0.1 sim: Simulating Quantities of Interest

Description

Simulate quantities of interest from the estimated model output from `zelig()` given specified values of explanatory variables established in `setx()`. For classical *maximum likelihood* models, `sim()` uses asymptotic normal approximation to the log-likelihood. For *Bayesian models*, Zelig simulates quantities of interest from the posterior density, whenever possible. For *robust Bayesian models*, simulations are drawn from the identified class of Bayesian posteriors. Alternatively, you may generate quantities of interest using bootstrapped parameters.

Usage

```
s.out <- sim(object, x, x1 = NULL, num = c(1000, 100), prev = NULL,
             bootstrap = FALSE,  bootfn = NULL, ...)
```

Arguments

`object`	the output object from `zelig`.
`x`	values of explanatory variables used for simulation, generated by `setx`.
`x1`	optional values of explanatory variables (generated by a second call of `setx`), used to simulate first differences and risk ratios. (Not available for conditional prediction.)
`num`	the number of simulations, i.e., posterior draws. If the `num` argument is omitted, `sim` draws 1,000 simulations by if `bootstrap = FALSE` (the default), or 100 simulations if `bootstrap = TRUE`. You may increase this value to improve accuracy. (Not available for conditional prediction.)
`bootstrap`	a logical value indicating if parameters should be generated by re-fitting the model for bootstrapped data, rather than from the likelihood or posterior. (Not available for conditional prediction.)
`bootfn`	a function which governs how the data is sampled, re-fits the model, and returns the bootstrapped model parameters. If `bootstrap = TRUE` and `bootfn = NULL`, `sim` will sample observations from the original data (with replacement) until it creates a sampled dataset with the same number of observations as the original data. Alternative bootstrap methods include sampling the residuals rather than the observations, weighted sampling, and parametric bootstrapping. (Not available for conditional prediction.)
`...`	additional optional arguments passed to `boot`.

Value

The output stored in `s.out` varies by model. Use the `names` command to view the output stored in `s.out`. Common elements include: normal-bracket109bracket-normal

`x` the `setx` values for the explanatory variables, used to calculate the quantities of interest (expected values, predicted values, etc.).

`x1` the optional `setx` object used to simulate first differences, and other model-specific quantities of interest, such as risk-ratios.

`call` the options selected for `sim`, used to replicate quantities of interest.

`zelig.call` the original command and options for `zelig`, used to replicate analyses.

`num` the number of simulations requested.

`par` the parameters (coefficients, and additional model-specific parameters). You may wish to use the same set of simulated parameters to calculate quantities of interest rather than simulating another set.

`qi$ev` simulations of the expected values given the model and `x`.

`qi$pr` simulations of the predicted values given by the fitted values.

`qi$fd` simulations of the first differences (or risk difference for binary models) for the given `x` and `x1`. The difference is calculated by subtracting the expected values given `x` from the expected values given `x1`. (If do not specify `x1`, you will not get first differences or risk ratios.)

`qi$rr` simulations of the risk ratios for binary and multinomial models. See specific models for details.

`qi$ate.ev` simulations of the average expected treatment effect for the treatment group, using conditional prediction. Let t_i be a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups. Then the average expected treatment effect for the treatment group is

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

where $Y_i(t_i = 1)$ is the value of the dependent variable for observation i in the treatment group. 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 $t_i = 0$.

`qi$ate.pr` simulations of the average predicted treatment effect for the treatment group, using conditional prediction. Let t_i be a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups. Then

the average predicted treatment effect for the treatment group is

$$\frac{1}{n}\sum_{i=1}^{n}[\,Y_i(t_i=1) - \widehat{Y_i(t_i=0)} \mid t_i=1\,],$$

where $Y_i(t_i = 1)$ is the value of the dependent variable for observation i in the treatment group. 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 $t_i = 0$.

normal-bracket109bracket-normal

In the case of censored Y in the exponential, Weibull, and lognormal models, `sim` first imputes the uncensored values for Y before calculating the ATE.

You may use the `$` operator to extract any of the above from `s.out`. For example, `s.out$qi$ev` extracts the simulated expected values.

Author(s)

Kosuke Imai <⟨kimai@princeton.edu⟩>; Gary King <⟨king@harvard.edu⟩>; Olivia Lau <⟨olau@fas.harvard.edu⟩>

See Also

The full Zelig at `http://gking.harvard.edu/zelig`, and `boot`.