Package ‘ri’

February 20, 2015

Type Package

Title ri: R package for performing randomization-based inference for experiments

Version 0.9

Date 2012-05-10

Author Peter M. Aronow <peter.aronow@yale.edu> and Cyrus Samii <cds2083@nyu.edu>

Maintainer Cyrus Samii <cds2083@nyu.edu>

Description This package provides a set of tools for conducting exact or approximate randomization-based inference for experiments of arbitrary design. The primary functionality of the package is in the generation, manipulation and use of permutation matrices implied by given experimental designs. Among other features, the package facilitates estimation of average treatment effects, constant effects variance estimation, randomization inference for significance testing against sharp null hypotheses and visualization of data and results.

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-05-15 06:44:09

NeedsCompilation no

R topics documented:

`ri-package` *ri: R package for performing randomization-based inference for experiments*

Description

This package provides a set of tools for conducting exact or approximate randomization-based inference for experiments of arbitrary design. The primary functionality of the package is in the generation, manipulation and use of permutation matrices implied by given experimental designs. Among other features, the package facilitates estimation of average treatment effects, constant effects variance estimation, randomization inference for significance testing against sharp null hypotheses and visualization of data and results.

Details

Package:	ri
Type:	Package
Version:	0.9
Date:	2012-05-10
License:	GPL (>= 2)

This package provides a set of tools for conducting exact or approximate inference for randomized experiments of arbitrary design. The primary functionality of the package is in the generation, manipulation and use of permutation matrices implied by given experimental designs. Among other features, the package facilitates estimation of average treatment effects, constant effects variance estimation and randomization inference for significance testing against sharp null hypotheses.

Author(s)

Peter M. Aronow <peter.aronow@yale.edu> and Cyrus Samii <cds2083@nyu.edu>

Maintainer: Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

`dispdist` *Estimated ATE distribution display, summary and significance testing*

Description

Function for displaying, summarizing and producing p-values from the estimated average treatment effect (ATE) distribution

Usage

```
dispdist(distout, ate, quantiles = c(0.025, 0.975), display.plot = TRUE)
```

Arguments

`distout`	randomization distribution of estimated ATEs, as output from `gendist()`.
`ate`	scalar hypothesized treatment effect for significance testing.
`quantiles`	vector of quantiles of the randomization distribution to be returned. Default is equal-tailed 95% intervals.
`display.plot`	logical for displaying a histogram for the randomization distribution with hypothesized treatment effect overlay. Default is `TRUE`.

Value

`two.tailed.p.value`	two-tailed p-value: twice the smaller of the two one-tailed p-values, as advocated by Rosenbaum (2002)
`two.tailed.p.value.abs`	two-tailed p-value: proportion of randomizations yielding absolute estimated ATE greater than or equal to absolute hypothesized ATE
`greater.p.value`	one-tailed p-value: proportion of randomizations yielding estimated ATE greater than or equal to hypothesized ATE
`lesser.p.value`	one-tailed p-value: proportion of randomizations yielding estimated ATE less than or equal to hypothesized ATE
`quantile`	specified quantiles of the randomization distribution
`sd`	standard deviation of the randomization distribution
`exp.val`	expected value of the randomization distribution

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

Rosenbaum, Paul R. 2002. *Observational Studies*. 2nd ed. New York: Springer.

Samii, Cyrus and Peter M. Aronow. 2012. On Equivalencies Between Design-Based and Regression-Based Variance Estimators for Randomized Experiments. *Statistics and Probability Letters*. 82(2): 365-370. http://dx.doi.org/10.1016/j.spl.2011.10.024

See Also

gendist

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0,2,2,4,1,1)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0)
cluster <- c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9)
block <- c(rep(1,4),rep(2,6),rep(3,8))

perms <- genperms(Z,blockvar=block, clustvar=cluster) # all possible permutations
probs <- genprobexact(Z,blockvar=block, clustvar=cluster) # probability of treatment
ate <- estate(y,Z,prob=probs) # estimate the ATE

## Conduct Sharp Null Hypothesis Test of Zero Effect for Each Unit

Ys <- genouts(y,Z,ate=0) # generate potential outcomes under sharp null of no effect
distout <- gendist(Ys,perms, prob=probs) # generate sampling dist. under sharp null
dispdist(distout, ate)  # display characteristics of sampling dist. for inference

## Generate Sampling Distribution Around Estimated ATE

Ys <- genouts(y,Z,ate=ate) ## generate potential outcomes under tau = ATE
distout <- gendist(Ys,perms, prob=probs) # generate sampling dist. under tau = ATE
dispdist(distout, ate)  ## display characteristics of sampling dist. for inference
```

---

estate *Estimation of average treatment effects*

Description

Function for estimating the average treatment effect (ATE). Permits regression adjustment for covariates, difference estimation (with a pretreatment measure of the outcome variable), inverse probability weighting, and unbiased Horvitz-Thompson estimation.

Usage

```
estate(Y, Z, X = NULL, Ypre = NULL, prob = NULL, HT = FALSE)
```

Arguments

`Y` numeric vector of length N, outcome variable

`Z` binary vector (0 or 1) of length N, treatment indicator

`X` N-by-k numeric matrix of covariates for regression adjustment

`Ypre` numeric vector of length N, pretreatment measure of the outcome variable for difference estimation

`prob` numeric vector within the (0,1) interval of length N, probability of treatment assignment, as outputted by `genprob()` or `genprobexact()`. When `prob=NULL` (the default), assumes uniform probability of assignment to treatment equal to the mean of Z

`HT` when `HT=TRUE`, invokes the Horvitz-Thompson (difference-in-totals) estimator. When `HT=FALSE`, invokes the inverse-probability-weighted regression estimator

Value

a scalar, the estimated average treatment effect

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Peter M. Aronow and Joel A. Middleton. 2012. *A Class of Unbiased Estimators of the Average Treatment Effect in Randomized Experiments*. Working paper, Yale University. http://pantheon.yale.edu/~pma5/unbiasedestimators.pdf

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

Horvitz, D.G. and D.J. Thompson. 1952. A generalization of sampling without replacement from a finite universe. *J. Amer. Statist. Assoc.* 47 663-684.

See Also

`genprob`

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0,2,2,4,1,1)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0)
cluster <- c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9)
block <- c(rep(1,4),rep(2,6),rep(3,8))

probs <- genprobexact(Z,blockvar=block, clustvar=cluster) # probability of treatment
ate <- estate(y,Z,prob=probs) # estimate the ATE
```

estlate *Estimation of local average treatment effects under noncompliance*

Description

Function for estimating the local average treatment effect (LATE) via variants of Wald/2SLS estimation (taking the ratio of two average treatment effect estimators). Permits regression adjustment for covariates, difference estimation (with a pretreatment measure of the outcome variable), inverse probability weighting and Horvitz-Thompson estimation.

Usage

```
estlate(Y, D, Z, X = NULL, Ypre = NULL, Dpre = NULL, prob = NULL, HT = FALSE)
```

Arguments

`Y`	numeric vector of length N, outcome variable
`D`	binary vector (0 or 1) of length N, treatment receipt indicator
`Z`	binary vector (0 or 1) of length N, treatment assignment indicator
`X`	N-by-k numeric matrix of covariates for regression adjustment
`Ypre`	numeric vector of length N, pretreatment measure of the outcome variable for difference estimation
`Dpre`	numeric vector of length N, pretreatment measure of the treatment receipt variable for difference estimation
`prob`	numeric vector within the (0,1) interval of N-length, probability of treatment assignment, as output by `genprob()` or `genprobexact()`. When `prob=NULL` (by default), assumes 0.5 probability of assignment to treatment
`HT`	when `HT=TRUE`, invokes the Horvitz-Thompson (difference-in-totals) estimator. When `HT=FALSE`, invokes the inverse-probability-weighted regression estimator

Value

a numeric scalar, the estimated LATE

Note

Takes the ratio of two `estate` values, the numerator with Y as the outcome variable and Z as the treatment indicator, the denominator with D as the outcome variable and Z as the treatment indicator

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Angrist, Joshua D, Guido W. Imbens and Donald B. Rubin. 1996. Identification of Causal Effects Using Instrumental Variables. *J. Amer. Statist. Assoc.* 91: 444-55.

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

Horvitz, D.G. and D.J. Thompson. 1952. A generalization of sampling without replacement from a finite universe. *J. Amer. Statist. Assoc.* 47: 663-684.

See Also

`estate`

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0,2,2,4,1,1)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0)
D <- c(1,0,0,0,0,0,1,0,1,1,0,1,0,0,1,0,0,1)

cluster <- c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9)
block <- c(rep(1,4),rep(2,6),rep(3,8))
probs <- genprobexact(Z,block,cluster) # generate probability of assignment
late <- estlate(y,D,Z,prob=probs) # estimate the LATE; estimated LATE = 9
```

`gendist` *Generates randomization distribution of estimated ATEs*

Description

Takes hypothesized potential outcomes, a permutation matrix, and arguments for `estate()` to produce a randomization distribution of estimated average treatment effects (ATEs).

Usage

```
gendist(Ys, perms, X = NULL, Ypre = NULL, prob = NULL, HT = FALSE)
```

Arguments

`Ys` list consisting of two N-length numeric vectors labeled `Y0` and `Y1`, as output by `genouts()`

`perms` N-by-r permutation matrix, as output by `genperms` or `genperms.custom`

`X` N-by-k numeric matrix of covariates for regression adjustment

`Ypre` numeric vector of length N, pretreatment measure of the outcome variable for difference estimation `prob` numeric vector within the (0,1) interval of length N, probability of treatment assignment, as output by `genprob()` or `genprobexact()`. When `prob=NULL` (by default), assumes probability of assignment to treatment implied by the permutation matrix

`HT` when `HT=TRUE`, invokes the Horvitz-Thompson (difference-in-totals) estimator. When `HT=FALSE`, invokes the inverse-probability-weighted regression estimator

Value

An r-length vector of estimated ATEs

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

See Also

`estate`, `genouts`, `genprob`, `genperms`, `genperms.custom`

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0,2,2,4,1,1)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0)
cluster <- c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9)
block <- c(rep(1,4),rep(2,6),rep(3,8))

perms <- genperms(Z,blockvar=block, clustvar=cluster) # all possible permutations
probs <- genprobexact(Z,blockvar=block, clustvar=cluster) # probability of treatment
ate <- estate(y,Z,prob=probs) # estimate the ATE

## Conduct Sharp Null Hypothesis Test of Zero Effect for Each Unit

Ys <- genouts(y,Z,ate=0) # generate potential outcomes under sharp null of no effect
distout <- gendist(Ys,perms, prob=probs) # generate sampling dist. under sharp null
dispdist(distout, ate)  # display characteristics of sampling dist. for inference

## Generate Sampling Distribution Around Estimated ATE

Ys <- genouts(y,Z,ate=ate) ## generate potential outcomes under tau = ATE
distout <- gendist(Ys,perms, prob=probs) # generate sampling dist. under tau = ATE
dispdist(distout, ate)  ## display characteristics of sampling dist. for inference
```

genouts *Generates hypothesized potential outcomes under a constant effects hypothesis*

Description

Takes an outcome variable, a treatment assignment, and a hypothesized treatment effect and generates a set of hypothesized potential outcomes

Usage

```
genouts(Y, Z, ate = 0)
```

Arguments

Y	numeric vector of N-length, outcome variable
Z	binary vector (0 or 1) of N-length, treatment indicator
ate	numeric scalar, hypothesized treatment effect

Value

list consisting of two N-length numeric vectors labeled Y0 and Y1

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

See Also

estate

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0,2,2,4,1,1)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0)
cluster <- c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9)
block <- c(rep(1,4),rep(2,6),rep(3,8))

perms <- genperms(Z,blockvar=block, clustvar=cluster) # all possible permutations
probs <- genprobexact(Z,blockvar=block, clustvar=cluster) # probability of treatment
ate <- estate(y,Z,prob=probs) # estimate the ATE

## Conduct Sharp Null Hypothesis Test of Zero Effect for Each Unit
```

```
Ys <- genouts(y,Z,ate=0) # generate potential outcomes under sharp null of no effect
distout <- gendist(Ys,perms, prob=probs) # generate sampling dist. under sharp null
dispdist(distout, ate)  # display characteristics of sampling dist. for inference

## Generate Sampling Distribution Around Estimated ATE

Ys <- genouts(y,Z,ate=ate) ## generate potential outcomes under tau = ATE
distout <- gendist(Ys,perms, prob=probs) # generate sampling dist. under tau = ATE
dispdist(distout, ate)  ## display characteristics of sampling dist. for inference
```

genperms *Generates a permutation matrix for blocked, clustered (or simpler) designs*

Description

Given complete randomization of clusters (even of length 1) in blocks (even of length N), `genperms()` produces either an exact or approximate permutation matrix. When the number of actual permutations exceeds a user specified value (`maxiter`), the function produces an approximate permutations matrix via repeated randomization

Usage

```
genperms(Z, blockvar = NULL, clustvar = NULL, maxiter = 10000)
```

Arguments

`Z`	binary vector (0 or 1) of N-length, treatment indicator
`blockvar`	positive integer vector of N-length, with unique values indicating different blocks
`clustvar`	positive integer vector of N-length, with unique values indicating different clusters
`maxiter`	maximum number of permutations to be included in the permutation matrix

Value

N-by-r permutation matrix, where r is the smaller of maxiter and the true number of permutations

Warning

`genperms` may use large amounts of memory and computational power, and may not be well-suited for large datasets. We recommend starting with `maxiter` set at low values before attempting to create a permutation matrix with a large number of permutations.

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0,2,2,4,1,1)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0)
cluster <- c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9)
block <- c(rep(1,4),rep(2,6),rep(3,8))

perms <- genperms(Z,blockvar=block, clustvar=cluster) # all possible permutations
probs <- genprobexact(Z,blockvar=block, clustvar=cluster) # probability of treatment
ate <- estate(y,Z,prob=probs) # estimate the ATE

## Conduct Sharp Null Hypothesis Test of Zero Effect for Each Unit

Ys <- genouts(y,Z,ate=0) # generate potential outcomes under sharp null of no effect
distout <- gendist(Ys,perms, prob=probs) # generate sampling dist. under sharp null
dispdist(distout, ate)  # display characteristics of sampling dist. for inference

## Generate Sampling Distribution Around Estimated ATE

Ys <- genouts(y,Z,ate=ate) ## generate potential outcomes under tau = ATE
distout <- gendist(Ys,perms, prob=probs) # generate sampling dist. under tau = ATE
dispdist(distout, ate)  ## display characteristics of sampling dist. for inference
```

`genperms.custom` *Generates an approximate permutation matrix for an user-supplied randomization function*

Description

Generates a permutation matrix by replicating a user-supplied randomization function. Not intended to be used for designs handled by `genperms` (i.e., complete randomization of clusters within blocks)

Usage

```
genperms.custom(numiter = 10000, randfun = randfun.default, ...)
```

Arguments

`numiter`	a scalar for the number of replicates, default is 10000
`randfun`	a user supplied function outputting an N-length binary (0 or 1) vector. Default is an internal function.
`...`	other inputs for `randfun`

Value

an N-by-k permutation matrix, where k = numiter

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

See Also

genperms

Examples

```
## Rejected randomization scheme: reject if and only if there is significant imbalance

X <- c(1:200)

randfun <- function() {
  teststat <- -1
while (teststat < 0.05) {
Zri <- sample(c(rep(0,180),rep(1,20))) # imbalanced design
fstat <- summary(lm(Zri~X))$fstatistic
teststat <- pf(fstat[1],fstat[2],fstat[3],lower.tail=FALSE)  # extract F-test p-value
}
return(Zri)
}
perms <- genperms.custom(numiter=10000, randfun=randfun) # generate permutations
probs <- genprob(perms) # generate approximate probabilities from permutation matrix
cor(probs,(X-mean(X))^2) # observations with extreme X are less likely to be treated
```

---

genprob *Estimates probabilities of treatment assignment*

Description

Takes a permutation matrix and estimates the probabilities of treatment assignment for each unit

Usage

```
genprob(perms)
```

Arguments

perms	N-by-k permutation matrix as produced by genperms or genperms.custom.

Details

genprob is NOT intended to be used for complete randomization of clusters within blocks – instead, it takes an arbitrary permutation matrix and computes the proportions of random assignments for which each unit is in treatment. For simpler designs, genpermsexact should be used.

Value

N-length numeric vector of values within the (0,1) interval, probability of treatment assignment

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

See Also

genprobexact

Examples

```
## Rejected randomization scheme: reject if and only if there is significant imbalance

X <- c(1:200)

randfun <- function() {
  teststat <- -1
  while (teststat < 0.05) {
Zri <- sample(c(rep(0,180),rep(1,20))) # imbalanced design
fstat <- summary(lm(Zri~X))$fstatistic
teststat <- pf(fstat[1],fstat[2],fstat[3],lower.tail=FALSE)  # extract F-test p-value
}
return(Zri)
}
perms <- genperms.custom(numiter=10000, randfun=randfun) # generate permutations
probs <- genprob(perms) # generate approximate probabilities from permutation matrix
cor(probs,(X-mean(X))^2) # observations with extreme X are less likely to be treated
```

---

genprobexact *Production of exact probabilities of treatment assignment for blocked, clustered designs*

Description

Function takes a blocking variable and a clustering variable and yields exact probabilities of treatment under complete randomization of clusters within blocks

Usage

```
genprobexact(Z, blockvar = NULL, clustvar = NULL)
```

Arguments

`Z` binary vector (0 or 1) of length N, treatment indicator

`blockvar` positive integer vector of length N, with unique values indicating different blocks

`clustvar` positive integer vector of length N, with unique values indicating different clusters

Value

numeric vector with values within the (0,1) interval of length N, probability of treatment assignment

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

See Also

genprob

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0,2,2,4,1,1)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0)
cluster <- c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9)
block <- c(rep(1,4),rep(2,6),rep(3,8))

probs <- genprobexact(Z,block,cluster) # generate probability of assignment
ate <- estate(y,Z,prob=probs) # estimate the ATE; estimated ATE=2
```

---

invert.ci *Confidence intervals through test inversion*

Description

Experimental code to generate endpoints of Rosenbaum (2002)-style confidence intervals through inversion of a constant effects hypothesis. Only conducts inference with the difference in (weighted) means as the test statistic, no covariate adjustment.

Usage

```
invert.ci(Y, Z, prob, perms, targetp)
```

Arguments

`Y` numeric vector of length N, outcome variable

`Z` binary vector (0 or 1) of length N, treatment indicator

`prob` numeric vector within the (0,1) interval of length N, probability of treatment assignment, as outputted by `genprob()` or `genprobexact()`. When `prob=NULL` (the default), assumes uniform probability of assignment to treatment equal to the mean of Z

`perms` N-by-r permutation matrix, as output by `genperms` or `genperms.custom`

`targetp` target p-value for the endpoint of the confidence interval

Value

returns endpoint of the confidence interval with the target p-value associated

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

Rosenbaum, Paul R. 2002. *Observational Studies*. 2nd ed. New York: Springer.

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0)

perms <- genperms(Z) ## all possible permutations of assignment to treatment
probs <- genprobexact(Z) ## assuming complete randomization

c(invert.ci(y,Z,probs,perms,0.025),invert.ci(y,Z,probs,perms,0.975)) ## 95% CI
```

`omni.ate` *Omnibus function for estimation and testing*

Description

Estimates the average treatment effect (ATE) and inferential statistics under constant effects hypotheses. Estimation is without covariate adjustment, via weighted least squares.

Usage

```
omni.ate(Y, Z, perms, invert = FALSE, quantiles = c(0.025, 0.975))
```

Arguments

`Y` numeric vector of length N, outcome variable

`Z` binary vector (0 or 1) of length N, treatment indicator

`perms` N-by-r permutation matrix, as output by `genperms` or `genperms.custom`

`invert` logical for generating constant effects confidence intervals through exact test inversion, with the difference-in-means as a test statistic. Default is `FALSE`.

`quantiles` vector of quantiles of the randomization distribution to be returned. Quantiles also used to determine endpoints of confidence intervals. Default is equal-tailed 95% intervals.

Details

`omni.ate()` is a convenience function that implements a number of functions otherwise available in `ri`. Greater flexibility through use of the individual functions involved.

Value

`ate` estimated average treatment effect

`greater.p.value`
one-tailed p-value: proportion of randomizations yielding estimated ATE greater than or equal to hypothesized ATE

`lesser.p.value` one-tailed p-value: proportion of randomizations yielding estimated ATE less than or equal to hypothesized ATE

`p.value` two-tailed p-value: twice the smaller of the two one-tailed p-values, as advocated by Rosenbaum (2002)

`p.value.alt` two-tailed p-value: proportion of randomizations yielding absolute estimated ATE greater than or equal to absolute hypothesized ATE

`se.null` standard error of the randomization distribution assuming a zero treatment effect

`conf.int` confidence interval approximation under a constant effect hypothesis

`se` standard error of the randomization distribution assuming a constant treatment effect equal to the estimated ATE

`conf.intInv` (Optional, if `invert=TRUE`) confidence interval under an inverted exact test with the difference-in-means as a test statistic

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

Rosenbaum, Paul R. 2002. *Observational Studies*. 2nd ed. New York: Springer.

Samii, Cyrus and Peter M. Aronow. 2012. On Equivalencies Between Design-Based and Regression-Based Variance Estimators for Randomized Experiments. *Statistics and Probability Letters*. 82(2): 365-370. http://dx.doi.org/10.1016/j.spl.2011.10.024

See Also

ri

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0,2,2)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0,0,1)

perms <- genperms(Z) # all possible permutations of assignment

omni.ate(y,Z,perms,FALSE)
# omni.ate(y,Z,perms,TRUE) # may take some time to run
```

resresplot *Produces residual-residual (added-variable) plot*

Description

Residualizes the outcome variable and the treatment variable with covariates (via inverse probability weighted least squares regression) and plots the relationship. When weights are applied, the graph shows the relative weighting of each observation

Usage

```
resresplot(Y, Z, X, prob = NULL, scale = 1)
```

Arguments

Y	numeric vector of length N, outcome variable
Z	binary vector (0 or 1) of length N, treatment indicator
X	N-by-k numeric matrix of covariates for regression adjustment
prob	numeric vector within the (0,1) interval of length N, probability of treatment assignment, as outputted by `genprob()` or `genprobexact()`. When `prob=NULL` (the default), assumes uniform probability of assignment to treatment equal to the mean of Z
scale	a scalar parameter controlling the size of the plotted points

Value

produces a plot of residualized and weighted values

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Cyrus Samii <cds2083@nyu.edu>

References

Gerber, Alan S. and Donald P. Green. 2012. *Field Experiments: Design, Analysis, and Interpretation*. New York: W.W. Norton.

See Also

estate

Examples

```
y <- c(8,6,2,0,3,1,1,1,2,2,0,1,0,2,2,4,1,1)
Z <- c(1,1,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0)
X <- c(1:18)
cluster <- c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9)
block <- c(rep(1,4),rep(2,6),rep(3,8))

probs <- genprobexact(Z,block,cluster) # generate probability of assignment

resresplot(y,Z,X,prob=probs,scale=3) # produce res-res plot
```

ri-internal *Internal function*

Description

Internal function

Index