

Package ‘ivdesign’

October 13, 2022

Type Package

Title Hypothesis Testing in Cluster-Randomized Encouragement Designs

Version 0.1.0

Author Bo Zhang

Maintainer Bo Zhang <bozhan@wharton.upenn.edu>

Description An implementation of randomization-based hypothesis testing for three different estimands in a cluster-randomized encouragement experiment. The three estimands include (1) testing a cluster-level constant proportional treatment effect (Fisher's sharp null hypothesis), (2) pooled effect ratio, and (3) average cluster effect ratio. To test the third estimand, user needs to install 'Gurobi' ($\geq 9.0.1$) optimizer via its R API. Please refer to <https://www.gurobi.com/documentation/9.0/refman/ins_the_r_package.html>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Depends R (≥ 2.10)

Suggests gurobi, Matrix

Imports stats

NeedsCompilation no

Repository CRAN

Date/Publication 2020-07-14 16:30:02 UTC

R topics documented:

ACER	2
control_clusters	3
double_rank	4
double_rank_CI	6
encouraged_clusters	7

PER	8
PER_CI	10

Index	12
--------------	-----------

ACER	<i>Two-sided test for the average cluster effect ratio estimand</i>
------	---

Description

ACER tests (two-sided) if the average cluster effect ratio (ACER) is equal to lambda.

Usage

```
ACER(
  num_t,
  num_c,
  R_t,
  R_c,
  d_t,
  d_c,
  lambda,
  alpha = 0.05,
  kappa = 0.1,
  gap = 0.05,
  verbose = TRUE
)
```

Arguments

num_t	A length-K vector where K is equal to the number of clusters and the kth entry equal to the number of units in the encouraged cluster of the kth matched pair of two clusters.
num_c	A length-K vector with the kth entry equal to the number of units in the control cluster of the kth matched pair of two clusters.
R_t	A length-K vector with kth entry equal to the sum of unit-level outcomes in the encouraged cluster of the kth matched pair of two clusters.
R_c	A length-K vector with the kth entry equal to the sum of unit-level outcomes in the control cluster of the kth matched pair of two clusters.
d_t	A length-K vector with the kth entry equal to the sum of unit-level treatment received in the encouraged cluster of the kth matched pair of two clusters.
d_c	A length-K vector with the kth entry equal to the sum of unit-level treatment received in the control cluster of the kth matched pair of two clusters.
lambda	The magnitude of the average cluster effect ratio (ACER) to be tested.
alpha	The level of the test.
kappa	Minimum compliance rate.

gap Relative MIP optimality gap.
 verbose If true, the solver output is enabled; otherwise, the solver output is disabled.

Value

A list of three elements: the optimal solution, the optimal objective value, and an indicator of whether or not the test is rejected.

Examples

```
## Not run:
# To run the following example, Gurobi must be installed.

R_t = encouraged_clusters$aggregated_outcome
R_c = control_clusters$aggregated_outcome
d_t = encouraged_clusters$aggregated_treatment
d_c = control_clusters$aggregated_treatment
num_t = encouraged_clusters$number_units
num_c = control_clusters$number_units

# Test at level 0.05 if the ACER is equal
# to 0.2. Assume the minimum compliance rate across
# K clusters is at least 0.2. Set verbose = FALSE
# to suppress the output.
res = ACER(num_t, num_c, R_t, R_c, d_t, d_c,
           lambda = 0.2, alpha = 0.05, kappa = 0.2,
           verbose = FALSE)

# The test is rejected
res$Reject

## End(Not run)
```

control_clusters *100 matched control clusters*

Description

A dataset containing the covariates, aggregated outcome, aggregated treatment received, number of units, and the cluster-level IV of 100 matched control clusters. There is a one-to-one correspondence between this 100 matched control clusters and 100 matched encouraged clusters: the *k*th control cluster is matched to the *k*th encouraged cluster in [encouraged_clusters](#).

Usage

```
control_clusters
```

Format

A data frame with 100 rows and 14 columns:

V1 1st simulated covariate

V2 2nd simulated covariate

V3 3rd simulated covariate

V4 4th simulated covariate

V5 5th simulated covariate

V6 6th simulated covariate

V7 7th simulated covariate

V8 8th simulated covariate

V9 9th simulated covariate

V10 10th simulated covariate

aggregated_outcome Total number of death in the cluster

aggregated_treatment Total number of treatment received

number_units Number of units in each cluster

IV Cluster-level instrumental variable

Source

This is a simulated dataset.

double_rank	<i>Two-sided double-rank test for Fisher's sharp null hypothesis in a cluster-level proportional treatment effect model</i>
-------------	---

Description

double_rank returns the two-sided p-value testing Fisher's sharp null hypothesis in a cluster-level proportional treatment effect model.

Usage

```
double_rank(beta_0, R_t, R_c, d_t, d_c, Z_t, Z_c, psi = NULL)
```

Arguments

beta_0 The magnitude of the proportional treatment effect to be tested.

R_t A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level outcomes in the encouraged cluster of the kth matched pair of two clusters.

R_c	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level outcomes in the control cluster of the kth matched pair of two clusters.
d_t	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level treatment received in the encouraged cluster of the kth matched pair of two clusters.
d_c	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level treatment received in the control cluster of the kth matched pair of two clusters.
Z_t	A length-K vector where K is equal to the number of clusters and the kth entry equal to the encouragement dose, i.e., the magnitude of the instrumental variable, of the encouraged cluster in the kth matched pair of two clusters.
Z_c	A length-K vector where K is equal to the number of clusters and the kth entry equal to the encouragement dose, i.e., the magnitude of the instrumental variable, of the control cluster in the kth matched pair of two clusters.
psi	A function specifying the score used in the test statistic. See Details.

Details

Double-rank test statistics is a flexible family of nonparametric test statistics. Function `psi` is a function that specifies the relationship between d_k , the normalized rank of the absolute treated-minus-control dose difference in the instrumental variable, and q_k , the normalized rank of the absolute treated-minus-control dose difference in the observed outcome. For instance, $\text{psi}(d_k, q_k) = 1$ yields the sign test, $\text{psi}(d_k, q_k) = q_k$ yields the Wilcoxon signed rank test. The default setting, $\text{psi}(d_k, q_k) = d_k * q_k$, yields the dose-weighted signed rank test.

Value

A list of five elements: two-sided p-value, deviate, test statistics, expectation of the test statistic under the null hypothesis, and variance of the test statistic under the null hypothesis.

Examples

```
R_t = encouraged_clusters$aggregated_outcome
R_c = control_clusters$aggregated_outcome
d_t = encouraged_clusters$aggregated_treatment
d_c = control_clusters$aggregated_treatment
Z_t = encouraged_clusters$IV
Z_c = control_clusters$IV

# Test beta = 0 in the proportional treatment effect
# model with the help of the double rank test using
# default psi(d_k, q_k) = d_k * q_k:
res = double_rank(0, R_t, R_c, d_t, d_c, Z_t, Z_c)

# Define a new psi function: psi(d_k, q_k) = q_k
psi_2 <- function(x, y) y
```

```
# Using psi_2 and the double rank test is reduced to the
#Wilcoxon signed rank test.
res_2 = double_rank(0, R_t, R_c, d_t, d_c,
                    Z_t, Z_c, psi = psi_2)
```

double_rank_CI	<i>Construct a two-sided confidence interval for the proportional treatment effect in a cluster-level proportional treatment effect model</i>
----------------	---

Description

double_rank_CI returns the two-sided level-alpha confidence interval of the proportional treatment effect in a cluster-level proportional treatment effect model.

Usage

```
double_rank_CI(
  R_t,
  R_c,
  d_t,
  d_c,
  Z_t,
  Z_c,
  lower,
  upper,
  meshsize = 0.001,
  psi = NULL,
  alpha = 0.05
)
```

Arguments

R_t	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level outcomes in the encouraged cluster of the kth matched pair of two clusters.
R_c	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level outcomes in the control cluster of the kth matched pair of two clusters.
d_t	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level treatment received in the encouraged cluster of the kth matched pair of two clusters.
d_c	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level treatment received in the control cluster of the kth matched pair of two clusters.

Z_t	A length-K vector where K is equal to the number of clusters and the kth entry equal to the encouragement dose, i.e., the magnitude of the instrumental variable, of the encouraged cluster in the kth matched pair of two clusters.
Z_c	A length-K vector where K is equal to the number of clusters and the kth entry equal to the encouragement dose, i.e., the magnitude of the instrumental variable, of the control cluster in the kth matched pair of two clusters.
lower, upper	The lower and upper endpoints of the interval to be searched.
meshsize	The meshsize of the grid search.
psi	A function specifying the score used in the test statistic. See Details of double_rank .
alpha	The level of the confidence interval.

Details

`double_rank_CI` constructs a two-sided level-alpha confidence interval by interting the hypothesis test using a `double_rank` test. Function `double_rank_CI` conducts a grid search with user-specified endpoints and meshsize in order to construct the confidence interval. For more details on the `double_rank` test, see [double_rank](#).

Value

A length-2 vector of two endpoints of the confidence interval.

Examples

```
R_t = encouraged_clusters$aggregated_outcome
R_c = control_clusters$aggregated_outcome
d_t = encouraged_clusters$aggregated_treatment
d_c = control_clusters$aggregated_treatment
Z_t = encouraged_clusters$IV
Z_c = control_clusters$IV

# Construct a level 0.05 CI for the constant proportional
# treatment effect with the help of the double rank test using
# default psi(d_k, q_k) = d_k * q_k. Search from -0.1 to 0.1:
CI = double_rank_CI(R_t, R_c, d_t, d_c, Z_t, Z_c,
                    lower = -0.1, upper = 0.1)
```

Description

A dataset containing the covariates, aggregated outcome, aggregated treatment received, number of units, and the cluster-level IV of 100 matched encouraged clusters. There is a one-to-one correspondence between this 100 matched encouraged clusters and 100 matched control clusters: the k th encouraged cluster is matched to the k th control cluster in `control_clusters`.

Usage

```
encouraged_clusters
```

Format

A data frame with 100 rows and 14 columns:

V1 1st simulated covariate

V2 2nd simulated covariate

V3 3rd simulated covariate

V4 4th simulated covariate

V5 5th simulated covariate

V6 6th simulated covariate

V7 7th simulated covariate

V8 8th simulated covariate

V9 9th simulated covariate

V10 10th simulated covariate

aggregated_outcome Total number of death in the cluster

aggregated_treatment Total number of treatment received

number_units Number of units in each cluster

IV Cluster-level instrumental variable

Source

This is a simulated dataset.

PER

Two-sided test for the pooled effect ratio estimand

Description

PER returns the two-sided p-value testing the pooled effect ratio equal to `lambda_0` in a cluster-randomized encouragement experiment.

Usage

```
PER(lambda_0, R_t, R_c, d_t, d_c, Q = NULL)
```


Arguments

<code>lambda_0</code>	The magnitude of the pooled effect ratio estimand to be tested.
<code>R_t</code>	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level outcomes in the encouraged cluster of the kth matched pair of two clusters.
<code>R_c</code>	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level outcomes in the control cluster of the kth matched pair of two clusters.
<code>d_t</code>	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level treatment received in the encouraged cluster of the kth matched pair of two clusters.
<code>d_c</code>	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level treatment received in the control cluster of the kth matched pair of two clusters.
<code>Q</code>	A K times p design matrix containing the covariate information. See Details.

Details

`Q` is used to construct a regression-assisted variance estimator. `Q` is can in principle be any K times p design matrix such that $p < K$. When `Q` is a column vector of 1's, the variance estimator is the classical sample variance estimator. More generally, `Q` may contain any cluster-level or even unit-level covariate information that are predictive of the encouraged-minus-control difference in the observed aggregated outcomes.

Value

A list of five elements: two-sided p-value, deviate, test statistics, expectation of the test statistic under the null hypothesis, and variance of the test statistic under the null hypothesis.

Examples

```
R_t = encouraged_clusters$aggregated_outcome
R_c = control_clusters$aggregated_outcome
d_t = encouraged_clusters$aggregated_treatment
d_c = control_clusters$aggregated_treatment

# Test the pooled effect ratio estimand lambda = 0 using
# the default sample variance estimator, i.e., setting Q = NULL.
res = PER(0, R_t, R_c, d_t, d_c)

# We may leverage observed covariates from both the encouraged
# and control clusters to construct less conservative variance
# estimator. The variance estimator will be less conservative if
# these covariate predict the treated-minus-control difference
# in the outcome. In this illustrated dataset, V1-V10 are simulated
# white noise; it is not surprising that they do not help
# reduce the variance.
Q = cbind(encouraged_clusters[,1:10], control_clusters[,1:10])
```

```
res_2 = PER(0, R_t, R_c, d_t, d_c, Q)
```

 PER_CI

Construct a two-sided confidence interval for the pooled effect ratio

Description

PER_CI returns the two-sided level-alpha confidence interval of the pooled effect ratio in a cluster-randomized encouragement experiment.

Usage

```
PER_CI(  
  R_t,  
  R_c,  
  d_t,  
  d_c,  
  lower,  
  upper,  
  Q = NULL,  
  meshsize = 0.001,  
  alpha = 0.05  
)
```

Arguments

R_t	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level outcomes in the encouraged cluster of the kth matched pair of two clusters.
R_c	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level outcomes in the control cluster of the kth matched pair of two clusters.
d_t	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level treatment received in the encouraged cluster of the kth matched pair of two clusters.
d_c	A length-K vector where K is equal to the number of clusters and the kth entry equal to the sum of unit-level treatment received in the control cluster of the kth matched pair of two clusters.
lower, upper	The lower and upper endpoints of the interval to be searched.
Q	A K times p design matrix containing the covariate information. See Details of the function PER .
meshsize	The meshsize of the grid search.
alpha	The level of the confidence interval.

Details

PER_CI constructs a two-sided level-alpha confidence interval by interting the corresponding hypothesis test for the pooled effect ratio. See [PER](#) for details on the hypothesis testing. PER_CI conducts a grid search with user-specified endpoints and meshsize in order to construct the confidence interval.

Value

A length-2 vector of two endpoints of the confidence interval.

Examples

```
R_t = encouraged_clusters$aggregated_outcome
R_c = control_clusters$aggregated_outcome
d_t = encouraged_clusters$aggregated_treatment
d_c = control_clusters$aggregated_treatment

# Construct 95% CI for the pooled effect ratio estimand
# using the default sample variance estimator, i.e.,
# setting Q = NULL.
CI = PER_CI(R_t, R_c, d_t, d_c, lower = -0.1, upper = 0.1,
            alpha = 0.05)
```

Index

* datasets

control_clusters, [3](#)

encouraged_clusters, [7](#)

ACER, [2](#)

control_clusters, [3](#), [8](#)

double_rank, [4](#), [7](#)

double_rank_CI, [6](#)

encouraged_clusters, [3](#), [7](#)

PER, [8](#), [10](#), [11](#)

PER_CI, [10](#)