0.1 `ls`: Least Squares Regression for Continuous Dependent Variables

Use least squares regression analysis to estimate the best linear predictor for the specified dependent variables.

Syntax

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

Additional Inputs

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

- `robust`: defaults to `FALSE`. If `TRUE` is selected, `zelig()` computes robust standard errors based on sandwich estimators (see **?**, **?**, and **?**). The default type of robust standard error is heteroskedastic consistent (HC), *not* heteroskedastic and autocorrelation consistent (HAC).

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

  - `method`: choose from
    - `"vcovHC"`: (the default if `robust = TRUE`), HC standard errors.
    - `"vcovHAC"`: HAC standard errors without weights.
    - `"kernHAC"`: HAC standard errors using the weights given in **?**.
    - `"weave"`: HAC standard errors using the weights given in **?**.
  - `order.by`: only applies to the HAC methods above. Defaults to `NULL` (the observations are chronologically ordered as in the original data). Optionally, you may specify a time index (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 **?** for more options.

Examples

1. Basic Example with First Differences

   Attach sample data:

   ```
   > data(macro)
   ```

Estimate model:

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

Summarize regression coefficients:

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

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

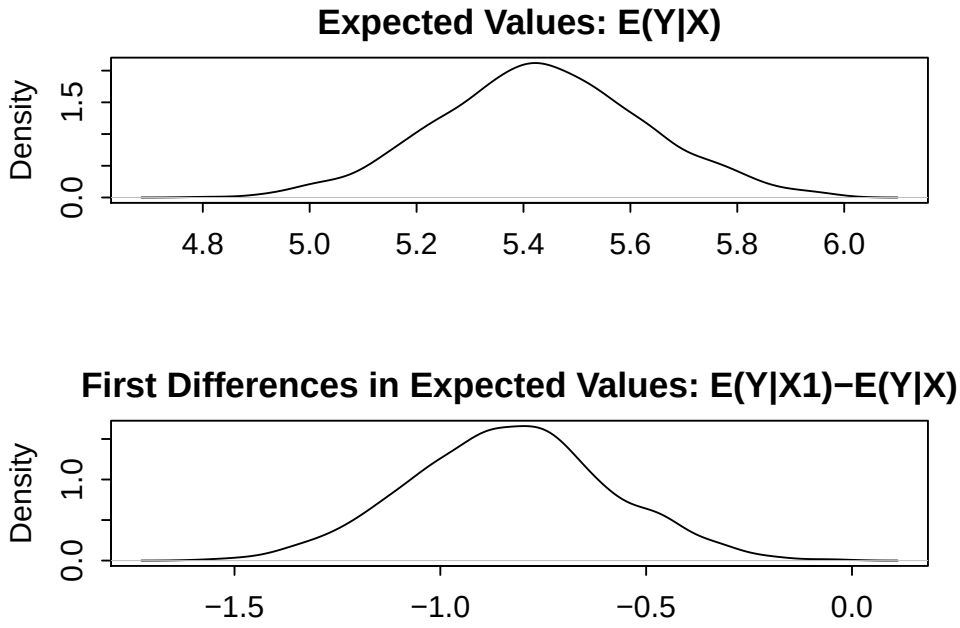
```
> x.high <- setx(z.out1, trade = quantile(macro$trade, 0.8))
> x.low <- setx(z.out1, trade = quantile(macro$trade, 0.2))
```

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)
```

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



2. Using Dummy Variables

   Estimate a model with fixed effects for each country (see Section **??** 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 variable into discrete levels.

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

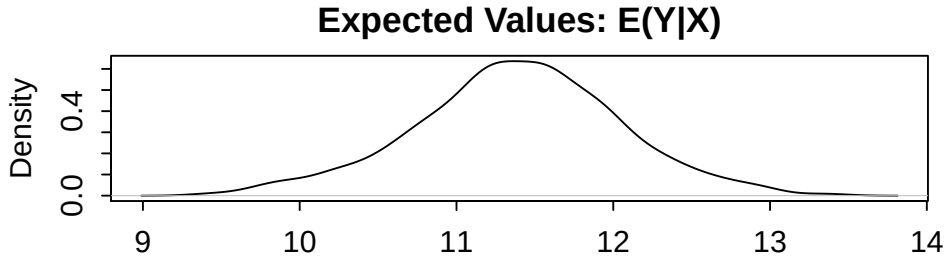
Set values for the explanatory variables, using the default mean/mode values, with country set to the United States and Japan, respectively:

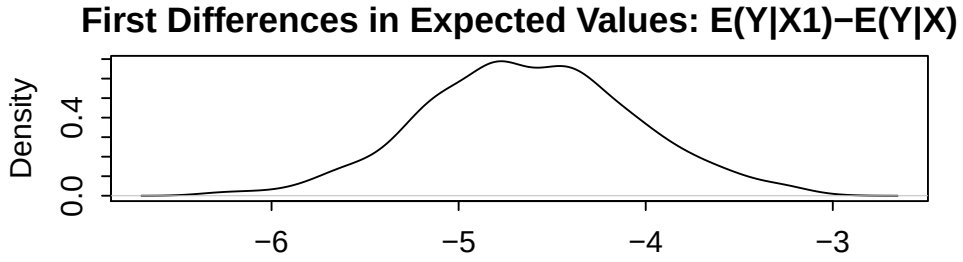
```
> x.US <- setx(z.out2, country = "United States")
> x.Japan <- setx(z.out2, country = "Japan")
```

Simulate quantities of interest:

```
> s.out2 <- sim(z.out2, x = x.US, x1 = x.Japan)
```

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





3. Multiple responses (least squares regression will be fitted separately to each dependent variable)

   Two responses for data set macro:

```
> z.out3 <- zelig(cbind(unem, gdp) ~ capmob + trade, model = "ls",
+     data = macro)
```

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

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

```
> x.US <- setx(z.out3, country = "United States")
> x.Japan <- setx(z.out3, country = "Japan")
```

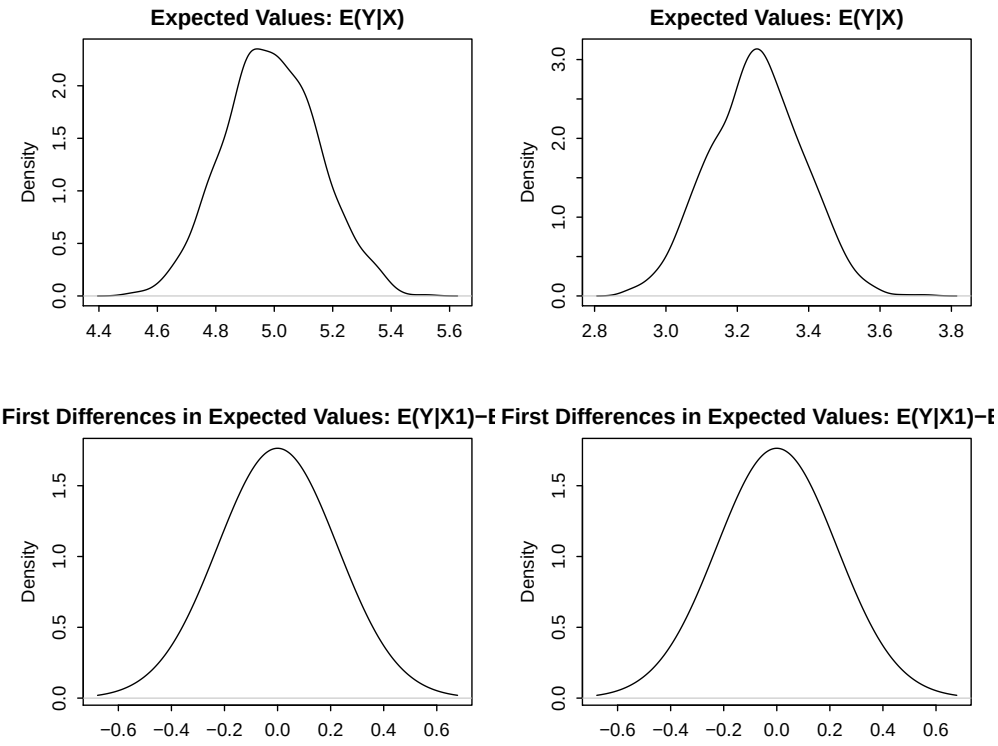
Simulate quantities of interest:

```
> s.out3 <- sim(z.out3, x = x.US, x1 = x.Japan)
```

Summary

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

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



Model

- The *stochastic component* is described by a density with mean μ_i and the common variance σ^2

$$Y_i \sim f(y_i \mid \mu_i, \sigma^2).$$

- The *systematic component* models the conditional mean as

$$\mu_i = x_i\beta$$

where x_i is the vector of covariates, and β is the vector of coefficients.

The least squares estimator is the best linear predictor of a dependent variable given x_i, and minimizes the sum of squared residuals, $\sum_{i=1}^{n}(Y_i - x_i\beta)^2$.

Quantities of Interest

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

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

given a draw of β from its sampling distribution.

- 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 $t_i = 0$.

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 = "ls", 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.
  - `df.residual`: the residual degrees of freedom.
  - `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.

$$\hat{\beta} \;=\; \left(\sum_{i=1}^{n} x_i' x_i\right)^{-1} \sum x_i y_i$$

  - `sigma`: the square root of the estimate variance of the random error e:

$$\hat{\sigma} \;=\; \frac{\sum (Y_i - x_i\hat{\beta})^2}{n-k}$$

  - `r.squared`: the fraction of the variance explained by the model.

$$R^2 \;=\; 1 - \frac{\sum (Y_i - x_i\hat{\beta})^2}{\sum (y_i - \bar{y})^2}$$

  - `adj.r.squared`: the above R^2 statistic, penalizing for an increased number of explanatory variables.
  - `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$ev`: the simulated expected values for the specified values of `x`.
  - `qi$fd`: the simulated first differences (or differences in expected values) for the specified values of `x` and `x1`.
  - `qi$att.ev`: the simulated average expected treatment effect for the treated from conditional prediction models.

How to Cite

To cite the *ls* Zelig model:

> Kosuke Imai, Gary King, and Oliva Lau. 2007. "ls: Least Squares 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

The least squares regression is part of the stats package by William N. Venables and Brian D. Ripley (**?**).In addition, advanced users may wish to refer to `help(lm)` and `help(lm.fit)`.Robust standard errors are implemented via the sandwich package by Achim Zeileis (**?**).Sample data are from **?**.