Package 'CRTsampleSearch'

February 4, 2021

Title CRTsampleSearch

Version 0.0

Author Ruoshui Zhai <ruoshui_zhai@brown.edu>

Maintainer Ruoshui Zhai <ruoshui_zhai@brown.edu>

Description Sample Size for Cluster Randomized Trials: A Simulation-Based Search Algorithm

Depends R (>= 3.5.0)

Imports parallel, pscl, plyr, lme4, lmtest

License GPL-3

Encoding UTF-8

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`CRTsearch` *Search for the optimal number of clusters*

Description

Binary search for the optimal number of clusters for a two-arm CRT using simulations

Usage

```
CRTsearch(nrep = 10000, nt, nc, tcRatio = 1, minpower = 0.8,
  alpha = 0.05, increaseSamplingBy = 1, PermutationTest = FALSE,
  Npermutationtest = 100, ...)
```

Arguments

nrep	the number of replications of the simulation procedure for generating the pseudo study data, default is "1000")
nt	the initial number of cluster in the treatment arm to start the search
nc	the initial number of cluster in the control arm to start the search
tcRatio	the ratio of number of clusters in the intervention arm vs that in the control arm, default value is "1" for balanced sample size allocation
minpower	the minimum power, default at 0.8
alpha	type-1 errer, default at 0.05
increaseSamplingBy	the rate to increas the number of replications (nrep) when power gets close to the minpower, default at 1 (not increase)
PermutationTest	indicator for if Fisher's sharp null hypothesis is test and permutation test power is of interest, default at FALSE (No)
Npermutationtest=100	if a permutation test is of interest, the number of replications when estimating the permutation test power

Value

a list of (nt, nc, power) for the optimal design

Examples

```
## Not run:
## distribution of cluster sizes
sim_cluster_size=function(N, ...){
  size=round(100*rnorm(N, 0, 1), 0)
  size[size<=0]=1
  return(size)
}
## distribution of the two potential outcomes
sim_potential_outcomes=function(m,...){
  muibar=rnorm(1, 0, 1)
  Y0=rnorm(m, muibar, 10)
  Y1=Y0 + 1
  re=cbind(Y0, Y1)
  colnames(re)=list("Y0", "Y1")
  return(re)
}

## Test statistics: pooled difference in the two study arms
calc_teststat=function(W, data, ...){
  Y0=data[,"Y0"]
  Y1=data[,"Y1"]
  re=sum(Y1[W==1])/sum(W==1) - sum(Y0[W==0])/sum(W==0)
  return(re)
}
## search for the optimal number of clusters
CRTsearch(nrep=1e4, nt=10, nc=10, FUN_clustersize=sim_cluster_size, FUN_Ys=sim_potential_outcomes, FUN_TestS

## End(Not run)
```

simPower *estimate hypothesis test power for a CRT using simulation*

Description

estimate hypothesis test power for a Cluster Randomization Trail (CRT) given the number of clusters in the treatment arm and the control arm for Neyman's Null hypothesis

Usage

```
simPower(nrep = 10000, nt, nc, alpha = 0.05, FUN_TestStat,
  uppersided = NULL, ...)
```

Arguments

nrep	the number of replications of the simulation procedure for generating the pseudo study data, default is "1000")
nt	the number of cluster in the treatment arm to start the search
nc	the number of cluster in the control arm to start the search
alpha	type-1 errer, default at 0.05
FUN_TestStat	user-defined function for the test statistics, should take in two arguments, W=the treatment/control assignment indicator at the individual level; data=the dataset which includes the two potential outcomes at least
uppersided	if a uppersided test is of interest, default at NULL (two-sided test), FALSE if a lowersided test if of interest
minpower	the minimum power, default at 0.8

Value

a vector (length=1) for the estimated power

Examples

```
## Not run:
## distribution of cluster sizes
sim_cluster_size=function(N, ...){
  size=round(100*rnorm(N, 0, 1), 0)
  size[size<=0]=1
  return(size)
}
## distribution of the two potential outcomes
sim_potential_outcomes=function(m,...){
  muibar=rnorm(1, 0, 1)
  Y0=rnorm(m, muibar, 10)
  Y1=Y0 + 1
  re=cbind(Y0, Y1)
  colnames(re)=list("Y0", "Y1")
  return(re)
}

## Test statistics: pooled difference in the two study arms
```

```
calc_teststat=function(W, data, ...){
  Y0=data[,"Y0"]
  Y1=data[,"Y1"]
  re=sum(Y1[W==1])/sum(W==1) - sum(Y0[W==0])/sum(W==0)
  return(re)
}
## search for the optimal number of clusters
simPower(nrep=1e4, nt=10, nc=10, FUN_clustersize=sim_cluster_size, FUN_Ys=sim_potential_outcomes, FUN_TestSt

## End(Not run)
```

`simPowerPT` *estimate hypothesis test power for a CRT using simulation*

Description

estimate hypothesis test power for a Cluster Randomization Trail (CRT) given the number of clusters in the treatment arm and the control arm for the Fisher's shrap null hypothesis and permutation test

Usage

```
simPowerPT(nrep = 10000, nt, nc, alpha = 0.05, FUN_TestStat,
  uppersided = NULL, Npermutationtest = 100, ...)
```

Arguments

`nrep`	the number of replications of the simulation procedure for generating the pseudo study data, default is `"1000"`)
`nt`	the number of cluster in the treatment arm to start the search
`nc`	the number of cluster in the control arm to start the search
`alpha`	type-1 errer, default at 0.05
`FUN_TestStat`	user-defined function for the test statistics, should take in two arguments, W=the treatment/control assignment indicator at the individual level; data=the dataset which includes the two potential outcomes at least #' @param Npermutationtest the number of replications when estimating the permutation test power
`uppersided`	if a uppersided test is of interest, default at NULL (two-sided test), FALSE if a lowersided test if of interest
`minpower`	the minimum power, default at 0.8

Value

a vector (length=1) for the estimated power

Examples

```
## Not run:
## distribution of cluster sizes
sim_cluster_size=function(N, ...){
  size=round(100*rnorm(N, 0, 1), 0)
  size[size<=0]=1
  return(size)
}
## distribution of the two potential outcomes
sim_potential_outcomes=function(m,...){
  muibar=rnorm(1, 0, 1)
  Y0=rnorm(m, muibar, 10)
  Y1=Y0 + 1
  re=cbind(Y0, Y1)
  colnames(re)=list("Y0", "Y1")
  return(re)
}

## Test statistics: pooled difference in the two study arms
calc_teststat=function(W, data, ...){
  Y0=data[,"Y0"]
  Y1=data[,"Y1"]
  re=sum(Y1[W==1])/sum(W==1) - sum(Y0[W==0])/sum(W==0)
  return(re)
}
## search for the optimal number of clusters
simPowerPT(nrep=1e4, nt=10, nc=10, FUN_clustersize=sim_cluster_size, FUN_Ys=sim_potential_outcomes, FUN_Test

## End(Not run)
```

`simulate_CRT` *simulate a smaple CRT study data*

Description

data generating mechanism

Usage

```
simulate_CRT(nt, nc, replacement = TRUE, FUN_clustersize = NULL,
  FUN_Ys = NULL, dataset = NULL, outcome = NULL, clusterID = NULL,
  ...)
```

Arguments

`nt`	the number of cluster in the treatment arm to start the search
`nc`	the number of cluster in the control arm to start the search
`FUN_clustersize`	user-defined function for the distribution of cluster sizes, should take in only one argument: n=the number of clsuters in the study
`FUN_Ys`	user-defined function for the distribution of the two potential outcomes, should take in only one argument: m=the number of individual units in the a clsuter

dataset	a dataset from which to generate the study data (no need to provide FUN_clustersize or FUN_Ys), should be individual level data, including columns for clutser ID, outcome of interest, and the variable for stratified randomization if desired.
outcome	the name of the outcome variable in the dataset
clusterID	the name of the cluster ID in the dataset
nrep	the number of replications of the simulation procedure for generating the pseudo study data, default is "1000")

Value

a dataframe

Examples

```
## Not run:
## distribution of cluster sizes
sim_cluster_size=function(N, ...){
  size=round(100*rnorm(N, 0, 1), 0)
  size[size<=0]=1
  return(size)
}
## distribution of the two potential outcomes
sim_potential_outcomes=function(m,...){
  muibar=rnorm(1, 0, 1)
  Y0=rnorm(m, muibar, 10)
  Y1=Y0 + 1
  re=cbind(Y0, Y1)
  colnames(re)=list("Y0", "Y1")
  return(re)
}

## simulate a CRT study
simulate_CRT(nt=10, nc=10, FUN_clustersize=sim_cluster_size, FUN_Ys=sim_potential_outcomes, ...)

## End(Not run)
```

Index