Package 'boolean'

February 14, 2012

Version 2.0-2

Date 2009-11-17

Title Boolean Binary Response Models

Author Bear F. Braumoeller <braumoeller.1@polisci.osu.edu>, Ben Goodrich <goodrich@fas.harvard.edu>, Jacob Kline <jkline@fas.harvard.edu>

Maintainer Bear F. Braumoeller <braumoeller.1@polisci.osu.edu>

Depends R (>= 2.0.1), Zelig

Suggests rgenoud (>= 5.6-6), MCMCpack (>= 0.7-1), MASS

Imports stats, Zelig

Description This package implements a partial-observability procedure for testing Boolean hypotheses that generalizes the binary response GLM.

License GPL (>= 2)

Repository CRAN

Date/Publication 2010-01-24 09:11:33

R topics documented:

`boolean` *Partial-Observability Binary Response Models for Testing Boolean Hypotheses*

Description

Boolean binary response models are a family of partial-observability *n*-variate models designed to permit researchers to model causal complexity, or multiple causal "paths" to a given outcome.

Usage

```
boolean(formula, data, link = "logit", start.values = NULL,
        method = "nlm", bootstrap = 0, control = list(fnscale = -1),
        weights = NULL, robust = FALSE, ...)
```

Arguments

`formula`	The object producted by `boolprep`; see `boolprep` for details.
`data`	An optional dataframe that contains the variables specified in the call to `boolprep`.
`link`	A character string or character vector of length *n* that identifies the CDF(s) to be used for each causal path. If only one link function is given, it is used for all *n* causal paths.The string may be "cauchit", "cloglog", "log", "logit" (the default), "probit", "scobit", "scobitL", "scobitR", or any combination thereof in a character vector of length *n*. Aside from "scobit", "scobitL", and "scobitR" these are the same as used in the `glm` function. For details, see `binomial`. For documentation of the scobit link function, see the Nagler reference below; "scobitR" is an alias for "scobit" and scobitL is a transformation of the scobit CDF that is left-skewed.
`start.values`	If not NULL (the default), a vector of starting values. Such a vector must have length equal to the number of estimated parameters and should be ordered so that the starting values for the coefficients come before the starting values for the ancillary parameter for any "scobit", "scobitL" or "scobitR" link functions. Otherwise, the parameters should be ordered left-to-right based on their order in the call to `boolprep`. The `start.values` argument is useful for checking whether the results are robust to overdispersion of starting values. If NULL, the starting values will be created from a series of calls to `glm` using the appropriate link function, assuming the ancillary parameter is one in the case of the various scobits.

`method` A character string or character vector of length two that specifies the algorithm used to estimate the boolean model. If `bootstrap > 0` and `method` is of length two, then the first element of `method` is used to obtain the maximum-likelihood estimates (presumably with a computationally intensive algorithm) and the second element is used for the bootstraps (possibly using a faster algorithm). Possible choices are: "nlm" (the default) "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN" (the preceding five represent the possibilities for `optim`), "constrOptim", "genoud", or "MH" (short for Metropolis-Hastings without simulating annealing). For "nlm", see `nlm`; for the optim-related methods, see `optim`; for "constrOptim", see `constrOptim`; for genoud, see `genoud`; for "MH" (Metropolis-Hastings), see `MCMCmetrop1R`. Additional arguments for each of these methods can (and often should) be passed through the `...` as indicated in the respective documentations. If `method = "constrOptim"`, the constrained optimization uses the BFGS algorithm. Note that for previous versions of this package, the `method` argument did something very different.

`bootstrap` The number of bootstraps to execute, which must be a non-negative integer. If zero (the default), standard errors are approximated using the Hessian. Note that bootstrapping is appropriate only if the `method` is something other than "MH".

`control` A list, identical in usage with that for `optim`.

`weights` A vector of weights with length equal to the number of valid observations, which is used to produce a weighted sum of the observation-specific log-likelihoods. If NULL (the default), a weight of one is used for each observation. Currently, only NULL is supported.

`robust` If TRUE a sandwich-style variance-covariance matrix would be estimated. However, this option is not current supported.

`...` Additional arguments to be passed to the procedure identified by the `method` argument. Such arguments are often necessary to obtain reasonable results but are too numerous to be listed here. However, it is not possible to pass a function that calculates the gradient.

Details

Boolean permits estimation of Boolean binary response models (see Braumoeller 2003 for derivation), which are a family of partial-observability *n*-variate models designed to permit researchers to model causal complexity, or multiple causal "paths" to a given outcome. The various "paths" are modeled as latent dependent variables that are multiplied together in a manner determined by the logic of their (Boolean) interaction. If, for example, we wanted to model a situation in which diet OR smoking causes heart failure, we would use one set of independent variables (caloric intake, fat intake, etc.) to predict the latent probability of diet-related coronary failure (y1*), use another set of variables (cigarettes smoked per day, exposure to second-hand smoke, etc.) to predict the latent probability of smoking-related coronary failure (y2*), and model the observed outcome (y, or coronary failure) as a function of the Boolean interaction of the two: Pr(y=1) = 1-([1-y1*] x [1-y2*]). Independent variables that have an impact on both latent dependent variables can be included in both paths. Any combination of ANDs and ORs can be posited, and the interaction of any number of latent dependent variables can be modeled, although the procedure becomes exponentially more data-intensive as the number of latent dependent variables increases.

Value

Returns an object of class booltest, with the following list elements:

value	The maximized value of the log-likelihood or NA, if method="MH".
counts	The number of calls to the function and the gradient, or the number of MCMC iterations if method="MH".
convergence	The convergence code for maximum-likelihood methods or NA if method = "MH", which does NOT indicate the presence, absence, or irrelevance of convergence of the Markov chain to the stationary distribution.
message	The convergence message for maximum-likelihood methods, (NA if method = "nlm") or NA if method = "MH", which does NOT indicate the presence, absence, or irrelevance of convergence of the Markov chain to the stationary distribution.
hessian	The Hessian matrix, unless method = "MH", in which case, NULL. If variance is a "caught" error, examining the Hessian may indicate where the problem is.
coefficients	Either a vector of maximum-likelihood estimates, or if method = "MH" or bootstap > 0, a matrix of MCMC or bootstrapped samples with rows equal to the number of samples and columns equal to the number of estimated parameters.
variance	A "try" of the variance-covariance matrix, unless method = "MH", in which case NA.
boolean.call	The call to the boolean function.

Note

Examining profile likelihoods with boolprof is highly recommended. These are partial observability models are generically starved for information; as a result, maximum likelihood estimation can encounter problems with plateaus in likelihood functions even with very large datasets. In principle, if method = "BHHH", the Hessian can be approximated in circumstances where other maximization methods might fail to estimate the Hessian due to plateaus or discontinuities at the maximum. However, such estimates should be analyzed with caution.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233. Nagler, Jonathon. (1994) "Scobit: An Alternative Estimator to Logit and Probit." *American Journal of Political Science* 38(1): 230-255.

See Also

boolprep to prepare the structure of the model, and boolprof to produce profile likelihoods after estimation.

Examples

```
set.seed(50)
x1 <- rnorm(1000)
x2 <- rnorm(1000)
x3 <- rnorm(1000)
x4 <- rnorm(1000)
x5 <- rnorm(1000)
x6 <- rnorm(1000)
y<-1-(1-pnorm(-2+0.33*x1+0.66*x2+1*x3)*1-(pnorm(1+1.5*x4-0.25*x5)*pnorm(1+0.2*x6)))
y <- y>runif(1000)
bp <- boolprep("(a|(b&c))", "y", a = ~ x1 + x2 + x3, b = ~ x4 + x5, c = ~ x6)
answer <- boolean(bp, link = c("probit", "logit", "cloglog"), start.values = ## For speed
                c(-1.750,  0.354,  0.698,  1.231,  1.473,  2.628, -0.452,  0.764,  0.173))

## Plot profiles
boolprof(answer)


## Examine coefficients
coef(answer)

## Examine coefficients, standard errors, etc.
summary(answer)
```

`boolfirst` *Deprecated Function*

Description

This function was implemented in previous versions, but is now deprecated. For an improved implementation, see the `sim` function in the Zelig library.

Usage

```
boolfirst(...)
```

Arguments

`...`	It does not matter.

Value

A message indicating deprecation is printed; nothing is returned.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

See Also

boolean, boolprep, and boolprof.

`boolplot` *Plot Probabilities from Boolean Models*

Description

This function plots the probability of the outcome as a function of an explanatory variable or the change in an explanatory variable.

Usage

```
boolplot(z.out, variable, delta = 0, suppression.factor = FALSE, CI = 95,
      truehist = TRUE, legend = TRUE, plot.both = FALSE, polygon = FALSE, yscale = NULL, ...)
```

Arguments

`z.out`	An object produced by Zelig.
`variable`	A character vector indicating which explanatory variable should constitute the x-axis of the plot.
`delta`	The amount by which to perturb the sample values of `variable` when plotting the change in probability as a function of the change in `variabel`. If `plot.both = TRUE`, then `delta` must be non-zero.
`suppression.factor`	If true, then the complement of the probability of the outcome constitutes the y-axis.
`CI`	Confidence interval for (the change in) the probability of the outcome. Set this equal to any negative number to suppress confidence intervals.
`truehist`	If TRUE, a histogram for `variable` is included in the plot.
`legend`	If TRUE, a legend is included in the plot.
`plot.both`	If TRUE, the probability of the outcome is plotted as a function of the level of `variable` and the change in the probability of the outcome is plotted as a function of `variable` plus `delta`.
`polygon`	If TRUE, then the confidence intervals as filled as ploygons.
`yscale`	If not NULL, then it should be a numeric vector of length two indicating the lower and upper limits of the y-axis in the plot.
`...`	Further arguments passed to `plot` or setx.

Value

Nothing is returned.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

---

boolprep *Generate Functional Form for Boolean binary choice model*

Description

Takes as arguments the logical form of a Boolean model, the dependent variable, and sets of independent variables, and produces a functional form necessary for inclusion in a `boolean` command.

Usage

```
boolprep(FORM, DEPVAR, ...)
```

Arguments

`FORM`	Form of Boolean model, e.g., `"a&b"`, `"a&(b|c)"`, etc. Unlike previous versions of this package, you are not restricted to single letters. For example, `"(lucky|good)"` is now allowed as well. However, `FORM` must be a character string.
`DEPVAR`	Dependent variable, which must be a logical vector or a numeric vector with zeroes and ones only, or the name of such a dependent variable can be given as a character string.
`...`	If there are n logical operators in FORM, users are required to pass n+1 arguments through the dots. These n+1 arguments should be one-sided formulas and follow the same syntax as in the glm() function (see `formula`), such as `~ x1 + x2`. Note that if `FORM = "(lucky|good)"`, then the required arguments would take a form such as `lucky = ~x1, good = ~x2`.

Details

Because the structure of a Boolean binary choice model can be arbitrarily complex, the layering of parentheses soon becomes tedious and unnecessarily difficult. To make life easier, the `boolprep` command automates the task of creating the appropriate argument and checks for syntactical errors.

Value

An object of class boolprep with the structure of the model and the appropriate formulas as list elements, none of which are expected to be accessed by the user. However, boolean requires that a boolprep object be passed to it.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

See Also

boolean

Examples

```
y  <- rbinom(1000, 1, 0.5)

bp <- boolprep("a & b", "y", a = ~x1+x2, b = ~x3+x4+x5)

bp <- boolprep("rich | (smart & diligent)", y, rich = ~ x1+x2+x3,
               smart = ~ x4+x5, diligent = ~ x6)

bp <- boolprep("(a|b|c)&(d|e)&f", y, a=~x1+x2, b = ~x3+x4+x5,
               c = ~x6, d = ~x7+x8, e = ~x9+x10+x11, f = ~x12+x13)
```

---

boolprof *Graph Profile Likelihoods for Boolean Binary Response Models*

Description

This command produces a graph of the profile likelihood for a given coefficient following the estimation of a Boolean binary response model.

Usage

```
boolprof(object, gvar = NULL, range = NULL, M = 100)
```

Arguments

`object`	The object created by the boolean command.
`gvar`	A character string indicating the name of variable associated with coefficient of interest. By default, `boolprof` plots the profile likelihood for each estimated parameter. Use of the default is encouraged. Otherwise, be sure to specify the name of the variable including the "id" that is tacked onto the end of the variable name so as to identify which instance of the variable is of interest, even if the variable appears only once.
`M`	The number of points to plot. The default (100) is usually sufficient.
`range`	Range of points over which graph is plotted (e.g., `range=seq(0,1,0.01)`). If NULL (the default), `M` random draws from a multivariate normal distribution are plotted consistent with the mode and Hessian for models estimated by maximum likelihood. For models estimated via bootstrapping or MCMC, `M` values are sampled randomly from the output.

Value

Nothing is returned. A graph is plotted.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

See Also

`boolean` and `boolprep`

Examples

```
set.seed(50)
x1<-rnorm(1000)
x2<-rnorm(1000)
x3<-rnorm(1000)
x4<-rnorm(1000)
x5<-rnorm(1000)
x6<-rnorm(1000)
y<-1-(1-pnorm(-2+0.33*x1+0.66*x2+1*x3)*1-(pnorm(1+1.5*x4-0.25*x5)*pnorm(1+0.2*x6)))
y <- y>runif(1000)
bp <- boolprep("(a|(b&c))", y, a = ~ x1 + x2 + x3, b = ~ x4 + x5, c = ~ x6)
answer <- boolean(bp, link = c("probit", "logit", "cloglog"))

## Plot profiles
```

```
boolprof(answer)
```

`coef.booltest` *Coefficients for Boolean Binary Response Models*

Description

This function implements a 'coef' S3 method for objects of class 'booltest'. If the model was estimated by maximum likelihood, the maximum likelihood estimates are returned. If the model was estimated by Metropolis-Hastings or bootstrapped, the mean over the iterations for each estimated parameter is returned with a warning.

Usage

```
## S3 method for class 'booltest'
## S3 method for class 'booltest'
coef(object, ...)
```

Arguments

`object`	An object of class 'booltest'
`...`	Anything passed will be ignored.

Value

A numeric vector with one element per estimated parameter.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

effectplot *Visually Assess Nonlinearity in Effects of Explanatory Variables*

Description

This function plots the change in probability of the outcome in one explanatory variable as a function of a change in another explanatory variable.

Usage

```
effectplot(z.out, variables, delta = 1, CI = 95, truehist = TRUE,
                        legend = TRUE, polygon = FALSE, ...)
```

Arguments

z.out	An object produced by Zelig.
variables	A character vector of length two. The first indicates which explanatory variable should constitute the x-axis of the plot. The second indicates which explanatory variable should be perturbed by delta
delta	The amount by which to perturb the sample values of the second element of variable.
CI	Confidence interval for (the change in) the probability of the outcome. Set this equal to any negative number to suppress confidence intervals.
truehist	If TRUE, a histogram for variable is included in the plot.
legend	If TRUE, a legend is included in the plot.
polygon	If TRUE, then the confidence intervals as filled as ploygons.
...	Further arguments passed to plot or setx.

Value

Nothing is returned.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

`fitted.booltest` *Fitted Values (Probabilities) for Boolean Binary Response Models*

Description

This function implements a 'fitted' (and an identical 'fitted.values') S3 method for objects of class 'booltest'. Since these pertain to a binary response model, the 'fitted values' are probabilities. If `method="MH"` or `bootstrap > 0` in the call to `boolean`, then the mean for each estimated parameter is used to create the fitted values.

Usage

```
## S3 method for class 'booltest'
## S3 method for class 'booltest'
fitted(object, ...)
## S3 method for class 'booltest'
fitted.values(object, ...)
```

Arguments

`object`	An object of class 'booltest'
`...`	Anything passed will be ignored

Value

A numeric vector of probabilities.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

latent *Extract Probabilities from Each Path from Boolean Models*

Description

Boolean models involve more than one 'path', e.g. if y~a&(b|c), then a, b, and c are paths that are parameterized with explanatory variables. This function extracts estimates of a, b, and c, which are probabilities but not the probability that the dependent variable occurs.

Usage

```
latent(object, probability = FALSE, invMills = FALSE)
```

Arguments

object	An object of class 'booltest'
probability	If TRUE, the linear predictor is transformed by the link function specified in the call to boolean to produce a vector of probabilities.
invMills	If TRUE, the inverse Mills ratio is calculated for paths where a probit link is specified in the call to boolean. Note that if both probabilities and invMills are FALSE (the default), then the linear predictor for each path is returned.

Value

A numeric matrix with columns equal to the number of paths and rows equal to the number of observations. Whether the matrix contains the linear predictions, probabilities, or inverse Mills ratios depends on the call to latent.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233. Nagler, Jonathon. (1994) "Scobit: An Alternative Estimator to Logit and Probit." *American Journal of Political Science* 38(1): 230-255.

`print.summary.booltest` *'print' Method for Boolean Binary Response Model Summaries*

Description

'print' method for objects of class 'summary.booltest'

Usage

```
## S3 method for class 'summary.booltest'
## S3 method for class 'summary.booltest'
print(x, ...)
```

Arguments

`x`	An object of class 'summary.booltest'
`...`	Further arguments.

Value

Does not return anything.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

`setx.booltest` *'booltest' Method for Post-Estimation Via 'zelig'*

Description

This function is an interface to the `setx` function in the Zelig library. It should not be called directly by the user.

Usage

```
## S3 method for class 'booltest'
## S3 method for class 'booltest'
setx(object, fn = list(numeric = mean, ordered = median, other = mode),
              data = NULL, cond = FALSE, counter = NULL, ...)  # good
```

Arguments

object	An object of class 'boolprep' estimated by Zelig.
fn	A list of functions; see setx for details.
data	An additional dataframe containing values for the explanatory variables or NULL to use the dataframe in the call to Zelig.
cond	A flag to indicate conditional prediction; see setx for details.
counter	Should always be NULL.
...	Additional arguments to set explanatory variables to particular values.

Value

A dataframe of explanatory variables set to the specified values; see setx for details.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>, and
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>

---

sim.booltest *'booltest' Method for Parameter Simulation Via 'zelig'*

Description

These functions provide an interface to the sim function in the Zelig library. They should not be called directly by the user.

Usage

```
## S3 method for class 'booltest'
## S3 method for class 'booltest'
sim(object, x, x1=NULL, num=c(1000, 100),
            prev = NULL, bootstrap = FALSE, bootfn=NULL, cond.data = NULL, ...)
```

Arguments

object	An object of class 'booltest'estimated by Zelig.
x	An object produced by setx.
x1	An optional, additional object produced by setx.
num	The number of simulations of the parameters.
prev	An optional matrix of parameter simulations supplied by the user.
bootstrap	Should be FALSE because bootstrapping is supported directly in the call to Zelig for boolean models. See also boolean.
bootfn	Should be NULL because bootstrapping is supported directly in the call to Zelig for boolean models. See also boolean.
cond.data	Should be NULL.
...	Should not be needed for a boolean model.

Value

The param and qi methods are used internally by sim, which in turn produces quantities of interest. See sim for details; the output is the same as for a probit or logit model estimated via zelig.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>, and
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>

---

summary.booltest *'summary' Method for Boolean Binary Response Models*

Description

'summary' method for objects of class 'booltest'

Usage

```
## S3 method for class 'booltest'
## S3 method for class 'booltest'
summary(object, ...)
```

Arguments

object	An object of class 'booltest'
...	Further arguments

Value

An object of class 'summary.booltest', a list with components

formula	The formula from boolprep
summary.mat	A matrix of coefficients, standard errors, etc.
log.likelihood	The log-likelihood at the optimum, NULL if Metropolis-Hastings was used to estimate
iterations	The number of iterations, NULL if Metropolis- Hastings was used to estimate.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

---

weights.booltest *Extract weights for Boolean Binary Response Models*

Description

This function implements a 'weights' S3 method for objects of class '"booltest"'. However, weights are not yet supported in boolean models, so a vector of ones is returned.

Usage

```
## S3 method for class 'booltest'
## S3 method for class 'booltest'
weights(object, ...)
```

Arguments

object	An object of class 'booltest'
...	Anything passed will be ignored

Value

A numeric vector of ones.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

`zelig.boolprep` *'boolprep' Method for Estimation Via 'zelig'*

Description

This function is an interface to the `Zelig` function in the Zelig library. It should not be called directly by the user.

Usage

```
## S3 method for class 'booltest'
## S3 method for class 'boolprep'
zelig(formula, model, data, by, save.data = FALSE, cite, ...)
```

Arguments

`formula`	An object of class 'boolprep'
`data`	A dataframe containing all the relevant variables.
`model`	A character vector, "boolean" in this case.
`by`	A character vector with the name of a stratifying variable so that the model is estimated on one subset of observations at a time.
`save.data`	Should the data be saved in the resulting object, defaults to FALSE
`cite`	I do not know what this does
`...`	Further arguments passed

Value

An object of class 'booltest', see boolean for details.

Author(s)

Bear F. Braumoeller, Harvard University, <braumoeller.1@polisci.osu.edu>,
Ben Goodrich, Harvard University, <goodrich@fas.harvard.edu>, and
Jacob Kline, Harvard University, <jkline@fas.harvard.edu>

References

Braumoeller, Bear F. (2003) "Causal Complexity and the Study of Politics." *Political Analysis* 11(3): 209-233.

Index