0.1 `normal.survey`: Survey-Weighted Normal Regression for Continuous Dependent Variables

The survey-weighted Normal regression model is appropriate for survey data obtained using complex sampling techniques, such as stratified random or cluster sampling (e.g., not simple random sampling). Like the least squares and Normal regression models (see Section **??** and Section **??**), survey-weighted Normal regression specifies a continuous dependent variable as a linear function of a set of explanatory variables. The survey-weighted normal model reports estimates of model parameters identical to least squares or Normal regression estimates, but uses information from the survey design to correct variance estimates.

The `normal.survey` model accommodates three common types of complex survey data. Each method listed here requires selecting specific options which are detailed in the "Additional Inputs" section below.

1. **Survey weights**: Survey data are often published along with weights for each observation. For example, if a survey intentionally over-samples a particular type of case, weights can be used to correct for the over-representation of that type of case in the dataset. Survey weights come in two forms:
   (a) *Probability* weights report the probability that each case is drawn from the population. For each stratum or cluster, this is computed as the number of observations in the sample drawn from that group divided by the number of observations in the population in the group.
   (b) *Sampling* weights are the inverse of the probability weights.
2. **Strata/cluster identification**: A complex survey dataset may include variables that identify the strata or cluster from which observations are drawn. For stratified random sampling designs, observations may be nested in different strata. There are two ways to employ these identifiers:
   (a) Use *finite population corrections* to specify the total number of cases in the stratum or cluster from which each observation was drawn.
   (b) For stratified random sampling designs, use the raw strata ids to compute sampling weights from the data.
3. **Replication weights**: To preserve the anonymity of survey participants, some surveys exclude strata and cluster ids from the public data and instead release only precomputed replicate weights.

Syntax

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

Additional Inputs

In addition to the standard `zelig` inputs (see Section **??**), survey-weighted Normal models accept the following optional inputs:

1. Datasets that include survey weights:

    - `probs`: An optional formula or numerical vector specifying each case's probability weight, the probability that the case was selected. Probability weights need not (and, in most cases, will not) sum to one. Cases with lower probability weights are weighted more heavily in the computation of model coefficients.
    - `weights`: An optional numerical vector specifying each case's sample weight, the inverse of the probability that the case was selected. Sampling weights need not (and, in most cases, will not) sum to one. Cases with higher sampling weights are weighted more heavily in the computation of model coefficients.

2. Datasets that include strata/cluster identifiers:

    - `ids`: An optional formula or numerical vector identifying the cluster from which each observation was drawn (ordered from largest level to smallest level). For survey designs that do not involve cluster sampling, `ids` defaults to `NULL`.
    - `fpc`: An optional numerical vector identifying each case's frequency weight, the total number of units in the population from which each observation was sampled.
    - `strata`: An optional formula or vector identifying the stratum from which each observation was sampled. Entries may be numerical, logical, or strings. For survey designs that do not involve cluster sampling, `strata` defaults to `NULL`.
    - `nest`: An optional logical value specifying whether primary sampling unites (PSUs) have non-unique ids across multiple strata. `nest=TRUE` is appropriate when PSUs reuse the same identifiers across strata. Otherwise, `nest` defaults to `FALSE`.
    - `check.strata`: An optional input specifying whether to check that clusters are nested in strata. If `check.strata` is left at its default, `!nest`, the check is not performed. If `check.strata` is specified as `TRUE`, the check is carried out.

3. Datasets that include replication weights:

    - `repweights`: A formula or matrix specifying replication weights, numerical vectors of weights used in a process in which the sample is repeatedly re-weighted and parameters are re-estimated in order to compute the variance of the model parameters.
    - `type`: A string specifying the type of replication weights being used. This input is required if replicate weights are specified. The following types of replication weights are recognized: `"BRR"`, `"Fay"`, `"JK1"`, `"JKn"`, `"bootstrap"`, or `"other"`.

- `weights`: An optional vector or formula specifying each case's sample weight, the inverse of the probability that the case was selected. If a survey includes both sampling weights and replicate weights separately for the same survey, both should be included in the model specification. In these cases, sampling weights are used to correct potential biases in in the computation of coefficients and replication weights are used to compute the variance of coefficient estimates.
- `combined.weights`: An optional logical value that should be specified as `TRUE` if the replicate weights include the sampling weights. Otherwise, `combined.weights` defaults to `FALSE`.
- `rho`: An optional numerical value specifying a shrinkage factor for replicate weights of type `"Fay"`.
- `bootstrap.average`: An optional numerical input specifying the number of iterations over which replicate weights of type `"bootstrap"` were averaged. This input should be left as `NULL` for `"bootstrap"` weights that were not were created by averaging.
- `scale`: When replicate weights are included, the variance is computed as the sum of squared deviations of the replicates from their mean. `scale` is an optional overall multiplier for the standard deviations.
- `rscale`: Like `scale`, `rscale` specifies an optional vector of replicate-specific multipliers for the squared deviations used in variance computation.
- `fpc`: For models in which `"JK1"`, `"JKn"`, or `"other"` replicates are specified, `fpc` is an optional numerical vector (with one entry for each replicate) designating the replicates' finite population corrections.
- `fpctype`: When a finite population correction is included as an `fpc` input, `fpctype` is a required input specifying whether the input to `fpc` is a sampling fraction (`fpctype="fraction"`) or a direct correction (`fpctype="correction"`).
- `return.replicates`: An optional logical value specifying whether the replicates should be returned as a component of the output. `return.replicates` defaults to `FALSE`.

Examples

1. A dataset that includes survey weights:

   Attach the sample data:

   ```
   > data(api, package = "survey")
   ```

   Suppose that a dataset included a continuous measure of public schools' performance (`api00`), a measure of the percentage of students at each school who receive subsidized meals (`meals`), an indicator for whether each school holds classes year round (`year.rnd`), and sampling weights computed by the survey house (`pw`). Estimate a model that regresses school performance on the `meals` and `year.rnd` variables:

```
> z.out1 <- zelig(api00 ~ meals + yr.rnd, model = "normal.survey",
+     weights = ~pw, data = apistrat)
```

Summarize regression coefficients:

```
> summary(z.out1)
```

Set explanatory variables to their default (mean/mode) values, and set a high (80th percentile) and low (20th percentile) value for "meals":

```
> x.low <- setx(z.out1, meals = quantile(apistrat$meals, 0.2))
> x.high <- setx(z.out1, meals = quantile(apistrat$meals, 0.8))
```

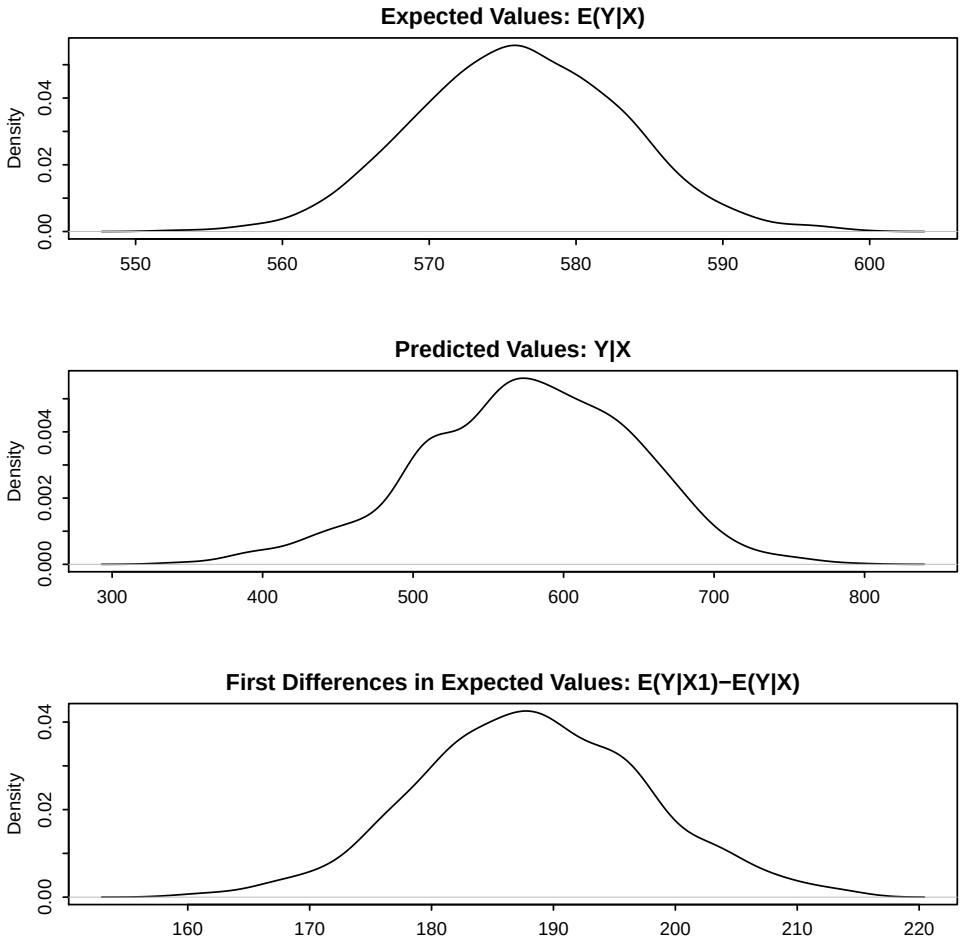
Generate first differences for the effect of high versus low concentrations of children receiving subsidized meals on academic performance:

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

```
> summary(s.out1)
```

Generate a visual summary of the quantities of interest:

```
> plot(s.out1)
```



2. A dataset that includes strata/cluster identifiers:

   Suppose that the survey house that provided the dataset used in the previous example excluded sampling weights but made other details about the survey design available. A user can still estimate a model without sampling weights that instead uses inputs that identify the stratum and/or cluster to which each observation belongs and the size of the finite population from which each observation was drawn.

   Continuing the example above, suppose the survey house drew at random a fixed number of elementary schools, a fixed number of middle schools, and a fixed number of high schools. If the variable `stype` is a vector of characters (`"E"` for elementary schools, `"M"` for middle schools, and `"H"` for high schools) that identifies the type of school each case represents and `fpc` is a numerical vector that identifies for each case the total number of schools of the same type in the population, then the user could estimate the following model:

```
> z.out2 <- zelig(api00 ~ meals + yr.rnd, model = "normal.survey",
+     strata = ~stype, fpc = ~fpc, data = apistrat)
```

Summarize the regression output:

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

The coefficient estimates for this example are identical to the point estimates in the first example, when pre-existing sampling weights were used. When sampling weights are omitted, they are estimated automatically for `"normal.survey"` models based on the user-defined description of sampling designs.

Moreover, because the user provided information about the survey design, the standard error estimates are lower in this example than in the previous example, in which the user omitted variables pertaining to the details of the complex survey design.

3. A dataset that includes replication weights:

   Consider a dataset that includes information for a sample of hospitals that includes counts of the number of out-of-hospital cardiac arrests that each hospital treats and the number of patients who arrive alive at each hospital:

   ```
   > data(scd, package = "survey")
   ```

   Survey houses sometimes supply replicate weights (in lieu of details about the survey design). For the sake of illustrating how replicate weights can be used as inputs in `normal.survey` models, create a set of balanced repeated replicate (BRR) weights:

   ```
   > BRRrep <- 2 * cbind(c(1, 0, 1, 0, 1, 0), c(1, 0, 0, 1, 0, 1),
   +     c(0, 1, 1, 0, 0, 1), c(0, 1, 0, 1, 1, 0))
   ```

   Estimate a model that regresses counts of patients who arrive alive in each hospital on the number of cardiac arrests that each hospital treats, using the BRR replicate weights in `BRRrep` to compute standard errors.

   ```
   > z.out3 <- zelig(alive ~ arrests, model = "poisson.survey", repweights = BRRrep,
   +     type = "BRR", data = scd)
   ```

   Summarize the regression coefficients:

   ```
   > summary(z.out3)
   ```

   Set `arrests` at its 20th and 80th quantiles:

   ```
   > x.low <- setx(z.out3, arrests = quantile(scd$arrests, 0.2))
   > x.high <- setx(z.out3, arrests = quantile(scd$arrests, 0.8))
   ```

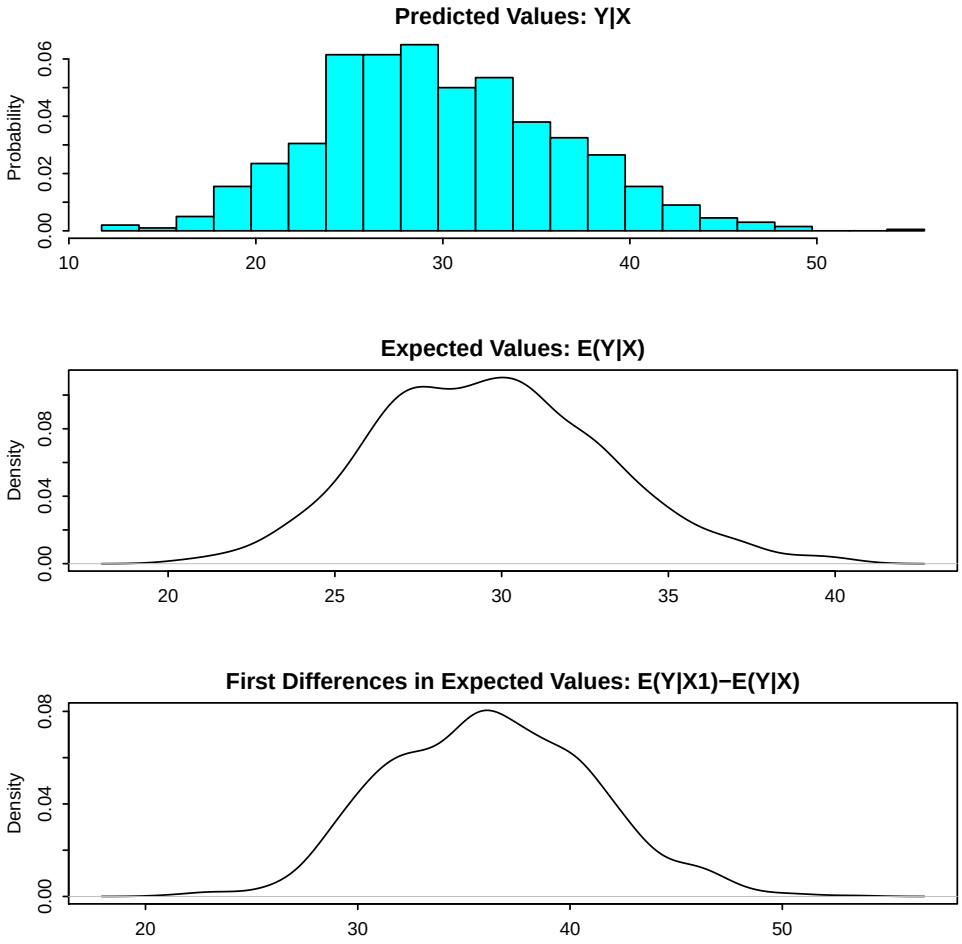
   Generate first differences for the effect of minimal versus maximal cardiac arrests on numbers of patients who arrive alive:

   ```
   > s.out3 <- sim(z.out3, x = x.low, x1 = x.high)
   ```

```
> summary(s.out3)
```

Generate a visual summary of quantities of interest:

```
> plot(s.out3)
```



Model

Let Y_i be the continuous dependent variable for observation i.

- The *stochastic component* is described by a univariate normal model with a vector of means μ_i and scalar variance σ^2:

$$Y_i \sim \text{Normal}(\mu_i, \sigma^2).$$

- The *systematic component* is

$$\mu_i = x_i\beta,$$

where x_i is the vector of k explanatory variables and β is the vector of coefficients.

Variance

When replicate weights are not used, the variance of the coefficients is estimated as

$$\hat{\mathbf{\Sigma}}\left[\sum_{i=1}^{n}\frac{(1-\pi_i)}{\pi_i^2}(\mathbf{X}_i(Y_i-\mu_i))'(\mathbf{X}_i(Y_i-\mu_i))+2\sum_{i=1}^{n}\sum_{j=i+1}^{n}\frac{(\pi_{ij}-\pi_i\pi_j)}{\pi_i\pi_j\pi_{ij}}(\mathbf{X}_i(Y_i-\mu_i))'(\mathbf{X}_j(Y_j-\mu_j))\right]\hat{\mathbf{\Sigma}}$$

where π_i is the probability of case i being sampled, $\mathbf{X}_i$ is a vector of the values of the explanatory variables for case i, Y_i is value of the dependent variable for case i, $\hat{\mu}_i$ is the predicted value of the dependent variable for case i based on the linear model estimates, and $\hat{\mathbf{\Sigma}}$ is the conventional variance-covariance matrix in a parametric glm. This statistic is derived from the method for estimating the variance of sums described in **?** and the Horvitz-Thompson estimator of the variance of a sum described in **?**.

When replicate weights are used, the model is re-estimated for each set of replicate weights, and the variance of each parameter is estimated by summing the squared deviations of the replicates from their mean.

Quantities of Interest

- The expected value (`qi$ev`) is the mean of simulations from the the stochastic component,

  $$E(Y) = \mu_i = x_i\beta,$$

  given a draw of β from its posterior.

- The predicted value (`qi$pr`) is drawn from the distribution defined by the set of parameters (μ_i, σ).

- The first difference (`qi$fd`) is:

  $$\text{FD} \;=\; E(Y \mid x_1) - E(Y \mid x)$$

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

  $$\frac{1}{\sum_{i=1}^{n} t_i}\sum_{i:t_i=1}^{n}\left\{Y_i(t_i=1) - E[Y_i(t_i=0)]\right\},$$

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

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

$$\frac{1}{\sum_{i=1}^{n} t_i} \sum_{i:t_i=1}^{n} \left\{ Y_i(t_i = 1) - \widehat{Y_i(t_i = 0)} \right\},$$

  where t_i is a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups. Variation in the simulations are due to uncertainty in simulating $\widehat{Y_i(t_i = 0)}$, the counterfactual predicted value of Y_i for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $t_i = 0$.

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.survey", 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 IWLS fit.
  - `fitted.values`: fitted values. For the survey-weighted normal model, these are identical to the `linear predictors`.
  - `linear.predictors`: fitted values. For the survey-weighted normal model, these are identical to `fitted.values`.
  - `aic`: Akaike's Information Criterion (minus twice the maximized log-likelihood plus twice the number of coefficients).
  - `df.residual`: the residual degrees of freedom.
  - `df.null`: the residual degrees of freedom for the null model.
  - `zelig.data`: the input data frame if `save.data = TRUE`.
- From `summary(z.out)`, you may extract:
  - `coefficients`: the parameter estimates with their associated standard errors, p-values, and t-statistics.
  - `cov.scaled`: a $k \times k$ matrix of scaled covariances.
  - `cov.unscaled`: a $k \times k$ matrix of unscaled covariances.

- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as matrices indexed by simulation × `x`-observation (for more than one `x`-observation). Available quantities are:
  - `qi$ev`: the simulated expected values for the specified values of `x`.
  - `qi$pr`: the simulated predicted values drawn from the distribution defined by (μ_i, σ).
  - `qi$fd`: the simulated first difference in the simulated expected values for the values specified in `x` and `x1`.
  - `qi$att.ev`: the simulated average expected treatment effect for the treated from conditional prediction models.
  - `qi$att.pr`: the simulated average predicted treatment effect for the treated from conditional prediction models.

When users estimate `normal.survey` models with replicate weights in `Zelig`, an object called `.survey.prob.weights` is created in the global environment. `Zelig` will over-write any existing object with that name, and users are therefore advised to re-name any object called `.survey.prob.weights` before using `normal.survey` models in `Zelig`.

How to Cite

To cite the *normal.survey* Zelig model:

> Nicholas Carnes. 2008. "normal.survey: Survey-Weighted Normal Regression for Continuous Dependent Variables" in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"`http://gking.harvard.edu/zelig`

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

Survey-weighted linear models and the sample data used in the examples above are a part of the `survey` package by Thomas Lumley. Users may wish to refer to the help files for the three functions that Zelig draws upon when estimating survey-weighted models, namely, `help(svyglm)`, `help(svydesign)`, and `help(svrepdesign)`. The Gamma model is part of the stats package by **?**. Advanced users may wish to refer to `help(glm)` and `help(family)`, as well as **?**.