic is divided by m, the number of linear combinations of coefficients to be tested (i.e., length(Terms) or nrow(L)). Under the null hypothesis H_0, this new statistic follows an $F(m, df)$ distribution.

Value

An object of class wald.test, printed with print.wald.test.

References

Diggle, P.J., Liang, K.-Y., Zeger, S.L., 1994. *Analysis of longitudinal data*. Oxford, Clarendon Press, 253 p.
Draper, N.R., Smith, H., 1998. *Applied Regression Analysis*. New York, John Wiley & Sons, Inc., 706 p.

Examples

```
data(orob2)
fm <- aodql(cbind(m, n - m) ~ seed * root, data = orob2, family = "qbin")
# Wald chi2 test for the effect of root
wald.test(b = coef(fm), varb = vcov(fm), Terms = 3:4)
L <- matrix(c(0, 0, 1, 0, 0, 0, 0, 1), nrow = 2, byrow = TRUE)
wald.test(b = coef(fm), varb = vcov(fm), L = L)
```

Index