Package 'experiment'

February 19, 2015

Version 1.1-1

Date 2013-03-28

Title experiment: R package for designing and analyzing randomized experiments

Author Kosuke Imai <kimai@princeton.edu>

Maintainer Kosuke Imai <kimai@princeton.edu>

Depends boot, MASS, R (>= 2.4.0)

Description The package provides various statistical methods for designing and analyzing randomized experiments. One main functionality of the package is the implementation of randomized-block and matched-pair designs based on possibly multivariate pre-treatment covariates. The package also provides the tools to analyze various randomized experiments including cluster randomized experiments, randomized experiments with noncompliance, and randomized experiments with missing data.

License GPL (>= 2)

URL http://imai.princeton.edu/software/experiment.html

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-03-29 08:00:15

R topics documented:

ATEbounds 2
ATEcluster 4
ATEnocov 6
CACEcluster 7
NoncompLI 8
randomize 12

Index **15**

ATEbounds *Bounding the Average Treatment Effect when some of the Outcome Data are Missing*

Description

This function computes the sharp bounds on the average treatment effect when some of the outcome data are missing. The confidence intervals for the bounds are also computed.

Usage

```
ATEbounds(formula, data = parent.frame(), maxY = NULL,
          minY = NULL, alpha = 0.05, n.reps = 0,
          strata = NULL, ratio = NULL, survey = NULL, ...)
```

Arguments

formula	A formula of the form `Y ~ X` where `Y` is the name of the outcome variable and `X` is the name of the (randomized) treatment variable. `X` should be a factor variable but its value can take more than two levels. The missing values for `Y` should be coded as `NA`.
data	A data frame containing the relevant variables.
maxY	A scalar. The maximum value of the outcome variable. The default is the maximum sample value.
minY	A scalar. The minimum value of the outcome variable. The default is the minimum sample value.
alpha	A positive scalar that is less than or equal to 0.5. This will determine the (1-`alpha`) level of confidence intervals. The default is `0.05`.
strata	The variable name indicating strata. If this is specified, the quantities of interest will be first calculated within each strata and then aggregated. The default is `NULL`.
ratio	A $J \times M$ matrix of probabilities where J is the number of strata and M is the number of treatment and control groups. Each element of the matrix specifies the probability of a unit falling into that category. The default is `NULL` in which case the sample estimates of these probabilities are used for computation.
survey	The variable name for survey weights. The default is `NULL`.
n.reps	A positive integer. The number of bootstrap replicates used for the construction of confidence intervals via B-method of Berran (1988). If it equals zero, the confidence intervals will not be constructed.
...	The arguments passed to other functions.

Details

For the details of the method implemented by this function, see the references.

Value

A list of class `ATEbounds` which contains the following items:

`call`	The matched call.
`Y`	The outcome variable.
`D`	The treatment variable.
`bounds`	The point estimates of the sharp bounds on the average treatment effect.
`bounds.Y`	The point estimates of the sharp bounds on the outcome variable within each treatment/control group.
`bmethod.ci`	The B-method confidence interval of the bounds on the average treatment effect.
`bonf.ci`	The Bonferroni confidence interval of the bounds on the average treatment effect.
`bonf.ci.Y`	The Bonferroni confidence interval of the bounds on the outcome variable within each treatment/control group.
`bmethod.ci.Y`	The B-method confidence interval of the bounds on the outcome variable within each treatment/control group.
`maxY`	The maximum value of the outcome variable used in the computation.
`minY`	The minimum value of the outcome variable used in the computation.
`nobs`	The number of observations.
`nobs.Y`	The number of observations within each treatment/control group.
`ratio`	The probability of treatment assignment (within each strata if `strata` is specified) used in the computation.

Author(s)

Kosuke Imai, Department of Politics, Princeton University <kimai@Princeton.Edu>, http://imai.princeton.edu;

References

Horowitz, Joel L. and Charles F. Manski. (1998). "Censoring of Outcomes and Regressors due to Survey Nonresponse: Identification and Estimation Using Weights and Imputations." *Journal of Econometrics*, Vol. 84, pp.37-58.

Horowitz, Joel L. and Charles F. Manski. (2000). "Nonparametric Analysis of Randomized Experiments With Missing Covariate and Outcome Data." *Journal of the Americal Statistical Association*, Vol. 95, No. 449, pp.77-84.

Harris-Lacewell, Melissa, Kosuke Imai, and Teppei Yamamoto. (2007). "Racial Gaps in the Responses to Hurricane Katrina: An Experimental Study", *Technical Report*. Department of Politics, Princeton University.

ATEcluster *Estimation of the Average Treatment Effects in Cluster-Randomized Experiments*

Description

This function estimates various average treatment effect in cluster-randomized experiments without using pre-treatment covariates. The treatment variable is assumed to be binary. Currently, only the matched-pair design is allowed. The details of the methods for this design are given in Imai, King, and Nall (2007).

Usage

```
ATEcluster(Y, Z, grp, data = parent.frame(), match = NULL,
           weights = NULL, fpc = TRUE)
```

Arguments

Y	The outcome variable of interest.
Z	The (randomized) cluster-level treatment variable. This variable should be binary. Two units in the same cluster should have the same value.
grp	A variable indicating clusters of units. Two units in the same cluster should have the same value.
data	A data frame containing the relevant variables.
match	A variable indicating matched-pairs of clusters. Two units in the same matched-pair of clusters should have the same value. The default is NULL (i.e., no matching).
weights	A variable indicating the population cluster sizes, which will be used to construct weights for each pair of clusters. Two units in the same cluster should have the same value. The default is NULL, in which case sample cluster sizes will be used for constructing weights.
fpc	A logical variable indicating whether or not finite population correction should be used for estimating the lower bound of CACE variance. This is relevant only when weights are specified.

Value

A list of class ATEcluster which contains the following items:

call	The matched call.
n	The total number of units.
n1	The total number of units in the treatment group.
n0	The total number of units in the control group.
Y	The outcome variable.

Y1bar	The cluster-specific (unweighted) average value of the observed outcome for the treatment group.
Y0bar	The cluster-specific (unweighted) average value of the observed outcome for the treatment group.
Y1var	The cluster-specific sample variance of the observed outcome for the treatment group.
Y0var	The cluster-specific sample variance of the observed outcome for the control group.
Z	The treatment variable.
grp	The cluster-indicator variable.
match	The matched-pair indicator variable.
weights	The weight variable in its original form.
est	The estimated average treatment effect based on the arithmetic mean weights.
var	The estimated variance of the average treatment effect estimator based on the arithmetic mean weights. This uses the variance formula provided in Imai, King, and Nall (2007).
var.lb	The estimated sharp lower bound of the cluster average treatment effect estimator using the arithmetic mean weights.
est.dk	The estimated average treatment effect based on the harmonic mean weights.
var.dk	The estimated variance of the average treatment effect estimator based on the harmonic mean weights. This uses the variance formula provided in Donner and Klar (1993).
dkvar	The estimated variance of the average treatment effect estimator based on the harmonic mean weights. This uses the variance formula provided in Imai, King, and Nall (2007).
eff	The estimated relative efficiency of the matched-pair design over the completely randomized design (the ratio of two estimated variances).
m	The number of pairs in the matched-pair design.
N1	The population cluster sizes for the treatment group.
N0	The population cluster sizes for the control group.
w1	Cluster-specific weights for the treatment group.
w0	Cluster-specific weights for the control group.
w	Pair-specific normalized arithmetic mean weights. These weights sum up to the total number of units in the sample, i.e., n.
w.dk	Pair-specific normalized harmonic mean weights. These weights sum up to the total number of units in the sample, i.e., n.
diff	Within-pair differences if the matched-pair design is analyzed. This equals the difference between Y1bar and Y0bar.

Author(s)

Kosuke Imai, Department of Politics, Princeton University <kimai@Princeton.Edu>, http://imai.princeton.edu;

References

Donner, A. and N. Klar (1993). "Confidence interval construction for effect measures arising from cluster randomized trials." Journal of Clinical Epidemiology. Vol. 46, No. 2, pp. 123-131.

Imai, Kosuke, Gary King, and Clayton Nall (2007). "The Essential Role of Pair Matching in Cluster-Randomized Experiments, with Application to the Mexican Universal Health Insurance Evaluation", Technical Report. Department of Politics, Princeton University.

`ATEnocov` *Estimation of the Average Treatment Effect in Randomized Experiments*

Description

This function computes the standard "difference-in-means" estimate of the average treatment effect in randomized experiments without using pre-treatment covariates. The treatment variable is assumed to be binary. Currently, the two designs are allowed: complete randomized design and matched-pair design.

Usage

```
ATEnocov(Y, Z, data = parent.frame(), match = NULL)
```

Arguments

`Y`	The outcome variable of interest.
`Z`	The (randomized) treatment variable. This variable should be binary.
`data`	A data frame containing the relevant variables.
`match`	A variable indicating matched-pairs. The two units in the same matched-pair should have the same value.

Value

A list of class `ATEnocov` which contains the following items:

`call`	The matched call.
`Y`	The outcome variable.
`Z`	The treatment variable.
`match`	The matched-pair indicator variable.
`ATEest`	The estimated average treatment effect.
`ATE.var`	The estimated variance of the average treatment effect estimator.
`diff`	Within-pair differences if the matched-pair design is analyzed.

Author(s)

Kosuke Imai, Department of Politics, Princeton University <kimai@Princeton.Edu>, http://imai.princeton.edu;

References

Imai, Kosuke, (2007). "Randomization-based Inference and Efficiency Analysis in Experiments under the Matched-Pair Design", Technical Report. Department of Politics, Princeton University.

`CACEcluster` *Estimation of the Complier Average Causal Effects in Cluster-Randomized Experiments with Unit-level Noncompliance*

Description

This function estimates various complier average causal effect in cluster-randomized experiments without using pre-treatment covariates when unit-level noncompliance exists. Both the encouragement and treatment variables are assumed to be binary. Currently, only the matched-pair design is allowed. The details of the methods for this design are given in Imai, King, and Nall (2007).

Usage

```
CACEcluster(Y, D, Z, grp, data = parent.frame(), match = NULL,
            weights = NULL, ...)
```

Arguments

`Y`	The outcome variable of interest.
`D`	The unit-level treatment receipt variable. This variable should be binary but can differ across units within each cluster.
`Z`	The (randomized) cluster-level encouragement variable. This variable should be binary. Two units in the same cluster should have the same value.
`grp`	A variable indicating clusters of units. Two units in the same cluster should have the same value.
`data`	A data frame containing the relevant variables.
`match`	A variable indicating matched-pairs of clusters. Two units in the same matched-pair of clusters should have the same value. The default is `NULL` (i.e., no matching).
`weights`	A variable indicating the population cluster sizes, which will be used to construct weights for each pair of clusters. Two units in the same cluster should have the same value. The default is `NULL`, in which case sample cluster sizes will be used for constructing weights.
`...`	Optional arguments passed to `ATEcluster`, which is called internally.

Value

A list of class `CACEcluster` which contains the following items:

`call`	The matched call.
`ITTY`	The output object from `ATEcluster` which is used to estimate the ITT effect of the encouragement on the outcome variable.
`ITTD`	The output object from `ATEcluster` which is used to estimate the ITT effect of the encouragement on the treatment receipt variable.
`n1`	The total number of units in the treatment group.
`n0`	The total number of units in the control group.
`Z`	The treatment variable.
`est`	The estimated complier average causal effect.
`var`	The estimated variance of the complier average causal effect estimator.
`cov`	The estimated covariance between two ITT estimator.
`m`	The number of pairs in the matched-pair design.
`N1`	The population cluster sizes for the treatment group.
`N0`	The population cluster sizes for the control group.
`w`	Pair-specific normalized arithmetic mean weights. These weights sum up to the total number of units in the sample, i.e., `n`.

Author(s)

Kosuke Imai, Department of Politics, Princeton University <kimai@Princeton.Edu>, http://imai.princeton.edu;

References

Imai, Kosuke, Gary King, and Clayton Nall (2007). "The Essential Role of Pair Matching in Cluster-Randomized Experiments, with Application to the Mexican Universal Health Insurance Evaluation", Technical Report. Department of Politics, Princeton University.

`NoncompLI` *Bayesian Analysis of Randomized Experiments with Noncompliance and Missing Outcomes Under the Assumption of Latent Ignorability*

Description

This function estimates the average causal effects for randomized experiments with noncompliance and missing outcomes under the assumption of latent ignorability (Frangakis and Rubin, 1999). The models are based on Bayesian generalized linear models and are fitted using the Markov chain Monte Carlo algorithms. Various types of the outcome variables can be analyzed to estimate the Intention-to-Treat effect and Complier Average Causal Effect.

Usage

```
NoncompLI(formulae, Z, D, data = parent.frame(), n.draws = 5000,
          param = TRUE, in.sample = FALSE, model.c = "probit",
          model.o = "probit", model.r = "probit",
          tune.c = 0.01, tune.o = 0.01, tune.r = 0.01,
          tune.v = 0.01, p.mean.c = 0, p.mean.o = 0,
          p.mean.r = 0, p.prec.c = 0.001,
          p.prec.o = 0.001, p.prec.r = 0.001,
          p.df.o = 10, p.scale.o = 1, p.shape.o = 1,
          mda.probit = TRUE, coef.start.c = 0,
          coef.start.o = 0, tau.start.o = NULL,
          coef.start.r = 0, var.start.o = 1,
          burnin = 0, thin = 0, verbose = TRUE)
```

Arguments

formulae	A list of formulae where the first formula specifies the (pre-treatment) covariates in the outcome model (the latent compliance covariate will be added automatically), the second formula specifies the compliance model, and the third formula defines the covariate specification for the model for missing-data mechanism (the latent compliance covariate will be added automatically). For the outcome model, the formula should take the two-sided standard R formula where the outcome variable is specified in the left hand side of the formula which is then separated by ~ from the covariate equation in the right hand side, e.g., y ~ x1 + x2. For the compliance and missing-data mechanism models, the one-sided formula should be used where the left hand side is left unspecified, e.g., ~ x1 + x2.
Z	A randomized encouragement variable, which should be a binary variable in the specified data frame.
D	A treatment variable, which should be a binary variable in the specified data frame.
data	A data frame which contains the variables that appear in the model formulae (formulae), the encouragement variable (Z), and the treatment variable (D).
n.draws	The number of MCMC draws. The default is 5000.
param	A logical variable indicating whether the Monte Carlo draws of the model parameters should be saved in the output object. The default is TRUE.
in.sample	A logical variable indicating whether or not the sample average causal effect should be calculated using the observed potential outcome for each unit. If it is set to FALSE, then the population average causal effect will be calculated. The default is FALSE.
model.c	The model for compliance. Either logit or probit model is allowed. The default is probit.
model.o	The model for outcome. The following five models are allowed: logit, probit, oprobit (ordered probit regression), gaussian (gaussian regression), negbin (negative binomial regression), and twopart (two part model where the first part is the probit regression for $Pr(Y > 0\|X)$ and the second part models $p(log(Y)\|X, Y > 0)$ using the gaussian regression). The default is probit.

`model.r` The model for (non)response. Either `logit` or `probit` model is allowed. The default is `probit`.

`tune.c` Tuning constants for fitting the compliance model. These positive constants are used to tune the (random-walk) Metropolis-Hastings algorithm to fit the logit model. Use either a scalar or a vector of constants whose length equals that of the coefficient vector. The default is `0.01`.

`tune.o` Tuning constants for fitting the outcome model. These positive constants are used to tune the (random-walk) Metropolis-Hastings algorithm to fit logit, ordered probit, and negative binomial models. Use either a scalar or a vector of constants whose length equals that of the coefficient vector for logit and negative binomial models. For the ordered probit model, use either a scalar or a vector of constants whose length equals that of cut-point parameters to be estimated. The default is `0.01`.

`tune.r` Tuning constants for fitting the (non)response model. These positive constants are used to tune the (random-walk) Metropolis-Hastings algorithm to fit the logit model. Use either a scalar or a vector of constants whose length equals that of the coefficient vector. The default is `0.01`.

`tune.v` A scalar tuning constant for fitting the variance component of the negative binomial (outcome) model. The default is `0.01`.

`p.mean.c` Prior mean for the compliance model. It should be either a scalar or a vector of appropriate length. The default is `0`.

`p.prec.c` Prior precision for the compliance model. It should be either a positive scalar or a positive semi-definite matrix of appropriate size. The default is `0.001`.

`p.mean.o` Prior mean for the outcome model. It should be either a scalar or a vector of appropriate length. The default is `0`.

`p.prec.o` Prior precision for the outcome model. It should be either a positive scalar or a positive semi-definite matrix of appropriate size. The default is `0.001`.

`p.mean.r` Prior mean for the (non)response model. It should be either a scalar or a vector of appropriate length. The default is `0`.

`p.prec.r` Prior precision for the (non)response model. It should be either a positive scalar or a positive semi-definite matrix of appropriate size. The default is `0.001`.

`p.df.o` A positive integer. Prior degrees of freedom parameter for the inverse chisquare distribution in the gaussian and twopart (outcome) models. The default is `10`.

`p.scale.o` A positive scalar. Prior scale parameter for the inverse chisquare distribution (for the variance) in the gaussian and twopart (outcome) models. For the negative binomial (outcome) model, this is used for the scale parameter of the inverse gamma distribution. The default is `1`.

`p.shape.o` A positive scalar. Prior shape for the inverse chisquare distribution in the negative binomial (outcome) model. The default is `1`.

`mda.probit` A logical variable indicating whether to use marginal data augmentation for probit models. The default is `TRUE`.

`coef.start.c` Starting values for coefficients of the compliance model. It should be either a scalar or a vector of appropriate length. The default is `0`.

coef.start.o	Starting values for coefficients of the outcome model. It should be either a scalar or a vector of appropriate length. The default is 0.
coef.start.r	Starting values for coefficients of the (non)response model. It should be either a scalar or a vector of appropriate length. The default is 0.
tau.start.o	Starting values for thresholds of the ordered probit (outcome) model. If it is set to NULL, then the starting values will be a sequence starting from 0 and then incrementing by 0.1. The default is NULL.
var.start.o	A positive scalar starting value for the variance of the gaussian, negative binomial, and twopart (outcome) models. The default is 1.
burnin	The number of initial burnins for the Markov chain. The default is 0.
thin	The size of thinning interval for the Markov chain. The default is 0.
verbose	A logical variable indicating whether additional progress reports should be prited while running the code. The default is TRUE.

Details

For the details of the model being fitted, see the references. Note that when always-takers exist we fit either two logistic or two probit models by first modeling whether a unit is a complier or a noncomplier, and then modeling whether a unit is an always-taker or a never-taker for those who are classified as non-compliers.

Value

An object of class NoncompLI which contains the following elements as a list:

call	The matched call.
Y	The outcome variable.
D	The treatment variable.
Z	The (randomized) encouragement variable.
R	The response indicator variable for Y.
A	The indicator variable for (known) always-takers, i.e., the control units who received the treatment.
C	The indicator variable for (known) compliers, i.e., the encouraged units who received the treatment when there is no always-takers.
Xo	The matrix of covariates used for the outcome model.
Xc	The matrix of covariates used for the compliance model.
Xr	The matrix of covariates used for the (non)response model.
n.draws	The number of MCMC draws.
QoI	The Monte carlo draws of quantities of interest from their posterior distributions. Quantities of interest include ITT (intention-to-treat) effect, CACE (complier average causal effect), Y1barC (The mean outcome value under the treatment for compliers), Y0barC (The mean outcome value under the control for compliers), YbarN (The mean outcome value for never-takers), YbarA (The mean outcome value for always-takers), pC (The proportion of compliers), pN (The proportion of never-takers), pA (The proportion of always-takers)

If `param` is set to `TRUE`, the following elments are also included:

`coef0`	The Monte carlo draws of coefficients of the outcome model from their posterior distribution.
`coef01`	If `model = "twopart"`, this element contains the Monte carlo draws of coefficients of the outcome model for $p(log(Y)|X, Y > 0)$ from their posterior distribution.
`coefC`	The Monte carlo draws of coefficients of the compliance model from their posterior distribution.
`coefA`	If always-takers exist, then this element contains the Monte carlo draws of coefficients of the compliance model for always-takers from their posterior distribution.
`coefR`	The Monte carlo draws of coefficients of the (non)response model from their posterior distribution.
`sig2`	The Monte carlo draws of the variance parameter for the gaussian, negative binomial, and twopart (outcome) models.

Author(s)

Kosuke Imai, Department of Politics, Princeton University <kimai@Princeton.Edu>, http://imai.princeton.edu;

References

Frangakis, Constantine E. and Donald B. Rubin. (1999). "Addressing Complications of Intention-to-Treat Analysis in the Combined Presence of All-or-None Treatment Noncompliance and Subsequent Missing Outcomes." *Biometrika*, Vol. 86, No. 2, pp. 365-379.

Hirano, Keisuke, Guido W. Imbens, Donald B. Rubin, and Xiao-Hua Zhou. (2000). "Assessing the Effect of an Influenza Vaccine in an Encouragement Design." *Biostatistics*, Vol. 1, No. 1, pp. 69-88.

Barnard, John, Constantine E. Frangakis, Jennifer L. Hill, and Donald B. Rubin. (2003). "Principal Stratification Approach to Broken Randomized Experiments: A Case Study of School Choice Vouchers in New York (with Discussion)", *Journal of the American Statistical Association*, Vol. 98, No. 462, pp299–311.

Horiuchi, Yusaku, Kosuke Imai, and Naoko Taniguchi (2007). "Designing and Analyzing Randomized Experiments: Application to a Japanese Election Survey Experiment." *American Journal of Political Science*, Vol. 51, No. 3 (July), pp. 669-687.

---

randomize *Randomization of the Treatment Assignment for Conducting Experiments*

Description

This function can be used to randomize the treatment assignment for randomized experiments. In addition to the complete randomization, it implements randomized-block and matched-pair designs.

Usage

```
randomize(data, group = c("Treat", "Control"), ratio = NULL,
          indx = NULL, block = NULL, n.block = NULL, match = NULL,
          complete = TRUE)
```

Arguments

data	A data frame containing the observations to which the treatments are randomly assigned.
group	A numerical or character vector indicating the treatment/control groups. The length of the vector equals the total number of such groups. The default specifies two groups called "Treat" and "Control".
ratio	An optional numerical vector which specifies the proportion of the treatment/control groups within the sample. The length of the vector should equal the number of groups. The default is the equal allocation.
indx	An optional variable name in the data frame to be used as the names of the observations. If not specified, the row names of the data frame will be used so long as they are available. If the row names are not available, the integer sequence starting from 1 will be used.
block	An optional variable name in the data frame or a formula to be used as the blocking variables for randomized-block designs. If a variable name is specified, then the unique values of that variable will form blocks unless n.block is specified (see below). If a formula is specified, it will be evaluated using data and then blocking will be based on the mahalanobis distance of the resulting model matrix. In this case, users may want to specify n.block to avoid creating blocks that have too few observations.
n.block	An optional scalar specifying the number of blocks to be created for randomized block designs. If unspecified, the unique values of the blocking variable will define blocks. If specified, the blocks of roughly equal size will be created based on the quantile of the blocking variable.
match	An optional variable name in the data frame or a formula to be used as the matching variables for matched-pair designs. This input is applicable only to the case where there are two groups. Pairs of observations will be formed based on the similar values of the matching variable. If a formula is specified, the mahalanobis distance of the resulting model matrix will be used.
complete	logical. If it equals TRUE (default), then complete randomization will be performed (within each block if randomized block designs are used). Otherwise, simple randomization will be implemented. For matched-pair designs, complete has to equal TRUE.

Details

Randomized-block designs refer to the complete randomization of the treatment within the pre-specified blocks which contain multiple observations. Matched-pair designs refer to the randomization of the binary treatment variable within the pre-specified pair of observations.

Value

A list of class `randomize` which contains the following items:

`call`	the matched call.
`treatment`	The vector of randomized treatments.
`data`	The data frame that was used to conduct the randomization.
`block`	The blocking variable that was used to implement randomized-block designs.
`match`	The matching variable that was used to implement matched-pair designs.
`block.id`	The variable indicating which observations belong to which blocks in randomized-block designs.
`match.id`	The variable indicating which observations belong to which pairs in matched-pair designs.

Author(s)

Kosuke Imai, Department of Politics, Princeton University <kimai@Princeton.Edu>, http://imai.princeton.edu;

Index

∗Topic **design**
ATEbounds, 2
ATEcluster, 4
ATEnocov, 6
CACEcluster, 7
randomize, 12
∗Topic **models**
NoncompLI, 8

ATEbounds, 2
ATEcluster, 4
ATEnocov, 6

CACEcluster, 7

NoncompLI, 8

Randomize *(*randomize*)*, 12
randomize, 12