## Package ‘geepack’

**Version** 1.2-1  
**Title** Generalized Estimating Equation Package  
**Author** Søren Højsgaard <sorenh@math.aau.dk>, Ulrich Halekoh <uhalekoh@health.sdu.dk>, Jun Yan <jun.yan@uconn.edu>,  
**Maintainer** Søren Højsgaard <sorenh@math.aau.dk>  
**Description** Generalized estimating equations solver for parameters in  
mean, scale, and correlation structures, through mean link,  
scale link, and correlation link. Can also handle clustered  
categorical responses.  
**Encoding** latin1  
**License** GPL (>= 3)  
**NeedsCompilation** yes  
**Repository** CRAN  
**Date/Publication** 2016-09-24 14:55:58

### R topics documented:

- compCoef .................................................. 2  
- dietox .................................................. 3  
- fixed2Zcor .............................................. 4  
- geeglm .................................................. 5  
- geese ................................................... 8  
- geese.control .......................................... 12  
- genZcor ................................................. 13  
- koch ................................................... 14  
- ohio .................................................... 15  
- ordgee .................................................. 15  
- relRisk .................................................. 18  
- respdis .................................................. 19  
- respiratory ............................................. 20  
- seizure ................................................. 21  
- sitka89 .................................................. 22  
- spruce ................................................... 23
compCoef

Description

Comparing regression coefficients between models when one model is nested within another for clustered data.

Usage

compCoef(fit0, fit1)

Arguments

fit0 a fitted object of class geese
fit1 another fitted object of class geese

Value

a list of two components:

delta estimated difference in the coefficients of common covariates from fit0 and fit1
variance estimated variance matrix of delta

Author(s)

Jun Yan <jyan.stat@gmail.com>

References


Examples

```r
## generate clustered data
gendat <- function(ncl, clsz) {
## ncl: number of clusters
## clsz: cluster size (all equal)
id <- rep(1:ncl, each = clsz)
visit <- rep(1:clsz, ncl)
n <- ncl * clsz
x1 <- rbinom(n, 1, 0.5)  ## within cluster varying binary covariate
x2 <- runif(n, 0, 1)  ## within cluster varying continuous covariate
## the true correlation coefficient rho for an ar(1)
## correlation structure is 2/3
rho <- 2/3
rhomat <- rho ^ outer(1:4, 1:4, function(x, y) abs(x - y))
chol.u <- chol(rhomat)
noise <- as.vector(sapply(1:ncl, function(x) chol.u %*% rnorm(clsz)))
y <- 1 + 3 * x1 - 2 * x2 + noise
dat <- data.frame(y, id, visit, x1, x2)
dat
}
simdat <- gendat(100, 4)
fit0 <- geese(y ~ x1, id = id, data = simdat, corstr = "un")
fit1 <- geese(y ~ x1 + x2, id = id, data = simdat, corstr = "un")
compCoef(fit0, fit1)
```

Description

The `dietox` data frame has 861 rows and 7 columns.

Usage

data(dietox)

Format

This data frame contains the following columns:

- **Weight** a numeric vector
- **Feed** a numeric vector
- **Time** a numeric vector
- **Pig** a numeric vector
- **Evit** a numeric vector
- **Cu** a numeric vector
- **Litter** a numeric vector
fixedZcor

Source

Examples
data(dietox)
dietox$Cu <- as.factor(dietox$Cu)
gee01 <- geeglm(Weight ~ Time + Cu + Cu * Time, id = Pig, data = dietox, family = gaussian, corstr = "ex")

mf <- formula(Weight ~ Cu * (Time + I(Time^2) + I(Time^3)))
gee1 <- geeglm(mf, data = dietox, id = Pig, family = poisson("identity"), corstr = "ar1")
summary(gee1)
anova(gee1)

fixed2Zcor

Construct zcor vector (of fixed correlations) from a fixed working correlation matrix

Description
Construct zcor vector (of fixed correlations) from a fixed working correlation matrix, a specification of clusters and a specification of waves.

Usage
fixed2Zcor(cor.fixed, id, waves)

Arguments
cor.fixed Matrix
id Clusters
waves Vector giving the ordering of observations within clusters.

Value
A vector which can be passed as the zcor argument to geeglm.

Author(s)
Søren Højsgaard <sorenh@agrsci.dk>

See Also
genZcor, geeglm
Examples

timeorder <- rep(1:5, 6)  
tvar <- timeorder + rnorm(length(timeorder))  
idvar <- rep(1:6, each=5)  
uuu <- rep(rnorm(6), each=5)  
yvar <- 1 + 2*tvar + uuu + rnorm(length(tvar))  
simdat <- data.frame(idvar, timeorder, tvar, yvar)  
head(simdat, 12)

simdatPerm <- simdat[sample(nrow(simdat))]

head(simdatPerm)

cor.fixed <- matrix(c(1, 0.5, 0.25, 0.125, 0.125,  
                      0.5, 1, 0.25, 0.125, 0.125, 0.125,  
                      0.25, 0.25, 1, 0.5, 0.125, 0.125,  
                      0.125, 0.125, 0.5, 1, 0.125, 0.125,  
                      0.125, 0.125, 0.125, 0.125, 1), nrow=5, ncol=5)

cor.fixed

cor.fixed <- matrix(c(1, 0.5, 0.25, 0.125, 0.125,  
                      0.5, 1, 0.25, 0.125, 0.125, 0.125,  
                      0.25, 0.25, 1, 0.5, 0.125, 0.125,  
                      0.125, 0.125, 0.5, 1, 0.125, 0.125,  
                      0.125, 0.125, 0.125, 0.125, 1), nrow=5, ncol=5)

cor.fixed

zcor <- fixed2Zcor(cor.fixed, id=simdatPerm$idvar, waves=simdatPerm$timeorder)

mod4 <- geeglm(yvar ~ tvar, id=idvar, data=simdatPerm, corstr="fixed", zcor=zcor)

mod4

---

geeglm

*Fit Generalized Estimating Equations (GEE)*

**Description**

The geeglm function fits generalized estimating equations using the 'geese.fit' function of the 'geepack' package for doing the actual computations. geeglm has a syntax similar to glm and returns an object similar to a glm object. An important feature of geeglm, is that an anova method exists for these models.

**Usage**

geeglm(formula, family = gaussian, data = parent.frame(), weights, subset,  
na.action, start = NULL, etastart, mustart, offset,  
control = geese.control(...),  
method = "glm.fit", x = FALSE, y = TRUE,  
contrasts = NULL,  
id, waves=NULL, zcor=NULL,  
corstr = "independence",  
scale.fix = FALSE,
scale.value =1, std.err="san.se",
...

Arguments

formula See corresponding documentation to glm
family See corresponding documentation to glm
data See corresponding documentation to glm
weights See corresponding documentation to glm
subset See corresponding documentation to glm
na.action No action is taken. Indeed geeglm only works on complete data.
start See corresponding documentation to glm
etastart See corresponding documentation to glm
mustart See corresponding documentation to glm
offset See corresponding documentation to glm
control See corresponding documentation to glm
method See corresponding documentation to glm
x See corresponding documentation to glm
y See corresponding documentation to glm
contrasts See corresponding documentation to glm
id a vector which identifies the clusters. The length of ‘id’ should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.
waves Variable specifying the ordering of repeated mesurements on the same unit. Also used in connection with missing values. See examples below.
zcor Used for entering a user defined working correlation structure.
corstr a character string specifying the correlation structure. The following are permitted: "independence", "exchangeable", "ar1", "unstructured" and "userdefined"
scale.fix a logical variable; if true, the scale parameter is fixed at the value of ‘scale.value’.
scale.value numeric variable giving the value to which the scale parameter should be fixed; used only if ‘scale.fix == TRUE’.
std.err Type of standard error to be calculated. Default ’san.se’ is the usual robust estimate. Other options are ’jack’: if approximate jackknife variance estimate should be computed. ’j1s’: if 1-step jackknife variance estimate should be computed. ’fij’: logical indicating if fully iterated jackknife variance estimate should be computed.

... further arguments passed to or from other methods.

Details

In the case of corstr="fixed" one must provide the zcor vector if the clusters have unequal sizes. Clusters with size one must not be represented in zcor.
Value

An object of type 'geeglm'

Warning

Use "unstructured" correlation structure only with great care. (It may cause R to crash).

Note

See the documentation for the 'geese' function for additional information. geeglm only works for complete data. Thus if there are NA's in data you can specify data=na.omit(mydata).

Author(s)

Søren Højsgaard, sorenh@agrsci.dk

References


See Also

gerese, glm, anova.geeglm

Examples

data(dietox)
dietox$Cu <- as.factor(dietox$Cu)
mf <- formula(Weight~Cu*(Time*I(Time^2)+I(Time^3)))
gee1 <- geeglm(mf, data=dietox, id=Pig, family=poisson("identity"),corstr="ar1")
summary(gee1)

mf2 <- formula(Weight~Cu*Time+I(Time^2)+I(Time^3))
gee2 <- geeglm(mf2, data=dietox, id=Pig, family=poisson("identity"),corstr="ar1")
anova(gee2)
geese

Function to solve a Generalized Estimating Equation Model

Description

Produces an object of class ‘geese’ which is a Generalized Estimating Equation fit of the data.

Usage

geese(formula = formula(data), sformula = ~1, id, waves = NULL,
data = parent.frame(), subset = NULL, na.action = na.omit,
contrasts = NULL, weights = NULL, zcor = NULL, corp = NULL,
control = geese.control(...), b = NULL, alpha = NULL, gm = NULL,
family = gaussian(), mean.link = NULL, variance = NULL,
cor.link = "identity", sca.link = "identity", link.same = TRUE,
scale.fix = FALSE, scale.value = 1, corstr = "independence", ...)

geese.fit(x, y, id, offset = rep(0, N), offset = rep(0, N),
weights = rep(1,N), waves = NULL, zscA = matrix(1, N, 1),
zcor = NULL, corp = NULL, control = geese.control(...),
b = NULL, alpha = NULL, gm = NULL, family = gaussian(),
mean.link = NULL, variance = NULL, cor.link = "identity",
sca.link = "identity", link.same = TRUE, scale.fit = FALSE,
scale.value = 1, corstr = "independence", ...)

Arguments

formula a formula expression as for glm, of the form response ~ predictors. See the documentation of lm and formula for details. As for glm, this specifies the linear predictor for modeling the mean. A term of the form offset(expression) is allowed.
sformula a formula expression of the form ~ predictor, the response being ignored. This specifies the linear predictor for modeling the dispersion. A term of the form offset(expression) is allowed.
id a vector which identifies the clusters. The length of ‘id’ should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.
waves an integer vector which identifies components in clusters. The length of waves should be the same as the number of observation. components with the same waves value will have the same link functions.
data an optional data frame in which to interpret the variables occurring in the formula, along with the id and n variables.
subset expression saying which subset of the rows of the data should be used in the fit. This can be a logical vector (which is replicated to have length equal to the number of observations), or a numeric vector indicating which observation numbers
are to be included, or a character vector of the row names to be included. All
observations are included by default.

**na.action**
a function to filter missing data. For gee only na.omit should be used here.

**contrasts**
a list giving contrasts for some or all of the factors appearing in the model
formula. The elements of the list should have the same name as the variable
and should be either a contrast matrix (specifically, any full-rank matrix with as
many rows as there are levels in the factor), or else a function to compute such
a matrix given the number of levels.

**weights**
an optional vector of weights to be used in the fitting process. The length of
weights should be the same as the number of observations. This weights is not
(yet) the weight as in sas proc genmod, and hence is not recommended to use.

**zcor**
a design matrix for correlation parameters.

**corp**
known parameters such as coordinates used for correlation coefficients.

**control**
a list of iteration and algorithmic constants. See geese.control for their names
and default values. These can also be set as arguments to geese itself.

**b**
an initial estimate for the mean parameters.

**alpha**
an initial estimate for the correlation parameters.

**gm**
an initial estimate for the scale parameters.

**family**
a description of the error distribution and link function to be used in the model,
as for glm.

**mean.link**
a character string specifying the link function for the means. The following are
allowed: "identity", "logit", "probit", "cloglog", "log", and "inverse". The
default value is determined from family.

**variance**
a character string specifying the variance function in terms of the mean. The
following are allowed: "gaussian", "binomial", "poisson", and "gamma". The
default value is determined from family.

**cor.link**
a character string specifying the link function for the correlation coefficients.
The following are allowed: "identity", and "fisherz".

**sca.link**
a character string specifying the link function for the scales. The following are
allowed: "identity", and "log".

**link.same**
a logical indicating if all the components in a cluster should use the same link.

**scale.fix**
a logical variable; if true, the scale parameter is fixed at the value of scale.value.

**scale.value**
numeric variable giving the value to which the scale parameter should be fixed;
used only if scale.fix == TRUE.

**corstr**
a character string specifying the correlation structure. The following are permitted:
"independence", "exchangeable", "ar1", "unstructured", "userdefined", and "fixed"

**x, y**
x is a design matrix of dimension n * p, and y is a vector of observations of
length n.

**offset, offset**
vector of offset for the mean and for the scale, respectively.

**zsca**
a design matrix of dimension n * r for the scales.

**...**
further arguments passed to or from other methods.
Details

when the correlation structure is fixed, the specification of Zcor should be a vector of length
\[ \text{sum(clusz} \times (\text{clusz} - 1)) / 2. \]

Value

An object of class "geese" representing the fit.

Author(s)

Jun Yan <jyan.stat@gmail.com>

References


See Also

glm, lm, ordgee.

Examples

data(seizure)
## Diggle, Liang, and Zeger (1994) pp166-168, compare Table 8.10
seiz.l <- reshape(seizure,
    varying=list(c("base","y1", "y2", "y3", "y4")),
    v.names="y", times=0:4, direction="long")
seiz.l$< <- order(seiz.l$id, seiz.l$time),]
seiz.l$t <- ifelse(seiz.l$time == 0, 8, 2)
seiz.l$x <- ifelse(seiz.l$time == 0, 0, 1)
m1 <- geese(y ~ offset(log(t)) + x + trt + x:trt, id = id,
    data=seiz.l, corstr="exch", family=poisson)
summary(m1)
m2 <- geese(y ~ offset(log(t)) + x + trt + x:trt, id = id,
    data = seiz.l, subset = id!=49,
    corstr = "exch", family=poisson)
summary(m2)
## Using fixed correlation matrix
cor.fixed <- matrix(c(1, 0.5, 0.25, 0.125, 0.125, 0.5, 1, 0.25, 0.125, 0.125, 0.25, 1, 0.5, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 1), 5, 5)
cor.fixed
zcor <- rep(cor.fixed[lower.tri(cor.fixed)], 59)
m3 <- geese(y ~ offset(log(t)) + x + trt + x:trt, id = id,
    data = seiz.l, family = poisson,
    corstr = "fixed", zcor = zcor)
summary(m3)
```r
data(o)hio
fit <- geese(resp ~ age + smoke + age:smoke, id = id, data = o, 
  family = binomial, corstr = "exch", scale.fix = TRUE)
summary(fit)
fit.ar1 <- geese(resp ~ age + smoke + age:smoke, id = id, data = o, 
  family = binomial, corstr = "ar1", scale.fix = TRUE)
summary(fit.ar1)

## simulated data
## a function to generate a dataset
gendat <- function() {
  id <- gl(50, 4, 200)
  visit <- rep(1:4, 50)
  x1 <- rbinom(200, 1, 0.6) ## within cluster varying binary covariate
  x2 <- runif(200, 0, 1) ## within cluster varying continuous covariate
  phi <- 1 + 2 * x1 ## true scale model
  ## the true correlation coefficient rho for an ar(1)
  ## correlation structure is 0.667.
  rhomat <- 0.667 ^ outer(1:4, 1:4, function(x, y) abs(x - y))
  chol.u <- chol(rhomat)
  noise <- as.vector(sapply(1:50, function(x) chol.u %*% rnorm(4)))
  e <- sqrt(phi) * noise
  y <- 1 + 3 * x1 - 2 * x2 + e
  dat <- data.frame(y, id, visit, x1, x2)
  dat
}
dat <- gendat()
fit <- geese(y ~ x1 + x2, id = id, data = dat, sformula = ~ x1, 
  corstr = "ar1", jack = TRUE, j1s = TRUE, fij = TRUE)
summary(fit)

## create user-defined design matrix of unstructured correlation.
## in this case, zcor has 4*3/2 = 6 columns, and 50 * 6 = 300 rows
zcor <- genZcor(clus = rep(4, 50), waves = dat$visit, "unstr")
zfit <- geese(y ~ x1 + x2, id = id, data = dat, sformula = ~ x1, 
  corstr = "userdefined", zcor = zcor, 
  jack = TRUE, j1s = TRUE, fij = TRUE)
summary(zfit)

## Now, suppose that we want the correlation of 1-2, 2-3, and 3-4
## to be the same. Then zcor should have 4 columns.
zz <- matrix(NA, 300, 4)
zz[,2:4] <- zcor[, c(2, 3, 5)]
summary(geese(y ~ x1 + x2, id = id, data = dat, sformula = ~ x1, 
  corstr = "userdefined", zcor = zz, 
  jack = TRUE, j1s = TRUE, fij = TRUE))

## Next, we introduce non-constant cluster sizes by
## randomly selecting 60 percent of the data
good <- sort(sample(1:nrow(dat), .6 * nrow(dat)))
```
geese.control

Auxiliary for Controlling GEE Fitting

Description

Auxiliary function as user interface for 'gee' fitting. Only used when calling 'geese' or 'geese.fit'.

Usage

geese.control(epsilon = 1e-04, maxit = 25, trace = FALSE,
                           scale.fix = FALSE, jack = FALSE, j1s = FALSE, fij = FALSE)

Arguments

text{positive convergence tolerance epsilon; the iterations converge when the absolute value of the difference in parameter estimate is below epsilon.

maxit
logical indicating if output should be produced for each iteration.

trace
logical indicating if approximate jackknife variance estimate should be computed.

scale.fix
logical indicating if the scale should be fixed.

jack
logical indicating if 1-step jackknife variance estimate should be computed.

j1s
logical indicating if fully iterated jackknife variance estimate should be computed.

Details

When 'trace' is true, output for each iteration is printed to the screen by the c++ code. Hence, 'options(digits = *)' does not control the precision.

Value

A list with the arguments as components.

Author(s)

Jun Yan <jyan.stat@gmail.com>

See Also

'geese.fit', the fitting procedure used by 'geese'.

mdata <- dat[good,]

summary(geese(y ~ x1 + x2, id = id, data = mdata, waves = visit,
            sformula = ~ x1, corstr="ar1",
            jack = TRUE, j1s = TRUE, fij = TRUE))
Description

constructs the design matrix for the correlation structures: independence, exchangeable, \( \text{ar1} \) and unstructured. The user will need this function only as a basis to construct a user defined correlation structure: use \text{genZcor} to get the design matrix \( Z \) for the unstructured correlation and define the specific correlation structure by linear combinations of the columns of \( Z \).

Usage

\text{genZcor}(clusz, \ waves, \ corstrv)

Arguments

\begin{itemize}
\item \textbf{clusz} integer vector giving the number of observations in each cluster
\item \textbf{waves} integer vector, observations in the same cluster with values of wave \( i \) and \( j \) have the correlation \( \text{latex} \)
\item \textbf{corstrv} correlation structures: 1=indepednce, 2=exchangeable, 3=ar1, 4=unstructured
\end{itemize}

Value

the design matrix for the correlation structure

Author(s)

Jun Yan <jyan.stat@gmail.com>

See Also

\text{fixed2Zcor}

Examples

\begin{verbatim}
# example to construct a Toeplitz correlation structure
# \( \sigma_{ij} = \sigma_{|i-j|} \)

data set with 5 clusters and maximally 4 observations (visits) per cluster
\text{gendat <- function()} {  
  id <- gl(5, 4, 20)
  visit <- rep(1:4, 5)
  y <- rnorm(id)
  dat <- data.frame(y, id, visit)[c(-2,-9),]
}

set.seed(88)
dat<-gendat()
\end{verbatim}
# generating the design matrix for the unstructured correlation
zcor <- genZcor(clusz = table(dat$id), waves = dat$visit, corstrv=4)
# defining the Toeplitz structure
zcor.toep<-matrix(NA, nrow(zcor),3)
zcor.toep[,1]<-apply(zcor[,c(1,4,6)],1,sum)
zcor.toep[,2]<-apply(zcor[,c(2,5)],1,sum)
zcor.toep[,3]<-zcor[,3]

zfit1 <- geese(y ~ 1, id = id, data = dat,
              corstr = "userdefined", zcor = zcor.toep)

zfit2 <- geeglm(y ~ 1, id = id, data = dat,
              corstr = "userdefined", zcor = zcor.toep)

---

**koch**

*Ordinal Data from Koch*

**Description**

The *koch* data frame has 288 rows and 4 columns.

**Usage**

data(koch)

**Format**

This data frame contains the following columns:

- **trt** a numeric vector
- **day** a numeric vector
- **y** an ordered factor with levels: `1 < 2 < 3`
- **id** a numeric vector

**Examples**

data(koch)

```r
fit <- ordgee(ordered(y) ~ trt + as.factor(day), id=id, data=koch, corstr="exch")
summary(fit)
```
Ohio Children Wheeze Status

Description

The ohio data frame has 2148 rows and 4 columns. The dataset is a subset of the six-city study, a longitudinal study of the health effects of air pollution.

Usage

```r
data(ohio)
```

Format

This data frame contains the following columns:

- `resp` an indicator of wheeze status (1=yes, 0=no)
- `id` a numeric vector for subject id
- `age` a numeric vector of age, 0 is 9 years old
- `smoke` an indicator of maternal smoking at the first year of the study

References


Examples

```r
data(ohio)
fit <- geese(resp ~ age + smoke + age:smoke, id=id, data=ohio, family=binomial, corstr="exch", scale.fix=TRUE)
summary(fit)
fit.ar1 <- geese(resp ~ age + smoke + age:smoke, id=id, data=ohio, family=binomial, corstr="ar1", scale.fix=TRUE)
summary(fit.ar1)
```

ordgee

GEE for Clustered Ordinal Responses

Description

Produces an object of class ‘geese’ which is a Generalized Estimating Equation fit of the clustered ordinal data.
Usage

```r
ordgee(formula = formula(data), ooffset = NULL, id, waves = NULL, data = parent.frame, subset = NULL, na.action = na.omit, contrasts = NULL, weights = NULL, z = NULL, mean.link = "logit", corstr = "independence", control = geese.control(...), b = NA, alpha = NA, scale.fix = TRUE, scale.val = 1, int.const = TRUE, rev = FALSE,...)
```

Arguments

- **formula**: a formula expression as for `glm`, of the form `response ~ predictors`. See the documentation of `lm` and `formula` for details. As for `glm`, this specifies the linear predictor for modelling the mean. A term of the form `offset(expression)` is allowed.
- **ooffset**: vector of offset for the odds ratio model.
- **id**: a vector which identifies the clusters. The length of `id` should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.
- **waves**: an integer vector which identifies components in clusters. The length of `waves` should be the same as the number of observation. components with the same `waves` value will have the same link functions.
- **data**: an optional data frame in which to interpret the variables occurring in the formula, along with the `id` and `n` variables.
- **subset**: expression saying which subset of the rows of the data should be used in the fit. This can be a logical vector (which is replicated to have length equal to the number of observations), or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.
- **na.action**: a function to filter missing data. For `gee` only `na.omit` should be used here.
- **contrasts**: a list giving contrasts for some or all of the factors appearing in the model formula. The elements of the list should have the same name as the variable and should be either a contrast matrix (specifically, any full-rank matrix with as many rows as there are levels in the factor), or else a function to compute such a matrix given the number of levels.
- **weights**: an optional vector of weights to be used in the fitting process. The length of weights should be the same as the number of observations.
- **z**: a design matrix for the odds ratio model. The number of rows of `z` is

\[
c^2 \sum n_i(n_i - 1)/2,
\]

where \( n_i \) is the cluster size, and \( c \) is the number of categories minus 1.
- **mean.link**: a character string specifying the link function for the means. The following are allowed: "logit", "probit", and "cloglog".
- **corstr**: a character string specifying the log odds. The following are allowed: "independence", "exchangeable", "unstructured", and "userdefined".
control a list of iteration and algorithmic constants. See geese.control for their names and default values. These can also be set as arguments to geese itself.

b an initial estimate for the mean parameters.

alpha an initial estimate for the odds ratio parameters.

scale.fix a logical variable indicating if scale is fixed; it is set at TRUE currently (it can not be FALSE yet!).

scale.val this argument is ignored currently.

int.const a logical variable; if true, the intercepts are constant, and if false, the intercepts are different for different components in the response.

rev a logical variable. For example, for a three level ordered response Y = 2, the accumulated indicator is coded as (1, 0, 0) if true and (0, 1, 1) if false.

... further arguments passed to or from other methods.

Value

An object of class "geese" representing the fit.

Author(s)

Jun Yan <jyan.stat@gmail.com>

References


See Also

glm, lm, geese.

Examples

data(respdis)
resp.l <- reshape(respdis, varying =list(c("y1", "y2", "y3", "y4")),
  v.names = "resp", direction = "long")
resp.l <- resp.l[order(resp.l$id, resp.l$time),]
fit <- ordgee(ordered(resp) ~ trt, id=id, data=resp.l, int.const=FALSE)
summary(fit)

data(ohio)
ohio$resp <- ordered(as.factor(ohio$resp))
fit <- ordgee(resp ~ age + smoke + age:smoke, id = id, data=ohio)
summary(fit)
**relRisk**  
*Fit a Relative Risk Model for Binary data with Log Link*

**Description**

Fit a Relative Risk Model for Binary data with Log Link using the COPY method.

**Usage**

```r
relRisk(formula, id, waves = NULL, data = parent.frame(),
        subset = NULL, contrasts = NULL, na.action = na.omit,
        corstr = "indep", ncopy = 1000,
        control = geese.control(), b = NULL, alpha = NULL)
```

**Arguments**

- `formula`: same as in `geese`
- `id`: same as in `geese`
- `waves`: same as in `geese`
- `data`: same as in `geese`
- `subset`: same as in `geese`
- `contrasts`: same as in `geese`
- `na.action`: same as in `geese`
- `corstr`: same as in `geese`
- `ncopy`: the number of copies of the original data in constructing weight.
- `control`: same as in `geese`
- `b`: initial values for regression coefficients as in `geese` but more difficult to obtain due to the log link.
- `alpha`: same as in `geese`

**Value**

An object of class "geese" representing the fit.

**Author(s)**

Jun Yan <jyan.stat@gmail.com>

**References**

Respdis

Examples

```r
## this example was used in Yu and Yan (2010, techreport)
data(respiratory)
respiratory$treat <- relevel(respiratory$treat, ref = "P")
respiratory$sex <- relevel(respiratory$sex, ref = "M")
respiratory$center <- as.factor(respiratory$center)
## 1 will be the reference level

fit <- relRisk(outcome ~ treat + center + sex + age + baseline + visit,
               id = id, corstr = "ar1", data = respiratory, ncopy=10000)
summary(fit)
## fit <- relRisk(outcome ~ treat + center + sex + age + baseline + visit,
##               id = id, corstr = "ex", data = respiratory)
## summary(fit)
## fit <- relRisk(outcome ~ treat + center + sex + age + baseline + visit,
##               id = id, corstr = "indep", data = respiratory)
## summary(fit)
```

Description

The respdis data frame has 111 rows and 3 columns. The study described in Miller et. al. (1993) is a randomized clinical trial of a new treatment of respiratory disorder. The study was conducted in 111 patients who were randomly assigned to one of two treatments (active, placebo). At each of four visits during the follow-up period, the response status of each patient was classified on an ordinal scale.

Usage

data(respdis)

Format

This data frame contains the following columns:

- **y1, y2, y3, y4** ordered factor measured at 4 visits for the response with levels, 1 < 2 < 3, 1 = poor, 2 = good, and 3 = excellent
- **trt** a factor for treatment with levels, 1 = active, 0 = placebo

References

Examples

data(resppd)
resp.l <- reshape(resppdis, varying = list(c("y1", "y2", "y3", "y4")),
   v.names = "resp", direction = "long")
resp.l <- resp.l[order(resp.l$id, resp.l$time),]
fit <- ordgee(ordered(resp) ~ trt, id = id, data = resp.l, int.const = FALSE)
summary(fit)

z <- model.matrix(~ trt - 1, data = respdis)
ind <- rep(1:111, 4*3/2 * 2*2)
zmat <- z[ind, drop=FALSE]
fit <- ordgee(ordered(resp) ~ trt, id = id, data = resp.l, int.const = FALSE,
   z = zmat, corstr = "exchangeable")
summary(fit)

respiratory

Data from a clinical trial comparing two treatments for a respiratory illness

Description

The data are from a clinical trial of patients with respiratory illness, where 111 patients from two different clinics were randomized to receive either placebo or an active treatment. Patients were examined at baseline and at four visits during treatment. At each examination, respiratory status (categorized as 1 = good, 0 = poor) was determined.

Usage

data(respiratory)

Format

A data frame with 111 observations on the following 7 variables.

center  a numeric vector
d  a numeric vector
age  a numeric vector
baseline  a numeric vector
active  a numeric vector
center2  a numeric vector
female  a numeric vector

Examples

data(respiratory)
## maybe str(respiratory) ; plot(respiratory) ...
Description

The seizure data frame has 59 rows and 7 columns. The dataset has the number of epileptic seizures in each of four two-week intervals, and in a baseline eight-week interval, for treatment and control groups with a total of 59 individuals.

Usage

data(seizure)

Format

This data frame contains the following columns:

- **y1** the number of epileptic seizures in the 1st 2-week interval
- **y2** the number of epileptic seizures in the 2nd 2-week interval
- **y3** the number of epileptic seizures in the 3rd 2-week interval
- **y4** the number of epileptic seizures in the 4th 2-week interval
- **trt** an indicator of treatment
- **base** the number of epileptic seizures in a baseline 8-week interval
- **age** a numeric vector of subject age

Source


References


Examples

data(seizure)

```r
## Diggle, Liang, and Zeger (1994) pp166-168, compare Table 8.10
seiz.l <- reshape(seizure,
                   varying=list(c("base","y1","y2","y3","y4")),
                   v.names="y", times=0:4, direction="long")
seiz.l <- seiz.l[order(seiz.l$id, seiz.l$time),]
seiz.l$t <- ifelse(seiz.l$time == 0, 8, 2)
seiz.l$x <- ifelse(seiz.l$time == 0, 0, 1)
m1 <- geese(y ~ offset(log(t)) + x + trt + x:trt, id = id,
            data=seiz.l, corstr="exch", family=poisson)
summary(m1)
```
m2 <- geese(y ~ offset(log(t)) + x + trt + x:trt, id = id,
             data = seiz.1, subset = id!=49,
             corstr = "exch", family = poisson)
summary(m2)

## Thall and Vail (1998)
seiz.1 <- reshape(seizure, varying=list(c("y1","y2","y3","y4")),
                  v.names="y", direction="long")
seiz.1$lbase <- log(seiz.1$base / 4)
seiz.1$lage <- log(seiz.1$age)
seiz.1$v4 <- ifelse(seiz.1$time == 4, 1, 0)
m3 <- geese(y ~ lbase + trt + lbase:trt + lage + v4,
             sformula = ~ as.factor(time) - 1, id = id,
             data = seiz.1, corstr = "exchangeable", family = poisson)
## compare to Model 13 in Table 4, noticeable difference
summary(m3)

## set up a design matrix for the correlation
z <- model.matrix(~ age, data = seizure)  # data is not seiz.1
## just to illustrate the scale link and correlation link
m4 <- geese(y ~ lbase + trt + lbase:trt + lage + v4,
             sformula = ~ as.factor(time)-1, id = id,
             data = seiz.1, corstr = "ar1", family = poisson,
             zcor = z, cor.link = "fisherz", sca.link = "log")
summary(m4)

---

**sitka89**

*Growth of Sitka Spruce Trees*

**Description**

Impact of ozone on the growth of sitka spruce trees.

**Usage**

data(sitka89)

**Format**

A dataframe

- **size**: size of the tree measured in \( \log(\text{height} \times \text{diameter}^2) \)
- **time**: days after the 1st January, 1988
- **tree**: id number of a tree
- **treat**: ozone: grown under ozone environment, control: ozone free

**Examples**

data(sitka89)
The spruce data frame has 1027 rows and 6 columns. The data consists of measurements on 79 sitka spruce trees over two growing seasons. The trees were grown in four controlled environment chambers, of which the first two, containing 27 trees each, were treated with introduced ozone at 70 ppb whilst the remaining two, containing 12 and 13 trees, were controls.

This data frame contains the following columns:

- **chamber**: a numeric vector of chamber numbers
- **ozone**: a factor with levels enriched and normal
- **id**: a numeric vector of tree id
- **time**: a numeric vector of the time when the measurements were taken, measured in days since Jan. 1, 1988
- **wave**: a numeric vector of the measurement number
- **logsize**: a numeric vector of the log-size


```r
data(spruce)

spruce$contr <- ifelse(spruce$ozone == "enriched", 0, 1)
sitka88 <- spruce[spruce$wave <= 5,]
sitka89 <- spruce[spruce$wave > 5,]
fit.88 <- geese(logsize ~ as.factor(wave) + contr +
               I(time/100*contr) - 1,
               id=id, data=sitka88, corstr="ar1")
summary(fit.88)

fit.89 <- geese(logsize ~ as.factor(wave) + contr - 1,
               id=id, data=sitka89, corstr="ar1")
summary(fit.89)
```
Index

∗Topic datasets
  dietox, 3
  koch, 14
  ohio, 15
  respdis, 19
  respiratory, 20
  seizure, 21
  sitka89, 22
  spruce, 23

∗Topic models
  compCoef, 2
  geeglm, 5
  geese, 8
  geese.control, 12
  ordgee, 15
  relRisk, 18

∗Topic nonlinear
  geese, 8
  ordgee, 15

∗Topic optimize
  geese.control, 12

∗Topic regression
  fixed2Zcor, 4
  genZcor, 13

anova.geeglm, 7
compCoef, 2
dietox, 3
fixed2Zcor, 4, 13
geeglm, 4, 5
geese, 7, 8, 17
geese.control, 9, 12, 17
genZcor, 4, 13
glm, 7, 9, 10, 17
humbelbee (genZcor), 13

koch, 14
lm, 10, 17
ohio, 15
ordgee, 10, 15
print.geese (geese), 8
print.summary.geese (geese), 8
relRisk, 18
respdis, 19
respiratory, 20
respiratoryWide (respiratory), 20
seizure, 21
sitka89, 22
spruce, 23
summary.geese (geese), 8