1 `blogit`: Bivariate Logistic Regression for Two Dichotomous Dependent Variables

Use the bivariate logistic regression model if you have two binary dependent variables (Y_1, Y_2), and wish to model them jointly as a function of some explanatory variables. Each pair of dependent variables (Y_{i1}, Y_{i2}) has four potential outcomes, $(Y_{i1} = 1, Y_{i2} = 1)$, $(Y_{i1} = 1, Y_{i2} = 0)$, $(Y_{i1} = 0, Y_{i2} = 1)$, and $(Y_{i1} = 0, Y_{i2} = 0)$. The joint probability for each of these four outcomes is modeled with three systematic components: the marginal $\Pr(Y_{i1} = 1)$ and $\Pr(Y_{i2} = 1)$, and the odds ratio ψ, which describes the dependence of one marginal on the other. Each of these systematic components may be modeled as functions of (possibly different) sets of explanatory variables.

1.0.1 Syntax

```
> z.out <- zelig(list(mu1 = Y1 ~ X1 + X2 ,
                      mu2 = Y2 ~ X1 + X3),
                model = "blogit", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

1.0.2 Input Values

In every bivariate logit specification, there are three equations which correspond to each dependent variable (Y_1, Y_2), and ψ, the odds ratio. You should provide a list of formulas for each equation or, you may use `cbind()` if the right hand side is the same for both equations

```
> formulae <- list(cbind(Y1,Y2) ~ X1 + X2)
```

which means that all the explanatory variables in equations 1 and 2 (corresponding to Y_1 and Y_2) are included, but only an intercept is estimated (all explanatory variables are omitted) for equation 3 (ψ).

You may use the function `tag()` to constrain variables across equations:

```
> formulae <- list(mu1 = y1 ~ x1 + tag(x3, "x3"),
+                   mu2 = y2 ~ x2 + tag(x3, "x3"))
```

where `tag()` is a special function that constrains variables to have the same effect across equations. Thus, the coefficient for `x3` in equation `mu1` is constrained to be equal to the coefficient for `x3` in equation `mu2`.

1.0.3 Examples

1. Basic Example

   Load the data and estimate the model:

```
>  data(sanction)
> ## sanction

>  z.out1 <- zelig(cbind(import, export) ~ coop + cost + target,
+                    model = "blogit", data = sanction)
```

By default, **zelig()** estimates two effect parameters for each explana-
tory variable in addition to the odds ratio parameter; this formulation
is parametrically independent (estimating unconstrained effects for each
explanatory variable), but stochastically dependent because the models
share an odds ratio.

Generate baseline values for the explanatory variables (with cost set to
1, net gain to sender) and alternative values (with cost set to 4, major loss
to sender):

```
>  x.low <- setx(z.out1, cost = 1)

> x.high <- setx(z.out1, cost = 4)
```
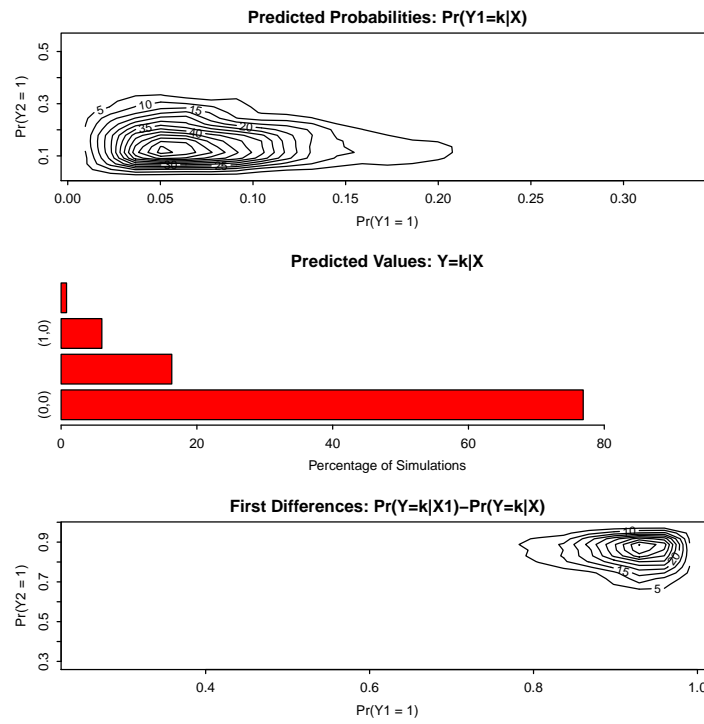
Simulate fitted values and first differences:

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

>  plot(s.out1)
```

Predicted Probabilities: Pr(Y1=k|X)

Predicted Values: Y=k|X

First Differences: Pr(Y=k|X1)–Pr(Y=k|X)

2. Joint Estimation of a Model with Different Sets of Explanatory Variables

   Using sample data `sanction`, estimate the statistical model, with `import` a function of `coop` in the first equation and `export` a function of `cost` and `target` in the second equation:

   ```
   > z.out2 <- zelig(list(import ~ coop, export ~ cost + target),
   +                 model = "blogit", data = sanction)
   > summary(z.out2)
   ```
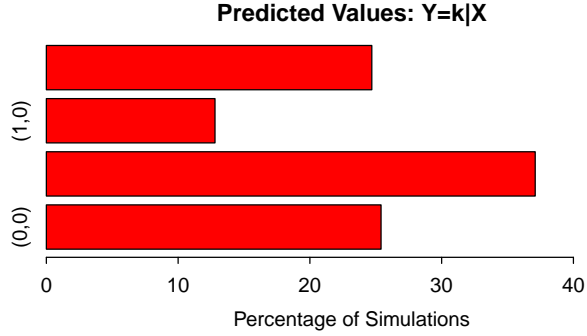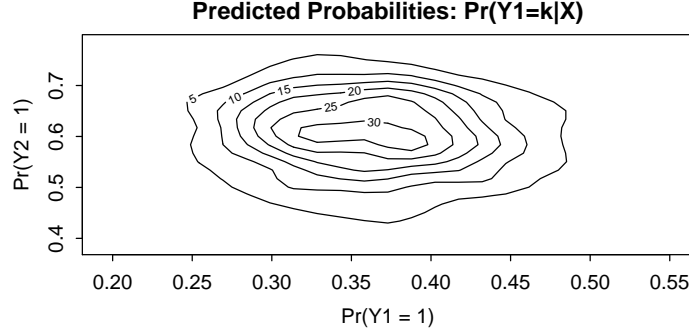
   Set the explanatory variables to their means:

   ```
   > x.out2 <- setx(z.out2)
   ```

   Simulate draws from the posterior distribution:

   ```
   > s.out2 <- sim(z.out2, x = x.out2)
   > summary(s.out2)

   > plot(s.out2)
   ```

Predicted Probabilities: Pr(Y1=k|X)

Predicted Values: Y=k|X

1.0.4 Model

For each observation, define two binary dependent variables, Y_1 and Y_2, each of which take the value of either 0 or 1 (in the following, we suppress the observation index). We model the joint outcome (Y_1, Y_2) using a marginal probability for each dependent variable, and the odds ratio, which parameterizes the relationship between the two dependent variables. Define Y_{rs} such that it is equal to 1 when $Y_1 = r$ and $Y_2 = s$ and is 0 otherwise, where r and s take a value of either 0 or 1. Then, the model is defined as follows,

- The *stochastic component* is

$$
\begin{aligned}
Y_{11} &\sim \text{Bernoulli}(y_{11} \mid \pi_{11}) \\
Y_{10} &\sim \text{Bernoulli}(y_{10} \mid \pi_{10}) \\
Y_{01} &\sim \text{Bernoulli}(y_{01} \mid \pi_{01})
\end{aligned}
$$

  where $\pi_{rs} = \Pr(Y_1 = r, Y_2 = s)$ is the joint probability, and $\pi_{00} = 1 - \pi_{11} - \pi_{10} - \pi_{01}$.

- The *systematic components* model the marginal probabilities, $\pi_j = \Pr(Y_j = 1)$, as well as the odds ratio. The odds ratio is defined as $\psi = \pi_{00}\pi_{01}/\pi_{10}\pi_{11}$ and describes the relationship between the two outcomes. Thus, for each

4

observation we have

$$\pi_j = \frac{1}{1 + \exp(-x_j\beta_j)} \quad \text{for} \quad j = 1, 2,$$
$$\psi = \exp(x_3\beta_3).$$

1.0.5 Quantities of Interest

- The expected values (`qi$ev`) for the bivariate logit model are the predicted joint probabilities. Simulations of β_1, β_2, and β_3 (drawn from their sampling distributions) are substituted into the systematic components (π_1, π_2, ψ) to find simulations of the predicted joint probabilities:

$$\pi_{11} = \begin{cases} \frac{1}{2}(\psi - 1)^{-1} - a - \sqrt{a^2 + b} & \text{for } \psi \neq 1 \\ \pi_1\pi_2 & \text{for } \psi = 1 \end{cases},$$
$$\pi_{10} = \pi_1 - \pi_{11},$$
$$\pi_{01} = \pi_2 - \pi_{11},$$
$$\pi_{00} = 1 - \pi_{10} - \pi_{01} - \pi_{11},$$

where $a = 1 + (\pi_1 + \pi_2)(\psi - 1)$, $b = -4\psi(\psi - 1)\pi_1\pi_2$, and the joint probabilities for each observation must sum to one. For n simulations, the expected values form an $n \times 4$ matrix for each observation in `x`.

- The predicted values (`qi$pr`) are draws from the multinomial distribution given the expected joint probabilities.

- The first differences (`qi$fd`) for each of the predicted joint probabilities are given by

$$\text{FD}_{rs} = \Pr(Y_1 = r, Y_2 = s \mid x_1) - \Pr(Y_1 = r, Y_2 = s \mid x).$$

- The risk ratio (`qi$rr`) for each of the predicted joint probabilities are given by

$$\text{RR}_{rs} = \frac{\Pr(Y_1 = r, Y_2 = s \mid x_1)}{\Pr(Y_1 = r, Y_2 = s \mid x)}$$

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

$$\frac{1}{\sum_{i=1}^{n} t_i} \sum_{i:t_i=1}^{n} \{Y_{ij}(t_i = 1) - E[Y_{ij}(t_i = 0)]\} \text{ for } j = 1, 2,$$

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 $E[Y_{ij}(t_i = 0)]$, the counterfactual expected value of Y_{ij} 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

$$\frac{1}{\sum_{i=1}^{n} t_i} \sum_{i:t_i=1}^{n} \left\{ Y_{ij}(t_i=1) - Y_{ij}\widehat{(t_i=0)} \right\} \text{ for } j = 1, 2,$$

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 $Y_{ij}\widehat{(t_i=0)}$, the counterfactual predicted value of Y_{ij} 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 = "blogit", data)`, then you may examine the available information in `z.out` by using `names(z.out)`, see the `coefficients` by using `z.out$coefficients`, and obtain a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output object `z.out`, you may extract:

  - `coefficients`: the named vector of coefficients.
  - `fitted.values`: an $n \times 4$ matrix of the in-sample fitted values.
  - `predictors`: an $n \times 3$ matrix of the linear predictors $x_j \beta_j$.
  - `residuals`: an $n \times 3$ matrix of the residuals.
  - `df.residual`: the residual degrees of freedom.
  - `df.total`: the total degrees of freedom.
  - `rss`: the residual sum of squares.
  - `y`: an $n \times 2$ matrix of the dependent variables.
  - `zelig.data`: the input data frame if `save.data = TRUE`.

- From `summary(z.out)`, you may extract:

  - `coef3`: a table of the coefficients with their associated standard errors and t-statistics.
  - `cov.unscaled`: the variance-covariance matrix.
  - `pearson.resid`: an $n \times 3$ matrix of the Pearson residuals.

- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as arrays indexed by simulation × quantity × x-observation (for more than one x-observation; otherwise the quantities are matrices). Available quantities are:

6

- `qi$ev`: the simulated expected joint probabilities (or expected values) for the specified values of `x`.
- `qi$pr`: the simulated predicted outcomes drawn from a distribution defined by the expected joint probabilities.
- `qi$fd`: the simulated first difference in the expected joint probabilities for the values specified in `x` and `x1`.
- `qi$rr`: the simulated risk ratio in the predicted probabilities for given `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 Bivariate 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 bivariate logit function is part of the VGAM package by Thomas Yee [7]. In addition, advanced users may wish to refer to `help(vglm)` in the VGAM library. Additional documentation is available at http://www.stat.auckland.ac.nz/~ yee.Sample data are from [4]

2   `bprobit`: Bivariate Probit Regression for Two Dichotomous Dependent Variables

Use the bivariate probit regression model if you have two binaryrun dependent variables (Y_1, Y_2), and wish to model them jointly as a function of some explanatory variables. Each pair of dependent variables (Y_{i1}, Y_{i2}) has four potential outcomes, $(Y_{i1} = 1, Y_{i2} = 1)$, $(Y_{i1} = 1, Y_{i2} = 0)$, $(Y_{i1} = 0, Y_{i2} = 1)$, and $(Y_{i1} = 0, Y_{i2} = 0)$. The joint probability for each of these four outcomes is modeled with three systematic components: the marginal $\Pr(Y_{i1} = 1)$ and $\Pr(Y_{i2} = 1)$, and the correlation parameter ρ for the two marginal distributions. Each of these systematic components may be modeled as functions of (possibly different) sets of explanatory variables.

2.0.7 Syntax

```
> z.out <- zelig(list(mu1 = Y1 ~ X1 + X2,
                      mu2 = Y2 ~ X1 + X3,
                      rho = ~ 1),
                 model = "bprobit", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

2.0.8 Input Values

In every bivariate probit specification, there are three equations which correspond to each dependent variable (Y_1, Y_2), and the correlation parameter ρ. Since the correlation parameter does not correspond to one of the dependent variables, the model estimates ρ as a constant by default. Hence, only two formulas (for μ_1 and μ_2) are required. If the explanatory variables for μ_1 and μ_2 are the same and effects are estimated separately for each parameter, you may use the following short hand:

```
> fml <- list(cbind(Y1,Y2) ~ X1 + X2)
```

which has the same meaning as:

```
> fml <- list(mu1 = Y1 ~ X1 + X2,
+             mu2 = Y2 ~ X1 + X2,
+             rho = ~ 1)
```

You may use the function `tag()` to constrain variables across equations. The `tag()` function takes a variable and a label for the effect parameter. Below, the constrained effect of `x3` in both equations is called the `age` parameter:

```
> fml <- list(mu1 = y1 ~ x1 + tag(x3, "age"),
+             mu2 = y2 ~ x2 + tag(x3, "age"))
```

You may also constrain different variables across different equations to have the same effect.

2.0.9 Examples

1. Basic Example

   Load the data and estimate the model:

   ```
   > data(sanction)
   ```

   ```
   > z.out1 <- zelig(cbind(import, export) ~ coop + cost + target,
   +                 model = "bprobit", data = sanction)
   ```

   By default, `zelig()` estimates two effect parameters for each explanatory variable in addition to the correlation coefficient; this formulation

is parametrically independent (estimating unconstrained effects for each explanatory variable), but stochastically dependent because the models share a correlation parameter.
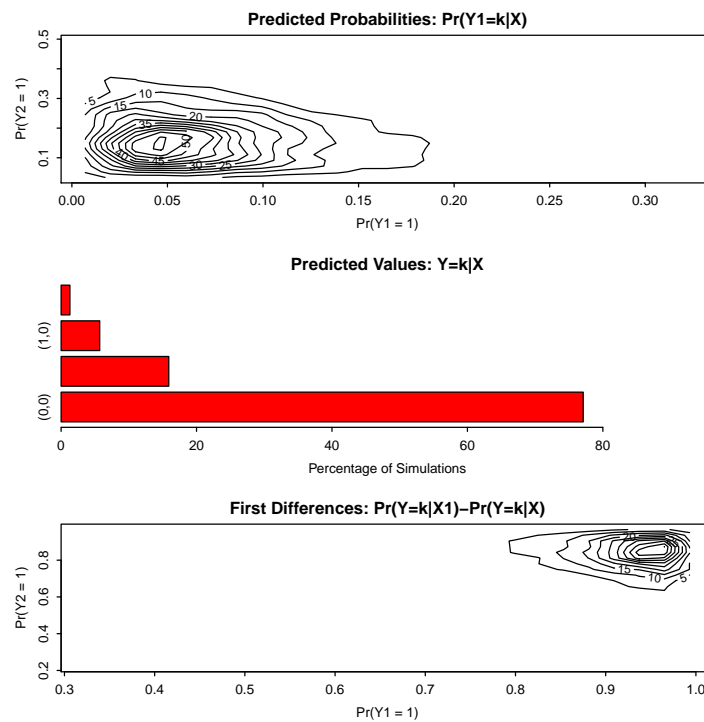
Generate baseline values for the explanatory variables (with cost set to 1, net gain to sender) and alternative values (with cost set to 4, major loss to sender):

```
>  x.low <- setx(z.out1, cost = 1)
>  x.high <- setx(z.out1, cost = 4)
```

Simulate fitted values and first differences:

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


>  plot(s.out1)
```


2. Joint Estimation of a Model with Different Sets of Explanatory Variables

Using the sample data sanction, estimate the statistical model, with import a function of coop in the first equation and export a function of cost and target in the second equation:

9

```
>  fml2 <- list(mu1 = import ~ coop,
+               mu2 = export ~ cost + target)

>  z.out2 <- zelig(fml2, model = "bprobit", data = sanction)
>  summary(z.out2)
```
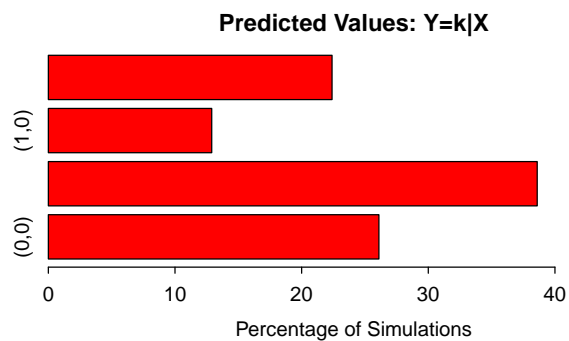
Set the explanatory variables to their means:

```
>  x.out2 <- setx(z.out2)
```
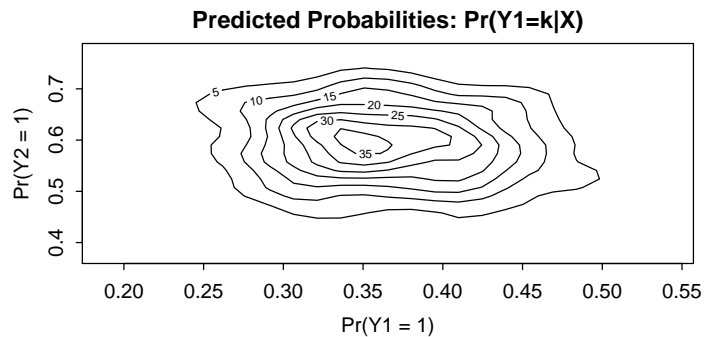
Simulate draws from the posterior distribution:

```
>  s.out2 <- sim(z.out2, x = x.out2)
>  summary(s.out2)

>  plot(s.out2)
```


2.0.10   Model

For each observation, define two binary dependent variables, Y_1 and Y_2, each of
which take the value of either 0 or 1 (in the following, we suppress the observation
index i). We model the joint outcome (Y_1, Y_2) using two marginal probabilities
for each dependent variable, and the correlation parameter, which describes how
the two dependent variables are related.

10

- The *stochastic component* is described by two latent (unobserved) continuous variables which follow the bivariate Normal distribution:

$$
\begin{pmatrix} Y_1^* \\ Y_2^* \end{pmatrix} \quad \sim \quad N_2 \left\{ \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right\},
$$

where μ_j is a mean for Y_j^* and ρ is a scalar correlation parameter. The following observation mechanism links the observed dependent variables, Y_j, with these latent variables

$$
Y_j \quad = \quad \begin{cases} 1 & \text{if } Y_j^* \geq 0, \\ 0 & \text{otherwise.} \end{cases}
$$

- The *systemic components* for each observation are

$$
\begin{aligned}
\mu_j &= x_j \beta_j \quad \text{for} \quad j = 1, 2, \\
\rho &= \frac{\exp(x_3 \beta_3) - 1}{\exp(x_3 \beta_3) + 1}.
\end{aligned}
$$

2.0.11 Quantities of Interest

For n simulations, expected values form an $n \times 4$ matrix.

- The expected values (`qi$ev`) for the binomial probit model are the predicted joint probabilities. Simulations of β_1, β_2, and β_3 (drawn form their sampling distributions) are substituted into the systematic components, to find simulations of the predicted joint probabilities $\pi_{rs} = \Pr(Y_1 = r, Y_2 = s)$:

$$
\begin{aligned}
\pi_{11} &= \Pr(Y_1^* \geq 0, Y_2^* \geq 0) &= \int_0^\infty \int_0^\infty \phi_2(\mu_1, \mu_2, \rho) \, dY_2^* \, dY_1^* \\
\pi_{10} &= \Pr(Y_1^* \geq 0, Y_2^* < 0) &= \int_0^\infty \int_{-\infty}^0 \phi_2(\mu_1, \mu_2, \rho) \, dY_2^* \, dY_1^* \\
\pi_{01} &= \Pr(Y_1^* < 0, Y_2^* \geq 0) &= \int_{-\infty}^0 \int_0^\infty \phi_2(\mu_1, \mu_2, \rho) \, dY_2^* \, dY_1^* \\
\pi_{11} &= \Pr(Y_1^* < 0, Y_2^* < 0) &= \int_{-\infty}^0 \int_{-\infty}^0 \phi_2(\mu_1, \mu_2, \rho) \, dY_2^* \, dY_1^*
\end{aligned}
$$

  where r and s may take a value of either 0 or 1, ϕ_2 is the bivariate Normal density.

- The predicted values (`qi$pr`) are draws from the multinomial distribution given the expected joint probabilities.

- The first difference (`qi$fd`) in each of the predicted joint probabilities are given by

$$
\text{FD}_{rs} = \Pr(Y_1 = r, Y_2 = s \mid x_1) - \Pr(Y_1 = r, Y_2 = s \mid x).
$$

- The risk ratio (qi$rr) for each of the predicted joint probabilities are given by
$$\text{RR}_{rs} = \frac{\Pr(Y_1 = r, Y_2 = s \mid x_1)}{\Pr(Y_1 = r, Y_2 = s \mid x)}.$$

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

$$\frac{1}{\sum_{i=1}^{n} t_i} \sum_{i:t_i=1}^{n} \{Y_{ij}(t_i = 1) - E[Y_{ij}(t_i = 0)]\} \text{ for } j = 1, 2,$$

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 $E[Y_{ij}(t_i = 0)]$, the counterfactual expected value of Y_{ij} 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

$$\frac{1}{\sum_{i=1}^{n} t_i} \sum_{i:t_i=1}^{n} \left\{ Y_{ij}(t_i = 1) - Y_{ij}\widehat{(t_i = 0)} \right\} \text{ for } j = 1, 2,$$

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 $Y_{ij}\widehat{(t_i = 0)}$, the counterfactual predicted value of Y_{ij} for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $t_i = 0$.

2.0.12 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 = "bprobit", 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 obtain 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: the named vector of coefficients.
    - fitted.values: an $n \times 4$ matrix of the in-sample fitted values.
    - predictors: an $n \times 3$ matrix of the linear predictors $x_j \beta_j$.
    - residuals: an $n \times 3$ matrix of the residuals.

- df.residual: the residual degrees of freedom.
- df.total: the total degrees of freedom.
- rss: the residual sum of squares.
- y: an $n \times 2$ matrix of the dependent variables.
- zelig.data: the input data frame if save.data = TRUE.

- From summary(z.out), you may extract:

    - coef3: a table of the coefficients with their associated standard errors and t-statistics.
    - cov.unscaled: the variance-covariance matrix.
    - pearson.resid: an $n \times 3$ matrix of the Pearson residuals.

- From the sim() output object s.out, you may extract quantities of interest arranged as arrays indexed by simulation $\times$ quantity $\times$ x-observation (for more than one x-observation; otherwise the quantities are matrices). Available quantities are:

    - qi$ev: the simulated expected values (joint predicted probabilities) for the specified values of x.
    - qi$pr: the simulated predicted outcomes drawn from a distribution defined by the joint predicted probabilities.
    - qi$fd: the simulated first difference in the predicted probabilities for the values specified in x and x1.
    - qi$rr: the simulated risk ratio in the predicted probabilities for given 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 Bivariate Probit 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 bivariate probit function is part of the VGAM package by Thomas Yee [7].
In addition, advanced users may wish to refer to `help(vglm)` in the VGAM library. Additional documentation is available at http://www.stat.auckland.ac.nz/˜ yee.Sample data are from [4]

3 `ologit`: Ordinal Logistic Regression for Ordered Categorical Dependent Variables

Use the ordinal logit regression model if your dependent variable is ordered and categorical, either in the form of integer values or character strings.

3.0.13 Syntax

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

If `Y` takes discrete integer values, the `as.factor()` command will order automatically order the values. If `Y` takes on values composed of character strings, such as "strongly agree", "agree", and "disagree", `as.factor()` will order the values in the order in which they appear in `Y`. You will need to replace your dependent variable with a factored variable prior to estimating the model through `zelig()`. See Example 1 for more information on creating ordered factors.

3.0.14 Example

1. Creating An Ordered Dependent Variable

   Load the sample data:

   ```
   > data(sanction)
   ```

   Create an ordered dependent variable:

   ```
   > sanction$ncost <- factor(sanction$ncost, ordered = TRUE,
   +                          levels = c("net gain", "little effect",
   +                          "modest loss", "major loss"))
   ```

   Estimate the model:

   ```
   > z.out <- zelig(ncost ~ mil + coop, model = "ologit", data = sanction)
   ```

   Set the explanatory variables to their observed values:

   ```
   > x.out <- setx(z.out, fn = NULL)
   ```

   Simulate fitted values given `x.out` and view the results:

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

>  summary(s.out)
```

2. First Differences

   Using the sample data `sanction`, estimate the empirical model and re-
   turning the coefficients:

```
>  z.out <- zelig(as.factor(cost) ~ mil + coop, model = "ologit",
+                  data = sanction)

> summary(z.out)
```

   Set the explanatory variables to their means, with `mil` set to 0 (no military
   action in addition to sanctions) in the baseline case and set to 1 (military
   action in addition to sanctions) in the alternative case:

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

   Generate simulated fitted values and first differences, and view the results:

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

3.0.15 Model

Let Y_i be the ordered categorical dependent variable for observation i that takes
one of the integer values from 1 to J where J is the total number of categories.

- The *stochastic component* begins with an unobserved continuous variable,
  Y_i^*, which follows the standard logistic distribution with a parameter μ_i,

$$Y_i^* \sim \text{Logit}(y_i^* \mid \mu_i),$$

  to which we add an observation mechanism

$$Y_i = j \quad \text{if} \quad \tau_{j-1} \le Y_i^* \le \tau_j \quad \text{for} \quad j = 1, \ldots, J.$$

  where τ_l (for $l = 0, \ldots, J$) are the threshold parameters with $\tau_l < \tau_m$ for
  all $l < m$ and $\tau_0 = -\infty$ and $\tau_J = \infty$.

- The *systematic component* has the following form, given the parameters
  τ_j and β, and the explanatory variables x_i:

$$\Pr(Y \le j) = \Pr(Y^* \le \tau_j) = \frac{\exp(\tau_j - x_i\beta)}{1 + \exp(\tau_j - x_i\beta)},$$

  which implies:

$$\pi_j = \frac{\exp(\tau_j - x_i\beta)}{1 + \exp(\tau_j - x_i\beta)} - \frac{\exp(\tau_{j-1} - x_i\beta)}{1 + \exp(\tau_{j-1} - x_i\beta)}.$$

15

3.0.16 Quantities of Interest

- The expected values (`qi$ev`) for the ordinal logit model are simulations of the predicted probabilities for each category:

$$E(Y = j) \; = \; \pi_j \; = \; \frac{\exp(\tau_j - x_i\beta)}{1 + \exp(\tau_j - x_i\beta)} - \frac{\exp(\tau_{j-1} - x_i\beta)}{1 + \exp(\tau_{j-1} - x_i\beta)},$$

given a draw of β from its sampling distribution.

- The predicted value (`qi$pr`) is drawn from the logit distribution described by μ_i, and observed as one of J discrete outcomes.

- The difference in each of the predicted probabilities (`qi$fd`) is given by

$$\Pr(Y = j \mid x_1) \; - \; \Pr(Y = j \mid x) \quad \text{for} \quad j = 1, \ldots, J.$$

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

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

where t_i is a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups, and n_j is the number of treated observations in category j.

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

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

where t_i is a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups, and n_j is the number of treated observations in category j.

3.0.17 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 = "ologit", 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.

- **zeta**: a vector containing the estimated class boundaries τ_j.
- **deviance**: the residual deviance.
- **fitted.values**: the $n \times J$ matrix of in-sample fitted values.
- **df.residual**: the residual degrees of freedom.
- **edf**: the effective degrees of freedom.
- **Hessian**: the Hessian matrix.
- **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, and t-statistics.

- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as arrays. Available quantities are:

  - **qi$ev**: the simulated expected probabilities for the specified values of x, indexed by simulation $\times$ quantity $\times$ x-observation (for more than one x-observation).

  - **qi$pr**: the simulated predicted values drawn from the distribution defined by the expected probabilities, indexed by simulation $\times$ x-observation.

  - **qi$fd**: the simulated first difference in the predicted probabilities for the values specified in x and x1, indexed by simulation $\times$ quantity $\times$ x-observation (for more than one x-observation).

  - **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 Ordinal 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 ordinal logit model is part of the MASS package by William N. Venable and Brian D. Ripley [6]. Advanced users may wish to refer to `help(polr)` as well as [5]. Sample data are from [4].

4 `oprobit`: Ordinal Probit Regression for Ordered Categorical Dependent Variables

Use the ordinal probit regression model if your dependent variables are ordered and categorical. They may take on either integer values or character strings.

4.0.18 Syntax

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

If `Y` takes discrete integer values, the `as.factor()` command will order it automatically. If `Y` takes on values composed of character strings, such as "strongly agree", "agree", and "disagree", `as.factor()` will order the values in the order in which they appear in `Y`. You will need to replace your dependent variable with a factored variable prior to estimating the model through `zelig()`.

4.0.19 Example

1. Creating An Ordered Dependent Variable

   Load the sample data:

   ```
   > data(sanction)
   ```

   Create an ordered dependent variable:

   ```
   > sanction$ncost <- factor(sanction$ncost, ordered = TRUE,
   +                          levels = c("net gain", "little effect",
   +                          "modest loss", "major loss"))
   ```

   Estimate the model:
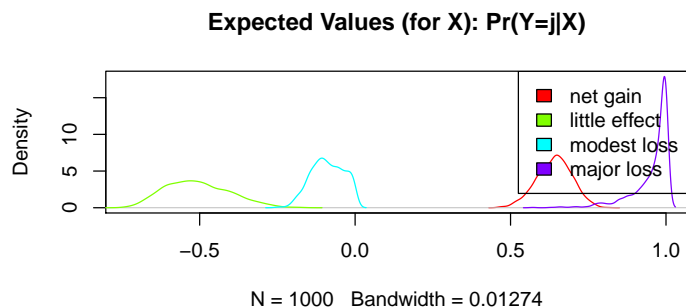
   ```
   > z.out <- zelig(ncost ~ mil + coop, model = "oprobit", data = sanction)
   > summary(z.out)
   ```

   Set the explanatory variables to their observed values:

   ```
   > x.out <- setx(z.out, fn = NULL)
   ```

   Simulate fitted values given `x.out` and view the results:

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

Expected Values (for X): Pr(Y=j|X)

N = 1000   Bandwidth = 0.01274

2. First Differences

Using the sample data `sanction`, let us estimate the empirical model and return the coefficients:

```
>   z.out <- zelig(as.factor(cost) ~ mil + coop, model = "oprobit",
+                    data = sanction)

> summary(z.out)
```

Set the explanatory variables to their means, with `mil` set to 0 (no military action in addition to sanctions) in the baseline case and set to 1 (military action in addition to sanctions) in the alternative case:

```
>   x.low <- setx(z.out, mil = 0)
>   x.high <- setx(z.out, mil = 1)
```
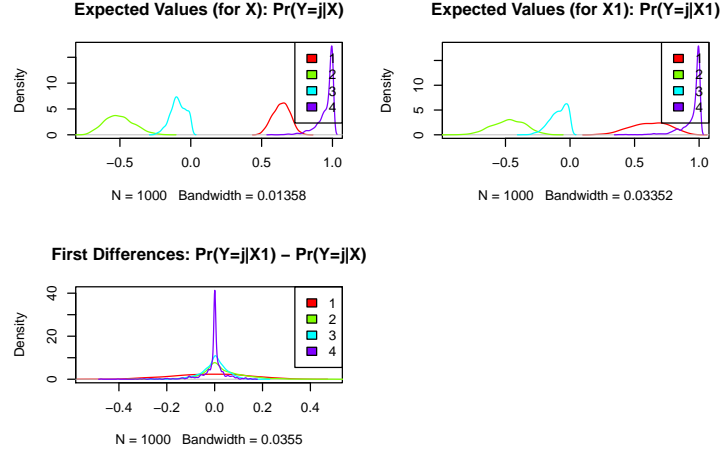
Generate simulated fitted values and first differences, and view the results:

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

19

```
> summary(s.out)

>  plot(s.out)
```

Expected Values (for X): Pr(Y=j|X)

Expected Values (for X1): Pr(Y=j|X1)

First Differences: Pr(Y=j|X1) – Pr(Y=j|X)

4.0.20   Model

Let Y_i be the ordered categorical dependent variable for observation i that takes one of the integer values from 1 to J where J is the total number of categories.

- The *stochastic component* is described by an unobserved continuous variable, Y_i^*, which follows the normal distribution with mean μ_i and unit variance

$$Y_i^* \ \sim \ N(\mu_i, 1).$$

The observation mechanism is

$$Y_i \ = \ j \quad \text{if} \quad \tau_{j-1} \leq Y_i^* \leq \tau_j \quad \text{for} \quad j = 1, \ldots, J.$$

where τ_k for $k = 0, \ldots, J$ is the threshold parameter with the following constraints; $\tau_l < \tau_m$ for all $l < m$ and $\tau_0 = -\infty$ and $\tau_J = \infty$.

Given this observation mechanism, the probability for each category, is given by

$$\Pr(Y_i = j) \ = \ \Phi(\tau_j \mid \mu_i) - \Phi(\tau_{j-1} \mid \mu_i) \quad \text{for} \quad j = 1, \ldots, J$$

20

where $\Phi(\mu_i)$ is the cumulative distribution function for the Normal distribution with mean μ_i and unit variance.

- The *systematic component* is given by

$$\mu_i = x_i\beta$$

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

4.0.21   Quantities of Interest

- The expected values (`qi$ev`) for the ordinal probit model are simulations of the predicted probabilities for each category:

$$E(Y_i = j) = \Pr(Y_i = j) = \Phi(\tau_j \mid \mu_i) - \Phi(\tau_{j-1} \mid \mu_i) \quad \text{for} \quad j = 1, \ldots, J,$$

given draws of β from its posterior.

- The predicted value (`qi$pr`) is the observed value of Y_i given the underlying standard normal distribution described by μ_i.

- The difference in each of the predicted probabilities (`qi$fd`) is given by

$$\Pr(Y = j \mid x_1) - \Pr(Y = j \mid x) \quad \text{for} \quad j = 1, \ldots, J.$$

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

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

where t_i is a binary explanatory variable defining the treatment $(t_i = 1)$ and control $(t_i = 0)$ groups, and n_j is the number of treated observations in category j.

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

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

where t_i is a binary explanatory variable defining the treatment $(t_i = 1)$ and control $(t_i = 0)$ groups, and n_j is the number of treated observations in category j.

4.0.22 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 = "oprobit", 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`: the named vector of coefficients.
  - `fitted.values`: an $n \times J$ matrix of the in-sample fitted values.
  - `predictors`: an $n \times (J-1)$ matrix of the linear predictors $x_i\beta_j$.
  - `residuals`: an $n \times (J-1)$ matrix of the residuals.
  - `zeta`: a vector containing the estimated class boundaries.
  - `df.residual`: the residual degrees of freedom.
  - `df.total`: the total degrees of freedom.
  - `rss`: the residual sum of squares.
  - `y`: an $n \times J$ matrix of the dependent variables.
  - `zelig.data`: the input data frame if `save.data = TRUE`.

- From `summary(z.out)`, you may extract:
  - `coef3`: a table of the coefficients with their associated standard errors and t-statistics.
  - `cov.unscaled`: the variance-covariance matrix.
  - `pearson.resid`: an $n \times (m-1)$ matrix of the Pearson residuals.

- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as arrays. Available quantities are:
  - `qi$ev`: the simulated expected probabilities for the specified values of `x`, indexed by simulation $\times$ quantity $\times$ x-observation (for more than one x-observation).
  - `qi$pr`: the simulated predicted values drawn from the distribution defined by the expected probabilities, indexed by simulation $\times$ x-observation.
  - `qi$fd`: the simulated first difference in the predicted probabilities for the values specified in `x` and `x1`, indexed by simulation $\times$ quantity $\times$ x-observation (for more than one x-observation).
  - `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 Ordinal 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 ordinal probit function is part of the VGAM package by Thomas Yee [7]. In addition, advanced users may wish to refer to `help(vglm)` in the VGAM library. Additional documentation is available at `http://www.stat.auckland.ac.nz/`~\protect\kern+.1667em\relaxyeehttp://www.stat.auckland.ac.nz/ yee.Sample data are from [4]

References

[1] Kosuke Imai, Olivia Lau, and Gary King. *ologit: Ordinal Logistic Regression for Ordered Categorical Dependent Variables*, 2011.

[2] Olivia Lau, Kosuke Imai, and Gary King. *Bivariate Logistic Regression for Two Dichotomous Dependent Variables*, 2011.

[3] Olivia Lau, Kosuke Imai, and Gary King. *Bivariate Probit Regression for Two Dichotomous Dependent Variables*, 2011.

[4] 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.

[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] T. W. Yee and T. J. Hastie. Reduced-rank vector generalized linear models. *Statistical Modelling*, 3:15–41, 2003.