

# Package 'PowerTOST'

April 5, 2012

**Type** Package

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

**Version** 0.9-8

**Date** 2012-04-05

**Author** Detlew Labes

**Maintainer** D. 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.

**Depends** stats, mvtnorm

**License** GPL (>= 2)

**LazyLoad** yes

**LazyData** yes

**Repository** CRAN

**Date/Publication** 2012-04-05 06:27:16

## R topics documented:

PowerTOST-package . . . . .	2
ct5.1+ct5.2+ct5.3+ct5.4.1 . . . . .	4
ct9.6.2+ct9.6.6 . . . . .	5
ct9.6.4+ct9.6.8 . . . . .	6
ctSJ.VIII.10+ctSJ.VIII.20+ctCW.III . . . . .	6
CV2se+se2CV . . . . .	7
CVfromCI . . . . .	8
CVpooled . . . . .	10
exppower.noninf . . . . .	12
exppower.TOST . . . . .	14
expsampleN.noninf . . . . .	15
expsampleN.TOST . . . . .	17
known.designs . . . . .	19
OwensQ . . . . .	21
OwensQOwen . . . . .	22
OwensT . . . . .	23
power.noninf . . . . .	24
power.RatioF . . . . .	26
power.TOST . . . . .	28
power2.TOST . . . . .	30
sampleN.noninf . . . . .	32
sampleN.RatioF . . . . .	34
sampleN.TOST . . . . .	36
<b>Index</b>	<b>40</b>

---

PowerTOST-package	<i>Power and sample size based on two one-sided t-test (TOST) procedure for bioequivalence studies</i>
-------------------	--

---

### 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).

These functions are intended for studies with clinical endpoints.

Contains further functions for power and sample size calculations based on non-inferiority test.

This is not a TOST procedure (but rather OOST ;-)) but eventually useful if the question of 'non-superiority' must be evaluated within a BE study.

The power and sample size calculations based on non-inferiority test may also be performed via 'expected' power in case of uncertain (estimated) variability.

## Details

Package: PowerTOST  
Type: Package  
Version: 0.9-8  
Date: 2012-04-05  
License: GPL (>=2)  
LazyLoad: yes  
LazyData: yes

Main functions are `sampleN.TOST()` and `power.TOST()` for usual power and sample size calculations.

If you prefer sample size based on 'expected' power see the functions `expsampleN.TOST()` and `exppower.TOST()`.

The main functions for equivalence of the ratio of means with normality on the original scale are `power.RatioF()` and `sampleN.RatioF()`.

The functions for calculating power and sample size for the non-inferiority case are `power.noninf()` and `sampleN.noninf()`.

The functions for calculating 'expected' power and sample size for the non-inferiority case are `exppower.noninf()` and `expsampleN.noninf()`.

The package contains further some utility functions (see Index).

## Author(s)

D. Labes

with inspirations from Kem Phillips function `power.equivalence.md()` in package MBESS.

Maintainer: D. Labes <DetlewLabes at gmx.de>

## 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

S.A. Julious, R.J. Owen (2006)

"Sample size calculations for clinical studies allowing for uncertainty in variance"

Pharmaceutical Statistics (2006), 5, 29-37

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

Hauschke D., Kieser M., Diletti E. and Burke M. (1999)  
 "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"  
 John Wiley & Sons, Chichester (2007)  
 Chapter 5 and 10.3

BEBAC forum: categories 'Power/Sample size' and 'R for BE/BA'  
<http://forum.bebac.at>

---

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,  $\theta_1=0.8$ ,  $\theta_2=1.25$  ( $1/\theta_1$ ), exact

ct5.2 is Table 5.2 from the same source  
 Multiplicative model,  $\theta_1=0.75$ ,  $\theta_2=1.3333$  ( $1/\theta_1$ ), exact

ct5.3 is Table 5.3 from the same source  
 Multiplicative model,  $\theta_1=0.9$ ,  $\theta_2=1.1111$  ( $1/\theta_1$ ), 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,  $\theta_1=-0.2$ ,  $\theta_2=+0.2$  (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,  $\theta_1 = -0.2$ ,  $\theta_2 = +0.2$  (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,  $\theta_1 = 0.8$ ,  $\theta_2 = 1.25$  ( $1/\theta_1$ ), 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,  $\theta_1 = -0.2$ ,  $\theta_2 = +0.2$  (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,  $\theta_1 = 0.8$ ,  $\theta_2 = 1.25$  ( $1/\theta_1$ ), 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,  $\theta_1=0.9$ ,  $\theta_2=1.1111$  ( $1/\theta_1$ ), 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,  $\theta_1=0.8$ ,  $\theta_2=1.25$  ( $1/\theta_1$ ), 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,  $\theta_1=-0.2$ ,  $\theta_2=+0.2$  (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

*Helper functions*

---

**Description**

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

**Usage**

CV2se(CV)

se2CV(se)

**Arguments**

CV                    coefficient of variation  
se                    standard error

**Value**

Returns  $se = \sqrt{\log(CV^2+1)}$   
or  $CV = \sqrt{\exp(se*se)-1}$

**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
```

---

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)
```

**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 <code>known.designs()</code> for designs covered in this package.
alpha	Error probability. Set it to $(1-\text{confidence})/2$ . Is 0.05 for the usual 90% confidence intervals.
robust	With <code>robust=FALSE</code> the usual degrees of freedom of the designs are used. With <code>robust=TRUE</code> the degrees of freedom for the so-called robust evaluation ( <code>df2</code> in <code>known.designs()</code> ) 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](http://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.

**Author(s)**

D. Labes

**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%
```

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 <code>known.designs()</code> 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.
alpha	A data.frame that contains the columns CV and giving the degrees of freedom df directly is also accepted as CVdata. 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 <a href="http://bebac.at/lectures/MU2010-CD2.pdf">http://bebac.at/lectures/MU2010-CD2.pdf</a>
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 <code>known.designs()\$df2</code> 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, alpha2 = 0.05, 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.
alpha2	Significance level for the calculation of an upper confidence interval for CV (for informal purposes only).
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, alpha2 = 0.05,
  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.
alpha2	Significance level for the calculation of an upper confidence interval for CV (for informal purposes only).
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

[expower.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 than 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 than 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	<i>Owen's Q-function</i>
--------	--------------------------

---

## 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.

See [../doc/BE\\_power\\_sample\\_size\\_excerpt.pdf](#) in the package sub-directory `/doc` for the definition of Owen's Q.

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

## Value

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

**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.

**Examples**

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

---

OwensQOwen

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

---

**Description**

This is an implementation of the algorithm given in Owen's original paper (Biometrika 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.

**Note**

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

**Note**

This function is only for comparative / validation purposes.  
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  $\exp(-0.5 \cdot h^2 \cdot (1+x^2)) / (1+x^2) / (2 \cdot \pi)$ .

**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()`.

**Author(s)**

D. Labes

**See Also**

[OwensQOwen](#), [OwensQ](#)

**Examples**

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

---

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 <a href="#">known.designs</a> 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 <code>known.designs()\$df2</code> 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.RatioF	<i>Power for equivalence of the ratio of two means with normality on original scale</i>
--------------	---

---

**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")
```

**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 <b>total</b> 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.

**Details**

The power is calculated exact using the bivariate non-central t-distribution via function `pmvt()` from the package `mvtnorm`.

**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] 0.731509
```

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 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.

The formulas used assume balanced studies, i.e. equal number of subjects in the (sequence) groups. 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](#)

**Examples**

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

---

power2.TOST	<i>Power of the classical TOST procedure with unbalanced (sequence) groups</i>
-------------	--

---

**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 <code>known.designs()</code> 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 <code>method="owenq"</code> . Approximate calculations can be chosen via <code>method="noncentral"</code> or <code>method="nct"</code> for the approximation using the non-central t-distribution or via <code>method="central"</code> or <code>method="shifted"</code> 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 <code>known.designs()\$df2</code> 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"` doesn't 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.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 <a href="#">known.designs</a> 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 <code>known.designs()\$df2</code> 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.RatioF	<i>Sample size for equivalence of the ratio of two means with normality on original scale</i>
----------------	---

---

**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)
```

**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.

### Details

The sample size is based on exact power calculated using the bivariate non-central t-distribution via function `pmvt()` from the package `mvtnorm`.  
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  $(MS(\text{subject within sequence}) - MS(\text{error}))/2$ .

### 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 0.807695
# 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 0.900681
# 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.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 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.
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 in case of the crossover designs and the paired means design.

**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=24 (per group) 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

## \*Topic **package**

- PowerTOST-package, 2
- ct5.1 (ct5.1+ct5.2+ct5.3+ct5.4.1), 4
- ct5.1+ct5.2+ct5.3+ct5.4.1, 4
- ct5.2 (ct5.1+ct5.2+ct5.3+ct5.4.1), 4
- ct5.3 (ct5.1+ct5.2+ct5.3+ct5.4.1), 4
- ct5.4.1 (ct5.1+ct5.2+ct5.3+ct5.4.1), 4
- ct9.6.2 (ct9.6.2+ct9.6.6), 5
- ct9.6.2+ct9.6.6, 5
- ct9.6.4 (ct9.6.4+ct9.6.8), 6
- ct9.6.4+ct9.6.8, 6
- ct9.6.6 (ct9.6.2+ct9.6.6), 5
- ct9.6.8 (ct9.6.4+ct9.6.8), 6
- ctCW. III
  - (ctSJ.VIII.10+ctSJ.VIII.20+ctCW. III), 6
- ctSJ.VIII.10
  - (ctSJ.VIII.10+ctSJ.VIII.20+ctCW. III), 6
- ctSJ.VIII.10+ctSJ.VIII.20+ctCW. III, 6
- ctSJ.VIII.20
  - (ctSJ.VIII.10+ctSJ.VIII.20+ctCW. III), 6
- CV2se (CV2se+se2CV), 7
- CV2se+se2CV, 7
- CVfromCI, 8, 11
- CVpooled, 10
- data2x2 (ct5.1+ct5.2+ct5.3+ct5.4.1), 4
- data2x2x3 (ct9.6.2+ct9.6.6), 5
- data2x4x4 (ct9.6.4+ct9.6.8), 6
- data\_parallel
  - (ctSJ.VIII.10+ctSJ.VIII.20+ctCW. III), 6
- exppower.noninf, 12, 17
- exppower.TOST, 14, 19
- expsampleN.noninf, 13, 15
- expsampleN.TOST, 15, 17, 17
- known.designs, 11, 19, 19, 25, 30, 32–34, 38
- OwensQ, 21, 23, 24
- OwensQOwen, 22, 24
- OwensT, 23, 23
- power.noninf, 13, 24, 34
- power.RatioF, 26, 36
- power.TOST, 13, 15, 28, 32, 38
- power2.TOST, 30
- PowerTOST (PowerTOST-package), 2
- PowerTOST-package, 2
- print.CVp (CVpooled), 10
- sampleN.noninf, 25, 32
- sampleN.RatioF, 27, 34
- sampleN.TOST, 19, 30, 36
- se2CV (CV2se+se2CV), 7