1 normal: Normal Regression for Continuous Dependent Variables

The Normal regression model is a close variant of the more standard least squares regression model (see Section **??**). Both models specify a continuous dependent variable as a linear function of a set of explanatory variables. The Normal model reports maximum likelihood (rather than least squares) estimates. The two models differ only in their estimate for the stochastic parameter σ.

1.0.1 Syntax

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

1.0.2 Additional Inputs

In addition to the standard inputs, `zelig()` takes the following additional options for normal regression:

- `robust`: defaults to `FALSE`. If `TRUE` is selected, `zelig()` computes robust standard errors via the `sandwich` package (see [7]). The default type of robust standard error is heteroskedastic and autocorrelation consistent (HAC), and assumes that observations are ordered by time index.

  In addition, `robust` may be a list with the following options:

  - `method`: Choose from
    - `"vcovHAC"`: (default if `robust = TRUE`) HAC standard errors.
    - `"kernHAC"`: HAC standard errors using the weights given in [1].
    - `"weave"`: HAC standard errors using the weights given in [4].
  - `order.by`: defaults to `NULL` (the observations are chronologically ordered as in the original data). Optionally, you may specify a vector of weights (either as `order.by = z`, where `z` exists outside the data frame; or as `order.by = ~z`, where `z` is a variable in the data frame). The observations are chronologically ordered by the size of `z`.
  - `...`: additional options passed to the functions specified in `method`. See the `sandwich` library and [7] for more options.

1.0.3 Examples

1. Basic Example with First Differences

   Attach sample data:

   ```
   > data(macro)
   ```

   Estimate model:

```
> z.out1 <- zelig(unem ~ gdp + capmob + trade, model = "normal",
+     data = macro)
```

Summarize of regression coefficients:

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

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

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

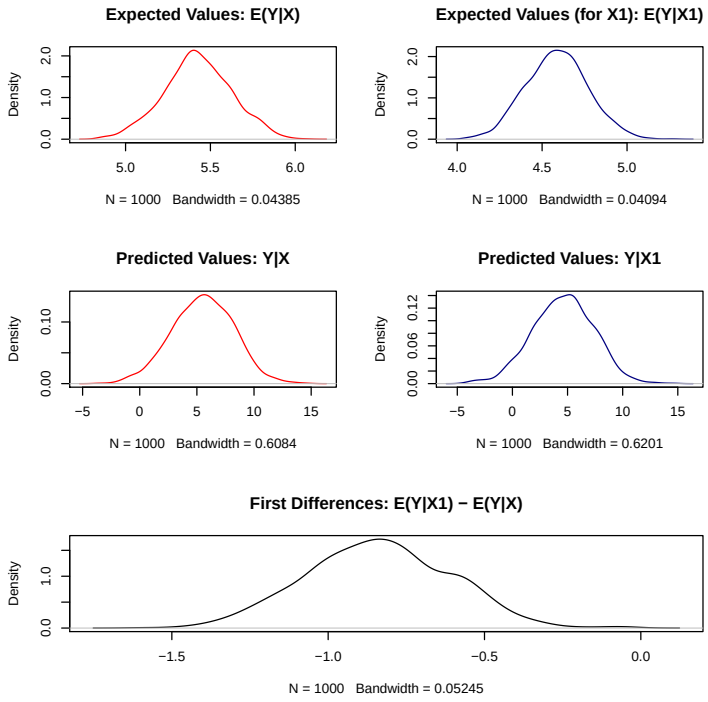
Generate first differences for the effect of high versus low trade on GDP:

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

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

A visual summary of quantities of interest:

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



2. Using Dummy Variables

   Estimate a model with a dummy variable for each year and country (see **??** for help with dummy variables). Note that you do not need to create dummy variables, as the program will automatically parse the unique values in the selected variables into dummy variables.

```
> z.out2 <- zelig(unem ~ gdp + trade + capmob + as.factor(year) +
+     as.factor(country), model = "normal", data = macro)
```

   Set values for the explanatory variables, using the default mean/mode variables, with country set to the United States and Japan, respectively: Simulate quantities of interest:

1.0.4 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.

1.0.5 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

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

  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 $Y_i(\widehat{t_i = 0})$, 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 $t_i = 0$.

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 = "normal", 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 normal model, these are identical to the `linear predictors`.
  - `linear.predictors`: fitted values. For the 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 $\times$ `x`-observation (for more than one `x`-observation). Available quantities are:
  - `qi$ev`: the simulated expected values for the specified values of `x`.
  - `qi$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.

How to Cite the Normal Regression Model

Kosuke Imai, Olivia Lau, and Gary King. *normal: Normal Regression for Continuous Dependent Variables*, 2011

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 normal model is part of the stats package by **(author?)** [6]. Advanced users may wish to refer to `help(glm)` and `help(family)`, as well as [5]. Robust standard errors are implemented via the sandwich package by **(author?)** [7]. Sample data are from [3].

References

[1] Donald W.K. Andrews. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica*, 59(3):817–858, May 1991.

[2] Kosuke Imai, Olivia Lau, and Gary King. *normal: Normal Regression for Continuous Dependent Variables*, 2011.

[3] 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.

[4] Thomas Lumley and Patrick Heagerty. Weighted empirical adaptive variance estimators for correlated data regression. *jrssb*, 61(2):459–477, 1999.

[5] Peter McCullagh and James A. Nelder. *Generalized Linear Models*. Number 37 in Monograph on Statistics and Applied Probability. Chapman & Hall, 2nd edition, 1989.

[6] William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S*. Springer-Verlag, 4th edition, 2002.

[7] Achim Zeileis. Econometric computing with hc and hac covariance matrix estimators. *Journal of Statistical Software*, 11(10):1–17, 2004.