Package ‘grouped’

February 20, 2015

Title Regression Analysis of Grouped and Coarse Data

Version 0.6-0

Date 2009-10-12

Author Dimitris Rizopoulos <`d.rizopoulos@erasmusmc.nl`> and Spyridoula Tsonaka <`s.tsonaka@lumc.nl`>

Maintainer Dimitris Rizopoulos <`d.rizopoulos@erasmusmc.nl`>

Description Regression models for grouped and coarse data, under the Coarsened At Random assumption.

Depends R(>= 2.0.0), MASS

LazyLoad yes

LazyData yes

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-10-12 10:23:08

NeedsCompilation no

R topics documented:

anova.grouped *Anova method for grouped objects*

Description

Performs a Likelihood Ratio Test between two nested grouped models.

Usage

```
## S3 method for class 'grouped'
anova(object, object2, ...)
```

Arguments

object	an object inheriting from class grouped, nested to object2.
object2	an object inheriting from class grouped.
...	additional arguments; currently none is used.

Value

a list of class aov.grouped with the following components:

name0	the name of the null model represented by object.
L0	the log-likelihood under object.
df0	the number of parameters in object.
AIC0	the AIC under object.
BIC0	the BIC under object.
name1	the name of the alternative model represented by object2.
L1	the log-likelihood under object2.
df1	the number of parameters in object2.
AIC1	the AIC under object2.
BIC1	the BIC under object2.
L01	the value of the likelihood ratio test statistic.
p.value	the p-value of the test.

Warning

The function does only partial checking whether the two models are nested; the user is responsible to supply nested models in order to perform a valid test.

Note

anova.grouped performs a likelihood ratio test between two nested models; for simple Wald tests for the estimated parameters use summary.grouped.

Author(s)

Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

See Also

grouped, summary.grouped

Examples

```

m1 <- grouped(cbind(lo, up) ~ treat, link = "logit", data = Sdata)
m2 <- grouped(cbind(lo, up) ~ treat * x, link = "logit", data = Sdata)
anova(m1, m2)

m1 <- grouped(equispaced(r, n) ~ x1, link = "logit", data = Seeds)
m2 <- grouped(equispaced(r, n) ~ x1 * x2, link = "logit", data = Seeds)
anova(m1, m2)
```

equispaced *Equispaced Coarsening Mechanism*

Description

Creates the lower and upper limits of the interval in which the true response lies for grouped data in $[0, 1]$.

Usage

```
equispaced(y, m)
```

Arguments

y	the score obtained or number of successes.
m	the maximum score or number of trials.

Details

After splitting the $[0, 1]$ interval in `m + 1` intervals of equal length, `equispaced` returns the limits of the intervals into which the rounded version, namely `y/m`, of the true response lies.

Value

a 2-dimensional matrix containing the lower and upper limits of the intervals.

Author(s)

Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

See Also

grouped

Examples

```
equispaced(Seeds$r, Seeds$n)
```

grouped *Regression for Grouped Data - Coarse Data*

Description

grouped is used to fit regression models for grouped or coarse data under the assumption that the data are Coarsened At Random.

Usage

```
grouped(formula, link = c("identity", "log", "logit"),
            distribution = c("normal", "t", "logistic"), data,
            subset, na.action, str.values, df = NULL, iter = 3, ...)
```

Arguments

formula — a two-sided formula describing the model structure. In the left-hand side, a two-column response matrix must be supplied, specifying the lower and upper limits (1st and 2nd column, respectively) of the interval in which the true response lies. They can be defined arbitrarily or you can use the functions equispaced and rounding.

link — the link function under which the underlying response variable follows the distribution given by the distribution argument. Available choices are "identity", "log" and "logit". See **Details** for more info.

distribution — the assumed distribution for the true latent response variable. Available choices are "normal", "t" and "logistic". See **Details** for more info.

data — an optional data.frame containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which grouped is called.

subset — an optional vector specifying a subset of observations to be used in the fitting process.

na.action — a function which indicates what should happen when the data contain NAs.

str.values — a numeric vector of starting values.

df — a scalar numeric value denoting the degrees of freedom when the underlying distribution for the response variable is assumed to be Student's-t.

iter — the number of extra times to call optim in case the first optimization has not converged.

... — additional arguments; currently none is used.

Details

Let $Z_i, i = 1, \ldots, n$ be a random sample from a response variable of interest. In many problems one can think of the sample space S_i of Z_i as being partitioned into a number of groups; one then observes not the exact value of Z_i but the group into which it falls. Data generated in this way are called grouped (Heitjan, 1989). The function `grouped` and this package are devoted in the analysis of such data in the case the data are Coarsened At Random (Heitjan and Rubin, 1991).

The framework we use assumes a latent variable Z_i which is coarsely measured and for which we only know Y_{li} and Y_{ui}, i.e., the interval in which Z_i lies. Given some covariates X_i, $Z_i | X_i$ may assume either a Normal, a Logistic or (generalized) Student's-t distribution. In addition three link functions are available for greater flexibility. In particular, the likelihood is of the following form

$$L(\beta, \sigma) = \prod_{i=1}^{n} F([y^*_{ui} - x_i^t \beta]/\sigma) - F([y^*_{li} - x_i^t \beta]/\sigma),$$

where $F(\cdot)$ denotes the cdf of the assumed distribution given by the argument `distribution` and $y^*_{li} = \phi(y_{li})$, where $\phi(\cdot)$ denotes the link function, and y^*_{ui} is defined analogously.

An interesting example of coarse data is the various quality of life indexes. The observed value of such indexes can be thought of as a rounded version of the *true* latent quality of life that the index attempts to capture. Applications of this approach can be found in Lesaffre et al. (2005) and Tsonaka et al. (2005). Various other examples of grouped and coarse data can be found in Heitjan (1989; 1993).

Value

an object of class `grouped` is a list with the following components:

`coefficients` the estimated coefficients, including the standard deviation σ.

`hessian` the approximate Hessian matrix at convergence returned by `optim`.

`fitted` the fitted values.

`details` a list with components: (i) `X` the design matrix, (ii) `y` the response data matrix, (iii) `convergence` the convergence identifier returned by `optim`, (iv) `logLik` the value of the log-likelihood at convergence, (v) `k` the number of outer iterations used, (vi) `n` the sample size, (vii) `df` the degrees of freedom; `NULL` except for the t distribution, (viii) `link` the link function used, (ix) `distribution` the distribution assumed for the true latent response variable and (x) `max.sc` the maximum absolute value of the score vector at convergence.

`call` the matched call.

Author(s)

Dimitris Rizopoulos `<d.rizopoulos@erasmusmc.nl>`

References

Heitjan, D. (1989) Inference from grouped continuous data: A review (with discussion). *Statistical Science*, **4**, 164–183.

Heitjan, D. (1993) Ignorability and coarse data: some biomedical examples. *Biometrics*, **49**, 1099–1109.

Heitjan, D. and Rubin, D. (1991) Ignorability and coarse data. *Annals of Statistics*, **19**, 2244–2253.

Lesaffre, E., Rizopoulos, D. and Tsonaka, S. (2007) The logistic-transform for bounded outcome scores. *Biostatistics*, **8**, 72–85.

Tsonaka, S., Rizopoulos, D. and Lesaffre, E. (2006) Power and sample size calculations for discrete bounded outcomes. *Statistics in Medicine*, **25**, 4241–4252.

See Also

anova.grouped, plot.grouped, residuals.grouped, summary.grouped, power.grouped

Examples

```
grouped(cbind(lo, up) ~ treat * x, link = "logit", data = Sdata)

grouped(equispaced(r, n) ~ x1 * x2, link = "logit", data = Seeds)

# See Figure 1 and Table 1 in Heitjan (1989)
y <- iris[iris$Species == "setosa", "Petal.Width"]
index <- cbind(seq(0.05, 0.55, 0.1), seq(0.15, 0.65, 0.1))
n <- length(y)
a <- b <- numeric(n)
for(i in 1:n){
    ind <- which(index[, 2] - y[i] > 0)[1]
    a[i] <- index[ind, 1]
    b[i] <- index[ind, 2]
}
summary(grouped(cbind(a, b) ~ 1))

# See Figure 1 and Table 1 in Heitjan (1989)
y <- iris[iris$Species == "setosa", "Petal.Length"]
index <- cbind(seq(0.95, 1.75, 0.2), seq(1.15, 1.95, 0.2))
n <- length(y)
a <- b <- numeric(n)
for(i in 1:n){
    ind <- which(index[, 2] - y[i] > 0)[1]
    a[i] <- index[ind, 1]
    b[i] <- index[ind, 2]
}
summary(grouped(cbind(a, b) ~ 1))
```

plot.grouped *Plot method for grouped objects*

Description

Produces the plot of residuals versus the fitted values for a fitted `grouped` model.

Usage

```
## S3 method for class 'grouped'
plot(x, B = 100, sub.caption = deparse(formula(x)), ...)
```

Arguments

`x` an object of class `grouped`.

`B` the number of multiple imputations used to estimate the residuals (see `residuals.grouped` for more info).

`sub.caption` a sub-title to be used in the plot.

`...` extra graphical parameters to be passed in `plot`.

Author(s)

Dimitris Rizopoulos <`d.rizopoulos@erasmusmc.nl`>

See Also

`grouped`

Examples

```

m <- grouped(cbind(lo, up) ~ treat * x, link = "logit", data = Sdata)
plot(m)
```

`power.grouped` *Power and sample size calculations for grouped data*

Description

Uses the method of Tsonaka, Rizopoulos and Lesaffre (2005) to estimate the power (or sample size to achieve desired power) of the Wald's test statistic for two-tailed two group comparisons in grouped data.

Usage

```
power.grouped(n = NULL, m, X = NULL, theta, sigma, sign.level = 0.05,
                type.power = c("conditional", "marginal"), MC.iter = 10,
                type.lik = c("original", "approximate"),
                gr.mech = c("rounding", "equispaced"), dist.t, dist.x,
                power. = NULL, limits = c(10, 3000))
```

Arguments

`n` Total number of observations. For sample size calculations it must be `NULL`.

`m` Maximum value of the outcome.

`X` The model design matrix which must be provided in the conditional power calculations and `NULL` in the marginal power calculations. Default is `NULL`.

`theta` A vector of the assumed regression coefficient values corresponding to the intercept, treatment indicator and additional (when needed) covariates.

`sigma` The residual standard deviation.

`sign.level` Significance level (Type I error probability). Default value fixed at 0.05.

`type.power` Type of power calculations.

`MC.iter` Number of Monte Carlo iterations for the marginal power calculations. Default value fixed at 10.

`type.lik` Type of the likelihood function to be used in the sample size calculations. For the power calculations always the original likelihood function is used.

`gr.mech` Grouping mechanism.

`dist.t` A `data.frame` with 1 row that gives for the treatment indicator the name of the assumed distribution and the assumed values of its parameters. This argument must always be provided for marginal power calculations and missing for conditional power calculations.

Currently available are two choices for this distribution: "bernoulli" and "no distr". For the "bernoulli", `n` values are simulated from the Bernoulli distribution using the function `rbinom` and thus the parameters that must be specified are `prob` and `size`. In the case of "no distr", only the sample size of the two treatment groups must be specified.

`dist.x` A `data.frame` with number of rows the number of the additional covariates (except from the treatment indicator). In each row it is given for each covariate the name of the assumed distribution and the assumed values of their parameters. When adjustment for additional covariates is not envisaged this argument must be missing.

Currently available are the following options for the covariate distribution: "normal", "gamma", "beta", "chisquare", "uniform" and "bernoulli".

`power.` Power of test (1 minus Type II error probability). For power calculations it must be `NULL`.

`limits` A vector giving the limits of the interval to be searched for the sample size. Default interval fixed at (10, 3000)

Details

`power.grouped` performs power or sample size calculations for bounded outcome scores under the model described in `grouped` using the Wald's test statistic. An important feature of this method is that it allows for covariate adjustments that can considerably increase the power.

Two types of the power function are considered: the conditional and the marginal (i.e., argument `type.power`). The conditional power function $p_c(X)$ assumes that the values of the covariates are known and can be used for post-hoc power analysis. In particular, it is assumed that the Wald's test follows a non central Student's-t under the alternative hypothesis with power function given by

$$p_c(X) = 1 - F_{\nu,\delta}(t_{\nu,1-\alpha/2}|H_a; X) + F_{\nu,\delta}(t_{\nu,\alpha/2}|H_a; X),$$

where $F_{\nu,\delta}$ is the distribution function of the non-central Student's-t distribution with ν degrees of freedom and non-centrality parameter δ, α is the type I error (i.e., argument `sign.level`) and X the realized values of the covariates. The marginal power function (mainly used for sample size calculations) acknowledges that prior to a study the actual values of the covariates are not known and is defined as the expected value of the conditional power

$$p_m = \int p_c(X) dH(X),$$

with respect to the assumed distribution of the covariates $H(X)$, based on pilot or historical data. This expectation is approximated using Monte Carlo integration.

In order to reduce the computational burden (induced by the Monte Carlo integration) for sample size calculation, an approximation to the likelihood is performed using a first order Taylor series expansion (i.e., argument `type.lik`). The approximate likelihood is suggested to be used for sample size calculations to get a better initial search area than the default (i.e., argument `limits`). Then the sample size calculations can be made using the original likelihood function. See **Examples** below.

Value

An object of class `"power.grouped"`, is a list of the arguments (including the computed one).

Note

`power.grouped` currently performs power or sample size calculations for the two-sided test.

`uniroot` is used to solve power equation for unknowns, so you may see errors from it, notably about inability to bracket the root when invalid arguments are given.

Author(s)

Spyridoula Tsonaka `<spyridoula.tsonaka@med.kuleuven.be>`

References

Tsonaka, S., Rizopoulos, D. and Lesaffre, E. (2005) Power and sample size calculations for discrete bounded outcomes. *submitted for publication*.

See Also

`grouped`, `rounding`, `equispaced`, `uniroot`

Examples

```
## Not run:
power.grouped(n = NULL, X = NULL, m = 20, theta = c(0, 1, 0.7),
    sigma = 1, type.power = "marginal", type.lik = "approximate",
    gr.mech = "equispaced", dist.t = data.frame("bernoulli", 0.5, 1),
    dist.x = data.frame("normal", 0, 1), power. = 0.7, limits = c(10,1000))
    # to get an initial search area using the approximate likelihood

power.grouped(n = NULL, X = NULL, m = 20, theta = c(0, 1, 0.7),
    sigma = 1, type.power = "marginal", MC.iter = 20, gr.mech = "equispaced",
    dist.t = data.frame("bernoulli", 0.5, 1), dist.x = data.frame("normal", 0, 1),
    power. = 0.7, limits = c(10,50))
    # redefine the search area and use the original likelihood

## End(Not run)

power.grouped(n = 20, X = NULL, m = 20, theta = c(0, 1, 0.7),
    sigma = 1, type.power = "marginal", gr.mech = "equispaced",
    dist.t = data.frame("bernoulli", 0.5, 1),
    dist.x = data.frame("normal", 0, 1), power. = NULL)
```

`residuals.grouped` *Residuals for grouped objects*

Description

Computes a version of Bayesian latent residuals for grouped models.

Usage

```
## S3 method for class 'grouped'
residuals(object, standardized = FALSE, B = 100, ...)
```

Arguments

`object`	an object of class grouped.
`standardized`	logical; if TRUE the standardized residuals are computed.
`B`	the number of multiple imputations to be used to estimate the residuals.
`...`	additional parameters; currently none is used.

Details

In a grouped-data setting the ordinary definition of residuals is problematic since, in fact the value of the true response is known only up to the interval in which it lies. A possible solution to this problem provides the notion of Bayesian residuals (see e.g., Johnson and Albert, Section 3.4). In particular, the Bayesian residuals in the grouped-data setting are defined as follows:

$$r_i = Z_i - x_i^t \beta,$$

where Z_i denotes the value of the underlying true response of the ith sample unit, x_i^t is the covariate vector of the ith sample unit, β are the regression coefficients and let also Y_i denote the observed data.

An estimation for r_i can be obtained under the following Multiple Imputation (MI) scheme:

Step 1: Simulate new parameter values, say θ^*, from $N(\hat{\theta}, C(\hat{\theta}))$, where $\hat{\theta}$ are the MLEs (including both β and σ, see `grouped`) and $C(\hat{\theta})$ is their large sample covariance matrix.

Step 2: Draw a value, say z_i^*, from the predictive distribution $Z_i|Y_i$ under θ^* and compute the residuals $r_i^* = z_i^* - x_i^t \beta^*$. In fact, $p(z_i|y_i; \theta^* = (\beta^*, \sigma^*))$ is a truncated F distribution in the interval given by y_i, where F denotes the distribution implied by the value of the `distribution` argument used in `grouped`.

Step 3: Repeat steps 1-2 `B` times and combine the estimates using the known formulas of MI.

This procedure explicitly acknowledges the ignorance of the true parameter values by drawing from their large sample posterior distribution while taking into account the sampling error.

Value

an object of class `resid.grouped` with the following components:

`residuals` a vector of the estimated residuals.

`mat.res` a numeric matrix containing the B realization of the latent residuals. If `standardized = TRUE`, then `mat.res` contains the values of r_i^* / σ^*. See **Details** above.

`nam.res` a character vector specifying the sample units names.

`B` the value of the `B` argument defined above.

`standardized` the value of the `standardized` argument defined above.

`fitted` a numeric vector of the fitted values of `object`.

Author(s)

Dimitris Rizopoulos <`d.rizopoulos@erasmusmc.nl`>

References

Johnson, V. and Albert, J. (1999), *Ordinal Data Modeling*, New York: Springer-Verlag.

See Also

`grouped`, `summary.resid.grouped`

Examples

```
m1 <- grouped(cbind(lo, up) ~ treat * x, link = "logit", data = Sdata)
resid(m1)

m2 <- grouped(equispaced(r, n) ~ x1 * x2, link = "logit", data = Seeds)
resid(m2)
```

`rounding` *Rounding Coarsening Mechanism*

Description

Creates the lower and upper limits of the interval in which the true response lies for grouped data in $[0, 1]$.

Usage

```
rounding(y, m)
```

Arguments

y	the score obtained or number of successes.
m	the maximum score or number of trials.

Details

Under the rounding coarsening mechanism, we assume that the true response lies in the interval `[y/m - 0.5/(m + 1), y/m + 0.5/(m + 1)]`.

Value

a 2-dimensional matrix containing the lower and upper limits of the intervals.

Author(s)

Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

See Also

`grouped`

Examples

```
rounding(Seeds$r, Seeds$n)
```

Sdata *Simulated Data*

Description

A simulated data-set used for the illustration of grouped for grouped data coming from a logit-normal distribution.

Format

A data frame with 250 observations on the following 4 variables:

`lo` the lower limits of the response intervals.

`up` the upper limits of the response intervals.

`treat` the treatment indicator.

`x` a continuous covariate.

Details

The data set has been produced with the code in the **Examples** below.

Author(s)

Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

Examples

```
## Not run:
n <- 250
treat <- rbinom(n, 1, 0.5)
x <- runif(n, -4, 4)
mu <- 1 + 0.5 * treat -1 * x + 0.8 * treat * x
u <- plogis(rnorm(n, mu, 2))

index <- cbind(c(0, 0.25, 0.5, 0.75), c(0.25, 0.5, 0.75, 1))
a <- b <- numeric(n)
for(i in 1:n){
    ind <- which(index[, 2] - u[i] > 0)[1]
    a[i] <- index[ind, 1]
    b[i] <- index[ind, 2]
}
Sdata <- data.frame(lo = a, up = b, treat = factor(treat), x)

## End(Not run)

str(Sdata)
summary(Sdata)
```

Seeds *Seeds Data*

Description

This example is taken from Table 3 of Crowder (1978), and concerns the proportion of seeds that germinated on each of 21 plates arranged according to a 2 by 2 factorial layout by seed and type of root extract.

Format

A data frame with 21 observations (denoting plates) on the following 4 variables.

r the number of germinated seeds.

n the number of total seeds.

x1 seed type.

x2 root type.

Author(s)

Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

Source

Crowder, M. (1978) Beta-Binomial ANOVA for proportions. *Applied Statistics*, 35, 34–37.

Examples

```
str(Seeds)
summary(Seeds)
```

summary.grouped *Summary method for grouped objects*

Description

Summarizes the fit of grouped objects.

Usage

```
## S3 method for class 'grouped'
summary(object, ...)
```

Arguments

`object` an object of class `grouped`.

`...` additional parameters; currently none is used.

Details

`summary.grouped` provides summaries of the fit for `grouped` objects, including computation of Wald tests for the estimated parameters.

Value

a list of class `summ.grouped` with the following components:

`object` the fitted `object`.

`coefficients` a numeric matrix containing the estimated coefficients, their standard errors, t-values and p-values.

`sigma` the estimated standard deviation of the underlying latent variable.

`se.sigma` the estimated standard error for the estimation of `sigma`.

`logLik` the value of the log-likelihood under the estimated parameters.

`AIC` the AIC under the fitted model.

`BIC` the BIC under the fitted model.

Author(s)

Dimitris Rizopoulos <d.rizopoulos@erasmusmc.nl>

See Also

`grouped`

Examples

```
m1 <- grouped(cbind(lo, up) ~ treat * x, link = "logit", data = Sdata)
summary(m1)

m2 <- grouped(equispaced(r, n) ~ x1 * x2, link = "logit", data = Seeds)
summary(m2)
```

summary.resid.grouped *Summary method for resid.grouped objects*

Description

The main use of this function is for identification of outliers.

Usage

```
## S3 method for class 'resid.grouped'
summary(object, K = 2, observed = NULL, ...)
```

Arguments

`object`	an object of class `resid.grouped`.
`K`	the cutoff point to identify outliers
`observed`	a numeric vector of possible observed data, e.g., the mean of the interval in which the true data lie.
`...`	additional arguments; currently none is used.

Details

Taking into advantage the realizations of the standardized residuals r_i provided by the Multiple Imputation scheme, we can estimate the probability

$$Pr(|r_i| > K), i = 1, \ldots, n,$$

which can be regarded as the probability of the ith sample unit being an outlier.

Value

a numeric matrix with columns, the fitted values, the estimated residuals, and the percentage of each sample unit having an absolute residual greater than the value given by `K`. If `!is.null(observed)` its value is given as the first column of the matrix.

Author(s)

Dimitris Rizopoulos <`d.rizopoulos@erasmusmc.nl`>

See Also

`residuals.grouped`

Examples

```
m <- grouped(cbind(lo, up) ~ treat * x, link = "logit", data = Sdata)
summary(resid(m, TRUE))
```

Index