Package 'boolean3'

February 19, 2015

Version 3.1.6

Date 2014-11-15

Title Boolean Binary Response Models

Author Jason W. Morgan <morgan.746@osu.edu>

Maintainer Jason W. Morgan <morgan.746@osu.edu>

Depends R (>= 3.0), utils

Imports stats, optimx (>= 2013.8.6), numDeriv, lattice, rgenoud, parallel, mvtnorm, rlecuyer

Description This package implements a partial-observability procedure for testing Boolean hypotheses that generalizes the binary response GLM as outlined in Braumoeller (2003).

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2014-11-18 11:28:19

R topics documented:

boolean3-package 2
boolboot 6
boolean 7
boolean-class 9
boolfit-class 9
boolprep 10
boolprob 12
boolprob-class 13
boolprof 14
boolprof-class 15
boolsum-class 16
coef.boolean 16
logLik.boolean 17

model.matrix.boolean 17
nobs.boolean 18
plot.boolprob 19
plot.boolprof 19
predict.boolean 20
print.boolboot 21
print.boolean 21
print.boolprob 22
print.boolprof 23
print.boolsum 23
summary.boolboot 24
summary.boolean 24
vcov.boolean 25

Index **26**

`boolean3-package` *Modeling Causal Complexity*

Description

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

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: $\Pr(y = 1) = 1 - ([1 - y1^*] \times [1 - y2^*])$. Independent variables that have an impact on both latent dependent variables can be included in both paths. Any combination of any number of ANDs and ORs can be modeled, though the procedure becomes exponentially more data-intensive as the number of interactions increases.

Package:	boolean3
Type:	Package
Version:	3.1.6
Date:	2014-11-15
License:	GPL-3
LazyLoad:	yes

Note

boolean3 was developed by Jason W. Morgan under the direction of Bear Braumoeller with support from The Ohio State University's College of Social and Behavioral Sciences. The package represents a significant re-write of the original boolean implementation developed by Bear Braumoeller, Ben Goodrich, and Jacob Kline. Please see the release notes and accompanying documentation for details regarding the many changes made in this version.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

References

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

See Also

See boolprep for model setup and boolean for estimation.

Examples

```
## Generate some fake data
require(mvtnorm)
set.seed(12345)
N  <- 2000
Df <- cbind(1, rmvnorm(N, mean=rep(0, 5)))

## Set coefficients
beta.a <- c(-2.00, 0.33, 0.66, 1.00)
beta.b <- c(0.00, 1.50, -0.25)

## Generate path probabilities following a normal model.
y.a <- as.vector(pnorm(tcrossprod(beta.a, Df[, 1:4])))
y.b <- as.vector(pnorm(tcrossprod(beta.b, Df[, c(1, 5, 6)])))

## AND and OR-model
or <- function(x, y) { x + y - x * y }
and <- function(x, y) { x * y }
y.star.OR  <- or(y.a, y.b)
y.star.AND <- and(y.a, y.b)

## Observed responses
y.OR <- rbinom(N, 1, y.star.OR)
y.AND <- rbinom(N, 1, y.star.AND)

## Set up data.frame for estimation
Df <- cbind(1, Df)
Df <- as.data.frame(Df)
Df[,1] <- y.OR
Df[,2] <- y.AND
```

```
names(Df) <- c("y.OR", "y.AND", "x1", "x2", "x3", "x4", "x5")

## Before estimating, boolean models need to be specified using the
## boolprep function.

## OR model, specified to use a probit link for each causal path. This
## matches the data generating process above.
mod.OR <- boolprep(y.OR ~ (a | b), a ~ x1 + x2 + x3, b ~ x4 + x5,
                   data = Df, family=list(binomial("probit")))

## IF you really want to, it's also possible to specify a different
## link function for each causal path. These should be in the same
## order as specified in the model formula.
mod.OR2 <- boolprep(y.OR ~ (a | b), a ~ x1 + x2 + x3, b ~ x4 + x5,
                    data = Df, family=list(binomial("probit"),
                                  binomial("logit")))

## Fit the prepared model using the nlminb optimizer (the default).
(fit.OR <- boolean(mod.OR, method="nlminb", control=list(trace=1)))

## Multiple optimizers can be specified in a single call to boolean.
## Here we fit with the nlm and nlminb optimizers.
(fit1.OR <- boolean(mod.OR, method=c("nlm", "nlminb")))

## Re-fit, with BFGS and a higher maximum number of iterations. All
## of the options that go along with nlm(), optim(), and genoud() should
## be transparently passable through boolean.
(fit2.OR <- boolean(mod.OR, method="BFGS", control = list(maxit = 500)))

## Induce a convergence warning message.
(fit3.OR <- boolean(mod.OR, method="BFGS", control = list(maxit = 5)))

## Not run:
## Now estimate model with genoud, a genetic optimizer. This has the
## capability of using multiple processors via parallel.
(fit6.OR <- boolean(mod.OR, method="genoud",
                    cluster=c("localhost", "localhost"),
                    print.level=2))

## The default SANN optimizer is also available.
(fit7.OR <- boolean(mod.OR, method="SANN"))

## End(Not run)

## The fit is stored as "model.fit", within the boolean object.
str(fit.OR$model.fit)

## Create a summary object, saving and printing it. Then take a look at
## the objects stored in the summary object.
(smry <- summary(fit.OR))
str(smry)

## Extract log-likelihood and coefficient vector.
```

```
logLik(fit.OR)
coef(fit.OR)

## Not run:
## Display the contours of the likelihood given a change the value of
## the coefficients. Despite the function name, these are not true
## profile likelihoods as they hold all other coefficients fixed at
## their MLE.
(prof <- boolprof(fit.OR))

## Extract the plots for x1_a and x4_b.
plot(prof, y = c("x1_a", "x4_b"))
plot(prof, y = c(1, 3), scales = list(y = list(relation = "free")))

## You can also use variable or index matching with boolprof to select
## particular covariates of interest.
boolprof(fit.OR, vars = c(1, 3))
boolprof(fit.OR, vars = c("x1_a", "x4_b"))

## Plots of predicted probabilities are available through boolprob.
## With boolprob, either vars or newdata *must* be specified.
boolprob(fit.OR, vars = c("x1_a", "x4_b"))
boolprob(fit.OR, vars = c(2, 3, 4, 6))

## Specifying conf.int = TRUE produces simulated confidence intervals.
## The number of samples to pull from the distribution of the estimated
## coefficients is controlled by n; n=100 is default. This can take a
## while.
(prob <- boolprob(fit.OR, vars = c(2, 3, 4, 6), n = 1000,
                  conf.int = TRUE))

## Choose a different method estimate upon which to base the estimates.
(prob <- boolprob(fit1.OR, method="nlm", vars=c(2, 3, 4, 6), n=1000,
                  conf.int=TRUE))

## As with the other components of the model, you can extract the
## predicted probabilities.
str(prob)
prob$est

## Bootstrapping is also possible, and is the recommended method of
## making inferences. The boolbool function uses a simple sampling scheme:
## it resamples repeatedly from the provided data.frame. More the complex
## data structures with, for example, clustering, need to be dealt with
## manually.
(bs <- boolboot(fit, n=10))

## boolboot supports bootstrapping across multiple processors.
(bs <- boolboot(fit, n=100, cluster=c("localhost", "localhost")))


## End(Not run)
```

boolboot *Bootstrap for Boolean Models*

Description

Performs bootstrap estimates of a boolean model.

Usage

```
boolboot(obj, n = 100, method = "nlminb", cluster = NULL, ...)
```

Arguments

obj	boolean model object as produced by boolprep and first estimated with boolean.
n	integer specifying the number of bootstrap estimates. Defaults to 100.
method	string specifying the method of estimation. The specified method should be one of those available from the optimx or optim functions. Defaults to "nlminb".
cluster	string vector specifying hosts to use for parallel processing through parallel (see makeCluster). Defaults to NULL indicating no clustering.
...	additional parameters to pass on to subsequent functions.

Details

boolboot performs bootstrap estimated of a boolean model specified by boolprep and first estimated with boolean.

Value

boolboot returns a boolboot model object. This object is identical to a boolean model object but with an additional model.boot slot containing the results of the bootstrap. A separate object type is used to help prevent the accidential loss of bootstrap estimates.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

References

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

See Also

See boolprep for model setup, boolean for estimation, the parallel package for details on clustering, and optimx and optim for estimation methods.

boolean *Fit a Boolean Model*

Description

Performs a fit of a boolean model as specified by boolprep.

Usage

```
boolean(obj, method = "nlminb", start = NULL, ...)
```

Arguments

obj	boolean model object as produced by boolprep.
method	string (or string vector) specifying the method(s) of estimation. The specified method(s) should be one of those available from the optimx or optim functions. A genetic algorithm is available from genoud (method="genoud"). method defaults to "nlminb".
start	numeric vector specifying starting values for each parameter in the model. Must have a length equal to the number of parameters being estimated. Defaults to NULL, which instructs boolean to estimate "sensible" starting values (currently the coefficient values estimated from a glm model).
...	additional arguments to be passed on to optimizers. Each optimizer provides numerous optional parameters to help improve estimation results. See the provided examples and the documentation for the estimation method of interest.

Details

boolean performs a fit of a boolean model as specified by boolprep.

Value

A boolean model object containing the fit results (detailed results available in the model.fit slot).

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

References

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

See Also

See boolprep for model setup, boolean the snow package for details on clustering (useful when using genoud). See optimx, optim, and genoud for detailed documentation on the estimation methods available.

Examples

```
## Generate some fake data
require(mvtnorm)
set.seed(12345)
N  <- 2000
Df <- cbind(1, rmvnorm(N, mean=rep(0, 5)))

## Set coefficients
beta.a <- c(-2.00, 0.33, 0.66, 1.00)
beta.b <- c(0.00, 1.50, -0.25)

## Generate path probabilities following a normal model.
y.a <- as.vector(pnorm(tcrossprod(beta.a, Df[, 1:4])))
y.b <- as.vector(pnorm(tcrossprod(beta.b, Df[, c(1, 5, 6)])))

## AND and OR-model
or <- function(x, y) { x + y - x * y }
and <- function(x, y) { x * y }
y.star.OR  <- or(y.a, y.b)
y.star.AND <- and(y.a, y.b)

## Observed responses
y.OR <- rbinom(N, 1, y.star.OR)
y.AND <- rbinom(N, 1, y.star.AND)

## Set up data.frame for estimation
Df <- cbind(1, Df)
Df <- as.data.frame(Df)
Df[,1] <- y.OR
Df[,2] <- y.AND
names(Df) <- c("y.OR", "y.AND", "x1", "x2", "x3", "x4", "x5")

## Before estimating, boolean models need to be specified using the
## boolprep function.

## OR model, specified to use a probit link for each causal path. This
## matches the data generating process above.
mod.OR <- boolprep(y.OR ~ (a | b), a ~ x1 + x2 + x3, b ~ x4 + x5,
                   data = Df, family=list(binomial("probit")))

## Fit a model using the nlminb optimizer (the default). Verbose output is
## requested by specifying trace=1 as a control parameter.
(fit <- boolean(mod.OR, method="nlminb", control=list(trace=1)))

## Multiple optimizers can be specified in a single call to boolean. Here
## we fit with the nlm and nlminb optimizers.
(fit1 <- boolean(mod.OR, method=c("nlm", "nlminb")))

## The summary function will report the detailed results for each
## optimization method.
summary(fit1)
```

boolean-class *Boolean Model Class*

Description

An object of class boolean as returned from boolprep.

Value

A boolean class contains the following items:

call call to estimate the model.

links link functions used to estimate the model.

model list, specifying the structure of the boolean model (used internally).

N number of observations.

k number of covariates.

coef.labels preformatted labels for the estimated coefficients.

coef.idx indicies for the coefficients in the model matrix (used internally).

response response vector.

frame model frame used to estimate the model.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

boolfit-class *A boolfit class*

Description

An object of class boolfit.

Value

A boolfit object is a component added to a boolean-class object (as model.fit) once the model is fit with boolean. It consists of a list of results returned by the particular optimization routines used to fit the model.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

boolprep *Specify Boolean Model*

Description

Construct a boolean object that can then be fit with boolean.

Usage

```
boolprep(..., data, subset, weights, na.action, offset,
  family = list(binomial(link = "logit")))
```

Arguments

...	formula specification for boolean model. See the details and examples sections below.
data	data.frame containing the data to be used in the model.
subset	select subset of data. See lm for details.
weights	specify model weights (not implemented).
na.action	set na.action. See lm for details.
offset	specify an offset. Offsets are not implemented and this parameter is simply ignored.
family	a model family to use for estimation. This can be comprised of a list of link functions when the desire is to have model components with different links. binomial is the only family supported at this time. See family for more details.

Details

boolprep sets up a boolean model object that can then be fit with boolean. A properly specified model (contained in ...) will contain at least three components. The first component must specify the boolean logic to be employed. For instance, the y ~ (a | b) formula would indicate a logical or between the a and b submodels, while y ~ (a & b) would indicate a logical and. y is the name of the response variable of interest. Logical operators can be nested; e.g., y ~ (a | (b & c)) is valid. The second and third components are submodels and are specified as usual: a ~ x1 + x2 and b ~ x3 + x4 + x5, where are the x-variables are covariates. a, b, and c are labels indicating the submodel position in the boolean specification.

Value

boolprep returns a boolean-class object containing the model components needed to estimate a boolean model.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

References

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

See Also

See boolean for model estimation.

Examples

```
## Generate some fake data
require(mvtnorm)
set.seed(12345)
N  <- 2000
Df <- cbind(1, rmvnorm(N, mean=rep(0, 5)))

## Set coefficients
beta.a <- c(-2.00, 0.33, 0.66, 1.00)
beta.b <- c(0.00, 1.50, -0.25)

## Generate path probabilities following a normal model.
y.a <- as.vector(pnorm(tcrossprod(beta.a, Df[, 1:4])))
y.b <- as.vector(pnorm(tcrossprod(beta.b, Df[, c(1, 5, 6)])))

## AND and OR-model
or <- function(x, y) { x + y - x * y }
and <- function(x, y) { x * y }
y.star.OR  <- or(y.a, y.b)
y.star.AND <- and(y.a, y.b)

## Observed responses
y.OR <- rbinom(N, 1, y.star.OR)
y.AND <- rbinom(N, 1, y.star.AND)

## Set up data.frame for estimation
Df <- cbind(1, Df)
Df <- as.data.frame(Df)
Df[,1] <- y.OR
Df[,2] <- y.AND
names(Df) <- c("y.OR", "y.AND", "x1", "x2", "x3", "x4", "x5")

## Before estimating, boolean models need to be specified using the
## boolprep function.

## OR model, specified to use a probit link for each causal path. This
## matches the data generating process above.
mod.OR <- boolprep(y.OR ~ (a | b), a ~ x1 + x2 + x3, b ~ x4 + x5,
                   data = Df, family=list(binomial("probit")))
```

boolprob *Calculate predicted probabilities*

Description

This function calculates predicted probabilities for the selected covariate profiles.

Usage

```
boolprob(obj, vars = NULL, newdata = NULL, k = 50, conf.int = FALSE,
  n = 100, as.table = TRUE, scales = list(x = list(relation = "free")),
  between = list(x = 1, y = 1), xlab = "x",
  ylab = "Predicted probability", ...)
```

Arguments

obj	object of boolean-class containing a fit boolean model.
vars	vector selecting a set of covariates from the fitted model. This can be a character vector of covariate names (as output from summary(obj)), or a numeric vector indexing the desired covariates.
newdata	data.frame with the same structure as model.matrix(boolean).
k	integer indicating the number of points at which the predicted probability should be calculated.
conf.int	logical; should confidence intervals be simulated.
n	number of draws to take from the estimated parameter space.
as.table	logical (default TRUE), to be passed to xyplot.
scales	list of settings for the scales argument passed to xyplot.
between	numeric specifying the space between panels.
xlab	string, the x-axis label.
ylab	string, the y-axis label.
...	Additional arguments to pass to xyplot. See that documentation for details.

Value

Returns an object of boolprob-class, the default action being to present the default plot.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

Examples

```
## Not run:

## Note: This example assumes a boolean model has already been fit.

## Plot predicted probabilities for a fitted model. Either vars or
## newdata *must* be specified.
boolprob(fit, vars = c("x1_a", "x4_b"))
boolprob(fit, vars = c(2, 3, 4, 6))

## Specifying conf.int = TRUE produces simulated confidence intervals.
## The number of samples to pull from the distribution of the estimated
## coefficients is controlled by n; n=100 is default. This can take a
## while.
(prob <- boolprob(fit, vars = c(2, 3, 4, 6), n = 1000, conf.int = TRUE))

## End(Not run)
```

---

boolprob-class *A boolprob class*

Description

An object of class boolprob.

Value

The object returned by `boolprob`. It's comprised of the following items:

est a data.frame of five columns: `llik`, the calculated range for the predicted probability `x`, the covariate value; `coef`, the preformatted variable name.

coef.labels preformatted coefficient labels.

default.plot a lattice plot produced by `xyplot`.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

boolprof *Calculate estimated likelihood-profiles.*

Description

This function calculates log-likelihood profiles for the selected variables. Despite the function name, these are not true profile likelihoods as they hold all other coefficients fixed at their MLE.

Usage

```
boolprof(obj, method = names(obj$model.fit)[1], vars = 1:obj$k, k = 50,
  as.table = TRUE, scales = list(x = list(relation = "free")),
  between = list(x = 1, y = 1), main = "Estimated likelihood profiles",
  xlab = "beta", ylab = "Log-likelihood", ...)
```

Arguments

obj	object of boolean-class containing a fit boolean model.
method	estimation method to use
vars	numeric vector selecting a set of covariates from the fitted model
k	integer indicating the number of points at which the log-likelihood should be calculated.
as.table	logical (default TRUE), to be passed to xyplot.
scales	list of settings for the scales argument passed to xyplot.
between	numeric specifying the space between panels.
main	string, plot title
xlab	string, the x-axis label.
ylab	string, the y-axis label.
...	Additional arguments to pass to xyplot. See that cumentation for details.

Value

Returns an object of boolprof-class, the default action being to present the default plot.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

Examples

```
## Not run:

## Note: This example assumes a boolean model has already been fit.

## Display the contours of the likelihood given a change the value of
## the coefficients.
(prof <- boolprof(fit))

## Extract the plots for x1_a and x4_b.
plot(prof, y = c("x1_a", "x4_b"))
plot(prof, y = c(1, 3), scales = list(y = list(relation = "free")))

## You can also use variable or index matching with boolprof to select
## particular covariates of interest.
boolprof(fit, vars = c(1, 3))
boolprof(fit, vars = c("x1_a", "x4_b"))

## End(Not run)
```

---

boolprof-class *A boolprof class*

Description

An object of class boolprof.

Value

The object returned by `boolprof`. It's comprised of the following items:

est a data.frame of three columns: `llik`, the calculated log-likelihood; `x`, the coefficient value; `var`, the preformatted variable name.

coef.labels preformatted coefficient labels.

default.plot a lattice plot produced by `xyplot`.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

boolsum-class *A boolsum class*

Description

An object of class boolsum.

Value

The object returned by `summary.boolean`.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

coef.boolean *Return the model estimates.*

Description

This function returns a vector of coefficient estimates.

Usage

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

Arguments

`object`	boolean model object.
`...`	Additional parameters to be passed.

Value

A vector of the coefficient estimates.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

logLik.boolean *Return the log-likelihood for a boolean model.*

Description

This function returns the log-likelihood of the estimated boolean model.

Usage

```
## S3 method for class 'boolean'
logLik(object, ...)
```

Arguments

object	boolean model object.
...	Additional parameters to be passed.

Value

This function returns an object of class logLik.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

model.matrix.boolean *Model Matrix from boolean object*

Description

Extract model matrix from boolean model.

Usage

```
## S3 method for class 'boolean'
model.matrix(object, ...)
```

Arguments

object	A boolean object.
...	Arguments to pass on.

Details

This function extracts the model matrix from the specified boolean model. Note that this model wouldn't always be appropriate for estimating a model since multiple intercepts are included, making the columns of the matrix perfectly collinear.

Value

An n-by-k model matrix.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

`nobs.boolean` *Extract the Number of Observations from a Fit.*

Description

Extract the Number of Observations from a Fit.

Usage

```
## S3 method for class 'boolean'
nobs(object, ...)
```

Arguments

`object`	boolean model object
`...`	further arguments to be passed to other methods.

Details

Extract the Number of Observations from a Fit.

Value

An integer.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

plot.boolprob *Profile Predicted Probability Plot*

Description

Plot predicted probabilities for selected covariates. Adjust default plot produced by boolprob to fit the user's preferences.

Usage

```
## S3 method for class 'boolprob'
plot(x, y = NULL, ...)
```

Arguments

x	boolprob object.
y	character or numeric vector specifying the covariates for which the predicted probabilities should be plotted.
...	additional arguments to pass to update.trellis. See xyplot for details.

Value

Plots the predicted probabilities for the selected covariates.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

plot.boolprof *Profile Likelihood Plot*

Description

Plot the profile likelihoods for selected covariates. Adjust default plot produced by boolprof to fit the user's preferences.

Usage

```
## S3 method for class 'boolprof'
plot(x, y = NULL, ...)
```

Arguments

x	boolprof object.
y	character or numeric vector specifying the covariates for which the profiled likelihood should be plotted.
...	additional arguments to pass to update.trellis. See xyplot for details.

Value

Plots the profile likelihood for the selected covariates.

Author(s)

Jason W. Morgan <morgan.746@osu.edu>

---

predict.boolean *Predicted Probabilities*

Description

Calculate predicted probabilities

Usage

```
## S3 method for class 'boolean'
predict(object, newdata = NULL, ...)
```

Arguments

object	boolean object.
newdata	optionally, a data.frame containing the covariate profiles to predict. If omitted, the original data used in fitting the model will be used. Note that newdata must have the same structure as that returned by model.matrix.
...	Additional parameters to pass on.

Details

Calculate predicted probabilities for each observation.

Value

A vector of predicted probabilities for the covariate profiles specified in newdata.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

print.boolboot *Print Bootstrap Results for Boolean Object*

Description

Default print for boolboot objects.

Usage

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

Arguments

x	boolboot object.
...	Additional parameters to pass on.

Details

Default print for boolboot objects.

Value

Print results in abbreviated summary.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

print.boolean *Print a brief summary of a boolean model fit.*

Description

This function prints a brief summary of the model fit to the user. The details of a model fit are provided by summary.boolean. If the model was not fit, this simply reports some summary statistics about the unfit model.

Usage

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

Arguments

x	boolean object.
...	Additional parameters to be passed.

Value

print brief summary of the model and the model fit (if available).

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

---

print.boolprob *Default print method for boolprob objects.*

Description

This is the default print object for boolprob objects.

Usage

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

Arguments

x	object of the boolprob-class.
...	Additional arguments that are passed to the lattice plot object.

Value

Plots the estimated predicted probabilities.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

print.boolprof *Default print method for boolprof objects.*

Description

This is the default print object for `boolprof` objects.

Usage

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

Arguments

`x`	object of the `boolprof-class`.
`...`	Additional arguments that are passed to the `lattice` plot object.

Value

Plots the estimated log-likelihood profiles.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

print.boolsum *Print summary of boolean model as described in* `boolsum` *object.*

Description

This function prints a summary of a boolean model as described in a `boolsum`, itself the result of calling `summary` on an object of class `boolean`.

Usage

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

Arguments

`x`	object of class `boolsum`.
`...`	Additional parameters to be passed.

Value

Prints summary information about the boolean model fit.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

`summary.boolboot` *Print Bootstrap Results for Boolean Object*

Description

Summary function for boolboot objects.

Usage

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

Arguments

`object`	object of class boolboot.
`...`	additional parameters to pass on.

Details

Summary function for boolboot objects.

Value

Prints a summary.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

`summary.boolean` *Create a summary object from a boolean estimate.*

Description

This function calculates the standard model fit summary information given the model object.

Usage

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

Arguments

object	object of class boolean.
...	Additional parameters to be passed.

Value

A list of class boolsum-class.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

---

vcov.boolean *Variance-Covariance Matrix for Boolean Model*

Description

Calculate the variance-covariance matrix for a boolean object.

Usage

```
## S3 method for class 'boolean'
vcov(object, ...)
```

Arguments

object	boolean object.
...	Additional parameters to pass on.

Details

Calculate the variance-covariance matrix for a boolean object.

Value

A matrix, the negative of the inverse Hessian.

Author(s)

Jason W. Morgan (<morgan.746@osu.edu>)

Index

∗Topic **package**
boolean3-package, 2

boolboot, 6
boolean, *3*, *6*, *7*, 7, *9–11*
boolean-class, 9
boolean3 *(*boolean3-package*)*, 2
boolean3-package, 2
boolfit-class, 9
boolprep, *3*, *6*, *7*, *9*, 10
boolprob, 12, *13*, *19*
boolprob-class, 13
boolprof, 14, *15*, *19*
boolprof-class, 15
boolsum-class, 16

coef.boolean, 16

family, *10*

genoud, *7*
glm, *7*

lm, *10*
logLik, *17*
logLik.boolean, 17

makeCluster, *6*
model.matrix.boolean, 17

nobs.boolean, 18

optim, *6*, *7*
optimx, *6*, *7*

plot.boolprob, 19
plot.boolprof, 19
predict.boolean, 20
print.boolboot, 21
print.boolean, 21
print.boolprob, 22
print.boolprof, 23
print.boolsum, 23

summary.boolboot, 24
summary.boolean, *16*, 24

update.trellis, *19*, *20*

vcov.boolean, 25

xyplot, *12*, *14*, *19*, *20*