Zelig v4.0-10 Core Model Reference Manual

Matt Owen, Olivia Lau, Kosuke Imai, and Gary King

November 13, 2012

1 gamma: Gamma Regression for Continuous, Positive Dependent Variables

Use the gamma regression model if you have a positive-valued dependent variable such as the number of years a parliamentary cabinet endures, or the seconds you can stay airborne while jumping. The gamma distribution assumes that all waiting times are complete by the end of the study (censoring is not allowed).

1.1 Syntax

```
> z.out <- zelig(Y ~ X1 + X2, model = "gamma", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out, x1 = NULL)
```

1.2 Additional Inputs

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

- `robust`: defaults to `FALSE`. If `TRUE` is selected, `zelig()` computes robust standard errors via the `sandwich` package (see [16]). 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 [11].
  - `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 [16] for more options.

1.3 Example

Attach the sample data:

```
> data(coalition)
```

Estimate the model:

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

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

    polar


 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "gamma: Gamma Regression for Continuous, Positive Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig
```

View the regression output:

```
> summary(z.out)
```

```
Call:
glm(formula = formula, family = Gamma(), data = data, model = F)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2510  -0.9112  -0.2278   0.4132   1.5360

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.296e-02  1.329e-02  -0.975  0.33016
fract        1.149e-04  1.723e-05   6.668 1.19e-10 ***
numst2      -1.739e-02  5.881e-03  -2.957  0.00335 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for Gamma family taken to be 0.6291004)

    Null deviance: 300.74  on 313  degrees of freedom
Residual deviance: 272.19  on 311  degrees of freedom
AIC: 2428.1

Number of Fisher Scoring iterations: 6
```

Set the baseline values (with the ruling coalition in the minority) and the alternative values (with the ruling coalition in the majority) for X:

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

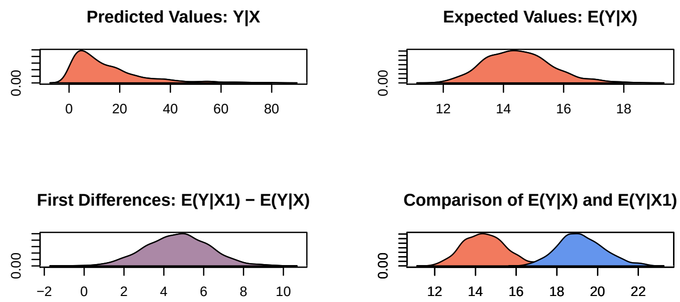
Simulate expected values (`qi$ev`) and first differences (`qi$fd`):

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

```
> summary(s.out)
```

```
Model:  gamma
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: E(Y|X)
   mean    sd    50%  2.5%  97.5%
 14.504 1.097 14.428 12.58 16.933

Expected Values (for X1): E(Y|X1)
   mean    sd    50%  2.5%  97.5%
 19.195 1.121 19.106 17.16 21.597

Predicted Values: Y|X
   mean     sd    50%  2.5% 97.5%
 14.802 13.928 10.706 0.616 55.51

Predicted Values: Y|X1
   mean     sd    50%  2.5%  97.5%
 19.848 18.458 14.961 1.208 73.676

First Differences: E(Y|X1) - E(Y|X)
  mean    sd   50%  2.5% 97.5%
 4.691 1.591 4.727 1.679 7.737
```

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



1.4 Model

- The Gamma distribution with scale parameter α has a *stochastic component*:

$$\begin{aligned} Y &\sim \text{Gamma}(y_i \mid \lambda_i, \alpha) \\ f(y) &= \frac{1}{\alpha^{\lambda_i}\,\Gamma\lambda_i}\, y_i^{\lambda_i - 1} \exp - \left\{ \frac{y_i}{\alpha} \right\} \end{aligned}$$

  for $\alpha, \lambda_i, y_i > 0$.

- The *systematic component* is given by

$$\lambda_i = \frac{1}{x_i \beta}$$

1.5 Quantities of Interest

- The expected values (`qi$ev`) are simulations of the mean of the stochastic component given draws of α and β from their posteriors:

$$E(Y) = \alpha\lambda_i.$$

- The predicted values (qi$pr) are draws from the gamma distribution for each given set of parameters (α, λ_i).

- If x1 is specified, sim() also returns the differences in the expected values (qi$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 $t_i = 0$.

1.6 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", 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`: the vector of fitted values.
- `linear.predictors`: the vector of $x_i \beta$.
- `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 a distribution defined by (α, λ_i).
  - `qi$fd`: the simulated first difference in the expected values for the specified values in `x` and `x1`.
  - `qi$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 Gamma Model

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 gamma model is part of the stats package by [14]. Advanced users may wish to refer to `help(glm)` and `help(family)`, as well as [13]. Robust standard errors are implemented via the sandwich package by [16]. Sample data are from [9].

2 `logit`: Logistic Regression for Dichotomous Dependent Variables

Logistic regression specifies a dichotomous dependent variable as a function of a set of explanatory variables.

2.1 Syntax

```
> z.out <- zelig(Y ~ X1 + X2, model = "logit", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out, x1 = NULL)
```

2.2 Additional Inputs

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

- `robust`: defaults to `FALSE`. If `TRUE` is selected, `zelig()` computes robust standard errors via the `sandwich` package (see [16]). 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 [11].
  - `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 [16] for more options.

2.3 Examples

1. Basic Example

   Attaching the sample turnout dataset:

```
> data(turnout)
```

   Estimating parameter values for the logistic regression:

```
> z.out1 <- zelig(vote ~ age + race,  model = "logit", data = turnout)
```

```
 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "logit: Logistic Regression for Dichotomous Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig

>
```

   Setting values for the explanatory variables:

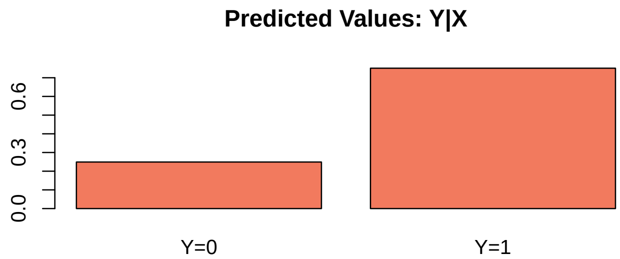
```
> x.out1 <- setx(z.out1, age = 36, race = "white")
```

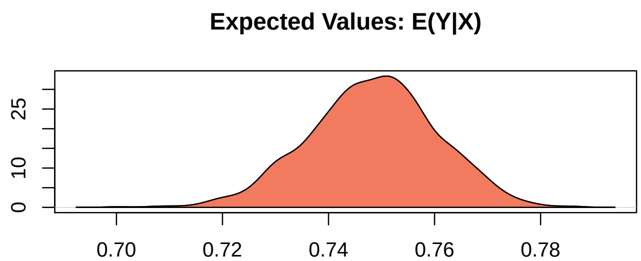
   Simulating quantities of interest from the posterior distribution.

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

> summary(s.out1)
```

```
Model:  logit
Number of simulations:  1000

Values of X
  (Intercept) age racewhite
1           1  36         1
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$race
[1] "contr.treatment"


Expected Values: E(Y|X)
  mean    sd   50%  2.5% 97.5%
 0.748 0.012 0.749 0.725 0.771

Predicted Values: Y|X
     0     1
 0.249 0.751
```

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



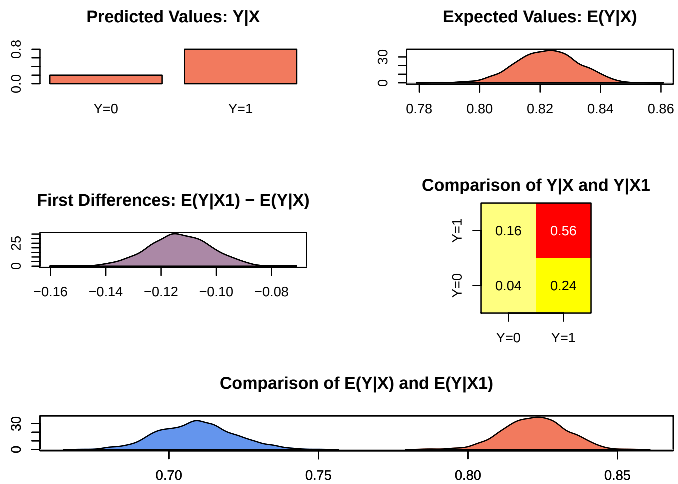


2. Simulating First Differences

   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.

```
> z.out2 <- zelig(vote ~ race + educate, model = "logit", data = turnout)

 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "logit: Logistic Regression for Dichotomous Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig

> x.high <- setx(z.out2, educate = quantile(turnout$educate, prob = 0.75))
> x.low <- setx(z.out2, educate = quantile(turnout$educate, prob = 0.25))

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

> summary(s.out2)
```

```
Model:  logit
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: E(Y|X)
  mean   sd   50%  2.5% 97.5%
 0.822 0.01 0.823 0.803 0.841

Expected Values (for X1): E(Y|X1)
  mean    sd   50%  2.5% 97.5%
 0.709 0.012 0.709 0.686 0.734

Predicted Values: Y|X
   0   1
 0.2 0.8

Predicted Values (for X1): Y|X1
     0     1
 0.287 0.713

First Differences: E(Y|X1) - E(Y|X)
   mean    sd    50%   2.5%  97.5%
 -0.113 0.011 -0.113 -0.136 -0.091

>  plot(s.out2)
```

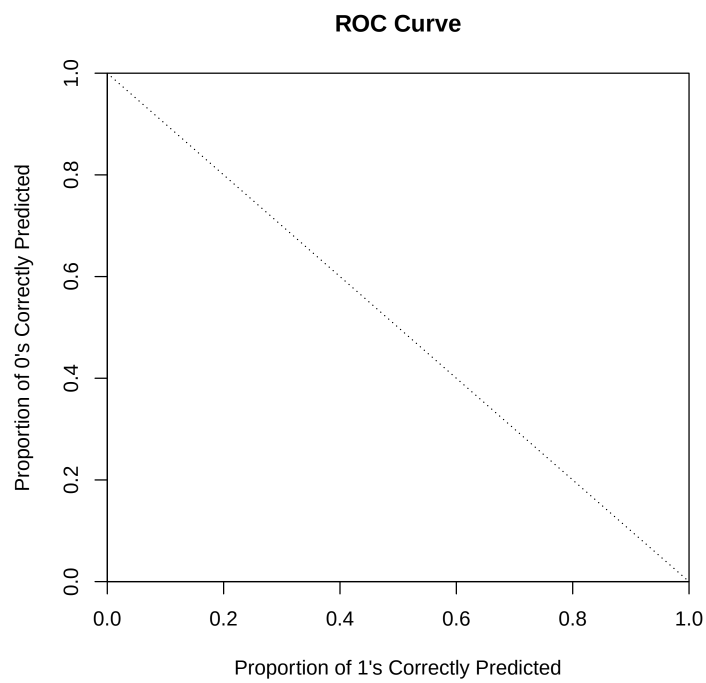


3. Presenting Results: An ROC Plot

   One can use an ROC plot to evaluate the fit of alternative model specifications. (Use `demo(roc)` to view this example, or see King and Zeng (2002).)

```
>  z.out1 <- zelig(vote ~ race + educate + age, model = "logit",
+                  data = turnout)

 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "logit: Logistic Regression for Dichotomous Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig

>  z.out2 <- zelig(vote ~ race + educate, model = "logit", data = turnout)

 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "logit: Logistic Regression for Dichotomous Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig
```

```
> rocplot(z.out1$y, z.out2$y, fitted(z.out1), fitted(z.out2))
```



2.4 Model

Let Y_i be the binary dependent variable for observation i which takes the value of either 0 or 1.

- The *stochastic component* is given by

$$\begin{aligned} Y_i &\sim \text{Bernoulli}(y_i \mid \pi_i) \\ &= \pi_i^{y_i}(1-\pi_i)^{1-y_i} \end{aligned}$$

  where $\pi_i = \Pr(Y_i = 1)$.

- The *systematic component* is given by:

$$\pi_i = \frac{1}{1+\exp(-x_i\beta)}.$$

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

2.5 Quantities of Interest

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

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

given draws of β from its sampling distribution.

- The predicted values (`qi$pr`) are draws from the Binomial distribution with mean equal to the simulated expected value π_i.

- The first difference (`qi$fd`) for the 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

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

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 $t_i = 0$.

2.6 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 = "logit", 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`: the vector of fitted values for the systemic component, π_i.
  - `linear.predictors`: the vector of $x_i \beta$
  - `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.
  - `data`: the name of the input data frame.
- 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$ev`: the simulated expected probabilities for the specified values of `x`.
  - `qi$pr`: the simulated predicted values for the specified values of `x`.
  - `qi$fd`: the simulated first difference in the expected probabilities for the values specified in `x` and `x1`.
  - `qi$rr`: the simulated risk ratio for the expected probabilities simulated from `x` and `x1`.
  - `qi$att.ev`: the simulated average expected treatment effect for the treated from conditional prediction models.
  - `qi$att.pr`: the simulated average predicted treatment effect for the treated from conditional prediction models.

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

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

3.1 Syntax

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

3.2 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 [16], [2], and [15]). 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 [1].
    - `"weave"`: HAC standard errors using the weights given in [11].
  - `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 [16] for more options.

3.3 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)

 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "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
```

Summarize regression coefficients:

```
>  summary(z.out1)

Call:
lm(formula = formula, data = data, weights = weights, model = F)

Residuals:
    Min      1Q  Median      3Q     Max
-5.3008 -2.0768 -0.3187  1.9789  7.7715

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.181294   0.450572  13.719  < 2e-16 ***
gdp         -0.323601   0.062820  -5.151 4.36e-07 ***
capmob       1.421939   0.166443   8.543 4.22e-16 ***
trade        0.019854   0.005606   3.542 0.000452 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 2.746 on 346 degrees of freedom
Multiple R-squared: 0.2878,        Adjusted R-squared: 0.2817
F-statistic: 46.61 on 3 and 346 DF,  p-value: < 2.2e-16
```

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)

Model:  ls
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: E(Y|X)
   mean    sd  50%  2.5% 97.5%
1 5.435 0.191 5.44 5.065 5.812

Expected Values (of X1): E(Y|X1)
   mean    sd   50% 2.5% 97.5%
1 4.597 0.184 4.593 4.23 4.968

First Difference in Expected Values: E(Y|X1) - E(Y|X)
    mean   sd    50%   2.5% 97.5%
1 -0.838 0.24 -0.848 -1.321 -0.34
```

2. Using Dummy Variables

   Estimate a model with fixed effects for each country. 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)
```

```
 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "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
```

   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)
```

3.4 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, $\sum_{i=1}^{n}(Y_i - x_i\beta)^2$.

3.5 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 $t_i = 0$.

3.6 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 = "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 the Least Squares Model

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 least squares regression is part of the stats package by William N. Venables and Brian D. Ripley [14].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 [16].Sample data are from [9].

How to Cite the Logit Model

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 logit model is part of the stats package by [14]. Advanced users may wish to refer to `help(glm)` and `help(family)`, as well as [13]. Robust standard errors are implemented via the sandwich package by [16]. Sample data are from [9].

4 `negbinom`: Negative Binomial Regression for Event Count Dependent Variables

Use the negative binomial regression if you have a count of events for each observation of your dependent variable. The negative binomial model is frequently used to estimate over-dispersed event count models.

4.1 Syntax

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

4.2 Additional Inputs

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

- `robust`: defaults to `FALSE`. If `TRUE` is selected, `zelig()` computes robust standard errors via the `sandwich` package (see [16]). 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 [11].
  - `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 [16] for more options.

4.3 Example

Load sample data:

```
>  data(sanction)
```

Estimate the model:

```
>  z.out <- zelig(num ~ target + coop, model = "negbinom", data = sanction)
```

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

    coop


 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "negbinom: Negative Binomial Regression for Event Count Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig
```

```
> summary(z.out)

Call:
glm.nb(formula = num ~ target + coop, data = Data.frame, weights = NULL,
    init.theta = 1.841603403, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0302  -0.5118  -0.1418  -0.0191   3.9987

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.5641     0.5941  -2.633   0.0103 *
target        0.1510     0.2262   0.668   0.5063
coop          1.2857     0.1761   7.302 2.51e-10 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for Negative Binomial(1.8416) family taken to be 1.520566)

    Null deviance: 237.094  on 77  degrees of freedom
Residual deviance:  56.545  on 75  degrees of freedom
AIC: 360.19

Number of Fisher Scoring iterations: 1
```

Set values for the explanatory variables to their default mean values:

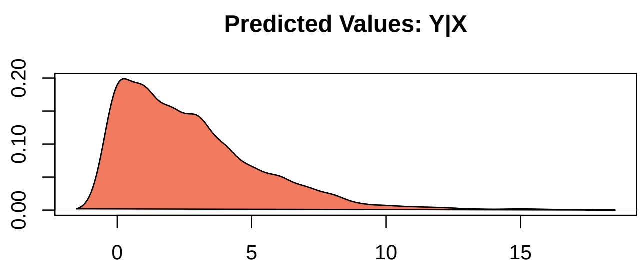
```
>  x.out <- setx(z.out)
```

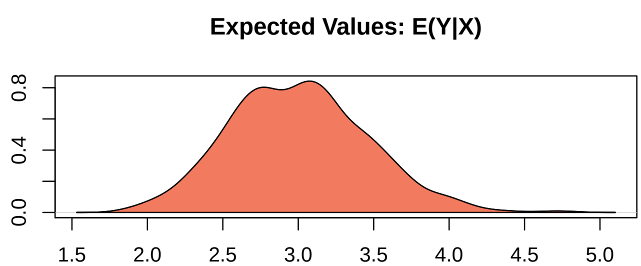
Simulate fitted values:

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

> summary(s.out)

Model:  negbinom
Number of simulations:  1000

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

Expected Values: E(Y|X)
  mean    sd   50%  2.5% 97.5%
 3.003 0.456 3.002 2.175 3.959
```

```
Predicted Values: Y|X
  mean   sd 50% 2.5% 97.5%
 2.732 2.65   2    0     9
```

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





4.4 Model

Let Y_i be the number of independent events that occur during a fixed time period. This variable can take any non-negative integer value.

- The negative binomial distribution is derived by letting the mean of the Poisson distribution vary according to a fixed parameter ζ given by the Gamma distribution. The *stochastic component* is given by

$$\begin{aligned} Y_i \mid \zeta_i &\sim \text{Poisson}(\zeta_i \mu_i), \\ \zeta_i &\sim \frac{1}{\theta}\text{Gamma}(\theta). \end{aligned}$$

  The marginal distribution of Y_i is then the negative binomial with mean

μ_i and variance $\mu_i + \mu_i^2/\theta$:

$$\begin{aligned} Y_i &\sim \text{NegBinom}(\mu_i, \theta), \\ &= \frac{\Gamma(\theta + y_i)}{y!\,\Gamma(\theta)} \frac{\mu_i^{y_i}\,\theta^\theta}{(\mu_i + \theta)^{\theta + y_i}}, \end{aligned}$$

where θ is the systematic parameter of the Gamma distribution modeling ζ_i.

- The *systematic component* is given by

$$\mu_i = \exp(x_i \beta)$$

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

4.5 Quantities of Interest

- The expected values (`qi$ev`) are simulations of the mean of the stochastic component. Thus,

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

given simulations of β.

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

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

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

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

4.6 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 = "negbinom", 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.
  - `theta`: the maximum likelihood estimate for the stochastic parameter θ.
  - `SE.theta`: the standard error for `theta`.
  - `residuals`: the working residuals in the final iteration of the IWLS fit.
  - `fitted.values`: a vector of the fitted values for the systemic component λ.
  - `linear.predictors`: a vector of $x_i\beta$.
  - `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$ev: the simulated expected values given the specified values of x.
- qi$pr: the simulated predicted values drawn from the distribution defined by (μ_i, θ).
- qi$fd: the simulated first differences in the simulated expected values given the specified values of x and x1.
- qi$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 Negative Binomial Model

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.

5 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 3). 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.

5.1 Syntax

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

5.2 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 [16]). 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 [11].
  - `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 [16] for more options.

5.3 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)
```

```
 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "normal: Normal Regression for Continuous Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig
```

   Summarize of regression coefficients:

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

```
Call:
glm(formula = formula, family = gaussian, data = data, weights = weights,
    model = F)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-5.3008  -2.0768  -0.3187   1.9789   7.7715

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.181294   0.450572  13.719  < 2e-16 ***
gdp         -0.323601   0.062820  -5.151 4.36e-07 ***
capmob       1.421939   0.166443   8.543 4.22e-16 ***
trade        0.019854   0.005606   3.542 0.000452 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for gaussian family taken to be 7.54307)

    Null deviance: 3664.8  on 349  degrees of freedom
Residual deviance: 2609.9  on 346  degrees of freedom
AIC: 1706.5

Number of Fisher Scoring iterations: 2
```

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))
```

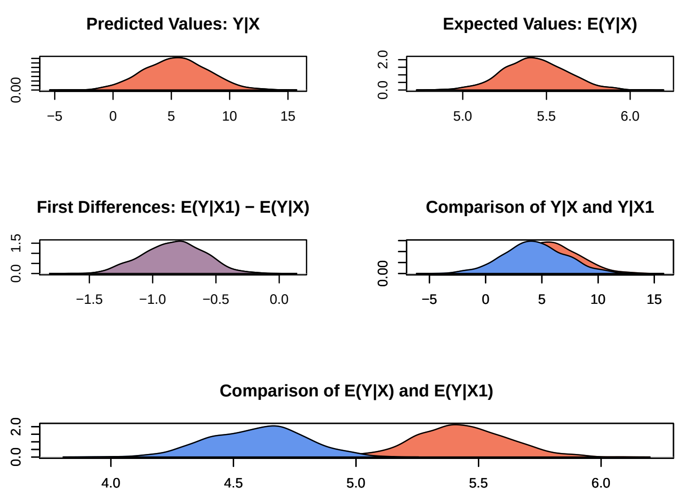
Generate first differences for the effect of high versus low trade on GDP:

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

>  summary(s.out1)

Model:  normal
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: E(Y|X)
```

```
  mean    sd   50%  2.5% 97.5%
 5.432 0.188 5.425 5.065  5.81

Expected Values (for X1): E(Y|X1)
  mean    sd   50%  2.5% 97.5%
 4.598 0.193 4.609 4.219 4.966

Predicted Values: Y|X
 mean    sd   50%  2.5% 97.5%
  5.4 2.716 5.414 0.194 10.62

Predicted Values (for X1): Y|X1
  mean    sd   50%   2.5% 97.5%
 4.362 2.741 4.298 -1.197 9.929

First Differences: E(Y|X1) - E(Y|X)
   mean    sd    50%   2.5%  97.5%
 -0.834 0.242 -0.826 -1.307 -0.372
```

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 factors 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)
```

```
 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "normal: Normal Regression for Continuous Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig
```

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

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

   Simulate quantities of interest:

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

> ###try(summary(s.out2))
```

5.4 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.

5.5 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

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

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 $t_i = 0$.

5.6 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

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 [14]. Advanced users may wish to refer to `help(glm)` and `help(family)`, as well as [13]. Robust standard errors are implemented via the sandwich package by [16]. Sample data are from [9].

6 `poisson`: Poisson Regression for Event Count Dependent Variables

Use the Poisson regression model if the observations of your dependent variable represents the number of independent events that occur during a fixed period of time (see the negative binomial model, Section 4, for over-dispersed event counts.).

6.1 Syntax

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

6.2 Additional Inputs

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

- `robust`: defaults to `FALSE`. If `TRUE` is selected, `zelig()` computes robust standard errors via the `sandwich` package (see [16]). 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 [11].
  - `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 [16] for more options.

6.3 Example

Load sample data:

```
> data(sanction)
```

Estimate Poisson model:

```
> z.out <- zelig(num ~ target + coop, model = "poisson", data = sanction)

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

    coop


 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "poisson: Poisson Regression for Event Count Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig

> summary(z.out)

Call:
glm(formula = formula, family = poisson(), data = data, weights = weights,
    model = F)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.2127  -1.1831  -0.2080  -0.1856  17.6514

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.96772    0.17545  -5.516 3.48e-08 ***
target      -0.02102    0.05823  -0.361    0.718
coop         1.21082    0.04662  25.970  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1583.77  on 77  degrees of freedom
```

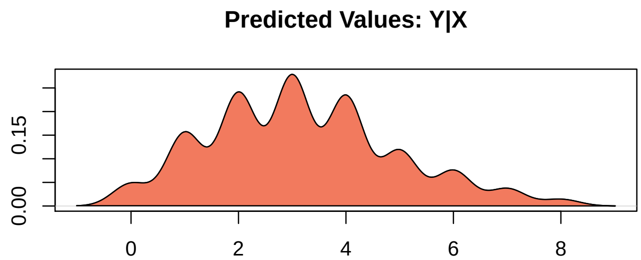
```
Residual deviance:  720.84  on 75  degrees of freedom
AIC: 944.35

Number of Fisher Scoring iterations: 6
```

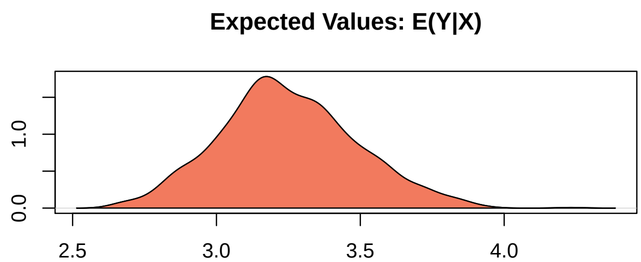
Set values for the explanatory variables to their default mean values:

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

Simulate fitted values:

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

> summary(s.out)

Model:  poisson
Number of simulations:  1000

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

Expected Values: E(Y|X)
  mean    sd  50% 2.5% 97.5%
 3.249 0.237 3.23 2.82 3.742

Predicted Values: Y|X
  mean    sd 50% 2.5% 97.5%
 3.184 1.732   3    0     7

>  plot(s.out)
```





6.4 Model

Let Y_i be the number of independent events that occur during a fixed time period. This variable can take any non-negative integer.

- The Poisson distribution has *stochastic component*

  $$Y_i \sim \text{Poisson}(\lambda_i),$$

  where λ_i is the mean and variance parameter.

- The *systematic component* is

  $$\lambda_i = \exp(x_i \beta),$$

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

6.5 Quantities of Interest

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

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

  given draws of β from its sampling distribution.

- The predicted value (qi$pr) is a random draw from the poisson distribution defined by mean λ_i.
- The first difference in the expected values (qi$fd) is given by:

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

- 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 $t_i = 0$.

6.6 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 = "poisson", 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`: a vector of the fitted values for the systemic component λ.
- `linear.predictors`: a vector of $x_i\beta$.
- `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$ev`: the simulated expected values given the specified values of `x`.
  - `qi$pr`: the simulated predicted values drawn from the distributions defined by λ_i.
  - `qi$fd`: the simulated first differences in the expected values given the specified values of `x` and `x1`.
  - `qi$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 Poisson Regression Model

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 poisson model is part of the stats package by [14]. Advanced users may wish to refer to `help(glm)` and `help(family)`, as well as [13]. Robust standard errors are implemented via the sandwich package by [16]. Sample data are from [12].

7 `probit`: Probit Regression for Dichotomous Dependent Variables

Use probit regression to model binary dependent variables specified as a function of a set of explanatory variables.

7.1 Syntax

```
> z.out <- zelig(Y ~ X1 + X2, model = "probit", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out, x1 = NULL)
```

7.2 Additional Inputs

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

- `robust`: defaults to `FALSE`. If `TRUE` is selected, `zelig()` computes robust standard errors via the `sandwich` package (see [16]). 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 [11].
  - `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 [16] for more options.

7.3 Examples

Attach the sample turnout dataset:

```
>  data(turnout)
```

Estimate parameter values for the probit regression:

```
>  z.out <- zelig(vote ~ race + educate,  model = "probit", data = turnout)

 How to cite this model in Zelig:
  Kosuke Imai, Gary King, and Olivia Lau. 2012.
  "probit: Probit Regression for Dichotomous Dependent Variables"
  in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software,"
  http://gking.harvard.edu/zelig

>  summary(z.out)

Call:
glm(formula = formula, family = binomial(link = "probit"), data = data,
    weights = weights, model = F)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2586  -0.8982   0.6712   0.7232   1.7045

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.725949   0.128635  -5.643 1.67e-08 ***
racewhite    0.299076   0.084648   3.533 0.000411 ***
educate      0.097119   0.009571  10.147  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2266.7  on 1999  degrees of freedom
Residual deviance: 2136.0  on 1997  degrees of freedom
AIC: 2142

Number of Fisher Scoring iterations: 4
```

Set values for the explanatory variables to their default values.

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

Simulate quantities of interest from the posterior distribution.

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

```
> summary(s.out)

Model:  probit
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: E(Y|X)
  mean    sd   50%  2.5% 97.5%
 0.772 0.011 0.772 0.751 0.794

Predicted Values: Y|X
    0    1
 0.24 0.76
```

7.4 Model

Let Y_i be the observed binary dependent variable for observation i which takes the value of either 0 or 1.

- The *stochastic component* is given by

$$Y_i \sim \text{Bernoulli}(\pi_i),$$

  where $\pi_i = \Pr(Y_i = 1)$.

- The *systematic component* is

$$\pi_i = \Phi(x_i \beta)$$

  where $\Phi(\mu)$ is the cumulative distribution function of the Normal distribution with mean 0 and unit variance.

7.5 Quantities of Interest

- The expected value (`qi$ev`) is a simulation of predicted probability of success

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

  given a draw of β from its sampling distribution.

- The predicted value (`qi$pr`) is a draw from a Bernoulli distribution with mean π_i.

- The first difference (`qi$fd`) in expected values 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

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

  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 $t_i = 0$.

7.6 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 = "probit", 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`: a vector of the in-sample fitted values.
- `linear.predictors`: a vector of $x_i\beta$.
- `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.
- `data`: the name of the input data frame.

- 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$ev`: the simulated expected values, or predicted probabilities, for the specified values of `x`.
  - `qi$pr`: the simulated predicted values drawn from the distributions defined by the predicted probabilities.
  - `qi$fd`: the simulated first differences in the predicted probabilities for the values specified in `x` and `x1`.
  - `qi$rr`: the simulated risk ratio for the predicted probabilities simulated from `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 Logit Model

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 probit model is part of the stats package by [14]. Advanced users may wish to refer to `help(glm)` and `help(family)`, as well as [13]. Robust standard errors are implemented via the sandwich package by [16]. Sample data are from [9].

8 Bibliography

References

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

[2] Peter J. Huber. *Robust Statistics*. Wiley, 1981.

[3] Kosuke Imai, Olivia Lau, and Gary King. *logit: Logistic Regression for Dichotomous Dependent*, 2011.

[4] Kosuke Imai, Olivia Lau, and Gary King. *ls: Least Squares Regression for Continuous Dependent Variables*, 2011.

[5] Kosuke Imai, Olivia Lau, and Gary King. *negbinom: Negative Binomial Regression for Event Count Dependent Variables*, 2011.

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

[7] Kosuke Imai, Olivia Lau, and Gary King. *poisson: Poisson Regression for Event Count Dependent Variables*, 2011.

[8] Kosuke Imai, Olivia Lau, and Gary King. *probit: Probit Regression for Dichotomous Dependent Variables*, 2011.

[9] 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.

[10] Gary King and Langche Zeng. Improving forecasts of state failure. *World Politics*, 53(4):623–658, July 2002. http://gking.harvard.edu/files/abs/civil-abs.shtml.

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

[12] 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.

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

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

[15] Halbert White. A heteroscedastic-consistent covariance matrix estimator and a direct test for heteroscedasticity. *Econometrica*, 48(4):817–838, 1980.

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