Package: osDesign (via r-universe)

November 3, 2026

Type Package

Title Design, Planning and Analysis of Observational Studies

Version 1.8

Date 2020-12-11

Author Sebastien Haneuse, Takumi Saegusa, Nilanjan Chaterjee, Norman Breslow

Maintainer Sebastien Haneuse <shaneuse@hsph.harvard.edu>

Description A suite of functions for the design of case-control and two-phase studies, and the analysis of data that arise from them. Functions in this packages provides Monte Carlo based evaluation of operating characteristics such as powers for estimators of the components of a logistic regression model. For additional detail see: Haneuse S, Saegusa T and Lumley T (2011)<doi:10.18637/jss.v043.i11>.

Depends R (>= 3.5.0)

License GPL (>= 3)

LazyLoad yes

NeedsCompilation no

Repository https://cranhaven.r-universe.dev

Date/Publication 2026-11-03 06:02:00 UTC

RemoteUrl https://github.com/cranhaven/cranhaven.r-universe.dev

RemoteRef package/osDesign

RemoteSha bdb5bb4281a534339498b1aea221a0bac83ecb4a

RemoteSubdir osDesign

Contents

beta0 2
ccPower 3
ccSim 6

infants 10
Ohio 11
phaseI 12
plotPower 14
rmvhyper 16
tps 17
tpsPower 21
tpsSim 24

Index **31**

`beta0` *Calculate the intercept of a logistic regression model, given a vector of log-odds ratio parameters and an overall prevalence.*

Description

When conducting power calculations, one is often interested in examining power for various 'effect sizes'. Suppose the logistic regression is specified via the vector of coefficients (`beta0`, `betaX`); the first element is the intercept and the second consists of a vector of log odds ratio parameters. In many settings, the overall outcome prevalence in the population of interest is known or, at least, fixed. Modifying any given element of `betaX` will automatically modify the overall prevalence, unless there is a corresponding change in `beta0`. The function `beta0()` calculates the value of `beta0` that minimizes the difference between the target outcome prevalence, `rhoY`, and prevalence induced by the model in conjuction with the assumed marginal exposure distribution.

Usage

```
beta0(betaX, X, N, rhoY, expandX="all")
```

Arguments

`betaX`	Numeric vector of log-odds ratio parameters for the logistic regression model.
`X`	Design matrix for the logistic regression model. The first column should correspond to intercept. For each exposure, the baseline group should be coded as 0, the first level as 1, and so on.
`N`	A numeric vector providing the sample size for each row of the design matrix, `X`.
`rhoY`	Target outcome prevalence in the population.
`expandX`	Character vector indicating which columns of `X` to expand as a series of dummy variables. Useful when at least one exposure is continuous (and should not be expanded). Default is 'all'; the other option is 'none' or character vector of column names.

Details

The minimization is performed using the `optimize` function.

Value

Numeric value of the intercept parameter in a logistic regression model.

Author(s)

Sebastien Haneuse, Takumi Saegusa

References

Haneuse, S. and Saegusa, T. and Lumley, T. (2011) "osDesign: An R Package for the Analysis, Evaluation, and Design of Two-Phase and Case-Control Studies." Journal of Statistical Software, 43(11), 1-29.

Examples

```
##
data(Ohio)

##
XM   <- cbind(Int=1, Ohio[,1:3])
fitM <- glm(cbind(Death, N-Death) ~ factor(Age) + Sex + Race, data=Ohio,
            family=binomial)

## Overall prevalence in the observed data
##
sum(Ohio$Death)/sum(Ohio$N)

## Intercept corresponding to the original vector of log-odds ratios
##
fitM$coef
beta0(betaX=fitM$coef[-1], X=XM, N=Ohio$N, rhoY=sum(Ohio$Death)/sum(Ohio$N))

## Reduction of Sex effect by 50%
##
betaXm    <- fitM$coef[-1]
betaXm[3] <- betaXm[3] * 0.5
beta0(betaX=betaXm, X=XM, N=Ohio$N, rhoY=sum(Ohio$Death)/sum(Ohio$N))

## Doubling of Race effect
##
betaXm    <- fitM$coef[-1]
betaXm[4] <- betaXm[4] * 2
beta0(betaX=betaXm, X=XM, N=Ohio$N, rhoY=sum(Ohio$Death)/sum(Ohio$N))
```

`ccPower` *Simulation-based estimation of power for the case-control design*

Description

Monte Carlo based estimation of statistical power for maximum likelihood estimator (MLE) of the components of a logistic regression model, based on the case-control design.

Usage

```
ccPower(B=1000, betaTruth, X, N, expandX="all", etaTerms=NULL,
        nCC, r=1, alpha=0.05,
        digits=1, betaNames=NULL, monitor=NULL)
```

Arguments

B	The number of datasets generated by the simulation.
betaTruth	Regression coefficients from the logistic regression model.
X	Design matrix for the logistic regression model. The first column should correspond to intercept. For each exposure, the baseline group should be coded as 0, the first level as 1, and so on.
N	A numeric vector providing the sample size for each row of the design matrix, X.
expandX	Character vector indicating which columns of X to expand as a series of dummy variables. Useful when at least one exposure is continuous (and should not be expanded). Default is 'all'; other options include 'none' or character vector of column names. See Details, below.
etaTerms	Character vector indicating which columns of X are to be included in the model. See Details, below.
nCC	A numeric value indicating the total case-control sample size. If a vector is provided, separate simulations are run for each value.
r	A numeric value indicating the control:case ratio in the case-control sample.
alpha	Type I error rate assumed for the evaluation of coverage probabilities and power.
digits	Integer indicating the precision to be used for the output.
betaNames	An optional character vector of names for the regression coefficients, betaTruth.
monitor	Numeric value indicating how often ccPower() reports real-time progress on the simulation, as the B datasets are generated and evaluated. The default of NULL indicates no output.

Details

A simulation study is conducted to evaluate statistical power for the MLE of a logistic regression model, based on the case-control design. The overall simulation approach is the same as that described in ccSim. Power is estimated as the proportion of simulated datasets for which a hypothesis test of no effect is rejected. Each hypothesis test is performed using the generic glm function.

The correspondence between betaTruth and X, specifically the ordering of elements, is based on successive use of factor to each column of X which is expanded via the expandX argument. Each exposure that is expanded must conform to a 0, 1, 2, ... integer-based coding convention.

The etaTerms argument is useful when only certain columns in X are to be included in the model.

A balanced case-control design is specified by setting r=1; setting r=2 indicates twice as many controls are sampled, relative to the number cases, from the total nCC.

When evaluating operating characteristics of the MLE, some simulated datasets may result in unusually large or small estimates. Particularly, when the the case-control sample size, nCC, is small. In some settings, it may be desirable to truncate the Monte Carlo sampling distribution prior to evaluating operating characteristics. The threshold argument indicates the interval beyond which MLEs are ignored. The default is such that all B datasets are kept.

Value

ccPower() returns an object of class "ccPower", a list containing all the input arguments, as well as the following components:

betaPower	Power against the null hypothesis that the regression coefficient is zero for a Wald-based test with an alpha type I error rate.
failed	A vector consisting of the number of datasets excluded from the power calculations (i.e. set to NA), for each simulation performed. For power calculations, the two reasons are: (1) lack of convergence indicated by NA point estimates returned by glm, (2) lack of convergence indicated by NA standard error point estimates returned by glm.

Note

A generic print method provides formatted output of the results.

A generic plot function plotPower provides plots of powers against different sample sizes for each estimate of a regression coefficient.

Author(s)

Sebastien Haneuse, Takumi Saegusa

References

Prentice, R. and Pyke, R. (1979) "Logistic disease incidence models and case-control studies." Biometrika 66:403-411.

Haneuse, S. and Saegusa, T. and Lumley, T. (2011) "osDesign: An R Package for the Analysis, Evaluation, and Design of Two-Phase and Case-Control Studies." Journal of Statistical Software, 43(11), 1-29.

Examples

```
##
data(Ohio)

##
XM   <- cbind(Int=1, Ohio[,1:3])
fitM <- glm(cbind(Death, N-Death) ~ factor(Age) + Sex + Race, data=Ohio,
            family=binomial)
betaNamesM <- c("Int", "Age1", "Age2", "Sex", "Race")
```

```
## Power for a single CC design
##
ccResult1 <- ccPower(B=100, betaTruth=fitM$coef, X=XM, N=Ohio$N, r=1,
                     nCC=500, betaNames=betaNamesM, monitor=10)
ccResult1

## Power for the CC design, based on a balanced design with
## various sample sizes
##

ccResult2 <- ccPower(B=1000, betaTruth=fitM$coef, X=XM, N=Ohio$N, r=1,
                     nCC=seq(from=100, to=500, by=50),
                     betaNames=betaNamesM, monitor=100)
ccResult2

## Recalculate power for the setting where the age coefficients are
## halved from their observed true values
##   * the intercept is modified, accordingly, using the beta0() function
##
newBetaM      <- fitM$coef
newBetaM[2:3] <- newBetaM[2:3] / 2
newBetaM[1]   <- beta0(betaX=newBetaM[-1], X=XM, N=Ohio$N,
rhoY=sum(Ohio$Death)/sum(Ohio$N))
##

ccResult3 <- ccPower(B=1000, betaTruth=newBetaM, X=XM, N=Ohio$N,
                     r=1, nCC=seq(from=100, to=500, by=50),
                     betaNames=betaNamesM,
                     monitor=100)
ccResult3
```

---

ccSim *Simulation function for case-control study designs.*

Description

Monte Carlo based evaluation of operating characteristics of the maximum likelihood estimator (MLE) for the coefficients of a logistic regression model, based on the case-control.

Usage

```
ccSim(B=1000, betaTruth, X, N,  expandX="all", etaTerms=NULL,
      nCC, r, refDesign=1, alpha=0.05,
      threshold=c(-Inf, Inf), digits=1, betaNames=NULL,
      monitor=NULL, returnRaw=FALSE)
```

Arguments

B	The number of datasets generated by the simulation.
betaTruth	Regression coefficients from the logistic regression model.
X	Design matrix for the logistic regression model. The first column should correspond to intercept. For each exposure, the baseline group should be coded as 0, the first level as 1, and so on.
N	A numeric vector providing the sample size for each row of the design matrix, X.
expandX	Character vector indicating which columns of X to expand as a series of dummy variables. Useful when at least one exposure is continuous (and should not be expanded). Default is 'all'; other options include 'none' or character vector of column names. See Details, below.
etaTerms	Character vector indicating which columns of X are to be included in the model. See Details, below.
nCC	A numeric value indicating the total case-control sample size.
r	A numeric value indicating the control:case ratio in the case-control sample. If a vector is provided, separate simulations are run for each value.
refDesign	A numeric value indicating the control:case ratio for the referent design (for the relative uncertainty calculation).
alpha	Type I error rate assumed for the evaluation of coverage probabilities and power.
threshold	An interval that specifies truncation of the Monte Carlo sampling distribution of the MLE.
digits	Integer indicating the precision to be used for the output.
betaNames	An optional character vector of names for the regression coefficients, betaTruth.
monitor	Numeric value indicating how often ccSim() reports real-time progress on the simulation, as the B datasets are generated and evaluated. The default of NULL indicates no output.
returnRaw	Logical indicator of whether or not the raw coefficient and standard error estimates for each of the design/estimator combinations should be returned.

Details

A simulation study is performed to evaluate the operating characteristics of the MLE for betaTruth from a case-control design (Prentice and Pyke, 1979). The operating characteristics are evaluated using the Monte Carlo sampling distribution of the estimator. The latter is generated using the following steps:

- (i) Specify the (joint) marginal exposure distribution of underlying population, using X and N.
- (ii) Simulate outcomes for all sum(N) individuals in the population, based on an underlying logistic regression model specified via betaTruth.
- (iii) Sample n0 controls and n1 cases, on the basis of nCC and r.
- (iv) Evaluate the MLE estimator, its estimated standard error and store the results.
- (v) Repeat steps (ii)-(iv) B times.

All case-control MLEs are evaluated using the generic glm function.

The correspondence between betaTruth and X, specifically the ordering of elements, is based on successive use of factor to each column of X which is expanded via the expandX argument. Each exposure that is expanded must conform to a 0, 1, 2, ... integer-based coding convention.

The etaTerms argument is useful when only certain columns in X are to be included in the model.

When evaluating operating characteristics of the MLE, some simulated datasets may result in unusually large or small estimates. Particularly, when the the case-control sample size, nCC, is small. In some settings, it may be desirable to truncate the Monte Carlo sampling distribution prior to evaluating operating characteristics. The threshold argument indicates the interval beyond which MLEs are ignored. The default is such that all B datasets are kept.

Value

ccSim() returns an object of class "ccSim", a list containing all the input arguments, as well list results with the following components:

betaMean	Mean of the Monte Carlo sampling distribution for each regression coefficient estimator.
betaMeanBias	Bias based on the mean, calculated as betaMean - betaTruth.
betaMeanPB	Percent bias based on mean, calculated as ((betaMean - betaTruth) / betaTruth) x 100. If a regression coefficient is zero, percent bias is not calculated and an NA is returned.
betaMedian	Median of the Monte Carlo sampling distribution for each regression coefficient estimator.
betaMedianBias	Bias based on the median, calculated as betaMedian - betaTruth.
betaMedianPB	Percent bias based on median, calculated as ((betaMedian - betaTruth) / betaTruth) x 100. If a regression coefficient is zero, median percent bias is not calculated and an NA is returned.
betaSD	Standard deviation of the Monte Carlo sampling distribution for each regression coefficient estimator.
betaMSE	Mean squared error of the Monte Carlo sampling distribution for each regression coefficient estimator.
seMean	Mean of the Monte Carlo sampling distribution for the standard error estimates reported by glm().
seRatio	Ratio of the mean reported standard error to the standard deviation of the Monte Carlo sampling distribution for each regression coefficient estimator. The ratio is multiplied by 100.
betaCP	Coverage probability for Wald-based confidence intervals, evaluated on the basis of an alpha type I error rate.
betaPower	Power against the null hypothesis that the regression coefficient is zero for a Wald-based test with an alpha type I error rate.
betaRU	The ratio of the standard deviation of the Monte Carlo sampling distribution for each estimator to the standard deviation of the Monte Carlo sampling distribution for the estimator corresponding to refDesign. The ratio is multiplied by 100.

Also returned is an object `failed` which is a vector consisting of the number of datasets excluded from the power calculations (i.e. set to `NA`), for each simulation performed. For the evaluation of general operating characteristics, the three reasons are: (1) lack of convergence indicated by `NA` point estimates returned by `glm`, (2) lack of convergence indicated by `NA` standard error point estimates returned by `glm`, (3) exclusion on the basis of the `threshold` argument.

Note

A generic print method provides formatted output of the results.

Author(s)

Sebastien Haneuse, Takumi Saegusa

References

Prentice, R. and Pyke, R. (1979) "Logistic disease incidence models and case-control studies." Biometrika 66:403-411.

Haneuse, S. and Saegusa, T. and Lumley, T. (2011) "osDesign: An R Package for the Analysis, Evaluation, and Design of Two-Phase and Case-Control Studies." Journal of Statistical Software, 43(11), 1-29.

See Also

`plotPower`.

Examples

```
##
data(Ohio)

##
XM   <- cbind(Int=1, Ohio[,1:3])
fitM <- glm(cbind(Death, N-Death) ~ factor(Age) + Sex + Race, data=Ohio,
            family=binomial)
betaNamesM <- c("Int", "Age1", "Age2", "Sex", "Race")

## Single case-control design
##
ccResults1 <- ccSim(B=100, betaTruth=fitM$coef, X=XM, N=Ohio$N,
                    nCC=500, r=1, betaNames=betaNamesM, monitor=10)
ccResults1

## Examining unbalanced case-control designs
##

ccResults2 <- ccSim(B=1000, betaTruth=fitM$coef, X=XM, N=Ohio$N,
                    nCC=500, r=c(0.25, 0.33, 0.5, 1, 2, 3, 4),
                    betaNames=betaNamesM, monitor=100)
ccResults2
```

infants *Infant mortality data from North Carolina*

Description

Individual-level infant mortality data on 235,272 births in the U.S. state of North Carolina, in 2003 and 2004.

Usage

```
data(infants)
```

Format

A data frame consisting of 235,464 observations, with the following columns:

`year` Year of birth; either 2003 or 2004.

`race` A 9-level categorical variable indicating the race of the baby. See Details, below.

`male` A binary variable; 0=female; 1=male.

`mage` Age of the mother, years.

`weeks` Number of completed weeks of gestation.

`cignum` Average number of cigarettes. A value of '98' indicates smoking but unknown amount.

`gained` Weight gained during pregnancy, lbs.

`weight` Birth weight, grams.

`death` A binary variable indicating death within 1st year of life; 0=alive; 1=death.

Details

The data were compiled by the North Carolina State Center for Health Statistics (http://www.irss.unc.edu/).

The `race` variable is coded as follows: 0 = Other non-white 1 = White 2 = Black 3 = American Indian 4 = Chinese 5 = Japanese 6 = Hawaiin 7 = Filipino 8 = Other Asian or Pacific Islander

Examples

```
## Code to generate an aggregated dataset
##
data(infants)
##
infants$smoker  <- as.numeric(infants$cignum > 0)
infants$teen    <- as.numeric(infants$mage < 20)
infants$lowgain <- as.numeric(infants$gained < 20)
infants$early   <- as.numeric(infants$weeks < 32)
infants$lbw     <- as.numeric(infants$weight < 2500)
##
listAgg <- list(year=infants$year,
                smoker=infants$smoker,
```

```
                  teen=infants$teen,
                  lowgain=infants$lowgain,
                  race=infants$race,
                  male=infants$male,
                  early=infants$early,
                  lbw=infants$lbw)
infantsAgg <- aggregate(rep(1, nrow(infants)), listAgg, FUN=sum)
names(infantsAgg)[ncol(infantsAgg)] <- "N"
infantsAgg$Y <- aggregate(infants$death, listAgg, FUN=sum)$x
```

`Ohio` *Ohio lung cancer data*

Description

Population estimates and lung cancer death counts for the state of Ohio in 1988, among 55-84 year olds. Counts are stratified by age, sex and race.

Usage

```
data(Ohio)
```

Format

A data frame consisting of 12 observations, with the following columns:

`Age` A 3-level categorical level variable; 0=55-64 years; 1=65-74 years; 2=75-84 years.

`Sex` A binary variable; 0=male; 1=female.

`Race` A binary variable; 0=white; 1=non-white.

`N` A numeric vector of estimated population counts.

`Death` A numeric vector of lung cancer death counts.

Details

The data were obtained from the National Center for Heatlh Statistics Compressed Mortality File and correspond to a population of 2,220,177 individuals with 5,533 lung cancer deaths. A more comprehensive dataset, providing counts further stratified by county as well as for the years 1968 to 1988, is described by Xia and Carlin (1998).

Source

Xia H, Carlin B (1998). Spatio-temporal models with errors in covariates: Mapping Ohio lung cancer mortality. Statistics in Medicine, 17, 2025-2043.

Haneuse, S. and Saegusa, T. and Lumley, T. (2011) "osDesign: An R Package for the Analysis, Evaluation, and Design of Two-Phase and Case-Control Studies." Journal of Statistical Software, 43(11), 1-29.

phaseI *Expected phase I stratification*

Description

phaseI() provides the expected phase I counts, based on a pre-specified population and outcome model. If phase II sample sizes are provided, the (expected) phase II sampling probabilities are also reported.

Usage

```
phaseI(betaTruth, X, N, strata=NULL, expandX="all", etaTerms=NULL,
      nII0=NULL, nII1=NULL,
      cohort=TRUE, NI=NULL, digits=NULL)
```

Arguments

betaTruth	Regression coefficients from the logistic regression model.
X	Design matrix for the logistic regression model. The first column should correspond to intercept. For each exposure, the baseline group should be coded as 0, the first level as 1, and so on.
N	A numeric vector providing the sample size for each row of the design matrix, X.
strata	A numeric vector indicating which columns of the design matrix, X, are used to form the phase I stratification variable. strata=1 specifies the intercept and is, therefore, equivalent to a case-control study.
expandX	Character vector indicating which columns of X to expand as a series of dummy variables. Useful when at least one exposure is continuous (and should not be expanded). Default is 'all'; other options include 'none' or character vector of column names. See Details, below.
etaTerms	Character vector indicating which columns of X are to be included in the model. See Details, below.
nII0	A vector of sample sizes at phase II for controls. The length must correspond to the number of unique values for phase I stratification variable.
nII1	A vector of sample sizes at phase II for cases. The length must correspond to the number of unique values phase I stratification variable.
cohort	Logical flag. TRUE indicates phase I is drawn as a cohort; FALSE indicates phase I is drawn as a case-control sample.
NI	A pair of integers providing the outcome-specific phase I sample sizes when the phase I data are drawn as a case-control sample. The first element corresponds to the controls and the second to the cases.
digits	Integer indicating the precision to be used for the reporting of the (expected) sampling probabilities

Details

The correspondence between betaTruth and X, specifically the ordering of elements, is based on successive use of factor to each column of X which is expanded via the expandX argument. Each exposure that is expanded must conform to a 0, 1, 2, ... integer-based coding convention.

The etaTerms argument is useful when only certain columns in X are to be included in the model. In the context of the two-phase design, this might be the case if phase I stratifies on some surrogate exposure and a more detailed/accurate measure is to be included in the main model.

Value

phaseI() returns an object of class "phaseI" that, at a minimum includes:

phaseI	Expected phase I counts, based on a pre-specified population and outcome model
covm	Model based variance-covariance matrix. This is available for method = "PL" and "ML".
cove	Empirical variance-covariance matrix. This is available for all the three methods.
fail	Indicator of whether or not the phase I and/or the phase II constraints are satisfied; only relevant for the ML estimator.

If phase II sample sizes were provided as part of the initial call, the object additionally includes:

phaseII	The supplied phase II sample sizes.
phaseIIprobs	Expected phase II sampling probabilities.

Author(s)

Sebastien Haneuse, Takumi Saegusa

References

Haneuse, S. and Saegusa, T. and Lumley, T. (2011) "osDesign: An R Package for the Analysis, Evaluation, and Design of Two-Phase and Case-Control Studies." Journal of Statistical Software, 43(11), 1-29.

Examples

```
##
data(Ohio)

## Design matrix that forms the basis for model and phase I
## stata specification
##
XM <- cbind(Int=1, Ohio[,1:3])      ## main effects only
XI <- cbind(XM, SbyR=XM[,3]*XM[,4]) ## interaction between sex and race

## 'True' values for the underlying logistic model
##
fitM <- glm(cbind(Death, N-Death) ~ factor(Age) + Sex + Race, data=Ohio,
```

```
            family=binomial)
fitI <- glm(cbind(Death, N-Death) ~ factor(Age) + Sex * Race, data=Ohio,
            family=binomial)

## Stratified sampling by race
##
phaseI(betaTruth=fitM$coef, X=XM, N=Ohio$N, strata=4,
       nII0=c(125, 125),
       nII1=c(125, 125))

## Stratified sampling by age and sex
##
phaseI(betaTruth=fitM$coef, X=XM, N=Ohio$N, strata=c(2,3))
##
phaseI(betaTruth=fitM$coef, X=XM, N=Ohio$N, strata=c(2,3),
       nII0=(30+1:6),
       nII1=(40+1:6))
```

---

plotPower *Plot function for power, based on two-phase and case-control design*

Description

The plotPower function plots estimates of power obtained from objects returned by either the tpsPower or ccPower functions.

Usage

```
plotPower(x, coefNum=1, include="All", yAxis=seq(from=0, to=100, by=20),
          xAxis=NULL, main=NULL, legendXY=NULL)
```

Arguments

x	An object in a class tpsPower or ccPower obtained as a result of tpsPower or ccPower functions, respectively.
coefNum	A numeric vector number specifying the regression coefficient in beta for the plot.
include	Character string indicating which estimators from a tpsPower object are to be printed. The default is "All" in which case all four estimators (two-phase WL, PL, ML and case-control CC) are presented. Other options include "TPS" which solely presents the three two-phase estimators; options "WL", "PL", "ML" and "CC" solely present the corresponding estimators. If the object is of class ccPower then only the case-control MLE (CC) is presented (i.e., the include argument is ignored).
yAxis	A scale marking the y-axis for the plot.
xAxis	A scale marking the x-axis for the plot. If left as the default NULL, the x-axis scale is taken from nII in the tpsResults object.

main	Title for the plot.
legendXY	Optional vector indicating the co-ordinates for the top-left hand corner of the legend box.

Details

Produces a plot of statistical power (to reject a null hypothesis H0: beta = 0), for estimators of a regression coefficient from a logistic regression model, based on a two-phase and/or case-control design.

Author(s)

Sebastien Haneuse, Takumi Saegusa

References

Haneuse, S. and Saegusa, T. and Lumley, T. (2011) "osDesign: An R Package for the Analysis, Evaluation, and Design of Two-Phase and Case-Control Studies." Journal of Statistical Software, 43(11), 1-29.

See Also

tpsPower.

Examples

```
##
data(Ohio)

##
XM   <- cbind(Int=1, Ohio[,1:3])
fitM <- glm(cbind(Death, N-Death) ~ factor(Age) + Sex + Race, data=Ohio,
            family=binomial)
betaNamesM <- c("Int", "Age1", "Age2", "Sex", "Race")

## Power for the TPS design where phase I stratification is based on Age
##
newBetaM      <- fitM$coef
newBetaM[2:3] <- newBetaM[2:3] / 2
##

powerRaceTPS <- tpsPower(B=10000, betaTruth=fitM$coef, X=XM, N=Ohio$N,
                    strata=4,
                    nII=seq(from=100, to=1000, by=100),
   betaNames=c("Int", "Age1", "Age2", "Sex", "Race"), monitor=1000)
##
old.par <- par()
par(mfrow=c(2,2))
plotPower(powerRaceTPS, include="TPS", coefNum=2,
          xAxis=seq(from=100, to=1000, by=100),
          main=expression("Age effect (65-74 vs. 55-64 years), " * beta[A1]),
```

```
          legendXY=c(800, 65))
plotPower(powerRaceTPS, include="ML", coefNum=2,
          xAxis=seq(from=100, to=1000, by=100),
          main=expression("Age effect (65-74 vs. 55-64 years), " * beta[A1]),
          legendXY=c(800, 65))
plotPower(powerRaceTPS, include="WL", coefNum=2,
          xAxis=seq(from=100, to=1000, by=100),
          main=expression("Age effect (65-74 vs. 55-64 years), " * beta[A1]),
          legendXY=c(800, 65))
plotPower(powerRaceTPS, include="CC", coefNum=2,
          xAxis=seq(from=100, to=1000, by=100),
          main=expression("Age effect (65-74 vs. 55-64 years), " * beta[A1]),
          legendXY=c(800, 65))
##
par(old.par)

## Power
##

ccResult <- ccPower(B=1000, betaTruth=newBetaM, X=XM, N=Ohio$N, r=0.5,
                    nCC=seq(from=100, to=500, by=50), betaNames=betaNamesM,
                    monitor=100)
##
par(mfrow=c(2,2))
plotPower(ccResult, coefNum=2, yAxis=seq(from=0, to=100, by=20),
            xAxis=seq(from=100, to=500, by=100),
            main=expression("Age effect (65-74 vs. 55-64 years), " * beta[A1]))
plotPower(ccResult, coefNum=3, yAxis=seq(from=0, to=100, by=20),
            xAxis=seq(from=100, to=500, by=100),
            main=expression("Age effect (75-84 vs. 55-64 years), " * beta[A2]))
plotPower(ccResult, coefNum=4, yAxis=seq(from=0, to=100, by=20),
            xAxis=seq(from=100, to=500, by=100),
            main=expression("Sex effect, " * beta[S]))
plotPower(ccResult, coefNum=5, yAxis=seq(from=0, to=100, by=20),
            xAxis=seq(from=100, to=500, by=100),
            main=expression("Race effect, " * beta[R]))
```

`rmvhyper` *Random generation for the multivariate hypergeometric distribution*

Description

Generates a single random deviate from a multivariate hypergeometric distribution.

Usage

```
rmvhyper(Mk, m)
```

Arguments

Mk	A numeric vector describing the population from which the sub-sample will be drawn.
m	Number of elements to be drawn from the population

Details

The multivariate hypergeometric distribution is for sampling without replacement from a population with a finite number of element types. The number of element types is given by the length of the vector Mk.

Value

A numeric vector of elements, totally to m, drawn without replacement from the population described by Mk.

Author(s)

Sebastien Haneuse

See Also

rhyper.

Examples

```
##
rmvhyper(c(1000, 500, 200, 50), 200)

## Check the properties (first two moments) of the generated deviates
##
M  <- 100
Qx <- c(0.7, 0.15, 0.1, 0.05)
temp <- matrix(NA, nrow=10000, ncol=length(Qx))
for(i in 1:nrow(temp)) temp[i,] <- rmvhyper(M*Qx, 1)

##
rbind(Qx, apply(temp, 2,  mean))
rbind(sqrt(Qx * (1-Qx)), apply(temp, 2, sd))
```

---

tps *Estimation for two-phase designs.*

Description

Fits a logistic regression model to data arising from two phase designs

Usage

```
tps(formula=formula(data), data=parent.frame(), nn0, nn1, group,
    contrasts=NULL, method="PL", cohort=TRUE, alpha=1)
```

Arguments

formula	A formula expression as for other binomial response regression models, of the form response ~ predictors, where both the response and predictors corresponds to observations at phase II sample. The response can be either a vector of 0's and 1's or else a matrix with two columns representing number of cases (response=1) and controls (response=0) corresponding to the rows of the design matrix.
data	An optional data frame for phase two sample in which to interpret the variables occurring in the formula.
nn0	A numeric vector of length K, indicating the numbers of controls for each Phase I strata.
nn1	A numeric vector of length K, indicating the numbers of cases for each Phase I strata.
group	A numeric vector providing stratum identification for phase II data. Values should be in {1,...,K}, where K is the number of strata (vector of same length as the response and predictors). A vector indicating a stratum for each row of the design matrix.
contrasts	A list of contrasts to be used for some or all of the factors appearing as variables in the model formula. See the documentation of glm for more details.
method	Three different procedures are available. The default method is "PL" which implements pseudo-likelihood as developed by Breslow and Cain (1988). Other possible choices are "WL" and "ML" which implements, respectively, weighted likelihood (Flanders and Greenland, 1991; Zhao and Lipsitz, 1992) and maximum likelihood (Breslow and Holubkov, 1997; Scott and Wild, 1997).
cohort	Logical flag. TRUE indicates phase I is drawn as a cohort; FALSE indicates phase I is drawn as a case-control sample.
alpha	Marginal odds of observing a case in the population. This is only used when cohort=F is specified and must be correctly specified in order to obtain a correct estimate of the intercept.

Details

Returns estimates and standard errors from logistic regression fit to data arising from two phase designs. Three semiparametric methods are implemented to obtain estimates of the regression coefficients and their standard errors. Use of this function requires existence of a finite number of strata (K) so that the phase one data consist of a joint classification into 2K cells according to binary outcome and stratum. This function can also handle certain missing value and measurement error problems with validation data.

The phase I sample can involve either cohort or case-control sampling. This software yields correct estimates (and standard errors) of all the regression coefficients (including the intercept) under cohort sampling at phase I. When phase I involves case-control sampling one cannot estimate the

intercept, except, when the marginal odds of observing a case in the population is specified. Then the software yields a correct estimate and standard error for the intercept also.

The WL method fits a logistic regression model to the phase II data with a set of weights. Each unit is weighted by the ratio of frequencies (phase I/phase II) for the corresponding outcome X stratum cell. This estimator has its origins in sampling theory and is well known as Horvitz-Thompson method. The PL method maximizes the product of conditional probabilities of "being a case" given the covariates and the fact of inclusion in the phase II sample. This is called the "complete data likelihood" by some researchers. The estimate is obtained by fitting a logistic regression model to the phase II data with a set of offsets. The ML procedure maximizes the full likelihood of the data (phase I and II) jointly with respect to the regression parameters and the marginal distribution of the covariates. The resulting concentrated score equations (Breslow and Holubkov (1997) , eq. 18) were solved using a modified Newton-Raphson algorithm. Schill's (1993) partial likelihood estimates are used as the starting values.

NOTE: In some settings, the current implementation of the ML estimator returns point estimates that do not satisfy the phase I and/or phase II constraints. If this is the case a warning is printed and the "fail" elements of the returned list is set to TRUE. An example of this is phenomenon is given below. When this occurs, users are encouraged to either report the PL estimator or consider using Chris Wild's 'missreg' package.

Value

`tps()` returns a list that includes estimated regression coefficients and one or two estimates of their asymptotic variance-covariance matrix:

`coef`	Regression coefficient estimates
`covm`	Model based variance-covariance matrix. This is available for `method = "PL"` and "ML".
`cove`	Empirical variance-covariance matrix. This is available for all the three methods.
`fail`	Indicator of whether or not the phase I and/or the phase II constraints are satisfied; only relevant for the ML estimator.

Author(s)

Nilanjan Chaterjee, Norman Breslow, Sebastien Haneuse

References

Flanders W. and Greenland S. (1991) "Analytic methods for two-stage case-control studies and other stratified designs." Statistics in Medicine 10:739-747.

Zhao L. and Lipsitz S. (1992) "Design and analysis of two-stage studies."Statistics in Medicine 11:769-782.

Schill, W., Jockel K-H., Drescher, K. and Timm, J.(1993). "Logistic analysis in case-control studies under validation sampling." Biometrika 80:339-352.

Scott, A. and Wild, C. (1997) "Fitting regression models to case control data by maximum likelihood." Biometrika 78:705-717.

Breslow, N. and Holubkov, R. (1997) "Maximum likelihood estimation for logistic regression parameters under two-phase, outcome dependent sampling." J. Roy. Statist. Soc. B. 59:447-461.

Breslow, N. and Cain, K. (1988) "Logistic regression for two-stage case control data." Biometrika 75:11-20.

Breslow, N. and Chatterjee, N. (1999) "Design and analysis of two phase studies with binary outcome applied to Wilms tumour prognosis." Applied Statistics 48:457-468.

Haneuse, S. and Saegusa, T. and Lumley, T. (2011) "osDesign: An R Package for the Analysis, Evaluation, and Design of Two-Phase and Case-Control Studies." Journal of Statistical Software, 43(11), 1-29.

Examples

```
##
data(Ohio)

## Phase I stratification based on age
##
Ohio$S <- Ohio$Age + 1
K <- length(unique(Ohio$S))

## Phase I data
##
Ohio$nonDeath <- Ohio$N-Ohio$Death
nn0 <- aggregate(Ohio$nonDeath, list(S=Ohio$S), FUN=sum)$x
nn1 <- aggregate(Ohio$Death, list(S=Ohio$S), FUN=sum)$x

## Phase II sample sizes
##
nPhIIconts <- rep(100, 3)
nPhIIcases <- rep(100, 3)

## 'Generate' phase II data
##
Ohio$conts <- NA
Ohio$cases <- NA
for(k in 1:K)
{
  Ohio$conts[Ohio$S == k] <- rmvhyper(Ohio$nonDeath[Ohio$S == k],
                                      nPhIIconts[k])
  Ohio$cases[Ohio$S == k] <- rmvhyper(Ohio$Death[Ohio$S == k],
                                      nPhIIcases[k])
}

## Three estimators
##
tps(cbind(cases, conts) ~ factor(Age) + Sex + Race, data=Ohio,
    nn0=nn0, nn1=nn1,
    group=Ohio$S, method="WL")
tps(cbind(cases, conts) ~ factor(Age) + Sex + Race, data=Ohio,
    nn0=nn0, nn1=nn1,
    group=Ohio$S, method="PL")
```

```
tps(cbind(cases, conts) ~ factor(Age) + Sex + Race, data=Ohio,
    nn0=nn0, nn1=nn1,
    group=Ohio$S, method="ML")

## An example where (most of the time) the constraints are not satisfied and a warning is returned
##
tps(cbind(cases, conts) ~ Sex + Race, data=Ohio,
    nn0=nn0, nn1=nn1,
    group=Ohio$S, method="ML")
```

`tpsPower` *Simulation-based estimation of power for the two-phase study design*

Description

Monte Carlo based estimation of statistical power for estimators of the components of a logistic regression model, based on balanced two-phase and case-control study designs (Breslow and Chatterjee, 1999; Prentice and Pykle, 1979).

Usage

```
tpsPower(B=1000, betaTruth, X, N, strata, expandX="all", etaTerms=NULL,
         nII, alpha=0.05, digits=1, betaNames=NULL,
         monitor=NULL, cohort=TRUE, NI=NULL)
```

Arguments

`B`	The number of datasets generated by the simulation.
`betaTruth`	Regression coefficients from the logistic regression model.
`X`	Design matrix for the logistic regression model. The first column should correspond to intercept. For each exposure, the baseline group should be coded as 0, the first level as 1, and so on.
`N`	A numeric vector providing the sample size for each row of the design matrix, `X`.
`strata`	A numeric vector indicating which columns of the design matrix, `X`, are used to form the phase I stratification variable. `strata=1` specifies the intercept and is, therefore, equivalent to a case-control study. `strata=0` is not permitted in tpsPower(), although multiple two-phase stratifications can be investigated with tpsSim().
`expandX`	Character vector indicating which columns of `X` to expand as a series of dummy variables. Useful when at least one exposure is continuous (and should not be expanded). Default is 'all'; other options include 'none' or character vector of column names. See Details, below.
`etaTerms`	Character vector indicating which columns of `X` are to be included in the model. See Details, below.

nII	A numeric value indicating the phase II sample size. If a vector is provided, separate simulations are run for each element.
alpha	Type I error rate assumed for the evaluation of coverage probabilities and power.
digits	Integer indicating the precision to be used for the output.
betaNames	An optional character vector of names for the regression coefficients, betaTruth.
monitor	Numeric value indicating how often tpsPower() reports real-time progress on the simulation, as the B datasets are generated and evaluated. The default of NULL indicates no output.
cohort	Logical flag. TRUE indicates phase I is drawn as a cohort; FALSE indicates phase I is drawn as a case-control sample.
NI	A pair of integers providing the outcome-specific phase I sample sizes when the phase I data are drawn as a case-control sample. The first element corresponds to the controls and the second to the cases.

Details

A simulation study is performed to estimate power for various estimators of beta:

- (a) complete data maximum likelihood (CD)
- (b) case-control maximum likelihood (CC)
- (c) two-phase weighted likelihood (WL)
- (d) two-phase pseudo- or profile likelihood (PL)
- (e) two-phase maximum likelihood (ML)

The overall simulation approach is the same as that described in tpsSim.

In each case, power is estimated as the proportion of simulated datasets for which a hypothesis test of no effect is rejected.

The correspondence between betaTruth and X, specifically the ordering of elements, is based on successive use of factor to each column of X which is expanded via the expandX argument. Each exposure that is expanded must conform to a 0, 1, 2, ... integer-based coding convention.

The etaTerms argument is useful when only certain columns in X are to be included in the model. In the context of the two-phase design, this might be the case if phase I stratifies on some surrogate exposure and a more detailed/accurate measure is to be included in the main model.

Only balanced designs are considered by tpsPower(). For unbalanced designs, power estimates can be obtained from tpsSim.

NOTE: In some settings, the current implementation of the ML estimator returns point estimates that do not satisfy the phase I and/or phase II constraints. If this is the case a warning is printed and the "fail" elements of the returned list is set to TRUE. An example of this is phenomenon is given the help file for tps. When this occurs, tpsPower() considers ML estimation for the particular dataset to have failed.

Value

`tpsPower()` returns an object of class 'tpsPower', a list containing all the input arguments, as well as the following components:

`betaPower` Power against the null hypothesis that the regression coefficient is zero for a Wald-based test with an `alpha` type I error rate.

`failed` A vector consisting of the number of datasets excluded from the power calculations (i.e. set to `NA`), for each simulation performed. For power calculations, the three reasons are: (1) lack of convergence indicated by `NA` point estimates returned by `glm` or `tps`; (2) lack of convergence indicated by `NA` standard error point estimates returned by `glm` or `tps`; and (3) for the ML estimator only, the phase I and/or phase II constraints are not satisfied.

Note

A generic print method provides formatted output of the results.

A generic plot function `plotPower` provides plots of powers against different sample sizes for each estimate of a regression coefficient.

Author(s)

Sebastien Haneuse, Takumi Saegusa

References

Prentice, R. and Pyke, R. (1979) "Logistic disease incidence models and case-control studies." Biometrika 66:403-411.

Breslow, N. and Chatterjee, N. (1999) "Design and analysis of two phase studies with binary outcome applied to Wilms tumour prognosis." Applied Statistics 48:457-468.

Haneuse, S. and Saegusa, T. and Lumley, T. (2011) "osDesign: An R Package for the Analysis, Evaluation, and Design of Two-Phase and Case-Control Studies." Journal of Statistical Software, 43(11), 1-29.

See Also

`plotPower`.

Examples

```
##
data(Ohio)

##
XM   <- cbind(Int=1, Ohio[,1:3])
fitM <- glm(cbind(Death, N-Death) ~ factor(Age) + Sex + Race, data=Ohio,
            family=binomial)
betaNamesM <- c("Int", "Age1", "Age2", "Sex", "Race")

## Power for the TPS design where phase I stratification is based on Race.
```

```
##

tpsResult1 <- tpsPower(B=1000, beta=fitM$coef, X=XM, N=Ohio$N, strata=4,
                       nII=seq(from=100, to=1000, by=100),
                       betaNames=betaNamesM, monitor=100)
tpsResult1

## Power for the TPS design where phase I stratification is based on Age
##   * consider the setting where the age coefficients are halved from
##     their observed true values
##   * the intercept is modified, accordingly, using the beta0() function
##
newBetaM      <- fitM$coef
newBetaM[2:3] <- newBetaM[2:3] / 2
newBetaM[1]   <- beta0(betaX=newBetaM[-1], X=XM, N=Ohio$N,
                       rhoY=sum(Ohio$Death)/sum(Ohio$N))
##

tpsResult2 <- tpsPower(B=1000, beta=fitM$coef, X=XM, N=Ohio$N, strata=2,
                       nII=seq(from=100, to=500, by=50),
                       betaNames=betaNamesM, monitor=100)
tpsResult2
```

`tpsSim` *Simulation function for two-phase study designs.*

Description

Monte Carlo based evaluation of operating characteristics for estimators of the components of a logistic regression model, based on the two-phase and case-control study designs (Breslow and Chatterjee, 1999; Prentice and Pykle, 1979).

Usage

```
tpsSim(B=1000, betaTruth, X, N, strata, expandX="all", etaTerms=NULL,
       nII0=NULL, nII1=NULL, nII=NULL, nCC=NULL,
       alpha=0.05, threshold=c(-Inf, Inf), digits=1, betaNames=NULL,
       referent=2, monitor=NULL, cohort=TRUE, NI=NULL, returnRaw=FALSE)
```

Arguments

`B`	The number of datasets generated by the simulation.
`betaTruth`	Regression coefficients from the logistic regression model.
`X`	Design matrix for the logistic regression model. The first column should correspond to intercept. For each exposure, the baseline group should be coded as 0, the first level as 1, and so on.
`N`	A numeric vector providing the sample size for each row of the design matrix, `X`.

strata	A list of numeric vectors indicating which columns of the design matrix, X, are used to form the phase I stratification schemes. strata=1 specifies the intercept and is, therefore, equivalent to a case-control study. strata=0 indicates all possible stratified two-phase sampling schemes. strata=list(2,3) indicates 2 two-phase designs (that stratify on the 2nd and 3rd columns, separately) are to be considered.
expandX	Character vector indicating which columns of X to expand as a series of dummy variables. Useful when at least one exposure is continuous (and should not be expanded). Default is 'all'; other options include 'none' or character vector of column names. See Details, below.
etaTerms	Character vector indicating which columns of X are to be included in the model. See Details, below.
nII0	A numeric vector of sample sizes at phase II for controls. The length must correspond to the number of unique values for the strata specification.
nII1	A numeric vector of sample sizes at phase II for cases. The length must correspond to the number of unique values for the strata specification.
nII	A pair of numbers providing the sample sizes for controls and cases at phase II. This is only used when simulating all stratified two-phase sampling schemes (i.e., strata=0).
nCC	A pair of sample sizes at phase II for controls and cases in a case-control design. If left NULL, the values case-control sample sizes are taken as the sums of n1 and n0, respectively.
alpha	Type I error rate assumed for the evaluation of coverage probabilities and power.
threshold	An interval that specifies truncation of the Monte Carlo sampling distribution of each estimator.
digits	Integer indicating the precision to be used for the output.
betaNames	An optional character vector of names for the regression coefficients, betaTruth.
referent	An numeric value specifying which estimator is taken as the referent (denominator) for the relative uncertainty calculation. 1=CD, 2=CC, 3=WL, 4=PL, 5=ML (see Details below).
monitor	Numeric value indicating how often tpsSim() reports real-time progress on the simulation, as the B datasets are generated and evaluated. The default of NULL indicates no output.
cohort	Logical flag. TRUE indicates phase I is drawn as a cohort; FALSE indicates phase I is drawn as a case-control sample.
NI	A pair of integers providing the outcome-specific phase I sample sizes when the phase I data are drawn as a case-control sample. The first element corresponds to the controls and the second to the cases.
returnRaw	Logical indicator of whether or not the raw coefficient and standard error estimates for each of the design/estimator combinations should be returned.

Details

A simulation study is performed to evaluate the operating characteristics of various designs/estimators for betaTruth:

- (a) complete data maximum likelihood (CD)
- (b) case-control maximum likelihood (CC)
- (c) two-phase weighted likelihood (WL)
- (d) two-phase pseudo- or profile likelihood (PL)
- (e) two-phase maximum likelihood (ML)

The operating characteristics are evaluated using the Monte Carlo sampling distribution of each estimator. The latter is generated using the following steps:

- (i) Specify the (joint) marginal exposure distribution of underlying population, using `X` and `N`.
- (ii) Simulate outcomes for all sum(`N`) individuals in the population, based on an underlying logistic regression model specified via `betaTruth`.
- (iii) Evaluate the CD estimator on the basis of the complete data.
- (iv) Sample either (a) `ccDesign` controls and cases or (b) sum(`n0`) controls and sum(`n1`) cases, (without regard to the `strata` variable) and evaluate the CC estimator.
- (v) Stratify the population according to outcome and the `strata` argument, to form the phase I data.
- (vi) Sample `n0` controls and `n1` cases from their respective phase I strata.
- (vii) Evaluate the WL, PL and ML estimators.
- (viii) Repeat steps (ii)-(vii) `B` times.

Both the CD and CC estimators are evaluated using the generic `glm` function. The three two-phase estimators are based on the `tps` function.

The correspondence between `betaTruth` and `X`, specifically the ordering of elements, is based on successive use of `factor` to each column of `X` which is expanded via the `expandX` argument. Each exposure that is expanded must conform to a 0, 1, 2, ... integer-based coding convention.

The `etaTerms` argument is useful when only certain columns in `X` are to be included in the model. In the context of the two-phase design, this might be the case if phase I stratifies on some surrogate exposure and a more detailed/accurate measure is to be included in the main model.

When evaluating operating characteristics, some simulated datasets may result in unusually large or small estimates. Particularly, when the the case-control/phase II sample sizes are small. In some settings, it may be desirable to truncate the Monte Carlo sampling distribution prior to evaluating operating characteristics. The `threshold` argument indicates the interval beyond which point estimates are ignored. The default is such that all `B` datasets are kept.

NOTE: In some settings, the current implementation of the ML estimator returns point estimates that do not satisfy the phase I and/or phase II constraints. If this is the case a warning is printed and the 'fail' elements of the returned list is set to TRUE. An example of this is phenomenon is given the help file for `tps`. When this occurs, `tpsSim()` considers ML estimation for the particular dataset to have failed.

Value

`tpsSim()` returns an object of class 'tpsSim', a list containing all the input arguments, as well list `results` with the following components:

betaMean	Mean of the Monte Carlo sampling distribution for each regression coefficient estimator.
betaMeanBias	Bias based on the mean, calculated as betaMean - betaTruth.
betaMeanPB	Percent bias based on mean, calculated as ((betaMean - betaTruth) / betaTruth) x 100. If a regression coefficient is zero, percent bias is not calculated and an NA is returned.
betaMedian	Median of the Monte Carlo sampling distribution for each regression coefficient estimator.
betaMedianBias	Bias based on the median, calculated as betaMedian - betaTruth.
betaMedianPB	Percent bias based on median, calculated as ((betaMedian - betaTruth) / betaTruth) x 100. If a regression coefficient is zero, median percent bias is not calculated and an NA is returned.
betaSD	Standard deviation of the Monte Carlo sampling distribution for each regression coefficient estimator.
betaMSE	Mean squared error of the Monte Carlo sampling distribution for each regression coefficient estimator.
seMean	Mean of the Monte Carlo sampling distribution for the standard error estimates reported by glm().
seRatio	Ratio of the mean reported standard error to the standard deviation of the Monte Carlo sampling distribution for each regression coefficient estimator. The ratio is multiplied by 100.
betaCP	Coverage probability for Wald-based confidence intervals, evaluated on the basis of an alpha type I error rate.
betaPower	Power against the null hypothesis that the regression coefficient is zero for a Wald-based test with an alpha type I error rate.
betaRU	The ratio of the standard deviation of the Monte Carlo sampling distribution for each estimator to the standard deviation of the Monte Carlo sampling distribution for the estimator corresponding to refDesign. The ratio is multiplied by 100.

Also returned is an object failed which is a vector consisting of the number of datasets excluded from the power calculations (i.e. set to NA), for each simulation performed. For the evaluation of general operating characteristics, the four reasons are: (1) lack of convergence indicated by NA point estimates returned by glm or tps; (2) lack of convergence indicated by NA standard error point estimates returned by glm or tps; (3) exclusion on the basis of the threshold argument; and (4) for the ML estimator only, the phase I and/or phase II constraints are not satisfied.

Note

A generic print method provides formatted output of the results.

Author(s)

Sebastien Haneuse, Takumi Saegusa

References

Prentice, R. and Pyke, R. (1979) "Logistic disease incidence models and case-control studies." Biometrika 66:403-411.

Breslow, N. and Chatterjee, N. (1999) "Design and analysis of two phase studies with binary outcome applied to Wilms tumour prognosis." Applied Statistics 48:457-468.

Haneuse, S. and Saegusa, T. and Lumley, T. (2011) "osDesign: An R Package for the Analysis, Evaluation, and Design of Two-Phase and Case-Control Studies." Journal of Statistical Software, 43(11), 1-29.

Examples

```
##
data(Ohio)

## Design matrix that forms the basis for model and
## phase I strata specification
##
XM <- cbind(Int=1, Ohio[,1:3])      ## main effects only
XI <- cbind(XM, SbyR=XM[,3]*XM[,4]) ## interaction between sex and race

## 'True' values for the underlying logistic model
##
fitM <- glm(cbind(Death, N-Death) ~ factor(Age) + Sex + Race, data=Ohio,
            family=binomial)
fitI <- glm(cbind(Death, N-Death) ~ factor(Age) + Sex * Race, data=Ohio,
            family=binomial)
##
betaNamesM <- c("Int", "Age1", "Age2", "Sex", "Race")
betaNamesI <- c("Int", "Age1", "Age2", "Sex", "Race", "SexRace")

## Two-phase design stratified by age
##   * sample 50 from each of 6 phase I strata
##   * show primary output (% bias, 95% CP, relative uncertainty)
##

ocAge <- tpsSim(B=1000, betaTruth=fitM$coef, X=XM, N=Ohio$N, strata=2,
                nII0=c(50,50,50), nII1=c(50,50,50), betaNames=betaNamesM,
                monitor=100)
ocAge

## All possible balanced two-phase designs
##  * 250 controls and 250 cases
##  * only show the relative uncertainty output
##

ocAll <- tpsSim(B=1000, betaTruth=fitM$coef, X=XM, N=Ohio$N, strata=0,
                nII=c(250, 250), betaNames=betaNamesM,
                monitor=100)
ocAll

## Two-phase design stratified by race
```

```
##  * balanced solely on outcome
##  * only show the relative uncertainty output
##

ocRace <- tpsSim(B=1000, betaTruth=fitI$coef, X=XI, N=Ohio$N, strata=4,
                 nII0=c(200, 50), nII1=c(200, 50), betaNames=betaNamesI,
                 monitor=100)
ocRace

## Comparison of two case-control designs
##  * 240 controls and 260 cases
##  * 240 controls and 260 cases
##  * only show the relative uncertainty output
##

ocCC <- tpsSim(B=1000, betaTruth=fitM$coef, X=XM, N=Ohio$N, strata=1,
               nII0=240, nII1= 260, nCC=c(200,300),
               betaNames=betaNamesM,
               monitor=100)
ocCC

## Illustration of setting where one of the covariates is continuous
##  * restrict to black and white children born in 2003
##  * dichotomize smoking, mothers age, weight gain during pregnancy and weight weight
##  * note the use of 'etaTerms' to restrict to specific variables (the majority of which
##    are created)
##  * note the use of 'strata=list(11,12)' to simultaneously investigate stratification by
##     - 11th column in XM: derived 'smoker' variable
##     - 12th column in XM: derived 'teen' variable
##
## Warning: takes a long time!
##

data(infants)
##
infants <- infants[infants$year == 2003,]
##
infants$race[!is.element(infants$race, c(1,2))] <- NA  ## White/Black = 0/1
infants$race <- infants$race - 1
infants <- na.omit(infants)
##
infants$smoker  <- as.numeric(infants$cignum > 0)
infants$teen    <- as.numeric(infants$mage < 20)
infants$lowgain <- as.numeric(infants$gained < 20)
infants$lbw     <- as.numeric(infants$weight < 2500)
infants$weeks   <- (infants$weeks - 36) / 4       ## estimate a 4-week contrast
##
fitM  <- glm(death ~ smoker + teen + race + male+ lowgain + lbw + weeks,
             data=infants,
             family=binomial)
betaM <- fitM$coef
XM    <- cbind(Int=1, infants)
etaM  <- c("Int", "smoker", "teen", "race", "male", "lowgain", "lbw", "weeks")
```

```
##
tpsSim(B=1000, betaTruth=fitM$coef, X=XM, N=rep(1, nrow(XM)), strata=list(11,12),
       expand="none", etaTerms=etaM, nII=c(1000,1000),
       threshold=c(-20,20),
       monitor=100)
```

Index

∗ **datasets**
infants, 10
Ohio, 11

beta0, 2

ccPower, 3, *14*
ccSim, *4*, 6

factor, *4*, *8*, *13*, *22*, *26*

glm, *4, 5*, *8, 9*, *18*, *23*, *26, 27*

infants, 10

Ohio, 11
optimize, *2*

phaseI, 12
plotPower, *5*, *9*, 14, *23*

rhyper, *17*
rmvhyper, 16

tps, 17, *22, 23*, *26, 27*
tpsPower, *14, 15*, 21
tpsSim, *22*, 24