Package 'PowerTOST'

September 30, 2014

Type Package

Title Power and Sample size based on two one-sided t-tests (TOST) for (bio)equivalence studies

Version 1.2-01

Date 2014-10-30

Maintainer Detlew Labes <DetlewLabes@gmx.de>

Description Contains functions to calculate power and sample size for
various study designs used for bioequivalence studies.
See function known.designs() for study designs covered.
Moreover the package contains functions for power and sample size
based on 'expected' power in case of uncertain (estimated) variability.
-----Added are functions for the power and sample size for the ratio of
two means with normally distributed data on the original scale
(based on Fieller's confidence ('fiducial') interval).
-----Contains further functions for power and sample size calculations based on
non-inferiority t-test. This is not a TOST procedure but eventually useful
if the question of 'non-superiority' must be evaluated.
The power and sample size calculations based on non-inferiority test may
also performed via 'expected' power in case of uncertain (estimated) variability.
-----Contains functions power.scABEL() and sampleN.scABEL() to calculate power
and sample size for the BE decision via scaled (widened) BE acceptance
limits based on simulations.
Contains further functions power.RSABE() and sampleN.RSABE() to calculate
power and sample size for the BE decision via reference scaled ABE criterion
according to the FDA procedure based on simulations.
Contains further functions power.NTIDFDA() and sampleN.NTIDFDA() to calculate
power and sample size for the BE decision via the FDA procedure for NTID's
based on simulations. -----Contains functions for power analysis of a sample size plan for ABE
(pa.ABE()), scaled ABE (pa.scABE()) and scaled ABE for NTID's (pa.NTIDFDA())
analysing power if deviating from assumptions of the plan.
-----Contains further functions for power calculations / samplesize estimation
for dose proportionality studies using the Power model.

Depends mvtnorm

ByteCompile yes

License GPL (>= 2)

LazyLoad yes

LazyData yes

Author Detlew Labes [aut, cre],Helmut Schuetz [aut]

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-30 14:41:47

R topics documented:

CI.BE . . . 3
CI.RatioF . . . 4
ct5.1+ct5.2+ct5.3+ct5.4.1 . . . 6
ct9.6.2+ct9.6.6 . . . 7
ct9.6.4+ct9.6.8 . . . 7
ctSJ.VIII.10+ctSJ.VIII.20+ctCW.III . . . 8
CV2se+se2CV+CV2mse+mse2CV . . . 9
CVCL . . . 10
CVfromCI . . . 11
CVp2CV . . . 12
CVpooled . . . 13
exppower.noninf . . . 16
exppower.TOST . . . 17
expsampleN.noninf . . . 19
expsampleN.TOST . . . 21
known.designs . . . 23
OwensQ . . . 24
OwensQOwen . . . 26
OwensT . . . 27
pa.ABE . . . 28
pa.NTIDFDA . . . 30
pa.scABE . . . 32
power.dp . . . 35
power.noninf . . . 36
power.NTIDFDA . . . 38
power.RatioF . . . 40
power.RSABE . . . 42
power.scABEL . . . 44
power.TOST . . . 47
power2.TOST . . . 50
sampleN.dp . . . 52
sampleN.noninf . . . 54

sampleN.NTIDFDA 56
sampleN.RatioF 58
sampleN.RSABE 60
sampleN.scABEL 63
sampleN.TOST 65

Index **69**

`CI.BE` *1-2*alpha confidence interval given point est., CV and n*

Description

Utility function to calculate the 1-2*alpha CI's given point est., CV and n for the various designs covered in this package.

Usage

```
CI.BE(alpha = 0.05, pe, CV, n, design = "2x2", robust = FALSE)
```

Arguments

`alpha`	Type I error probability, significance level. Defaults to 0.05.
`pe`	Point estimator (GMR).
`CV`	Coefficient of variation of error variability as ratio.
`n`	Total number of subjects if a scalar is given. Number of subjects in (sequence) groups if given as vector.
`design`	Character string describing the study design. See `known.designs()` for designs covered in this package.
`robust`	Defaults to FALSE. Set to `TRUE` will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as `n-seq`. See `known.designs()$df2` for designs covered in this package.

Value

Returns the 1-2*alpha confidence interval.
Returns a vector with named elements lower, upper if arguments pe and CV are scalars, else a matrix with columns lower, upper is returned.

Note

The function assumes an evaluation using log-transformation.
The function assumes equal varainces in case of `design="parallel"` and the higher order crossover designs.
The formula implemented covers balanced and unbalanced designs.

If the function vectorizes propperly is not thoroughly tested.

Author(s)

D. Labes

Examples

```
# 90% confidence interval for the 2x2 crossover
# n(total) = 24
CI.BE(pe=0.95, CV=0.3, n=24)
# should give
#    lower     upper
#0.8213465 1.0988055
# same with number of subjects in sequence groups
CI.BE(pe=0.95, CV=0.3, n=c(12, 12))
```

CI.RatioF *1-2*alpha Fieller confidence interval given point est., CV (,CVb) and n*

Description

Utility function to calculate the 1-2*alpha Fieller CI's given point est., CV (, CVb) and n for the parallel group and 2x2 crossover.

Usage

```
CI.RatioF(alpha = 0.025, pe, CV, CVb, n, design = c("2x2", "parallel"))
```

Arguments

alpha	Type I error probability, aka significance level. Defaults here to 0.025 because this function is intended for studies with clinical endpoints.
pe	point estimator (ratio T/R).
CV	Coefficient of variation as ratio. In case of `design="parallel"` this is the CV of the total variability, in case of `design="2x2"` the intra-subject CV.
CVb	CV of the between-subject variability. Only necessary for `design="2x2"`.
n	Total number of subjects if a scalar is given. Number of subjects in (sequence) groups if given as vector.
design	A character string describing the study design. `design="parallel"` or `design="2x2"` allowed for a two-parallel group design or a classical TR/RT crossover design.

Details

The CV(within) and CVb(etween) in case of `design="2x2"` are obtained via an appropriate ANOVA from the error term and from the difference `(MS(subject within sequence)-MS(error))/2`.

Value

Returns the 1-2*alpha confidence interval.

Note

The function assumes an evaluation using un-transformed data.
The function assumes equal varainces in case of `design="parallel"`.
The formula implemented covers balanced and unbalanced designs.

Note that when the mean of the denominator of the ratio is close to zero, confidence intervals might be degenerated and are returned as NA. In that case a warning is issued.

If the function vectorizes propperly is not thoroughly tested.

This function is intended for studies with clinical endpoints. In such studies the 95% confidence intervals are usually used for equivalence testing. Therefore alpha defaults here to 0.025.
See CPMP/EWP/482/99 "Points to consider on switching between superiority and non-inferiority" EMEA, London (2000).

Author(s)

D. Labes

References

Locke C.S.
"An exact confidence interval from untransformed data for the ratio of two formulation means."
J Pharmacokinet Biopharm. 12(6):649-55 (1984)

Hauschke D., Steinijans V. and Pigeot I.
"Bioequivalence Studies in Drug Development"
Chapter 10., John Wiley & Sons, Chichester (2007)

See Also

`CI.BE`, `power.RatioF`

Examples

```
# 95% Fieller CI for the 2x2 crossover
CI.RatioF(pe=1.05,CV=0.3,CVb=0.6, n=24)
```

`ct5.1+ct5.2+ct5.3+ct5.4.1` *Sample size tables for the classical 2x2 crossover*

Description

These data.frames give sample size tables calculated with `sampleN.TOST()` for the 2x2 design.

Details

The data.frame's can be accessed by their names or by `data("name")`.

ct5.1 is Table 5.1 from
Hauschke D., Steinijans V. and Pigeot I.
"Bioequivalence studies in Drug Development"
John Wiley & Sons, Chichester (2007)
Multiplicative model, theta1=0.8, theta2=1.25 (1/theta1), exact

ct5.2 is Table 5.2 from the same source
Multiplicative model, theta1=0.75, theta2=1.3333 (1/theta1), exact

ct5.3 is Table 5.3 from the same source
Multiplicative model, theta1=0.9, theta2=1.1111 (1/theta1), exact

ct5.4.1 is Table 5.4.1 from
Chow S.C., Liu J.P.
"Design and Analysis of Bioavailability and Bioequivalence Studies"
Third edition, CRC Press, Chapman & Hall, Boca Raton (2009)
Additive model, theta1=-0.2, theta2=+0.2 (BE limits 0.80 - 1.20), exact

Note

Scripts for creation of these data.frame's can be found in the \test sub-directory of the package. Comparing the results of that scripts to the corresponding data.frames can be used for validation purposes.

Author(s)

PowerTOST

ct9.6.2+ct9.6.6 *Sample size tables for the 2x2x3 replicate crossover*

Description

These data.frames give sample size tables calculated with `sampleN.TOST()` for the 2x2x3 replicate crossover design (2-treatment-2-sequence-3-period design.

Details

The data.frame's can be accessed by their names or by `data("name")`.

ct9.6.2 is Table 9.6.2 from
Chow S.C., Liu J.P.
"Design and Analysis of Bioavailability and Bioequivalence Studies",
Third edition, CRC Press, Chapman & Hall, Boca Raton (2009)
Additive model, theta1=-0.2, theta2=+0.2 (BE limits 0.80 - 1.20),
approximate power via shifted non-central t-distribution.

ct9.6.6 is Table 9.6.6 from the same reference.
Multiplicative model, theta1=0.8, theta2=1.25 (1/theta1), power via shifted non-central t-distribution.
Attention! Chow and Liu's CV is se (standard error) of residuals.

Note

Scripts for creation of these data.frame's can be found in the \test sub-directory of the package.
Comparing the results of that scripts to the corresponding data.frames can be used for validation purposes.

Author(s)

PowerTOST

ct9.6.4+ct9.6.8 *Sample size tables for the 2x4x4 replicate crossover*

Description

These data.frames give sample size tables calculated with `sampleN.TOST()` for the 2x4x4 replicate crossover design (2-treatment-4-sequence-4-period design).

Details

The data.frame's can be accessed by their names or by `data("name")`.

ct9.6.4 is Table 9.6.4 from
Chow S.C., Liu J.P.
"Design and Analysis of Bioavailability and Bioequivalence Studies",
Third edition, CRC Press, Chapman & Hall, Boca Raton (2009)
Additive model, theta1=-0.2, theta2=+0.2 (BE limits 0.80 - 1.20),
approximate power via shifted non-central t-distribution.

ct9.6.8 is Table 9.6.8 from the same reference.
Multiplicative model, theta1=0.8, theta2=1.25 (1/theta1), power via shifted non-central t-distribution.
Attention! Chow and Liu's CV in case of multiplicative model is se (standard error) of residuals.

Note

Scripts for creation of these data.frame's can be found in the \test sub-directory of the package.
Comparing the results of that scripts to the corresponding data.frames can be used for validation purposes.

Author(s)

PowerTOST

`ctSJ.VIII.10+ctSJ.VIII.20+ctCW.III`
Sample size tables for the parallel group design

Description

These data.frames give sample size tables calculated with `sampleN.TOST()` for the parallel group design (2 groups).

Details

The data.frame's can be accessed by their names or by `data("name")`.

ctSJ.VIII.10 is Table VIII, column 'level of bioequivalence 10%' from
S.A.Julious
"Tutorial in Biostatistics
Sample sizes for clinical trials with Normal data"
'Statistics in Medicine, Vol. 23, 1921-1986 (2004)
Multiplicative model, theta1=0.9, theta2=1.1111 (1/theta1), target power=90%,
power approximate via non-central t-distribution.
Attention! Julious gives sample size per group.

ctSJ.VIII.20 is Table VIII from the same source
column 'level of bioequivalence 20%'
Multiplicative model, theta1=0.8, theta2=1.25 (1/theta1), target power=90%,
power approximate via non-central t.

ctCW.III is Table III from
Chow and Wang
"On Sample Size Calculation in Bioequivalence Trials"
J. Pharmacokin. Biopharm. Vol. 28(2), 155-169 (2001)
Additive model, theta1=-0.2, theta2=+0.2 (BE limits 0.80 - 1.20), exact.

Seems the last reference is not very reliable (compare to the Table in the paper).

Note

Scripts for creation of these data.frame's can be found in the \test sub-directory of the package. Comparing the results of that scripts to the corresponding data.frames can be used for validation purposes.

Author(s)

PowerTOST

`CV2se+se2CV+CV2mse+mse2CV`
Helper functions

Description

Calculates the standard error or the mean squared error from a given CV and vice versa for log-normal data.

Usage

```
CV2se(CV)
se2CV(se)
CV2mse(CV)
mse2CV(mse)
```

Arguments

`CV`	coefficient of variation
`se`	standard error
`mse`	mean squared error

Value

Returns `se = sqrt(log(CV^2+1))`
or `CV = sqrt(exp(se*se)-1)`
or `mse = log(CV^2+1)`
or `CV = sqrt(exp(mse)-1)`

Note

These functions were originally intended for internal use only.
But may be useful for others.

Author(s)

D. Labes

Examples

```
# these functions are one liners:
CV2se <- function(CV) return(sqrt(log(1.0 + CV^2)))
se2CV <- function(se) return(sqrt(exp(se*se)-1))

CV2se(0.3)
# should give: [1] 0.2935604

se2CV(0.2935604)
#[1] 0.3
```

CVCL *Confidence limits of a CV for log-normal data*

Description

The function calculates the 1-alpha confidence limits (either 1-sided or 2-sided) via the chi-squared distribution of the error variance the CV is based on.

Usage

```
CVCL(CV, df, side = c("upper", "lower", "2-sided"), alpha = 0.05)
```

Arguments

`CV`	Coefficient of variation
`df`	degrees of freedom of the CV (error variance)
`side`	Side(s) to calculate the confidence limits for
`alpha`	Type I error probability, aka significance level

Value

Numeric vector of the confidence limits named as 'lower CL' and 'upper CL'.
In case of the one-sided upper confidence limit the 'lower CL' is = 0.
In case of the one-sided lower confidence limit the 'upper CL' is = Inf.

Author(s)

D. Labes

Examples

```
# upper one-sided 95% CL of a CV=0.3
# from a study with df=22 (f.i. a 2x2 crossover with n=24)
# side="upper" is standard if not explicitly given
CVCL(0.3, df=22)
# should give:
# lower CL  upper CL
#0.0000000 0.4075525
```

`CVfromCI` *CV from a given Confidence interval*

Description

Calculates the CV (coefficient of variation) from a known confidence interval of a BE study.
Useful if no CV but the 90% CI was given in literature.

Usage

```
CVfromCI(point, lower, upper, n, design = "2x2", alpha = 0.05, robust=FALSE)
CI2CV(point, lower, upper, n, design = "2x2", alpha = 0.05, robust=FALSE)
```

Arguments

`point`	Point estimator of the BE ratio. The point estimator can be missing. In that case it will be calculated as geometric mean of lower and upper.
`lower`	Lower confidence limit of the BE ratio.
`upper`	Upper confidence limit of the BE ratio.
`n`	Total number of subjects under study.
`design`	Character string describing the study design. See `known.designs()` for designs covered in this package.
`alpha`	Error probability. Set it to `(1-confidence)/2`. Is 0.05 for the usual 90% confidence intervals.
`robust`	With `robust=FALSE` the usual degrees of freedom of the designs are used. With `robust=TRUE` the degrees of freedom for the so-called robust evaluation (df2 in known.designs()) will be used. This may be helpful if the CI was evaluated via mixed model or via intra-subject contrasts (aka Senn's basic estimator).

Details

See Helmut Schuetz lectures at www.bebac.at/lectures.htm for a description of the algebra underlying this function.

Value

Numeric value of the CV as ratio.

Note

The calculations are based on the assumption of evaluation via log-transformed values.
The calculations are further based on a common variance of Test and Reference treatments in replicate crossover studies or parallel group study, respectively.
It is assumed that the sequence groups in a crossover study or the treatment arms in a parallel-group study are balanced.
The estimated CV is conservative (i.e. greater than actually observed) in case of unbalanced studies.

`CI2CV()` is simply an alias to `CVfromCI()`.

Author(s)

D. Labes and H. Schuetz

Examples

```
# Given a 90% confidence interval (without point estimator)
# from a classical 2x2 crossover with 22 subjects
CVfromCI(lower=0.91, upper=1.15, n=22, design="2x2")
# will give
# [1] 0.2279405 i.e a CV ~ 23%
```

CVp2CV *Decompose CV(T) and CV(R) from 'pooled' CV of T/R*

Description

Helper function to calculate CV(T) and CV(R) from a pooled CV(T/R) assuming a ratio of the intra-subject variances.

Usage

```
CVp2CV(CV, ratio = 1.5)
```

Arguments

`CV`	'pooled' CV of T/R.
`ratio`	Ratio of the intra-subject variances s^2(T)/s^2(R). May be a vector.

Details

In case of knowing only the CV(T/R) f.i. from an ordinary cross-over you can calculate the components CV(T) and CV(R) assuming a ratio of the intra-subject variances.
The formula the function is based on:
`log(1.0 + CV^2) = (sWT^2 + sWR^2)/2`
Insert sWT^2 = ratio* sWR^2 and solve for sWR^2.

Value

Returns a numeric vector of the CV values for Test and Reference if only one ratio is given.
Returns a matrix with named columns 'CVwT' and C'VwR' if ratio is given as vector.

Author(s)

D. Labes

Examples

```
CVp2CV(0.4, ratio=2)
# gives
# [1] 0.4677952 0.3225018
```

CVpooled *Pooled CV from several studies*

Description

This function calculates a pooled CV from CV's from several studies.

Usage

```
CVpooled(CVdata, alpha = 0.2, logscale=TRUE, robust = FALSE)
## S3 method for class 'CVp'
print(x, digits=4, verbose=FALSE, ...)
```

Arguments

`CVdata`	A data.frame that must contain the columns `CV`, `n` and `design` where CV are the error CVs from the studies, n the number of subjects and design is a character string describing the study design. See `known.designs()` for designs covered in this package. If the design column is missing the classical 2x2 crossover is assumed for each study. A message is displayed under that circumstances. A data.frame that contains the columns `CV` and giving the degrees of freedom `df` directly is also accepted as `CVdata`.

alpha	Error probability for calculating an upper confidence limit of the pooled CV. Recommended 0.2-0.25 for use in subsequent sample size estimation. See f.i one of H. Schuetz lectures http://bebac.at/lectures/MU2010-CD2.pdf
logscale	Defaults to TRUE. Should the calculations be done for log-transformed data?
robust	Defaults to FALSE. Set to TRUE will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df's are calculated as n-seq. They are also often more appropriate if the CV comes from a 'true' mixed model evaluation (FDA model for average bioequivalence). See known.designs()$df2 for the designs covered in this package.
x	An object of class "CVp".
digits	Number of digits for CV.
verbose	Defaults to FALSE. Prints only the pooled CV and the df. If set to TRUE the upper confidence limit is also printed.
...	More args to print(). None used.

Details

The pooled CV is obtained from the weighted average of the error variances obtained from the CV's of the single studies, weights are the df (degrees of freedom).
If only n is given in the input CVdata, the df's are calculated via the formulas given in known.designs().
If both n and df are given the df column precedes.

If logscale=TRUE the error variances are obtained via function CV2se(). Otherwise the pooled CV is obtained via pooling the CV^2.

Value

A list of class "CVp" with components

CV	value of the pooled CV
df	pooled degrees of freedom
CVupper	upper confidence interval of the pooled CV
alpha	input value

The class "CVp" has a S3 methods print.CVp.

Warning

Pooling of CV's from parallel-group and cross-over designs does not make any sense.
Also the function does not throw an error if you do so.

Note

The calculations for `logscale=FALSE` are not described in the references. They are implemented by analogy to the case via log-transformed data.
The calculations are based on a common variance of Test and Reference formulations in replicate crossover studies or parallel group study, respectively.

Author(s)

D. Labes

References

H. Schuetz lectures about sample size challenges
at http://bebac.at/lectures.htm.

Patterson, Jones
"Bioequivalence and Statistics in Clinical Pharmacology"
Chapter 5.7 "Determining Trial Size"
Chapman & Hall/CRC, Boca Raton 2006

See Also

known.designs, CVfromCI

Examples

```
# some data:
# the values for AUC, study 1 and study 2 are Example 3 of H. Schuetz lecture
CVs <- ("
 PKmetric | CV   | n  |design|source
    AUC   | 0.20 | 24 | 2x2  | study 1
    Cmax  | 0.25 | 24 | 2x2  | study 1
    AUC   | 0.30 | 12 | 2x2  | study 2
    Cmax  | 0.31 | 12 | 2x2  | study 2
    AUC   | 0.25 | 12 | 2x2x4| study 3 (replicate)
")
txtcon <- textConnection(CVs)
CVdata <- read.table(txtcon, header=TRUE, sep="|", strip.white=TRUE, as.is=TRUE)
close(txtcon)

# evaluation of the AUC CV's
CVsAUC <- subset(CVdata,PKmetric=="AUC")
CVpooled(CVsAUC, alpha=0.2, logscale=TRUE)
# df of the 'robust' evaluation
CVpooled(CVsAUC, alpha=0.2, logscale=TRUE, robust=TRUE)
#print also the upper CL, data example 3
CVsAUC3 <- subset(CVsAUC,design != "2x2x4")
print(CVpooled(CVsAUC3, alpha=0.2, robust=TRUE), digits=3, verbose=TRUE)
# will give the output:
#Pooled CV = 0.235 with 32 degrees of freedom (robust df's)
#Upper 80% confidence limit of CV = 0.266
```

exppower.noninf *'Expected' power of non-inferiority test*

Description

Calculates the 'expected' power according to Julious for a variety of study designs used in bioequivalence studies.

Usage

```
exppower.noninf(alpha = 0.025, logscale=TRUE, theta0, margin,
                CV, dfCV, n, design = "2x2", robust=FALSE)
```

Arguments

alpha	Type I error probability, significance level. Defaults here to 0.025.
logscale	Should the data used on log-transformed or on original scale? TRUE or FALSE. Defaults to TRUE.
theta0	'True' or assumed bioequivalence ratio or difference. Typically set to 0.95 (default if missing) if logscale=TRUE. Defaults to -0.05 if logscale=FALSE.
margin	Non-inferiority margin. In case of logscale=TRUE it must be given as ratio, otherwise as diff. Defaults to 0.8 if logscale=TRUE or to -0.2 if logscale=FALSE.
CV	Coefficient of variation as ratio.
dfCV	Degrees of freedom for the CV (error/residual degree of freedom).
n	Number of subjects to be planned (ntotal).
design	Character string describing the study design. See known.designs() for designs covered in this package.
robust	Defaults to FALSE. Set to TRUE will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as n-seq. See known.designs()$df2 for designs covered in this package.

Details

This function calculates the so-called 'expected' power based on formulas according to S.A. Julious. These take into account that usually the CV is not known but estimated from a previous study / studies with an uncertainty. See references.

Value

Value of expected power according to the input.

Author(s)

D. Labes

References

S.A. Julious
"Sample sizes for Clinical Trials"
CRC Press, Chapman & Hall 2010

See Also

expsampleN.noninf, power.noninf, power.TOST

Examples

```
# expected power for non-inferiority test of a 2x2 crossover
# CV 30% known from a pilot study with 12 subjects (-> dfCV=10)
# using all the defaults for other parameters
# should give: [1] 0.6751358
exppower.noninf(CV=0.3, dfCV=10, n=40)

# Compare this to the usual power (CV known, "carved in stone")
# should give: [1] 0.7228685
power.noninf(CV=0.3, n=40)
```

---

exppower.TOST *'Expected' power of TOST procedure*

Description

Calculates the 'expected' power according to Julious for a variety of study designs used in bioequivalence studies.

Usage

```
exppower.TOST(alpha = 0.05, logscale=TRUE, theta0, theta1, theta2,
              CV, dfCV, n, design = "2x2", robust=FALSE)
```

Arguments

alpha	Level of significance. Commonly set to 0.05.
logscale	Should the data used on log-transformed or on original scale? TRUE or FALSE. Defaults to TRUE.
theta0	'True' or assumed bioequivalence ratio or difference. Typically set to 0.95 (default if missing) if logscale=TRUE. Defaults to 0.05 if logscale=FALSE.

theta1	Lower bioequivalence limit as ratio if logscale=TRUE or as difference. Can be missing. Defaults then to 0.8 if logscale=TRUE or to -0.2 if logscale=FALSE.
theta2	Upper bioequivalence limit as ratio if logscale=TRUE or as difference. If not given theta2 will be calculated as 1/theta1 if logscale=TRUE, else as -theta1.
CV	Coefficient of variation as ratio.
dfCV	Degrees of freedom for the CV (error/residual degree of freedom).
n	Number of subjects to be planned (ntotal).
design	Character string describing the study design. See known.designs() for designs covered in this package.
robust	Defaults to FALSE. Set to TRUE will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as n-seq. See known.designs()$df2 for designs covered in this package.

Details

This function calculates the so-called 'expected' power based on S.A. Julious taking into account that usually the CV is not known but estimated from a previous study / studies with an uncertainty. See references.

Value

Value of expected power according to the input.

Note

This function is now also implemented for logscale=FALSE.
Previous versions only had logscale=TRUE implemented with lack of this argument.

Author(s)

D. Labes

References

S.A. Julious, R.J. Owen
"Sample size calculations for clinical studies allowing for uncertainty in variance"
Pharmaceutical Statistics (2006), 5, 29-37

S.A. Julious
"Sample sizes for Clinical Trials"
CRC Press, Chapman & Hall 2010

See Also

expsampleN.TOST, power.TOST

Examples

```
# expected power for a 2x2 crossover
# CV 30% known from a pilot study with 12 subjects (-> dfCV=10)
# using all the defaults for other parameters
# should give: [1] 0.735977
exppower.TOST(CV=0.3, dfCV=10, n=40)

# Compare this to the usual power (CV known, "carved in stone")
# gives: [1] 0.8158453
power.TOST(CV=0.3, n=40)
```

---

expsampleN.noninf *Sample size based on 'expected' power for the non-inferiority test*

Description

Calculates the sample size based on Julious 'expected' power for a variety of study designs used in bioequivalence studies.
See `known.designs()` for the study designs covered.

Usage

```
expsampleN.noninf(alpha = 0.025, targetpower = 0.8, logscale=TRUE,
                  theta0, margin, CV, dfCV, design = "2x2",
                  robust=FALSE, print = TRUE, details = FALSE, imax=100)
```

Arguments

`alpha`	Error probability. Typically set to 0.025 for one-sided test.
`targetpower`	Power to achieve at least. Must be >0 and <1. Typical values are 0.8 or 0.9.
`logscale`	Should the data used on log-transformed or on original scale? TRUE or FALSE. Defaults to TRUE.
`theta0`	'True' or assumed bioequivalence ratio or difference. Maybe missing. Defaults then to 0.95 if `logscale=TRUE` or to -0.05 if `logscale=FALSE`.
`margin`	Non-inferiority margin. In case of `logscale=TRUE` it must be given as ratio, otherwise as diff. Defaults to 0.8 if `logscale=TRUE` or to -0.2 if `logscale=FALSE`.
`CV`	Coefficient of variation as ratio. May be given as vector. Then the CV's were pooled as weighted mean (of s2) with their df (degrees of freedom) as weights.
`dfCV`	Degrees of freedom for the CV's. Must be a vector of same length as CV.
`design`	Character string describing the study design. See `known.designs()` for designs covered in this package.

robust	Defaults to FALSE. With that value the usual degrees of freedom will be used. Set to TRUE will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as n-seq. See known.designs()$df2 for designs covered in this package.
print	If TRUE (default) the function prints its results. If FALSE only a data.frame with the results will be returned.
details	If TRUE the design characteristics and the steps during sample size calculations will be shown. Defaults to FALSE.
imax	Maximum number of steps in sample size search. Defaults to 100. Adaption only in very rare cases needed. Never seen a need for adaption up to now.

Details

The sample size is calculated based on iterative evaluation of 'expected' power via Julious formulas based on non-central t-distribution.
The start value of the sample size search is taken from a large sample approximation.
The sample size is bound to 4 as minimum.

Value

A data.frame with the input values and the result of the sample size estimation.
The "Sample size" column contains the **total** sample size in case of all design implemented.

Author(s)

D. Labes

References

S.A. Julious
"Sample sizes for Clinical Trials"
CRC Press, Chapman & Hall, Boca Raton 2010

See Also

exppower.noninf, expsampleN.TOST

Examples

```
# Classical 2x2 cross-over, target power = 80%, alpha=0.025
# logscale=TRUE, 'non-superiority' margin 125%, assumed true BE ratio = 105%,
# intra-subject CV=30% estimated with 10 df
# using all the defaults
expsampleN.noninf(theta0=1.05, margin=1.25, CV=0.3, dfCV=10)
# -> gives n=56 with achieved expected power 0.807719
# Compare this to the usual sample size with CV known as 'carved in stone'
sampleN.noninf(theta0=1.05, margin=1.25, CV=0.3)
```

```
# More then one CV with corresponding degrees of freedom
# other parameters as above
CVs <- c(0.25, 0.3)
dfs <- c(  22, 10)
expsampleN.noninf(theta0=1.05, margin=1.25, CV=CVs, dfCV=dfs)
# -> gives a pooled CV=0.2664927 with df=32
# and a sample size n=34 with achieved expected power 0.815019
```

`expsampleN.TOST` *Sample size based on 'expected' power*

Description

Calculates the sample size based on Julious 'expected' power for a variety of study designs used in bioequivalence studies. See known.designs() for the study designs covered.

Usage

```
expsampleN.TOST(alpha = 0.05, targetpower = 0.8, logscale=TRUE,
                theta0, theta1, theta2, CV, dfCV, design = "2x2",
                robust=FALSE, print = TRUE, details = FALSE, imax=100)
```

Arguments

`alpha`	Error probability. Typically set to 0.05.
`targetpower`	Power to achieve at least. Must be >0 and <1. Typical values are 0.8 or 0.9.
`logscale`	Should the data used on log-transformed or on original scale? TRUE or FALSE. Defaults to TRUE.
`theta0`	'True' or assumed bioequivalence ratio or difference. Maybe missing. Defaults the to 0.95 if `logscale=TRUE` or to 0.05 if `logscale=FALSE`.
`theta1`	Lower bioequivalence limit as ratio if `logscale=TRUE` or as difference. Can be missing. Defaults then to 0.8 if `logscale=TRUE` or to -0.2 if `logscale=FALSE`.
`theta2`	Upper bioequivalence limit as ratio if `logscale=TRUE` or as difference. If not given `theta2` will be calculated as `1/theta1` if `logscale=TRUE`, else as `-theta1`.
`CV`	Coefficient of variation as ratio. May be given as vector. Then the CV's were pooled as weighted mean with their df=degrees of freedom as weights.
`dfCV`	Degrees of freedom for the CV's. Must be a vector of same length as CV.
`design`	Character string describing the study design. See `known.designs()` for designs covered in this package.
`robust`	Defaults to FALSE. With that value the usual degrees of freedom will be used. Set to `TRUE` will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as `n-seq`. See `known.designs()$df2` for designs covered in this package.

print	If TRUE (default) the function prints its results. If FALSE only a data.frame with the results will be returned.
details	If TRUE the design characteristics and the steps during sample size calculations will be shown. Defaults to FALSE.
imax	Maximum number of steps in sample size search. Defaults to 100. Adaption only in very rare cases needed. Never seen a need for adaption up to now.

Details

The sample size is calculated based on iterative evaluation of 'expected' power via Julious formulas based on non-central t-distribution.
The start value of the sample size search is taken from a large sample approximation.

Value

A data.frame with the input values and the result of the sample size estimation.
The "Sample size" column contains the **total** sample size in case of all design implemented.

Author(s)

D. Labes

References

S.A. Julious, R.J. Owen
"Sample size calculations for clinical studies allowing for uncertainty in variance"
Pharmaceutical Statistics (2006), 5, 29-37

S.A. Julious
"Sample sizes for Clinical Trials"
CRC Press, Chapman & Hall, Boca Raton 2010

S. Senn
"Cross-over Trials in Clinical Research" Second edition
Wiley, Chichester 2002

See Also

exppower.TOST, known.designs, sampleN.TOST

Examples

```
# Classical 2x2 cross-over, target power = 80%,
# BE limits 80 ... 125%, assumed true BE ratio = 95%,
# intra-subject CV=30% estimated with 10 df
# using all the defaults
expsampleN.TOST(CV=0.3, dfCV=10)
# -> gives n=48 with achieved expected power 0.805082
# Compare this to the usual sample size with CV known as 'carved in stone'
```

```
sampleN.TOST(CV=0.3)

# More then one CV with corresponding degrees of freedom
# other parameters as above
CVs <- c(0.25, 0.3)
dfs <- c(  22, 10)
expsampleN.TOST(CV=CVs, dfCV=dfs)
# -> gives a pooled CV=0.2664927 with df=32
# and a sample size n=34 with achieved expected power 0.815019
```

---

known.designs *Show the 'known' designs*

Description

Returns the known study designs for which power and sample size can be calculated within this package.

Usage

```
known.designs()
```

Details

This function is for informal purposes and will be used internal
for obtaining characteristics of the designs used in calculation formulas.

Value

Returns a data.frame with

`no` = number of the design

`design` = character string for identifying the design

`df` = degrees of freedom of the design

`df2` = 'robust' degrees of freedom of the design

`steps` = step width in the iterative sample size estimation

`bk` = so-called design constant in terms of total n

`bkni` = design constant in terms of number of subjects in (sequence) groups

The design character string has to be used in the functions calls for power and sample size.

Note

The design string for higher order crossover designs is named as:
`treatments x sequences x periods` in case of replicate designs and
`treatments x periods` in case of crossover designs for more then 2 treatments with number of sequences equal number of treatments.

The df for the replicate crossover designs are those without carry-over in the model.
Chen, Chow and Liu used models with carry-over, i.e. one df lower than here.

The design constant bk in case of design 2x2x4 is here bk=1.
Chen, Chow and Liu used bk=1.1 due to carry-over in the model.

n is the **total** number of subjects for all designs implemented.
df2 = degrees of freedom for the so-called 'robust' analysis (aka Senn's basic estimator).
These degrees of freedom are often also more appropriate in case of evaluation via a 'true' mixed model (FDA model for replicate designs).

Author(s)

D. Labes

References

K.-W. Chen, S.-C. Chow and G. Liu
"A Note on Sample Size Determination for Bioequivalence Studies with Higher-order Crossover Designs"
J. Pharmacokinetics and Biopharmaceutics, Vol. 25, No. 6, p753-765 (1997)

S. Senn
"Cross-over Trials in Clinical Research"
Second Edition, John Wiley & Sons, Chichester 2002

FDA Guidance for Industry.
"Statistical Approaches to Establishing Bioequivalence"
U.S. Department of Health and Human Services,
Food and Drug Administration,
Center for Drug Evaluation and Research (CDER). January 2001

Examples

```
known.designs()
```

`OwensQ` *Owen's Q-function*

Description

Calculates Owen's Q function.

Usage

```
OwensQ(nu, t, delta, a, b)
```

Arguments

nu	degree of Owen's Q
t	parameter t
delta	parameter delta
a	lower integration limit
b	upper integration limit

Details

Uses `integrate()` from package stats to perform the numerical evaluation of the definite integral in Owen's Q function.
See `../doc/BE_power_sample_size_excerpt.pdf` in the package sub-directory /doc for the definition of Owen's Q and implmentation details.
In case of high delta and/or high upper integration limit b where the implementation via R's function `integrate()` may fail the function `OwensQOwen()` is used.

The arguments to the function must be scalars. No vectors allowed.

Value

Numeric value of Owen's Q-function at given input arguments.

Warning

Since for really large values of nu and the upper integration limit b the integrand is a function which is zero over nearly all its range, the `integrate()` function may fail (see `?integrate`) and `OwensQ()` then returns erroneously 0.
The function now tries to return a value via non-central t-approximation in such cases. This approximation is up to 6 decimals correct as far as tested.
`OwensQ()` issues a warning if the nct-approximation is used.

Note

This function is intended for internal use in the power calculations.
But may be useful for others.

Author(s)

D. Labes

References

Owen, D. B. (1965)
"A Special Case of a Bivariate Non-central t-Distribution"
Biometrika, 52, 437-446.

See Also

OwensQOwen

Examples

```
# This function is mainly intended for internal use.
OwensQ(10,2.5,5,0,2)
#should give [1] 9.388137e-06
OwensQ(10,-2.5,-5,0,2)
#should give [1] 0.05264363
```

OwensQOwen *Owen's Q-function via repeated integration by parts*

Description

This is an implementation of the algorithm given in Owen's original paper (Biometrica 1965) via repeated integration by parts.

Usage

```
OwensQOwen(nu, t, delta, a=0, b)
```

Arguments

nu	degree of Owen's Q
t	parameter t
delta	parameter delta
a	lower integration limit. Only a=0 implemented, other values give an error.
b	upper integration limit

Value

numeric value of Owen's Q function.

Note

The argument a=0 could be dropped but is retained for sake of completeness.

Note

This function is mainly for comparative / validation purposes.
It is used in OwensQ() in case of high nu and/or high upper integration limit where the implementation via R's function integrate() may fail. The implementation needs OwensT() function.

Author(s)

D. Labes

References

Owen, D.B. (1965)
"A Special Case of a Bivariate Non-central t-Distribution"
Biometrika Vol. 52, p437-446.

See Also

OwensQ, OwensT

Examples

```
# comparison of the results of both implementations
# both should give [1] 0.0731726
OwensQ(2, 2.92, 4.2135, 0, 2.0407)
OwensQOwen(2, 2.92, 4.2135, 0, 2.0407)
```

OwensT *Owen's T-function*

Description

Calculates the definite integral from 0 to a of exp(-0.5*h^2*(1+x^2))/(1+x^2)/(2*pi).

Usage

```
OwensT(h, a)
```

Arguments

h	parameter h
a	upper limit of integration

Details

The function is simply implemented via stats function `integrate()`.

Value

Numeric value of the definite integral.

Note

This function is only needed in `OwensQOwen()`.
But may be useful for others.

Author(s)

D. Labes

See Also

OwensQOwen, OwensQ

Examples

```
OwensT(2.5,0.75)
# should give [1] 0.002986697
```

---

pa.ABE *Power analysis for average bioequivalence (ABE)*

Description

An analysis tool for exploration/visualization of the impact of expected values (CV, GMR, reduced sample size due to drop-outs) on power of BE decision via ABE if these values deviate from the ones assumed in planning the sample size of the study.

Usage

```
pa.ABE(CV, theta0 = 0.95, targetpower = 0.8, minpower = 0.7, design = "2x2", ...)
## S3 method for class 'pwrA'
print(x, digits=4, plotit=TRUE, ...)
## S3 method for class 'pwrA'
plot(x, pct=TRUE, cols=c("blue", "red"), ...)
```

Arguments

CV	Coefficient of variation as ratio. In case of cross-over studies this is the within-subject CV, in case of a parallel-group design the CV of the total variability.
theta0	'True' or assumed bioequivalence ratio. Often named GMR. Must be given as ratio.
targetpower	Power to achieve at least in sample size estimation. Must be >0 and <1. Typical values are 0.8 or 0.9. Defaults to 0.8. Note that targetpower < 0.5 doesn't make many sense.
minpower	Minimum acceptable power to have if deviating from assumptions for sample size plan. Has to be < as targetpower. Defaults to 0.7. minpower or targetpower <0.5 does'nt make many sense.
design	Character string describing the study design. See known.designs() for designs covered in this package.

...	More arguments to pass to power2.TOST(). F. i. alpha, theta1, theta2 or robust if other values then the defaults for these arguments are needed. See man page of power2.TOST().
	More arguments passed to the S3 methods. Here currently ignored.
	Additional arguments of the S3 methods:
x	Object of class 'pwrA'.
digits	Digits for rounding power in printing. The '...' argument is currently ignored in print().
plotit	If set to TRUE, the default, the print method calls plot(x) if R is running interactively.
pct	If set to TRUE (the default) scales CV and power in percent in plot(). Else they will be given as ratios, the usual standard in PowerTOST.
cols	Colors for the plots. cols[1] gives the color for plotting points with power>targetpower. From targetpower toward minpower the color changes gradually to cols[2].

Details

Power calculations are done via power2.TOST() and calculations of CV and theta0 which gave a power=minpower are derived via R base uniroot(). While one of the parameters (CV, GMR, n) is varied, the respective two others are kept constant. The tool shows the relative impact of single parameters on power.
The tool takes a minimum of 12 subjects as demanded in most BE guidances into account.

It should be kept in mind that this is **not** a substitute for the "Sensitivity Analysis" recommended in ICH-E9. In a real study a combination of all effects occurs simultaneously. It's upto **you** to decide on reasonable combinations and analyze the power of them.

Value

Returns a list with class "pwrA" with the components

plan	A data.frame with the result of the sample size estimation. See output of sampleN.TOST().
paCV	A data.frame with value pairs CV, pwr for impact of deviations from CV.
paGMR	A data.frame with value pairs theta0, pwr for impact of deviations from theta0 (GMR).
paN	A data.frame with value pairs N, pwr for impact of deviations from planned N (drop-outs).
method	Method of BE decision. Here fix = "ABE".
minpower	Minimum acceptable power.

The class 'pwrA' has the S3 methods print() and plot(). See pa.scABE for usage.

Note

The code of deviations from planned sample size tries to keep the degree of imbalance as low as possible between (sequence) groups. This results in a lesser drop of power than more extreme drop-out patterns.

Author(s)

Idea and original code by Helmut Schuetz
with modifications by D. Labes to use PowerTOST infrastructure.

References

See http://forum.bebac.at/mix_entry.php?id=13353.

See Also

power2.TOST, known.designs, pa.scABE

Examples

```
# using the defaults
# design="2x2", targetpower=0.8, minpower=0.7, theta0/GMR=0.95
# BE acceptance range from defaults of sampleN.TOST() 0.8 ... 1.25
# print & plot implicit
pa.ABE(CV=0.2)
# print & plot
## Not run:
res <- pa.ABE(CV=0.2)
print(res, plotit=FALSE) # print only
plot(res)
## End(Not run)
```

---

pa.NTIDFDA *Power analysis for scaled ABE for NTID according to FDA*

Description

An analysis tool for exploration/visualization of the impact of expected values (CV, theta0, reduced sample size due to drop-outs) on power of BE decision via scABE for narrow therapeutic drugs (NTID) if these values deviate from the ones assumed in planning the sample size of the study.
The only implemented design is the full replicate design "2x2x4" according to the FDA Warfarin guidance.

Usage

```
pa.NTIDFDA(CV, theta0=0.975, targetpower=0.8, minpower=0.7, ...)
```

Arguments

CV	Coefficient of variation of the intra-subject vriabilities of Test and Reference as ratio. Here only the case CVwT=CVwR is implemented, i.e. CV has to be a scalar.
theta0	'True' or assumed bioequivalence ratio. Often named GMR. Must be given as ratio. Defaults here to 0.975.
targetpower	Power to achieve at least in sample size estimation. Must be >0 and <1. Typical values are 0.8 or 0.9. Defaults to 0.8. Note that targetpower < 0.5 doesn't make many sense.
minpower	Minimum acceptable power to have if deviating from assumptions for sample size plan. Has to be lower than targetpower. Defaults to 0.7. Note that minpower < 0.5 doesn't make many sense.
...	More arguments to pass to power.NTIDFDA(). F. i. alpha, theta1, theta2 or nsims if other values then the defaults for these arguments are needed. See man page of power.NTIDFDA().

Details

Power calculations are done via power.NTIDFDA() and calculations of CV and theta0 which result in minpower are derived via uniroot().
While one of the parameters (CV, GMR, n) is varied, the respective two others are kept constant.
The tool shows the relative impact of single parameters on power.
The tool takes a minimum of 12 subjects into account as demanded in most BE guidances.

It should be kept in mind that this is **not** a substitute for the "Sensitivity Analysis" recommended in ICH-E9. In a real study a combination of all effects occurs simultaneously. It's upto **you** to decide on reasonable combinations and analyze the power of them.

Value

Returns a list with class 'pwrA' with the components

plan	A data.frame with the result of the sample size estimation. See output of sampleN.NTIDFDA()
.	
paCV	A data.frame with value pairs CV, pwr for impact of deviations from CV.
paGMR	A data.frame with value pairs theta0, pwr for impact of deviations from theta0 (GMR).
paN	A data.frame with value pairs N, pwr for impact of deviations from planned N (drop-outs).
method	Method of BE decision. Here fix = "NTID FDA".
regulator	Here fix = "FDA".
minpower	Minimum acceptable power from the call of the function.

The class 'pwrA' has the S3 methods print() and plot(). See pa.ABE for usage.

Warning

Be extremly carefull if your sample size plan has extremly small CV near or below 0.05 (5%). Adapt in that case your expected true ratio (`theta0`) to values nearer to 1 to not run into errors and/or long execution times.

Note

The code for impact of deviations from planned sample size tries to keep the degree of imbalance as low as possible between (sequence) groups. This results in a lesser drop of power than more extreme drop-out patterns.

Author(s)

D. Labes
according to code by Helmut Schuetz for pa.ABE() and pa.scABE()

References

FDA "Draft Guidance on Warfarin Sodium"
Recommended Dec 2012
http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM201283.pdf

See Also

power.NTIDFDA, pa.ABE, pa.scABE print.pwrA, plot.pwrA

Examples

```
# using the defaults:
# targetpower=0.8, minpower=0.7, theta0/GMR=0.975
# BE acceptance range from defaults of sampleN.NTIDFDA() 0.8 ... 1.25
# 1E5 sims in power.NTIDFDA()
# not run due to timing policy of CRAN for examples
# may run some ten seconds or more
## Not run:
plot(pa.NTIDFDA(CV=0.1))
## End(Not run)
```

---

pa.scABE *Power analysis for scaled average bioequivalence (scABE)*

Description

An analysis tool for exploration/visualization of the impact of expected values (CV, theta0, reduced sample size due to drop-outs) on power of BE decision via scABE (for highly variable drugs) if these values deviate from the ones assumed in planning the sample size of the study.

Usage

```
pa.scABE(CV, theta0=0.9, targetpower=0.8, minpower=0.7,
         design=c("2x3x3", "2x2x4", "2x2x3"), regulator=c("EMA", "FDA"), ...)
```

Arguments

CV	Coefficient of variation of the intra-subject vriability as ratio. Here only the case CVwT=CVwR is implemented, i.e. CV has to be a scalar.
theta0	'True' or assumed bioequivalence ratio. Often named GMR. Must be given as ratio. Defaults to 0.9 here since HVD have a greater scatter in point estimator of T/R.
targetpower	Power to achieve at least in sample size estimation. Must be >0 and <1. Typical values are 0.8 or 0.9. Defaults to 0.8. targetpower < 0.5 doesn't make many sense.
minpower	Minimum acceptable power to have if deviating from assumptions for sample size plan. Has to be < as targetpower. Defaults to 0.7.
design	Character string describing the study design. Defaults to 2x3x3, the partial replicate design (TRR/RTR/RRT).
regulator	Character string describing the scaled ABE method recommended by the regulatory bodies EMA or FDA. Defaults to EMA, method of scaled (widened) bioequivalence limits.
...	More arguments to pass to power.scABEL() or power.RSABE(). F. i. alpha, theta1, theta2 or nsims if other values then the defaults for these arguments are needed. See man pages of power.scABEL() or power.RSABE().

Details

Power calculations are done via power.scABEL() or power.RSABE() and calculations of CV and theta0 which result in minpower are derived via uniroot().
While one of the parameters (CV, GMR, n) is varied, the respective two others are kept constant.
The tool shows the relative impact of single parameters on power.
The tool takes a minimum of 12 subjects as demanded in most BE guidances into account.

It should be kept in mind that this is **not** a substitute for the "Sensitivity Analysis" recommended in ICH-E9. In a real study a combination of all effects occurs simultaneously. It's upto **you** to decide on reasonable combinations and analyze the power of them.

Value

Returns a list with class 'pwrA' with the components

plan	A data.frame with the result of the sample size estimation. See output of sampleN.scABEL() or sampleN.RSABE()

.

paCV	A data.frame with value pairs CV, pwr for impact of deviations from CV.
paGMR	A data.frame with value pairs theta0, pwr for impact of deviations from theta0 (GMR).
paN	A data.frame with value pairs N, pwr for impact of deviations from planned N (drop-outs).
method	Method of BE decision. Here fix = "scABE".
regulator	"EMA" or "FDA".
minpower	Minimum acceptable power from the call of the function.

The class 'pwrA' has the S3 methods print() and plot(). See pa.ABE for usage.

Note

The code for impact of deviations from planned sample size tries to keep the degree of imbalance as low as possible between (sequence) groups. This results in a lesser drop of power than more extreme drop-out patterns.

Author(s)

Idea and original code by Helmut Schuetz
with modifications by D. Labes to use PowerTOST infrastructure.

References

See http://forum.bebac.at/mix_entry.php?id=13376.

See Also

power.scABEL, power.RSABE, print.pwrA, plot.pwrA, pa.ABE

Examples

```
# using the defaults:
# design="2x3x3", targetpower=0.8, minpower=0.7, theta0/GMR=0.90
# BE acceptance range from defaults of sampleN.scABEL() 0.8 ... 1.25
# 1E5 sims in power.scABEL()
# not run due to timing policy of CRAN, may run some ten seconds
## Not run:
# implicit print & plot
pa.scABE(CV=0.4)
## End(Not run)
```

power.dp *Power of dose-proportionality studies evaluated via Power model*

Description

Calculates the power of dose-proportionality studies using the Power model for cossover (Latin square) or parallel group designs via a confidence interval equivalence criterion.

Usage

```
power.dp(alpha = 0.05, CV, doses, n, beta0 = 1, theta1 = 0.8, theta2 = 1/theta1,
         design = c("crossover", "parallel"))
```

Arguments

alpha	Type 1 error. Usually taken as 0.05.
CV	Coefficient of variation. Is intra-subject CV for design="crossover" and CV of total variability in case of design="parallel"
doses	Vector of dose values. At least 2 doses have to be given.
n	Number of subjects. Is total number if given as scalar, else number of subjects in the (sequence) groups. In the latter case the length of n vector has to be the same as length of vector doses.
beta0	'True' slope of power model.
theta1	Lower acceptance limit for the ratio of dose normalized means (Rdmn). Transformes into slope acceptance range as described under item beta0.
theta2	Upper acceptance limit for the ratio of dose normalized means (Rdmn).
design	Crossover design (default) or parallel group design. Crossover design means Latin square design with number of doses as dimension.

Details

The power calculations are based on TOST for testing equivalence of the slope = 1.
Power is calculated via non-central t approximation only.

Value

Value of power according to the input arguments.

Warning

This function is 'experimental' only since there is a pending discussion of differences to the SAS code given in Patterson, Jones.
Additionally the function is not thorougly tested yet.

Author(s)

D. Labes

References

Patterson, Jones
"Bioequivalence and Statistics in Clinical Pharmacology"
Chapman & Hall/CRC, Boca Raton, 2006, page 239

Hummel J, McKendrick S, Brindley C, and R French
"Exploratory assessment of dose proportionality: review of current approaches and proposal for a practical criterion"
Pharmaceut Statist 8(1), 38-49 (2009)

Sethuraman VS, Leonov S, Squassante L, Mitchell TR, Hale MD
"Sample size calculation for the Power Model for dose proportionality studies"
Pharm Stat. Vol. 6(1):35-41 (2007)

See Also

sampleN.dp

Examples

```
# using all the defaults, i.e. crossover design, alpha=0.05, beta0=1
# theta1=0.8, theta2=1.25
power.dp(CV=0.2, doses=c(1, 2, 8), n=12)
```

`power.noninf` *Power of the one-sided non-inferiority t-test*

Description

Function calculates of the power of the one-sided non-inferiority t-test for normal or log-normal distributed data.

Usage

```
power.noninf(alpha = 0.025, logscale = TRUE, margin, theta0, CV, n,
             design = "2x2", robust = FALSE)
```

Arguments

`alpha`	Type I error probability, significance level. Defaults here to 0.025.
`logscale`	Should the data used on log-transformed or on original scale? TRUE or FALSE. Defaults to TRUE.

margin	Non-inferiority margin. In case of logscale=TRUE it must be given as ratio, otherwise as diff. to 1. Defaults to 0.8 if logscale=TRUE or to -0.2 if logscale=FALSE.
theta0	'True' or assumed bioequivalence ratio or difference. In case of logscale=TRUE it must be given as ratio, otherwise as difference to 1. See examples. Defaults to 0.95 if logscale=TRUE or to -0.05 if logscale=FALSE.
CV	Coefficient of variation as ratio. In case of cross-over studies this is the within-subject CV, in case of a parallel-group design the CV of the total variability.
n	Number of subjects under study.
design	Character string describing the study design. See known.designs for designs covered in this package.
robust	Defaults to FALSE. With that value the usual degrees of freedom will be used. Set to TRUE will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as n-seq. See known.designs()$df2 for designs covered in this package. Has only effect for higher-order crossover designs.

Details

The power is calculated via non-central t-distribution.

Value

Value of power according to the input arguments.

Warning

The function does not vectorize if design is a vector.
The function vectorize properly if n or CV or theta0 are vectors.
Other vector input is not tested yet.

Note

This function does not rely on TOST but may be useful in planning BE studies if the question is not equivalence but 'non-superiority'.
Hint: Evaluation of Fluctuation in the EMA MR NfG (1999) between modified release formulation and immediate release product.

Author(s)

D. Labes

References

S.A. Julious
"TUTORIAL IN BIOSTATISTICS
Sample sizes for clinical trials with Normal data"
Statist. Med. 2004; 23: 1921-1986

See Also

known.designs, sampleN.noninf

Examples

```
# using all the defaults: margin=0.8, theta0=0.95, alpha=0.025
# log-transformed, design="2x2"
# should give: 0.4916748
power.noninf(CV=0.3, n=24)
#
# vectorized answer
# should give: [1] 0.7146648 0.8227931 0.8926036
power.noninf(alpha=0.05, CV=0.3, n=c(30,40,50), design="2x2")
```

---

power.NTIDFDA *(Empirical) Power for BE decision via FDA method for NTID's*

Description

This function performs the power calculation of the BE decision via the FDA method for narrow therapeutic index drugs (NTID's) by simulations. The study design is the full replicate design 2x2x4 (2-treatment-2-sequence-4period design)

Usage

```
power.NTIDFDA(alpha = 0.05, theta1, theta2, theta0, CV, n, nsims = 1e+05,
              details = FALSE, setseed = TRUE)
```

Arguments

alpha	Type I error probability, significance level. Conventionally mostly set to 0.05.
theta1	Conventional lower ABE limit to be applied in the FDA procedure. Defaults to 0.8 if not given explicitly.
theta2	Conventional upper ABE limit to be applied in the FDA procedure. Defaults to 1.25 if not given explicitly.
theta0	'True' or assumed bioequivalence ratio. Attention! Defaults here to 0.975 if not given explicitly. The value was chosen nearer to 1 because the potency (contents) settings for NTID's are tightened by the FDA.

CV	Coefficient(s) of variation as ratio. If length(CV) = 1 the same CV is assumed for Test and Reference. If length(CV) = 2 the CV for Test must be given in CV[1] and for Reference in CV[2].
n	Number of subjects under study. May be given as vector. In that case it is assumed that n contains the number of subjects per sequence groups. If n is given as single number (total sample size) and this number is not divisible by the number of sequences of the design an unbalanced design is assumed. A corresponding message is thrown showing the numbers of subjects in the sequence groups.
nsims	Number of simulations to be performed to obtain the empirical power. Defaults to 100 000 = 1e+5.
details	If set to TRUE the computational time is shown as well as the components for the BE decision. p(BE-ABE) is the simulated probability for the conventional ABE test. p(BE-SABEc) is the probability that the 95% CI of the ABE criterion is <0. p(BE-sratio) is the probability that the ratio of sWT/sWR is < 2.5.
setseed	Simulations are dependent on the starting point of the (pseudo) random number generator. To avoid differences in power for different runs a set.seed(123456) is issued if setseed=TRUE, the default.

Details

The linearized scaled ABE criterion is calculated according to the SAS code given in the FDA Warfarine guidance. For deciding BE the study must pass that criterion, the conventional ABE test and additional the test that the ratio of sWT/sWR is < 2.5.

The simulations are done via the distributional properties of the statistical quantities necessary for deciding BE based on these method.
Details can be found in a document "Implementation_scaledABE_sims" located in the doc subdirectory of the package.

Value

Returns the value of the empirical power.

Note

The FDA method is described for the ABE limits 0.8 ... 1.25 only. Setting theta1, theta2 to other values may not be reasonable and is not tested.

Author(s)

Detlew Labes

References

FDA "Draft Guidance on Warfarin Sodium"
Recommended Dec 2012
http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM201283.pdf

See Also

sampleN.NTIDFDA

Examples

```
# using the defaults:
# GMR=0.975, theta1=0.8, theta2=1.25, 100 000 sims
# and a CV of 0.1 (=10%) with 12 subjects, balanced
power.NTIDFDA(CV=0.1, n=12)
# should give a power of 0.62553
```

`power.RatioF` *Power for equivalence of the ratio of two means with normality on original scale*

Description

Calculates the power of the test of equivalence of the ratio of two means with normality on original scale.
This test is based on Fieller's confidence ('fiducial') interval and Sasabuchi's test (again a TOST procedure).

Usage

```
power.RatioF(alpha = 0.025, theta1 = 0.8, theta2, theta0 = 0.95, CV, CVb, n,
             design = "2x2", setseed=TRUE)
```

Arguments

`alpha`	Type I error probability, aka significance level. Defaults here to 0.025 because this function is intended for studies with clinical endpoints.
`theta1`	Lower bioequivalence limit. Typically 0.8 (default).
`theta2`	Upper bioequivalence limit. Typically 1.25. Is set to `1/theta1` if missing.
`theta0`	True ('null') assumed bioequivalence ratio. Typically set to 0.95 for planning.
`CV`	Coefficient of variation as ratio. In case of `design="parallel"` this is the CV of the total variability, in case of `design="2x2"` the intra-subject CV (CVw in the reference).
`CVb`	CV of the between-subject variability. Only necessary for `design="2x2"`.

n	Number of subjects to be planned. n is for both designs implemented the **total** number of subjects.
design	A character string describing the study design. design="parallel" or design="2x2" allowed for a two-parallel group design or a classical TR/RT crossover design.
setseed	If set to TRUE the dependence of the power from the state of the random number generator is avoided. With setseed = FALSE you may see the dependence from the state of the random number generator.

Details

The power is calculated exact using the bivariate non-central t-distribution via function pmvt() from the package mvtnorm.
Due to the calculation method of the used package mvtnorm, randomized Quasi-Monte-Carlo, these probabilities are dependent from the state of the random number generator within the precision of the power. See argument setseed.

Value

Value of power according to the input.

Note

This function is intended for studies with clinical endpoints.
In such studies the 95% confidence intervals are usually used for equivalence testing.
Therefore alpha defaults here to 0.025.
See CPMP/EWP/482/99 "Points to consider on switching between superiority and non-inferiority" EMEA, London (2000).

The formulas given in the references rely on the assumption of equal variances in the two treatment groups for the parallel group design or on assuming equal within-subject and between-subject variabilities for the 2x2 crossover design.

Author(s)

D. Labes

References

Hauschke D., Kieser M., Diletti E. and Burke M.
"Sample size determination for proving equivalence based on the ratio
of two means for normally distributed data"
Stat. Med. 18(1) p93-105 (1999)

Hauschke D., Steinijans V. and Pigeot I.
"Bioequivalence Studies in Drug Development"
Chapter 10., John Wiley & Sons, Chichester (2007)

See Also

sampleN.RatioF

Examples

```
# power for alpha=0.025, ratio0=0.95, theta1=0.8, theta2=1/theta1=1.25
# within-subject CV=0.2, between-subject CV=0.4
# 2x2 crossover study, n=24
# using all the defaults:
power.RatioF(CV=0.2, CVb=0.4, n=24)
# gives [1] 0.7315357
```

---

power.RSABE *(Empirical) Power for BE decision via linearized scaled ABE criterion*

Description

This function performs the power calculation of the BE decision via linearized scaled ABE criterion by simulations.

Usage

```
power.RSABE(alpha = 0.05, theta1, theta2, theta0, CV, n,
            design = c("2x3x3", "2x2x4", "2x2x3"), regulator = c("FDA", "EMA"),
            nsims = 1e+05, details = FALSE, setseed=TRUE)
```

Arguments

alpha	Type I error probability, significance level. Conventionally mostly set to 0.05.
theta1	Conventional lower ABE limit to be applied in the mixed procedure if CVsWR <= CVswitch. Also lower limit for the point estimator constraint. Defaults to 0.8 if not given explicitly.
theta2	Conventional upper ABE limit to be applied in the mixed procedure if CVsWR <= CVswitch. Also upper limit for the point estimator constraint. Defaults to 1.25 if not given explicitly.
theta0	'True' or assumed bioequivalence ratio. Defaults to 0.95 if not given explicitly.
CV	Coefficient(s) of variation as ratio. If `length(CV) = 1` the same CV is assumed for Test and Reference. If `length(CV) = 2` the CV for Test must be given in CV[1] and for Reference in CV[2].
n	Number of subjects under study. May be given as vector. In that case it is assumed that n contains the number of subjects in the sequence groups. If n is given as single number (total sample size) and this number is not divisible

by the number of sequences of the design an unbalanced design is assumed. A corresponding message is thrown showing the numbers of subjects in sequence groups used.
Attention! In case of the 2x2x3 (TRT|RTR) design the order of n's important if given as vector. n[1] is for sequence group 'TRT' and n[2] is for sequence group 'RTR'.

`design` Design of the study to be planned.
2x3x3 is the partial replicate design (TRR|RTR|RRT).
2x2x4 is the full replicate design with 2 sequences and 4 periods.
2x2x3 is the 3-period design with sequences TRT|RTR.
Defaults to `design="2x3x3"`.

`regulator` Regulatory body settings for the scaled ABE criterion.
Defaults to `regulator="FDA"`.
Also the linearized scaled ABE criterion is usually calculated with the FDA constant r_const=log(1.25)/0.25 you can override this behavior to use the EMA setting r_const=0.76 to avoid the discontinuity at CV=30% and be more stringent.

`nsims` Number of simulations to be performed to obtain the empirical power. Defaults to 100 000 = 1e+5.
If simulations are aimed for empirical alpha `nsims=1e+06` is recommended.

`details` If set to `TRUE` the computational time is shown as well as the components for the BE decision.
p(BE-ABE) is the simulated probability for the conventional ABE test. p(BE-SABEc) is the probability that the 95% CI of the ABE criterion is <0.
p(BE-PE) is the probability that the point estimate is within theta1 ... theta2.

`setseed` Simulations are dependent on the starting point of the (pseudo) random number generator. To avoid differences in power for different runs a set.seed() is issued if `setseed=TRUE`, the default.

Details

The linearized scaled ABE criterion is calculated according to the SAS code given in the FDA progesterone guidance.

The simulations are done via the distributional properties of the statistical quantities necessary for deciding BE based on scaled ABE criterion.
Details can be found in a document "Implementation_scaledABE_simsVx.yy.pdf" located in the doc subdirectory of the package.

Value

Returns the value of the empirical power.

Warning

In case of the design 2x2x3 heteroscedasticity (CVwT not equal to CVwR) may lead to poor agreement of the power values compared to those calculated via the 'classical' way of subject data sims if

the design is unbalanced in respect to the number of subjects in the sequence groups. The function therefore issues a warning for that cases.

Author(s)

D. Labes

References

FDA "Draft Guidance on Progesterone"
Recommended Apr 2010; Revised Feb 2011
http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM209294.pdf

Laszlo Tothfalusi and Laszlo Endrenyi
"Sample Sizes for Designing Bioequivalence Studies for Highly Variable Drugs"
J. Pharm. Pharmaceut. Sci. (www.cspsCanada.org) 15(1) 73 - 84, 2011

Tothfalusi L., Endrenyi L. and A. Garcia Arieta
"Evaluation of Bioequivalence for Highly Variable Drugs with Scaled Average Bioequivalence"
Clin. Pharmacokin. 48/11, 725-743 (2009)

See Also

sampleN.RSABE, power.scABEL

Examples

```
# using all the defaults:
# design="2x3x3" -> partial replicate
# ABE limits, PE constraint 0.8-1.25
# true ratio =0.95, 1E+5 simulations
power.RSABE(CV=0.4, n=24)
# should give
# [1] 0.80864
#
# to explore the simulation error due to the state of the
# random number generator
power.RSABE(CV=0.4, n=24, setseed=FALSE)
# will give something like
# [1] 0.8081
```

---

power.scABEL *(Empirical) Power for BE decision via scaled (widened) BE acceptance limits*

Description

This function performs the power calculation of the BE decision via scaled (widened) BE acceptance limits by simulations.

Usage

```
power.scABEL(alpha = 0.05, theta1, theta2, theta0, CV, n,
             design = c("2x3x3", "2x2x4", "2x2x3"), regulator = c("EMA", "FDA"),
              nsims = 1e+05, details = FALSE, setseed = TRUE)
```

Arguments

alpha	Type I error probability, significance level. Conventionally mostly set to 0.05.
theta1	Conventional lower ABE limit to be applied in the mixed procedure if CVsWR <= CVswitch. Also lower limit for the point estimator constraint. Defaults to 0.8 if not given explicitly.
theta2	Conventional upper ABE limit to be applied in the mixed procedure if CVsWR <= CVswitch. Also upper limit for the point estimator constraint. Defaults to 1.25 if not given explicitly.
theta0	'True' or assumed bioequivalence ratio. Defaults to 0.95 if not given explicitly.
CV	Coefficient(s) of variation as ratio. If `length(CV) = 1` the same CV is assumed for Test and Reference. If `length(CV) = 2` the CV for Test must be given in CV[1] and for Reference in CV[2].
n	Number of subjects under study. May be given as vector. In that case it is assumed that n contains the number of subjects in the sequence groups. If n is given as single number (total sample size) and this number is not divisible by the number of sequences of the design an unbalanced design is assumed. A corresponding message is thrown showing the numbers of subjects in sequence groups. Attention! In case of the 2x2x3 (TRT|RTR) design the order of n's is important if given as vector. n[1] is for sequence group 'TRT' and n[2] is for sequence group 'RTR'.
design	Design of the study to be planned. 2x3x3 is the partial replicate design (TRR/RTR/RRT). 2x2x4 is the full replicate design with 2 sequences and 4 periods. 2x2x3 is the 3-period design with sequences TRT|RTR. Defaults to `design="2x3x3"`.
regulator	Regulatory body settings for the widening of the BE acceptance limits. Defaults to `regulator="EMA"`
nsims	Number of simulations to be performed to obtain the empirical power. Defaults to 100 000 = 1e+05. If simulations are aimed for empirical alpha `nsims=1e+06` is recommended.
details	If set to `TRUE` the computational time is shown as well as the components for the BE decision. p(BE-wABEL) is the probability that the CI is within widened limits. p(BE-PE) is the probability that the point estimate is within theta1 ... theta2. p(BE-ABE) is the simulated probability for the conventional ABE test.

setseed — Simulations are dependent on the starting point of the (pseudo) random number generator. To avoid differences in power for different runs a `set.seed()` is issued if `setseed=TRUE`, the default.

Details

The methods rely on the analysis of log-transformed data, i.e. assume a log-normal distribution on the original scale.

The widened BE acceptance limits will be calculated by the formula
`[lBEL,uBEL] =exp(-+ r_const*sWR)`
with r_const the regulatory constant and sWR the standard deviation of the within subjects variability of the Reference. r_const=0.76 is used in case of `regulator="EMA"` and in case of `regulator="FDA"` r_const=0.89257...=log(1.25)/0.25. If the CVwR of the Reference is < CVswitch=0.3 the conventional ABE limits apply (mixed procedure). In case of `regulator="EMA"` a cap is placed on the widened limits if CVwr>0.5, i.e. the widened limits are held at value calculated for CVwR=0.5.

The simulations are done via the distributional properties of the statistical quantities necessary for deciding BE based on widened ABEL.
For more details see a document "Implementation_scaledABE_simsVx.yy.pdf" in the /doc subdirectory of the package.

Value

Returns the value of the 'empirical' power.

Warning

Cross-validation of the simulations as implemented here and via the 'classical' subject data simulation have shown somewhat unsatisfactory results for the 2x3x3 design if the variabilities for Test and Reference are different.
The function therefore gives a warning if calculations with different CVwT, CVwR are requested for the 2x3x3 partial replicate design.
For more details see the above mentioned document "Implementation_scaledABE_simsVy.xx.pdf"

Note

In case of `regulator="FDA"` the (empirical) power is only approximate since the BE decision method is not exactly what is expected by the FDA. But the two Laszlos state that the scABEL method should be 'operational equivalent' to the FDA method.
To get the power for the FDA favored method via linearized scaled ABE criterion use function `power.RSABE()`.

Author(s)

D. Labes

References

Laszlo Tothfalusi and Laszlo Endrenyi
"Sample Sizes for Designing Bioequivalence Studies for Highly Variable Drugs"
J. Pharm. Pharmaceut. Sci. (www.cspsCanada.org) 15(1) 73 - 84, 2011

See Also

sampleN.scABEL, power.RSABE

Examples

```
# using all the defaults:
# design="2x3x3", EMA regulatory settings
# PE constraint 0.8-1.25, cap on widening if CV>0.5
# true ratio =0.95, 1E+6 simulations
power.scABEL(CV=0.4, n=29)
# should give:
# Unbalanced design. n(i)=10/10/9 assumed.
# [1] 0.82854
# with details=TRUE to view the computational time
power.scABEL(CV=0.5, n=54, theta0=1.15, details=TRUE)
# should give (times may differ depending on your machine).
# 1e+05 sims. Time elapsed (sec):
#    user  system elapsed
#    0.09    0.00    0.10
# p(BE-ABE)= 0.27542 ; p(BE-wABEL)= 0.82078 ; p(BE-PE)= 0.85385
# 0.81727
```

---

power.TOST *Power of the classical TOST procedure*

Description

Calculates the exact or approximate power of the two-one-sided t-tests (TOST) procedure for various study designs used in BE studies.

Usage

```
power.TOST(alpha = 0.05, logscale = TRUE, theta1, theta2, theta0, CV, n,
           design = "2x2", method="exact", robust=FALSE)
```

Arguments

alpha	Type I error probability, significance level. Conventionally mostly set to 0.05.
logscale	Should the data used on log-transformed or on original scale? TRUE or FALSE. Defaults to TRUE.

theta1	Lower bioequivalence limit. In case of logscale=TRUE it is given as ratio, otherwise as diff. to 1. Defaults to 0.8 if logscale=TRUE or to -0.2 if logscale=FALSE.
theta2	Upper bioequivalence limit. If not given theta2 will be calculated as 1/theta1 if logscale=TRUE or as -theta1 if logscale=FALSE.
theta0	'True' or assumed bioequivalence ratio. In case of logscale=TRUE it must be given as ratio, otherwise as difference to 1. See examples. Defaults to 0.95 if logscale=TRUE or to 0.05 if logscale=FALSE
CV	Coefficient of variation as ratio. In case of cross-over studies this is the within-subject CV, in case of a parallel-group design the CV of the total variability.
n	Number of subjects under study.
design	Character string describing the study design. See known.designs() for designs covered in this package.
method	Defaults to "exact" in which case the calculation is done based on formulas with Owen's Q. The exact calculation can also be choosen with method="owenq" Approximate calculations can be choosen via method="noncentral" or method="nct" for the approximation using the non-central t-distribution or via method="central" or method="shifted" for the approximation via 'shifted' central t-distribution. The strings for method may be abbreviated.
robust	Defaults to FALSE. With that value the usual degrees of freedom will be used. Set to TRUE will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as n-seq. See known.designs()$df2 for designs covered in this package. Has only effect for higher-order crossover designs.

Details

The exact calculations of power are based on Owen's Q-function.
The approximate power is implemented via non-central t-distribution or via 'shifted' central t-distribution.

The formulas used assume balanced studies, i.e. equal number of subjects in the (sequence) groups.
If the design is imbalanced consider function power2.TOST() instead.

In case of parallel group design and higher order crossover designs (replicate crossover or crossover with more than two treatments) the calculations are based on the assumption of equal variances for Test and Reference products under consideration.

The formulas for the paired means 'design' do not take a correlation parameter into account.
They are solely based on the paired t-test (TOST of differences = zero).

Value

Value of power according to the input arguments.

Warning

The function does not vectorize if design is a vector.
The function vectorize properly if n or CV or theta0 are vectors.
Other vector input is not tested yet.

Note

Of course it is highly recommended to use the default `method="exact"` :-)).
There is no reason beside testing and comparative purposes to use an approximation if the exact method is available.

Author(s)

D. Labes

References

Phillips, K. F. (1990)
"Power of the Two One-Sided Tests Procedure in Bioequivalence"
Journal of Pharmacokinetics and Biopharmaceutics, 18, 137-144.

Diletti D., Hauschke D., and Steinijans V. W. (1991)
"Sample Size Determination for Bioequivalence Assessment by Means of Confidence Intervals"
Int. J. of Clinical Pharmacology, Therapy and Toxicology, 29, 1-8

See here for a short description:
../doc/BE_power_sample_size_excerpt.pdf.

See Also

sampleN.TOST, known.designs, power2.TOST

Examples

```
# power for the 2x2 cross-over design with 24 subjects and CV 25%
# using all the other default values
# should give: [1] 0.7391155
power.TOST(CV=0.25, n=24)
```

power2.TOST *Power of the classical TOST procedure with unbalanced (sequence) groups*

Description

Calculates the exact or approximate power of the two-one-sided t-tests procedure for various study designs used in BE studies in case of unbalanced number of subjects in the (sequence) groups.

Usage

```
power2.TOST(alpha = 0.05, logscale = TRUE, theta1, theta2, theta0, CV, n,
           design = "2x2", method="exact", robust=FALSE)
```

Arguments

alpha	Type I error probability, significance level. Conventionally mostly set to 0.05.
logscale	Should the data used on log-transformed or on original scale? TRUE or FALSE. Defaults to TRUE.
theta1	Lower bioequivalence limit. In case of logscale=TRUE it is given as ratio, otherwise as diff. to 1. Defaults to 0.8 if logscale=TRUE or to -0.2 if logscale=FALSE.
theta2	Upper bioequivalence limit. If not given theta2 will be calculated as 1/theta1 if logscale=TRUE or as -theta1 if logscale=FALSE.
theta0	'True' or assumed bioequivalence ratio. In case of logscale=TRUE it must be given as ratio, otherwise as difference to 1. See examples. Defaults to 0.95 if logscale=TRUE or to 0.05 if logscale=FALSE
CV	Coefficient of variation as ratio. In case of cross-over studies this is the within-subject CV, in case of a parallel-group design the CV of the total variability.
n	Number of subjects in the (sequence) groups under study. Must be a vector of length = (sequence) groups.
design	Character string describing the study design. See known.designs() for designs covered in this package.
method	Defaults to "exact" in which case the calculation is done based on formulas with Owen's Q. The exact calculation can also be chosen with method="owenq". Approximate calculations can be chosen via method="noncentral" or method="nct" for the approximation using the non-central t-distribution or via method="central" or method="shifted" for the approximation via 'shifted' central t-distribution. The strings for method may be abbreviated.

`robust` Defaults to FALSE. With that value the usual degrees of freedom will be used.
Set to `TRUE` will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as `n-seq`.
See `known.designs()$df2` for designs covered in this package.
Has only effect for higher-order crossover designs.

Details

The exact calculations of power are based on Owen's Q-function.
The approximate power is implemented via non-central t-distribution or via 'shifted' central t-distribution.

In case of parallel group design and higher order crossover designs (replicate crossover or crossover with more than two treatments) the calculations are based on the assumption of equal variances for Test and Reference products under consideration.

Value

Value of power according to the input arguments.

Note

Using this function in case of `design="paired"` does'nt make many sense because we can't have imbalance here.

Of course it is highly recommended to use the default `method="exact"` :-)).
There is no reason beside testing and comparative purposes to use an approximation if the exact method is available at no extra costs.

Author(s)

D. Labes

References

Phillips, K. F. (1990)
"Power of the Two One-Sided Tests Procedure in Bioequivalence"
Journal of Pharmacokinetics and Biopharmaceutics, 18, 137-144.

Diletti D., Hauschke D., and Steinijans V. W. (1991)
"Sample Size Determination for Bioequivalence Assessment by Means of Confidence Intervals"
Int. J. of Clinical Pharmacology, Therapy and Toxicology, 29, 1-8

See Also

`power.TOST`, `known.designs`

Examples

```
# power for the 2x2 cross-over design with 24 subjects balanced,
# CV=25% using all the other default values
# should give: [1] 0.7391155
power2.TOST(CV=0.25, n=c(12,12))
# the same result
power.TOST(CV=0.25, n=24)

# power for the 2x2 cross-over study with 24 subjects, same CV
# and 2 drop-outs in the same sequence group
# should give: [1] 0.6912935
power2.TOST(CV=0.25, n=c(10,12))
# not the same compared to
power.TOST(CV=0.25,n=22)
power2.TOST(CV=0.25,n=c(11,11))
# both should give: [1] 0.6953401
```

`sampleN.dp` *Sample size estimation of dose-proportionality studies evaluated via Power model*

Description

Performes a sample size estimation for dose-proportionality studies using the Power model for cossover (Latin square) or parallel group designs via a confidence interval equivalence criterion.

Usage

```
sampleN.dp(alpha = 0.05, CV, doses, targetpower = 0.8, beta0 = 1, theta1 = 0.8,
           theta2 = 1/theta1, design = c("crossover", "parallel"),
           print = TRUE, details = FALSE, imax = 100)
```

Arguments

`alpha`	Type 1 error. Usually taken as 0.05.
`CV`	Coefficient of variation. Is intra-subject CV for `design="crossover"` and CV of total variability in case of `design="parallel"`
`doses`	Vector of dose values under study. At least 2 doses have to be given.
`targetpower`	Power to achieve at least. Must be >0 and <1. Typical values are 0.8 or 0.9.
`beta0`	'True' slope of power model. Defaults to 1 also a real planing should be done with some deviation to 1, but within slope acceptance range according to `1+log(theta1)/log(rd)` and `1+log(theta2)/log(rd)` were rd is the ratio of highest to lowest dose. Otherwise the function issues an error.
`theta1`	Lower acceptance limit for the ratio of dose normalized means (Rdmn). Transformes into slope acceptance range as described under item `beta0`.

theta2	Upper acceptance limit for the ratio of dose normalized means (Rdmn).
design	Crossover design (default) or parallel group design. Crossover design means Latin square design with number of doses as dimension.
print	If TRUE (default) the function prints its results. If FALSE only the data.frame with the results will be returned.
details	If TRUE the design characteristics and the steps during sample size calculations will be shown. Defaults to FALSE.
imax	Maximum number of steps in sample size search. Defaults to 100. Adaption only in rare cases if any needed.

Details

The sample size is calculated via iterative evaluation of power.dp().
Start value for the sample size search is taken from a large sample approximation.
The sample size is bound to 2*number of doses as minimum.
Balanced designs are used although this is not absolutely necessary.

Value

A data.frame with the input and results will be returned.
The "Sample size" column contains the total sample size.

Warning

This function is 'experimental' only since there is a pending discussion of differences to the SAS code given in Patterson, Jones.
Additionally the function is not thorougly tested yet.

Author(s)

D. Labes

References

Patterson, Jones
"Bioequivalence and Statistics in Clinical Pharmacology"
Chapman & Hall/CRC, Boca Raton, 2006, page 239

Hummel J, McKendrick S, Brindley C, and R French
"Exploratory assessment of dose proportionality: review of current approaches
and proposal for a practical criterion"
Pharmaceut Statist 8(1), 38-49 (2009)

Sethuraman VS, Leonov S, Squassante L, Mitchell TR, Hale MD
"Sample size calculation for the Power Model for dose proportionality studies"
Pharm Stat. Vol. 6(1):35-41 (2007)

See Also

power.dp

Examples

```
# using all the defaults, i.e. crossover design, alpha=0.05
# theta1=0.8, theta2=1.25
sampleN.dp(CV=0.2, doses=c(1, 2, 8), beta0=1.02)
# should give n=18
```

---

sampleN.noninf *Sample size for the non-inferiority t-test*

Description

Function for calculating the sample size needed to have a pre-specified power for the one-sided non-inferiority t-test for normal or log-normal distributed data.

Usage

```
sampleN.noninf(alpha = 0.025, targetpower = 0.8, logscale = TRUE, margin,
               theta0, CV, design = "2x2", robust = FALSE,
               details = FALSE, print = TRUE, imax=100)
```

Arguments

alpha	Type I error probability, significance level. Defaults here to 0.025.
targetpower	Power to achieve at least. Must be >0 and <1. Typical values are 0.8 or 0.9.
logscale	Should the data used on log-transformed or on original scale? TRUE or FALSE. Defaults to TRUE.
margin	Non-inferiority margin. In case of logscale=TRUE it must be given as ratio, otherwise as diff. to 1. Defaults to 0.8 if logscale=TRUE or to -0.2 if logscale=FALSE.
theta0	'True' or assumed bioequivalence ratio or difference. In case of logscale=TRUE it must be given as ratio, otherwise as difference to 1. See examples. Defaults to 0.95 if logscale=TRUE or to 0.05 if logscale=FALSE
CV	Coefficient of variation as ratio. In case of cross-over studies this is the within-subject CV, in case of a parallel-group design the CV of the total variability.
design	Character string describing the study design. See known.designs for designs covered in this package.

robust	Defaults to FALSE. With that value the usual degrees of freedom will be used. Set to TRUE will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as n-seq. See known.designs()$df2 for designs covered in this package. Has only effect for higher-order crossover designs.
details	If TRUE the design characteristics and the steps during sample size calculations will be shown. Defaults to FALSE.
print	If TRUE (default) the function prints its results. If FALSE only the data.frame with the results will be returned.
imax	Maximum number of steps in sample size search. Defaults to 100. Adaption only in rare cases needed.

Details

The sample size is calculated via iterative evaluation of power.noninf().
Start value for the sample size search is taken from a large sample approximation.
The sample size is bound to 4 as minimum.

Value

A data.frame with the input and results will be returned.
Explore it with str(sampleN.noninf(...)

Warning

The function does not vectorize properly.
If you need sample sizes with varying CV's f.i. use for-loops or the apply-family.

Author(s)

D. Labes

References

S.A. Julious
"TUTORIAL IN BIOSTATISTICS
Sample sizes for clinical trials with Normal data"
Statist. Med. 2004; 23: 1921-1986

See Also

known.designs, power.noninf

Examples

```
# using all the defaults: margin=0.8, theta0=0.95, alpha=0.025
# log-transformed, design="2x2"
sampleN.noninf(CV=0.3)
# should give n=48
#
# 'non-superiority' case, log-transformed data
# with assumed 'true' ratio somewhat above 1
sampleN.noninf(CV=0.3, targetpower=0.9, margin=1.25, theta0=1.05)
# should give n=62
```

`sampleN.NTIDFDA` *Sample size estimation for BE decision via FDA method for narrow therapeutic index drugs (NTID's)*

Description

This function performs the Sample size estimation for the BE decision via FDA method for NTID's based on simulations. The study design is the full replicate design 2x2x4.

Usage

```
sampleN.NTIDFDA(alpha = 0.05, targetpower = 0.8, theta0, theta1, theta2, CV,
                nsims = 1e+05, nstart, print = TRUE, details = TRUE,
                setseed = TRUE)
```

Arguments

`alpha`	Type I error probability. Per convention mostly set to 0.05.
`targetpower`	Power to achieve at least. Must be >0 and <1. Typical values are 0.8 or 0.9.
`theta0`	'True' or assumed bioequivalence ratio. Attention! Defaults here to 0.975 if not given explicitly. The value was chosen nearer to 1 because the potency (contents) settings for NTID's are tightened by the FDA.
`theta1`	Conventional lower ABE limit to be applied in the FDA procedure. Defaults to 0.8 if not given explicitly.
`theta2`	Conventional upper ABE limit to be applied in the FDA procedure. Defaults to 1.25 if not given explicitly.
`CV`	Coefficient(s) of variation as ratio. If `length(CV) = 1` the same CV is assumed for Test and Reference. If `length(CV) = 2` the CV for Test must be given in CV[1] and for Reference in CV[2].
`nsims`	Number of simulations to be performed to obtain the empirical power. Defaults to 100 000 = 1e+5.

nstart	Set this to a start value for the sample size if a previous run failed. May be missing.
print	If TRUE (default) the function prints its results. If FALSE only the resulting dataframe will be returned.
details	If set to TRUE, the default, the steps during sample size search are shown. Moreover the details of the method settings are printed.
setseed	Simulations are dependent on the starting point of the (pseudo) random number generator. To avoid differences in power values for different runs a set.seed(123456) is issued if setseed=TRUE, the default.

Details

The linearized scaled ABE criterion is calculated according to the SAS code given in the FDA Warfarine guidance. For deciding BE the study must pass that criterion, the conventional ABE test and additional the test that the ratio of sWT/sWR is < 2.5.

The simulations are done via the distributional properties of the statistical quantities necessary for deciding BE based on these method.
Details can be found in a document "Implementation_scaledABE_sims" located in the doc subdirectory of the package.

Value

Returns now a data.frame with the input and sample size results.
The "Sample size" column contains the total sample size.
The "nlast" column contains the last n value. May be useful for re-starting.

Warning

For some input constellations the sample size search may be very time consuming and will eventually also fail since the start values chosen may not really reasonable for them. This applies especially for theta0 values near to the implied scaled (tightened/widened) ABE limits according to exp(+-1.053605*swR).
In case of a failed sample size search you may restart with setting tha argument nstart.
In case of theta0 values outside the implied scaled (tightened/widened) ABE limits no sample size estimation is possible and the function throws an error (f.i. CV=0.04, theta0=0.95).

Note

The design recommended by the FDA is the full replicate design 2x2x4. Only this design is implemented.
The sample size estimation is done only for balanced studies since the break down of the total subject number in case of unbalanced sequence groups is not unique. Moreover the formulas used are only valid for balanced designs.
The FDA method is described for the ABE limits 0.8 ... 1.25 only. Setting theta1, theta2 to other values may not be reasonable and is not tested.

Author(s)

D. Labes

References

FDA "Draft Guidance on Warfarin Sodium"
Recommended Dec 2012
http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM201283.pdf

See Also

power.NTIDFDA

Examples

```
sampleN.NTIDFDA(CV=0.04,theta0=0.975)
# should give
# n=54 with an (empirical) power of 0.809590
#
# Test formulation with lower variability
sampleN.NTIDFDA(CV=c(0.04,0.06),theta0=0.975)
# should give
# n=20 with an (empirical) power of 0.0.814610
```

`sampleN.RatioF` *Sample size for equivalence of the ratio of two means with normality on original scale*

Description

Calculates the necessary sample size to have at least a given power based on Fieller's confidence ('fiducial') interval.

Usage

```
sampleN.RatioF(alpha = 0.025, targetpower = 0.8, theta1 = 0.8, theta2,
               theta0 = 0.95, CV, CVb, design = "2x2",
               print = TRUE, details = FALSE, imax=100, setseed=TRUE)
```

Arguments

`alpha`	Type I error probability. Defaults here to 0.025 because this function is intended for studies with clinical endpoints.
`targetpower`	Power to achieve at least. Must be >0 and <1. Typical values are 0.8 or 0.9.
`theta1`	Lower bioequivalence limit. Typically 0.8 (default).

theta2	Upper bioequivalence limit. Typically 1.25. Is set to 1/theta1 if missing.
theta0	True ('null') assumed bioequivalence ratio. Typically set to 0.95.
CV	Coefficient of variation as ratio. In case of design="parallel" this is the CV of the total variability, in case of design="2x2" the intra-subject CV (CVw in the reference).
CVb	CV of the between-subject variability. Only necessary for design="2x2".
design	A character string describing the study design. design="parallel" or design="2x2" allowed for a two-parallel group design or a classical TR/RT crossover design.
print	If TRUE (default) the function prints its results. If FALSE only a data.frame with the results will be returned.
details	If TRUE the steps during sample size calculations will be shown. Defaults to FALSE.
imax	Maximum number of steps in sample size search. Defaults to 100. Adaption only in rare cases needed.
setseed	If set to TRUE the dependence of the power from the state of the random number generator is avoided.

Details

The sample size is based on exact power calculated using the bivariate non-central t-distribution via function pmvt() from the package mvtnorm.
Due to the calculation method used in package mvtnorm these probabilities are dependent from the state of the random number generator within the precision of the power.

The CV(within) and CVb(etween) in case of design="2x2" are obtained via an appropriate ANOVA from the error term and from the difference (MS(subject within sequence)-MS(error))/2.

Value

A data.frame with the input values and results will be returned.
The sample size n returned is the **total** sample size for **both** designs.

Note

This function is intended for studies with clinical endpoints.
In such studies the 95% confidence intervals are usually used for equivalence testing.
Therefore alpha defaults here to 0.025.
See CPMP/EWP/482/99 "Points to consider on switching between superiority and non-inferiority" EMEA, London (2000).

Author(s)

D. Labes

References

Hauschke D., Kieser M., Diletti E. and Burke M.
"Sample size determination for proving equivalence based on the ratio
of two means for normally distributed data"
Stat. Med. 18(1) p93-105 (1999)

Hauschke D., Steinijans V. and Pigeot I.
"Bioequivalence studies in Drug Development"
Chapter 10., John Wiley & Sons, Chichester (2007)

See Also

power.RatioF

Examples

```
# sample size for a 2x2 cross-over study
# with CVw=0.2, CVb=0.4
# alpha=0.025 (95% CIs), target power = 80%
# 'true' ratio = 95%, BE acceptance limits 80-125%
# using all the defaults:
sampleN.RatioF(CV=0.2, CVb=0.4)
# gives n=28 with an achieved power of 0.807774
# see Hauschke et.al. (2007) Table 10.3a

# sample size for a 2-group parallel study
# with CV=0.4 (total variability)
# alpha=0.025 (95% CIs), target power = 90%
# 'true' ratio = 90%, BE acceptance limits 75-133.33%
sampleN.RatioF(targetpower=0.9, theta1=0.75, theta0=0.90, CV=0.4, design="parallel")
# gives n=236 with an achieved power of 0.900685
# see Hauschke et.al. (2007) Table 10.2

# a rather strange setting of ratio0! have a look at n.
# it would be better this is not the sample size but your account balance ;-).
sampleN.RatioF(theta0=0.801, CV=0.2, CVb=0.4)
```

---

sampleN.RSABE *Sample size estimation for BE decision via linearized scaled ABE criterion*

Description

This function performs the Sample size estimation for the BE decision via linearized scaled ABE criterion based on simulations.

Usage

```
sampleN.RSABE(alpha = 0.05, targetpower = 0.8, theta0, theta1, theta2, CV,
             design = c("2x3x3", "2x2x4", "2x2x3"), regulator = c("FDA", "EMA"),
              nsims = 1e+05, nstart, print = TRUE, details = TRUE, setseed=TRUE)
```

Arguments

alpha	Type I error probability. Per convention mostly set to 0.05.
targetpower	Power to achieve at least. Must be >0 and <1. Typical values are 0.8 or 0.9.
theta0	'True' or assumed bioequivalence ratio. Defaults to 0.95 if given explicitly.
theta1	Conventional lower ABE limit to be applied in the mixed procedure if CVsWR <= CVswitch. Also Lower limit for the point estimator constraint. Defaults to 0.8 if not given explicitly.
theta2	Conventional upper ABE limit to be applied in the mixed procedure if CVsWR <= CVswitch. Also upper limit for the point estimator constraint. Defaults to 1.25 if not given explicitly.
CV	Coefficient(s) of variation as ratio. If `length(CV) = 1` the same CV is assumed for Test and Reference. If `length(CV) = 2` the CV for Test must be given in CV[1] and for Reference in CV[2].
design	Design of the study to be planned. 2x3x3 is the partial replicate design (TRR|RTR|RRT). 2x2x4 is the full replicate design with 2 sequences and 4 periods. 2x2x3 is the 3-period design with sequences (TRT|RTR). Defaults to `design="2x3x3"`
regulator	Regulatory body settings for the scaled ABE criterion. Defaults to `design="FDA"`. Also the scaled ABE criterion is usually calculated with the FDA constant r_const=log(1.25)/0.25 you can override this behavior to use the EMA setting r_const=0.76 to avoid the discontinuity at CV=30% and be more stringent.
nsims	Number of simulations to be performed to obtain the (empirical) power.
nstart	Set this to a start for the sample size search if a previous run failed. After reworking the start n in version 1.1-05 seldom needed.
print	If `TRUE` (default) the function prints its results. If `FALSE` only the result data.frame will be returned.
details	If set to `TRUE`, the default, the steps during sample size search are shown.
setseed	Simulations are dependent on the starting point of the (pseudo) random number generator. To avoid differences in power for different runs a `set.seed(123456)` is issued if `setseed=TRUE`, the default.

Details

The linearized scaled ABE criterion is calculated according to the SAS code given in the FDA progesterone guidance.

The simulations are done via the distributional properties of the statistical quantities necessary for deciding BE based on scaled ABE. For more details see a document "Implementation_scaledABE_simsVx.yy.pdf" in the doc subdirectory of the package.

Value

Returns now a data.frame with the input and sample size results.
The "Sample size" column contains the total sample size.
The "nlast" column contains the last n value. May be useful for restarting.

Warning

The sample size estimation for theta0 >1.2 and <0.85 may be very time consuming and will eventually also fail since the start values chosen are not really reasonable in that ranges. This is especially true in the range about CV = 0.3 and regulatory constant according to FDA.
If you really need sample sizes in that range be prepared to restart the sample size estimation via the argument nstart.
Since the dependence of power from n is very flat in the mentioned region you may also consider to adapt the number of simulations not to tap in the simulation error trap.

Note

The sample size estimation is done only for balanced designs since the break down of the total subject number in case of unbalanced sequence groups is not unique. Moreover the formulas used are only for balanced designs.

Author(s)

D. Labes

References

FDA "Draft Guidance on Progesterone"
Recommended Apr 2010; Revised Feb 2011
http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/ Guidances/UCM209294.pdf

Laszlo Tothfalusi and Laszlo Endrenyi
"Sample Sizes for Designing Bioequivalence Studies for Highly Variable Drugs"
J. Pharm. Pharmaceut. Sci. (www.cspsCanada.org) 15(1) 73 - 84, 2011

Tothfalusi L., Endrenyi L. and A. Garcia Arieta
"Evaluation of Bioequivalence for Highly Variable Drugs with Scaled Average Bioequivalence"
Clin. Pharmacokin. 48/11, 725-743 (2009)

See Also

power.RSABE, power.scABEL

Examples

```
# using all the defaults:
# design=2x3x3 (partial replicate design), theta0=0.95,
# ABE limits, PE constraint 0.8 - 1.25
# targetpower=80%, alpha=0.05, 1E5 sims
sampleN.RSABE(CV=0.3)
# results in a sample size n=27, power=0.84132
```

`sampleN.scABEL` *Sample size estimation for BE decision via scaled (widened) BE acceptance limits*

Description

This function performs the Sample size estimation for the BE decision via scaled (widened) BE acceptance limits based on simulations.

Usage

```
sampleN.scABEL(alpha = 0.05, targetpower = 0.8, theta0, theta1, theta2, CV,
             design = c("2x3x3", "2x2x4", "2x2x3"), regulator = c("EMA", "FDA"),
             nsims = 1e+05, nstart, print = TRUE, details = TRUE, setseed = TRUE)
```

Arguments

`alpha`	Type I error probability. Per convention mostly set to 0.05.
`targetpower`	Power to achieve at least. Must be >0 and <1. Typical values are 0.8 or 0.9.
`theta0`	'True' or assumed bioequivalence ratio. Defaults to 0.95 if given explicitly.
`theta1`	Conventional lower ABE limit to be applied in the mixed procedure if CVsWR <= CVswitch. Also Lower limit for the point estimator constraint. Defaults to 0.8 if not given explicitly.
`theta2`	Conventional upper ABE limit to be applied in the mixed procedure if CVsWR <= CVswitch. Also upper limit for the point estimator constraint. Defaults to 1.25 if not given explicitly.
`CV`	Coefficient(s) of variation as ratio. If `length(CV) = 1` the same CV is assumed for Test and Reference. If `length(CV) = 2` the CV for Test must be given in CV[1] and for Reference in CV[2].
`design`	Design of the study to be planned. 2x3x3 is the partial replicate design (TRR|RTR|RRT). 2x2x3 is the 3-period replicate design (TRT|RTR). 2x2x4 is the full replicate design with 2 sequences and 4 periods. Defaults to `design="2x3x3"`

regulator	Regulatory body settings for the widening of the BE acceptance limits. Defaults to design="EMA".
nsims	Number of simulations to be performed to obtain the (empirical) power. The default value 100 000 = 1e+5 is usually sufficient. Consider to rise this value if theta0<=0.85 or >=1.25. But see the warning section.
nstart	Set this to a start for the sample size search if a previous run failed. After reworking the start n in version 1.1-05 seldom needed.
print	If TRUE (default) the function prints its results.
details	If set to TRUE, the default, the steps during sample size search are shown.
setseed	Simulations are dependent on the starting point of the (pseudo) random number generator. To avoid differences in power for different runs a set.seed(123456) is issued if setseed=TRUE, the default.

Details

The simulations are done via the distributional properties of the statistical quantities necessary for deciding BE based on widened ABEL. For more details see a document in the doc subdirectory of the package.

Value

Returns now a data.frame with the input and sample size results.
The "Sample size" column contains the total sample size.
The "nlast" column contains the last n value. May be useful for restarting.

Warning

The sample size estimation for very extreme theta0 (<0.83 or >1.21) may be very time consuming and will eventually also fail since the start values chosen are not really reasonable in that ranges. This is especially true in the range around CV = 0.3 and regulatory constant according to FDA.
If you really need sample sizes in that range be prepared to restart the sample size estimation via the argument nstart.
Since the dependence of power from n is very flat in the mentioned region you may also consider to adapt the number of simulations not to tap in the simulation error trap.

See also the Warning section of the function power.scABEL() concerning the power value agreement to those obtained from simulations via subject data.

Note

We are doing the sample size estimation only for balanced designs since the break down of the total subject number in case of unbalanced sequence groups is not unique. Moreover the formulas used are only for balanced designs.
In case of regulator="FDA" the sample size is only approximate since the BE decision method is not exactly what is expected by the FDA. But the two Laszlo's state that the scABEL method should be 'operational' equivalent to the FDA method. Thus the sample size should be comparable.
Consider in case of regulator="FDA" to use the function sampleN.RSABE().

Author(s)

D. Labes

References

Laszlo Tothfalusi and Laszlo Endrenyi
"Sample Sizes for Designing Bioequivalence Studies for Highly Variable Drugs"
J. Pharm. Pharmaceut. Sci. (www.cspsCanada.org) 15(1) 73 - 84, 2011

See Also

power.scABEL, power.RSABE, sampleN.RSABE

Examples

```
# using all the defaults:
# partial replicate design, targetpower=80%,
# true assumed ratio = 0.95, 1E+5 simulated studies
# ABE limits, PE constraint 0.8 - 1.25
# EMA regulatory settings
sampleN.scABEL(CV=0.3)
# results in a sample size n=27, power=0.82566

# for the full replicate design, target power = 90%
# true assumed ratio = 0.9, FDA regulatory settings
sampleN.scABEL(CV=0.4, targetpower=0.9, theta0=0.9, design="2x2x4", regulator="FDA")
# should result in a sample size n=30, power=0.9074
```

---

sampleN.TOST *Sample size based on power of TOST*

Description

Calculates the necessary sample size to have at least a given power.

Usage

```
sampleN.TOST(alpha = 0.05, targetpower = 0.8, logscale = TRUE,
             theta0, theta1, theta2, CV, design = "2x2", method="exact",
             robust=FALSE, print = TRUE, details = FALSE, imax=100)
```

Arguments

alpha	Type I error probability. Per convention mostly set to 0.05.
targetpower	Power to achieve at least. Must be >0 and <1. Typical values are 0.8 or 0.9.

logscale	Should the data used on log-transformed or on original scale? TRUE or FALSE. Defaults to TRUE.
theta0	'True' or assumed bioequivalence ratio. In case of logscale=TRUE it must be given as ratio, otherwise as difference to 1. See examples. Defaults to 0.95 if logscale=TRUE or to 0.05 if logscale=FALSE.
theta1	Lower bioequivalence limit. In case of logscale=TRUE it is given as ratio, otherwise as diff. to 1. Defaults to 0.8 if logscale=TRUE or to -0.2 if logscale=FALSE.
theta2	Upper bioequivalence limit. If not given theta2 will be calculated as 1/theta1 if logscale=TRUE or as -theta1 if logscale=FALSE.
CV	Coefficient of variation as ratio.
design	Character string describing the study design. See known.designs() for designs covered in this package.
method	Defaults to "exact" in which case the calculation is done based on formulas with Owen's Q. The exact calculation can also be choosen with method="owenq". Approximate calculations can be choosen via method="noncentral" or method="nct" for the approximation using the non-central t-distribution or via method="central" or method="shifted" for the approximation via 'shifted' central t-distribution. The strings for method may be abbreviated.
robust	Defaults to FALSE. With that value the usual degrees of freedom will be used. Set to TRUE will use the degrees of freedom according to the 'robust' evaluation (aka Senn's basic estimator). These df are calculated as n-seq. See known.designs()$df2 for designs covered in this package. Has only effect for higher-order crossover designs.
print	If TRUE (default) the function prints its results. If FALSE only the data.frame with the results will be returned.
details	If TRUE the design characteristics and the steps during sample size calculations will be shown. Defaults to FALSE.
imax	Maximum number of steps in sample size search. Defaults to 100. Adaption only in rare cases needed.

Details

The sample size is calculated via iterative evaluation of power.TOST().
Start value for the sample size search is taken from a large sample approximation.
The sample size is bound to 4 as minimum.

Value

A data.frame with the input and results will be returned.
The "Sample size" column contains the total sample size.

Warning

The function does not vectorize properly.
If you need sample sizes with varying CV's f.i. use for-loops or the apply-family.

Note

Of course it is highly recommended to use the default `method="exact"` :-)).
There is no reason beside testing and comparative purposes to use an approximation if the exact method is available at no extra costs.

Author(s)

D. Labes

References

Phillips, K. F. (1990)
"Power of the Two One-Sided Tests Procedure in Bioequivalence"
Journal of Pharmacokinetics and Biopharmaceutics, 18, 137-144.

Diletti, D., Hauschke, D., and Steinijans, V. W. (1991)
"Sample Size Determination for Bioequivalence Assessment
by Means of Confidence Intervals"
Int. J. of Clinical Pharmacology, Therapy and Toxicology, 29 (1), 1-8 (1991)
30 Suppl.No.1, S51-58 (1992)

Diletti, D., Hauschke, D., and Steinijans, V. W. (1992)
"Sample size determination : Extended tables for the multiplicative model
and bioequivalence ranges of 0.9 to 1.11 and 0.7 to 1.43"
Int. J. of Clinical Pharmacology, Therapy and Toxicology, 30 Suppl.No.1, S59-62

See here (R_HOME/library/PowerTOST/doc) for a short description:
../doc/BE_power_sample_size_excerpt.pdf.

See Also

power.TOST, known.designs

Examples

```
# Exact calculation for a classical 2x2 cross-over (TR/RT),
# BE limits 80 ... 125%, assumed true BE ratio 0.95, intra-subject CV=30%,
# using all the default values
# should give n=40 power=0.815845
sampleN.TOST(CV=0.3)

# Exact calculation for a parallel group design
# evaluation on the original (untransformed) scale
# BE limits 80 ... 120% = -20% ... +20% of reference,
# assumed true BE ratio 0.95% = -5% to reference mean,
# total CV=20%
# should give n=48 (total) power=0.815435
```

```
sampleN.TOST(logscale=FALSE, theta1=-0.2, theta0=-0.05, CV=0.2, design="parallel")

# A rather strange setting of theta0! Have a look at n.
# It would be better this is not the sample size but the running total
# of my bank account. But the first million is the hardest ;-).
sampleN.TOST(CV=0.2, theta0=0.8005, theta1=0.8)
```

Index

CI.BE, 3, *5*
CI.RatioF, 4
CI2CV *(*CVfromCI*)*, 11
ct5.1 *(*ct5.1+ct5.2+ct5.3+ct5.4.1*)*, 6
ct5.1+ct5.2+ct5.3+ct5.4.1, 6
ct5.2 *(*ct5.1+ct5.2+ct5.3+ct5.4.1*)*, 6
ct5.3 *(*ct5.1+ct5.2+ct5.3+ct5.4.1*)*, 6
ct5.4.1 *(*ct5.1+ct5.2+ct5.3+ct5.4.1*)*, 6
ct9.6.2 *(*ct9.6.2+ct9.6.6*)*, 7
ct9.6.2+ct9.6.6, 7
ct9.6.4 *(*ct9.6.4+ct9.6.8*)*, 7
ct9.6.4+ct9.6.8, 7
ct9.6.6 *(*ct9.6.2+ct9.6.6*)*, 7
ct9.6.8 *(*ct9.6.4+ct9.6.8*)*, 7
ctCW.III
*(*ctSJ.VIII.10+ctSJ.VIII.20+ctCW.III*)*, 8
ctSJ.VIII.10
*(*ctSJ.VIII.10+ctSJ.VIII.20+ctCW.III*)*, 8
ctSJ.VIII.10+ctSJ.VIII.20+ctCW.III, 8
ctSJ.VIII.20
*(*ctSJ.VIII.10+ctSJ.VIII.20+ctCW.III*)*, 8
CV2mse *(*CV2se+se2CV+CV2mse+mse2CV*)*, 9
CV2se *(*CV2se+se2CV+CV2mse+mse2CV*)*, 9
CV2se+se2CV+CV2mse+mse2CV, 9
CVCL, 10
CVfromCI, 11, *15*
CVp2CV, 12
CVpooled, 13

data2x2 *(*ct5.1+ct5.2+ct5.3+ct5.4.1*)*, 6
data2x2x3 *(*ct9.6.2+ct9.6.6*)*, 7
data2x4x4 *(*ct9.6.4+ct9.6.8*)*, 7
data_parallel
*(*ctSJ.VIII.10+ctSJ.VIII.20+ctCW.III*)*, 8

exppower.noninf, 16, *20*
exppower.TOST, 17, *22*
expsampleN.noninf, *17*, 19
expsampleN.TOST, *18*, *20*, 21

known.designs, *15*, *22*, 23, *30*, *37*, *38*, *49*, *51*, *54*, *55*, *67*

mse2CV *(*CV2se+se2CV+CV2mse+mse2CV*)*, 9

OwensQ, 24, *27*, *28*
OwensQOwen, *26*, 26, *28*
OwensT, *27*, 27

pa.ABE, 28, *31*, *32*, *34*
pa.NTIDFDA, 30
pa.scABE, *29*, *30*, *32*, 32
plot.pwrA, *32*, *34*
plot.pwrA *(*pa.ABE*)*, 28
power.dp, 35, *54*
power.noninf, *17*, 36, *55*
power.NTIDFDA, *32*, 38, *58*
power.RatioF, *5*, 40, *60*
power.RSABE, *34*, 42, *47*, *62*, *65*
power.scABEL, *34*, *44*, 44, *62*, *65*
power.TOST, *17*, *18*, 47, *51*, *67*
power2.TOST, *30*, *49*, 50
print.CVp *(*CVpooled*)*, 13
print.pwrA, *32*, *34*
print.pwrA *(*pa.ABE*)*, 28

sampleN.dp, *36*, 52
sampleN.noninf, *38*, 54
sampleN.NTIDFDA, *40*, 56
sampleN.RatioF, *42*, 58
sampleN.RSABE, *44*, 60, *65*
sampleN.scABEL, *47*, 63
sampleN.TOST, *22*, *49*, 65
se2CV *(*CV2se+se2CV+CV2mse+mse2CV*)*, 9