Package 'disprofas'

March 24, 2025

Type Package

Title Non-Parametric Dissolution Profile Analysis

Version 0.2.1

Description Similarity of dissolution profiles is assessed using the similarity factor f2 according to the EMA guideline (European Medicines Agency 2010) ``On the investigation of bioequivalence''. Dissolution profiles are regarded as similar if the f2 value is between 50 and 100. For the applicability of the similarity factor f2, the variability between profiles needs to be within certain limits. Often, this constraint is violated. One possibility in this situation is to resample the measured profiles in order to obtain a bootstrap estimate of f2 (Shah et al. (1998) <doi:10.1023/A:1011976615750>). Other alternatives are the model-independent non-parametric multivariate confidence region (MCR) procedure (Tsong et al. (1996) <doi:10.1177/009286159603000427>) or the T2-test for equivalence procedure (Hoffelder (2016) <https://www.ecv.de/suse_item.php?suseId=Z|pi|8430>). Functions for estimation of f1, f2, bootstrap f2, MCR / T2-test for equivalence procedure are implemented.

License GPL (>= 2)

URL https://github.com/piusdahinden/disprofas

BugReports https://github.com/piusdahinden/disprofas/issues

Depends R (>= 4.0)

Imports boot, ggplot2, rlang

Suggests testthat, T2EQ

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Pius Dahinden [aut, cre],
Tillotts Pharma AG [cph, fnd]

Maintainer Pius Dahinden <pius.dahinden@tillotts.com>

Repository CRAN

Date/Publication 2025-03-24 01:00:07 UTC

Contents

bootstrap_f2 *Bootstrap f2*

Description

The function bootstrap_f2() generates rr bootstrap replicates of the similarity factor f_2 based on resampling of complete profiles (nonparametric bootstrap) or on resampling per time point the values between profiles (parametric bootstrap). Estimates of "normal", "basic", "student", "percent" and of "bias-corrected, accelerated" (BCa) percentile intervals are returned.

Usage

```
bootstrap_f2(
  data,
  tcol,
  grouping,
  rand_mode = "complete",
  rr = 999,
  each = 12,
  new_seed = 100,
  confid = 0.9,
  use_ema = "no",
  bounds = c(1, 85),
  nsf = c(1, 2),
  ...
)
```

Arguments

data	A data frame with the dissolution profile data in wide format.
tcol	A vector of indices that specifies the columns in data that contain the % release values. The length of tcol must be three or longer.
grouping	A character string that specifies the column in data that contains the group names (i.e. a factorial variable, e.g., for the differentiation of batches or formulations of a drug product).
rand_mode	A character string that indicates whether complete profiles shall be randomised ("complete", the default) or individual data points ("individual").
rr	An integer that specifies the number of bootstrap replicates. The default is 999.
each	An integer that specifies the number of dissolution profiles to be selected per group per randomisation round. The default is 12.
new_seed	An integer for setting the seed for random number generation. The default is 100.
confid	A numeric value between 0 and 1 that specifies the confidence limit for the calculation of the bootstrap confidence intervals. The default is 0.9.

use_ema — A character string that indicates whether the similarity factor f_2 should be calculated according to the EMA guideline "On the investigation of bioequivalence" ("yes") or not ("no", the default). The default is "no" because the bootstrap f_2 method is one of the possible solutions if the condition concerning the variability between the profiles does not allow the evaluation of f_2 according to the EMA guideline. A third option is "ignore". If use_ema is "yes", the bounds are c(0, 85) per definition. If use_ema is "no", the appropriate profile portion is determined on the basis of the values of the parameter bounds. If use_ema is "ignore", the complete profiles are used as specified by the parameter tcol.

bounds — A numeric vector of the form c(lower, upper) that specifies the "lower" and "upper" limits, respectively, for the % drug release given that use_ema is "no". The default is c(1, 85). Mean % release values of any of the two groups being compared that are smaller than or equal to the lower bound are ignored and only the first mean % release value that is greater than or equal to the upper bound is included while all the subsequent values are ignored. If use_ema is "yes" the bounds are c(0, 85) per definition. If use_ema is "ignore" the bounds are disregarded.

nsf — A vector of positive integers that specify the "number of significant figures" (nsf) of the corresponding values of the bounds parameter. It must thus have the same length as the bounds parameter. Before the % release values are compared with the limits that are specified by the bounds parameter, they are rounded to the corresponding number of significant figures as specified by the nsf parameter.

... — Named parameters of the functions stat.fun(), ran.fun() and boot().

Details

Information on f_2 can be found in at least three FDA guidances and in the guideline of the European Medicines Agency (EMA) "On the investigation of bioequivalence" (EMA 2010). For the assessment of the similarity of dissolution profiles using the similarity factor f_2 according to the EMA guideline the following constraints do apply:

1. A minimum of three time points (without zero) are necessary.
2. The time points should be the same for the two formulations.
3. For every time point and for each formulation at least 12 data points are required.
4. A maximum of one mean value per formulation may be > 85% dissolved.
5. The coefficient of variation (%CV) should be < 20% for the first time point and < 10% from the second to the last time point for any formulation.

Dissolution profiles are regarded as similar if the f_2 value is between 50 and 100.

One often encountered problem is that the %CV constraint cannot be fulfilled. One possibility in this situation is the use of the bootstrap f_2 method (Shah 1998) by which the distribution of f_2 is simulated to obtain an unbiased estimate of the expected value of f_2 and the variability of the underlying distribution. For the f_2 calculation only those parts of the profiles are taken into account where the means (per formulation) are $> d\%$ dissolved (e.g., $d = 1$) and a maximum of one mean value per formulation is $> 85\%$ dissolved. In the literature it is suggested to make use of the lower 90% bias corrected and accelerated (BCa) confidence interval (CI) limit to come to a decision in terms of similarity (Stevens (2015)).

Value

An object of class 'bootstrap_f2' is returned, containing the following list elements:

Boot	An object of class 'boot' with the corresponding components.
Profile.TP	A named numeric vector of the columns in data specified by tcol and depending on the selection of use_ema. Given that the column names contain extractable numeric information, e.g., the testing time points of the dissolution profile, it contains the corresponding numeric values. Elements where no numeric information could be extracted are NA.
L	A vector of the Jackknife leave-one-out-values.
CI	An object of class 'bootci' which contains the intervals.
BCa_CI	The lower and upper limits of the BCa interval calculated by the boot.ci() function from the 'boot' package.
Shah_BCa_CI	The lower and upper limits of the BCa interval calculated according to Shah (Shah 1998).

References

United States Food and Drug Administration (FDA). Guidance for industry: dissolution testing of immediate release solid oral dosage forms. 1997.
https://www.fda.gov/media/70936/download

United States Food and Drug Administration (FDA). Guidance for industry: immediate release solid oral dosage form: scale-up and post-approval changes, chemistry, manufacturing and controls, *in vitro* dissolution testing, and *in vivo* bioequivalence documentation (SUPAC-IR). 1995.
https://www.fda.gov/media/70949/download

European Medicines Agency (EMA), Committee for Medicinal Products for Human Use (CHMP). Guideline on the Investigation of Bioequivalence. 2010; CPMP/EWP/QWP/1401/98 Rev. 1.

Stevens, R. E., Gray, V., Dorantes, A., Gold, L., and Pham, L. Scientific and regulatory standards for assessing product performance using the similarity factor, f_2. *AAPS Journal*. 2015; **17**(2): 301-306.
doi:10.1208/s122480159723y

Shah, V. P., Tsong, Y., Sathe, P., and Liu, J. P. *In vitro* dissolution profile comparison - statistics and analysis of the similarity factor, f_2. *Pharm Res*. 1998; **15**(6): 889-896.
doi:10.1023/A:1011976615750

See Also

boot, boot.ci, mimcr, mztia.

check_point_location *Check point location*

Description

The function check_point_location() checks if points that were found by the gep_by_nera() function sit on specified confidence region bounds (*CRB*) or not. This is necessary because the points found by aid of the "Method of Lagrange Multipliers" (MLM) and "Newton-Raphson" (nera) optimisation may not sit on the *CRB*.

Usage

```
check_point_location(lpt, lhs)
```

Arguments

lpt	A list returned by the gep_by_nera() function.
lhs	A list of the estimates of Hotelling's two-sample T^2 statistic for small samples as returned by the function get_T2_two().

Details

The function check_point_location() checks if points that were found by the gep_by_nera() function sit on specified confidence region bounds (*CRB*) or not. The gep_by_nera() function determines the points on the *CRB* for each of the n_p time points or model parameters by aid of the "Method of Lagrange Multipliers" (MLM) and by "Newton-Raphson" (nera) optimisation, as proposed by Margaret Connolly (Connolly 2000). However, since the points found may not sit on the specified *CRB*, it must be checked if the points returned by the gep_by_nera() function do sit on the *CRB* or not.

Value

The function returns the list that was passed in via the lpt parameter with a modified points.on.crb element, i.e. set as TRUE if the points sit on the *CRB* or FALSE if they do not sit on the *CRB*.

References

Tsong, Y., Hammerstrom, T., Sathe, P.M., and Shah, V.P. Statistical assessment of mean differences between two dissolution data sets. *Drug Inf J*. 1996; **30**: 1105-1112. doi:10.1177/009286159603000427

Connolly, M. SAS(R) IML Code to calculate an upper confidence limit for multivariate statistical distance; 2000; Wyeth Lederle Vaccines, Pearl River, NY. https://analytics.ncsu.edu/sesug/2000/p-902.pdf

See Also

mimcr, gep_by_nera.

Examples

```
# Collecting the required information
time_points <- suppressWarnings(as.numeric(gsub("([^0-9])", "",
                                                colnames(dip1))))
tcol <- which(!is.na(time_points))
b1 <- dip1$type == "R"
tol <- 1e-9

# Hotelling's T2 statistics
l_hs <- get_T2_two(m1 = as.matrix(dip1[b1, tcol]),
                   m2 = as.matrix(dip1[!b1, tcol]),
                   signif = 0.05)

# Calling gep_by_nera()
res <- gep_by_nera(n_p = as.numeric(l_hs[["Parameters"]]["df1"]),
                   kk = as.numeric(l_hs[["Parameters"]]["K"]),
                   mean_diff = l_hs[["means"]][["mean.diff"]],
                   m_vc = l_hs[["S.pool"]],
                   ff_crit = as.numeric(l_hs[["Parameters"]]["F.crit"]),
                   y = rep(1, times = l_hs[["Parameters"]]["df1"] + 1),
                   max_trial = 100, tol = tol)

# Expected result in res[["points.on.crb"]]
# [1] NA

# Check if points lie on the confidence region bounds (CRB)
check_point_location(lpt = res, lhs = l_hs)

# Expected result in res[["points.on.crb"]]
# [1] TRUE
```

---

dip1 *Dissolution data of a reference and a test batch*

Description

A data set containing the dissolution data of one reference batch and one test batch of $n = 6$ tablets each, i.e. the dissolution profiles of the % drug release observed within a period of 120 minutes.

Usage

```
data(dip1)
```

Format

A data frame with 12 observations and 10 variables:

type Factor with levels `R` (Reference) and `T` (Test)

tablet Factor with levels `1` to `6` representing individual tablets

t.5 Numeric of the % release at the 5 minutes testing point

t.10 Numeric of the % release at the 10 minutes testing point

t.15 Numeric of the % release at the 15 minutes testing point

t.20 Numeric of the % release at the 20 minutes testing point

t.30 Numeric of the % release at the 30 minutes testing point

t.60 Numeric of the % release at the 60 minutes testing point

t.90 Numeric of the % release at the 90 minutes testing point

t.120 Numeric of the % release at the 120 minutes testing point

Source

See reference: Example data set shown in Table 1.

References

Tsong, Y., Hammerstrom, T., Sathe, P.M., and Shah, V.P. Statistical assessment of mean differences between two dissolution data sets. *Drug Inf J*. 1996; **30**: 1105-1112.
doi:10.1177/009286159603000427

Examples

```
str(dip1)
```

`dip2` *Dissolution data of one reference batch and five test batches*

Description

A data set containing the dissolution data of one reference batch and five test batches of $n = 12$ tablets each, i.e. the dissolution profiles of the % drug release observed within a period of 180 minutes.

Usage

```
data(dip2)
```

Format

A data frame with 72 observations and 8 variables:

type Factor with levels `Reference` and `Test`

tablet Factor with levels `1` to `12` representing individual tablets

batch Factor with levels `b0`, `b1`, `b2`, `b3`, `b4` and `b5`

t.0 Numeric of the % release at the initial testing point

t.30 Numeric of the % release at the 30 minutes testing point

t.60 Numeric of the % release at the 60 minutes testing point

t.90 Numeric of the % release at the 90 minutes testing point

t.180 Numeric of the % release at the 180 minutes testing point

Source

See reference: Example data set shown in Table 4.

References

Shah, V. P., Tsong, Y., Sathe, P., and Liu, J. P. *In vitro* dissolution profile comparison - statistics and analysis of the similarity factor, f_2. *Pharm Res*. 1998; **15**(6): 889-896.
doi:10.1023/A:1011976615750

Examples

```
str(dip2)
```

---

dip3 *Dissolution data of two different capsule formulations*

Description

A data set containing the dissolution data of one reference batch and one test batch of $n = 12$ capsules each, i.e. the dissolution profiles of the % drug release observed at 15, 20 and 25 minutes.

Usage

```
data(dip3)
```

Format

A data frame with 24 observations and 6 variables:

cap Factor with levels 1 to 12 representing individual capsules

batch Factor with levels `white` and `blue` representing the colours of two different capsule formulations

type Factor with levels `ref` (Reference) and `test` (Test)

x.15 Numeric of the % release at the 15 minutes testing point

x.20 Numeric of the % release at the 20 minutes testing point

x.25 Numeric of the % release at the 25 minutes testing point

Source

See reference: Example data set shown in Table 1. Data set '`ex_data_JoBS`' from package '`T2EQ`'.

References

Hoffelder, T., Goessl, R., and Wellek, S. Multivariate equivalence tests for use in pharmaceutical development. *J Biopharm Stat*. 2015; **25**(3): 417-437.
doi:10.1080/10543406.2014.920344

Examples

```
str(dip3)

if (requireNamespace("T2EQ")) {
library(T2EQ)

  data(ex_data_JoBS, envir = environment())
  str(ex_data_JoBS)
  rm(ex_data_JoBS)
}
```

`dip4` *Dissolution data of two different formulations*

Description

A data set containing the dissolution data of one reference batch and one test batch of $n = 12$ items each, i.e. the dissolution profiles of the % drug release observed at 10, 20 and 30 minutes.

Usage

```
data(dip4)
```

Format

A data frame with 24 observations and 2 variables:

type Factor with levels `ref` (Reference) and `test` (Test)

x.10 Numeric of the % release at the 10 minutes testing point

x.20 Numeric of the % release at the 20 minutes testing point

x.30 Numeric of the % release at the 30 minutes testing point

Source

See reference: Example data set underlying Figure 1. Data set '`ex_data_pharmind`' from package '`T2EQ`'.

References

Hoffelder, T. Highly variable dissolution profiles. Comparison of T^2-test for equivalence and f_2 based methods. *Pharm Ind*. 2016; **78**(4): 587-592.
https://www.ecv.de/suse_item.php?suseId=Z|pi|8430

Examples

```
str(dip4)

if (requireNamespace("T2EQ")) {
library(T2EQ)

  data(ex_data_pharmind, envir = environment())
  str(ex_data_pharmind)
  rm(ex_data_pharmind)
}
```

`dip5` *Fluid weights of drink cans*

Description

The `response` values of this data set correspond to the values published in the SAS/QC(R) 13.1 (2013) User's Guide, Chapter 5 (The CAPABILITY Procedure). The data set is described on page 199: The fluid weights of 100 drink cans were measured in ounces. The filling process is assumed to be in statistical control.

Usage

```
data(dip5)
```

Format

A data frame with 100 observations and 3 variables:

type Factor with the single level `reference`

batch Factor with levels `b1` to `b100`

weight Weight of drink cans

Source

See reference: Chapter 5 (The CAPABILITY Procedure), Cans data set shown on page 199.

References

SAS Institute Inc. 2013. *SAS/QC(R) 13.1 User's Guide*. Cary, NC: SAS Institute Inc.
https://support.sas.com/documentation/cdl/en/qcug/66857/PDF/default/qcug.pdf

Examples

```
str(dip5)
```

dip6 *Dissolution data of a reference and a test batch*

Description

A data set containing the simulated dissolution data of one reference batch and one test batch of $n = 12$ tablets each, i.e. the dissolution profiles of the % drug release observed within a period of 140 minutes. The profiles are simulated to have a kink between 115 and 125 minutes.

Usage

```
data(dip6)
```

Format

A data frame with 24 observations and 31 variables:

type Factor with levels `R` (Reference) and `T` (Test)

tablet Factor with levels `1` to `12` representing individual tablets

t.0 Numeric of the % release at the initial testing point

t.5 Numeric of the % release at the 5 minutes testing point

t.10 Numeric of the % release at the 10 minutes testing point

t.15 Numeric of the % release at the 15 minutes testing point

t.20 Numeric of the % release at the 20 minutes testing point

t.25 Numeric of the % release at the 25 minutes testing point

t.30 Numeric of the % release at the 30 minutes testing point

t.35 Numeric of the % release at the 35 minutes testing point

t.40 Numeric of the % release at the 40 minutes testing point

t.45 Numeric of the % release at the 45 minutes testing point

t.50 Numeric of the % release at the 50 minutes testing point

t.55 Numeric of the % release at the 55 minutes testing point

t.60 Numeric of the % release at the 60 minutes testing point

t.65 Numeric of the % release at the 65 minutes testing point

t.70 Numeric of the % release at the 70 minutes testing point

t.75 Numeric of the % release at the 75 minutes testing point

t.80 Numeric of the % release at the 80 minutes testing point

t.85 Numeric of the % release at the 85 minutes testing point

t.90 Numeric of the % release at the 90 minutes testing point

t.95 Numeric of the % release at the 95 minutes testing point

t.100 Numeric of the % release at the 100 minutes testing point

t.105 Numeric of the % release at the 105 minutes testing point

t.110 Numeric of the % release at the 110 minutes testing point

t.115 Numeric of the % release at the 115 minutes testing point

t.120 Numeric of the % release at the 120 minutes testing point

t.125 Numeric of the % release at the 125 minutes testing point

t.130 Numeric of the % release at the 130 minutes testing point

t.135 Numeric of the % release at the 135 minutes testing point

t.140 Numeric of the % release at the 140 minutes testing point

Examples

```
str(dip6)
```

`dip7` *Parameter estimates of Weibull fit to individual dissolution profiles*

Description

A data set containing the Weibull parameter estimates obtained from fitting Weibull curves to the cumulative dissolution profiles of individual tablets of three reference batches and one test batch, $n = 12$ tablets each. The Weibull curve is fitted according to the formula $x(t) = x_{max}(1 - exp(-\alpha t^{\beta}))$, where $x(t)$ is the percent released at time t divided by 100, x_{max} is the maximal release (set to be 100, i.e. assumed to be a constant).

Usage

```
data(dip7)
```

Format

A data frame with 48 observations and 5 variables:

tablet Factor with levels `1` to `12` representing individual tablets

batch Factor with levels `b0`, `b1`, `b2`, `b3` and `b4`

type Factor with levels `ref` (Reference) and `test` (Test)

alpha Weibull parameter α, i.e. the scale parameter being a function of the undissolved proportion at $t = 1$

beta Weibull parameter β, i.e. the shape parameter which is related to the dissolution rate per unit of time

Source

See reference: Example data set shown in Table 4.

References

Tsong, Y., Hammerstrom, T., Chen, J.J. Multipoint dissolution specification and acceptance sampling rule based on profile modeling and principal component analysis. *J Biopharm Stat*. 1997; **7**(3): 423-439.
doi:10.1080/10543409708835198

Examples

```
str(dip7)
```

`dip8` *Parameter estimates of Weibull fit to individual dissolution profiles*

Description

A data set containing the Weibull parameter estimates obtained from fitting Weibull curves to the cumulative dissolution profiles of individual tablets of one reference batch and one test or post-change batch with a minor modification and a second test or post-change batch with a major modification, $n = 12$ tablets each.

Usage

```
data(dip8)
```

Format

A data frame with 36 observations and 4 variables:

tablet Factor with levels `1` to `12` representing individual tablets

type Factor with levels `ref` (Reference), `minor` (Test) and `major` (Test)

alpha Weibull parameter α, i.e. the scale parameter being a function of the undissolved proportion at $t = 1$

beta Weibull parameter β, i.e. the shape parameter which is related to the dissolution rate per unit of time

Source

See reference: Example data set shown in Table III.

References

Sathe, P.M., Tsong, Y., and Shah, V.P. *In-Vitro* dissolution profile comparison: Statistics and analysis, model dependent approach. *Pharm Res*. 1996; **13**(12): 1799-1803.
doi:10.1023/a:1016020822093

Examples

```
str(dip8)
```

f1 *Dissimilarity factor f1 for dissolution data*

Description

The function `f1()` calculates the dissimilarity factor f_1.

Usage

```
f1(data, tcol, grouping, use_ema = "yes", bounds = c(1, 85), nsf = c(1, 2))
```

Arguments

`data`	A data frame with the dissolution profile data in wide format.
`tcol`	A vector of indices that specifies the columns in `data` that contain the % release values. The length of `tcol` must be three or longer.
`grouping`	A character string that specifies the column in `data` that contains the group names (i.e. a factorial variable, e.g., for the differentiation of batches or formulations of a drug product).
`use_ema`	A character string indicating whether the dissimilarity factor f_1 should be calculated following the EMA guideline "On the investigation of bioequivalence" (`"yes"`, the default) or not (`"no"`), i.e. the recommendations concerning the similarity factor f_2. A third option is `"ignore"`. If `use_ema` is `"yes"` or `"no"` the appropriate profile portion is determined on the basis of the values of the parameter `bounds`. If it is `"ignore"`, the complete profiles are used as specified by the parameter `tcol`.
`bounds`	A numeric vector of the form `c(lower, upper)` that specifies the "lower" and "upper" limits, respectively, for the % drug release given that `use_ema` is `"no"`. The default is `c(1, 85)`. Mean % release values of any of the two groups being compared that are smaller than or equal to the lower bound are ignored and only the first mean % release value that is greater than or equal to the upper bound is included while all the subsequent values are ignored. If `use_ema` is `"yes"` the `bounds` are `c(0, 85)` per definition. If `use_ema` is `"ignore"` the `bounds` are disregarded.
`nsf`	A vector of positive integers that specify the "number of significant figures" (nsf) of the corresponding values of the `bounds` parameter. It must thus have the same length as the `bounds` parameter. Before the % release values are compared with the limits that are specified by the `bounds` parameter, they are rounded to the corresponding number of significant figures as specified by the `nsf` parameter.

Details

Similarity of dissolution profiles is often assessed using the similarity factor f_2, as recommended by the EMA guideline (European Medicines Agency 2010) "On the investigation of bioequivalence". The evaluation of the similarity factor is based on the following constraints:

1. A minimum of three time points (zero excluded).
2. The time points should be the same for the two formulations.
3. Twelve individual values for every time point for each formulation.
4. Not more than one mean value of > 85% dissolved for any of the formulations.
5. The relative standard deviation or coefficient of variation of any product should be less than 20% for the first time point and less than 10% from the second to the last time point.

The *dis*similarity factor, or difference factor, f_1, is the counterpart of the similarity factor f_2. The difference factor f_1 is a measure of the relative error between two curves. Current FDA guidelines suggest that two profiles can be considered similar if f_1 is less than 15 $(0 - 15)$ and f_2 is greater than 50 $(50 - 100)$, which is equivalent to an average difference of 10% at all sampling time points. The dissimilarity factor f_1 is calculated by aid of the equation

$$f_1 = 100 \times \frac{\sum_{t=1}^{n} \left(\left| \bar{R}(t) - \bar{T}(t) \right| \right)}{\sum_{t=1}^{n} (\bar{R}(t))}.$$

In this equation

f_1 is the dissimilarity factor,

n is the number of time points,

$\bar{R}(t)$ is the mean percent reference drug dissolved at time t after initiation of the study, and

$\bar{T}(t)$ is the mean percent test drug dissolved at time t after initiation of the study.

Value

A list with the following elements is returned:

`f1`	A numeric value representing the similarity factor f_1.
`Profile.TP`	A named numeric vector of the columns in `data` specified by `tcol` and depending on the selection of `use_ema`. Given that the column names contain extractable numeric information, e.g., the testing time points of the dissolution profile, it contains the corresponding numeric values. Elements where no numeric information could be extracted are `NA`.

References

United States Food and Drug Administration (FDA). Guidance for industry: dissolution testing of immediate release solid oral dosage forms. 1997.
https://www.fda.gov/media/70936/download

United States Food and Drug Administration (FDA). Guidance for industry: immediate release solid oral dosage form: scale-up and post-approval changes, chemistry, manufacturing and controls, *in vitro* dissolution testing, and *in vivo* bioequivalence documentation (SUPAC-IR). 1995.
https://www.fda.gov/media/70949/download

European Medicines Agency (EMA), Committee for Medicinal Products for Human Use (CHMP). Guideline on the Investigation of Bioequivalence. 2010; CPMP/EWP/QWP/1401/98 Rev. 1.

See Also

f2.

Examples

```
# Use of defaults, i.e. 'use_ema = "yes"', 'bounds = c(1, 85)'
# Comparison always involves only two groups.
f1(data = dip1, tcol = 3:10, grouping = "type")

# $f1
# [1] 18.19745
#
# $Profile.TP
# t.5 t.10 t.15 t.20 t.30 t.60 t.90
#   5   10   15   20   30   60   90

# Use of 'use_ema = "no"', 'bounds = c(5, 80)'
f1(data = dip1, tcol = 3:10, grouping = "type", use_ema = "no",
   bounds = c(5, 80), nsf = c(1, 2))

# $f1
# [1] 21.333
#
# $Profile.TP
# t.5 t.10 t.15 t.20 t.30 t.60
#   5   10   15   20   30   60

# Use of 'use_ema = "no"', 'bounds = c(1, 95)'
f1(data = dip1, tcol = 3:10, grouping = "type", use_ema = "no",
   bounds = c(1, 95), nsf = c(1, 2))

# $f1
# [1] 16.22299
#
# $Profile.TP
# t.5  t.10  t.15  t.20  t.30  t.60  t.90 t.120
#   5    10    15    20    30    60    90   120

# In this case, the whole profiles are used. The same result is obtained
# when setting 'use_ema = "ignore"' (ignoring values passed to 'bounds').
f1(data = dip1, tcol = 3:10, grouping = "type", use_ema = "ignore")

# Passing in a data frame with a grouping variable with a number of levels that
# differs from two produces an error.
## Not run:
  tmp <- rbind(dip1,
               data.frame(type = "T2",
                          tablet = as.factor(1:6),
                          dip1[7:12, 3:10]))

  tryCatch(
    f1(data = tmp, tcol = 3:10, grouping = "type"),
```

```
    error = function(e) message(e),
    finally = message("\nMaybe you want to remove unesed levels in data."))

## End(Not run)

# Error in f1(data = tmp, tcol = 3:10, grouping = "type") :
#   The number of levels in column type differs from 2.
```

---

f2 *Similarity factor f2 for dissolution data*

Description

The function `f2()` calculates the similarity factor f_2.

Usage

```
f2(data, tcol, grouping, use_ema = "yes", bounds = c(1, 85), nsf = c(1, 2))
```

Arguments

`data`	A data frame with the dissolution profile data in wide format.
`tcol`	A vector of indices that specifies the columns in `data` that contain the % release values. The length of `tcol` must be three or longer.
`grouping`	A character string that specifies the column in `data` that contains the group names (i.e. a factorial variable, e.g., for the differentiation of batches or formulations of a drug product).
`use_ema`	A character string indicating whether the dissimilarity factor f_1 should be calculated following the EMA guideline "On the investigation of bioequivalence" (`"yes"`, the default) or not (`"no"`), i.e. the recommendations concerning the similarity factor f_2. A third option is `"ignore"`. If `use_ema` is `"yes"` or `"no"` the appropriate profile portion is determined on the basis of the values of the parameter `bounds`. If it is `"ignore"`, the complete profiles are used as specified by the parameter `tcol`.
`bounds`	A numeric vector of the form `c(lower, upper)` that specifies the "lower" and "upper" limits, respectively, for the % drug release given that `use_ema` is `"no"`. The default is `c(1, 85)`. Mean % release values of any of the two groups being compared that are smaller than or equal to the lower bound are ignored and only the first mean % release value that is greater than or equal to the upper bound is included while all the subsequent values are ignored. If `use_ema` is `"yes"` the `bounds` are `c(0, 85)` per definition. If `use_ema` is `"ignore"` the `bounds` are disregarded.
`nsf`	A vector of positive integers that specify the "number of significant figures" (nsf) of the corresponding values of the `bounds` parameter. It must thus have the same length as the `bounds` parameter. Before the % release values are compared with the limits that are specified by the `bounds` parameter, they are rounded to the corresponding number of significant figures as specified by the `nsf` parameter.

Details

Similarity of dissolution profiles is assessed using the similarity factor f_2 according to the EMA guideline (European Medicines Agency 2010) "On the investigation of bioequivalence". The evaluation of the similarity factor is based on the following constraints:

1. A minimum of three time points (zero excluded).
2. The time points should be the same for the two formulations.
3. Twelve individual values for every time point for each formulation.
4. Not more than one mean value of > 85% dissolved for any of the formulations.
5. The relative standard deviation or coefficient of variation of any product should be less than 20% for the first time point and less than 10% from the second to the last time point.

The similarity factor f_2 is calculated by aid of the equation

$$f_2 = 50 \times \log \left(\frac{100}{\sqrt{1 + \frac{\sum_{t=1}^{n} \left(\bar{R}(t) - \bar{T}(t) \right)^2}{n}}} \right).$$

In this equation

f_2 is the similarity factor,

n is the number of time points,

$\bar{R}(t)$ is the mean percent reference drug dissolved at time t after initiation of the study, and

$\bar{T}(t)$ is the mean percent test drug dissolved at time t after initiation of the study.

Dissolution profiles are regarded as similar if the f_2 value is between 50 and 100.

Value

A list with the following elements is returned:

`f2`	A numeric value representing the similarity factor f_2.
`Profile.TP`	A named numeric vector of the columns in `data` specified by `tcol` and depending on the selection of `use_ema`. Given that the column names contain extractable numeric information, e.g., the testing time points of the dissolution profile, it contains the corresponding numeric values. Elements where no numeric information could be extracted are `NA`.

References

United States Food and Drug Administration (FDA). Guidance for industry: dissolution testing of immediate release solid oral dosage forms. 1997.
https://www.fda.gov/media/70936/download

United States Food and Drug Administration (FDA). Guidance for industry: immediate release solid oral dosage form: scale-up and post-approval changes, chemistry, manufacturing and controls, *in*

vitro dissolution testing, and *in vivo* bioequivalence documentation (SUPAC-IR). 1995. https://www.fda.gov/media/70949/download

European Medicines Agency (EMA), Committee for Medicinal Products for Human Use (CHMP). Guideline on the Investigation of Bioequivalence. 2010; CPMP/EWP/QWP/1401/98 Rev. 1.

See Also

f1.

Examples

```
# Use of defaults, i.e. 'use_ema = "yes"', 'bounds = c(1, 85)'
# Comparison always involves only two groups.
f2(data = dip1, tcol = 3:10, grouping = "type")

# $f2
# [1] 40.83405
#
# $Profile.TP
# t.5 t.10 t.15 t.20 t.30 t.60 t.90
#   5   10   15   20   30   60   90

# Use of 'use_ema = "no"', 'bounds = c(5, 80)'
f2(data = dip1, tcol = 3:10, grouping = "type", use_ema = "no",
   bounds = c(5, 80), nsf = c(1, 2))

# $f2
# [1] 39.24385
#
# $Profile.TP
# t.5 t.10 t.15 t.20 t.30 t.60
#   5   10   15   20   30   60

# Use of 'use_ema = "no"', 'bounds = c(1, 95)'
f2(data = dip1, tcol = 3:10, grouping = "type", use_ema = "no",
   bounds = c(1, 95), nsf = c(1, 2))

# $f2
# [1] 42.11197
#
# $Profile.TP
# t.5  t.10  t.15  t.20  t.30  t.60  t.90 t.120
#   5    10    15    20    30    60    90   120

# In this case, the whole profiles are used. The same result is obtained
# when setting 'use_ema = "ignore"' (ignoring values passed to 'bounds').
f2(data = dip1, tcol = 3:10, grouping = "type", use_ema = "ignore")

# Passing in a data frame with a grouping variable with a number of levels that
# differs from two produces an error.
## Not run:
  tmp <- rbind(dip1,
```

```
                data.frame(type = "T2",
                           tablet = as.factor(1:6),
                           dip1[7:12, 3:10]))

  tryCatch(
    f2(data = tmp, tcol = 3:10, grouping = "type"),
    error = function(e) message(e),
    finally = message("\nMaybe you want to remove unesed levels in data."))

## End(Not run)

# Error in f1(data = tmp, tcol = 3:10, grouping = "type") :
#   The number of levels in column type differs from 2.
```

---

gep_by_nera *Get points on confidence region bounds by Newton-Raphson search*

Description

The function gep_by_nera() is a function for finding points that ideally sit on specific confidence region bounds (*CRB*) by aid of the "Method of Lagrange Multipliers" (MLM) and by "Newton-Raphson" (nera) optimisation. The multivariate confidence interval for profiles with four time points, e.g., is an "ellipse" in four dimensions.

Usage

```
gep_by_nera(n_p, kk, mean_diff, m_vc, ff_crit, y, max_trial, tol)
```

Arguments

n_p	A positive integer that specifies the number of (time) points n_p.
kk	A non-negative numeric value that specifies the scaling factor kk for the calculation of the Hotelling's T^2 statistic.
mean_diff	A vector of the mean differences between the dissolution profiles or model parameters of the reference and the test batch(es) or the averages of the model parameters of a specific group of batch(es) (reference or test). It must have the length specified by the parameter n_p.
m_vc	The pooled variance-covariance matrix of the dissolution profiles or model parameters of the reference and the test batch(es) or the variance-covariance matrix of the model parameters of a specific group of batch(es) (reference or test). It must have the dimension $n_p \times n_p$.
ff_crit	The critical F value (i.e. a non-negative numeric).
y	A numeric vector of y values that serve as starting points for the Newton-Raphson search, i.e. values supposed to lie on or close to the confidence interval bounds. It must have a length of $n_p + 1$.
max_trial	A positive integer that specifies the maximum number of Newton-Raphson search rounds to be performed.

`tol` A non-negative numeric that specifies the accepted minimal difference between two consecutive search rounds.

Details

The function `gep_by_nera()` determines the points on the *CRB* for each of the n_p time points. It does so by aid of the "Method of Lagrange Multipliers" (MLM) and by "Newton-Raphson" (nera) optimisation, as proposed by Margaret Connolly (Connolly 2000).

For more information, see the sections "Comparison of highly variable dissolution profiles" and "Similarity limits in terms of MSD" below.

Value

A list with the following elements is returned:

`points` A matrix with one column and n_p+1 rows is returned, where rows 1 to n_p represent, for each time point or model parameter, the points on the *CRB*. For symmetry reasons, the points on the opposite side are obtained by addition/subtraction. The last row in the matrix, with index $n_p + 1$, represents the λ parameter of the MLM, also known as *lambda multiplier method*, that is used to optimise under constraint(s). The variable λ is thus called the *Lagrange multiplier*.

`converged` A logical indicating whether the NR algorithm converged or not.

`points.on.crb` A logical indicating whether the points found by the NR algorithm sit on the sit on the confidence region bounds (`TRUE`) or not (`FALSE`). Since it is not know a priori it is `NA` by default. The parameter is set by the `check_point_location()` function.

`n.trial` Number of trials until convergence.

`max.trial` Maximal number of trials.

`tol` A non-negative numeric value that specifies the accepted minimal difference between two consecutive search rounds, i.e. the tolerance.

Comparison of highly variable dissolution profiles

When comparing the dissolution data of a post-approval change product and a reference approval product, the goal is to assess the similarity between the mean dissolution values at the observed sample time points. A widely used method is the f_2 method that was introduced by Moore & Flanner (1996). Similarity testing criteria based on f_2 can be found in several FDA guidelines and in the guideline of the European Medicines Agency (EMA) "On the investigation of bioequivalence" (EMA 2010).

In situations where within-batch variation is greater than 15%, FDA guidelines recommend use of a multivariate confidence interval as an alternative to the f_2 method. This can be done using the following stepwise procedure:

1. Establish a similarity limit in terms of "Multivariate Statistical Distance" (MSD) based on inter-batch differences in % drug release from reference (standard approved) formulations, i.e. the so-called "Equivalence Margin" (EM).
2. Calculate the MSD between test and reference mean dissolutions.

3. Estimate the 90% confidence interval (CI) of the true MSD as determined in step 2.
4. Compare the upper limit of the 90% CI with the similarity limit determined in step 1. The test formulation is declared to be similar to the reference formulation if the upper limit of the 90% CI is less than or equal to the similarity limit.

Similarity limits in terms of MSD

For the calculation of the "Multivariate Statistical Distance" (MSD), the procedure proposed by Tsong et al. (1996) can be considered as well-accepted method that is actually recommended by the FDA. According to this method, a multivariate statistical distance, called Mahalanobis distance, is used to measure the difference between two multivariate means. This distance measure is calculated as

$$D_M = \sqrt{(\boldsymbol{x}_T - \boldsymbol{x}_R)^\top \boldsymbol{S}_{pooled}^{-1} (\boldsymbol{x}_T - \boldsymbol{x}_R)},$$

where $\boldsymbol{S}_{pooled}$ is the sample variance-covariance matrix pooled across the comparative groups, $\boldsymbol{x}_T$ and $\boldsymbol{x}_R$ are the vectors of the sample means for the test (T) and reference (R) profiles, and $\boldsymbol{S}_T$ and $\boldsymbol{S}_R$ are the variance-covariance matrices of the test and reference profiles. The pooled variance-covariance matrix $\boldsymbol{S}_{pooled}$ is calculated by

$$\boldsymbol{S}_{pooled} = \frac{(n_R - 1)\boldsymbol{S}_R + (n_T - 1)\boldsymbol{S}_T}{n_R + n_T - 2}.$$

In order to determine the similarity limits in terms of the MSD, i.e. the Mahalanobis distance between the two multivariate means of the dissolution profiles of the formulations to be compared, Tsong et al. (1996) proposed using the equation

$$D_M^{max} = \sqrt{\boldsymbol{d}_g^\top \boldsymbol{S}_{pooled}^{-1} \boldsymbol{d}_g},$$

where $\boldsymbol{d}_g$ is a $1 \times p$ vector with all p elements equal to an empirically defined limit $\boldsymbol{d}_g$, e.g., 15%, for the maximum tolerable difference at all time points, and p is the number of sampling points. By assuming that the data follow a multivariate normal distribution, the 90% confidence region (*CR*) bounds for the true difference between the mean vectors, $\boldsymbol{\mu}_T - \boldsymbol{\mu}_R$, can be computed for the resultant vector $\boldsymbol{\mu}$ to satisfy the following condition:

$$CR = K\,(\boldsymbol{\mu} - (\boldsymbol{x}_T - \boldsymbol{x}_R))^\top \boldsymbol{S}_{pooled}^{-1} (\boldsymbol{\mu} - (\boldsymbol{x}_T - \boldsymbol{x}_R)) \leq F_{p,n_T+n_R-p-1,0.9},$$

where K is the scaling factor that is calculated as

$$K = \frac{n_T n_R}{n_T + n_R} \frac{n_T + n_R - p - 1}{(n_T + n_R - 2)\,p},$$

and $F_{p,n_T+n_R-p-1,0.9}$ is the 90^{th} percentile of the F distribution with degrees of freedom p and $n_T + n_R - p - 1$, where n_T and n_R are the number of observations of the reference and the test group, respectively, and p is the number of sampling or time points, as mentioned already. It is obvious that $(n_T + n_R)$ must be greater than $(p + 1)$. The formula for *CR* gives a p-variate 90% confidence region for the possible true differences.

References

United States Food and Drug Administration (FDA). Guidance for industry: dissolution testing of immediate release solid oral dosage forms. 1997.
https://www.fda.gov/media/70936/download

United States Food and Drug Administration (FDA). Guidance for industry: immediate release solid oral dosage form: scale-up and post-approval changes, chemistry, manufacturing and controls, *in vitro* dissolution testing, and *in vivo* bioequivalence documentation (SUPAC-IR). 1995.
https://www.fda.gov/media/70949/download

European Medicines Agency (EMA), Committee for Medicinal Products for Human Use (CHMP). Guideline on the Investigation of Bioequivalence. 2010; CPMP/EWP/QWP/1401/98 Rev. 1..

Moore, J.W., and Flanner, H.H. Mathematical comparison of curves with an emphasis on *in-vitro* dissolution profiles. *Pharm Tech*. 1996; **20**(6): 64-74.

Tsong, Y., Hammerstrom, T., Sathe, P.M., and Shah, V.P. Statistical assessment of mean differences between two dissolution data sets. *Drug Inf J*. 1996; **30**: 1105-1112.
doi:10.1177/009286159603000427

Connolly, M. SAS(R) IML Code to calculate an upper confidence limit for multivariate statistical distance; 2000; Wyeth Lederle Vaccines, Pearl River, NY.
https://analytics.ncsu.edu/sesug/2000/p-902.pdf

See Also

check_point_location, mimcr, bootstrap_f2.

Examples

```
# Collecting the required information
time_points <- suppressWarnings(as.numeric(gsub("([^0-9])", "",
                                                colnames(dip1))))
tcol <- which(!is.na(time_points))
b1 <- dip1$type == "R"

# Hotelling's T2 statistics
l_hs <- get_T2_two(m1 = as.matrix(dip1[b1, tcol]),
                   m2 = as.matrix(dip1[!b1, tcol]),
                   signif = 0.05)

# Calling gep_by_nera()
res <- gep_by_nera(n_p = as.numeric(l_hs[["Parameters"]]["df1"]),
                   kk = as.numeric(l_hs[["Parameters"]]["K"]),
                   mean_diff = l_hs[["means"]][["mean.diff"]],
                   m_vc = l_hs[["S.pool"]],
                   ff_crit = as.numeric(l_hs[["Parameters"]]["F.crit"]),
                   y = rep(1, times = l_hs[["Parameters"]]["df1"] + 1),
                   max_trial = 100, tol = 1e-9)

# Expected result in res[["points"]]
#              [,1]
# t.5   -15.7600077
# t.10  -13.6501734
```

```
# t.15  -11.6689469
# t.20   -9.8429369
# t.30   -6.6632182
# t.60   -0.4634318
# t.90    2.2528551
# t.120   3.3249569
#       -17.6619995

# Rows t.5 to t.120 represent the points on the CR bounds.The unnamed last row
# represents the Lagrange multiplier lambda.

# If 'max_trial' is too small, the Newton-Raphson search may not converge.
## Not run:
  tryCatch(
    gep_by_nera(n_p = as.numeric(l_hs[["Parameters"]]["df1"]),
                kk = as.numeric(l_hs[["Parameters"]]["K"]),
                mean_diff = l_hs[["means"]][["mean.diff"]],
                m_vc = l_hs[["S.pool"]],
                ff_crit = as.numeric(l_hs[["Parameters"]]["F.crit"]),
                y = rep(1, times = l_hs[["Parameters"]]["df1"] + 1),
                max_trial = 5, tol = 1e-9),
    warning = function(w) message(w),
    finally = message("\nMaybe increasing the number of max_trial could help."))

## End(Not run)
```

`get_sim_lim` *Similarity limit*

Description

The function `get_sim_lim()` estimates a similarity limit in terms of the "Multivariate Statistical Distance" (MSD).

Usage

```
get_sim_lim(mtad, lhs)
```

Arguments

`mtad`	A numeric value that specifies the "maximum tolerable average difference" (MTAD) of the profiles of two formulations at all time points (in %). The default value is `10`. It determines the size of the similarity limit $\boldsymbol{d}_g$ (see the details section for more information).
`lhs`	A list of the estimates of Hotelling's two-sample T^2 statistic for small samples as returned by the function `get_T2_two()`.

Details

Details about the estimation of similarity limits in terms of the "Multivariate Statistical Distance" (MSD) are explained in the corresponding section below.

Value

A vector containing the following information is returned:

`dm`	The Mahalanobis distance of the samples.
`df1`	Degrees of freedom (number of variables or time points).
`df2`	Degrees of freedom (number of rows - number of variables - 1).
`alpha`	The provided significance level.
`K`	Scaling factor for F to account for the distribution of the T^2 statistic.
`k`	Scaling factor for the squared Mahalanobis distance to obtain the T^2 statistic.
`T2`	Hotelling's T^2 statistic (F-distributed).
`F`	Observed F value.
`ncp.Hoffelder`	Non-centrality parameter for calculation of the F statistic (T^2 test procedure).
`F.crit`	Critical F value (Tsong's procedure).
`F.crit.Hoffelder`	Critical F value (T^2 test procedure).
`p.F`	The p value for the Hotelling's T^2 test statistic.
`p.F.Hoffelder`	The p value for the Hotelling's T^2 statistic based on the non-central F distribution.
`MTAD`	Specified "maximum tolerable average difference" (MTAD) of the profiles of two formulations at each individual time point (in %).
`Sim.Limit`	Critical Mahalanobis distance or similarity limit (Tsong's procedure).

Similarity limits in terms of MSD

For the calculation of the "Multivariate Statistical Distance" (MSD), the procedure proposed by Tsong et al. (1996) can be considered as well-accepted method that is actually recommended by the FDA. According to this method, a multivariate statistical distance, called Mahalanobis distance, is used to measure the difference between two multivariate means. This distance measure is calculated as

$$D_M = \sqrt{(\boldsymbol{x}_T - \boldsymbol{x}_R)^\top \boldsymbol{S}_{pooled}^{-1} (\boldsymbol{x}_T - \boldsymbol{x}_R)},$$

where $\boldsymbol{S}_{pooled}$ is the sample variance-covariance matrix pooled across the comparative groups, $\boldsymbol{x}_T$ and $\boldsymbol{x}_R$ are the vectors of the sample means for the test (T) and reference (R) profiles, and $\boldsymbol{S}_T$ and $\boldsymbol{S}_R$ are the variance-covariance matrices of the test and reference profiles. The pooled variance-covariance matrix $\boldsymbol{S}_{pooled}$ is calculated by

$$\boldsymbol{S}_{pooled} = \frac{(n_R - 1)\boldsymbol{S}_R + (n_T - 1)\boldsymbol{S}_T}{n_R + n_T - 2}.$$

In order to determine the similarity limits in terms of the MSD, i.e. the Mahalanobis distance between the two multivariate means of the dissolution profiles of the formulations to be compared, Tsong et al. (1996) proposed using the equation

$$D_M^{max} = \sqrt{\boldsymbol{d}_g^\top \boldsymbol{S}_{pooled}^{-1} \boldsymbol{d}_g},$$

where $\boldsymbol{d}_g$ is a $1 \times p$ vector with all p elements equal to an empirically defined limit $\boldsymbol{d}_g$, e.g., 15%, for the maximum tolerable difference at all time points, and p is the number of sampling points. By assuming that the data follow a multivariate normal distribution, the 90% confidence region (*CR*) bounds for the true difference between the mean vectors, $\boldsymbol{\mu}_T - \boldsymbol{\mu}_R$, can be computed for the resultant vector $\boldsymbol{\mu}$ to satisfy the following condition:

$$CR = K\left(\boldsymbol{\mu} - (\boldsymbol{x}_T - \boldsymbol{x}_R)\right)^\top \boldsymbol{S}_{pooled}^{-1} \left(\boldsymbol{\mu} - (\boldsymbol{x}_T - \boldsymbol{x}_R)\right) \leq F_{p,n_T+n_R-p-1,0.9},$$

where K is the scaling factor that is calculated as

$$K = \frac{n_T n_R}{n_T + n_R} \, \frac{n_T + n_R - p - 1}{(n_T + n_R - 2)\, p},$$

and $F_{p,n_T+n_R-p-1,0.9}$ is the 90^{th} percentile of the F distribution with degrees of freedom p and $n_T + n_R - p - 1$, where n_T and n_R are the number of observations of the reference and the test group, respectively, and p is the number of sampling or time points, as mentioned already. It is obvious that $(n_T + n_R)$ must be greater than $(p + 1)$. The formula for *CR* gives a p-variate 90% confidence region for the possible true differences.

T2 test for equivalence

Based on the distance measure for profile comparison that was suggested by Tsong et al. (1996), i.e. the Mahalanobis distance, Hoffelder (2016) proposed a statistical equivalence procedure for that distance, the so-called T^2 test for equivalence (T2EQ). It is used to demonstrate that the Mahalanobis distance between reference and test group dissolution profiles is smaller than the "Equivalence Margin" (EM). Decision in favour of equivalence is taken if the p value of this test statistic is smaller than the pre-specified significance level α, i.e. if $p < \alpha$. The p value is calculated by aid of the formula

$$p = F_{p,n_T+n_R-p-1,ncp,\alpha} \, \frac{n_T + n_R - p - 1}{(n_T + n_R - 2)p} T^2,$$

where α is the significance level and ncp is the so-called "*non-centrality parameter*" that is calculated by

$$\frac{n_T n_R}{n_T + n_R} \left(D_M^{max}\right)^2.$$

The test statistic being used is Hotelling's two-sample T^2 test that is given as

$$T^2 = \frac{n_T n_R}{n_T + n_R} \left(\boldsymbol{x}_T - \boldsymbol{x}_R\right)^\top \boldsymbol{S}_{pooled}^{-1} \left(\boldsymbol{x}_T - \boldsymbol{x}_R\right).$$

As mentioned in paragraph "Similarity limits in terms of MSD", $\boldsymbol{d}_g$ is a $1 \times p$ vector with all p elements equal to an empirically defined limit d_g. Thus, the components of the vector $\boldsymbol{d}_g$ can be interpreted as upper bound for a kind of "*average*" allowed difference between test and reference profiles, the "*global similarity limit*". Since the EMA requires that "similarity acceptance limits should be pre-defined and justified and not be greater than a 10% difference", it is recommended to use 10%, not 15% as proposed by Tsong et al. (1996), for the maximum tolerable difference at all time points.

References

Tsong, Y., Hammerstrom, T., Sathe, P.M., and Shah, V.P. Statistical assessment of mean differences between two dissolution data sets. *Drug Inf J*. 1996; **30**: 1105-1112.
doi:10.1177/009286159603000427

Wellek S. (2010) *Testing statistical hypotheses of equivalence and noninferiority* (2nd ed.). Chapman & Hall/CRC, Boca Raton.
doi:10.1201/EBK1439808184

Hoffelder, T. Highly variable dissolution profiles. Comparison of T^2-test for equivalence and f_2 based methods. *Pharm Ind*. 2016; **78**(4): 587-592.
https://www.ecv.de/suse_item.php?suseId=Z|pi|8430

See Also

mimcr, get_T2_two.

Examples

```
# Estimation of the parameters for Hotelling's two-sample T2 statistic
# (for small samples)
hs <- get_T2_two(m1 = as.matrix(dip1[dip1$type == "R", c("t.15", "t.90")]),
                 m2 = as.matrix(dip1[dip1$type == "T", c("t.15", "t.90")]),
                 signif = 0.1)

# Estimation of the similarity limit in terms of the "Multivariate Statistical
# Distance" (MSD)for a "maximum tolerable average difference" (mtad) of 10
res <- get_sim_lim(mtad = 15, hs)

# Expected results in res
#            DM              df1              df2            alpha
#  1.044045e+01     2.000000e+00     9.000000e+00     1.000000e-01
#             K                k               T2                F
#  1.350000e+00     3.000000e+00     3.270089e+02     1.471540e+02
# ncp.Hoffelder           F.crit F.crit.Hoffelder              p.F
#  2.782556e+02     3.006452e+00     8.357064e+01     1.335407e-07
# p.F.Hoffelder             MTAD        Sim.Limit
#  4.822832e-01     1.500000e+01     9.630777e+00
```

get_T2_one *Hotelling's statistics (for one (small) sample)*

Description

The function `get_T2_one()` estimates the parameters for Hotelling's one-sample T^2 statistic for small samples.

Usage

```
get_T2_one(m, mu, signif, na_rm = FALSE)
```

Arguments

`m`	A matrix with the data of the reference group, e.g. a matrix for the different model parameters (columns) of different dosage unit (rows).
`mu`	A numeric vector of, e.g. the hypothetical model parameter mean values.
`signif`	A positive numeric value between `0` and `1` that specifies the significance level. The default value is `0.05`.
`na_rm`	A logical value that indicates whether observations containing `NA` (or `NaN`) values should be removed (`na_rm = TRUE`) or not (`na_rm = FALSE`). The default is `na_rm = FALSE`.

Details

The one-sample Hotelling's T^2 test statistic is given by

$$T^2 = n\,(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_0)^\top\, \boldsymbol{S}^{-1}\,(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_0)\,.$$

where $\bar{\boldsymbol{x}}$ is the vector of the sample means of the sample group, e.g. the vector of the average dissolution per time point or of the average model parameters, n is the numbers of observations of the sample group (i.e. the number of rows in matrix `m` handed over to the `get_T2_one()` function, and $\boldsymbol{S}$ is variance-covariance matrix. The matrix $\boldsymbol{S}^{-1}$ is the inverted variance-covariance matrix. The term

$$D_M = \sqrt{(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_0)^\top\, \boldsymbol{S}^{-1}\,(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_0)}$$

is the Mahalanobis distance measuring the difference between the sample mean vector and the vector of the hypothetical values $\boldsymbol{\mu}_0$. For large samples, T^2 is approximately chi-square distributed with p degrees of freedom, where p is the number of variables, i.e. the number of dissolution profile time points or the number of model parameters. In terms of the Mahalanobis distance, the one-sample Hotelling's T^2 statistic can be expressed has

$$n\; D_M^2 = k\; D_M^2.$$

To transform the one-sample Hotelling's T^2 statistic into an F-statistic, a conversion factor is necessary, i.e.

$$K = k \, \frac{n-p}{(n-1)p}.$$

With this transformation, the following test statistic can be applied:

$$K \; D_M^2 \leq F_{p,n-p,\alpha}.$$

Under the null hypothesis, $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$, this F-statistic is F-distributed with p and $n-p$ degrees of freedom. H_0 is rejected at a significance level of α if the test statistic F exceeds the critical value from the F-table evaluated at α, i.e. $F > F_{p,n-p,\alpha}$.

The following assumptions concerning the data are made:

- The data of population x has no sub-populations, i.e. there are no sub-populations of x with different means.
- The observations are based on a common variance-covariance matrix Σ.
- The observations have been independently sampled.
- The observations have been sampled from a multivariate normal distribution.

Confidence intervals:
Simultaneous $(1-\alpha)100\%$ confidence intervals for all linear combinations of the sample means are given by the expression

$$(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_0) \pm \sqrt{\frac{1}{K} \; F_{p,n-p,\alpha} \; \boldsymbol{s}},$$

where $\boldsymbol{s}$ is the vector of the diagonal elements of the variance-covariance matrix $\boldsymbol{S}$. With $(1-\alpha)100\%$ confidence, this interval covers the respective linear combination of the differences between the sample means and the hypothetical means. If not the linear combination of the variables is of interest but rather the individual variables, then the Bonferroni corrected confidence intervals should be used instead which are given by the expression

$$(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_0) \pm t_{n-1,\frac{\alpha}{2p}} \; \sqrt{\frac{1}{k} \; \boldsymbol{s}}.$$

Value

A list with the following elements is returned:

`Parameters`	Parameters determined for the estimation of Hotelling's T^2.
`cov`	The variance-covariance matrix of the reference group.
`means`	A list with the elements `mean.r`, `mean.t` and `mean.diff`, i.e. the average model parameters of the reference group, the hypothetical average model parameters (handed over via the `mu` parameter) and the corresponding differences, respectively.

CI	A list with the elements Hotelling and Bonferroni, i.e. data frames with columns LCL and UCL for the lower and upper $(1-\alpha)100\%$ confidence limits, respectively, and rows for each time point or model parameter.

The Parameters element contains the following information:

dm	Mahalanobis distance of the samples.
df1	Degrees of freedom (number of variables or time points).
df2	Degrees of freedom (number of rows - number of variables - 1).
alpha	Provided significance level.
K	Scaling factor for F to account for the distribution of the T^2 statistic.
k	Scaling factor for the squared Mahalanobis distance to obtain the T^2 statistic.
T2	Hotelling's T^2 statistic (F-distributed).
F	Observed F value.
F.crit	Critical F value.
t.crit	Critical t value.
p.F	p value for Hotelling's T^2 test statistic.

References

Hotelling, H. The generalisation of Student's ratio. *Ann Math Stat*. 1931; **2**(3): 360-378.

Hotelling, H. (1947) *Multivariate quality control illustrated by air testing of sample bombsights*. In: Eisenhart, C., Hastay, M.W., and Wallis, W.A., Eds., Techniques of Statistical Analysis, McGraw Hill, New York, 111-184.

See Also

get_T2_two, get_sim_lim.

Examples

```
# Estimation of the parameters for Hotelling's one-sample T2 statistic
# (for small samples)
# Check if there is a significant difference of the test batch results
# from the average reference batch results.
# Since p.F in res1$Parameters is smaller than 0.1, it is concluded that the
# new batch differs from the reference batch.
res1 <-
  get_T2_one(m = as.matrix(dip1[dip1$type == "T", c("t.15", "t.90")]),
             mu = colMeans(dip1[dip1$type == "R", c("t.15", "t.90")]),
             signif = 0.1, na_rm = FALSE)
res1$Parameters

# Expected results in res1$Parameters
#           dm          df1          df2       signif            K
# 1.314197e+01 2.000000e+00 4.000000e+00 1.000000e-01 2.400000e+00
#            k           T2            F       F.crit       t.crit
# 6.000000e+00 1.036268e+03 4.145072e+02 4.324555e+00 2.570582e+00
```

```
#          p.F
# 2.305765e-05

# In Tsong (1997) (see reference of dip7), the model-dependent approach is
# illustrated with an example data set of alpha and beta parameters obtained
# by fitting the Weibull curve function to a data set of dissolution profiles
# of three reference batches and one new batch (12 profiles per batch).
# Check if there is a significant difference of the test batch results
# from the average reference batch results.
# Since p.F in res2$Parameters is smaller than 0.05, it is concluded that the
# test batch differs from the reference batches.
res2 <-
  get_T2_one(m = as.matrix(dip7[dip7$type == "test", c("alpha", "beta")]),
             mu = colMeans(dip7[dip7$type == "ref", c("alpha", "beta")]),
             signif = 0.05, na_rm = FALSE)
res2$Parameters

# Expected results in res2$Parameters
#           dm          df1          df2       signif            K
# 5.984856e+00 2.000000e+00 1.000000e+01 5.000000e-02 5.454545e+00
#            k           T2            F       F.crit       t.crit
# 1.200000e+01 4.298220e+02 1.953736e+02 4.102821e+00 2.593093e+00
#          p.F
# 9.674913e-09

# In Sathe (1996) (see reference of dip8), the model-dependent approach is
# illustrated with an example data set of alpha and beta parameters obtained
# by fitting the Weibull curve function to a data set of dissolution profiles
# of one reference batch and one new batch with minor modifications and another
# new batch with major modifications (12 profiles per batch).
# Check if there is a significant difference of the results of the minor or
# major modificated batches from the average reference batch results.
# Since p.F in res3.minor$Parameters or in res3.major$Parameters are smaller
# than 0.1, it is concluded that the minor and the major modification batch
# differs from the reference batch.
res3.minor <-
  get_T2_one(m = log(as.matrix(dip8[dip8$type == "minor",
                                    c("alpha", "beta")])),
             mu = log(colMeans(dip8[dip8$type == "ref",
                                    c("alpha", "beta")])),
             signif = 0.1, na_rm = FALSE)
res3.major <-
  get_T2_one(m = log(as.matrix(dip8[dip8$type == "major",
                                    c("alpha", "beta")])),
             mu = log(colMeans(dip8[dip8$type == "ref",
                                    c("alpha", "beta")])),
             signif = 0.1, na_rm = FALSE)
res3.minor$Parameters
res3.major$Parameters

# Expected results in res3.minor$Parameters
#           dm          df1          df2       signif            K
# 2.718715e+00 2.000000e+00 1.000000e+01 1.000000e-01 5.454545e+00
```

```
#            k           T2            F       F.crit       t.crit
# 1.200000e+01 8.869691e+01 4.031678e+01 2.924466e+00 2.200985e+00
#          p.F
# 1.635140e-05

# Expected results in res3.major$Parameters
#           dm          df1          df2       signif            K
# 5.297092e+00 2.000000e+00 1.000000e+01 1.000000e-01 5.454545e+00
#            k           T2            F       F.crit       t.crit
# 1.200000e+01 3.367102e+02 1.530501e+02 2.924466e+00 2.200985e+00
#          p.F
# 3.168664e-08
```

`get_T2_two` *Hotelling's statistics (for two independent (small) samples)*

Description

The function `get_T2_two()` estimates the parameters for Hotelling's two-sample T^2 statistic for small samples.

Usage

```
get_T2_two(m1, m2, signif, na_rm = FALSE)
```

Arguments

`m1`	A matrix with the data of the reference group, e.g. a matrix representing dissolution profiles, i.e. with rows for the different dosage units and columns for the different time points, or a matrix for the different model parameters (columns) of different dosage units (rows).
`m2`	A matrix with the same dimensions as matrix `m1` with the data of the test group having the characteristics as the data of matrix `m1`.
`signif`	A positive numeric value between `0` and `1` that specifies the significance level. The default value is `0.05`.
`na_rm`	A logical value that indicates whether observations containing `NA` (or `NaN`) values should be removed (`na_rm = TRUE`) or not (`na_rm = FALSE`). The default is `na_rm = FALSE`.

Details

The two-sample Hotelling's T^2 test statistic is given by

$$T^2 = \frac{n_T n_R}{n_T + n_R} \left(\boldsymbol{x}_T - \boldsymbol{x}_R\right)^\top \boldsymbol{S}_{pooled}^{-1} \left(\boldsymbol{x}_T - \boldsymbol{x}_R\right),$$

where $\boldsymbol{x}_T$ and $\boldsymbol{x}_R$ are the vectors of the sample means of the test (T) and reference (R) group, e.g. vectors of the average dissolution per time point or of the average model parameters, n_T and n_R

are the numbers of observations of the reference and the test group, respectively (i.e. the number of rows in matrices `m1` and `m2` handed over to the `get_T2_two()` function), and $\boldsymbol{S}_{pooled}$ is the pooled variance-covariance matrix which is calculated by

$$\boldsymbol{S}_{pooled} = \frac{(n_R - 1)\boldsymbol{S}_R + (n_T - 1)\boldsymbol{S}_T}{n_R + n_T - 2},$$

where $\boldsymbol{S}_R$ and $\boldsymbol{S}_T$ are the estimated variance-covariance matrices which are calculated from the matrices of the two groups being compared, i.e. `m1` and `m2`. The matrix $\boldsymbol{S}_{pooled}^{-1}$ is the inverted variance-covariance matrix. As the number of columns of matrices `m1` and `m2` increases, and especially as the correlation between the columns increases, the risk increases that the pooled variance-covariance matrix $\boldsymbol{S}_{pooled}$ is ill-conditioned or even singular and thus cannot be inverted. The term

$$D_M = \sqrt{(\boldsymbol{x}_T - \boldsymbol{x}_R)^\top \boldsymbol{S}_{pooled}^{-1} (\boldsymbol{x}_T - \boldsymbol{x}_R)}$$

is the Mahalanobis distance which is used to measure the difference between two multivariate means. For large samples, T^2 is approximately chi-square distributed with p degrees of freedom, where p is the number of variables, i.e. the number of dissolution profile time points or the number of model parameters. In terms of the Mahalanobis distance, Hotelling's T^2 statistic can be expressed has

$$\frac{n_T n_R}{n_T + n_R}\, D_M^2 = k\, D_M^2.$$

To transform the Hotelling's T^2 statistic into an F-statistic, a conversion factor is necessary, i.e.

$$K = k\, \frac{n_T + n_R - p - 1}{(n_T + n_R - 2)\, p}.$$

With this transformation, the following test statistic can be applied:

$$K\, D_M^2 \leq F_{p, n_T + n_R - p - 1, \alpha}.$$

Under the null hypothesis, $H_0 : \boldsymbol{\mu}_T = \boldsymbol{\mu}_R$, this F-statistic is F-distributed with p and $n_T + n_R - p - 1$ degrees of freedom. H_0 is rejected at significance level α if the F-value exceeds the critical value from the F-table evaluated at α, i.e. $F > F_{p, n_T + n_R - p - 1, \alpha}$. The null hypothesis is satisfied if, and only if, the population means are identical for all variables. The alternative is that at least one pair of these means is different.

The following assumptions concerning the data are made:

- The data from population i is a sample from a population with mean vector μ_i. In other words, it is assumed that there are no sub-populations.
- The data from both populations have common variance-covariance matrix Σ.
- The elements from both populations are independently sampled, i.e. the data values are independent.
- Both populations are multivariate normally distributed.

Confidence intervals:
Confidence intervals for the mean differences at each time point or confidence intervals for the mean differences between the parameter estimates of the reference and the test group are calculated by aid of the formula

$$(\boldsymbol{x}_T - \boldsymbol{x}_R) \pm \sqrt{\frac{1}{K}\, F_{p,n_T+n_R-p-1,\alpha}\; \boldsymbol{s}_{pooled}},$$

where $\boldsymbol{s}_{pooled}$ is the vector of the diagonal elements of the pooled variance-covariance matrix $\boldsymbol{S}_{pooled}$. With $(1-\alpha)100\%$ confidence, this interval covers the respective linear combination of the differences between the means of the two sample groups. If not the linear combination of the variables is of interest but rather the individual variables, then the Bonferroni corrected confidence intervals should be used instead which are given by the expression

$$(\boldsymbol{x}_T - \boldsymbol{x}_R) \pm t_{n_T+n_R-2,\frac{\alpha}{2p}}\; \sqrt{\frac{1}{k}\; \boldsymbol{s}_{pooled}}.$$

Value

A list with the following elements is returned:

`Parameters`	Parameters determined for the estimation of Hotelling's T^2.
`S.pool`	Pooled variance-covariance matrix.
`covs`	A list with the elements `S.b1` and `S.b2`, i.e. the variance-covariance matrices of the reference and the test group, respectively.
`means`	A list with the elements `mean.b1`, `mean.b2` and `mean.diff`, i.e. the average dissolution profile values (for each time point) or the average model parameters of the reference and the test group and the corresponding differences, respectively.
`CI`	A list with the elements `Hotelling` and `Bonferroni`, i.e. data frames with columns `LCL` and `UCL` for the lower and upper $(1-\alpha)100\%$ confidence limits, respectively, and rows for each time point or model parameter.

The `Parameters` element contains the following information:

`dm`	Mahalanobis distance of the samples.
`df1`	Degrees of freedom (number of variables or time points).
`df2`	Degrees of freedom (number of rows - number of variables - 1).
`alpha`	Provided significance level.
`K`	Scaling factor for F to account for the distribution of the T^2 statistic.
`k`	Scaling factor for the squared Mahalanobis distance to obtain the T^2 statistic.
`T2`	Hotelling's T^2 statistic (F-distributed).
`F`	Observed F value.
`F.crit`	Critical F value.
`t.crit`	Critical t value.
`p.F`	p value for Hotelling's T^2 test statistic.

References

Hotelling, H. The generalisation of Student's ratio. *Ann Math Stat*. 1931; **2**(3): 360-378.

Hotelling, H. (1947) *Multivariate quality control illustrated by air testing of sample bombsights*. In: Eisenhart, C., Hastay, M.W., and Wallis, W.A., Eds., Techniques of Statistical Analysis, McGraw Hill, New York, 111-184.

See Also

get_T2_one, get_sim_lim, mimcr.

Examples

```
# Estimation of the parameters for Hotelling's two-sample T2 statistic
# (for small samples)
res1 <- get_T2_two(m1 = as.matrix(dip1[dip1$type == "R", c("t.15", "t.90")]),
                   m2 = as.matrix(dip1[dip1$type == "T", c("t.15", "t.90")]),
                   signif = 0.1)
res1$S.pool
res1$Parameters

# Results in res1$S.pool
#          t.15     t.90
# t.15 3.395808 1.029870
# t.90 1.029870 4.434833

# Results in res1$Parameters
#           dm          df1          df2       signif            K
# 1.044045e+01 2.000000e+00 9.000000e+00 1.000000e-01 1.350000e+00
#            k           T2            F       F.crit       t.crit
# 3.000000e+00 3.270089e+02 1.471540e+02 3.006452e+00 2.228139e+00
#          p.F
# 1.335407e-07

# The results above correspond to the values that are shown in Tsong (1996)
# (see reference of dip1 data set) under paragraph "DATA1 data (Comparing
# the 15- and 90-minute sample time points only).

# For the second assessment shown in Tsong (1996) (see reference of dip1 data
# set) under paragraph "DATA2 data (Comparing all eight time points), the
# following results are obtained.
res2 <- get_T2_two(m1 = as.matrix(dip1[dip1$type == "R", 3:10]),
                   m2 = as.matrix(dip1[dip1$type == "T", 3:10]),
                   signif = 0.1)
res2$Parameters

# Results in res2$Parameters
#           dm          df1          df2       signif            K
# 2.648562e+01 8.000000e+00 3.000000e+00 1.000000e-01 1.125000e-01
#            k           T2            F       F.crit       t.crit
# 3.000000e+00 2.104464e+03 7.891739e+01 5.251671e+00 3.038243e+00
#          p.F
```

```
# 2.116258e-03

# In Tsong (1997) (see reference of dip7), the model-dependent approach is
# illustrated with an example data set of alpha and beta parameters obtained
# by fitting the Weibull curve function to a data set of dissolution profiles
# of three reference batches and one new batch (12 profiles per batch).
res3 <-
  get_T2_two(m1 = as.matrix(dip7[dip7$type == "ref", c("alpha", "beta")]),
             m2 = as.matrix(dip7[dip7$type == "test", c("alpha", "beta")]),
             signif = 0.05)
res3$Parameters

# Results in res3$Parameters
#           dm          df1          df2       signif            K
# 3.247275e+00 2.000000e+00 4.500000e+01 5.000000e-02 4.402174e+00
#            k           T2            F       F.crit       t.crit
# 9.000000e+00 9.490313e+01 4.642001e+01 3.204317e+00 2.317152e+00
#          p.F
# 1.151701e-11

# In Sathe (1996) (see reference of dip8), the model-dependent approach is
# illustrated with an example data set of alpha and beta parameters obtained
# by fitting the Weibull curve function to a data set of dissolution profiles
# of one reference batch and one new batch with minor modifications and another
# new batch with major modifications (12 profiles per batch). Note that the
# assessment is performed on the (natural) logarithm scale.
res4.minor <- get_T2_two(m1 = log(as.matrix(dip8[dip8$type == "ref",
                                                 c("alpha", "beta")])),
                         m2 = log(as.matrix(dip8[dip8$type == "minor",
                                                 c("alpha", "beta")])),
                         signif = 0.1)
res4.major <- get_T2_two(m1 = log(as.matrix(dip8[dip8$type == "ref",
                                                 c("alpha", "beta")])),
                         m2 = log(as.matrix(dip8[dip8$type == "major",
                                                 c("alpha", "beta")])),
                         signif = 0.1)
res4.minor$Parameters
res4.minor$CI$Hotelling
res4.major$Parameters
res4.major$CI$Hotelling

# Expected results in res4.minor$Parameters
#          dm          df1          df2       signif            K
# 1.462603730  2.000000000 21.000000000  0.100000000  2.863636364
#           k           T2            F       F.crit       t.crit
# 6.000000000 12.835258028  6.125918604  2.574569390  2.073873068
#         p.F
# 0.008021181

# Results in res4.minor$CI$Hotelling
#              LCL         UCL
# alpha -0.2553037 -0.02814098
# beta  -0.1190028  0.01175691
```

```
# Expected results in res4.major$Parameters
#           dm          df1          df2       signif            K
# 4.508190e+00 2.000000e+00 2.100000e+01 5.000000e-02 2.863636e+00
#            k           T2            F       F.crit       t.crit
# 6.000000e+00 1.219427e+02 5.819992e+01 2.574569e+00 2.073873e+00
#          p.F
# 2.719240e-09

# Expected results in res4.major$CI$Hotelling
#              LCL        UCL
# alpha -0.4864736 -0.2360966
# beta   0.1954760  0.3035340
```

`mimcr` *Model-independent multivariate confidence region (MIMCR) procedure*

Description

The function `mimcr()` assesses the equivalence of highly variable dissolution profiles. It does so by applying different methods proposed in the literature, implementing the non-parametric "Model-Independent Multivariate Confidence Region" (MIMCR) procedure and the "T^2 test for equivalence" of dissolution data as proposed by Hoffelder (2016).

Usage

```
mimcr(
  data,
  tcol,
  grouping,
  fit_n_obs = FALSE,
  mtad = 10,
  signif = 0.05,
  max_trial = 50,
  bounds = c(1, 85),
  nsf = c(1, 2),
  tol = 1e-09
)
```

Arguments

`data`	A data frame with the dissolution profile data in wide format.
`tcol`	A vector of indices that specifies the columns in `data` which contain the % release values. The length of `tcol` must be two or longer.
`grouping`	A character string that specifies the column in `data` that contains the group names (i.e. a factorial variable, e.g., for the differentiation of batches or formulations of a drug product).

`fit_n_obs` A logical value that indicates whether the number of rows per level in the column specified by the `grouping` parameter should be adjusted to be equal given that they are not equal. The default is `FALSE` because for this type of analysis each group should have the same number of observations. If `fit_n_obs` is `TRUE`, redundant observations from the level with more observations are dropped, i.e. only the observations `1:n` (`n`: number of observations of the level with the fewer observations) will be used for the comparison of the two groups.

`mtad` A numeric value that specifies the "maximum tolerable average difference" (MTAD) of the profiles of two formulations at all time points (in %). The default value is `10`. It determines the size of the similarity limit $\boldsymbol{d}_g$ (see the details section for more information).

`signif` A positive numeric value between `0` and `1` that specifies the significance level for the calculation of the "Confidence Region" (CR). The coverage of CR is $(1 - signif)100\%$. The default value is `0.05`.

`max_trial` A positive integer that specifies the maximum number of Newton-Raphson search rounds to be performed.

`bounds` A numeric vector of the form `c(lower, upper)` that specifies the "lower" and "upper" limits, respectively, for the % drug release. The default is `c(1, 85)`. Mean % release values of any of the two groups being compared that are smaller than or equal to the lower bound are ignored and only the first mean % release value that is greater than or equal to the upper bound is included while all the subsequent values are ignored.

`nsf` A vector of positive integers that specify the "number of significant figures" (nsf) of the corresponding values of the `bounds` parameter. It must thus have the same length as the `bounds` parameter. Before the % release values are compared with the limits that are specified by the `bounds` parameter, they are rounded to the corresponding number of significant figures as specified by the `nsf` parameter.

`tol` A non-negative numeric that specifies the accepted minimal difference between two consecutive search rounds.

Details

The function `mimcr()` assesses the equivalence of highly variable dissolution profiles by aid of a "Model-Independent Multivariate Confidence Region" (MIMCR) procedure as proposed by Tsong et al. (1996) and by aid of a "T2 test for equivalence" as proposed by Hoffelder (2016).

For details see the sections "Comparison of highly variable dissolution profiles", "Similarity limits in terms of MSD" and "T2 test for equivalence" below.

Value

An object of class '`mimcr`' is returned, containing the following list elements:

`Similarity` Conclusion concerning similarity.

`Parameters` Parameters calculated during the assessment.

`NR.CI` List with results from the Newton-Raphson (NR) search.

`Profile.TP` A named numeric vector of the columns in `data` specified by `tcol`. Given that the column names contain extractable numeric information, e.g., the testing time points of the dissolution profile, it contains the corresponding numeric values. Elements where no numeric information could be extracted are `NA`.

The `Parameters` element contains the following information:

`dm` The Mahalanobis distance of the samples.

`df1` Degrees of freedom (number of variables or time points).

`df2` Degrees of freedom (number of rows - number of variables - 1).

`alpha` The provided significance level.

`K` Scaling factor for F to account for the distribution of the T^2 statistic.

`k` Scaling factor for the squared Mahalanobis distance to obtain the T^2 statistic.

`T2` Hotelling's T^2 statistic (F-distributed).

`F` Observed F value.

`ncp.Hoffelder` Non-centrality parameter for calculation of the F statistic (T^2 test procedure).

`F.crit` Critical F value (Tsong's procedure).

`F.crit.Hoffelder` Critical F value (T^2 test procedure).

`p.F` The p value for the Hotelling's T^2 test statistic.

`p.F.Hoffelder` The p value for the Hotelling's T^2 statistic based on the non-central F distribution.

`MTAD` Specified "maximum tolerable average difference" (MTAD) of the profiles of two formulations at each individual time point (in %).

`Sim.Limit` Critical Mahalanobis distance or similarity limit (Tsong's procedure).

`Obs.L` Observed lower limit (Tsong's procedure).

`Obs.U` Observed upper limit (Tsong's procedure).

The `NR.CI` element contains the following information:

`CI` A matrix of the points on the *CR* bounds for each time point.

`converged` A logical that indicates whether the NR algorithm converged or not.

`points.on.crb` A logical that indicates whether the points that were found by the NR algorithm sit on the confidence region boundary or not, i.e. whether the T^2 statistic of the found data points, in relation to the mean difference, is equal to the critical F value.

`n.trial` Number of trials until convergence.

`max.trial` Maximal number of trials.

`Warning` A warning message, if applicable, or otherwise NULL.

`Error` An error message, if applicable, or otherwise NULL.

Comparison of highly variable dissolution profiles

When comparing the dissolution data of a post-approval change product and a reference approval product, the goal is to assess the similarity between the mean dissolution values at the observed sample time points. A widely used method is the f_2 method that was introduced by Moore & Flanner (1996). Similarity testing criteria based on f_2 can be found in several FDA guidelines and in the guideline of the European Medicines Agency (EMA) "On the investigation of bioequivalence" (EMA 2010).

In situations where within-batch variation is greater than 15%, FDA guidelines recommend use of a multivariate confidence interval as an alternative to the f_2 method. This can be done using the following stepwise procedure:

1. Establish a similarity limit in terms of "Multivariate Statistical Distance" (MSD) based on inter-batch differences in % drug release from reference (standard approved) formulations, i.e. the so-called "Equivalence Margin" (EM).
2. Calculate the MSD between test and reference mean dissolutions.
3. Estimate the 90% confidence interval (CI) of the true MSD as determined in step 2.
4. Compare the upper limit of the 90% CI with the similarity limit determined in step 1. The test formulation is declared to be similar to the reference formulation if the upper limit of the 90% CI is less than or equal to the similarity limit.

Similarity limits in terms of MSD

For the calculation of the "Multivariate Statistical Distance" (MSD), the procedure proposed by Tsong et al. (1996) can be considered as well-accepted method that is actually recommended by the FDA. According to this method, a multivariate statistical distance, called Mahalanobis distance, is used to measure the difference between two multivariate means. This distance measure is calculated as

$$D_M = \sqrt{(\boldsymbol{x}_T - \boldsymbol{x}_R)^\top \boldsymbol{S}_{pooled}^{-1} (\boldsymbol{x}_T - \boldsymbol{x}_R)},$$

where $\boldsymbol{S}_{pooled}$ is the sample variance-covariance matrix pooled across the comparative groups, $\boldsymbol{x}_T$ and $\boldsymbol{x}_R$ are the vectors of the sample means for the test (T) and reference (R) profiles, and $\boldsymbol{S}_T$ and $\boldsymbol{S}_R$ are the variance-covariance matrices of the test and reference profiles. The pooled variance-covariance matrix $\boldsymbol{S}_{pooled}$ is calculated by

$$\boldsymbol{S}_{pooled} = \frac{(n_R - 1)\boldsymbol{S}_R + (n_T - 1)\boldsymbol{S}_T}{n_R + n_T - 2}.$$

In order to determine the similarity limits in terms of the MSD, i.e. the Mahalanobis distance between the two multivariate means of the dissolution profiles of the formulations to be compared, Tsong et al. (1996) proposed using the equation

$$D_M^{max} = \sqrt{\boldsymbol{d}_g^\top \boldsymbol{S}_{pooled}^{-1} \boldsymbol{d}_g},$$

where $\boldsymbol{d}_g$ is a $1 \times p$ vector with all p elements equal to an empirically defined limit $\boldsymbol{d}_g$, e.g., 15%, for the maximum tolerable difference at all time points, and p is the number of sampling points. By assuming that the data follow a multivariate normal distribution, the 90% confidence region

(*CR*) bounds for the true difference between the mean vectors, $\boldsymbol{\mu}_T - \boldsymbol{\mu}_R$, can be computed for the resultant vector $\boldsymbol{\mu}$ to satisfy the following condition:

$$CR = K\left(\boldsymbol{\mu} - (\boldsymbol{x}_T - \boldsymbol{x}_R)\right)^\top \boldsymbol{S}_{pooled}^{-1}\left(\boldsymbol{\mu} - (\boldsymbol{x}_T - \boldsymbol{x}_R)\right) \leq F_{p,n_T+n_R-p-1,0.9},$$

where K is the scaling factor that is calculated as

$$K = \frac{n_T n_R}{n_T + n_R}\,\frac{n_T + n_R - p - 1}{(n_T + n_R - 2)\,p},$$

and $F_{p,n_T+n_R-p-1,0.9}$ is the 90^{th} percentile of the F distribution with degrees of freedom p and $n_T + n_R - p - 1$, where n_T and n_R are the number of observations of the reference and the test group, respectively, and p is the number of sampling or time points, as mentioned already. It is obvious that $(n_T + n_R)$ must be greater than $(p + 1)$. The formula for *CR* gives a p-variate 90% confidence region for the possible true differences.

T2 test for equivalence

Based on the distance measure for profile comparison that was suggested by Tsong et al. (1996), i.e. the Mahalanobis distance, Hoffelder (2016) proposed a statistical equivalence procedure for that distance, the so-called T^2 test for equivalence (T2EQ). It is used to demonstrate that the Mahalanobis distance between reference and test group dissolution profiles is smaller than the "Equivalence Margin" (EM). Decision in favour of equivalence is taken if the p value of this test statistic is smaller than the pre-specified significance level α, i.e. if $p < \alpha$. The p value is calculated by aid of the formula

$$p = F_{p,n_T+n_R-p-1,ncp,\alpha}\,\frac{n_T + n_R - p - 1}{(n_T + n_R - 2)p}T^2,$$

where α is the significance level and ncp is the so-called "*non-centrality parameter*" that is calculated by

$$\frac{n_T n_R}{n_T + n_R}\left(D_M^{max}\right)^2.$$

The test statistic being used is Hotelling's two-sample T^2 test that is given as

$$T^2 = \frac{n_T n_R}{n_T + n_R}\left(\boldsymbol{x}_T - \boldsymbol{x}_R\right)^\top \boldsymbol{S}_{pooled}^{-1}\left(\boldsymbol{x}_T - \boldsymbol{x}_R\right).$$

As mentioned in paragraph "Similarity limits in terms of MSD", $\boldsymbol{d}_g$ is a $1 \times p$ vector with all p elements equal to an empirically defined limit d_g. Thus, the components of the vector $\boldsymbol{d}_g$ can be interpreted as upper bound for a kind of "*average*" allowed difference between test and reference profiles, the "*global similarity limit*". Since the EMA requires that "similarity acceptance limits should be pre-defined and justified and not be greater than a 10% difference", it is recommended to use 10%, not 15% as proposed by Tsong et al. (1996), for the maximum tolerable difference at all time points.

References

United States Food and Drug Administration (FDA). Guidance for industry: dissolution testing of immediate release solid oral dosage forms. 1997.
https://www.fda.gov/media/70936/download

United States Food and Drug Administration (FDA). Guidance for industry: immediate release solid oral dosage form: scale-up and post-approval changes, chemistry, manufacturing and controls, *in vitro* dissolution testing, and *in vivo* bioequivalence documentation (SUPAC-IR). 1995.
https://www.fda.gov/media/70949/download

European Medicines Agency (EMA), Committee for Medicinal Products for Human Use (CHMP). Guideline on the Investigation of Bioequivalence. 2010; CPMP/EWP/QWP/1401/98 Rev. 1.

Tsong, Y., Hammerstrom, T., Sathe, P.M., and Shah, V.P. Statistical assessment of mean differences between two dissolution data sets. *Drug Inf J*. 1996; **30**: 1105-1112.
doi:10.1177/009286159603000427

Tsong, Y., Hammerstrom, T., and Chen, J.J. Multipoint dissolution specification and acceptance sampling rule based on profile modeling and principal component analysis. *J Biopharm Stat*. 1997; **7**(3): 423-439.
doi:10.1080/10543409708835198

Wellek S. (2010) *Testing statistical hypotheses of equivalence and noninferiority* (2nd ed.). Chapman & Hall/CRC, Boca Raton.
doi:10.1201/EBK1439808184

Hoffelder, T. Highly variable dissolution profiles. Comparison of T^2-test for equivalence and f_2 based methods. *Pharm Ind*. 2016; **78**(4): 587-592.
https://www.ecv.de/suse_item.php?suseId=Z|pi|8430

See Also

gep_by_nera, get_T2_two, get_T2_one, bootstrap_f2, mztia.

Examples

```
# Using the defaults, only profile time points with an average release of >= 1%
# and only one time point with an average release of > 85% are taken into
# account.
res1 <- mimcr(data = dip3, tcol = 4:6, grouping = "batch")
res1$Similarity
res1$Parameters

# Expected results in res1$Similarity
#     Tsong Hoffelder
# "Similar" "Similar"

# Expected results in res1$Parameters
#            DM              df1              df2            alpha
#  2.384023e-01     3.000000e+00     2.000000e+01     5.000000e-02
#             K                k               T2                F
#  1.818182e+00     6.000000e+00     3.410141e-01     1.033376e-01
# ncp.Hoffelder           F.crit F.crit.Hoffelder              p.F
#  3.032296e+01     3.098391e+00     4.899274e+00     9.571526e-01
```

```
# p.F.Hoffelder              MTAD        Sim.Limit            Obs.L
#  2.890827e-08     1.000000e+01     2.248072e+00     1.067015e+00
#         Obs.U
#  1.543820e+00

# Comparison with T2-test for equivalence for dissolution data from the 'T2EQ'
# package
## Not run:
  if (requireNamespace("T2EQ")) {
    library(T2EQ)
    data(ex_data_JoBS)

    T2EQ.dissolution.profiles.hoffelder(
      X = as.matrix(dip3[dip3$type == "ref", c("x.15", "x.20", "x.25")]),
      Y = as.matrix(dip3[dip3$type == "test", c("x.15", "x.20", "x.25")]))
  }

  # Excerpt of output:
  # Hotelling's T2:                        0.3410141
  # Noncentrality parameter:                    30.32296
  # Significance level:                      0.05
  # Teststatistic:                           0.1033376
  # Quantile of noncent. F-distribution:        4.899274
  # p-value of the T2-test for equivalence: p = 2.890827e-08

## End(Not run)

# Use of 'bounds = c(1, 85)'
res2 <- mimcr(data = dip1, tcol = 3:10, grouping = "type", bounds = c(1, 85),
              nsf = c(1, 2))
res2$Similarity
res2$Profile.TP
res2[["Parameters"]][c("p.F.Hoffelder", "Sim.Limit", "Obs.U")]

# Expected results in res2$Similarity
#        Tsong    Hoffelder
# "Dissimilar" "Dissimilar"

# Expected results in res2$Profile.TP
# t.5 t.10 t.15 t.20 t.30 t.60 t.90
#   5   10   15   20   30   60   90

# Expected results in res2$Parameters
# res2[["Parameters"]][c("p.F.Hoffelder", "Sim.Limit", "Obs.U")]
# p.F.Hoffelder     Sim.Limit         Obs.U
#      0.740219     11.328041     31.679020

# Allow for a larger maximum tolerable average difference (MTAD), e.g., 15.
res3 <- mimcr(data = dip1, tcol = 3:10, grouping = "type", mtad = 15,
              bounds = c(1, 85), nsf = c(1, 2))
res3$Similarity
res3[["Parameters"]][c("p.F.Hoffelder", "Sim.Limit", "Obs.U")]
```

```
# Expected results in res3$Similarity
#        Tsong    Hoffelder
# "Dissimilar" "Dissimilar"

# Expected results in res3$Parameters
# res3[["Parameters"]][c("p.F.Hoffelder", "Sim.Limit", "Obs.U")]
# p.F.Hoffelder     Sim.Limit         Obs.U
#     0.3559019    16.9920622    31.6790198

# Use default 'mtad' but set 'signif = 0.1' and use 'bounds = c(1, 95)' so that
# the complete profiles are taken into account.
res4 <- mimcr(data = dip1, tcol = 3:10, grouping = "type", mtad = 10,
              signif = 0.1, bounds = c(1, 95), nsf = c(1, 2))
res4$Similarity
res4$Profile.TP
res4[["Parameters"]][c("p.F.Hoffelder", "Sim.Limit", "Obs.U")]

# Expected results in res4$Similarity
#        Tsong    Hoffelder
# "Dissimilar" "Dissimilar"

# Expected results in res4$Profile.TP
# t.5  t.10  t.15  t.20  t.30  t.60  t.90 t.120
#   5    10    15    20    30    60    90   120

# Expected results in res4$Parameters
# res2[["Parameters"]][c("p.F.Hoffelder", "Sim.Limit", "Obs.U")]
# p.F.Hoffelder     Sim.Limit         Obs.U
#     0.1449045    19.4271898    33.3180044

## Not run:
  # If 'max_trial' is too small, the Newton-Raphson search may not converge.
  tryCatch(
    mimcr(data = dip1, tcol = 3:10, grouping = "type", max_trial = 5),
    warning = function(w) message(w),
    finally = message("\nMaybe increasing the number of max_trial could help."))

  # If 'tol' is too big, the points found by the Newton-Raphson search may not
  # be located on the confidence region boundary.
  tryCatch(
    mimcr(data = dip3, tcol = 4:6, grouping = "batch", tol = 1),
    warning = function(w) message(w),
    finally = message("\nMaybe making tol smaller could help."))

  # Passing in a data frame with a grouping variable with a number of levels
  # that differs from two produces an error.
  tmp <- rbind(dip1,
               data.frame(type = "T2",
                          tablet = as.factor(1:6),
                          dip1[7:12, 3:10]))

  tryCatch(
    mimcr(data = tmp, tcol = 3:10, grouping = "type", bounds = c(1, 85)),
```

```
    error = function(e) message(e),
    finally = message("\nMaybe you want to remove unesed levels in data."))

  # Error in mimcr(data = tmp, tcol = 3:10, grouping = "type", bounds = ,  :
  #   The number of levels in column type differs from 2.

## End(Not run)
```

`mztia` *Martinez & Zhao Tolerance Interval Approach*

Description

The *Martinez & Zhao Tolerance Interval Approach* (`mztia`) is a simple approach for the comparison of dissolution profiles. The `mztia()` function calculates tolerance intervals (*TI*) at each time point of the dissolution profiles of a set of reference batches. By aid of a graphical display the test batches are checked to lie within the *TI* boundaries or within certain limits exceeding the *TI* boundaries by a specified percentage.

Usage

```
mztia(
  data,
  shape,
  tcol,
  grouping,
  reference,
  response = NULL,
  na_rm = FALSE,
  alpha = 0.05,
  pp = 0.99,
  cap = TRUE,
  bounds = c(0, 100),
  qs = c(5, 15),
  ...
)
```

Arguments

`data`	A data frame with the dissolution profile data in wide or in long format (see parameter `shape`). If the data frame is in wide format, it is tried to extract the information on the time points of dissolution testing from the column names of the columns specified by the `tcol` parameter. Thus, they must contain extractable numeric information, e.g., `(t_0, t_5, t_10)`. If the data frame is in long format, it must have a column of time points (column specified via the `tcol` parameter).
`shape`	A character string that indicates whether the data frame is in long or in wide format.

tcol	If shape is "wide" an integer vector of indices, if shape is "long" an integer that specifies the column(s) containing the profile time points. If the data frame is in wide format it is reshaped using the function reshape() from the 'stats' package.
grouping	A character string that specifies the column in data that contains the group names (i.e. a factorial variable, e.g., for the differentiation of batches or formulations of a drug product).
reference	A character string that specifies the name of the reference group from the grouping variable.
response	A character string that is expected if data is provided in long format to specify the column with the % drug release values. The default is NULL.
na_rm	A logical value that indicates whether observations containing NA (or NaN) values should be removed (na_rm = TRUE) or not (na_rm = FALSE). The default is na_rm = FALSE.
alpha	A numeric value between 0 and 1 that specifies the probability level. The default is 0.05.
pp	A numeric value between 0 and 1 that specifies the proportion of the population being enclosed by the tolerance interval boundaries. The default is 0.99.
cap	A logical variable that indicates whether the calculated tolerance limits should be limited (i.e. *cap*ped). The default is TRUE.
bounds	A numeric vector of the form c(lower, upper) that specifies the "lower" and "upper" limits, respectively, for the % drug release at which the calculated tolerance interval limits should be capped (see parameter *cap*. This parameter is only relevant if cap = TRUE. The default is c(0, 100).
qs	A numeric vector of the form c(Q S1, Q S2) that specifies the allowable deviations from the specifications in percent according to the $S1$ and $S2$ acceptance criteria of USP chapter <711> on dissolution. The default is c(5, 15).
...	Further arguments passed on to the reshape() from the 'stats' package.

Details

The tolerance interval approach proposed by Martinez & Zhao (2018) is a simple approach for the comparison of dissolution profiles. The authors propose to calculate for each time point of a set of reference dissolution profiles a tolerance interval (*TI*), i.e. intervals containing $pp\%$ of the population of potential values for reference product at a probability level of $alpha/2$ per tail (i.e., $(1 - alpha)100\%$ confidence). Based on these *TI*s the dissolution profiles of the test batch(es) is (are) compared, i.e. the corresponding data points should lie within the *TI*s. The *TI*s are calculated as

$$Y_{utl,ltl} = \bar{Y} \pm k \times s$$

where $\bar{Y}$ is the average, s is the sample standard deviation, and the factor k is calculated according to Hahn (Hahn & Meeker (1991)), as proposed in Martinez & Zhao (2018).

Since the goal of the comparison is not to confirm simply "*statistical sameness*" but "product comparability", Martinez & Zhao propose allowing acceptable deviations by utilizing the concepts described by the United States Pharmacopoeia (USP), chapter <711> on dissolution, defining *allowable deviations from a set of product specifications* (Q). The *TI*s serve as the target value Q at each

sampling time. The allowable deviations about Q are defined by the $S1$ and $S2$ acceptance criteria of USP chapter <711> on dissolution:

1. The $S1$ level boundary is defined by $Q \pm 5\%$ at each time point. For every 12 profiles tested, only one profile is allowed to exceed the $S1$ bounds.
2. The $S2$ level boundary is defined by $Q \pm 15\%$ at each time point. No observation from any of the test dissolution profiles is allowed to exceed the $S2$ bounds.

In situations where the reference formulation itself has more than one of twelve observations (profiles) exceeding $S1$ at one or more time points, additional runs of the reference product must be performed. It is deemed appropriate to use the same values of $S1$ and $S2$ across all time points because the high variability associated with the early sampling times is already factored into the *TI*s.

TI calculation according to Hahn is proposed because it appeared to be more numerically stable and gave more consistent *TI*s than the *TI* calculation method proposed by Howe (Howe 1969) when samples were very variable. The reason might be due to the less stringent requirements imposed by Hahn's method with respect to the normality of the data.

Value

An object of class '`mztia`' is returned, containing the following elements:

`Variables`	A list of the variables and the corresponding values.
`Limits`	A data frame of the limits calculated for each time point.
`Data`	A data frame consisting of the provided data, complemented by the calculated tolerance interval results.
`Profile.TP`	If `shape` is `"wide"` a named numeric vector of the columns in `data` specified by `tcol`. Given that the column names contain extractable numeric information, e.g., the testing time points of the dissolution profile, it contains the corresponding numeric values. Elements where no numeric information could be extracted are `NA`. If `shape` is `"long"` it is a numeric value that specifies the column containing the % release values.

References

Martinez, M.N., and Zhao, X. A simple approach for comparing the *in vitro* dissolution profiles of highly variable drug products: a proposal. *AAPS Journal*. 2018; **20**: 78.
doi:10.1208/s1224801802381

Howe, W.G. Two-sided tolerance limits for normal populations - some improvements. *J Am Stat Assoc*. 1969; **64**: 610-620.
doi:10.1080/01621459.1969.10500999

Hahn, G.J., and Meeker, W. Q. Statistical intervals: A guide for practitioners. (1991); John Wiley & Sons, New York. Hahn's method is also described in: SAS/QC 13.1: User's Guide. Chapter 5, sub-chapter "Details: INTERVALS Statement", pp 421-424. SAS Institute Inc. 2013. Cary, NC.
`https://support.sas.com/documentation/cdl/en/qcug/66857/PDF/default/qcug.pdf`

U.S. Pharmacopoeia. 2016 U.S. Pharmacopoeia-National Formulary (USP 39 NF 34). Volume 1. Rockville, Md: United States Pharmacopeial Convention, Inc; 2015. <711> Dissolution.

See Also

bootstrap_f2, mimcr.

Examples

```
# Calculation of tolerance intervals
m_alpha_P <- matrix(c(rep(c(0.01, 0.05, 0.1), each = 3),
                      1 - rep(c(0.1, 0.05, 0.01), times = 3)),
                    ncol = 2, byrow = FALSE)

ll <-
  apply(m_alpha_P, MARGIN = 1, FUN = function(x)
    mztia(data = dip5, shape = "long", tcol = 1, grouping = "type",
          reference = "reference", response = "weight", na_rm = FALSE,
          alpha = x[1], P = x[2], cap = FALSE)[["Data"]][102, "weight"])
ul <-
  apply(m_alpha_P, MARGIN = 1, FUN = function(x)
    mztia(data = dip5, shape = "long", tcol = 1, grouping = "type",
          reference = "reference", response = "weight", na_rm = FALSE,
          alpha = x[1], P = x[2], cap = FALSE)[["Data"]][103, "weight"])

# Expected results in ll and ul
rbind(ll, ul)
#        [,1]    [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
# ll 11.91648 11.8987 11.86395 11.92132 11.90446 11.87152 11.92373 11.90734
# ul 12.10212 12.1199 12.15465 12.09728 12.11414 12.14708 12.09487 12.11126
#       [,9]
# ll 11.8753
# ul 12.1433

# Use a data frame in wide format
# Using the defaults; Limits are capped to the range specified by 'bounds'
res1 <- mztia(data = dip1, shape = "wide", tcol = 3:10, grouping = "type",
              reference = "R")
res1$Limits

# Expected results in res1$Limits
# Time     Mean      LTL       UTL   S1.LTL    S1.UTL   S2.LTL    S2.UTL
# 1    5 46.77167 27.22641  66.31693 22.22641  71.31693 12.22641  81.31693
# 2   10 60.13333 46.15483  74.11184 41.15483  79.11184 31.15483  89.11184
# 3   15 67.27500 56.90417  77.64583 51.90417  82.64583 41.90417  92.64583
# 4   20 71.98667 65.44354  78.52979 60.44354  83.52979 50.44354  93.52979
# 5   30 78.07000 69.54259  86.59741 64.54259  91.59741 54.54259 101.59741
# 6   60 84.81667 77.20275  92.43058 72.20275  97.43058 62.20275 107.43058
# 7   90 89.09333 76.24588 100.00000 71.24588 105.00000 61.24588 115.00000
# 8  120 91.43833 80.29321 100.00000 75.29321 105.00000 65.29321 115.00000

# Without capping of limits to 105%
res2 <- mztia(data = dip1, shape = "wide", tcol = 3:10, grouping = "type",
              reference = "R", cap = FALSE)
res2$Limits
```

```
# Expected results in res1$Limits
# Time     Mean      LTL       UTL   S1.LTL    S1.UTL   S2.LTL    S2.UTL
# 1    5 46.77167 27.22641  66.31693 22.22641  71.31693 12.22641  81.31693
# 2   10 60.13333 46.15483  74.11184 41.15483  79.11184 31.15483  89.11184
# 3   15 67.27500 56.90417  77.64583 51.90417  82.64583 41.90417  92.64583
# 4   20 71.98667 65.44354  78.52979 60.44354  83.52979 50.44354  93.52979
# 5   30 78.07000 69.54259  86.59741 64.54259  91.59741 54.54259 101.59741
# 6   60 84.81667 77.20275  92.43058 72.20275  97.43058 62.20275 107.43058
# 7   90 89.09333 76.24588 101.94079 71.24588 106.94079 61.24588 116.94079
# 8  120 91.43833 80.29321 102.58346 75.29321 107.58346 65.29321 117.58346

# Tolerance intervals are calculated exclusively for the level of the
# grouping variable that is specified by the reference variable. Therefore,
# the following code produces the same limits summary as in res2$Limits.
tmp <- rbind(dip1,
             data.frame(type = "T2",
                        tablet = as.factor(1:6),
                        dip1[7:12, 3:10]))

res2 <- mztia(data = dip1, shape = "wide", tcol = 3:10, grouping = "type",
              reference = "R", cap = FALSE)
res3 <- mztia(data = tmp, shape = "wide", tcol = 3:10, grouping = "type",
              reference = "R", cap = FALSE)

isTRUE(all.equal(res2$Limits, res3$Limits))
# [1] TRUE
```

`plot.bootstrap_f2` *Plot of the bootstrap f2 simulation*

Description

This is a method for the function `plot()` for objects of class '`bootstrap_f2`'.

Usage

```
## S3 method for class 'bootstrap_f2'
plot(x, ...)
```

Arguments

`x`	An object of class '`bootstrap_f2`' returned by the `bootstrap_f2()` function.
`...`	Further arguments passed to or from other methods or arguments that can be passed down to the `plot.boot()` function.

Details

The element Boot of the 'bootstrap_f2' object that is returned by the function bootstrap_f2() is an object of class 'boot', generated by the function boot() from the 'boot' package. Thus, the corresponding plot method is used. Arguments to the plot.boot() function can be passed via the ... parameter. In addition to making the plot the function prints the result of Shah's lower 90% BCa confidence interval to the console.

Value

The 'bootstrap_f2' object passed to the x parameter is returned invisibly.

See Also

bootstrap_f2, boot, plot.boot, methods.

Examples

```
# Bootstrap assessment of data (two groups) by aid of bootstrap_f2() function
# by using 'rand_mode = "complete"' (the default, randomisation of complete
# profiles)
bs1 <- bootstrap_f2(data = dip2[dip2$batch %in% c("b0", "b4"), ],
                    tcol = 5:8, grouping = "batch", rand_mode = "complete",
                    rr = 200, new_seed = 421, use_ema = "no")

## Not run:
  pbs1 <- plot(bs1)

  # The plot() function returns the 'plot_mztia' object invisibly.
  class(bs1)
  class(pbs1)

## End(Not run)

# Use of 'rand_mode = "individual"' (randomisation per time point)
bs2 <- bootstrap_f2(data = dip2[dip2$batch %in% c("b0", "b4"), ],
                    tcol = 5:8, grouping = "batch", rand_mode = "individual",
                    rr = 200, new_seed = 421, use_ema = "no")

## Not run:
  plot(bs2)

## End(Not run)
```

---

plot.plot_mztia *Plot of the mztia simulation*

Description

This is a method for the function plot() for objects of class 'plot_mztia'.

Usage

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

Arguments

x	An object of class 'plot_mztia' returned by the plot_mztia() function.
...	Further arguments passed to or from other methods or arguments that can be passed down to the plot.boot() function.

Details

The element Graph of the 'plot_mztia' object that is returned by the function plot_mztia() is an object of class 'ggplot', generated by the function ggplot() from the 'ggplot2' package. Thus, the corresponding plot method is used for plotting. Arguments to the ggplot() function can be passed via the ... parameter.

Value

The 'plot_mztia' object passed to the x parameter is returned invisibly.

See Also

mztia, plot_mztia, ggplot(), methods.

Examples

```
# Assessment of data by aid of the mztia() function
res1 <- mztia(data = dip1, shape = "wide", tcol = 3:10, grouping = "type",
              reference = "R", cap = FALSE)

# The 'mztia' object can be passed on to the plot_mztia() function. This
# function does not produce any output but returns a 'plot_mztia' object.
## Not run:
  gg1 <- plot_mztia(res1)
  gg2 <- plot(gg1)

  # The plot() function returns the 'plot_mztia' object invisibly.
  class(gg1)
  class(gg2)

## End(Not run)
```

plot_mztia *Graphical representation of the of MZTIA estimation*

Description

The function `plot_mztia()` makes a graphical representation of the estimates done by the `mztia()` function.

Usage

```
plot_mztia(x, ...)
```

Arguments

`x`	An object of class 'mztia' returned by the `mztia()` function.
`...`	Additional parameters that can be passed on to the `ggplot()` function.

Details

A graphical representation of the information in the `Data` element of the object that is returned by `mztia()` function is made by aid of the `ggplot()` function from the 'ggplot2' package and added as new list element to the `mztia` object. Ideally, the data frame provided to the `mztia()` function allows drawing a time course of the % drug release values. If a single time point is available, the tolerance intervals of the groups specified by the `grouping` parameter (e.g., for the differentiation of batches or formulations of a drug product) are displayed.

Value

An object of class 'plot_mztia' is returned invisibly, consisting of the elements of the 'mztia' object and an additional element named `Graph`. The element `Graph` is a 'ggplot' object returned by calling the `ggplot()` function.

See Also

`mztia`, `ggplot`.

Examples

```
# Analyse the data by aid of the mztia() function.
res1 <- mztia(data = dip1, shape = "wide", tcol = 3:10, grouping = "type",
              reference = "R", cap = FALSE)

# The 'mztia' object can be passed on to the plot_mztia() function. This
# function does not produce any output. It returns a 'plot_mztia' object that
# is essentially an 'mztia' object augmented by a 'ggplot' object.
## Not run:
  gg1 <- plot_mztia(res1)
  gg1
```

```
## End(Not run)

# Since the element gg1$Graph is a 'ggplot' object it can be used for further
# manipulation by aid of 'ggplot2' functions.
## Not run:
  if (requireNamespace("ggplot2")) {
    library(ggplot2)

    gg1$Graph + labs(title = "Dissolution Data Assessment",
                     x = "Time [min]", y = "Drug Release [%]")
  }

## End(Not run)

# Use a data frame in long format.
res2 <- mztia(data = dip5, shape = "long", tcol = 3, grouping = "type",
              reference = "reference", response = "weight", cap = FALSE,
              QS = c(5, 15) / 100)

## Not run:
  gg2 <- plot_mztia(res2)
  gg2

  if (requireNamespace("ggplot2")) {
    library(ggplot2)

    gg2$Graph + labs(title = "Tolerance Intervals",
                     x = NULL, y = "Weight [ounces]")
  }

## End(Not run)
```

`print.bootstrap_f2` *Print a summary of the bootstrap f2 simulation*

Description

This is a method for the function `print()` for objects of class '`bootstrap_f2`'.

Usage

```
## S3 method for class 'bootstrap_f2'
print(x, ...)
```

Arguments

`x`	An object of class '`bootstrap_f2`' returned by the `bootstrap_f2()` function.
`...`	Further arguments passed to or from other methods or arguments that can be passed down to the `print.boot()` and `print.bootci()` functions.

Details

The elements Boot and CI of the 'bootstrap_f2' object that is returned by the function bootstrap_f2() are objects of type 'boot' and 'bootci', respectively, generated by the functions boot() and boot.ci(), respectively, from the 'boot' package. Thus, the corresponding print methods are used. Arguments to the print.boot() and print.bootci() functions can be passed via the ... parameter.

Value

The 'bootstrap_f2' object passed to the x parameter is returned invisibly.

See Also

bootstrap_f2, boot, boot.ci, print.boot, print.bootci, methods.

Examples

```
# Bootstrap assessment of data (two groups) by aid of bootstrap_f2() function
# by using 'rand_mode = "complete"' (the default, randomisation of complete
# profiles)
bs1 <- bootstrap_f2(data = dip2[dip2$batch %in% c("b0", "b4"), ],
                    tcol = 5:8, grouping = "batch", rand_mode = "complete",
                    rr = 200, new_seed = 421, use_ema = "no")

# Print of a summary of the assessment
print(bs1)

# STRATIFIED BOOTSTRAP
#
#
# Call:
#   boot(data = data, statistic = get_f2, R = R, strata = data[, grouping],
#        grouping = grouping, tcol = tcol[ok])
#
#
# Bootstrap Statistics :
#   original      bias    std. error
# t1* 50.07187 -0.02553234   0.9488015
#
#
# BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
# Based on 200 bootstrap replicates
#
# CALL :
#   boot.ci(boot.out = t_boot, conf = confid, type = "all", L = jack$loo.values)
#
# Intervals :
#   Level      Normal              Basic
# 90%   (48.54, 51.66 )   (48.46, 51.71 )
#
# Level     Percentile            BCa
```

```
# 90%   (48.43, 51.68 )   (48.69, 51.99 )
# Calculations and Intervals on Original Scale
# Some BCa intervals may be unstable
#
#
# Shah's lower 90% BCa confidence interval:
#  48.64613

# Use of 'rand_mode = "individual"' (randomisation per time point)
bs2 <- bootstrap_f2(data = dip2[dip2$batch %in% c("b0", "b4"), ],
                    tcol = 5:8, grouping = "batch", rand_mode = "individual",
                    rr = 200, new_seed = 421, use_ema = "no")

# Print of a summary of the assessment
print(bs2)

# PARAMETRIC BOOTSTRAP
#
#
# Call:
#   boot(data = data, statistic = get_f2, R = R, sim = "parametric",
#        ran.gen = rand_indiv_points, mle = mle, grouping = grouping,
#        tcol = tcol[ok], ins = seq_along(b1))
#
#
# Bootstrap Statistics :
#   original     bias    std. error
# t1* 50.07187 -0.1215656   0.9535517
#
#
# BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
# Based on 200 bootstrap replicates
#
# CALL :
#   boot.ci(boot.out = t_boot, conf = confid, type = "all", L = jack$loo.values)
#
# Intervals :
#   Level      Normal              Basic
# 90%   (48.62, 51.76 )   (48.44, 51.64 )
#
# Level     Percentile            BCa
# 90%   (48.50, 51.70 )   (48.88, 52.02 )
# Calculations and Intervals on Original Scale
# Some BCa intervals may be unstable
#
#
# Shah's lower 90% BCa confidence interval:
#  48.82488
```

`print.mimcr` *Print a summary of MIMCR estimation*

Description

This is a method for the function `print()` for objects of class '`mimcr`'.

Usage

```
## S3 method for class 'mimcr'
print(x, ...)
```

Arguments

`x`	An object of class '`mimcr`' returned by the `mimcr()` function.
`...`	Further arguments passed to or from other methods or arguments that can be passed down to the `formatC()` function.

Details

The most relevant information in an '`mimcr`' object is printed.

Value

The '`mimcr`' object passed to the `x` parameter is returned invisibly.

See Also

`mimcr`, `formatC`, `methods`.

Examples

```
# Assessment of data by aid of the mimcr() function
res1 <- mimcr(data = dip1, tcol = 3:10, grouping = "type")

# Print of a summary of the assessment
print(res1)

# Results of Model-Independent Multivariate Confidence Region (MIMCR)
# approach to assess equivalence of highly variable in-vitro
# dissolution profiles of two drug product formulations
#
# Did the Newton-Raphson search converge? Yes
#
# Parameters (general):
#   Significance level:                0.05
# Degrees of freedom (1):              7
# Degrees of freedom (2):              4
# Mahalanobis distance (MD):           25.72
# (F) scaling factor K:                0.1714
# (MD) scaling factor k:               3
# Hotelling's T2:                      1984
#
# Parameters specific for Tsong (1996) approach:
# Maximum tolerable average difference: 10
```

```
# Similarity limit:                     11.33
# Observed upper limit:                 31.68
#
# Parameters specific for Hoffelder (2016) approach:
# Noncentrality parameter:              385
# Critial F (Hoffelder):                23.16
# Probability p (Hoffelder):            0.7402
#
# Conclusions:
#       Tsong (1996):  Dissimilar
#   Hoffelder (2016):  Dissimilar

# Taking only the 15 and 90 minutes testing points into account produces a
# warning because profiles should comprise a minimum of three testing points.
## Not run:
  res2 <- mimcr(data = dip1, tcol = c(5, 9), grouping = "type", mtad = 15,
                signif = 0.1)
  print(res2)

  # Warning:
  #   In mimcr(data = dip1, tcol = c(5, 9), grouping = "type", mtad = 15,  :
  # The profiles should comprise a minimum of 3 time points. The actual profiles
  # comprise 2 points only.

  # Results of Model-Independent Multivariate Confidence Region (MIMCR)
  # approach to assess equivalence of highly variable in-vitro
  # dissolution profiles of two drug product formulations
  #
  # Did the Newton-Raphson search converge? Yes
  #
  # Parameters (general):
  #   Significance level:                 0.1
  # Degrees of freedom (1):               2
  # Degrees of freedom (2):               9
  # Mahalanobis distance (MD):            10.44
  # (F) scaling factor K:                 1.35
  # (MD) scaling factor k:                3
  # Hotelling's T2:                       327
  #
  # Parameters specific for Tsong (1996) approach:
  # Maximum tolerable average difference: 15
  # Similarity limit:                     9.631
  # Observed upper limit:                 11.93
  #
  # Parameters specific for Hoffelder (2016) approach:
  # Noncentrality parameter:              278.3
  # Critial F (Hoffelder):                83.57
  # Probability p (Hoffelder):            0.4823
  #
  # Conclusions:
  #       Tsong (1996):  Dissimilar
  #   Hoffelder (2016):  Dissimilar
```

```
## End(Not run)

# A successful comparison:
res3 <- mimcr(data = dip3, tcol = 4:6, grouping = "batch")
print(res3)

# Results of Model-Independent Multivariate Confidence Region (MIMCR)
# approach to assess equivalence of highly variable in-vitro
# dissolution profiles of two drug product formulations
#
# Did the Newton-Raphson search converge? Yes
#
# Parameters (general):
#   Significance level:                 0.05
# Degrees of freedom (1):               3
# Degrees of freedom (2):               20
# Mahalanobis distance (MD):            0.2384
# (F) scaling factor K:                 1.818
# (MD) scaling factor k:                6
# Hotelling's T2:                       0.341
#
# Parameters specific for Tsong (1996) approach:
# Maximum tolerable average difference: 10
# Similarity limit:                     2.248
# Observed upper limit:                 1.544
#
# Parameters specific for Hoffelder (2016) approach:
# Noncentrality parameter:              30.32
# Critial F (Hoffelder):                4.899
# Probability p (Hoffelder):            2.891e-08
#
# Conclusions:
#       Tsong (1996):  Similar
#   Hoffelder (2016):  Similar
```

`print.mztia` *Print a summary of MZTIA estimation*

Description

This is a method for the function `print()` for objects of class '`mztia`'.

Usage

```
## S3 method for class 'mztia'
print(x, ...)
```

Arguments

x	An object of class 'mztia' returned by the mztia() function.
...	Further arguments passed to or from other methods or arguments that can be passed down to the print.data.frame() function.

Details

The "limits" subset (see column "frame") of the data frame that is contained in the "Data" element of the 'mztia' object is printed.

Value

The 'mztia' object passed to the x parameter is returned invisibly.

See Also

mztia, print.data.frame, methods.

Examples

```
# Assessment of data (in wide format) by aid of the mztia() function
res1 <- mztia(data = dip1, shape = "wide", tcol = 3:10, grouping = "type",
              reference = "R", cap = FALSE)

# Print of a summary of the assessment
print(res1)

# Results of Martinez & Zhao Tolerance Interval (TI) Approach
# (TI limits calculated at each time point of the dissolution profiles of a set
# of reference batches)
#
# Time     Mean      LTL       UTL   S1.LTL    S1.UTL   S2.LTL    S2.UTL
# 1    5 46.77167 27.22641  66.31693 22.22641  71.31693 12.22641  81.31693
# 2   10 60.13333 46.15483  74.11184 41.15483  79.11184 31.15483  89.11184
# 3   15 67.27500 56.90417  77.64583 51.90417  82.64583 41.90417  92.64583
# 4   20 71.98667 65.44354  78.52979 60.44354  83.52979 50.44354  93.52979
# 5   30 78.07000 69.54259  86.59741 64.54259  91.59741 54.54259 101.59741
# 6   60 84.81667 77.20275  92.43058 72.20275  97.43058 62.20275 107.43058
# 7   90 89.09333 76.24588 101.94079 71.24588 106.94079 61.24588 116.94079
# 8  120 91.43833 80.29321 102.58346 75.29321 107.58346 65.29321 117.58346
#
# Abbreviations:
#   TL: Tolerance Interval Limit (TL); LTL: lower TL; UTL: upper TL;
#   S1: level 1 boundary (LTL - 5) or (UTL + 5); S2: level 2 boundary
#   (LTL - 15) or (UTL + 15).

# Assessment of data (in long format) by aid of the mztia() function
res2 <- mztia(data = dip5, shape = "long", tcol = 3, grouping = "type",
              reference = "reference", response = "weight", cap = FALSE,
              QS = c(5, 15) / 100)
```

```
# Print of a summary of the assessment
print(res2)

# Results of Martinez & Zhao Tolerance Interval (TI) Approach
# (TI limits calculated at each time point of the dissolution profiles of a set
# of reference batches)
#
# Time    Mean      LTL      UTL   S1.LTL   S1.UTL   S2.LTL   S2.UTL
# 1    1 12.0093 11.87152 12.14708 11.82152 12.19708 11.72152 12.29708
#
# Abbreviations:
#   TL: Tolerance Interval Limit (TL); LTL: lower TL; UTL: upper TL;
#   S1: level 1 boundary (LTL - 0.05) or (UTL + 0.05); S2: level 2 boundary
#   (LTL - 0.15) or (UTL + 0.15).
```

`print.plot_mztia` *Print a plot of MZTIA estimation*

Description

This is a method for the function `print()` for objects of class '`plot_mztia`'.

Usage

```
## S3 method for class 'plot_mztia'
print(x, ...)
```

Arguments

`x`	An object of class '`plot_mztia`' returned by the `plot_mztia()` function.
`...`	Further arguments passed to or from other methods or arguments that can be passed down to the `plot.boot()` function.

Details

The element `Graph` of the '`plot_mztia`' object that is returned by the function `plot_mztia()` is an object of class '`ggplot`', generated by the function `ggplot()` from the '`ggplot2`' package. Thus, the corresponding `plot` method is used for plotting. Arguments to the `ggplot()` function can be passed via the `...` parameter.

Value

The '`plot_mztia`' object passed to the `x` parameter is returned invisibly.

See Also

`mztia`, `plot_mztia`, `ggplot()`, `methods`.

Examples

```
# Assessment of data by aid of the mztia() function
res1 <- mztia(data = dip1, shape = "wide", tcol = 3:10, grouping = "type",
              reference = "R", cap = FALSE)

# The 'mztia' object can be passed on to the plot_mztia() function. This
# function does not produce any output but returns a 'plot_mztia' object.
## Not run:
  gg1 <- plot_mztia(res1)
  gg2 <- print(gg1)

  # The print() function returns the 'plot_mztia' object invisibly.
  class(gg1)
  class(gg2)

## End(Not run)
```

---

summary.bootstrap_f2 *Summary of the bootstrap f2 simulation*

Description

This is a method for the function `summary()` for objects of class '`bootstrap_f2`'.

Usage

```
## S3 method for class 'bootstrap_f2'
summary(object, ...)
```

Arguments

`object`	An object of class '`bootstrap_f2`' returned by the `bootstrap_f2()` function.
`...`	Further arguments passed to or from other methods or arguments that can be passed down to the `print.boot()` and `print.bootci()` functions.

Details

The elements `Boot` and `CI` of the '`bootstrap_f2`' object that is returned by the function `bootstrap_f2()` are objects of type '`boot`' and '`bootci`', respectively, generated by the functions `boot()` and `boot.ci()`, respectively, from the '`boot`' package. Thus, the corresponding `print` methods are used. Arguments to the `print.boot()` and `print.bootci()` functions can be passed via the `...` parameter.

Value

The '`bootstrap_f2`' object passed to the `object` parameter is returned invisibly.

See Also

bootstrap_f2, boot, boot.ci, print.boot, print.bootci, methods.

Examples

```
# Bootstrap assessment of data (two groups) by aid of bootstrap_f2() function
# by using 'rand_mode = "complete"' (the default, randomisation of complete
# profiles)
bs1 <- bootstrap_f2(data = dip2[dip2$batch %in% c("b0", "b4"), ],
                    tcol = 5:8, grouping = "batch", rand_mode = "complete",
                    rr = 200, new_seed = 421, use_ema = "no")

# Summary of the assessment
summary(bs1)

# STRATIFIED BOOTSTRAP
#
#
# Call:
#   boot(data = data, statistic = get_f2, R = R, strata = data[, grouping],
#        grouping = grouping, tcol = tcol[ok])
#
#
# Bootstrap Statistics :
#   original      bias    std. error
# t1* 50.07187 -0.02553234   0.9488015
#
#
# BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
# Based on 200 bootstrap replicates
#
# CALL :
#   boot.ci(boot.out = t_boot, conf = confid, type = "all", L = jack$loo.values)
#
# Intervals :
#   Level      Normal              Basic
# 90%   (48.54, 51.66 )   (48.46, 51.71 )
#
# Level     Percentile            BCa
# 90%   (48.43, 51.68 )   (48.69, 51.99 )
# Calculations and Intervals on Original Scale
# Some BCa intervals may be unstable
#
#
# Shah's lower 90% BCa confidence interval:
#  48.64613

# Use of 'rand_mode = "individual"' (randomisation per time point)
bs2 <- bootstrap_f2(data = dip2[dip2$batch %in% c("b0", "b4"), ],
                    tcol = 5:8, grouping = "batch", rand_mode = "individual",
                    rr = 200, new_seed = 421, use_ema = "no")
```

```
# Summary of the assessment
summary(bs2)

# PARAMETRIC BOOTSTRAP
#
#
# Call:
#   boot(data = data, statistic = get_f2, R = R, sim = "parametric",
#        ran.gen = rand_indiv_points, mle = mle, grouping = grouping,
#        tcol = tcol[ok], ins = seq_along(b1))
#
#
# Bootstrap Statistics :
#   original     bias    std. error
# t1* 50.07187 -0.1215656   0.9535517
#
#
# BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
# Based on 200 bootstrap replicates
#
# CALL :
#   boot.ci(boot.out = t_boot, conf = confid, type = "all", L = jack$loo.values)
#
# Intervals :
#   Level      Normal              Basic
# 90%   (48.62, 51.76 )   (48.44, 51.64 )
#
# Level     Percentile            BCa
# 90%   (48.50, 51.70 )   (48.88, 52.02 )
# Calculations and Intervals on Original Scale
# Some BCa intervals may be unstable
#
#
# Shah's lower 90% BCa confidence interval:
#  48.82488
```

`summary.mimcr` *Summary of MIMCR estimation*

Description

This is a method for the function `summary()` for objects of class '`mimcr`'.

Usage

```
## S3 method for class 'mimcr'
summary(object, ...)
```

Arguments

object	An object of class 'mimcr' returned by the mimcr() function.
...	Further arguments passed to or from other methods or arguments that can be passed down to the formatC() function.

Details

The most relevant information in an 'mimcr' object is printed.

Value

The 'mimcr' object passed to the object parameter is returned invisibly.

See Also

mimcr, formatC, methods.

Examples

```
# Assessment of data by aid of the mimcr() function
res1 <- mimcr(data = dip1, tcol = 3:10, grouping = "type")

# Summary of the assessment
summary(res1)

# Results of Model-Independent Multivariate Confidence Region (MIMCR)
# approach to assess equivalence of highly variable in-vitro
# dissolution profiles of two drug product formulations
#
# Did the Newton-Raphson search converge? Yes
#
# Parameters (general):
# Significance level:                   0.05
# Degrees of freedom (1):               7
# Degrees of freedom (2):               4
# Mahalanobis distance (MD):            25.72
# (F) scaling factor K:                 0.1714
# (MD) scaling factor k:                3
# Hotelling's T2:                       1984
#
# Parameters specific for Tsong (1996) approach:
# Maximum tolerable average difference: 10
# Similarity limit:                     11.33
# Observed upper limit:                 31.68
#
# Parameters specific for Hoffelder (2016) approach:
# Noncentrality parameter:              385
# Critial F (Hoffelder):                23.16
# Probability p (Hoffelder):            0.7402
#
# Conclusions:
```

```
#       Tsong (1996):  Dissimilar
#   Hoffelder (2016):  Dissimilar

# Taking only the 15 and 90 minutes testing points into account produces a
# warning because profiles should comprise a minimum of three testing points.
## Not run:
  res2 <- mimcr(data = dip1, tcol = c(5, 9), grouping = "type", mtad = 15,
                signif = 0.1)
  summary(res2)

  # Warning:
  #   In mimcr(data = dip1, tcol = c(5, 9), grouping = "type", mtad = 15,  :
  # The profiles should comprise a minimum of 3 time points. The actual profiles
  # comprise 2 points only.

  # Results of Model-Independent Multivariate Confidence Region (MIMCR)
  # approach to assess equivalence of highly variable in-vitro
  # dissolution profiles of two drug product formulations
  #
  # Did the Newton-Raphson search converge? Yes
  #
  # Parameters (general):
  # Significance level:                   0.1
  # Degrees of freedom (1):               2
  # Degrees of freedom (2):               9
  # Mahalanobis distance (MD):            10.44
  # (F) scaling factor K:                 1.35
  # (MD) scaling factor k:                3
  # Hotelling's T2:                       327
  #
  # Parameters specific for Tsong (1996) approach:
  # Maximum tolerable average difference: 15
  # Similarity limit:                     9.631
  # Observed upper limit:                 11.93
  #
  # Parameters specific for Hoffelder (2016) approach:
  # Noncentrality parameter:              278.3
  # Critial F (Hoffelder):                83.57
  # Probability p (Hoffelder):            0.4823
  #
  # Conclusions:
  #       Tsong (1996):  Dissimilar
  #   Hoffelder (2016):  Dissimilar

## End(Not run)

# A successful comparison:
res3 <- mimcr(data = dip3, tcol = 4:6, grouping = "batch")
summary(res3)

# Results of Model-Independent Multivariate Confidence Region (MIMCR)
# approach to assess equivalence of highly variable in-vitro
# dissolution profiles of two drug product formulations
```

```
#
# Did the Newton-Raphson search converge? Yes
#
# Parameters (general):
#   Significance level:                0.05
# Degrees of freedom (1):              3
# Degrees of freedom (2):              20
# Mahalanobis distance (MD):           0.2384
# (F) scaling factor K:                1.818
# (MD) scaling factor k:               6
# Hotelling's T2:                      0.341
#
# Parameters specific for Tsong (1996) approach:
# Maximum tolerable average difference: 10
# Similarity limit:                    2.248
# Observed upper limit:                1.544
#
# Parameters specific for Hoffelder (2016) approach:
# Noncentrality parameter:             30.32
# Critial F (Hoffelder):               4.899
# Probability p (Hoffelder):           2.891e-08
#
# Conclusions:
#       Tsong (1996):  Similar
#   Hoffelder (2016):  Similar
```

`summary.mztia` *Summary of MZTIA estimation*

Description

This is a method for the function `summary()` for objects of class '`mztia`'.

Usage

```
## S3 method for class 'mztia'
summary(object, ...)
```

Arguments

`object`	An object of class '`mztia`' returned by the `mztia()` function.
`...`	Further arguments passed to or from other methods or arguments that can be passed down to the `print.data.frame()` function.

Details

The "limits" subset (see column "frame") of the data frame that is contained in the "Data" element of the '`mztia`' object is printed.

Value

The 'mztia' object passed to the object parameter is returned invisibly.

See Also

mztia, print.data.frame, methods.

Examples

```
# Assessment of data (in wide format) by aid of the mztia() function
res1 <- mztia(data = dip1, shape = "wide", tcol = 3:10, grouping = "type",
              reference = "R", cap = FALSE)

# Summary of the assessment
summary(res1)

# Results of Martinez & Zhao Tolerance Interval (TI) Approach
# (TI limits calculated at each time point of the dissolution profiles of a set
# of reference batches)
#
# Time     Mean      LTL       UTL   S1.LTL    S1.UTL   S2.LTL    S2.UTL
# 1    5 46.77167 27.22641  66.31693 22.22641  71.31693 12.22641  81.31693
# 2   10 60.13333 46.15483  74.11184 41.15483  79.11184 31.15483  89.11184
# 3   15 67.27500 56.90417  77.64583 51.90417  82.64583 41.90417  92.64583
# 4   20 71.98667 65.44354  78.52979 60.44354  83.52979 50.44354  93.52979
# 5   30 78.07000 69.54259  86.59741 64.54259  91.59741 54.54259 101.59741
# 6   60 84.81667 77.20275  92.43058 72.20275  97.43058 62.20275 107.43058
# 7   90 89.09333 76.24588 101.94079 71.24588 106.94079 61.24588 116.94079
# 8  120 91.43833 80.29321 102.58346 75.29321 107.58346 65.29321 117.58346
#
# Abbreviations:
#   TL: Tolerance Interval Limit (TL); LTL: lower TL; UTL: upper TL;
#   S1: level 1 boundary (LTL - 5) or (UTL + 5); S2: level 2 boundary
#   (LTL - 15) or (UTL + 15).

# Assessment of data (in long format) by aid of the mztia() function
res2 <- mztia(data = dip5, shape = "long", tcol = 3, grouping = "type",
              reference = "reference", response = "weight", cap = FALSE,
              QS = c(5, 15) / 100)

# Summary of the assessment
summary(res2)

# Results of Martinez & Zhao Tolerance Interval (TI) Approach
# (TI limits calculated at each time point of the dissolution profiles of a set
# of reference batches)
#
# Time    Mean      LTL      UTL   S1.LTL   S1.UTL   S2.LTL   S2.UTL
# 1    1 12.0093 11.87152 12.14708 11.82152 12.19708 11.72152 12.29708
#
# Abbreviations:
#   TL: Tolerance Interval Limit (TL); LTL: lower TL; UTL: upper TL;
```

```
#   S1: level 1 boundary (LTL - 0.05) or (UTL + 0.05); S2: level 2 boundary
#   (LTL - 0.15) or (UTL + 0.15).
```

Index