Package

February 17, 2019

Title Calculate Power and Sample Size with BetaPSS

Version 0.0.1

Description Calculate power and sample size(incomplete).

Depends R (>= 3.5.1)

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

Imports betareg,
lmtest,
reshape,
ggplot2,
ggpubr,
Rcpp

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

betapower *Find Power with Beta DBN*

Description

Find the power for a given sample size when testing the null hypothesis that the means for the control and treatment groups are equal against a two-sided alternative.

Usage

```
betapower(mu0, sd0, mu1.start, mu1.end, mu1.by,
ss.start, ss.end, ss.by, trials, seed, link.type="logit")
```

Arguments

`mu0`	the mean for the control group
`sd0`	the standard deviation for the control group
`mu1.start`	the starting value of mean for the treatment group under the alternative mu1
`mu1.end`	the ending value of mean for the treatment group under the alternative mu1
`mu1.by`	the step length of mean for the treatment group under the alternative mu1
`ss.start`	the starting value of sample size
`ss.end`	the ending value of sample size
`ss.by`	the step length of sample size
`trials`	the number of trials
`seed`	the seed used in the simulation
`link.type`	the type of link used in the beta regression. Default value is "logit", or you can choose one or more of the following: "logit", "probit", "cloglog", "cauchit", "log", "loglog", "all"

Details

betapower function allows you to control the number of trials in the simulation, the sample sizes used, and the alternative means. You can fix the alternative and vary sample size to match a desired power; You can fix the sample size and vary the alternative to see which will match a desired power; You can vary both; Start with a small number of trials (say 100) to determine the rough range of sample sizes or alternatives; Use a larger number of trials (say 1000) to get better estimates.

Value

Return a matrix with 7 to 12 columns:

`power.of.GLM: link name`	the power using regression method; it will return the power with every links if you use link.type = "all" statement.
`power.of.Wilcoxon.test`	the power from Wilcoxon Rank sum test.
`sample size`	sample size.
`mu1`	the mean for the treatment group under the alternative.

`mu0`	the mean for the control group.
`sd0`	the standard deviation for the control group.
`trials`	the number of trials.

Examples

```
betapower(0.56,0.255,.70,.75,.05,30,50, 20,40,610201501)
betapower(0.56,0.255,.60,.75,.05,30,50, 5,100,617201501,"all")
betapower(0.56,0.255,.70,.75,.05,30,50, 20,40,610201501,c("logit","loglog","log"))
```

`betapower_plot` *Plots of Beta power*

Description

Generate several comparison plots of power.

Usage

```
betapower_plot(betapower_matrix,link.type)
```

Arguments

`link.type`	the type of link used in the beta regression. You can choose one or more of the following: "logit", "probit", "cloglog", "cauchit", "log", "loglog", "all"
`plot.type`	the type of plot. see details.
`betapower_matrix`	a matrix obtained by the function betapower.(the formula was described as the output formula in the function betapower)

Details

betapower_plot() returns different plots depends on plot.type
plot.type = 1: betapower_plot() returns graphs that plot power against mu1, where mu1 is the mean for the treatment group under the alternative. The number of plots will vary depending on the number of link types selected with the last plot showing power based on Wilcoxon Rank Sum Test. The first one or several plots show comparisons of power with different sample size, using GLM method with one or several link types. The last plot shows a comparison of the power with different sample size using Wilcoxon Rank Sum Test. Y-axis denotes power and X-axis denotes mu1, the mean for the treatment group under the alternative.
plot.type = 2: betapower_plot() returns a number of plots equal to the number of sample sizes tested. Each plot compares power calculated with different link types and the Wilcoxon Rank Sum Test. Y-axis denotes power and X-axis denotes mu1, the mean for the treatment group under the alternative.
plot.type = 3: betapower_plot() returns a number of plots equal to the number of mu1 used in the procedure. Each plot compares power calculated with different link types and the Wilcoxon Rank Sum Test. Y-axis denotes power and X-axis denotes sample size.

Examples

```
BPmat <- betapower(0.56,0.255,.70,.75,.05,30,50, 20,40,610201501,"all")
betapower_plot(BPmat,link.type = "all",plot.type=1)
betapower_plot(BPmat,link.type = "all",plot.type=2)
betapower_plot(BPmat,link.type = "all",plot.type=3)
BPmat2 <- betapower(0.56,0.255,.560,.76,.05,30,45, 5,200,610201511,c("logit","loglog","log"))
betapower_plot(BPmat2,link.type = c("logit","loglog","log"),plot.type=1)
betapower_plot(BPmat2,link.type = c("logit","loglog","log"),plot.type=2)
betapower_plot(BPmat2,link.type = c("logit","loglog","log"),plot.type=3)
```

LFL *Check linearity assumption*

Description

Check linearity assumption in a model using a given link function.

Usage

```
LFL(dsn,Y,X,nkat,link.type)
```

Arguments

dsn	data set name containing X and Y;
Y	outcome variable
X	continuous covariate
nkat	number of categories into which X should be divided (an even number)
link.type	link function to be used: link.type is coded as follows: "identity" = identity link; "logit" = logit link; "probit" = probit link; "log" = log link; "clog-log" = complementary log-log link; "log-log" = log-log link; "reciprocal" = reciprocal link;
outlier	exclude outliers from the plot.

Value

Return a matrix with 6 columns:

X	intervals of continuous covariate.
Frequency	number of observations in each group.
mean	group means of outcome variable.
link	name of link function.
gmu	group means of outcome variable with link function.
midpt	median of continuous covariate in each group.

Examples

```
Data.X <- runif(1000)
Data.Y <- exp(Data.X+1)+rnorm(1000,0,1)
Data <- data.frame(cbind(Data.X,Data.Y))
LFL(Data,"Data.Y","Data.X",20,4)
##Use NH data:
NH$quality_rating <- (scale(NH$quality_rating)/2+0.5)
LFL(dsn = NH,Y = "quality_rating", X = "RNHRD", nkat = 40, link.type = "logit", outlier = T)
```

NH *Nursing homes data on Nursing Home Compare site.*

Description

These are the official datasets used on the Medicare.gov Nursing Home Compare Website provided by the Centers for Medicare & Medicaid Services. These data allow you to compare the quality of care at every Medicare and Medicaid-certified nursing home in the country, including over 15,000 nationwide. This data set only used 3 variables of original data.

Usage

```
data(NH)
```

Format

An object of data frame.

Source

Nursing homes data

Examples

```
data(NH)
summary(NH)
```

---

sample.size2 *Find minimum sample size with Beta DBN*

Description

Find minimum sample sizes with Beta DBN and given mu0,sd0,mu1 and target powers.

Usage

```
sample.size(mu0, sd0, mu1.start, mu1.end, mu1.by, power.start, power.end, power.by,
sig.level = 0.05, N = 100, accuracy = NuLL, seed = 1, link.type = "logit")
```

Arguments

`mu0`	the mean for the control group
`sd0`	the standard deviation for the control group
`mu1.start`	the starting value of mean for the treatment group under the alternative mu1
`mu1.end`	the ending value of mean for the treatment group under the alternative mu1
`mu1.by`	the step length of mean for the treatment group under the alternative mu1
`power.start`	the starting value of target power
`power.end`	the ending value of target power
`power.by`	the step length of target power
`sig.level`	significant level; default value is 0.05
`N`	the number of trials; default value is 100
`accuracy`	the accuracy of the result; must be integer
`seed`	the seed used in the simulation
`link.type`	link options include: "logit", "probit", "cloglog", "cauchit", "log", "loglog". Default link is "logit".

Details

The sample.size function allows you to control the number of trials in the simulation, the target power, accuracy, and the alternative means. You can fix the alternative and vary power to match a desired sample size; Use default values for the number of trials and accuracy for a quick view; Use a larger number of trials (say 1000) and a smaller accuracy (say 1) to get better estimates.

Examples

```
sample.size2(mu0=0.56, sd0=0.255, mu1.start = 0.60, mu1.end = 0.70, mu1.by = 0.05,
power.start = 0.7, power.end = 0.9, power.by = 0.1)
sample.size2(mu0=0.56, sd0=0.255, mu1.start = 0.60, mu1.end = 0.70, mu1.by = 0.05,
power.start = 0.7, power.end = 0.9, power.by = 0.1, link.type = c("logit","loglog","log"))
sample.size2(mu0=0.56, sd0=0.255, mu1.start = 0.60, mu1.end = 0.70, mu1.by = 0.05,
power.start = 0.7, power.end = 0.9, power.by = 0.1, link.type = "all")
```

`samplesize_plot` *Plots by mu1*

Description

Generate the comparison plots using GLM method and Wilcoxon Rank Sum Test with different mu1.

Usage

```
samplesize_plot(SS_matrix,link.type)
```

Arguments

`link.type`	the type of link used in the beta regression(or Wilcoxon Rank Sum Test). You can choose one or more of the following: "logit", "probit", "cloglog", "cauchit", "log", "loglog", "wilcoxon", "all"
`SS_matrix`	the matrix obtained by the function sample.size2.(the formula was described as the output formula in the function sample.size2)

Details

samplesize_plot() returns a series of plots equal to the number of mu1 used in the procedure. Y-axis denotes minimum sample size and X-axis denotes minimum power.

Examples

```
SSmat <- sample.size2(mu0=0.56, sd0=0.255, mu1.start = 0.60, mu1.end = 0.70, mu1.by = 0.05,
power.start = 0.7, power.end = 0.9, power.by = 0.1, link.type = "all")
samplesize_plot(SSmat, "all")
SSmat2 <- sample.size2(mu0=0.56, sd0=0.255, mu1.start = 0.60, mu1.end = 0.70, mu1.by = 0.05,
power.start = 0.7, power.end = 0.9, power.by = 0.1, link.type = c("logit","loglog","log"))
samplesize_plot(SSmat2,link.type = c("logit","loglog","log"))
```

Index