Package ‘CBPS’

December 30, 2016

Version 0.13
Date 2016-12-27
Title Covariate Balancing Propensity Score
Author Christian Fong <christianfong@stanford.edu>,
Marc Ratkovic <ratkovic@princeton.edu>,
Chad Hazlett <chazlett@ucla.edu>,
Xiaolin Yang <xiaoliny@princeton.edu>,
Kosuke Imai <kimai@princeton.edu>
Maintainer Christian Fong <christianfong@stanford.edu>
Depends R (>= 2.14), MASS, MatchIt, nnet, numDeriv
Imports
Description Implements the covariate balancing propensity score (CBPS) proposed
by Imai and Ratkovic (2014) <DOI:10.1111/rssb.12027>. The propensity score is
estimated such that it maximizes the resulting covariate balance as well as the
prediction of treatment assignment. The method, therefore, avoids an iteration
between model fitting and balance checking. The package also implements several
extensions of the CBPS beyond the cross-sectional, binary treatment setting.
The current version implements the CBPS for longitudinal settings so that it can
be used in conjunction with marginal structural models from Imai and Ratkovic
(2015) <DOI:10.1080/01621459.2014.956872>, treatments with three- and four-valued
treatment variables, continuous-valued treatments from Fong, Hazlett,
situation with multiple distinct binary treatments administered simultaneously.
In the future it will be extended to other settings including the generalization
of experimental and instrumental variable estimates. Recently add the optimal
CBPS which chooses the optimal balancing function and results in doubly robust
and efficient estimator for the treatment effect.

LazyLoad yes
LazyData yes
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2016-12-30 00:13:52
Description

Returns the mean and standardized mean associated with each treatment group, before and after weighting.

Usage

```r
## S3 method for class 'CBPS'
balance(object, ...)
```

Arguments

- `object` A CBPS, npCBPS, or CBMSM object.
- `...` Additional arguments to be passed to balance.

Details

For binary and multi-valued treatments as well as marginal structural models, each of the matrices' rows are the covariates and whose columns are the weighted mean, and standardized mean associated with each treatment group. The standardized mean is the weighted mean divided by the standard deviation of the covariate for the whole population. For continuous treatments, returns the absolute Pearson correlation between the treatment and each covariate.

Value

Returns a list of two matrices, "original" (before weighting) and "balanced" (after weighting).

Author(s)

Christian Fong, Marc Ratkovic, and Kosuke Imai.
Examples

### Example: Assess Covariate Balance
###
data(LaLonde)
## Estimate CBPS
fit <- CBPS(treat ~ age + educ + re75 + re74 +
I(re75==0) + I(re74==0),
data = LaLonde, ATT = TRUE)
balance(fit)

Blackwell Data for Covariate Balancing Propensity Score

Description

This data set gives the outcomes as well as treatment assignments and covariates for the example from Blackwell (2013).

Usage

Blackwell

Format

A data frame consisting of 13 columns (including treatment assignment, time, and identifier vectors) and 570 observations.

Source

d.gone.neg is the treatment. d.gone.neg.l1, d.gone.neg.l2, and d.gone.neg.l3 are lagged treatment variables. camp.length, deminc, base.poll, base.und, and office covariates. year is the year of the particular race, and time goes from the first measurement (time = 1) to the election (time = 5). demName is the identifier, and demprcnt is the outcome.

References

CBMSM estimates propensity scores such that both covariate balance and prediction of treatment assignment are maximized. With longitudinal data, the method returns marginal structural model weights that can be entered directly into a linear model. The method also handles multiple binary treatments administered concurrently.

**Usage**

```r
CBMSM(formula, id, time, data, type="MSM", twostep = TRUE,
msm.variance = "approx", time.vary = FALSE, ...)
CBMSM.fit(treat, X, id, time, MultiBin.fit, twostep,
msm.variance, time.vary, ...)
```

**Arguments**

- **formula**: A list of formulas of the form treat ~ X. The function assumes that there is one formula for each time, and they are ordered from the first time to the last time.
- **id**: A vector which identifies the unit associated with each row of treat and X.
- **time**: A vector which identifies the time period associated with each row of treat and X.
- **data**: An optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `CBMSM` is called.
- **twostep**: Set to `TRUE` to use a two-step estimator, which will run substantially faster than continuous-updating. Default is `FALSE`, which uses the continuous-updating estimator described by Imai and Ratkovic (2014).
- **msm.variance**: Default is `FALSE`, which uses the low-rank approximation of the variance described in Imai and Ratkovic (2014). Set to `TRUE` to use the full variance matrix.
- **time.vary**: Default is `FALSE`, which uses the same coefficients across time period. Set to `TRUE` to fit one set per time period.
- **treat**: A vector of treatment assignments. For N observations over T time periods, the length of treat should be N*T.
- **X**: A covariate matrix. For N observations over T time periods, X should have N*T rows.
- **type**: "MSM" for a marginal structural model, with multiple time periods or "Multi-Bin" for multiple binary treatments at the same time period.
Multibin.fit  A parameter for whether the multiple binary treatments occur concurrently (FALSE) or over consecutive time periods (TRUE) as in a marginal structural model. Setting type = "MultiBin" when calling CBMSM will set Multibin.fit to TRUE when CBMSM.fit is called.

... Other parameters to be passed through to optim()

Details

Fits covariate balancing propensity scores for marginal structural models.

Value

- weights  The optimal weights.
- fitted.values  The fitted propensity score for each observation.
- y  The treatment vector used.
- x  The covariate matrix.
- id  The vector id used in CBMSM.fit.
- time  The vector time used in CBMSM.fit.
- model  The model frame.
- call  The matched call.
- formula  The formula supplied.
- data  The data argument.
- treat.hist  A matrix of the treatment history, with each observation in rows and time in columns.
- treat.cum  A vector of the cumulative treatment history, by individual.

Author(s)

Marc Ratkovic, Christian Fong, and Kosuke Imai; The CBMSM function is based on the code for version 2.15.0 of the glm function implemented in the stats package, originally written by Simon Davies. This documentation is likewise modeled on the documentation for glm and borrows its language where the arguments and values are the same.

References


See Also

plot.CBMSM
**Examples**

```r
# Load Blackwell data
data(Blackwell)

## Quickly fit a short model to test
form0 <- "d.gone.neg ~ d.gone.neg.l1 + camp.length"
fit0 <- CBMSM(formula = form0, time=Blackwell$time, id=Blackwell$demName,
   data=Blackwell, type="MSM", iterations = NULL, twostep = TRUE,
   msm.variance = "approx", time.vary = FALSE)

## Not run:
## Fitting the models in Imai and Ratkovic (2014)
## Warning: may take a few mintues; setting time.vary to FALSE
## Usually, it is best to use time.vary TRUE
form1<-"d.gone.neg ~ d.gone.neg.l1 + d.gone.neg.l2 + d.neg.frac.l3 +
camp.length + camp.length + deminc + base.poll + year.2002 +
year.2004 + year.2006 + base.und + office"
fit1 <- CBMSM(formula = form1, time=Blackwell$time, id=Blackwell$demName,
   data=Blackwell, type="MSM", iterations = NULL, twostep = TRUE,
   msm.variance = "full", time.vary = TRUE)

fit2 <- CBMSM(formula = form1, time=Blackwell$time, id=Blackwell$demName,
   data=Blackwell, type="MSM", iterations = NULL, twostep = TRUE,
   msm.variance = "approx", time.vary = TRUE)

## Assessing balance
bal1 <- balance.CBMSM(fit1)
bal2 <- balance.CBMSM(fit2)

## Effect estimation: Replicating Effect Estimates in
## Table 3 of Imai and Ratkovic (2014)

lm1 <- lm(demprcnt[time==1]-fit1$treat.hist, data=Blackwell,
   weights=fit1$glm.weights)
lm2 <- lm(demprcnt[time==1]-fit1$treat.hist, data=Blackwell,
   weights=fit1$weights)
lm3 <- lm(demprcnt[time==1]-fit1$treat.hist, data=Blackwell,
   weights=fit2$weights)
lm4 <- lm(demprcnt[time==1]-fit1$treat.cum, data=Blackwell,
   weights=fit1$glm.weights)
lm5 <- lm(demprcnt[time==1]-fit1$treat.cum, data=Blackwell,
   weights=fit1$weights)
lm6 <- lm(demprcnt[time==1]-fit1$treat.cum, data=Blackwell,
   weights=fit2$weights)
```
CBPS

Covariate Balancing Propensity Score (CBPS) Estimation

Description

CBPS estimates propensity scores such that both covariate balance and prediction of treatment assignment are maximized. The method, therefore, avoids an iterative process between model fitting and balance checking and implements both simultaneously. For cross-sectional data, the method can take continuous treatments and treatments with a control (baseline) condition and either 1, 2, or 3 distinct treatment conditions.

Usage

CBPS(formula, data, na.action, ATT = 1, iterations = 1000, standardize = TRUE, method = "over", twostep = TRUE, baseline.formula = NULL, diff.formula = NULL, ...)
CBPS.fit(treat, X, baselineX, diffX, ATT, method, iterations, standardize, twostep, ...)

Arguments

formula An object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data: An optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which CBPS is called.

na.action: A function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset.

ATT: Default is 1, which finds the average treatment effect on the treated interpreting the second level of the treatment factor as the treatment. Set to 2 to find the ATT interpreting the first level of the treatment factor as the treatment. Set to 0 to find the average treatment effect. For non-binary treatments, only the ATE is available.

iterations: An optional parameter for the maximum number of iterations for the optimization. Default is 1000.

standardize: Default is TRUE, which normalizes weights to sum to 1 within each treatment group. For continuous treatments, normalizes weights to sum up to 1 for the entire sample. Set to FALSE to return Horvitz-Thompson weights.

method: Choose "over" to fit an over-identified model that combines the propensity score and covariate balancing conditions; choose "exact" to fit a model that only contains the covariate balancing conditions.

twostep: Default is TRUE for a two-step estimator, which will run substantially faster than continuous-updating. Set to FALSE to use the continuous-updating estimator described by Imai and Ratkovic (2014).

treat: A vector of treatment assignments. Binary or multi-valued treatments should be factors. Continuous treatments should be numeric.

X: A covariate matrix.

baseline.formula: Used only to fit CBPSOptimal (see Fan et al). Currently only works with binary treatments. A formula specifying the balancing covariates in the baseline outcome model. Namely $h_1()$. Eg. ~X2+X4 means X_2 and X_4 are the baseline outcome model covariates.

diff.formula: Used only to fit CBPSOptimal (see Fan et al). Currently only works with binary treatments. A formula specifying the balancing covariates in the difference between the treatment and baseline outcome model. Namely $h_2()$. Eg. ~X1+X3 means X_1 and X_3 are covariates in the difference between the treatment and baseline outcome models.

baselineX: Similar to baseline.formula, but in matrix form.

diffX: Similar to diff.formula, but in matrix form.

...: Other parameters to be passed through to optim().

Details

Fits covariate balancing propensity scores.
Value

fitted.values
The fitted propensity score

deviance
Minus twice the log-likelihood of the CBPS fit

weights
The optimal weights. Let \( \pi_i = f(T_i | X_i) \). For binary ATE, these are given by
\[
\frac{T_i^*}{\pi_i} + \frac{(1-T_i^*)}{(1-\pi_i)}. \]
For binary ATT, these are given by \( \frac{\pi_i - T_i}{1-\pi_i} \). For multi_valued treatments, these are given by \( \sum_{j=0}^{J-1} T_{i,j} / \pi_{i,j} \). For continuous treatments, these are given by \( f(T_i) / f(T_i | X_i) \). These expressions for weights are all before standardization (i.e. with standardize=FALSE). Standardization will make weights sum to 1 within each treatment group. For continuous treatment, standardization will make all weights sum to 1.

y
The treatment vector used

x
The covariate matrix

model
The model frame

converged
Convergence value. Returned from the call to optim().

call
The matched call

formula
The formula supplied

data
The data argument

coefficients
A named vector of coefficients

sigmasq
The sigma-squared value, for continuous treatments only

J
The J-statistic at convergence

mle.J
The J-statistic for the parameters from maximum likelihood estimation

var
The covariance matrix, evaluated numerically from optim().

Author(s)

Christian Fong, Marc Ratkovic, Kosuke Imai, and Xiaolin Yang; The CBPS function is based on the code for version 2.15.0 of the glm function implemented in the stats package, originally written by Simon Davies. This documentation is likewise modeled on the documentation for glm and borrows its language where the arguments and values are the same.

References


### Example: propensity score matching

```r
# Load the LaLonde data
data(LaLonde)

# Estimate CBPS
fit <- CBPS(treat ~ age + educ + re75 + re74 + I(re75==0) + I(re74==0),
data = LaLonde, ATT = TRUE)
summary(fit)

# Not run:
# matching via MatchIt: one to one nearest neighbor with replacement
library(MatchIt)
m.out <- matchit(treat ~ fitted(fit), method = "nearest",
data = LaLonde, replace = TRUE)
```

### Example: propensity score weighting

```r
# Simulation from Kang and Shafer (2007).
set.seed(123456)
n <- 500
X <- mvrnorm(n, mu = rep(0, 4), Sigma = diag(4))
prop <- 1 / (1 + exp(X[,1] - 0.5 * X[,2] + 0.25*X[,3] + 0.1 * X[,4]))
treat <- rbinom(n, 1, prop)

# Estimate CBPS with a misspecified model
X.mis <- cbind(exp(X[,1]/2), X[,2]*exp(X[,1]), X[,3]+exp(X[,3])/25+.6, X[,2]*X[,4]+20)^2
fit1 <- CBPS(treat - X.mis, ATT = 0)
summary(fit1)

# Horwitz-Thompson estimate
mean(treat*y/fit1$fitted.values)
# Inverse propensity score weighting
sum(treat*y/fit1$fitted.values)/sum(treat/fit1$fitted.values)
```

### Example: Continuous Treatment

```r
set.seed(123456)
n <- 1000
X < mvrnorm(n, mu = rep(0,2), Sigma = diag(2))
beta <- rnorm(ncol(X)+1, sd = 1)
treat <- cbind(1,X)%%beta + rnorm(n, sd = 5)
```
LaLonde

### Example: Improved CBPS from Fan et al
```r
set.seed(123456)
n <- 500
X <- mvrnorm(n, mu = rep(0, 4), Sigma = diag(4))
prop <- 1 / (1 + exp(X[,1] - 0.5 * X[,2] + 0.25*X[,3] + 0.1 * X[,4]))
treat <- rbinom(n, 1, prop)
# Estimate CBPS with a misspecified model
X.mis <- cbind(exp(X[,1]/10), X[,2]*(1+exp(X[,1]))^(-1)+10, 
              (X[,1]*X[,3]/25+.6)^3, (X[,2]+X[,4]+20)^2)
fit1 <- CBPS(treat ~ X.mis, baseline.formula=~X.mis[,2:4], 
             diff.formula=~X.mis[,1], ATT = FALSE)
summary(fit1)
```

### LaLonde Data for Covariate Balancing Propensity Score

**Description**

This data set gives the outcomes as well as treatment assignments and covariates for the econometric evaluation of training programs in LaLonde (1986).

**Usage**

LaLonde

**Format**

A data frame consisting of 12 columns (including a treatment assignment vector) and 3212 observations.
Source

Data from the National Supported Work Study. A benchmark matching dataset. Columns consist of an indicator for whether the observed unit was in the experimental subset; an indicator for whether the individual received the treatment; age in years; schooling in years; indicators for black and Hispanic; an indicator for marriage status, one of married; an indicator for no high school degree; reported earnings in 1974, 1975, and 1978; and whether the 1974 earnings variable is missing. Data not missing 1974 earnings are the Dehejia-Wahba subsample of the LaLonde data. Missing values for 1974 earnings set to zero. 1974 and 1975 earnings are pre-treatment. 1978 earnings is taken as the outcome variable.

References


npCBPS

Non-Parametric Covariate Balancing Propensity Score (npCBPS) Estimation

Description

npCBPS is a method to estimate weights interpretable as (stabilized) inverse generalized propensity score weights, \( w_i = \frac{f(T|X)}{f(T_i|X)} \), without actually estimating a model for the treatment to arrive at \( f(T|X) \) estimates. In brief, this works by maximizing the empirical likelihood of observing the values of treatment and covariates that were observed, while constraining the weights to be those that (a) ensure balance on the covariates, and (b) maintain the original means of the treatment and covariates.

In the continuous treatment context, this balance on covariates means zero correlation of each covariate with the treatment. In binary or categorical treatment contexts, balance on covariates implies equal means on the covariates for observations at each level of the treatment. When given a numeric treatment the software handles it continuously. To handle the treatment as binary or categorical is must be given as a factor.

Furthermore, we apply a Bayesian variant that allows the correlation of each covariate with the treatment to be slightly non-zero, as might be expected in a a given finite sample.

Usage

```r
npCBPS(formula, data, na.action, corprior = 0.01,
      print.level = 0, ...)
npCBPS.fit(treat, X, corprior, print.level, ...)
```

Arguments

- `formula`: An object of class `formula` (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data

An optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which CBPS is called.

na.action

A function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset.

corprior

Prior hyperparameter controlling the expected amount of correlation between each covariate and the treatment. Specifically, the amount of correlation between the k-dimensional covariates, X, and the treatment T after weighting is assumed to have prior distribution MVN(0,sigma^2 I_k). We conceptualize sigma^2 as a tuning parameter to be used pragmatically. It’s default of 0.1 ensures that the balance constraints are not too harsh, and that a solution is likely to exist. Once the algorithm works at such a high value of sigma^2, the user may wish to attempt values closer to 0 to get finer balance.

print.level

Controls verbosity of output to the screen while npCBPS runs. At the default of print.level=0, little output is produced. It print.level>0, it outputs diagnostics including the log posterior (log_post), the log empirical likelihood associated with the weights (log_el), and the log prior probability of the (weighted) correlation of treatment with the covariates.

treat

A vector of treatment assignments. Binary or multi-valued treatments should be factors. Continuous treatments should be numeric.

X

A covariate matrix.

... Other parameters to be passed.

Details

Estimates non-parametric covariate balancing propensity score weights.

Value

weights

The optimal weights

y

The treatment vector used

x

The covariate matrix

model

The model frame

call

The matched call

formula

The formula supplied

data

The data argument

log.p.eta

The log density for the (weighted) correlation of the covariates with the treatment, given the choice of prior (corprior)

log.el

The log empirical likelihood of the observed data at the chosen set of IPW weights.

eta

A vector describing the correlation between the treatment and each covariate on the weighted data at the solution.
The sum of weights, provided as a check on convergence. This is always 1 when convergence occurs unproblematically. If it differs from 1 substantially, no solution perfectly satisfying the conditions was found, and the user may consider a larger value of corprior.

Author(s)

Christian Fong, Chad Hazlett, and Kosuke Imai

References


Examples

```r
## Generate data
data(LaLonde)

## Restricted two only two covariates so that it will run quickly.
## Performance will remain good if the full LaLonde specification is used
fit <- npCBPS(treat ~ age + educ, data = LaLonde, corprior = 1/nrow(LaLonde))
plot(fit)
```

plot.CBMSM

Plotting CBPS Estimation for Marginal Structural Models

Description

Plots the absolute difference in standardized means before and after weighting.

Usage

```r
## S3 method for class 'CBMSM'
plot(x, covars = NULL, silent = TRUE, boxplot = FALSE, ...)
```

Arguments

- `x`: an object of class “CBMSM”.
- `covars`: Indices of the covariates to be plotted (excluding the intercept). For example, if only the first two covariates from `balance` are desired, set `covars` to 1:2. The default is `NULL`, which plots all covariates.
- `silent`: If set to `FALSE`, returns the absolute imbalance for each treatment history pair before and after weighting. This helps the user to create his or her own customized plot. Default is `TRUE`, which returns nothing.
- `boxplot`: If set to `TRUE`, returns a boxplot summarizing the imbalance on the covariates instead of a point for each covariate. Useful if there are many covariates.
- `...`: Additional arguments to be passed to `plot`.
Details
Covariate balance is improved if the plot’s points are below the plotted line of y=x.

Value
The x-axis gives the imbalance for each covariate-treatment history pair without any weighting, and the y-axis gives the imbalance for each covariate-treatment history pair after CBMSM weighting. Imbalance is measured as the absolute difference in standardized means for the two treatment histories. Means are standardized by the standard deviation of the covariate in the full sample.

Author(s)
Marc Ratkovic and Christian Fong

See Also
CBMSM, plot

plot.CBPS
Plotting Covariate Balancing Propensity Score Estimation

Description
Plots the absolute difference in standardized means before and after weighting.

Usage
## S3 method for class 'CBPS'
plot(x, covars = NULL, silent = TRUE, boxplot = FALSE, ...)

Arguments
- **x**: an object of class “CBPS” or “npCBPS”, usually, a result of a call to CBPS or npCBPS.
- **covars**: Indices of the covariates to be plotted (excluding the intercept). For example, if only the first two covariates from balance are desired, set covars to 1:2. The default is NULL, which plots all covariates.
- **silent**: If set to FALSE, returns the imbalances used to construct the plot. Default is TRUE, which returns nothing.
- **boxplot**: If set to TRUE, returns a boxplot summarizing the imbalance on the covariates instead of a point for each covariate. Useful if there are many covariates.
- **...**: Additional arguments to be passed to plot.

Details
The "Before Weighting" plot gives the balance before weighting, and the "After Weighting" plot gives the balance after weighting.
Value

For binary and multi-valued treatments, plots the absolute difference in standardized means by contrast for all covariates before and after weighting. This quantity for a single covariate and a given pair of treatment conditions is given by

\[
\frac{\sum_{i=1}^{n} w_i \cdot (T_i = 1) \cdot x_i}{\sum_{i=1}^{n} w_i \cdot (T_i = 1)} - \frac{\sum_{i=1}^{n} w_i \cdot (T_i = 0) \cdot x_i}{\sum_{i=1}^{n} w_i}.
\]

For continuous treatments, plots the weighted absolute Pearson correlation between the treatment and each covariate. See https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient#Weighted_correlation_coefficient.

Author(s)

Christian Fong, Marc Ratkovic, and Kosuke Imai.

See Also

CBPS, plot

Description

Prints a summary of a fitted CBPS object.

Usage

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

Arguments

- `object`: an object of class “CBPS”, usually, a result of a call to CBPS.
- `...`: Additional arguments to be passed to summary.

Details

Prints a summary of a CBPS object, in a format similar to glm. The variance matrix is calculated from the numerical Hessian at convergence of CBPS.

Value

- `call`: The matched call.
- `deviance.residuals`: The five number summary and the mean of the deviance residuals.
- `coefficients`: A table including the estimate for each coefficient and the standard error, z-value, and two-sided p-value for these estimates.
- `J`: Hansen’s J-Statistic for the fitted model.
- `Log-Likelihood`: The log-likelihood of the fitted model.
vcov.CBPS

Author(s)
Christian Fong, Marc Ratkovic, and Kosuke Imai.

See Also
CBPS, summary

vcov.CBPS

Calculate Variance-Covariance Matrix for a Fitted CBPS Object

Description
vcov.CBPS Returns the variance-covariance matrix of the main parameters of a fitted CBPS object.

Usage
## S3 method for class 'CBPS'
vcov(object,...)

Arguments
object An object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted.
... Additional arguments to be passed to vcov.CBPS

Details
This is the CBPS implementation of the generic function vcov().

Value
A matrix of the estimated covariances between the parameter estimates in the linear or non-linear predictor of the model.

Author(s)
Christian Fong, Marc Ratkovic, and Kosuke Imai.

References
This documentation is modeled on the documentation of the generic vcov.

See Also
vcov
Examples

```r
###
### Example: Variance-Covariance Matrix
###

## Load the LaLonde data
data(LaLonde)
## Estimate CBPS via logistic regression
fit <- CBPS(treat ~ age + educ + re75 + re74 + I(re75==0) + I(re74==0),
            data = LaLonde, ATT = TRUE)
## Get the variance-covariance matrix.
vcov(fit)
```
Index

*Topic **datasets**
  - Blackwell, 3
  - LaLonde, 11

balance, 2
Blackwell, 3

CBMSM, 4, 15
CBPS, 7, 16, 17
LaLonde, 11
npCBPS, 12

plot, 15, 16
plot.CBMSM, 5, 14
plot.CBPS, 15
plot.npCBPS (plot.CBPS), 15
print.CBPS (CBPS), 7

summary, 17
summary.CBPS, 10, 16

vcov, 17
vcov.CBPS, 17