Package ‘rPowerSampleSize’
October 14, 2022

Type Package
Title Sample Size Computations Controlling the Type-II Generalized
Family-Wise Error Rate
Version 1.0.2
Date 2018-05-10
License GPL (> 2)
LazyLoad yes
Author Pierre Lafaye de Micheaux, Benoit Liquet and Jeremie Riou
Maintainer Pierre Lafaye de Micheaux <lafaye@unsw.edu.au>
Description The significance of mean difference tests in clinical trials is estab-
lished if at least r null hypotheses are rejected among m that are simultane-
ously tested. This package enables one to compute necessary sample sizes for single-step (Bon-
ferroni) and step-wise procedures (Holm and Hochberg). These three procedures control the q-
genralized family-wise error rate (probability of making at least q false rejections). Sam-
ple size is computed (for these single-step and step-wise procedures) in a such a way that the r-
power (probability of rejecting at least r false null hypotheses, i.e. at least r significant end-
points among m) is above some given threshold, in the context of tests of differ-
ence of means for two groups of continuous endpoints (variables). Various types of struc-
ture of correlation are considered. It is also possible to analyse data (i.e., actually test difference
in means) when these are available. The case r equals 1 is treated in separate func-
tions that were used in Lafaye de Micheaux et al. (2014) <doi:10.1080/10543406.2013.860156>.
Depends R (>= 2.10.0), mvtnorm, stats, ssanv, parallel
NeedsCompilation no
Repository CRAN
Date/Publication 2018-05-10 12:16:42 UTC

R topics documented:

rPowerSampleSize-package ................................................. 2
bonferroni.1m.ssc ..................................................... 3
complexity ............................................................. 4
data ........................................................................... 5
The package rPowerSampleSize has been developed in order to compute the sample size, and to analyse the data, in the context of multiple primary continuous endpoints in clinical trials. Two approaches are available for the decision rule “at least one significant endpoint among m”. All the procedures which correspond to this decision rule include 1m in the name of the function. An individual procedure based on Union-Intersection Procedure, and a global method based on multivariate linear model. This package for this decision rule contains four functions: two for the sample size determination (one for individual procedure, and one for a global method), and two for data analysis in order to resolve the multiple testing problems (one for individual procedure, and one for a global method). The sample size computation for the Bonferroni Procedure is also available. A generalisation of the first case is also available with a decision rule “at least r significant endpoint among the m primary endpoint”. The procedure which corresponds to this decision rule include rm in the name of the function. We have one functions here where we study the sample size computation.

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>rPowerSampleSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>2012-03-21</td>
</tr>
<tr>
<td>License:</td>
<td>GPL (&gt;2)</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>
Author(s)
Pierre Lafaye de Micheaux, Benoit Liquet and Jeremie Riou
Maintainer: Pierre Lafaye de Micheaux
<lafaye@unsw.edu.au>

Sample Size Computation with Single Step Bonferroni Method in the
Context of Multiple Continuous Endpoints.

Description
This function computes the sample size for an analysis of multiple test
with a single step Bonferroni procedure.

Usage

bonferroni.1m.ssc(mean.diff, sd, cor, power = 0.8, alpha = 0.05,
alternative = "two.sided")

Arguments

mean.diff vector of the mean differences of the m endpoints between
both groups under the alternative hypothesis.

sd vector of the standard deviations of the m endpoints. These are assumed identical
for both groups.

cor correlation matrix between the endpoints. These are assumed identical for both
groups.

power value which corresponds to the chosen power.

alpha value which corresponds to the chosen Type-I error rate bound.

alternative character string specifying the alternative hypothesis, must be one of "two.sided"
(default), "greater" or "less".

Value
Sample size The required sample size.

Author(s)
P. Lafaye de Micheaux, B. Liquet and J. Riou

References
Determination in Clinical Trials With Multiple Primary Continuous Correlated Endpoints,
46:261-283.
complexity

See Also
global.1m.analysis, indiv.1m.ssc, indiv.1m.analysis, global.1m.ssc

Examples

```r
## Not run:
# Sample size computation for the global methodonferroni.1m.ssc(mean.diff = c(0.1, 0.2, 0.3), sd = c(1, 1,1 ), cor =
diag(1, 3))
## End(Not run)
```

complexity

Computation of the complexity of the numerical computations.

Description

This function computes the complexity of our formulas.

Usage

```r
complexity(method, exchangeable, r, p, display = TRUE)
```

Arguments

- `method`:  "Bonferroni", "Hochberg" or "Holm".
- `exchangeable`: Logical indicating if we are in the exchangeable case or not.
- `r`: integer, r = 1, ..., m. Desired number of endpoints to be declared significant.
- `p`: integer, p = 1, ..., m. Indicates the number of false null hypotheses.
- `display`: Logical. Should we display the complexity.

Value

Complexity of the formulas (i.e., number of individual probabilities computed using the function pmvt (or pmvnorm) in the non exchangeable case).

Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

References

Simulated data

Description

This simulated dataset contains three outcomes, and one adjustment variable, divided in two groups.

Usage

data(data.sim)

Format

A data frame with 100 observations on the following 5 variables.

Y1 numeric vector of the first outcome
Y2 numeric vector of the second outcome
Y3 numeric vector of the third outcome
Group binary vector of the group variable
A numeric vector of the adjustment variable

Examples

data(data.sim)

df.compute Computation of degrees of freedom.

Description

This function computes the degrees of freedom.

Usage

df.compute(nE, nC, SigmaE = NULL, SigmaC = NULL, matrix.type = NULL, equalSigmas = NULL, m = NULL)
Arguments

nE Sample size for the experimental (test) group.
nC Sample size for the control group.
SigmaE NULL or a matrix indicating the covariances between the primary endpoints in
the experimental (test) group. See Details.
SigmaC NULL or a matrix indicating the covariances between the primary endpoints in
the control group. See Details.
matrix.type NULL or an integer among 1, 2, 3, 4, giving the type of the matrices 'SigmaE'
and 'SigmaC'. See Details.
equalSigmas NULL or a logical indicating if 'SigmaC' and 'SigmaE' are equal. See Details.
m NULL or the value for m. See Details.

Details

You should provide either both SigmaE, SigmaC or both matrix.type, equalSigmas. When you
provide the former, the latter should be set to NULL. And vice versa.

Value

df The degrees of freedom.

Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

References

Error Rate Formulas with Application to Sample Size Determination. Statistics in Medicine.

See Also

global.1m.analysis, indiv.1m.ssc, indiv.1m.analysis, global.1m.ssc

Examples

```r
## Not run:
# standard deviation of the treatment effect
var <- c(0.3520^2,0.6219^2,0.5427^2,0.6075^2,0.6277^2,0.5527^2,0.8066^2)

# Correlation matrix
cov <- matrix(1,ncol=7,nrow=7)
cov[1,2:7] <- cov[2:7,1] <- c(0.1341692,0.1373891,0.07480123,0.1401267,0.1280336,0.1614103)
cov[2,3:7] <- cov[3:7,2] <- c(0.2874531,0.18451960,0.3156895,0.2954996,0.3963837)
cov[3,4:7] <- cov[4:7,3] <- c(0.19903400,0.2736123,0.2369907,0.3423579)
cov[4,5:7] <- cov[5:7,4] <- c(0.1915028,0.1558958,0.2376056)
```
Data analysis with a global method in the context of multiple continuous endpoints

Description
This function aims at analysing \( m \) multiple continuous endpoints with a global procedure. The clinical aim is to be able to detect a mean difference between the test \( T \) and the control \( C \) product for at least one endpoint among \( m \). This method is based on a multivariate model taking into account the correlations between the \( m \) endpoints and possibly some adjustment variables. The result gives only a global decision.

Usage
\[
global.1m.analysis(XC, XT, A, alpha = 0.05, n = NULL)
\]

Arguments
\begin{itemize}
  \item \textit{XC} \hspace{1cm} matrix of the outcome for the control group.
  \item \textit{XT} \hspace{1cm} matrix of the outcome for the test group.
  \item \textit{A} \hspace{1cm} matrix of the adjustment variables.
  \item \textit{n} \hspace{1cm} sample size of a group. The sample size needs to be the same for each group.
  \item \textit{alpha} \hspace{1cm} value which corresponds to the chosen Type-I error rate bound.
\end{itemize}

Value
\[
\text{Pvalue} \hspace{1cm} \text{the p-value of the global test.}
\]

Author(s)
P. Lafaye de Micheaux, B. Liquet and J. Riou

References
**See Also**

*global.1m.ssc, indiv.1m.ssc, indiv.1m.analysis, bonferroni.1m.ssc*

**Examples**

```r
# Calling the data
data(data.sim)

# Data analysis for the global method
n <- nrow(data) / 2
XC <- data[1:n, 1:3]
XT <- data[(n + 1):(2 * n), 1:3]

global.1m.analysis(XC = XC, XT = XT, A = data[, 5])
```

---

**global.1m.ssc**  
*Sample Size Computation Based on a Global Procedure in the Context of Multiple Continuous Endpoints*

**Description**

This function computes the sample size with a global method in the context of $m$ multiple continuous endpoints. Two groups are considered: $C$ for control and $T$ for treatment. The clinical aim is to be able to detect a mean difference between the test and the control product for at least one endpoint among $m$. This method is based on a multivariate model with co-variates taking into account the correlations between the endpoints.

**Usage**

```r
global.1m.ssc(method, mean.diff, sd, cor, v = NULL, M = NULL, power = 0.8, alpha = 0.05)
```

**Arguments**

- `method` either "Model" if no co-variates are involved and "Adj.Model" for a model with co-variates.
- `mean.diff` vector of the mean differences of the $m$ endpoints between both groups under the alternative hypothesis.
- `sd` vector of the standard deviations of the $m$ endpoints. These are assumed identical for both groups.
- `cor` correlation matrix between the endpoints. These are assumed identical for both groups.
- `v` $v$ is a $p \times 1$ vector whose $l^{th}$ component is $v_l = \bar{a}_l^C - \bar{a}_l^T$, where $p$ is the number of adjustment variables, and $\bar{a}_l^i$ is the mean of the adjustment variable $a_l$ for the group $i$, $i = C, T$. 

---
**M** is a $p \times p$ matrix with general term $M_{l,l'} = (\bar{a}_lC - \bar{a}'_lC) + (\bar{a}_lT - \bar{a}'_lT)$. 

**Power** value which corresponds to the chosen power. 

**Alpha** value which corresponds to the chosen Type-I error rate bound. 

### Value 

**Sample size** The required sample size. 

### Author(s) 

P. Lafaye de Micheaux, B. Liquet and J. Riou 

### References 


### See Also 

`global.1m.ssc, indiv.1m.ssc, indiv.1m.analysis, bonferroni.1m.ssc` 

### Examples 

```r 
# Sample size computation for the global method 
global.1m.ssc(method = "Adj.Model", mean.diff = c(0.1, 0.2, 0.3), sd = c(1, 1, 1), cor = diag(1, 3), v = -0.2, M = 0.46) 

# Table 2 in our 2014 paper: 
Sigma2 <- matrix(c(5.58, 2, 1.24, 2, 4.29, 1.59, 1.24, 1.59, 4.09), ncol = 3) 
sd2 <- sqrt(diag(Sigma2)) 
cor2 <- diag(1 / sd2) %*% Sigma2 %*% diag(1 / sd2) 
mu2 <- c(0.35, 0.28, 0.46) 
m <- 3 
global.1m.ssc(method = "Model", mean.diff = mu2, sd = sd2, cor = cor2) 
```

### Description 

This function aims at analysing some multiple continuous endpoints with an individual testing procedure. This method, based on Union-Intersection test procedure, allows one to take into account the correlations between the different endpoints in the analysis.
Usage

`indiv.1m.analysis(method, XC, XT, varX = NULL, alpha = 0.05, alternative = "two.sided", n = NULL)`

Arguments

- `method` description of the covariance matrix estimation. Two choices are possible: "Unknown" (normality assumption and unknown covariance matrix) and "Asympt" (asymptotic context).
- `XC` matrix of the outcomes for the control group.
- `XT` matrix of the outcomes for the test group.
- `varX` covariance matrix. Should be provided when `method` = 'Known'.
- `alpha` value which corresponds to the chosen Type-I error rate bound.
- `alternative` character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
- `n` NULL. Sample size of a group, computed from `XC`.

Value

- `UnAdjPvalue` unadjusted p-values.
- `AdjPvalue` corrected p-values.

Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

References


See Also

`global.1m.ssc`, `indiv.1m.ssc`, `global.1m.analysis`, `bonferroni.1m.ssc`

Examples

```r
# Calling the data
data(data.sim)

# Data analysis for the individual method
n <- nrow(data) / 2

XC <- data[1:n, 1:3]
XT <- data[(n + 1):(2 * n), 1:3]

indiv.1m.analysis(method = "UnKnown", XC = XC, XT = XT)
```
Sample size computation with an individual testing procedure in the context of multiple continuous endpoints

Description

This function computes the sample size with an individual testing procedure in the context of multiple continuous endpoints. This method, based on the Union-Intersection testing procedure, allows one to take into account the correlation between the different endpoints in the computation of the sample size.

Usage

```r
indiv.1m.ssc(method, ES, cor, power = 0.8, alpha = 0.05, alternative = "two.sided", tol = 1e-04, maxiter = 1000, tol.uniroot = 1e-04)
```

Arguments

- `method`: description of the covariance matrix estimation. Two choices are possible: "Unknown" (normality assumption and unknown covariance matrix) and "Asympt" (asymptotic context).
- `ES`: vector indicating the values of the effect size. The definition of the effect size is presented in the "Details" section.
- `cor`: matrix indicating the correlation matrix between the endpoints.
- `power`: value which corresponds to the chosen power.
- `alpha`: value which corresponds to the chosen Type-I error rate bound.
- `alternative`: character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
- `tol`: the desired accuracy (convergence tolerance) for our algorithm.
- `maxiter`: maximum number of iterations.
- `tol.uniroot`: desired accuracy (convergence tolerance) for the uniroot.integer function.

Details

ES: The effect size definition parameter for the \( k \)th endpoint is defined as \( \frac{\mu_T^k - \mu_C^k}{\sigma_k^*} \), where \( \sigma_k^* \) refers to the standard deviation of the population from which the different treatment groups were taken and \( \mu_T^k - \mu_C^k \) is the true mean difference between the test and the control group for the \( k \)th group.

We consider that: \( \sigma_k^* = \frac{\sigma_T^2 + \sigma_C^2}{2} \).

Value

- Adjusted Type-I error rate
  - adjusted Type-I error rate.
- Sample size
  - the required sample size.
Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

References


See Also

global.1m.ssc, global.1m.analysis, indiv.1m.analysis, bonferroni.1m.ssc

Examples

# Sample size computation for the individual method
indiv.1m.ssc(method = "Known", ES = c(0.1, 0.2, 0.3), cor = diag(1, 3))

# Table 2 in our 2014 paper:
Sigma2 <- matrix(c(5.58, 2, 1.24, 2, 4.29, 1.59, 1.24, 1.59, 4.09), ncol = 3)
sd2 <- sqrt(diag(Sigma2))
cor2 <- diag(1 / sd2) %*% Sigma2 %*% diag(1 / sd2)
mu2 <- c(0.35, 0.28, 0.46)
m <- 3
indiv.1m.ssc(method = "Known", ES = mu2 / sd2, cor = cor2)

Data analysis using an individual testing procedure controlling the q-gFWER in the context of m multiple continuous endpoints

Description

This function aims at analysing some multiple continuous endpoints with individual testing procedures (Bonferroni, Holm, Hochberg). These procedures, based on a Union-Intersection test procedure, allow to take into account the correlation between the different endpoints in the analysis. This function uses critical values from Romano et al. to control the q-gFWER. Different structures of the covariance matrices between endpoints are considered.

Usage

indiv.1m.ssc(method, XE, XC, d, matrix.type, equalSigmas, alpha = 0.05, q = 1, rho = NULL, alternative = "greater", orig.Hochberg = FALSE)
Arguments

method
"Bonferroni", "Holm" or "Hochberg". When method = "Hochberg", we use critical values involving the D1 term in formula (11) of Romano et al. in order to control strongly the $q$-FWER. If you want to use the original Hochberg’s procedure, set orig.Hochberg to TRUE. Even for $q = 1$, this is a bad idea except when the p-values can be assumed independent.

XE
matrix (of size $n_E \times m$) of the outcome for the experimental (test) group.

XC
matrix (of size $n_C \times m$) of the outcome for the control group.

d
vector of length $m$ indicating the true value of the differences in means under the null hypothesis.

matrix.type
integer value equal to 1, 2, 3, 4 or 5. A value of 1 indicates multisample sphericity. A value of 2 indicates multisample variance components. A value of 3 indicates multisample compound symmetry. A value of 4 indicates multisample compound symmetry with unequal individual (endpoints) variances. A value of 5 indicates unstructured variance components.

equalSigmass
logical. Indicates if $\Sigma_E$ is equal to $\Sigma_C$.

alpha
value which corresponds to the chosen $q$-FWER type-I error rate control bound.

q
integer. Value of ‘$q$’ ($q=1,\ldots,m$) in the $q$-FWER of Romano et al., which is the probability to make at least $q$ false rejections. The default value $q=1$ corresponds to the classical FWER control.

rho
NULL or should be provided only if matrix.type is equal to 3 or 4. This is the value of correlation for the compound symmetry case.

alternative
NOT USED YET. Character string specifying the alternative hypothesis, must be one of "two.sided", "greater" or "less".

orig.Hochberg
logical. To use the standard Hochberg’s procedure.

Value

list(stat = statvec, pvals = pvals, AdjPvals = pvals.adj, sig2hat = varhatvec)

stat
individual test statistic values.

pvals
non corrected p-values.

pvals.adj
corrected p-values.

sig2hat
estimated variance (i.e., square of denominator of the test statistic).

Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

References


See Also

indiv.rm.ssc.

indiv.rm.ssc  Sample size determination in the context of multiple continuous endpoints with a control of the q-gFWER, for a given value of r-power (generalized disjunctive power).

Description

This function computes the sample size for an analysis of \( m \) multiple tests with a control of the q-gFWER.

Usage

```r
indiv.rm.ssc(method, asympt = FALSE, r, m, p = m, nCovernE = 1, 
muC = NULL, muE = NULL, d = NULL, delta = NULL, SigmaC = NULL, 
SigmaE = NULL, power = 0.8, alpha = 0.05, interval = c(2, 2000), q = 1, 
maxpts = 25000, abseps = 0.001, releps = 0, nbcores = 1, LB = FALSE, 
orig.Hochberg = FALSE)
```

Arguments

- `method` : "Bonferroni", "Hochberg" or "Holm". When `method = "Hochberg"`, we use critical values involving the D1 term in formula (11) of Romano et al. in order to control strongly the q-FWER. If you want to use the original Hochberg's procedure, set `orig.Hochberg` to `TRUE`. Even for \( q = 1 \), this is a bad idea except when the p-values can be assumed independent.
- `asympt` : logical. `TRUE` for the use of the asymptotic approximation by a multivariate normal distribution or `FALSE` for the multivariate Student distribution.
- `r` : integer, \( r = 1, ..., m \). Desired number of endpoints to be declared significant.
- `m` : integer. Number of endpoints.
- `p` : integer, \( p = 1, ..., m \). Indicates the number of false null hypotheses.
- `nCovernE` : ratio of \( n_C \) over \( n_E \).
- `muC` : \( \mu_C \) or a vector of length \( m \) of the true means of the control group for all endpoints under the alternative hypothesis. If \( \mu_C \), \( \mu_E \) and \( d \) are \( \text{NULL} \), then \( \delta \) should be provided instead.
- `muE` : \( \mu_E \) or a vector of length \( m \) of the true means of the experimental (test) group for all endpoints under the alternative hypothesis.
- `d` : \( d \) or a vector of length \( m \) indicating the true value of the differences in means under the null hypothesis.
- `delta` : should be \( \text{NULL} \) if \( \mu_C \), \( \mu_E \) and \( d \) are provided. If not, it is equal to \( \mu_E - \mu_C - d \) and these parameters should be set to \( \text{NULL} \).
SigmaC  matrix giving the covariances between the m primary endpoints in the control group.

SigmaE  matrix giving the covariances between the m primary endpoints in the experimental (test) group.

power  a value which corresponds to the chosen r-power.

alpha  a value which corresponds to the chosen q-gFWER type-I control bound.

interval  an interval of values in which to search for the sample size. Left endpoint should be greater than or equal to 2.

q  integer. Value of 'q' (q=1,...,m) in the q-gFWER of Romano et al., which is the probability to make at least q false rejections. The default value q=1 corresponds to the classical FWER control.

maxpts  convergence parameter used in the GenzBretz function. A good choice is \( \min(25000 \times 10^{\text{true.complexity}}, \text{.Machine}$integer.max) where true.complexity is computed with the complexity function. But note that this might considerably increase the computation time!

abseps  convergence parameter used in the GenzBretz function. A good choice is \( \max(0.001 / \text{true.complexity}, 1e-08) \) where true.complexity is computed with the complexity function. But note that this might considerably increase the computation time!

releps  relative error tolerance as double used in the GenzBretz function.

nbcores  integer. Number of cores to use for parallel computations.

LB  logical. Should we use a load balancing parallel computation.

orig.Hochberg  logical. To use the standard Hochberg’s procedure.

Value

The required sample size.

Note

Results can differ from one time to another because the results of the function pmvt are random. If this is the case, you should consider increasing maxpts and decreasing abseps. In any case, you should always double check using one of the functions Psirms, Psirmu or Psirmd if the sample size you obtained gives you the intended power, with an acceptable error (or at least compute the power a few times with various seeds to see if results are stable).

Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

References


See Also

indiv.analysis.

Examples

## Not run:
# Pneumovacs example (takes 37 mn to compute on 1 core)

# Treatment effect
delta <- c(0.55, 0.34, 0.38, 0.20, 0.70, 0.38, 0.86)

# Variances of the m endpoints
var <- c(0.3520, 0.6219, 0.5427, 0.6075, 0.6277, 0.5527, 0.8066) ^ 2

# Covariance matrix
cov <- matrix(1, ncol = 7, nrow = 7)
cov[1, 2:7] <- cov[2:7, 1] <- c(0.1341692, 0.1373891, 0.07480123, 0.1401267, 0.1280336, 0.1614103)
cov[2, 3:7] <- cov[3:7, 2] <- c(0.2874531, 0.18451960, 0.3156895, 0.2954996, 0.3963837)
cov[3, 4:7] <- cov[4:7, 3] <- c(0.19983400, 0.2736123, 0.236907, 0.3423579)
cov[4, 5:7] <- cov[5:7, 4] <- c(0.1915028, 0.1558958, 0.2376056)
cov[5, 6:7] <- cov[6:7, 5] <- c(0.2642217, 0.3969920)
cov[6, 7] <- cov[7, 6] <- 0.3352029
diag(cov) <- var

indiv.rm.ssc(method = "Hochberg", asympt = FALSE, r = 3, m = 7, p = 7, nCovernE = 1,
muC = NULL, muE = NULL, d = NULL, delta = delta, SigmaC = cov,
SigmaE = cov, power = 0.8, alpha = 0.05, interval = c(10, 2000), q = 1)

# Pre-RELAX-AHF example from the paper by Teerlink et al. (2009),
# Relaxin for the treatment of patients with acute heart failure
# (Pre-RELAX-AHF): a multicentre, randomised,
# placebo-controlled, parallel-group, dose-finding phase IIb
# study, Lancet, 373: 1429--39

# Table 2 page 1432:
# ---------------
# Proportion with moderately or markedly better dyspnoea at 6 h, 12 h, and 24 h (Likert): 23% 40%
# Dyspnoea AUC change from baseline to day 5 (VAS [mmxh]): 1679 (2556) 2567 (2898)
# Worsening heart failure through day 5: 21% 12%
# Length of stay (days): 12.0 (7.3) 10.2 (6.1)
# Days alive out of hospital: 44.2 (14.2) 47.9 (10.1)
# KM cardiovascular death or readmission (HR, 95% CI): 17.2% 2.6% (0.13, 0.02--1.03); p=0.053
# KM cardiovascular death (HR, 95% CI): 14.3% 0.0% (0.00, 0.00--0.98); p=0.046

# Table 4 page 1436:
# ---------------
# >=25% increase at day 5: 8 (13%) 9 (21%)
# >=26 micro-mol/L increase at days 5 and 14: 4 (7%) 3 (7%)
muC <- c(23 / 100, 1679, 1 - 21 / 100, -12.0, 44.2, 1 - 17.2 / 100, 1 -
matrix.type.compute

Computation of matrix type

Description

This function determines the type of matrix structure of $\Sigma_E$ and $\Sigma_C$, which can be multisample sphericity (type 1), multisample variance components (type 2), multisample compound symmetry (type 3) or unstructured variance components (type 4).

Usage

matrix.type.compute(SigmaE, SigmaC, display.type = FALSE)

Arguments

SigmaE    matrix giving the covariances between the $m$ primary endpoints in the experimental (test) group.
SigmaC    matrix giving the covariances between the $m$ primary endpoints in the control group.
display.type Logical. Should we display the (name of) type of the matrices.
Value

Integer indicating the structure of the matrices: 1 if both are of type 1, 2 if both are of type 2, 3 if both are of type 3 or 4 if one of them is of type 4.

Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

See Also

indiv.analysis,

Examples

```r
## Not run:
# Variances of the m endpoints
var <- c(0.3520, 0.6219, 0.5427, 0.6075, 0.6277, 0.5527, 0.8066) ^ 2

# Covariance matrix
cov <- matrix(1, ncol = 7, nrow = 7)
cov[1, 2:7] <- cov[2:7, 1] <- c(0.1341692, 0.1373891, 0.07480123,
0.1401267, 0.1280336, 0.1614103)
cov[2, 3:7] <- cov[3:7, 2] <- c(0.2874531, 0.18451960, 0.3156895,
0.2954996, 0.3963837)
cov[3, 4:7] <- cov[4:7, 3] <- c(0.19903400, 0.2736123, 0.2369907, 0.3423579)
cov[4, 5:7] <- cov[5:7, 4] <- c(0.1915028, 0.1558958, 0.2376056)
cov[5, 6:7] <- cov[6:7, 5] <- c(0.2642217, 0.3969920)
cov[6, 7] <- cov[7, 6] <- 0.3352029
diag(cov) <- var
diag(cov) <- var
diag(cov) <- var
matrix.type.compute(SigmaE = cov, SigmaC = cov, display = TRUE)
## End(Not run)
```

---

montercarlo

Monte carlo computation of power.

Description

This function approximates the power for a given sample size using a Monte Carlo simulation.

Usage

```r
montecarlo(method, M = 100000, nE, r, m, nCoverE = 1, muC,
muE, d = rep(0.0, m), SigmaE, SigmaC, alpha = 0.05, q = 1, nbcores = parallel::detectCores() - 1,
alternative = "greater", orig.Hochberg = FALSE)
```
Arguments

method
"Bonferroni", "Holm" or "Hochberg". When method = "Hochberg", we use critical values involving the D1 term in formula (11) of Romano et al. in order to control strongly the q-FWER. If you want to use the original Hochberg’s procedure, set orig.Hochberg to TRUE. Even for q = 1, this is a bad idea except when the p-values can be assumed independent.

M
number of Monte Carlo repetitions. Dmitrienko et al. (2013) suggested to take $M = 10^5$.

nE
integer. Sample size for the experimental (test) group.

r
integer, r = 1, ..., m. Desired number of endpoints to be declared significant.

m
integer. Number of endpoints.

nCover/nE
ratio of nC over nE.

muC
vector of length m of the true means of the control group for all endpoints under the alternative hypothesis.

muE
vector of length m of the true means of the experimental (test) group for all endpoints under the alternative hypothesis.

d
vector of length m indicating the true value of the differences in means under the null hypothesis.

SigmaE
matrix indicating the covariances between the m primary endpoints in the experimental (test) group. See Details.

SigmaC
matrix indicating the covariances between the m primary endpoints in the control group. See Details.

alpha
a value which corresponds to the chosen q-gFWER type-I control bound.

q
integer. Value of ’q’ (q = 1, ..., m) in the q-gFWER of Romano et al., which is the probability to make at least q false rejections. The default value q=1 corresponds to the classical FWER control.

nbcores
number of cores to use for the computations.

alternative
NOT USED YET. Character string specifying the alternative hypothesis, must be one of "two.sided", "greater" or "less".

orig.Hochberg
logical. To use the standard Hochberg’s procedure.

Value

rpowBonf or rpowHoch or rpowHolm
List with one element giving the computed power.

Author(s)

P. Lafaye de Michele, B. Liquet and J. Riou
References


See Also

`indiv.analysis`,

---

### plot.rPower

*Plot of an rPower object*

#### Description

For an `rPower` object which has been produced by `montecarlo`, this function plots the distribution (estimated by Monte-Carlo) of the number of significant results (i.e, the realized values $r$).

#### Usage

```r
## S3 method for class 'rPower'
plot(x, ...)
```

#### Arguments

- `x` The name of an object of class `rPower`.
- `...` Some methods for this generic require additional arguments. None are used in this method.

#### Value

Returns graphs.

#### Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

#### References


#### See Also

`montecarlo`
Description

This function computes the power for an analysis of $m$ multiple tests with a control of the $q$-gFWER with the Holm procedure.

Usage

Psirmd(r, m, p = m, nE, nCovernE = 1, delta, SigmaC, SigmaE, alpha = 0.05, q = 1, asympt = FALSE, maxpts = 25000, abseps = 0.001, releps = 0, nbcores = 1, LB = FALSE)

Arguments

- $r$: integer, $r = 1, \ldots, m$. Desired number of endpoints to be declared significant.
- $m$: integer. Number of endpoints.
- $p$: integer, $p = 1, \ldots, m$. Indicates the number of false null hypotheses.
- $nE$: integer. Sample size for the experimental (test) group.
- $nCovernE$: Ratio of $nC$ over $nE$.
- $delta$: vector of length $m$ equal to $muE - muC - d$.
- $SigmaC$: matrix giving the covariances between the $m$ primary endpoints in the control group.
- $SigmaE$: matrix giving the covariances between the $m$ primary endpoints in the experimental (test) group.
- $alpha$: a value which corresponds to the chosen $q$-gFWER type-I control bound.
- $q$: integer. Value of ‘$q’ ($q=1, \ldots, m$) in the $q$-gFWER of Romano et al., which is the probability to make at least $q$ false rejections. The default value $q=1$ corresponds to the classical FWER control.
- $asympt$: logical. TRUE for the use of the asymptotic approximation by a multivariate normal distribution or FALSE for the multivariate Student distribution.
- $maxpts$: convergence parameter used in the GenzBretz function. A suggested choice is $\min(25000 \times \text{true.complexity}, \, .\text{Machine}$integer$\text{.max})$ where true.complexity is computed with the complexity function. But note that this might considerably increase the computation time!
- $abseps$: convergence parameter used in the GenzBretz function. A suggested choice is $\max(0.001 \div \text{true.complexity}, \, 1e^{-08})$ where true.complexity is computed with the complexity function. But note that this might considerably increase the computation time!
- $releps$: relative error tolerance as double used in the GenzBretz function.
- $nbcores$: integer. Number of cores to use for parallel computations.
- $LB$: logical. Should we use a load balancing parallel computation.
Value

List with two components:

- **pow**: The computed power.
- **error**: The sum of the absolute estimated errors for each call to the pmvt (or pmvnorm) function. The number of such calls is given (in the non-exchangeable case) by the function complexity. Note that in the exchangeable case, some probabilities are weighted. So an error committed on such a probability is also inflated with the same weight. Note also that this global error does not take into account the signs of the individual errors and is thus most certainly higher than the true error. In other words, you are 99 percent sure that the true power is between 'pow' - 'error' and 'pow' + 'error', but it is also probably much closer to 'pow', particularly if the complexity is large.

Note

Results can differ from one time to another because the results of the function pmvt are random. If this is the case, you should consider increasing maxpts and decreasing abseps.

Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

References


See Also

- indiv.analysis

Description

This function computes the power for an analysis of \( m \) multiple tests with a control of the q-gFWER by a Bonferroni procedure.

Usage

\[
\text{Psirms}(r, m, p = m, nE, nCovernE = 1, \delta, \SigmaC, \SigmaE, \\
\alpha = 0.05, q = 1, \text{asympt} = \text{FALSE}, \\
\text{maxpts} = 25000, \text{abseps} = 0.001, \text{releps} = 0, \text{nbcores} = 1, \text{LB} = \text{FALSE})
\]
Arguments

- \( r \) integer, \( r = 1, \ldots, m \). Desired number of endpoints to be declared significant.
- \( m \) integer. Number of endpoints.
- \( p \) integer, \( p = 1, \ldots, m \). Indicates the number of false null hypotheses.
- \( n \) integer. Sample size for the experimental (test) group.
- \( nC/\text{over} nE \) Ratio of \( nC \) over \( nE \).
- \( \delta \) vector of length \( m \) equal to \( \muE - \muC - d \).
- \( \SigmaC \) matrix giving the covariances between the \( m \) primary endpoints in the control group.
- \( \SigmaE \) matrix giving the covariances between the \( m \) primary endpoints in the experimental (test) group.
- \( \alpha \) a value which corresponds to the chosen q-gFWER type-I control bound.
- \( q \) integer. Value of \( 'q' \) (\( q=1, \ldots, m \)) in the q-gFWER of Romano et al., which is the probability to make at least \( q \) false rejections. The default value \( q=1 \) corresponds to the classical FWER control.
- \( \text{asympt} \) logical. \( \text{TRUE} \) for the use of the asymptotic approximation by a multivariate normal distribution or \( \text{FALSE} \) for the multivariate Student distribution.
- \( \text{maxpts} \) convergence parameter used in the GenzBretz function. A suggested choice is \( \min(25000 \times \text{true.complexity}, .Machine\$integer.max) \) where \( \text{true.complexity} \) is computed with the \text{complexity} function. But note that this might considerably increase the computation time!
- \( \text{abseps} \) convergence parameter used in the GenzBretz function. A suggested choice is \( \max(0.001 / \text{true.complexity}, 1e-08) \) where \( \text{true.complexity} \) is computed with the \text{complexity} function. But note that this might considerably increase the computation time!
- \( \text{releps} \) relative error tolerance as double used in the GenzBretz function.
- \( \text{nbcores} \) integer. Number of cores to use for parallel computations.
- \( \text{LB} \) logical. Should we use a load balancing parallel computation.

Value

List with two components:

- \( \text{pow} \) The computed power.
- \( \text{error} \) The total sum of the absolute estimated errors for each call to the \text{pmvt} (or \text{pmvnorm}) function. The number of such calls is given (in the non-exchangeable case) by the function \text{complexity}. Note that in the exchangeable case, some probabilities are weighted. So an error committed on such a probability is also inflated with the same weight. Note also that this global error does not take into account the signs of the individual errors and is thus most certainly higher than the true error. In other words, you are 99 percent sure that the true power is between \( \text{pow} - \text{error} \) and \( \text{pow} + \text{error} \), but it is also probably much closer to \( \text{pow} \), particularly if the complexity is large.
Note

Results can differ from one time to another because the results of the function pmvt are random. If this is the case, you should consider increasing maxpts and decreasing abseps.

Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

References


See Also

indiv.analysis.

Presimu

Computation of power for step-up (Hochberg) procedure.

Description

This function computes the power for an analysis of m multiple tests with a control of the q-gFWER with the Hochberg procedure.

Usage

Psirmu(r, m, p = m, nE, nCovernE = 1, delta, SigmaC, SigmaE,
   alpha = 0.05, q = 1, asympt = FALSE,
   maxpts = 25000, abseps = 0.001, releps = 0, nbcores = 1, LB = FALSE,
   orig.Hochberg = FALSE)

Arguments

r integer, r = 1, ..., m. Desired number of endpoints to be declared significant.

m integer. Number of endpoints.

p integer, p = 1, ..., m. Indicates the number of false null hypotheses.

nE integer. Sample size for the experimental (test) group.

nCovernE Ratio of nC over nE.

delta vector of length m equal to muE - muC - d.

SigmaC matrix giving the covariances between the m primary endpoints in the control group.
Psirmu

SigmaE matrix giving the covariances between the m primary endpoints in the experimental (test) group.

alpha a value which corresponds to the chosen q-gFWER type-I control bound.

q integer. Value of 'q' (q=1,...,m) in the q-gFWER of Romano et al., which is the probability to make at least q false rejections. The default value q=1 corresponds to the classical FWER control.

asympt logical. TRUE for the use of the asymptotic approximation by a multivariate normal distribution or FALSE for the multivariate Student distribution.

maxpts convergence parameter used in the GenzBretz function. A suggested choice is \( \min(25000 * \text{true.complexity}, \ .Machine\$\text{integer.max}) \) where true.complexity is computed with the complexity function. But note that this might considerably increase the computation time!

abseps convergence parameter used in the GenzBretz function. A suggested choice is \( \max(0.001 / \text{true.complexity}, 1e-08) \) where true.complexity is computed with the complexity function. But note that this might considerably increase the computation time!

releps relative error tolerance as double used in the GenzBretz function.

nbcores integer. Number of cores to use for parallel computations.

LB logical. Should we use a load balancing parallel computation.

orig.Hochberg logical. To use the standard Hochberg’s procedure.

Value

List with two components:

pow The computed power.

error The sum of the absolute estimated errors for each call to the pmvt (or pmvnorm) function. The number of such calls is given (in the non exchangeable case) by the function complexity. Note that in the exchangeable case, some probabilities are weighted. So an error committed on such a probability is also inflated with the same weight. Note also that this global error does not take into account the signs of the individual errors and is thus most certainly higher than the true error. In other words, you are 99 percent sure that the true power is between ‘pow’ - ‘error’ and ‘pow’ + ‘error’, but it is also probably much closer to ‘pow’, particularly if the complexity is large.

Note

Note that we use critical values involving the D1 term in formula (11) of Romano et al. in order to control strongly the q-FWER. If you want to use the original Hochberg’s procedure, set orig.Hochberg to TRUE. Even for \( q = 1 \), this is a bad idea except when the p-values can be assumed independent.

Results can differ from one time to another because the results of the function pmvt are random. If this is the case, you should consider increasing maxpts and decreasing abseps.
Author(s)

P. Lafaye de Micheaux, B. Liquet and J. Riou

References


See Also

indiv.analysis,
Index

* Bonferroni Procedure
  bonferroni.1m.ssc, 3
  indiv.rm.ssc, 14
  Psirns, 22

* Hochberg Procedure
  Psirmu, 24

* Holm Procedure
  complexity, 4
  df.compute, 5
  matrix.type.compute, 17
  montecarlo, 18
  Psirmd, 21

* data analysis
  global.1m.analysis, 7
  indiv.1m.analysis, 9
  indiv.analysis, 12
  rPowerSampleSize-package, 2

* datasets
  data, 5

* individual procedure
  indiv.1m.analysis, 9
  indiv.1m.ssc, 11
  indiv.analysis, 12
  rPowerSampleSize-package, 2

* multiple testing
  bonferroni.1m.ssc, 3
  complexity, 4
  df.compute, 5
  global.1m.analysis, 7
  global.1m.ssc, 8
  indiv.1m.analysis, 9
  indiv.1m.ssc, 11
  indiv.analysis, 12
  indiv.rm.ssc, 14
  matrix.type.compute, 17
  montecarlo, 18
  Psirmd, 21
  Psirms, 22
  Psirmu, 24
  rPowerSampleSize-package, 2

* multivariate models
  global.1m.analysis, 7
  global.1m.ssc, 8
  rPowerSampleSize-package, 2

* package
  bonferroni.1m.ssc, 3
  complexity, 4
  df.compute, 5
  global.1m.analysis, 7
  global.1m.ssc, 8
  indiv.1m.analysis, 9
  indiv.1m.ssc, 11
  indiv.analysis, 12
  indiv.rm.ssc, 14
  matrix.type.compute, 17
  montecarlo, 18
  Psirmd, 21
  Psirms, 22
  Psirmu, 24
  rPowerSampleSize-package, 2

* r power
  complexity, 4
  df.compute, 5
  indiv.rm.ssc, 14
  matrix.type.compute, 17
  montecarlo, 18
  Psirmd, 21
  Psirms, 22
  Psirmu, 24

* sample size computation
  bonferroni.1m.ssc, 3
  complexity, 4
  df.compute, 5
  global.1m.ssc, 8
  indiv.1m.ssc, 11
  indiv.rm.ssc, 14
  matrix.type.compute, 17
  montecarlo, 18
Psirmd, 21
Psirms, 22
Psirmu, 24

rPowerSampleSize-package, 2

asympt.ssc (indiv.1m.ssc), 11
bonferroni.1m.ssc, 3, 8–10, 12
complexity, 4
data, 5
df.compute, 5
fwer (indiv.1m.analysis), 9
global.1m.analysis, 4, 6, 7, 9, 10, 12
global.1m.ssc, 4, 6, 8, 10, 12
indiv.1m.analysis, 4, 6, 8, 9, 12
indiv.1m.ssc, 4, 6, 8–10, 11
indiv.analysis, 12, 16, 18, 20, 22, 24, 26
indiv.rm.ssc, 14, 14
known.ssc (indiv.1m.ssc), 11
matrix.type.compute, 17
montecarlo, 18, 20
plot.rPower, 20
Psirmd, 21
Psirms, 22
Psirmu, 24

rPowerSampleSize
  (rPowerSampleSize-package), 2
rPowerSampleSize-package, 2
test.indiv (indiv.1m.analysis), 9
unknown.ssc (indiv.1m.ssc), 11