# Package 'geepack'

June 7, 2024

**Version** 1.3.11

**Title** Generalized Estimating Equation Package

**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. See e.g. Halekoh and Højsgaard, (2005, <doi:10.18637/jss.v015.i02>), for details.

**Encoding** UTF-8

**LazyData** true

**License** GPL (>= 3)

**NeedsCompilation** yes

**Depends** R (>= 3.5.0), methods

**Imports** MASS, broom, magrittr

**RoxygenNote** 7.2.3

**Author** Søren Højsgaard [aut, cre, cph],
Ulrich Halekoh [aut, cph],
Jun Yan [aut, cph],
Claus Thorn Ekstrøm [ctb]

**Repository** CRAN

**Date/Publication** 2024-06-06 22:40:06 UTC

## Contents

---

`compCoef` *Compare Regression Coefficiente between Nested Models*

---

## 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

Allison, P. D. (1995). The impact of random predictors on comparisons of coefficients between models: Comment on Clogg, Petkova, and Haritou. *American Journal of Sociology*, **100**(5), 1294–1305.

Clogg, C. C., Petkova, E., and Haritou, A. (1995). Statistical methods for comparing regression coefficients between models. *American Journal of Sociology*, **100**(5), 1261–1293.

Yan, J., Aseltine, R., and Harel, O. (2011). Comparing Regression Coefficients Between Nested Linear Models for Clustered Data with Generalized Estimating Equations. *Journal of Educational and Behaviorial Statistics*, Forthcoming.

## Examples

```
## 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)
```

---

`dietox` *Growth curves of pigs in a 3x3 factorial experiment*

---

## Description

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

## Usage

```
dietox
```

## Format

This data frame contains the following columns:

**Weight** Weight in Kg

**Feed** Cumulated feed intake in Kg

**Time** Time (in weeks) in the experiment

**Pig** Factor; id of each pig

**Evit** Factor; vitamin E dose; see 'details'.

**Cu** Factor, copper dose; see 'details'

**Start** Start weight in experiment, i.e. weight at week 1.

**Litter** Factor, id of litter of each pig

## Details

Data contains weight of slaughter pigs measured weekly for 12 weeks. Data also contains the startweight (i.e. the weight at week 1). The treatments are 3 different levels of Evit = vitamin E (dose: 0, 100, 200 mg dl-alpha-tocopheryl acetat /kg feed) in combination with 3 different levels of Cu=copper (dose: 0, 35, 175 mg/kg feed) in the feed. The cumulated feed intake is also recorded. The pigs are littermates.

## Source

Lauridsen, C., Højsgaard, S.,Sørensen, M.T. C. (1999) Influence of Dietary Rapeseed Oli, Vitamin E, and Copper on Performance and Antioxidant and Oxidative Status of Pigs. J. Anim. Sci.77:906-916

## Examples

```
data(dietox)
head(dietox)
## Not run:
if (require(ggplot2)){
  qplot(Time, Weight, data=dietox, col=Pig) + geom_line() +
        theme(legend.position = "none") + facet_grid(Evit~Cu)
} else {
  coplot(Weight ~ Time | Evit * Cu, data=dietox)
}

## End(Not run)
```

`fixed2Zcor` *Construct zcor vector*

## Description

Construct zcor vector (of fixed correlations) from a fixed working correlation matrix, a specification of clusters and a specifcation 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@math.aau.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)),]
simdatPerm <- simdatPerm[order(simdatPerm$idvar),]
head(simdatPerm)

cor.fixed <- matrix(c(1    , 0.5  , 0.25,  0.125, 0.125,
                      0.5  , 1    , 0.25,  0.125, 0.125,
                      0.25 , 0.25 , 1   ,  0.5  , 0.125,
                      0.125, 0.125, 0.5  , 1    , 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)
zcor

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",
  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 |
| 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 each cluster appear as contiguous rows in data. If data is not sorted this way, the function will not identify the clusters correctly. If data is not sorted this way, a warning will be issued. Please consult the package vignette for details. |
| waves | Wariable specifying the ordering of repeated mesurements on the same unit. Also used in connection with missing values. Please consult the package vignette for details. |
| 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. Defualt '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@math.aau.dk>

## References

Halekoh, U.; Højsgaard, S. and Yan, J (2006) The R Package geepack for Generalized Estimating Equations. Journal of Statistical Software, 15, 2, 1-11"

Liang, K.Y. and Zeger, S.L. (1986) Longitudinal data analysis using generalized linear models. Biometrika, *73* 13-22.

Prentice, R.L. and Zhao, L.P. (1991). Estimating equations for parameters in means and covariances of multivariate discrete and continuous responses. Biometrics, *47* 825-839.

## See Also

geese, 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")
gee1
coef(gee1)
vcov(gee1)
summary(gee1)
coef(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 fit 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",
  ...
)
```

## 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" |
| ... | 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 sum(clusz * (clusz - 1)) / 2.

## Value

An object of class "geese" representing the fit.

## Author(s)

Jun Yan <jyan.stat@gmail.com>

## References

Yan, J. and J.P. Fine (2004) Estimating Equations for Association Structures. *Statistics in Medicine*, **23**, 859–880.

## 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 <- 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, 0.25, 1, 0.5, 0.125,
                      0.125, 0.125, 0.5, 1, 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)

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)

###### 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 unstrctured correlation.
#### in this case, zcor has 4*3/2 = 6 columns, and 50 * 6 = 300 rows
zcor <- genZcor(clusz = 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.
z2 <- matrix(NA, 300, 4)
z2[,1] <- zcor[,1] + zcor[,4] + zcor[,6]
```

```
z2[,2:4] <- zcor[, c(2, 3, 5)]
summary(geese(y ~ x1 + x2, id = id, data = dat, sformula = ~ x1,
              corstr = "userdefined", zcor = z2,
              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)))
mdat <- dat[good,]

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

---

`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

| | |
|---|---|
| `epsilon` | positive convergence tolerance epsilon; the iterations converge when the absolute value of the difference in parameter estimate is below `epsilon`. |
| `maxit` | integer giving the maximal number of Fisher Scoring iteration. |
| `trace` | logical indicating if output should be produced for each iteration. |
| `scale.fix` | logical indicating if the scale should be fixed. |
| `jack` | logical indicating if approximate jackknife variance estimate should be computed. |
| `j1s` | logical indicating if 1-step jackknife variance estimate should be computed. |
| `fij` | 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'.

---

genZcor *genZcor*

---

## Description

constructs the design matrix for the correlation structures: independence, echangeable, ar1 and unstructured The user will need this function only as a basis to construct a user defined correlation structure: use 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

```
genZcor(clusz, waves, corstrv)
```

## Arguments

| | |
|---|---|
| `clusz` | integer vector giving the number of observations in each cluster. |
| `waves` | integer vector, obervations in the same cluster with values of wave i and j have the correlation $latex$. |
| `corstrv` | correlation structures: 1=independence, 2=exchangeable, 3=ar1, 4=unstructured. |

## Value

The design matrix for the correlation structure.

## Author(s)

Jun Yan <jyan.stat@gmail.com>

## See Also

`fixed2Zcor`

## Examples

```
# example to construct a Toeplitz correlation structure
#    sigma_ij=sigma_|i-j|

# data set with 5 clusters and maximally 4 observations (visits) per cluster
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()

# 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

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

---

`muscatine` *Data on Obesity from the Muscatine Coronary Risk Factor Study.*

---

## Description

The data are from the Muscatine Coronary Risk Factor (MCRF) study, a longitudinal survey of school-age children in Muscatine, Iowa. The MCRF study had the goal of examining the development and persistence of risk factors for coronary disease in children. In the MCRF study, weight and height measurements of five cohorts of children, initially aged 5-7, 7-9, 9-11, 11-13, and 13-15 years, were obtained biennially from 1977 to 1981. Data were collected on 4856 boys and girls. On the basis of a comparison of their weight to age-gender specific norms, children were classified as obese or not obese.

## Usage

```
muscatine
```

## Format

A dataframe with 14568 rows and 7 variables:

**id** identifier of child.

**gender** gender of child

**base_age** baseline age

**age** current age

**occasion** identifier of occasion of recording

**obese** 'yes' or 'no'

**numobese** obese in numerical form: 1 corresponds to 'yes' and 0 corresponds to 'no'.

## Source

https://content.sph.harvard.edu/fitzmaur/ala2e/muscatine.txt

Woolson, R.F. and Clarke, W.R. (1984). Analysis of categorical incompletel longitudinal data. Journal of the Royal Statistical Society, Series A, 147, 87-99.

## Examples

```
muscatine$cage <- muscatine$age - 12
muscatine$cage2 <- muscatine$cage^2

f1 <- numobese ~ gender
f2 <- numobese ~ gender + cage + cage2 +
    gender:cage + gender:cage2

gee1 <- geeglm(formula = f1, id = id,
               waves = occasion, data = muscatine, family = binomial(),
               corstr = "independence")

gee2 <- geeglm(formula = f2, id = id,
               waves = occasion, data = muscatine, family = binomial(),
               corstr = "independence")

tidy(gee1)
tidy(gee2)
QIC(gee1)
QIC(gee2)
```

---

# ohio *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

```
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

Fitzmaurice, G.M. and Laird, N.M. (1993) A likelihood-based method for analyzing longitudinal binary responses, *Biometrika* **80**: 141–151.

## Examples

```
data(ohio)

fit.ex <- geeglm(resp ~ age + smoke + age:smoke, id=id, data=ohio,
   family=binomial, corstr="exch", scale.fix=TRUE)
QIC(fit.ex)

fit.ar <- geeglm(resp ~ age + smoke + age:smoke, id=id, data=ohio,
   family=binomial, corstr="ar1", scale.fix=TRUE)
QIC(fit.ex)
```

---

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

```
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

Heagerty, P.J. and Zeger, S.L. (1996) Marginal regression models for clustered ordinal measurements. *JASA*, **91** 1024–1036.

## 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)
```

---

# QIC.geeglm *Quasi Information Criterion*

---

## Description

Function for calculating the quasi-likelihood under the independence model information criterion (QIC), quasi-likelihood, correlation information criterion (CIC), and corrected QIC for one or several fitted geeglm model object from the geepack package.

## Usage

```
## S3 method for class 'geeglm'
QIC(object, ..., tol = .Machine$double.eps, env = parent.frame())

## S3 method for class 'ordgee'
QIC(object, ..., tol = .Machine$double.eps, env = parent.frame())

## S3 method for class 'geekin'
QIC(object, ..., tol = .Machine$double.eps, env = parent.frame())

QIC(object, ..., tol = .Machine$double.eps, env = parent.frame())
```

## Arguments

| | |
|---|---|
| object | a fitted GEE model from the geepack package. Currently only works on geeglm objects. |
| ... | optionally more fitted geeglm model objects. |
| tol | the tolerance used for matrix inversion. |
| env | environment. |

## Details

QIC is used to select a correlation structure. The QICu is used to compare models that have the same working correlation matrix and the same quasi-likelihood form but different mean specifications. CIC has been suggested as a more robust alternative to QIC when the model for the mean may not fit the data very well and when models with different correlation structures are compared.

Models with smaller values of QIC, CIC, QICu, or QICC are preferred.

If the MASS package is loaded then the `ginv` function is used for matrix inversion. Otherwise the standard `solve` function is used.

## Value

A vector or matrix with the QIC, QICu, quasi likelihood, CIC, the number of mean effect parameters, and the corrected QIC for each GEE object

## Author(s)

Claus Ekstrom <claus@rprimer.dk>, Brian McLoone <bmcloone@pdx.edu> and Steven Orzack <orzack@freshpond.org>

## References

Pan, W. (2001). *Akaike's information criterion in generalized estimating equations*. Biometrics, 57, 120-125.
Hardin, J.W. and Hilbe, J.M. (2012). *Generalized Estimating Equations, 2nd Edition*, Chapman and Hall/CRC: New York.

Hin, L.-Y. and Wang, Y-G.
(2009). \emph{Working-correlation-structure identification in
generalized estimating equations}, Statistics in Medicine 28:
generalized estimating equations}, Statistics in Medicine 28:
642-658. \cr Thall, P.F.  and Vail, S.C. (1990). \emph{Some
Covariance Models for Longitudinal Count Data with
Overdispersion}.  Biometrics, 46, 657-671.

## See Also

geeglm

## Examples

```
library(geepack)
data(ohio)
fit <- geeglm(resp ~ age + smoke + age:smoke, id=id, data=ohio,
            family=binomial, corstr="exch", scale.fix=TRUE)
fit2 <- geeglm(resp ~ age + smoke + age:smoke, id=id, data=ohio,
            family=binomial, corstr="ar1", scale.fix=TRUE)
QIC(fit, fit2)
```

---

# 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

```
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

Lumley, T., Kornmal, R. and Ma, S. (2006). Relative risk regression in medical research: models, contrasts, estimators, and algorithms. UW Biostatistics Working Paper Series 293, University of Washington.

## Examples

```
## 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)
```

respdis *Clustered Ordinal Respiratory Disorder*

## 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 patients was classified on an ordinal scale.

## Usage

```
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

Miller, M.E., David, C.S., and Landis, R.J. (1993) The analysis of longitudinal polytomous data: Generalized estimating equation and connections with weighted least squares, *Biometrics* **49**: 1033-1048.

## 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)

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. The respiratory status (categorized as 1 = good, 0 = poor) was determined at each visit.

## Usage

```
respiratory
```

## Format

A data frame with 444 observations on the following 8 variables.

**center** a numeric vector

**id** a numeric vector

**treat** treatment or placebo

**sex** M or F

**age** in years at baseline

**baseline** resporatory status at baseline

**visit** id of each of four visits

**outcome** respiratory status at each visit

## Examples

```
data(respiratory)
data(respiratory, package="geepack")
respiratory$center <- factor(respiratory$center)
head(respiratory)

m1 <- glm(outcome ~ center + treat + age + baseline, data=respiratory,
          family=binomial())
gee.ind <- geeglm(outcome ~ center + treat + age + baseline, data=respiratory, id=id,
          family=binomial(), corstr="independence")
gee.exc <- geeglm(outcome ~ center + treat + age + baseline, data=respiratory, id=id,
             family=binomial(), corstr="exchangeable")
gee.uns <- geeglm(outcome ~ center + treat + age + baseline, data=respiratory, id=id,
             family=binomial(), corstr="unstructured")
gee.ar1 <- geeglm(outcome ~ center + treat + age + baseline, data=respiratory, id=id,
             family=binomial(), corstr="ar1")

mlist <- list(gee.ind, gee.exc, gee.uns, gee.ar1)
```

```
do.call(rbind, lapply(mlist, QIC))
lapply(mlist, tidy)
```

---

`seizure` *Epiliptic Seizures*

---

## Description

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

## Usage

```
seizure
```

## Format

This data frame contains the following columns:

**y1** the number of epiliptic seizures in the 1st 2-week interval

**y2** the number of epiliptic seizures in the 2nd 2-week interval

**y3** the number of epiliptic seizures in the 3rd 2-week interval

**y4** the number of epiliptic seizures in the 4th 2-week interval

**trt** an indicator of treatment

**base** the number of epilitic seizures in a baseline 8-week interval

**age** a numeric vector of subject age

## Source

Thall, P.F. and Vail S.C. (1990) Some covariance models for longitudinal count data with overdispersion. *Biometrics* **46**: 657–671.

## References

Diggle, P.J., Liang, K.Y., and Zeger, S.L. (1994) Analysis of Longitudinal Data. Clarendon Press.

## 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 <- 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)

## Thall and Vail (1990)
seiz.l <- reshape(seizure, varying=list(c("y1","y2","y3","y4")),
                  v.names="y", direction="long")
seiz.l <- seiz.l[order(seiz.l$id, seiz.l$time),]
seiz.l$lbase <- log(seiz.l$base / 4)
seiz.l$lage <- log(seiz.l$age)
seiz.l$v4 <- ifelse(seiz.l$time == 4, 1, 0)
m3 <- geese(y ~ lbase + trt + lbase:trt + lage + v4,
            sformula = ~ as.factor(time) - 1, id = id,
            data = seiz.l, 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.l
## 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.l, 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

```
sitka89
```

## Format

A dataframe

**size:** size of the tree measured in $log(height * diamter^2)$

**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)
```

---

spruce *Log-size of 79 Sitka spruce trees*

---

## Description

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.

## Usage

```
spruce
```

## Format

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

## Source

Diggle, P.J., Liang, K.Y., and Zeger, S.L. (1994) Analysis of Longitudinal Data, Clarendon Press.

## Examples

```
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