

Package ‘exact.n’

July 22, 2025

Type Package

Title Exact Samples Sizes and Inference for Clinical Trials with Binary Endpoint

Version 1.1.1

Date 2023-11-08

Author Chris J. Lloyd

Maintainer Chris J. Lloyd <c.lloyd@mbs.edu>

Description Allows the user to determine minimum sample sizes that achieve target size and power at a specified alternative. For more information, see “Exact sample sizes for clinical trials subject to size and power constraints” by Lloyd, C.J. (2022) Preprint <[doi:10.13140/RG.2.2.11828.94085](https://doi.org/10.13140/RG.2.2.11828.94085)>.

License GPL-2

Depends R(>= 3.5)

Imports graphics, stats, http

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2023-11-09 02:30:02 UTC

Contents

fetch.data	2
inference	3
main.function	4
n1.get	5
n1.get.solution	7
POWER	8
power.image	10
Index	12

`fetch.data`*Download a target power library*

Description

Function downloads one of 85 power libraries from chrislloyd.com.au. The libraries are all four column matrices with roughly 20 million rows and will be around 1.4Gb within R. Download should take roughly a minute. The object will have a name of the form LIB.alpha.delta.Rdata.

Usage

```
fetch.data(alpha, delta, prin = FALSE)
```

Arguments

alpha	value of nominal size of test
delta	value of clinically relevant difference
prin	If TRUE, error and/or warning messages are printed.

Value

No return value. A library object will appear in the global environment.

Author(s)

Chris J. Lloyd

References

C.J. Lloyd (2022) Exact samples sizes for clinical trials subject to size and power constraints.
[doi:10.13140/RG.2.2.11828.94085](https://doi.org/10.13140/RG.2.2.11828.94085)

Examples

```
#'  
## Not run:  
fetch.data(alpha=0.05,delta=0.10)  
# Object LIB.a050.d10 will appear with 83,289,960 elements  
  
## End(Not run)
```

inference*Approximate and exact tests for single data set*

Description

For a single provided data sets (y_0 , n_0 , y_1 , n_1) calculate approximate and quasi-exact test statistics of the null value $p_1 - p_0 = \text{psi}$.

Usage

```
inference(y0, n0, y1, n1, psi = 0, sided = 1, dec.places = 4)
```

Arguments

y_0	number of successes for control
n_0	control sample size
y_1	number of successes for treatment
n_1	treatment sample size
psi	null value of risk difference $p_1 - p_0$
sided	1-sided or 2-sided test
dec.places	decimal places of output T and P values

Value

List with elements

y	the data vector y_0, y_1
n	the sample size vector n_0, n_1
approx	approximate T-value and p-value
quasi.exact	quasi.exact T-value and p-value
pmle	profile maximum likelihood estimate of baseline probability
psi	null value of $p_1 - p_0$

Author(s)

Chris J. Lloyd

References

C.J. Lloyd (2022) Exact samples sizes for clinical trials subject to size and power constraints. Preprint. doi:[10.13140/RG.2.2.11828.94085](https://doi.org/10.13140/RG.2.2.11828.94085)

Examples

```

y0=25
y1=41
n0=65
n1=82
# Non-inferiority test of  $p_1 - p_0 > -0.1$ . Evidence is strong.
inference(y0,n0,y1,n1,psi=-0.1)

```

main.function

Provide sample size solutions for target size and power.

Description

Function gives smallest values for n_1 as function of n_0 that achieve target size and power.

Usage

```

main.function(
  alpha,
  delta,
  beta = 0.75,
  p0 = 0.5,
  type = 2,
  plt = FALSE,
  out = (-1),
  b.lim = 5,
  prin = TRUE
)

```

Arguments

alpha	value of nominal size of test
delta	value of clinically relevant difference
beta	scalar target for power
p0	single value or range of values for baseline probability
type	type of maximisation (see n1.get documentation)
plt	If TRUE, plot n_1 solutions versus
out	More solutions output if $out > 0$ than $out < 0$ (see details)
b.lim	maximum imbalance of sample sizes
prin	If TRUE, error messages will be printed.

Details

If $\text{out} > 0$ all solutions (including $n1 = \text{Inf}$) are returned. If $\text{out} = 0$, infinite values are suppressed. If $\text{out} < 0$, only output satisfying the balance criterion are output.

Value

list with elements $n0$ and $n1$

Note

The appropriate data file needs to have been downloaded corresponding to the desired value of α and δ . This can be done with the `fetch.data()` function.

Author(s)

Chris J. Lloyd

References

C.J. Lloyd (2022) Exact samples sizes for clinical trials subject to size and power constraints. Preprint. doi:[10.13140/RG.2.2.11828.94085](https://doi.org/10.13140/RG.2.2.11828.94085)

Examples

```
# We are interested in designs with power at least 0.75 when exact size
# 0.025 and delta=0.20. Therefore, you would need to have downloaded
# LIB.a025.d20 using fetch(0.015,0.20). The example below instead uses
# the toy data that comes with the package. The baseline probability is
# assumed to be between 0.3 and 0.5.
rdata_file = system.file('files', 'LIB.a025.d20.Rdata', package = 'exact.n')
load(rdata_file)
#' main.function(.025,0.20,p0=c(0.3,0.5),beta=0.75,plt=TRUE)
# The value of the function is the minimum value of n1 for a range
# of values of n0. The sample size ratio is limited to 5 by default.
```

n1.get

Find smallest value of n1 that achieves target power

Description

Function calculates minimum value of $n1$ that achieves power β . If there is no solution less than 500, it models observed powers as a function of $n1$ and then extrapolates. It returns an infinite value if the power is unattainable.

Usage

```
n1.get(data, n0, beta, delta, alpha, type = 1, plt = FALSE)
```

Arguments

<code>data</code>	a matrix with column names <code>beta</code> , <code>p0.vals</code> , <code>n0</code> , <code>n1</code> and 481^2 rows (See details)
<code>n0</code>	scalar value of <code>n0</code> between 20 and 500 inclusive
<code>beta</code>	scalar target for power
<code>delta</code>	value of clinically relevant difference
<code>alpha</code>	value of nominal size of test
<code>type</code>	type of minimum solution (See details)
<code>plt</code>	if true produce diagnostic plots

Details

This function is called by `n1.get.vector` and will likely never be run by the user.

The data matrix will be a subset of one of the 85 main databases. Supplying `alpha` and `delta` loads the appropriate database and selecting a value of `p0` further subsets this data base. The resulting matrix will have 4 columns and 481^2 rows corresponding to all values of `n0` and `n1` from 20 to 500 inclusive and is suitable for input into `n1.get`.

If `type=1`, the smallest value `N1` of `n1` so that $\text{power} > \text{beta}$ is returned. If `type=2`, the smallest value `N1` of `n1` so that $\text{power} > \text{beta}$ for all $n1 \geq N1$ is returned.

Value

a single scalar value of `n1`. An integer indicates a solution was found in the database. A non-integer indicates an extrapolated solution. An Inf value indicates no extrapolated solution was found.

Author(s)

Chris J. Lloyd

References

C.J. Lloyd (2022) Exact samples sizes for clinical trials subject to size and power constraints. Preprint. doi:[10.13140/RG.2.2.11828.94085](https://doi.org/10.13140/RG.2.2.11828.94085)

Examples

```
# Load toy version of power library for alpha=0.025, delta=0.20.
# Alternatively, load the full library using fetch.data(alpha=0.025,delta=0.20)
rdata_file = system.file('files', 'LIB.a025.d20.Rdata', package = 'exact.n')
load(rdata_file)
data=LIB.a025.d20[LIB.a025.d20[,2]==0.5,] # select subset with p0=.5
# For given value of n0, what minimum value of n1 ensures power at least 0.7?
n1.get(data, n0=70, beta=.7, delta=.2, alpha=0.025,
       type = 1, plt = TRUE) # Explicit solution 63 found in data base
n1.get(data, n0=50, beta=.7, delta=.2, alpha=0.025,
       type = 1, plt = TRUE) # Approximate solution 131 extrapolated from toy data base
# You can check the accuracy of this extrapolated result:
# POWER(n0=50,n1=131,alpha=0.025,delta=0.2,p0=0.5,type="elr")
# The minimum power at p0=0.5 is 0.699, slightly less than 0.7.
```

```
# With the full library n1.get returns the correct answer of n1=136.
n1.get(data, n0=30, beta=.7, delta=.2, alpha=0.025,
       type = 1, plt = TRUE) # Extrapolated solution is infinite
```

n1.get.solution	<i>Find minimal n1 achieving target power for range of values of p0</i>
-----------------	---

Description

This function calls `n1.get.vector`, runs it for all values of `p0` within the supplied range and then takes the worst (i.e. largest) solution for `n1`

Usage

```
n1.get.solution(data, beta, p0, delta, alpha, type = 1)
```

Arguments

<code>data</code>	power database for selected value of <code>alpha</code> and <code>delta</code>
<code>beta</code>	scalar target for power
<code>p0</code>	single value or range of values for baseline probability
<code>delta</code>	value of clinically relevant difference
<code>alpha</code>	value of nominal size of test
<code>type</code>	type of minimum solution (See details)

Value

vector of solutions for `n1` with name vector equal to range of `n0` values

Author(s)

Chris J. Lloyd

References

C.J. Lloyd (2022) Exact sample sizes for clinical trials subject to size and power constraints. Preprint. doi:[10.13140/RG.2.2.11828.94085](https://doi.org/10.13140/RG.2.2.11828.94085)

Examples

```
# Load toy version of power library for alpha=0.025, delta=0.20.
rdata_file = system.file('files', 'LIB.a025.d20.Rdata', package = 'exact.n')
load(rdata_file)
# n0 solutions when p0=0.5
n1.get.solution(LIB.a025.d20,beta=.7,p0=0.5,delta=0.2,alpha=0.025,type=1)
# n0 solutions for p0 between 0.4 ad 0.5
n1.get.solution(LIB.a025.d20,beta=.7,p0=c(0.4,0.5),delta=0.2,alpha=0.025,type=1)
```

POWER

Exact Power of Test for Selected Sample Sizes.

Description

For known values of the sizes n_0 , this function computes the exact probability of rejecting the null as a function of baseline probability.

Usage

```
POWER(
  n0,
  n1,
  alpha = 0.05,
  delta = 0,
  psi = 0,
  type = "lr",
  sided = 1,
  p0 = NULL,
  obj = NULL
)
```

Arguments

n_0	control sample size
n_1	treatment sample size
α	value of nominal size of test
δ	value of clinically relevant difference
ψ	null value of risk difference $p_1 - p_0$
type	either "lr" for approximate or "elr" for quasi-exact test
sided	(1 or 2 sided test)
p_0	baseline probability. If missing, a grid of values is created for plotting. A scalar value can also be supplied.
obj	Optional object with all possible p-values. Must be a list with elements y_0 , y_1 , P (typically output of <code>lr.rd</code> or <code>ESTEP.rd</code>). If not supplied then object is generated from n_0 , n_1 and ψ .

Details

p_0 values must be between $\max(0, -\delta)$ and $\min(1, 1 - \delta)$

Value

list with element x (containing values of baseline probability) and element y (containing corresponding exact powers)

Author(s)

Chris J. Lloyd

References

C.J. Lloyd (2022) Exact samples sizes for clinical trials subject to size and power constraints. Preprint. doi:[10.13140/RG.2.2.11828.94085](https://doi.org/10.13140/RG.2.2.11828.94085)

Examples

```
oldpar <- graphics::par()
# Typical usage
#     POWER(n0=65,n1=82,psi=0,type="lr") # Exact size of approximate lr test
#     POWER(n0=65,n1=82,psi=0,type="elr",delta=.2) # Exact power of quasi exact test
# To make examples run faster, the package includes objects that contain
# all possible values of various tests when n0=65, n1=82.
load(system.file('files', "lr.stats.Rdata", package = 'exact.n'))
# All possible values of LR statistic p-values for testing  $p_1 - p_0 > 0$ 
load(system.file('files', "elr.stats.Rdata", package = 'exact.n'))
# All possible values of ELR statistic p-values for testing  $p_1 - p_0 > 0$ 
load(system.file('files', "elr10.stats.Rdata", package = 'exact.n'))
# Object contains all exact p-values for testing if  $p_1 - p_0 > 0.1$ 
# All possible values of ELR statistic p-values for testing  $p_1 - p_0 > 0.1$ 
#
graphics::par(mfrow=c(1,2))
# When delta=0 this gives type 1 error. The first plot is for the approximate
# lr based p-value, the second is for the quasi-exact e-p-value (alpha=0.05)
plot(POWER(n0=65,n1=82,alpha=0.05,psi=0,delta=0),type="l",
     xlab=expression("p"[0]),ylab="exact size")
abline(h=.05,lty=2)
plot(POWER(obj=elr.stats,alpha=0.05,delta=0),type="l",
     xlab=expression("p"[0]),ylab="exact size")
abline(h=.05,lty=2)
#
# For these sample sizes, power curve is calculated below for
# values of delta=0.1, 0.12, 0.14, 0.16, 0.18, 0.20. Power
# is poor for detecting a difference of 0.1 (see red).
plot(POWER(obj=lr.stats,alpha=0.05,delta=0.1),type="l",
     xlab=expression("p"[0]),ylab="exact power")
TITLE=expression('Exact power of LR test of  $p'[1] - p'[0] > 0.$ ')
title(main=TITLE,cex.main=0.8)
lines(POWER(obj=lr.stats,alpha=0.05,delta=0.12))
lines(POWER(obj=lr.stats,alpha=0.05,delta=0.14))
```

```

lines(POWER(obj=lr.stats,alpha=0.05,delta=0.16))
lines(POWER(obj=lr.stats,alpha=0.05,delta=0.18))
lines(POWER(obj=lr.stats,alpha=0.05,delta=0.20))
lines(POWER(obj=lr.stats,alpha=0.05,delta=0.10),col="red")
#
# The results below are for testing  $p_1 - p_0 > 0.1$ .
plot(c(0,.9),c(0,1),type="n",
     xlab=expression("p"[0]),ylab="Pr(reject null)")
lines(POWER(obj=elr10.stats,alpha=0.05,psi=0.1,delta=0.1)) # Note delta=psi
abline(h=0.05,lty=2)
lines(POWER(obj=elr10.stats,alpha=0.05,delta=0.25),col="blue")
TITLE=expression('Exact size and power of test of  $p'[1]*'-p'[0]*'>0.1.'$ ')
title(main=TITLE,cex.main=0.8)
legend(.4,.4,lty=c(1,1),col=c("black","blue"),box.col="white",
      legend=c("size:  $p_1 - p_0 = 0.1$ ", "power:  $p_1 - p_0 = 0.25$ "),cex=.7)
# When using the package the above plots would be generated by
# lines(POWER(n0=65,n1=82,alpha=0.05,psi=0.1,delta=0.10,type="elr"))
# lines(POWER(n0=65,n1=82,alpha=0.05,psi=0.1,delta=0.25,type="elr"))
suppressWarnings(graphics::par(oldpar))

```

power.image

Image plot of power

Description

Creates image plot of exact power for range of values of n_0 and n_1 . Optionally, the image plot will show where power exceeds a provided power target.

Usage

```
power.image(data, p = 0.2, binary = TRUE, beta = 0.75)
```

Arguments

data	a matrix of powers with columns named beta p0.vals, n_0 and n_1 .
p	a scalar value for p0.vals that is used to select the subset of data.
binary	If TRUE only the binary indicator of power exceeding beta is displayed.
beta	target value of power

Details

data will be one of the 85 saved objects corresponding to a selected value of alpha and delta.

Value

No return value and if the 'binary' parameter is TRUE then an image will be displayed

Author(s)

Chris J. Lloyd

Examples

```
oldpar <- graphics::par()
# Load toy version of power library for alpha=0.025, delta=0.20.
rdata_file = system.file('files', 'LIB.a025.d20.Rdata', package = 'exact.n')
load(rdata_file)
graphics::par(mfrow=c(1,2))
power.image(LIB.a025.d20,p=.5,binary=FALSE)
power.image(LIB.a025.d20,p=.2,beta=.7,binary=TRUE)
suppressWarnings(graphics::par(oldpar))
```

Index

`fetch.data`, [2](#)

`inference`, [3](#)

`main.function`, [4](#)

`n1.get`, [5](#)

`n1.get.solution`, [7](#)

`POWER`, [8](#)

`power.image`, [10](#)