1 gamma.gee: Generalized Estimating Equation for Gamma Regression

The GEE gamma is similar to standard gamma regression (appropriate when you have an uncensored, positive-valued, continuous dependent variable such as the time until a parliamentary cabinet falls). Unlike in gamma regression, GEE gamma allows for dependence within clusters, such as in longitudinal data, although its use is not limited to just panel data. GEE models make no distributional assumptions but require three specifications: a mean function, a variance function, and a "working" correlation matrix for the clusters, which models the dependence of each observation with other observations in the same cluster. The "working" correlation matrix is a $T \times T$ matrix of correlations, where T is the size of the largest cluster and the elements of the matrix are correlations between within-cluster observations. The appeal of GEE models is that it gives consistent estimates of the parameters and consistent estimates of the standard errors can be obtained using a robust "sandwich" estimator even if the "working" correlation matrix is incorrectly specified. If the "working" correlation matrix is correctly specified, GEE models will give more efficient estimates of the parameters. GEE models measure population-averaged effects as opposed to cluster-specific effects (See **(author?)** [4]).

1.0.1 Syntax

```
> z.out <- zelig(Y ~ X1 + X2, model = "gamma.gee",
                 id = "X3", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

where `id` is a variable which identifies the clusters. The data should be sorted by `id` and should be ordered within each cluster when appropriate.

1.0.2 Additional Inputs

- `robust`: defaults to `TRUE`. If `TRUE`, consistent standard errors are estimated using a "sandwich" estimator.

Use the following arguments to specify the structure of the "working" correlations within clusters:

- `corstr`: defaults to `"independence"`. It can take on the following arguments:
  - Independence (`corstr = "independence"`): $\text{cor}(y_{it}, y_{it'}) = 0$, $\forall t, t'$ with $t \neq t'$. It assumes that there is no correlation within the clusters and the model becomes equivalent to standard gamma regression. The "working" correlation matrix is the identity matrix.

– Fixed (`corstr = "fixed"`): If selected, the user must define the "working" correlation matrix with the `R` argument rather than estimating it from the model.

– Stationary m dependent (`corstr = "stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{|t-t'|} & \text{if} \quad |t-t'| \leq m \\ 0 & \text{if} \quad |t-t'| > m \end{cases}$$

If (`corstr = "stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. Choose this option when the correlations are assumed to be the same for observations of the same $|t-t'|$ periods apart for $|t-t'| \leq m$.

Sample "working" correlation for Stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_1 & \alpha_2 & 0 & 0 \\ \alpha_1 & 1 & \alpha_1 & \alpha_2 & 0 \\ \alpha_2 & \alpha_1 & 1 & \alpha_1 & \alpha_2 \\ 0 & \alpha_2 & \alpha_1 & 1 & \alpha_1 \\ 0 & 0 & \alpha_2 & \alpha_1 & 1 \end{pmatrix}$$

– Non-stationary m dependent (`corstr = "non_stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{tt'} & \text{if} \quad |t-t'| \leq m \\ 0 & \text{if} \quad |t-t'| > m \end{cases}$$

If (`corstr = "non_stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. This option relaxes the assumption that the correlations are the same for all observations of the same $|t-t'|$ periods apart.

Sample "working" correlation for Non-stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_{12} & \alpha_{13} & 0 & 0 \\ \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} & 0 \\ \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} & \alpha_{35} \\ 0 & \alpha_{24} & \alpha_{34} & 1 & \alpha_{45} \\ 0 & 0 & \alpha_{35} & \alpha_{45} & 1 \end{pmatrix}$$

– Exchangeable (`corstr = "exchangeable"`): $\text{cor}(y_{it}, y_{it'}) = \alpha$, $\forall t, t'$ with $t \neq t'$. Choose this option if the correlations are assumed to be the same for all observations within the cluster.

Sample "working" correlation for Exchangeable

$$\begin{pmatrix} 1 & \alpha & \alpha & \alpha & \alpha \\ \alpha & 1 & \alpha & \alpha & \alpha \\ \alpha & \alpha & 1 & \alpha & \alpha \\ \alpha & \alpha & \alpha & 1 & \alpha \\ \alpha & \alpha & \alpha & \alpha & 1 \end{pmatrix}$$

– Stationary mth order autoregressive (`corstr = "AR-M"`): If (`corstr = "AR-M"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. For example, the first order autoregressive model (AR-1) implies $\text{cor}(y_{it}, y_{it'}) = \alpha^{|t-t'|}, \forall t, t'$ with $t \neq t'$. In AR-1, observation 1 and observation 2 have a correlation of α. Observation 2 and observation 3 also have a correlation of α. Observation 1 and observation 3 have a correlation of α^2, which is a function of how 1 and 2 are correlated (α) multiplied by how 2 and 3 are correlated (α). Observation 1 and 4 have a correlation that is a function of the correlation between 1 and 2, 2 and 3, and 3 and 4, and so forth.

Sample "working" correlation for Stationary AR-1 (`Mv`=1)

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ \alpha & 1 & \alpha & \alpha^2 & \alpha^3 \\ \alpha^2 & \alpha & 1 & \alpha & \alpha^2 \\ \alpha^3 & \alpha^2 & \alpha & 1 & \alpha \\ \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \end{pmatrix}$$

– Unstructured (`corstr = "unstructured"`): $\text{cor}(y_{it}, y_{it'}) = \alpha_{tt'}, \forall t, t'$ with $t \neq t'$. No constraints are placed on the correlations, which are then estimated from the data.

- `Mv`: defaults to 1. It specifies the number of periods of correlation and only needs to be specified when `corstr` is `"stat_M_dep"`, `"non_stat_M_dep"`, or `"AR-M"`.

- `R`: defaults to `NULL`. It specifies a user-defined correlation matrix rather than estimating it from the data. The argument is used only when `corstr` is `"fixed"`. The input is a $T \times T$ matrix of correlations, where T is the size of the largest cluster.

1.0.3 Examples

1. Example with Exchangeable Dependence

   Attaching the sample turnout dataset:

   ```
   > data(coalition)
   ```

   Sorted variable identifying clusters

   ```
   > coalition$cluster <- c(rep(c(1:62),5),rep(c(63),4))
   > sorted.coalition <- coalition[order(coalition$cluster),]
   ```

   Estimating model and presenting summary:

   ```
   > z.out <- zelig(duration ~ fract + numst2, model = "gamma.gee", id = "cluster", data =
   ```

```
The following object(s) are masked from 'package:boot':

    polar
  (Intercept)         fract        numst2
-0.0129597411  0.0001148931 -0.0173874664


 How to cite this model in Zelig:
  Patrick Lam. 2012.
  "gamma.gee: General Estimating Equation for Gamma Regression"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig

> summary(z.out)

 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Reciprocal
 Variance to Mean Relation: Gamma
 Correlation Structure:     Exchangeable

Call:
gee(formula = duration ~ fract + numst2, id = NUMERIC314, data = Data.frame,
    family = Function, corstr = "exchangeable")

Summary of Residuals:
       Min         1Q     Median         3Q        Max
-51.662849 -10.922635  -3.338295   9.384375  33.481595


Coefficients:
                 Estimate   Naive S.E.    Naive z  Robust S.E.  Robust z
(Intercept) -0.0129634095 1.328203e-02 -0.9760113 0.0126829873 -1.022110
fract        0.0001149138 1.719796e-05  6.6818299 0.0000147418  7.795104
numst2      -0.0174009000 5.886821e-03 -2.9559076 0.0062943755 -2.764516

Estimated Scale Parameter:  0.6291527
Number of Iterations:  1

Working Correlation
             [,1]         [,2]         [,3]         [,4]         [,5]
[1,]  1.000000000 -0.008042939 -0.008042939 -0.008042939 -0.008042939
[2,] -0.008042939  1.000000000 -0.008042939 -0.008042939 -0.008042939
[3,] -0.008042939 -0.008042939  1.000000000 -0.008042939 -0.008042939
```

```
[4,] -0.008042939 -0.008042939 -0.008042939  1.000000000 -0.008042939
[5,] -0.008042939 -0.008042939 -0.008042939 -0.008042939  1.000000000
```

Setting the explanatory variables at their default values (mode for factor variables and mean for non-factor variables), with numst2 set to the vector 0 = no crisis, 1 = crisis.

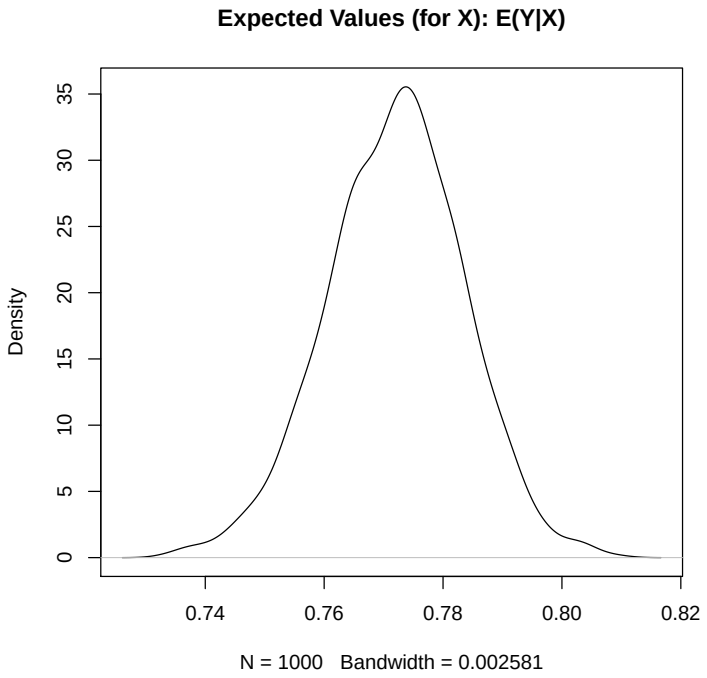
```
> x.low <- setx(z.out, numst2 = 0)
> x.high <- setx(z.out, numst2 = 1)
```

Simulate quantities of interest

```
> s.out <- sim(z.out, x = x.low, x1 = x.high)
> summary(s.out)

Model:  gamma.gee
Number of simulations:  1000

Values of X
  (Intercept)    fract numst2
1           1 718.8121      0
attr(,"assign")
[1] 0 1 2

Values of X1
  (Intercept)    fract numst2
1           1 718.8121      1
attr(,"assign")
[1] 0 1 2

Expected Values (for x): E(Y|X)
  mean    sd   50%  2.5% 97.5%
 0.528 0.002 0.528 0.524 0.532

Expected Values (for x1): E(Y|X1)
  mean    sd   50%  2.5% 97.5%
 0.521 0.001 0.521 0.519 0.523

First Differences: E(Y|X1) - E(Y|X)
   mean    sd    50%   2.5%  97.5%
 -0.007 0.003 -0.007 -0.012 -0.002
```

Generate a plot of quantities of interest:

```
> plot(s.out)
```



1.0.4 The Model

Suppose we have a panel dataset, with Y_{it} denoting the positive-valued, continuous dependent variable for unit i at time t. Y_i is a vector or cluster of correlated data where y_{it} is correlated with $y_{it'}$ for some or all t, t'. Note that the model assumes correlations within i but independence across i.

- The *stochastic component* is given by the joint and marginal distributions

$$\begin{aligned} Y_i &\sim f(y_i \mid \lambda_i) \\ Y_{it} &\sim g(y_{it} \mid \lambda_{it}) \end{aligned}$$

  where f and g are unspecified distributions with means λ_i and λ_{it}. GEE models make no distributional assumptions and only require three specifications: a mean function, a variance function, and a correlation structure.

- The *systematic component* is the *mean function*, given by:

$$\lambda_{it} = \frac{1}{x_{it}\beta}$$

  where x_{it} is the vector of k explanatory variables for unit i at time t and β is the vector of coefficients.

- The *variance function* is given by:

$$V_{it} = \lambda_{it}^2 = \frac{1}{(x_{it}\beta)^2}$$

- The *correlation structure* is defined by a $T \times T$ "working" correlation matrix, where T is the size of the largest cluster. Users must specify the structure of the "working" correlation matrix *a priori*. The "working" correlation matrix then enters the variance term for each i, given by:

$$V_i = \phi\, A_i^{\frac{1}{2}} R_i(\alpha) A_i^{\frac{1}{2}}$$

  where A_i is a $T \times T$ diagonal matrix with the variance function $V_{it} = \lambda_{it}^2$ as the tth diagonal element, $R_i(\alpha)$ is the "working" correlation matrix, and ϕ is a scale parameter. The parameters are then estimated via a quasi-likelihood approach.

- In GEE models, if the mean is correctly specified, but the variance and correlation structure are incorrectly specified, then GEE models provide consistent estimates of the parameters and thus the mean function as well, while consistent estimates of the standard errors can be obtained via a robust "sandwich" estimator. Similarly, if the mean and variance are correctly specified but the correlation structure is incorrectly specified, the parameters can be estimated consistently and the standard errors can be estimated consistently with the sandwich estimator. If all three are specified correctly, then the estimates of the parameters are more efficient.

- The robust "sandwich" estimator gives consistent estimates of the standard errors when the correlations are specified incorrectly only if the number of units i is relatively large and the number of repeated periods t is relatively small. Otherwise, one should use the "naïve" model-based standard errors, which assume that the specified correlations are close approximations to the true underlying correlations. See **?**] for more details.

1.0.5 Quantities of Interest

- All quantities of interest are for marginal means rather than joint means.

- The method of bootstrapping generally should not be used in GEE models. If you must bootstrap, bootstrapping should be done within clusters, which is not currently supported in Zelig. For conditional prediction models, data should be matched within clusters.

- The expected values (`qi$ev`) for the GEE gamma model is the mean:

$$E(Y) = \lambda_c = \frac{1}{x_c\beta},$$

  given draws of β from its sampling distribution, where x_c is a vector of values, one for each independent variable, chosen by the user.

- The first difference (`qi$fd`) for the GEE gamma model is defined as

$$\text{FD} = \Pr(Y = 1 \mid x_1) - \Pr(Y = 1 \mid x).$$

- In conditional prediction models, the average expected treatment effect (`att.ev`) for the treatment group is

$$\frac{1}{\sum_{i=1}^{n}\sum_{t=1}^{T} tr_{it}} \sum_{i:tr_{it}=1}^{n} \sum_{t:tr_{it}=1}^{T} \left\{ Y_{it}(tr_{it} = 1) - E[Y_{it}(tr_{it} = 0)] \right\},$$

  where tr_{it} is a binary explanatory variable defining the treatment ($tr_{it} = 1$) and control ($tr_{it} = 0$) groups. Variation in the simulations are due to uncertainty in simulating $E[Y_{it}(tr_{it} = 0)]$, the counterfactual expected value of Y_{it} for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $tr_{it} = 0$.

1.0.6 Output Values

The output of each Zelig command contains useful information which you may view. For example, if you run `z.out <- zelig(y ~ x, model = "gamma.gee", id, data)`, then you may examine the available information in `z.out` by using `names(z.out)`, see the `coefficients` by using `z.out$coefficients`, and a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output object `z.out`, you may extract:
  - `coefficients`: parameter estimates for the explanatory variables.
  - `residuals`: the working residuals in the final iteration of the fit.
  - `fitted.values`: the vector of fitted values for the systemic component.
  - `linear.predictors`: the vector of $x_{it}\beta$
  - `max.id`: the size of the largest cluster.
- From `summary(z.out)`, you may extract:
  - `coefficients`: the parameter estimates with their associated standard errors, p-values, and z-statistics.
  - `working.correlation`: the "working" correlation matrix
- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as matrices indexed by simulation $\times$ `x`-observation (for more than one `x`-observation). Available quantities are:
  - `qi$ev`: the simulated expected values for the specified values of `x`.

– `qi$fd`: the simulated first difference in the expected probabilities for the values specified in `x` and `x1`.

– `qi$att.ev`: the simulated average expected treatment effect for the treated from conditional prediction models.

How To Cite the *gamma.gee* Zelig model

Patrick Lam. 2007. "gamma.gee: Generalized Estimating Equation for Gamma Regression," in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software," `http://gking.harvard.edu/zelig`.

How to Cite the Zelig Software Package

To cite Zelig as a whole, please reference these two sources:

Kosuke Imai, Gary King, and Olivia Lau. 2007. "Zelig: Everyone's Statistical Software," `http://GKing.harvard.edu/zelig`.

Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The gee function is part of the gee package by Vincent J. Carey, ported to R by Thomas Lumley and Brian Ripley. Advanced users may wish to refer to `help(gee)` and `help(family)`. Sample data are from [1].

2 `logit.gee`: Generalized Estimating Equation for Logistic Regression

The GEE logit estimates the same model as the standard logistic regression (appropriate when you have a dichotomous dependent variable and a set of explanatory variables). Unlike in logistic regression, GEE logit allows for dependence within clusters, such as in longitudinal data, although its use is not limited to just panel data. The user must first specify a "working" correlation matrix for the clusters, which models the dependence of each observation with other observations in the same cluster. The "working" correlation matrix is a $T \times T$ matrix of correlations, where T is the size of the largest cluster and the elements of the matrix are correlations between within-cluster observations. The appeal of GEE models is that it gives consistent estimates of the parameters and consistent estimates of the standard errors can be obtained using a robust "sandwich" estimator even if the "working" correlation matrix is

incorrectly specified. If the "working" correlation matrix is correctly specified, GEE models will give more efficient estimates of the parameters. GEE models measure population-averaged effects as opposed to cluster-specific effects (See **(author?)** [4]).

2.0.7 Syntax

```
> z.out <- zelig(Y ~ X1 + X2, model = "logit.gee",
                 id = "X3", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

where `id` is a variable which identifies the clusters. The data should be sorted by `id` and should be ordered within each cluster when appropriate.

2.0.8 Additional Inputs

- `robust`: defaults to `TRUE`. If `TRUE`, consistent standard errors are estimated using a "sandwich" estimator.

Use the following arguments to specify the structure of the "working" correlations within clusters:

- `corstr`: defaults to `"independence"`. It can take on the following arguments:
  - Independence (`corstr = "independence"`): $\text{cor}(y_{it}, y_{it'}) = 0$, $\forall t, t'$ with $t \neq t'$. It assumes that there is no correlation within the clusters and the model becomes equivalent to standard logistic regression. The "working" correlation matrix is the identity matrix.
  - Fixed (`corstr = "fixed"`): If selected, the user must define the "working" correlation matrix with the `R` argument rather than estimating it from the model.
  - Stationary m dependent (`corstr = "stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{|t-t'|} & \text{if} \quad |t-t'| \leq m \\ 0 & \text{if} \quad |t-t'| > m \end{cases}$$

    If (`corstr = "stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. Choose this option when the correlations are assumed to be the same for observations of the same $|t-t'|$ periods apart for $|t-t'| \leq m$.

    Sample "working" correlation for Stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_1 & \alpha_2 & 0 & 0 \\ \alpha_1 & 1 & \alpha_1 & \alpha_2 & 0 \\ \alpha_2 & \alpha_1 & 1 & \alpha_1 & \alpha_2 \\ 0 & \alpha_2 & \alpha_1 & 1 & \alpha_1 \\ 0 & 0 & \alpha_2 & \alpha_1 & 1 \end{pmatrix}$$

– Non-stationary m dependent (`corstr = "non_stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{tt'} & \text{if} \quad |t - t'| \leq m \\ 0 & \text{if} \quad |t - t'| > m \end{cases}$$

If (`corstr = "non_stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. This option relaxes the assumption that the correlations are the same for all observations of the same $|t - t'|$ periods apart.

Sample "working" correlation for Non-stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_{12} & \alpha_{13} & 0 & 0 \\ \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} & 0 \\ \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} & \alpha_{35} \\ 0 & \alpha_{24} & \alpha_{34} & 1 & \alpha_{45} \\ 0 & 0 & \alpha_{35} & \alpha_{45} & 1 \end{pmatrix}$$

– Exchangeable (`corstr = "exchangeable"`): $\text{cor}(y_{it}, y_{it'}) = \alpha$, $\forall t, t'$ with $t \neq t'$. Choose this option if the correlations are assumed to be the same for all observations within the cluster.

Sample "working" correlation for Exchangeable

$$\begin{pmatrix} 1 & \alpha & \alpha & \alpha & \alpha \\ \alpha & 1 & \alpha & \alpha & \alpha \\ \alpha & \alpha & 1 & \alpha & \alpha \\ \alpha & \alpha & \alpha & 1 & \alpha \\ \alpha & \alpha & \alpha & \alpha & 1 \end{pmatrix}$$

– Stationary mth order autoregressive (`corstr = "AR-M"`): If (`corstr = "AR-M"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. For example, the first order autoregressive model (AR-1) implies $\text{cor}(y_{it}, y_{it'}) = \alpha^{|t-t'|}, \forall t, t'$ with $t \neq t'$. In AR-1, observation 1 and observation 2 have a correlation of α. Observation 2 and observation 3 also have a correlation of α. Observation 1 and observation 3 have a correlation of α^2, which is a function of how 1 and 2 are correlated (α) multiplied by how 2 and 3 are correlated (α). Observation 1 and 4 have a correlation that is a function of the correlation between 1 and 2, 2 and 3, and 3 and 4, and so forth.

Sample "working" correlation for Stationary AR-1 (`Mv`=1)

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ \alpha & 1 & \alpha & \alpha^2 & \alpha^3 \\ \alpha^2 & \alpha & 1 & \alpha & \alpha^2 \\ \alpha^3 & \alpha^2 & \alpha & 1 & \alpha \\ \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \end{pmatrix}$$

  - Unstructured (`corstr = "unstructured"`): $\text{cor}(y_{it}, y_{it'}) = \alpha_{tt'}, \forall t, t'$ with $t \neq t'$. No constraints are placed on the correlations, which are then estimated from the data.

- `Mv`: defaults to 1. It specifies the number of periods of correlation and only needs to be specified when `corstr` is `"stat_M_dep"`, `"non_stat_M_dep"`, or `"AR-M"`.

- `R`: defaults to `NULL`. It specifies a user-defined correlation matrix rather than estimating it from the data. The argument is used only when `corstr` is `"fixed"`. The input is a $T \times T$ matrix of correlations, where T is the size of the largest cluster.

2.0.9 Examples

1. Example with Stationary 3 Dependence

   Attaching the sample turnout dataset:

   ```
   > data(turnout)
   ```

   Variable identifying clusters

   ```
   > turnout$cluster <- rep(c(1:200),10)
   ```

   Sorting by cluster

   ```
   > sorted.turnout <- turnout[order(turnout$cluster),]
   ```

   Estimating parameter values for the logistic regression:

   ```
   > z.out1 <- zelig(vote ~ race + educate, model = "logit.gee", id = "cluster", data = so

   (Intercept)   racewhite     educate
    -1.2189037   0.5022257   0.1610007


    How to cite this model in Zelig:
     Patrick Lam. 2012.
     "logit.gee: General Estimating Equation for Logistic Regression"
     in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
     http://gking.harvard.edu/zelig
   ```

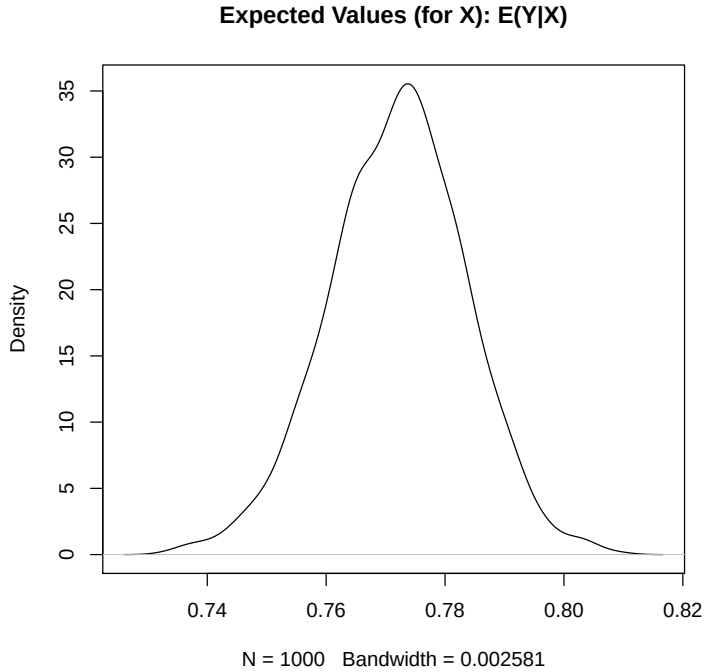
   Setting values for the explanatory variables to their default values:

   ```
   > x.out1 <- setx(z.out1)
   ```

   Simulating quantities of interest:

   ```
   > s.out1 <- sim(z.out1, x = x.out1)

   > summary(s.out1)
   ```

```
Model:  logit.gee
Number of simulations:  1000

Values of X
  (Intercept) racewhite  educate
1           1         1 12.06675
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$race
[1] "contr.treatment"


Expected Values (for x): E(Y|X)
  mean    sd   50%  2.5% 97.5%
 0.773 0.012 0.773 0.751 0.796

> plot(s.out1)
```



2. Simulating First Differences

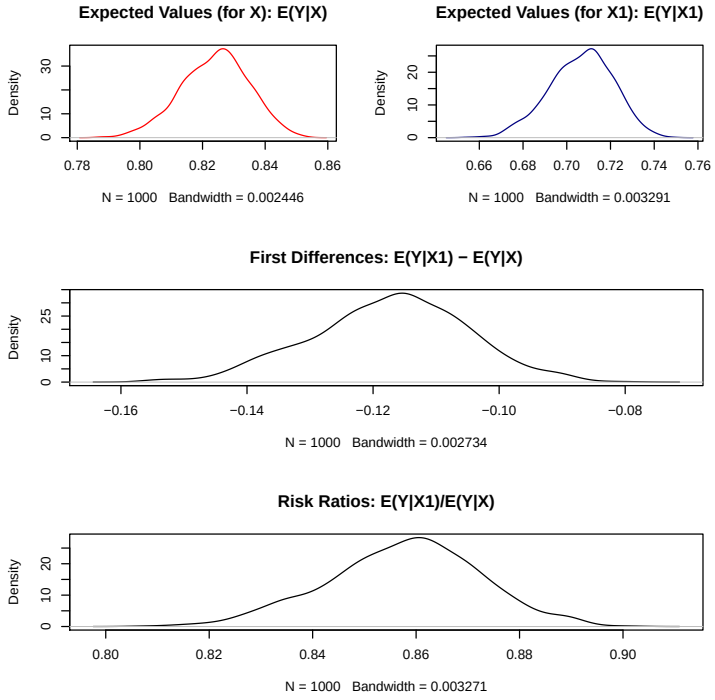
   Estimating the risk difference (and risk ratio) between low education (25th

percentile) and high education (75th percentile) while all the other variables held at their default values.

```
> x.high <- setx(z.out1, educate = quantile(turnout$educate, prob = 0.75))
> x.low <- setx(z.out1, educate = quantile(turnout$educate, prob = 0.25))

> s.out2 <- sim(z.out1, x = x.high, x1 = x.low)

> summary(s.out2)

Model:  logit.gee
Number of simulations:  1000

Values of X
  (Intercept) racewhite educate
1           1         1      14
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$race
[1] "contr.treatment"


Values of X1
  (Intercept) racewhite educate
1           1         1      10
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$race
[1] "contr.treatment"


Expected Values (for x): E(Y|X)
  mean    sd   50%  2.5% 97.5%
 0.823 0.011 0.823 0.802 0.844

Expected Values (for x1): E(Y|X1)
 mean    sd   50%  2.5% 97.5%
 0.71 0.014 0.709 0.682 0.738

First Differences: E(Y|X1) - E(Y|X)
   mean    sd    50%   2.5% 97.5%
 -0.113 0.011 -0.112 -0.137 -0.09

Risk Ratios: E(Y|X1)/E(Y|X)
```

```
  mean    sd   50%  2.5% 97.5%
 0.863 0.014 0.863 0.835 0.889
```

```
> plot(s.out2)
```



3. Example with Fixed Correlation Structure

   User-defined correlation structure

```
> corr.mat <- matrix(rep(0.5,100), nrow=10, ncol=10)
> diag(corr.mat) <- 1
```

   Generating empirical estimates:

```
> z.out2 <- zelig(vote ~ race + educate, model = "logit.gee", id = "cluster", data = so
```

```
(Intercept)   racewhite     educate
 -1.2189037   0.5022257   0.1610007
```

```
How to cite this model in Zelig:
 Patrick Lam. 2012.
 "logit.gee: General Estimating Equation for Logistic Regression"
 in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
 http://gking.harvard.edu/zelig
```

Viewing the regression output:

```
> summary(z.out2)

 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Logit
 Variance to Mean Relation: Binomial
 Correlation Structure:     Fixed

Call:
gee(formula = vote ~ race + educate, id = INTEGER2000, data = Data.frame,
    R = corr.mat, family = structure(list(family = "binomial",
        link = "logit", linkfun = function (mu)
        .Call(C_logit_link, mu), linkinv = function (eta)
        .Call(C_logit_linkinv, eta), variance = function (mu)
        mu * (1 - mu), dev.resids = function (y, mu, wt)
        .Call(C_binomial_dev_resids, y, mu, wt), aic = function (y,
            n, mu, wt, dev)
        {
            m <- if (any(n > 1))
                n
            else wt
            -2 * sum(ifelse(m > 0, (wt/m), 0) * dbinom(round(m *
                y), round(m), mu, log = TRUE))
        }, mu.eta = function (eta)
        .Call(C_logit_mu_eta, eta), initialize = expression({
            if (NCOL(y) == 1) {
                if (is.factor(y))
                  y <- y != levels(y)[1L]
                n <- rep.int(1, nobs)
                y[weights == 0] <- 0
                if (any(y < 0 | y > 1))
                  stop("y values must be 0 <= y <= 1")
                mustart <- (weights * y + 0.5)/(weights + 1)
                m <- weights * y
                if (any(abs(m - round(m)) > 0.001))
                  warning("non-integer #successes in a binomial glm!")
            }
            else if (NCOL(y) == 2) {
                if (any(abs(y - round(y)) > 0.001))
                  warning("non-integer counts in a binomial glm!")
                n <- y[, 1] + y[, 2]
                y <- ifelse(n == 0, 0, y[, 1]/n)
```

```
            weights <- weights * n
            mustart <- (n * y + 0.5)/(n + 1)
        }
        else stop("for the binomial family, y must be a vector of 0 and 1's\n",
            "or a 2 column matrix where col 1 is no. successes and col 2 is no. fai
    }), validmu = function (mu)
    all(mu > 0) && all(mu < 1), valideta = function (eta)
    TRUE, simulate = function (object, nsim)
    {
        ftd <- fitted(object)
        n <- length(ftd)
        ntot <- n * nsim
        wts <- object$prior.weights
        if (any(wts%%1 != 0))
            stop("cannot simulate from non-integer prior.weights")
        if (!is.null(m <- object$model)) {
            y <- model.response(m)
            if (is.factor(y)) {
              yy <- factor(1 + rbinom(ntot, size = 1, prob = ftd),
                labels = levels(y))
              split(yy, rep(seq_len(nsim), each = n))
            }
            else if (is.matrix(y) && ncol(y) == 2) {
              yy <- vector("list", nsim)
              for (i in seq_len(nsim)) {
                Y <- rbinom(n, size = wts, prob = ftd)
                YY <- cbind(Y, wts - Y)
                colnames(YY) <- colnames(y)
                yy[[i]] <- YY
              }
              yy
            }
            else rbinom(ntot, size = wts, prob = ftd)/wts
        }
        else rbinom(ntot, size = wts, prob = ftd)/wts
    }), .Names = c("family", "link", "linkfun", "linkinv",
"variance", "dev.resids", "aic", "mu.eta", "initialize",
"validmu", "valideta", "simulate"), class = "family"), corstr = "fixed")

Summary of Residuals:
       Min         1Q     Median         3Q        Max
-0.9067826 -0.3018991  0.2112738  0.2390951  0.7887027


Coefficients:
             Estimate Naive S.E.   Naive z Robust S.E.  Robust z
```

```
(Intercept) -1.3171233 0.16878117 -7.803733  0.22423028 -5.873976
racewhite    0.5593612 0.10125663  5.524193  0.14572484  3.838475
educate      0.1596174 0.01220733 13.075544  0.01657297  9.631189

Estimated Scale Parameter:  0.9574347
Number of Iterations:  3

Working Correlation
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]  1.0  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5   0.5
 [2,]  0.5  1.0  0.5  0.5  0.5  0.5  0.5  0.5  0.5   0.5
 [3,]  0.5  0.5  1.0  0.5  0.5  0.5  0.5  0.5  0.5   0.5
 [4,]  0.5  0.5  0.5  1.0  0.5  0.5  0.5  0.5  0.5   0.5
 [5,]  0.5  0.5  0.5  0.5  1.0  0.5  0.5  0.5  0.5   0.5
 [6,]  0.5  0.5  0.5  0.5  0.5  1.0  0.5  0.5  0.5   0.5
 [7,]  0.5  0.5  0.5  0.5  0.5  0.5  1.0  0.5  0.5   0.5
 [8,]  0.5  0.5  0.5  0.5  0.5  0.5  0.5  1.0  0.5   0.5
 [9,]  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  1.0   0.5
[10,]  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5   1.0
```

2.0.10 The Model

Suppose we have a panel dataset, with Y_{it} denoting the binary dependent variable for unit i at time t. Y_i is a vector or cluster of correlated data where y_{it} is correlated with $y_{it'}$ for some or all t, t'. Note that the model assumes correlations within i but independence across i.

- The *stochastic component* is given by the joint and marginal distributions

$$\begin{aligned} Y_i &\sim f(y_i \mid \pi_i) \\ Y_{it} &\sim g(y_{it} \mid \pi_{it}) \end{aligned}$$

  where f and g are unspecified distributions with means π_i and π_{it}. GEE models make no distributional assumptions and only require three specifications: a mean function, a variance function, and a correlation structure.

- The *systematic component* is the *mean function*, given by:

$$\pi_{it} = \frac{1}{1 + \exp(-x_{it}\beta)}$$

  where x_{it} is the vector of k explanatory variables for unit i at time t and β is the vector of coefficients.

- The *variance function* is given by:

$$V_{it} = \pi_{it}(1 - \pi_{it})$$

- The *correlation structure* is defined by a $T \times T$ "working" correlation matrix, where T is the size of the largest cluster. Users must specify the structure of the "working" correlation matrix *a priori*. The "working" correlation matrix then enters the variance term for each i, given by:

$$V_i = \phi\, A_i^{\frac{1}{2}} R_i(\alpha) A_i^{\frac{1}{2}}$$

  where A_i is a $T \times T$ diagonal matrix with the variance function $V_{it} = \pi_{it}(1-\pi_{it})$ as the tth diagonal element, $R_i(\alpha)$ is the "working" correlation matrix, and ϕ is a scale parameter. The parameters are then estimated via a quasi-likelihood approach.

- In GEE models, if the mean is correctly specified, but the variance and correlation structure are incorrectly specified, then GEE models provide consistent estimates of the parameters and thus the mean function as well, while consistent estimates of the standard errors can be obtained via a robust "sandwich" estimator. Similarly, if the mean and variance are correctly specified but the correlation structure is incorrectly specified, the parameters can be estimated consistently and the standard errors can be estimated consistently with the sandwich estimator. If all three are specified correctly, then the estimates of the parameters are more efficient.

- The robust "sandwich" estimator gives consistent estimates of the standard errors when the correlations are specified incorrectly only if the number of units i is relatively large and the number of repeated periods t is relatively small. Otherwise, one should use the "naïve" model-based standard errors, which assume that the specified correlations are close approximations to the true underlying correlations. See **?**] for more details.

2.0.11 Quantities of Interest

- All quantities of interest are for marginal means rather than joint means.

- The method of bootstrapping generally should not be used in GEE models. If you must bootstrap, bootstrapping should be done within clusters, which is not currently supported in Zelig. For conditional prediction models, data should be matched within clusters.

- The expected values (`qi$ev`) for the GEE logit model are simulations of the predicted probability of a success:

$$E(Y) = \pi_c = \frac{1}{1+\exp(-x_c\beta)},$$

  given draws of β from its sampling distribution, where x_c is a vector of values, one for each independent variable, chosen by the user.

- The first difference (`qi$fd`) for the GEE logit model is defined as

$$\text{FD} = \Pr(Y = 1 \mid x_1) - \Pr(Y = 1 \mid x).$$

- The risk ratio (`qi$rr`) is defined as

$$\text{RR} = \Pr(Y = 1 \mid x_1) \;/\; \Pr(Y = 1 \mid x).$$

- In conditional prediction models, the average expected treatment effect (`att.ev`) for the treatment group is

$$\frac{1}{\sum_{i=1}^{n}\sum_{t=1}^{T} tr_{it}} \sum_{i:tr_{it}=1}^{n} \sum_{t:tr_{it}=1}^{T} \left\{ Y_{it}(tr_{it}=1) - E[Y_{it}(tr_{it}=0)] \right\},$$

  where tr_{it} is a binary explanatory variable defining the treatment ($tr_{it} = 1$) and control ($tr_{it} = 0$) groups. Variation in the simulations are due to uncertainty in simulating $E[Y_{it}(tr_{it} = 0)]$, the counterfactual expected value of Y_{it} for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $tr_{it} = 0$.

2.0.12 Output Values

The output of each Zelig command contains useful information which you may view. For example, if you run `z.out <- zelig(y ~ x, model = "logit.gee", id, data)`, then you may examine the available information in `z.out` by using `names(z.out)`, see the `coefficients` by using `z.out$coefficients`, and a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output object `z.out`, you may extract:
  - `coefficients`: parameter estimates for the explanatory variables.
  - `residuals`: the working residuals in the final iteration of the fit.
  - `fitted.values`: the vector of fitted values for the systemic component, π_{it}.
  - `linear.predictors`: the vector of $x_{it}\beta$
  - `max.id`: the size of the largest cluster.
- From `summary(z.out)`, you may extract:
  - `coefficients`: the parameter estimates with their associated standard errors, p-values, and z-statistics.
  - `working.correlation`: the "working" correlation matrix
- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as matrices indexed by simulation $\times$ `x`-observation (for more than one `x`-observation). Available quantities are:
  - `qi$ev`: the simulated expected probabilities for the specified values of `x`.

- `qi$fd`: the simulated first difference in the expected probabilities for the values specified in `x` and `x1`.
- `qi$rr`: the simulated risk ratio for the expected probabilities simulated from `x` and `x1`.
- `qi$att.ev`: the simulated average expected treatment effect for the treated from conditional prediction models.

How To Cite the *logit.gee* Zelig Model

Patrick Lam. 2007. "logit.gee: Generalized Estimating Equation for Logit Regression," in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software," `http://gking.harvard.edu/zelig`.

How to Cite the Zelig Software Package

To cite Zelig as a whole, please reference these two sources:

Kosuke Imai, Gary King, and Olivia Lau. 2007. "Zelig: Everyone's Statistical Software," `http://GKing.harvard.edu/zelig`.

Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The gee function is part of the gee package by Vincent J. Carey, ported to R by Thomas Lumley and Brian Ripley. Advanced users may wish to refer to `help(gee)` and `help(family)`. Sample data are from [2].

3 `normal.gee`: Generalized Estimating Equation for Normal Regression

The GEE normal estimates the same model as the standard normal regression. Unlike in normal regression, GEE normal allows for dependence within clusters, such as in longitudinal data, although its use is not limited to just panel data. The user must first specify a "working" correlation matrix for the clusters, which models the dependence of each observation with other observations in the same cluster. The "working" correlation matrix is a $T \times T$ matrix of correlations, where T is the size of the largest cluster and the elements of the matrix are correlations between within-cluster observations. The appeal of GEE models is that it gives consistent estimates of the parameters and consistent estimates of the standard errors can be obtained using a robust "sandwich" estimator even

if the "working" correlation matrix is incorrectly specified. If the "working" correlation matrix is correctly specified, GEE models will give more efficient estimates of the parameters. GEE models measure population-averaged effects as opposed to cluster-specific effects (See **(author?)** [4]).

3.0.13 Syntax

```
> z.out <- zelig(Y ~ X1 + X2, model = "normal.gee",
                 id = "X3", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

where `id` is a variable which identifies the clusters. The data should be sorted by `id` and should be ordered within each cluster when appropriate.

3.0.14 Additional Inputs

- `robust`: defaults to `TRUE`. If `TRUE`, consistent standard errors are estimated using a "sandwich" estimator.

Use the following arguments to specify the structure of the "working" correlations within clusters:

- `corstr`: defaults to `"independence"`. It can take on the following arguments:
  - Independence (`corstr = "independence"`): $\text{cor}(y_{it}, y_{it'}) = 0$, $\forall t, t'$ with $t \neq t'$. It assumes that there is no correlation within the clusters and the model becomes equivalent to standard normal regression. The "working" correlation matrix is the identity matrix.
  - Fixed (`corstr = "fixed"`): If selected, the user must define the "working" correlation matrix with the `R` argument rather than estimating it from the model.
  - Stationary m dependent (`corstr = "stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{|t-t'|} & \text{if} \quad |t-t'| \leq m \\ 0 & \text{if} \quad |t-t'| > m \end{cases}$$

    If (`corstr = "stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. Choose this option when the correlations are assumed to be the same for observations of the same $|t-t'|$ periods apart for $|t-t'| \leq m$.

    Sample "working" correlation for Stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_1 & \alpha_2 & 0 & 0 \\ \alpha_1 & 1 & \alpha_1 & \alpha_2 & 0 \\ \alpha_2 & \alpha_1 & 1 & \alpha_1 & \alpha_2 \\ 0 & \alpha_2 & \alpha_1 & 1 & \alpha_1 \\ 0 & 0 & \alpha_2 & \alpha_1 & 1 \end{pmatrix}$$

– Non-stationary m dependent (`corstr = "non_stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{tt'} & \text{if} \quad |t - t'| \leq m \\ 0 & \text{if} \quad |t - t'| > m \end{cases}$$

If (`corstr = "non_stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. This option relaxes the assumption that the correlations are the same for all observations of the same $|t - t'|$ periods apart.

Sample "working" correlation for Non-stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_{12} & \alpha_{13} & 0 & 0 \\ \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} & 0 \\ \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} & \alpha_{35} \\ 0 & \alpha_{24} & \alpha_{34} & 1 & \alpha_{45} \\ 0 & 0 & \alpha_{35} & \alpha_{45} & 1 \end{pmatrix}$$

– Exchangeable (`corstr = "exchangeable"`): $\text{cor}(y_{it}, y_{it'}) = \alpha$, $\forall t, t'$ with $t \neq t'$. Choose this option if the correlations are assumed to be the same for all observations within the cluster.

Sample "working" correlation for Exchangeable

$$\begin{pmatrix} 1 & \alpha & \alpha & \alpha & \alpha \\ \alpha & 1 & \alpha & \alpha & \alpha \\ \alpha & \alpha & 1 & \alpha & \alpha \\ \alpha & \alpha & \alpha & 1 & \alpha \\ \alpha & \alpha & \alpha & \alpha & 1 \end{pmatrix}$$

– Stationary mth order autoregressive (`corstr = "AR-M"`): If (`corstr = "AR-M"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. For example, the first order autoregressive model (AR-1) implies $\text{cor}(y_{it}, y_{it'}) = \alpha^{|t-t'|}, \forall t, t'$ with $t \neq t'$. In AR-1, observation 1 and observation 2 have a correlation of α. Observation 2 and observation 3 also have a correlation of α. Observation 1 and observation 3 have a correlation of α^2, which is a function of how 1 and 2 are correlated (α) multiplied by how 2 and 3 are correlated (α). Observation 1 and 4 have a correlation that is a function of the correlation between 1 and 2, 2 and 3, and 3 and 4, and so forth.

Sample "working" correlation for Stationary AR-1 (`Mv`=1)

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ \alpha & 1 & \alpha & \alpha^2 & \alpha^3 \\ \alpha^2 & \alpha & 1 & \alpha & \alpha^2 \\ \alpha^3 & \alpha^2 & \alpha & 1 & \alpha \\ \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \end{pmatrix}$$

  - Unstructured (`corstr = "unstructured"`): $\text{cor}(y_{it}, y_{it'}) = \alpha_{tt'}, \forall t, t'$ with $t \neq t'$. No constraints are placed on the correlations, which are then estimated from the data.

- `Mv`: defaults to 1. It specifies the number of periods of correlation and only needs to be specified when `corstr` is `"stat_M_dep"`, `"non_stat_M_dep"`, or `"AR-M"`.

- `R`: defaults to `NULL`. It specifies a user-defined correlation matrix rather than estimating it from the data. The argument is used only when `corstr` is `"fixed"`. The input is a $T \times T$ matrix of correlations, where T is the size of the largest cluster.

3.0.15 Examples

1. Example with AR-1 Dependence

   Attaching the sample turnout dataset:

```
> data(macro)
```

   Estimating model and presenting summary:

```
> z.out <- zelig(unem ~ gdp + capmob + trade, model = "normal.gee", id = "country", dat

(Intercept)         gdp      capmob       trade
 6.18129445 -0.32360059  1.42193926  0.01985421


 How to cite this model in Zelig:
  Patrick Lam. 2012.
  "normal.gee: General Estimating Equation for Normal Regression"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig

> summary(z.out)

 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Identity
 Variance to Mean Relation: Gaussian
 Correlation Structure:     AR-M , M = 1

Call:
gee(formula = unem ~ gdp + capmob + trade, id = Factor, data = Data.frame,
    family = structure(list(family = "gaussian", link = "identity",
        linkfun = function (mu)
```

```
        mu, linkinv = function (eta)
        eta, variance = function (mu)
        rep.int(1, length(mu)), dev.resids = function (y, mu,
            wt)
        wt * ((y - mu)^2), aic = function (y, n, mu, wt, dev)
        {
            nobs <- length(y)
            nobs * (log(dev/nobs * 2 * pi) + 1) + 2 - sum(log(wt))
        }, mu.eta = function (eta)
        rep.int(1, length(eta)), initialize = expression({
            n <- rep.int(1, nobs)
            if (is.null(etastart) && is.null(start) && is.null(mustart) &&
                ((family$link == "inverse" && any(y == 0)) ||
                  (family$link == "log" && any(y <= 0))))
                stop("cannot find valid starting values: please specify some")
            mustart <- y
        }), validmu = function (mu)
        TRUE, valideta = function (eta)
        TRUE), .Names = c("family", "link", "linkfun", "linkinv",
    "variance", "dev.resids", "aic", "mu.eta", "initialize",
    "validmu", "valideta"), class = "family"), corstr = "AR-M",
    Mv = 1)

Summary of Residuals:
       Min         1Q     Median         3Q        Max
-3.8177474 -1.9082221  0.1124818  2.8287233  8.0261275


Coefficients:
              Estimate  Naive S.E.   Naive z Robust S.E.  Robust z
(Intercept)  3.96818567 0.912532716  4.348541  0.77474335  5.121936
gdp         -0.06553926 0.016222536 -4.040013  0.01737177 -3.772744
capmob       0.33382352 0.125003704  2.670509  0.16212556  2.059043
trade        0.01616045 0.009822323  1.645278  0.01102485  1.465820

Estimated Scale Parameter:  9.512685
Number of Iterations:  4

Working Correlation
           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
 [1,] 1.0000000 0.9701417 0.9411749 0.9130731 0.8858103 0.8593615 0.8337024
 [2,] 0.9701417 1.0000000 0.9701417 0.9411749 0.9130731 0.8858103 0.8593615
 [3,] 0.9411749 0.9701417 1.0000000 0.9701417 0.9411749 0.9130731 0.8858103
 [4,] 0.9130731 0.9411749 0.9701417 1.0000000 0.9701417 0.9411749 0.9130731
 [5,] 0.8858103 0.9130731 0.9411749 0.9701417 1.0000000 0.9701417 0.9411749
 [6,] 0.8593615 0.8858103 0.9130731 0.9411749 0.9701417 1.0000000 0.9701417
```

```
 [7,] 0.8337024 0.8593615 0.8858103 0.9130731 0.9411749 0.9701417 1.0000000
 [8,] 0.8088095 0.8337024 0.8593615 0.8858103 0.9130731 0.9411749 0.9701417
 [9,] 0.7846598 0.8088095 0.8337024 0.8593615 0.8858103 0.9130731 0.9411749
[10,] 0.7612312 0.7846598 0.8088095 0.8337024 0.8593615 0.8858103 0.9130731
[11,] 0.7385021 0.7612312 0.7846598 0.8088095 0.8337024 0.8593615 0.8858103
[12,] 0.7164517 0.7385021 0.7612312 0.7846598 0.8088095 0.8337024 0.8593615
[13,] 0.6950597 0.7164517 0.7385021 0.7612312 0.7846598 0.8088095 0.8337024
[14,] 0.6743064 0.6950597 0.7164517 0.7385021 0.7612312 0.7846598 0.8088095
[15,] 0.6541728 0.6743064 0.6950597 0.7164517 0.7385021 0.7612312 0.7846598
[16,] 0.6346403 0.6541728 0.6743064 0.6950597 0.7164517 0.7385021 0.7612312
[17,] 0.6156910 0.6346403 0.6541728 0.6743064 0.6950597 0.7164517 0.7385021
[18,] 0.5973075 0.6156910 0.6346403 0.6541728 0.6743064 0.6950597 0.7164517
[19,] 0.5794730 0.5973075 0.6156910 0.6346403 0.6541728 0.6743064 0.6950597
[20,] 0.5621709 0.5794730 0.5973075 0.6156910 0.6346403 0.6541728 0.6743064
[21,] 0.5453854 0.5621709 0.5794730 0.5973075 0.6156910 0.6346403 0.6541728
[22,] 0.5291011 0.5453854 0.5621709 0.5794730 0.5973075 0.6156910 0.6346403
[23,] 0.5133031 0.5291011 0.5453854 0.5621709 0.5794730 0.5973075 0.6156910
[24,] 0.4979767 0.5133031 0.5291011 0.5453854 0.5621709 0.5794730 0.5973075
[25,] 0.4831080 0.4979767 0.5133031 0.5291011 0.5453854 0.5621709 0.5794730
          [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
 [1,] 0.8088095 0.7846598 0.7612312 0.7385021 0.7164517 0.6950597 0.6743064
 [2,] 0.8337024 0.8088095 0.7846598 0.7612312 0.7385021 0.7164517 0.6950597
 [3,] 0.8593615 0.8337024 0.8088095 0.7846598 0.7612312 0.7385021 0.7164517
 [4,] 0.8858103 0.8593615 0.8337024 0.8088095 0.7846598 0.7612312 0.7385021
 [5,] 0.9130731 0.8858103 0.8593615 0.8337024 0.8088095 0.7846598 0.7612312
 [6,] 0.9411749 0.9130731 0.8858103 0.8593615 0.8337024 0.8088095 0.7846598
 [7,] 0.9701417 0.9411749 0.9130731 0.8858103 0.8593615 0.8337024 0.8088095
 [8,] 1.0000000 0.9701417 0.9411749 0.9130731 0.8858103 0.8593615 0.8337024
 [9,] 0.9701417 1.0000000 0.9701417 0.9411749 0.9130731 0.8858103 0.8593615
[10,] 0.9411749 0.9701417 1.0000000 0.9701417 0.9411749 0.9130731 0.8858103
[11,] 0.9130731 0.9411749 0.9701417 1.0000000 0.9701417 0.9411749 0.9130731
[12,] 0.8858103 0.9130731 0.9411749 0.9701417 1.0000000 0.9701417 0.9411749
[13,] 0.8593615 0.8858103 0.9130731 0.9411749 0.9701417 1.0000000 0.9701417
[14,] 0.8337024 0.8593615 0.8858103 0.9130731 0.9411749 0.9701417 1.0000000
[15,] 0.8088095 0.8337024 0.8593615 0.8858103 0.9130731 0.9411749 0.9701417
[16,] 0.7846598 0.8088095 0.8337024 0.8593615 0.8858103 0.9130731 0.9411749
[17,] 0.7612312 0.7846598 0.8088095 0.8337024 0.8593615 0.8858103 0.9130731
[18,] 0.7385021 0.7612312 0.7846598 0.8088095 0.8337024 0.8593615 0.8858103
[19,] 0.7164517 0.7385021 0.7612312 0.7846598 0.8088095 0.8337024 0.8593615
[20,] 0.6950597 0.7164517 0.7385021 0.7612312 0.7846598 0.8088095 0.8337024
[21,] 0.6743064 0.6950597 0.7164517 0.7385021 0.7612312 0.7846598 0.8088095
[22,] 0.6541728 0.6743064 0.6950597 0.7164517 0.7385021 0.7612312 0.7846598
[23,] 0.6346403 0.6541728 0.6743064 0.6950597 0.7164517 0.7385021 0.7612312
[24,] 0.6156910 0.6346403 0.6541728 0.6743064 0.6950597 0.7164517 0.7385021
[25,] 0.5973075 0.6156910 0.6346403 0.6541728 0.6743064 0.6950597 0.7164517
          [,15]     [,16]     [,17]     [,18]     [,19]     [,20]     [,21]
```

```
 [1,] 0.6541728 0.6346403 0.6156910 0.5973075 0.5794730 0.5621709 0.5453854
 [2,] 0.6743064 0.6541728 0.6346403 0.6156910 0.5973075 0.5794730 0.5621709
 [3,] 0.6950597 0.6743064 0.6541728 0.6346403 0.6156910 0.5973075 0.5794730
 [4,] 0.7164517 0.6950597 0.6743064 0.6541728 0.6346403 0.6156910 0.5973075
 [5,] 0.7385021 0.7164517 0.6950597 0.6743064 0.6541728 0.6346403 0.6156910
 [6,] 0.7612312 0.7385021 0.7164517 0.6950597 0.6743064 0.6541728 0.6346403
 [7,] 0.7846598 0.7612312 0.7385021 0.7164517 0.6950597 0.6743064 0.6541728
 [8,] 0.8088095 0.7846598 0.7612312 0.7385021 0.7164517 0.6950597 0.6743064
 [9,] 0.8337024 0.8088095 0.7846598 0.7612312 0.7385021 0.7164517 0.6950597
[10,] 0.8593615 0.8337024 0.8088095 0.7846598 0.7612312 0.7385021 0.7164517
[11,] 0.8858103 0.8593615 0.8337024 0.8088095 0.7846598 0.7612312 0.7385021
[12,] 0.9130731 0.8858103 0.8593615 0.8337024 0.8088095 0.7846598 0.7612312
[13,] 0.9411749 0.9130731 0.8858103 0.8593615 0.8337024 0.8088095 0.7846598
[14,] 0.9701417 0.9411749 0.9130731 0.8858103 0.8593615 0.8337024 0.8088095
[15,] 1.0000000 0.9701417 0.9411749 0.9130731 0.8858103 0.8593615 0.8337024
[16,] 0.9701417 1.0000000 0.9701417 0.9411749 0.9130731 0.8858103 0.8593615
[17,] 0.9411749 0.9701417 1.0000000 0.9701417 0.9411749 0.9130731 0.8858103
[18,] 0.9130731 0.9411749 0.9701417 1.0000000 0.9701417 0.9411749 0.9130731
[19,] 0.8858103 0.9130731 0.9411749 0.9701417 1.0000000 0.9701417 0.9411749
[20,] 0.8593615 0.8858103 0.9130731 0.9411749 0.9701417 1.0000000 0.9701417
[21,] 0.8337024 0.8593615 0.8858103 0.9130731 0.9411749 0.9701417 1.0000000
[22,] 0.8088095 0.8337024 0.8593615 0.8858103 0.9130731 0.9411749 0.9701417
[23,] 0.7846598 0.8088095 0.8337024 0.8593615 0.8858103 0.9130731 0.9411749
[24,] 0.7612312 0.7846598 0.8088095 0.8337024 0.8593615 0.8858103 0.9130731
[25,] 0.7385021 0.7612312 0.7846598 0.8088095 0.8337024 0.8593615 0.8858103
          [,22]     [,23]     [,24]     [,25]
 [1,] 0.5291011 0.5133031 0.4979767 0.4831080
 [2,] 0.5453854 0.5291011 0.5133031 0.4979767
 [3,] 0.5621709 0.5453854 0.5291011 0.5133031
 [4,] 0.5794730 0.5621709 0.5453854 0.5291011
 [5,] 0.5973075 0.5794730 0.5621709 0.5453854
 [6,] 0.6156910 0.5973075 0.5794730 0.5621709
 [7,] 0.6346403 0.6156910 0.5973075 0.5794730
 [8,] 0.6541728 0.6346403 0.6156910 0.5973075
 [9,] 0.6743064 0.6541728 0.6346403 0.6156910
[10,] 0.6950597 0.6743064 0.6541728 0.6346403
[11,] 0.7164517 0.6950597 0.6743064 0.6541728
[12,] 0.7385021 0.7164517 0.6950597 0.6743064
[13,] 0.7612312 0.7385021 0.7164517 0.6950597
[14,] 0.7846598 0.7612312 0.7385021 0.7164517
[15,] 0.8088095 0.7846598 0.7612312 0.7385021
[16,] 0.8337024 0.8088095 0.7846598 0.7612312
[17,] 0.8593615 0.8337024 0.8088095 0.7846598
[18,] 0.8858103 0.8593615 0.8337024 0.8088095
[19,] 0.9130731 0.8858103 0.8593615 0.8337024
[20,] 0.9411749 0.9130731 0.8858103 0.8593615
```

```
[21,] 0.9701417 0.9411749 0.9130731 0.8858103
[22,] 1.0000000 0.9701417 0.9411749 0.9130731
[23,] 0.9701417 1.0000000 0.9701417 0.9411749
[24,] 0.9411749 0.9701417 1.0000000 0.9701417
[25,] 0.9130731 0.9411749 0.9701417 1.0000000
```

Set explanatory variables to their default (mean/mode) values, with high (80th percentile) and low (20th percentile) values:

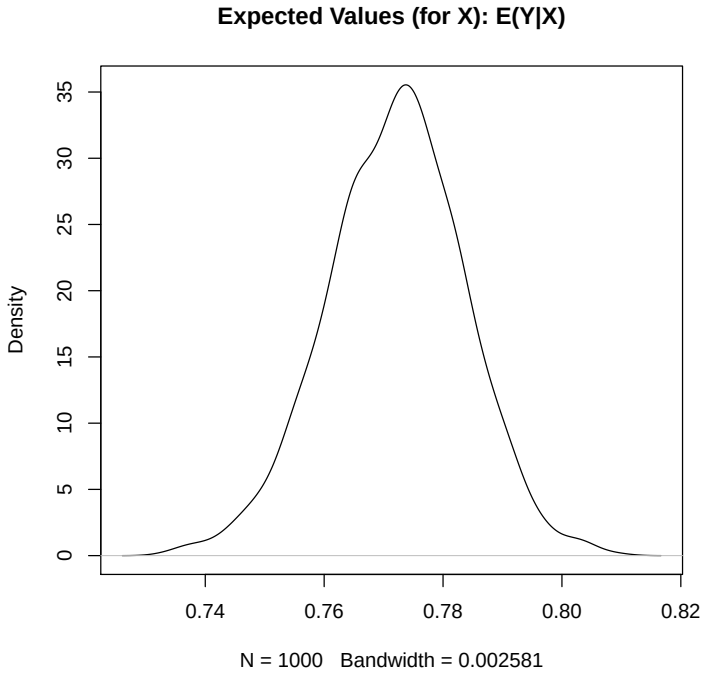
```
> x.high <- setx(z.out, trade = quantile(macro$trade, 0.8))
> x.low <- setx(z.out, trade = quantile(macro$trade, 0.2))
```

Generate first differences for the effect of high versus low trade on GDP:

```
> s.out <- sim(z.out, x = x.high, x1 = x.low)

> summary(s.out)

Model:  normal.gee
Number of simulations:  1000

Values of X
  (Intercept)      gdp     capmob    trade
1           1 3.254223 -0.8914286 79.10131
attr(,"assign")
[1] 0 1 2 3

Values of X1
  (Intercept)      gdp     capmob    trade
1           1 3.254223 -0.8914286 37.29106
attr(,"assign")
[1] 0 1 2 3

Expected Values (for x): E(Y|X)
  mean    sd   50%  2.5% 97.5%
 4.741 0.555 4.739 3.702 5.805

Expected Values (for x1): E(Y|X1)
  mean    sd   50%  2.5% 97.5%
 4.083 0.536 4.083 3.044 5.145

First Differences: E(Y|X1) - E(Y|X)
   mean    sd   50%   2.5% 97.5%
 -0.658 0.456 -0.67 -1.538 0.232
```

Generate a plot of quantities of interest:

```
> plot(s.out)
```



3.0.16 The Model

Suppose we have a panel dataset, with Y_{it} denoting the continuous dependent variable for unit i at time t. Y_i is a vector or cluster of correlated data where y_{it} is correlated with $y_{it'}$ for some or all t, t'. Note that the model assumes correlations within i but independence across i.

- The *stochastic component* is given by the joint and marginal distributions

$$\begin{aligned} Y_i &\sim f(y_i \mid \mu_i) \\ Y_{it} &\sim g(y_{it} \mid \mu_{it}) \end{aligned}$$

  where f and g are unspecified distributions with means μ_i and μ_{it}. GEE models make no distributional assumptions and only require three specifications: a mean function, a variance function, and a correlation structure.

- The *systematic component* is the *mean function*, given by:

$$\mu_{it} = x_{it}\beta$$

  where x_{it} is the vector of k explanatory variables for unit i at time t and β is the vector of coefficients.

- The *variance function* is given by:

$$V_{it} = 1$$

- The *correlation structure* is defined by a $T \times T$ "working" correlation matrix, where T is the size of the largest cluster. Users must specify the structure of the "working" correlation matrix *a priori*. The "working" correlation matrix then enters the variance term for each i, given by:

$$V_i = \phi \, A_i^{\frac{1}{2}} R_i(\alpha) A_i^{\frac{1}{2}}$$

  where A_i is a $T \times T$ diagonal matrix with the variance function $V_{it} = 1$ as the tth diagonal element (in the case of GEE normal, A_i is the identity matrix), $R_i(\alpha)$ is the "working" correlation matrix, and ϕ is a scale parameter. The parameters are then estimated via a quasi-likelihood approach.

- In GEE models, if the mean is correctly specified, but the variance and correlation structure are incorrectly specified, then GEE models provide consistent estimates of the parameters and thus the mean function as well, while consistent estimates of the standard errors can be obtained via a robust "sandwich" estimator. Similarly, if the mean and variance are correctly specified but the correlation structure is incorrectly specified, the parameters can be estimated consistently and the standard errors can be estimated consistently with the sandwich estimator. If all three are specified correctly, then the estimates of the parameters are more efficient.

- The robust "sandwich" estimator gives consistent estimates of the standard errors when the correlations are specified incorrectly only if the number of units i is relatively large and the number of repeated periods t is relatively small. Otherwise, one should use the "naïve" model-based standard errors, which assume that the specified correlations are close approximations to the true underlying correlations. See **?**] for more details.

3.0.17 Quantities of Interest

- All quantities of interest are for marginal means rather than joint means.

- The method of bootstrapping generally should not be used in GEE models. If you must bootstrap, bootstrapping should be done within clusters, which is not currently supported in Zelig. For conditional prediction models, data should be matched within clusters.

- The expected values (`qi$ev`) for the GEE normal model is the mean of simulations from the stochastic component:

$$E(Y) = \mu_c = x_c \beta,$$

  given draws of β from its sampling distribution, where x_c is a vector of values, one for each independent variable, chosen by the user.

- The first difference (`qi$fd`) for the GEE normal model is defined as

$$\text{FD} = \Pr(Y = 1 \mid x_1) - \Pr(Y = 1 \mid x).$$

- In conditional prediction models, the average expected treatment effect (`att.ev`) for the treatment group is

$$\frac{1}{\sum_{i=1}^{n}\sum_{t=1}^{T} tr_{it}} \sum_{i:tr_{it}=1}^{n} \sum_{t:tr_{it}=1}^{T} \left\{ Y_{it}(tr_{it} = 1) - E[Y_{it}(tr_{it} = 0)] \right\},$$

  where tr_{it} is a binary explanatory variable defining the treatment ($tr_{it} = 1$) and control ($tr_{it} = 0$) groups. Variation in the simulations are due to uncertainty in simulating $E[Y_{it}(tr_{it} = 0)]$, the counterfactual expected value of Y_{it} for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $tr_{it} = 0$.

3.0.18 Output Values

The output of each Zelig command contains useful information which you may view. For example, if you run `z.out <- zelig(y ~ x, model = "normal.gee", id, data)`, then you may examine the available information in `z.out` by using `names(z.out)`, see the `coefficients` by using `z.out$coefficients`, and a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output object `z.out`, you may extract:
  - `coefficients`: parameter estimates for the explanatory variables.
  - `residuals`: the working residuals in the final iteration of the fit.
  - `fitted.values`: the vector of fitted values for the systemic component, μ_{it}.
  - `linear.predictors`: the vector of $x_{it}\beta$
  - `max.id`: the size of the largest cluster.
- From `summary(z.out)`, you may extract:
  - `coefficients`: the parameter estimates with their associated standard errors, p-values, and z-statistics.
  - `working.correlation`: the "working" correlation matrix
- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as matrices indexed by simulation $\times$ `x`-observation (for more than one `x`-observation). Available quantities are:
  - `qi$ev`: the simulated expected values for the specified values of `x`.

- `qi$fd`: the simulated first difference in the expected probabilities for the values specified in `x` and `x1`.
- `qi$att.ev`: the simulated average expected treatment effect for the treated from conditional prediction models.

How To Cite the *normal.gee* Zelig model

Patrick Lam. 2007. "normal.gee: Generalized Estimating Equation for Normal Regression," in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software," `http://gking.harvard.edu/zelig`.

How to Cite the Zelig Software Package

To cite Zelig as a whole, please reference these two sources:

Kosuke Imai, Gary King, and Olivia Lau. 2007. "Zelig: Everyone's Statistical Software," `http://GKing.harvard.edu/zelig`.

Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

4 `poisson.gee`: Generalized Estimating Equation for Poisson Regression

The GEE poisson estimates the same model as the standard poisson regression (appropriate when your dependent variable represents the number of independent events that occur during a fixed period of time). Unlike in poisson regression, GEE poisson allows for dependence within clusters, such as in longitudinal data, although its use is not limited to just panel data. The user must first specify a "working" correlation matrix for the clusters, which models the dependence of each observation with other observations in the same cluster. The "working" correlation matrix is a $T \times T$ matrix of correlations, where T is the size of the largest cluster and the elements of the matrix are correlations between within-cluster observations. The appeal of GEE models is that it gives consistent estimates of the parameters and consistent estimates of the standard errors can be obtained using a robust "sandwich" estimator even if the "working" correlation matrix is incorrectly specified. If the "working" correlation matrix is correctly specified, GEE models will give more efficient estimates of the parameters. GEE models measure population-averaged effects as opposed to cluster-specific effects (See **(author?)** [4]).

4.0.19 Syntax

```
> z.out <- zelig(Y ~ X1 + X2, model = "poisson.gee",
                 id = "X3", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

where `id` is a variable which identifies the clusters. The data should be sorted by `id` and should be ordered within each cluster when appropriate.

4.0.20 Additional Inputs

- `robust`: defaults to `TRUE`. If `TRUE`, consistent standard errors are estimated using a "sandwich" estimator.

Use the following arguments to specify the structure of the "working" correlations within clusters:

- `corstr`: defaults to `"independence"`. It can take on the following arguments:
  - Independence (`corstr = "independence"`): $\text{cor}(y_{it}, y_{it'}) = 0$, $\forall t, t'$ with $t \neq t'$. It assumes that there is no correlation within the clusters and the model becomes equivalent to standard poisson regression. The "working" correlation matrix is the identity matrix.
  - Fixed (`corstr = "fixed"`): If selected, the user must define the "working" correlation matrix with the `R` argument rather than estimating it from the model.
  - Stationary m dependent (`corstr = "stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{|t-t'|} & \text{if} \quad |t-t'| \leq m \\ 0 & \text{if} \quad |t-t'| > m \end{cases}$$

    If (`corstr = "stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. Choose this option when the correlations are assumed to be the same for observations of the same $|t-t'|$ periods apart for $|t-t'| \leq m$.

    Sample "working" correlation for Stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_1 & \alpha_2 & 0 & 0 \\ \alpha_1 & 1 & \alpha_1 & \alpha_2 & 0 \\ \alpha_2 & \alpha_1 & 1 & \alpha_1 & \alpha_2 \\ 0 & \alpha_2 & \alpha_1 & 1 & \alpha_1 \\ 0 & 0 & \alpha_2 & \alpha_1 & 1 \end{pmatrix}$$

– Non-stationary m dependent (`corstr = "non_stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{tt'} & \text{if} \quad |t - t'| \leq m \\ 0 & \text{if} \quad |t - t'| > m \end{cases}$$

If (`corstr = "non_stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. This option relaxes the assumption that the correlations are the same for all observations of the same $|t - t'|$ periods apart.

Sample "working" correlation for Non-stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_{12} & \alpha_{13} & 0 & 0 \\ \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} & 0 \\ \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} & \alpha_{35} \\ 0 & \alpha_{24} & \alpha_{34} & 1 & \alpha_{45} \\ 0 & 0 & \alpha_{35} & \alpha_{45} & 1 \end{pmatrix}$$

– Exchangeable (`corstr = "exchangeable"`): $\text{cor}(y_{it}, y_{it'}) = \alpha$, $\forall t, t'$ with $t \neq t'$. Choose this option if the correlations are assumed to be the same for all observations within the cluster.

Sample "working" correlation for Exchangeable

$$\begin{pmatrix} 1 & \alpha & \alpha & \alpha & \alpha \\ \alpha & 1 & \alpha & \alpha & \alpha \\ \alpha & \alpha & 1 & \alpha & \alpha \\ \alpha & \alpha & \alpha & 1 & \alpha \\ \alpha & \alpha & \alpha & \alpha & 1 \end{pmatrix}$$

– Stationary mth order autoregressive (`corstr = "AR-M"`): If (`corstr = "AR-M"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. For example, the first order autoregressive model (AR-1) implies $\text{cor}(y_{it}, y_{it'}) = \alpha^{|t-t'|}, \forall t, t'$ with $t \neq t'$. In AR-1, observation 1 and observation 2 have a correlation of α. Observation 2 and observation 3 also have a correlation of α. Observation 1 and observation 3 have a correlation of α^2, which is a function of how 1 and 2 are correlated (α) multiplied by how 2 and 3 are correlated (α). Observation 1 and 4 have a correlation that is a function of the correlation between 1 and 2, 2 and 3, and 3 and 4, and so forth.

Sample "working" correlation for Stationary AR-1 (`Mv`=1)

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ \alpha & 1 & \alpha & \alpha^2 & \alpha^3 \\ \alpha^2 & \alpha & 1 & \alpha & \alpha^2 \\ \alpha^3 & \alpha^2 & \alpha & 1 & \alpha \\ \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \end{pmatrix}$$

  - Unstructured (`corstr = "unstructured"`): $\text{cor}(y_{it}, y_{it'}) = \alpha_{tt'}, \forall t, t'$ with $t \neq t'$. No constraints are placed on the correlations, which are then estimated from the data.

- `Mv`: defaults to 1. It specifies the number of periods of correlation and only needs to be specified when `corstr` is `"stat_M_dep"`, `"non_stat_M_dep"`, or `"AR-M"`.

- `R`: defaults to `NULL`. It specifies a user-defined correlation matrix rather than estimating it from the data. The argument is used only when `corstr` is `"fixed"`. The input is a $T \times T$ matrix of correlations, where T is the size of the largest cluster.

4.0.21 Examples

1. Example with Exchangeable Dependence

   Attaching the sample turnout dataset:

   ```
   > data(sanction)
   ```

   Variable identifying clusters

   ```
   > sanction$cluster <- c(rep(c(1:15),5),rep(c(16),3))
   ```

   Sorting by cluster

   ```
   > sorted.sanction <- sanction[order(sanction$cluster),]
   ```

   Estimating model and presenting summary:

   ```
   > z.out <- zelig(num ~ target + coop, model = "poisson.gee", id = "cluster", data = sor

   The following object(s) are masked from 'package:MASS':

       coop
   (Intercept)      target        coop
   -0.96771994 -0.02102351  1.21081908


    How to cite this model in Zelig:
     Patrick Lam. 2012.
     "poisson.gee: General Estimating Equation for Poisson Regression"
     in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
     http://gking.harvard.edu/zelig

   > summary(z.out)
   ```

```
 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Logarithm
 Variance to Mean Relation: Poisson
 Correlation Structure:     Exchangeable

Call:
gee(formula = num ~ target + coop, id = NUMERIC78, data = Data.frame,
    family = structure(list(family = "poisson", link = "log",
        linkfun = function (mu)
        log(mu), linkinv = function (eta)
        pmax(exp(eta), .Machine$double.eps), variance = function (mu)
        mu, dev.resids = function (y, mu, wt)
        2 * wt * (y * log(ifelse(y == 0, 1, y/mu)) - (y - mu)),
        aic = function (y, n, mu, wt, dev)
        -2 * sum(dpois(y, mu, log = TRUE) * wt), mu.eta = function (eta)
        pmax(exp(eta), .Machine$double.eps), initialize = expression(
            {
                if (any(y < 0))
                  stop("negative values not allowed for the Poisson family")
                n <- rep.int(1, nobs)
                mustart <- y + 0.1
            }), validmu = function (mu)
        all(mu > 0), valideta = function (eta)
        TRUE, simulate = function (object, nsim)
        {
            wts <- object$prior.weights
            if (any(wts != 1))
                warning("ignoring prior weights")
            ftd <- fitted(object)
            rpois(nsim * length(ftd), ftd)
        }), .Names = c("family", "link", "linkfun", "linkinv",
    "variance", "dev.resids", "aic", "mu.eta", "initialize",
    "validmu", "valideta", "simulate"), class = "family"), corstr = "exchangeable")

Summary of Residuals:
        Min          1Q      Median          3Q         Max
-39.1944672  -2.1913793  -0.2236836  -0.2047618 106.5134108


Coefficients:
               Estimate Naive S.E.    Naive z Robust S.E.    Robust z
(Intercept) -0.96680028  0.7196563 -1.3434195   0.4550626 -2.12454341
target      -0.02083277  0.2413024 -0.0863347   0.3341394 -0.06234754
```

```
coop         1.21033147  0.1914327  6.3224910   0.2631640  4.59915320

Estimated Scale Parameter:  17.10881
Number of Iterations:  2

Working Correlation
            [,1]        [,2]        [,3]        [,4]        [,5]
[1,]  1.00000000 -0.01180279 -0.01180279 -0.01180279 -0.01180279
[2,] -0.01180279  1.00000000 -0.01180279 -0.01180279 -0.01180279
[3,] -0.01180279 -0.01180279  1.00000000 -0.01180279 -0.01180279
[4,] -0.01180279 -0.01180279 -0.01180279  1.00000000 -0.01180279
[5,] -0.01180279 -0.01180279 -0.01180279 -0.01180279  1.00000000
```

Set explanatory variables to their default values:

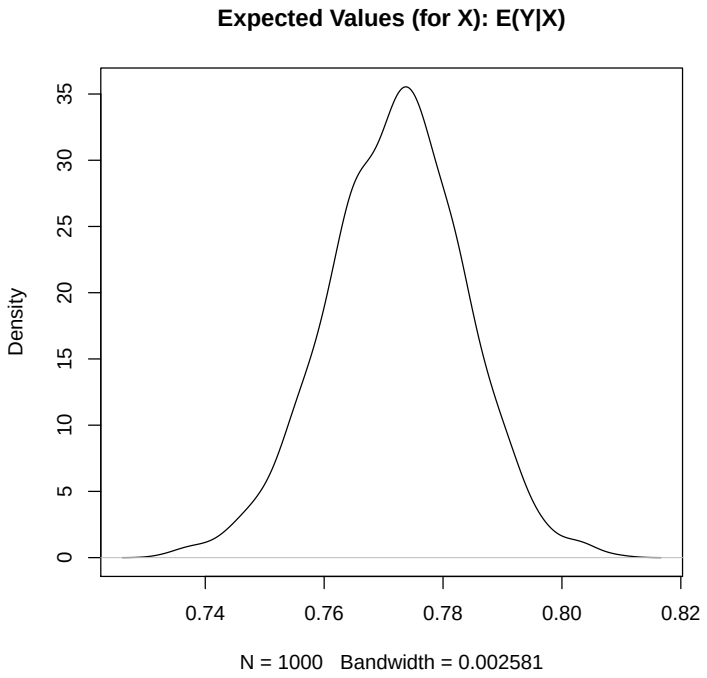
```
> x.out <- setx(z.out)
```

Simulate quantities of interest

```
> s.out <- sim(z.out, x = x.out)
> summary(s.out)

Model:  poisson.gee
Number of simulations:  1000

Values of X
  (Intercept)   target     coop
1           1 2.141026 1.807692
attr(,"assign")
[1] 0 1 2

Expected Values (for x): E(Y|X)
 mean    sd   50% 2.5% 97.5%
 0.88 0.225 0.832  0.6 1.414
```

Generate a plot of quantities of interest:

```
> plot(s.out)
```



4.0.22 The Model

Suppose we have a panel dataset, with Y_{it} denoting the dependent variable of the number of independent events for a fixed period of time for unit i at time t. Y_i is a vector or cluster of correlated data where y_{it} is correlated with $y_{it'}$ for some or all t, t'. Note that the model assumes correlations within i but independence across i.

- The *stochastic component* is given by the joint and marginal distributions

$$\begin{aligned} Y_i &\sim f(y_i \mid \lambda_i) \\ Y_{it} &\sim g(y_{it} \mid \lambda_{it}) \end{aligned}$$

  where f and g are unspecified distributions with means λ_i and λ_{it}. GEE models make no distributional assumptions and only require three specifications: a mean function, a variance function, and a correlation structure.

- The *systematic component* is the *mean function*, given by:

$$\lambda_{it} = \exp(x_{it}\beta)$$

  where x_{it} is the vector of k explanatory variables for unit i at time t and β is the vector of coefficients.

- The *variance function* is given by:

$$V_{it} = \lambda_{it}$$

- The *correlation structure* is defined by a $T \times T$ "working" correlation matrix, where T is the size of the largest cluster. Users must specify the structure of the "working" correlation matrix *a priori*. The "working" correlation matrix then enters the variance term for each i, given by:

$$V_i = \phi\, A_i^{\frac{1}{2}} R_i(\alpha) A_i^{\frac{1}{2}}$$

  where A_i is a $T \times T$ diagonal matrix with the variance function $V_{it} = \lambda_{it}$ as the tth diagonal element, $R_i(\alpha)$ is the "working" correlation matrix, and ϕ is a scale parameter. The parameters are then estimated via a quasi-likelihood approach.

- In GEE models, if the mean is correctly specified, but the variance and correlation structure are incorrectly specified, then GEE models provide consistent estimates of the parameters and thus the mean function as well, while consistent estimates of the standard errors can be obtained via a robust "sandwich" estimator. Similarly, if the mean and variance are correctly specified but the correlation structure is incorrectly specified, the parameters can be estimated consistently and the standard errors can be estimated consistently with the sandwich estimator. If all three are specified correctly, then the estimates of the parameters are more efficient.

- The robust "sandwich" estimator gives consistent estimates of the standard errors when the correlations are specified incorrectly only if the number of units i is relatively large and the number of repeated periods t is relatively small. Otherwise, one should use the "naïve" model-based standard errors, which assume that the specified correlations are close approximations to the true underlying correlations. See **?**] for more details.

4.0.23 Quantities of Interest

- All quantities of interest are for marginal means rather than joint means.

- The method of bootstrapping generally should not be used in GEE models. If you must bootstrap, bootstrapping should be done within clusters, which is not currently supported in Zelig. For conditional prediction models, data should be matched within clusters.

- The expected values (`qi$ev`) for the GEE poisson model is the mean of simulations from the stochastic component:

$$E(Y) = \lambda_c = \exp(x_c \beta),$$

  given draws of β from its sampling distribution, where x_c is a vector of values, one for each independent variable, chosen by the user.

- The first difference (`qi$fd`) for the GEE poisson model is defined as

$$\text{FD} = \Pr(Y = 1 \mid x_1) - \Pr(Y = 1 \mid x).$$

- In conditional prediction models, the average expected treatment effect (`att.ev`) for the treatment group is

$$\frac{1}{\sum_{i=1}^{n}\sum_{t=1}^{T} tr_{it}} \sum_{i:tr_{it}=1}^{n} \sum_{t:tr_{it}=1}^{T} \left\{ Y_{it}(tr_{it} = 1) - E[Y_{it}(tr_{it} = 0)] \right\},$$

  where tr_{it} is a binary explanatory variable defining the treatment ($tr_{it} = 1$) and control ($tr_{it} = 0$) groups. Variation in the simulations are due to uncertainty in simulating $E[Y_{it}(tr_{it} = 0)]$, the counterfactual expected value of Y_{it} for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $tr_{it} = 0$.

4.0.24 Output Values

The output of each Zelig command contains useful information which you may view. For example, if you run `z.out <- zelig(y ~ x, model = "poisson.gee", id, data)`, then you may examine the available information in `z.out` by using `names(z.out)`, see the `coefficients` by using `z.out$coefficients`, and a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output object `z.out`, you may extract:
  - `coefficients`: parameter estimates for the explanatory variables.
  - `residuals`: the working residuals in the final iteration of the fit.
  - `fitted.values`: the vector of fitted values for the systemic component, λ_{it}.
  - `linear.predictors`: the vector of $x_{it}\beta$
  - `max.id`: the size of the largest cluster.
- From `summary(z.out)`, you may extract:
  - `coefficients`: the parameter estimates with their associated standard errors, p-values, and z-statistics.
  - `working.correlation`: the "working" correlation matrix
- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as matrices indexed by simulation $\times$ `x`-observation (for more than one `x`-observation). Available quantities are:
  - `qi$ev`: the simulated expected values for the specified values of `x`.

– `qi$fd`: the simulated first difference in the expected probabilities for the values specified in `x` and `x1`.

– `qi$att.ev`: the simulated average expected treatment effect for the treated from conditional prediction models.

How To Cite *poisson.gee* Zelig model

Patrick Lam. 2007. "poisson.gee: Generalized Estimating Equation for Poisson Regression," in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software," `http://gking.harvard.edu/zelig`.

How to Cite the Zelig Software Package

To cite Zelig as a whole, please reference these two sources:

Kosuke Imai, Gary King, and Olivia Lau. 2007. "Zelig: Everyone's Statistical Software," `http://GKing.harvard.edu/zelig`.

Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The gee function is part of the gee package by Vincent J. Carey, ported to R by Thomas Lumley and Brian Ripley. Advanced users may wish to refer to `help(gee)` and `help(family)`. Sample data are from [3]. Please inquire with Lisa Martin before publishing results from these data, as this dataset includes errors that have since been corrected.

5 `probit.gee`: Generalized Estimating Equation for Probit Regression

The GEE probit estimates the same model as the standard probit regression (appropriate when you have a dichotomous dependent variable and a set of explanatory variables). Unlike in probit regression, GEE probit allows for dependence within clusters, such as in longitudinal data, although its use is not limited to just panel data. The user must first specify a "working" correlation matrix for the clusters, which models the dependence of each observation with other observations in the same cluster. The "working" correlation matrix is a $T \times T$ matrix of correlations, where T is the size of the largest cluster and the elements of the matrix are correlations between within-cluster observations. The appeal of GEE models is that it gives consistent estimates of the

parameters and consistent estimates of the standard errors can be obtained using a robust "sandwich" estimator even if the "working" correlation matrix is incorrectly specified. If the "working" correlation matrix is correctly specified, GEE models will give more efficient estimates of the parameters. GEE models measure population-averaged effects as opposed to cluster-specific effects (See **(author?)** [4]).

5.0.25 Syntax

```
> z.out <- zelig(Y ~ X1 + X2, model = "probit.gee",
                 id = "X3", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

where `id` is a variable which identifies the clusters. The data should be sorted by `id` and should be ordered within each cluster when appropriate.

5.0.26 Additional Inputs

- `robust`: defaults to `TRUE`. If `TRUE`, consistent standard errors are estimated using a "sandwich" estimator.

Use the following arguments to specify the structure of the "working" correlations within clusters:

- `corstr`: defaults to `"independence"`. It can take on the following arguments:
  - Independence (`corstr = "independence"`): $\text{cor}(y_{it}, y_{it'}) = 0$, $\forall t, t'$ with $t \neq t'$. It assumes that there is no correlation within the clusters and the model becomes equivalent to standard probit regression. The "working" correlation matrix is the identity matrix.
  - Fixed (`corstr = "fixed"`): If selected, the user must define the "working" correlation matrix with the `R` argument rather than estimating it from the model.
  - Stationary m dependent (`corstr = "stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{|t-t'|} & \text{if} \quad |t-t'| \leq m \\ 0 & \text{if} \quad |t-t'| > m \end{cases}$$

    If (`corstr = "stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. Choose this option when the correlations are assumed to be the same for observations of the same $|t-t'|$ periods apart for $|t-t'| \leq m$.

    Sample "working" correlation for Stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_1 & \alpha_2 & 0 & 0 \\ \alpha_1 & 1 & \alpha_1 & \alpha_2 & 0 \\ \alpha_2 & \alpha_1 & 1 & \alpha_1 & \alpha_2 \\ 0 & \alpha_2 & \alpha_1 & 1 & \alpha_1 \\ 0 & 0 & \alpha_2 & \alpha_1 & 1 \end{pmatrix}$$

– Non-stationary m dependent (`corstr = "non_stat_M_dep"`):

$$\text{cor}(y_{it}, y_{it'}) = \begin{cases} \alpha_{tt'} & \text{if} \quad |t - t'| \leq m \\ 0 & \text{if} \quad |t - t'| > m \end{cases}$$

If (`corstr = "non_stat_M_dep"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. This option relaxes the assumption that the correlations are the same for all observations of the same $|t - t'|$ periods apart.

Sample "working" correlation for Non-stationary 2 dependence (`Mv`=2)

$$\begin{pmatrix} 1 & \alpha_{12} & \alpha_{13} & 0 & 0 \\ \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} & 0 \\ \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} & \alpha_{35} \\ 0 & \alpha_{24} & \alpha_{34} & 1 & \alpha_{45} \\ 0 & 0 & \alpha_{35} & \alpha_{45} & 1 \end{pmatrix}$$

– Exchangeable (`corstr = "exchangeable"`): $\text{cor}(y_{it}, y_{it'}) = \alpha$, $\forall t, t'$ with $t \neq t'$. Choose this option if the correlations are assumed to be the same for all observations within the cluster.

Sample "working" correlation for Exchangeable

$$\begin{pmatrix} 1 & \alpha & \alpha & \alpha & \alpha \\ \alpha & 1 & \alpha & \alpha & \alpha \\ \alpha & \alpha & 1 & \alpha & \alpha \\ \alpha & \alpha & \alpha & 1 & \alpha \\ \alpha & \alpha & \alpha & \alpha & 1 \end{pmatrix}$$

– Stationary mth order autoregressive (`corstr = "AR-M"`): If (`corstr = "AR-M"`), you must also specify `Mv` $= m$, where m is the number of periods t of dependence. For example, the first order autoregressive model (AR-1) implies $\text{cor}(y_{it}, y_{it'}) = \alpha^{|t-t'|}, \forall t, t'$ with $t \neq t'$. In AR-1, observation 1 and observation 2 have a correlation of α. Observation 2 and observation 3 also have a correlation of α. Observation 1 and observation 3 have a correlation of α^2, which is a function of how 1 and 2 are correlated (α) multiplied by how 2 and 3 are correlated (α). Observation 1 and 4 have a correlation that is a function of the correlation between 1 and 2, 2 and 3, and 3 and 4, and so forth.

Sample "working" correlation for Stationary AR-1 (`Mv`=1)

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ \alpha & 1 & \alpha & \alpha^2 & \alpha^3 \\ \alpha^2 & \alpha & 1 & \alpha & \alpha^2 \\ \alpha^3 & \alpha^2 & \alpha & 1 & \alpha \\ \alpha^4 & \alpha^3 & \alpha^2 & \alpha & 1 \end{pmatrix}$$

  - Unstructured (`corstr = "unstructured"`): $\text{cor}(y_{it}, y_{it'}) = \alpha_{tt'}, \forall t, t'$ with $t \neq t'$. No constraints are placed on the correlations, which are then estimated from the data.

- `Mv`: defaults to 1. It specifies the number of periods of correlation and only needs to be specified when `corstr` is `"stat_M_dep"`, `"non_stat_M_dep"`, or `"AR-M"`.

- `R`: defaults to `NULL`. It specifies a user-defined correlation matrix rather than estimating it from the data. The argument is used only when `corstr` is `"fixed"`. The input is a $T \times T$ matrix of correlations, where T is the size of the largest cluster.

5.0.27 Examples

1. Example with Stationary 3 Dependence

   Attaching the sample turnout dataset:

   ```
   > data(turnout)
   ```

   Variable identifying clusters

   ```
   > turnout$cluster <- rep(c(1:200),10)
   ```

   Sorting by cluster

   ```
   > sorted.turnout <- turnout[order(turnout$cluster),]
   ```

   Estimating parameter values:

   ```
   > z.out1 <- zelig(vote ~ race + educate, model = "probit.gee", id = "cluster", data = s
   ```

   ```
   (Intercept)   racewhite     educate
   -0.72594913  0.29907642  0.09711897
   ```

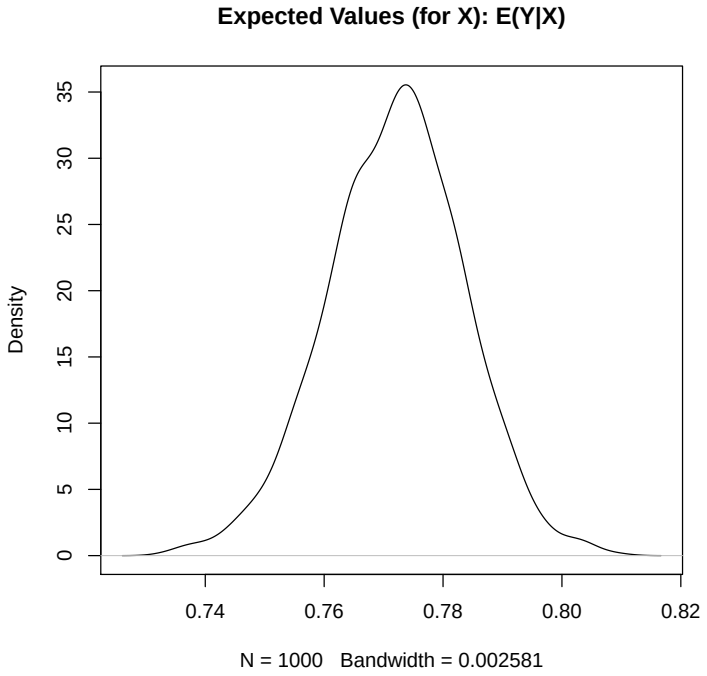
   ```
   How to cite this model in Zelig:
    Patrick Lam. 2012.
    "probit.gee: General Estimating Equation for Poisson Regression"
    in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
    http://gking.harvard.edu/zelig
   ```

Setting values for the explanatory variables to their default values:

```
> x.out1 <- setx(z.out1)
```

Simulating quantities of interest:

```
> s.out1 <- sim(z.out1, x = x.out1)

> summary(s.out1)

Model:  probit.gee
Number of simulations:  1000

Values of X
  (Intercept) racewhite  educate
1           1         1 12.06675
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$race
[1] "contr.treatment"


Expected Values (for x): E(Y|X)
  mean    sd   50%  2.5% 97.5%
 0.772 0.012 0.773 0.748 0.793

> plot(s.out1)
```

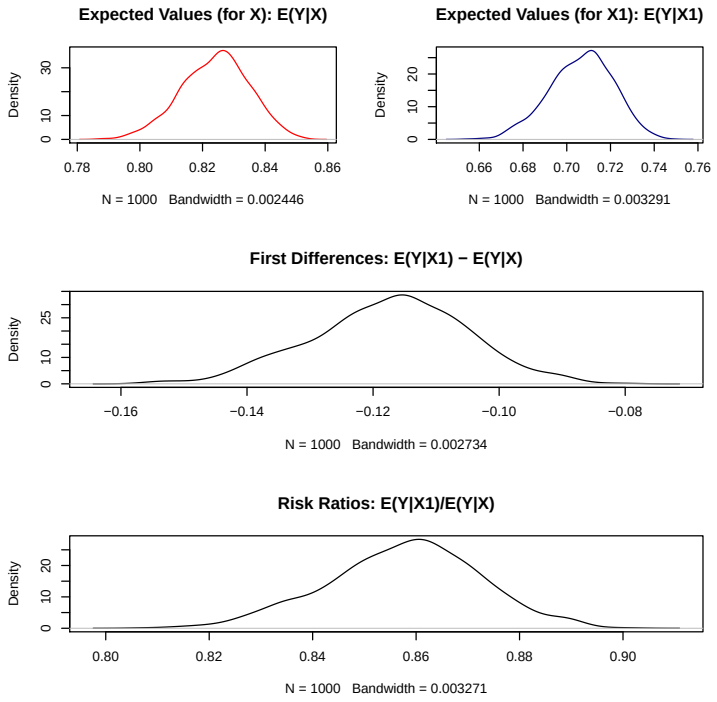


2. Simulating First Differences

   Estimating the risk difference (and risk ratio) between low education (25th percentile) and high education (75th percentile) while all the other variables held at their default values.

```
> x.high <- setx(z.out1, educate = quantile(turnout$educate, prob = 0.75))
> x.low <- setx(z.out1, educate = quantile(turnout$educate, prob = 0.25))

> s.out2 <- sim(z.out1, x = x.high, x1 = x.low)

> summary(s.out2)

Model:  probit.gee
Number of simulations:  1000

Values of X
  (Intercept) racewhite educate
1           1         1      14
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$race
```

```
[1] "contr.treatment"


Values of X1
  (Intercept) racewhite educate
1           1         1      10
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$race
[1] "contr.treatment"


Expected Values (for x): E(Y|X)
  mean    sd   50%  2.5% 97.5%
 0.824 0.011 0.825 0.801 0.844

Expected Values (for x1): E(Y|X1)
  mean    sd   50%  2.5% 97.5%
 0.707 0.015 0.708 0.677 0.733

First Differences: E(Y|X1) - E(Y|X)
   mean    sd    50%   2.5%  97.5%
 -0.117 0.012 -0.117 -0.142 -0.094

Risk Ratios: E(Y|X1)/E(Y|X)
  mean    sd   50%  2.5% 97.5%
 0.858 0.015 0.858 0.827 0.887

> plot(s.out2)
```



3. Example with Fixed Correlation Structure

   User-defined correlation structure

```
> corr.mat <- matrix(rep(0.5,100), nrow=10, ncol=10)
> diag(corr.mat) <- 1
```

   Generating empirical estimates:

```
> z.out2 <- zelig(vote ~ race + educate, model = "probit.gee", id = "cluster", data = s
```

```
(Intercept)   racewhite     educate
-0.72594913  0.29907642  0.09711897


 How to cite this model in Zelig:
  Patrick Lam. 2012.
  "probit.gee: General Estimating Equation for Poisson Regression"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig
```

   Viewing the regression output:

```
> summary(z.out2)
```

```
 GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:                      Probit
 Variance to Mean Relation: Binomial
 Correlation Structure:     Fixed

Call:
gee(formula = vote ~ race + educate, id = INTEGER2000, data = Data.frame,
    R = corr.mat, family = structure(list(family = "binomial",
        link = "probit", linkfun = function (mu)
        qnorm(mu), linkinv = function (eta)
        {
            thresh <- -qnorm(.Machine$double.eps)
            eta <- pmin(pmax(eta, -thresh), thresh)
            pnorm(eta)
        }, variance = function (mu)
        mu * (1 - mu), dev.resids = function (y, mu, wt)
        .Call(C_binomial_dev_resids, y, mu, wt), aic = function (y,
            n, mu, wt, dev)
        {
            m <- if (any(n > 1))
                n
            else wt
            -2 * sum(ifelse(m > 0, (wt/m), 0) * dbinom(round(m *
                y), round(m), mu, log = TRUE))
        }, mu.eta = function (eta)
        pmax(dnorm(eta), .Machine$double.eps), initialize = expression(
            {
                if (NCOL(y) == 1) {
                  if (is.factor(y))
                    y <- y != levels(y)[1L]
                  n <- rep.int(1, nobs)
                  y[weights == 0] <- 0
                  if (any(y < 0 | y > 1))
                    stop("y values must be 0 <= y <= 1")
                  mustart <- (weights * y + 0.5)/(weights + 1)
                  m <- weights * y
                  if (any(abs(m - round(m)) > 0.001))
                    warning("non-integer #successes in a binomial glm!")
                }
                else if (NCOL(y) == 2) {
                  if (any(abs(y - round(y)) > 0.001))
                    warning("non-integer counts in a binomial glm!")
                  n <- y[, 1] + y[, 2]
```

```
                y <- ifelse(n == 0, 0, y[, 1]/n)
                weights <- weights * n
                mustart <- (n * y + 0.5)/(n + 1)
            }
            else stop("for the binomial family, y must be a vector of 0 and 1's\n",
              "or a 2 column matrix where col 1 is no. successes and col 2 is no. f
        }), validmu = function (mu)
    all(mu > 0) && all(mu < 1), valideta = function (eta)
    TRUE, simulate = function (object, nsim)
    {
        ftd <- fitted(object)
        n <- length(ftd)
        ntot <- n * nsim
        wts <- object$prior.weights
        if (any(wts%%1 != 0))
            stop("cannot simulate from non-integer prior.weights")
        if (!is.null(m <- object$model)) {
            y <- model.response(m)
            if (is.factor(y)) {
              yy <- factor(1 + rbinom(ntot, size = 1, prob = ftd),
                labels = levels(y))
              split(yy, rep(seq_len(nsim), each = n))
            }
            else if (is.matrix(y) && ncol(y) == 2) {
              yy <- vector("list", nsim)
              for (i in seq_len(nsim)) {
                Y <- rbinom(n, size = wts, prob = ftd)
                YY <- cbind(Y, wts - Y)
                colnames(YY) <- colnames(y)
                yy[[i]] <- YY
              }
              yy
            }
            else rbinom(ntot, size = wts, prob = ftd)/wts
        }
        else rbinom(ntot, size = wts, prob = ftd)/wts
    }), .Names = c("family", "link", "linkfun", "linkinv",
"variance", "dev.resids", "aic", "mu.eta", "initialize",
"validmu", "valideta", "simulate"), class = "family"), corstr = "fixed")

Summary of Residuals:
       Min         1Q     Median         3Q        Max
-0.9191419 -0.3146504  0.2063033  0.2349483  0.7801544


Coefficients:
```

```
              Estimate  Naive S.E.   Naive z Robust S.E.  Robust z
(Intercept) -0.77271488 0.105618565 -7.316090 0.133841982 -5.773337
racewhite    0.33534707 0.061921463  5.415684 0.088349410  3.795691
educate      0.09666793 0.007082234 13.649355 0.009711359  9.954110

Estimated Scale Parameter:  0.9734069
Number of Iterations:  3

Working Correlation
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]  1.0  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5   0.5
 [2,]  0.5  1.0  0.5  0.5  0.5  0.5  0.5  0.5  0.5   0.5
 [3,]  0.5  0.5  1.0  0.5  0.5  0.5  0.5  0.5  0.5   0.5
 [4,]  0.5  0.5  0.5  1.0  0.5  0.5  0.5  0.5  0.5   0.5
 [5,]  0.5  0.5  0.5  0.5  1.0  0.5  0.5  0.5  0.5   0.5
 [6,]  0.5  0.5  0.5  0.5  0.5  1.0  0.5  0.5  0.5   0.5
 [7,]  0.5  0.5  0.5  0.5  0.5  0.5  1.0  0.5  0.5   0.5
 [8,]  0.5  0.5  0.5  0.5  0.5  0.5  0.5  1.0  0.5   0.5
 [9,]  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  1.0   0.5
[10,]  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5   1.0
```

5.0.28 The Model

Suppose we have a panel dataset, with Y_{it} denoting the binary dependent variable for unit i at time t. Y_i is a vector or cluster of correlated data where y_{it} is correlated with $y_{it'}$ for some or all t, t'. Note that the model assumes correlations within i but independence across i.

- The *stochastic component* is given by the joint and marginal distributions

$$\begin{aligned} Y_i &\sim f(y_i \mid \pi_i) \\ Y_{it} &\sim g(y_{it} \mid \pi_{it}) \end{aligned}$$

  where f and g are unspecified distributions with means π_i and π_{it}. GEE models make no distributional assumptions and only require three specifications: a mean function, a variance function, and a correlation structure.

- The *systematic component* is the *mean function*, given by:

$$\pi_{it} = \Phi(x_{it}\beta)$$

  where $\Phi(\mu)$ is the cumulative distribution function of the Normal distribution with mean 0 and unit variance, x_{it} is the vector of k explanatory variables for unit i at time t and β is the vector of coefficients.

- The *variance function* is given by:

$$V_{it} = \pi_{it}(1 - \pi_{it})$$

- The *correlation structure* is defined by a $T \times T$ "working" correlation matrix, where T is the size of the largest cluster. Users must specify the structure of the "working" correlation matrix *a priori*. The "working" correlation matrix then enters the variance term for each i, given by:

$$V_i = \phi \, A_i^{\frac{1}{2}} R_i(\alpha) A_i^{\frac{1}{2}}$$

  where A_i is a $T \times T$ diagonal matrix with the variance function $V_{it} = \pi_{it}(1 - \pi_{it})$ as the tth diagonal element, $R_i(\alpha)$ is the "working" correlation matrix, and ϕ is a scale parameter. The parameters are then estimated via a quasi-likelihood approach.

- In GEE models, if the mean is correctly specified, but the variance and correlation structure are incorrectly specified, then GEE models provide consistent estimates of the parameters and thus the mean function as well, while consistent estimates of the standard errors can be obtained via a robust "sandwich" estimator. Similarly, if the mean and variance are correctly specified but the correlation structure is incorrectly specified, the parameters can be estimated consistently and the standard errors can be estimated consistently with the sandwich estimator. If all three are specified correctly, then the estimates of the parameters are more efficient.

- The robust "sandwich" estimator gives consistent estimates of the standard errors when the correlations are specified incorrectly only if the number of units i is relatively large and the number of repeated periods t is relatively small. Otherwise, one should use the "naïve" model-based standard errors, which assume that the specified correlations are close approximations to the true underlying correlations. See **?**] for more details.

5.0.29 Quantities of Interest

- All quantities of interest are for marginal means rather than joint means.

- The method of bootstrapping generally should not be used in GEE models. If you must bootstrap, bootstrapping should be done within clusters, which is not currently supported in Zelig. For conditional prediction models, data should be matched within clusters.

- The expected values (`qi$ev`) for the GEE probit model are simulations of the predicted probability of a success:

$$E(Y) = \pi_c = \Phi(x_c \beta),$$

  given draws of β from its sampling distribution, where x_c is a vector of values, one for each independent variable, chosen by the user.

- The first difference (`qi$fd`) for the GEE probit model is defined as

$$\text{FD} = \Pr(Y = 1 \mid x_1) - \Pr(Y = 1 \mid x).$$

- The risk ratio (`qi$rr`) is defined as

$$\text{RR} = \Pr(Y = 1 \mid x_1) \;/\; \Pr(Y = 1 \mid x).$$

- In conditional prediction models, the average expected treatment effect (`att.ev`) for the treatment group is

$$\frac{1}{\sum_{i=1}^{n}\sum_{t=1}^{T} tr_{it}} \sum_{i:tr_{it}=1}^{n} \sum_{t:tr_{it}=1}^{T} \left\{ Y_{it}(tr_{it}=1) - E[Y_{it}(tr_{it}=0)] \right\},$$

  where tr_{it} is a binary explanatory variable defining the treatment ($tr_{it} = 1$) and control ($tr_{it} = 0$) groups. Variation in the simulations are due to uncertainty in simulating $E[Y_{it}(tr_{it} = 0)]$, the counterfactual expected value of Y_{it} for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $tr_{it} = 0$.

5.0.30 Output Values

The output of each Zelig command contains useful information which you may view. For example, if you run `z.out <- zelig(y ~ x, model = "probit.gee", id, data)`, then you may examine the available information in `z.out` by using `names(z.out)`, see the `coefficients` by using `z.out$coefficients`, and a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output object `z.out`, you may extract:
  - `coefficients`: parameter estimates for the explanatory variables.
  - `residuals`: the working residuals in the final iteration of the fit.
  - `fitted.values`: the vector of fitted values for the systemic component, π_{it}.
  - `linear.predictors`: the vector of $x_{it}\beta$
  - `max.id`: the size of the largest cluster.
- From `summary(z.out)`, you may extract:
  - `coefficients`: the parameter estimates with their associated standard errors, p-values, and z-statistics.
  - `working.correlation`: the "working" correlation matrix
- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as matrices indexed by simulation $\times$ `x`-observation (for more than one `x`-observation). Available quantities are:
  - `qi$ev`: the simulated expected probabilities for the specified values of `x`.

- `qi$fd`: the simulated first difference in the expected probabilities for the values specified in `x` and `x1`.
- `qi$rr`: the simulated risk ratio for the expected probabilities simulated from `x` and `x1`.
- `qi$att.ev`: the simulated average expected treatment effect for the treated from conditional prediction models.

How To Cite *probit.gee* Zelig model

Patrick Lam. 2007. "probit.gee: Generalized Estimating Equation for Probit Regression," in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software," `http://gking.harvard.edu/zelig`.

How to Cite the Zelig Software Package

To cite Zelig as a whole, please reference these two sources:

Kosuke Imai, Gary King, and Olivia Lau. 2007. "Zelig: Everyone's Statistical Software," `http://GKing.harvard.edu/zelig`.

Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The gee function is part of the gee package by Vincent J. Carey, ported to R by Thomas Lumley and Brian Ripley. Advanced users may wish to refer to `help(gee)` and `help(family)`. Sample data are from [2].

References

[1] Gary King, James Alt, Nancy Burns, and Michael Laver. A unified model of cabinet dissolution in parliamentary democracies. *American Journal of Political Science*, 34(3):846–871, August 1990. http://gking.harvard.edu/files/abs/coal-abs.shtml.

[2] Gary King, Michael Tomz, and Jason Wittenberg. Making the most of statistical analyses: Improving interpretation and presentation. *American Journal of Political Science*, 44(2):341–355, April 2000. http://gking.harvard.edu/files/abs/making-abs.shtml.

[3] Lisa Martin. *Coercive Cooperation: Explaining Multilateral Economic Sanctions*. Princeton University Press, 1992. Please inquire with Lisa Martin before publishing results from these data, as this dataset includes errors that have since been corrected.

[4] Christopher Zorn. Generalized estimating equation models for correlated data: A review with applications. *American Journal of Political Science*, 45:470–490, April 2001.