Package 'expirest'

February 25, 2025

Type Package

Title Expiry Estimation Procedures

Version 0.1.7

Description The Australian Regulatory Guidelines for Prescription Medicines (ARGPM), guidance on ``Stability testing for prescription medicines", recommends to predict the shelf life of chemically derived medicines from stability data by taking the worst case situation at batch release into account. Consequently, if a change over time is observed, a release limit needs to be specified. Finding a release limit and the associated shelf life is supported, as well as the standard approach that is recommended by guidance Q1E ``Evaluation of stability data" from the International Council for Harmonisation (ICH).

License GPL (>= 2)

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

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

Depends R (>= 4.0)

Imports ggplot2, rlang, lifecycle

Suggests testthat

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 2025-02-25 08:30:02 UTC

Contents

`exp1` *Stability potency data of five batches*

Description

A data set containing the potency stability data (in % of label claim (LC)) of five batches of a drug product obtained over a 24 months period. A total of $n = 53$ independent measurements are available (corresponding to data shown in Tables IV, VI and VIII in LeBlond et al. (2011)). Data in Table IV (batches `b2`, `b5` and `b7`) are compatible with a common intercept / common slope (cics) model, data in Table VI (batches `b3`, `b4` and `b5`) with a different intercept / common slope (dics) model and data in Table VIII (bathes `b4`, `b5` and `b8`) with a different intercept / different slope (dids) model.

Usage

```
data(exp1)
```

Format

A data frame with 53 observations and 3 variables:

Batch Factor with levels `b2`, `b3`, `b4`, `b5`, `b7` and `b8`

Month Numeric representing the time points of testing from the start (0 months) to the end (24 months) of the study.

Potency Numeric of the measured potency values in %LC

Source

See reference: Example data sets shown in Tables IV, VI and VIII.

References

LeBlond, D., Griffith, D. and Aubuchon, K. Linear Regression 102: Stability Shelf Life Estimation Using Analysis of Covariance. *J Valid Technol* (2011) **17**(3): 47-68.

Examples

```
str(exp1)
```

`exp2` *Stability related substance data of three batches*

Description

A data set containing the related substance stability data (in % of label claim (LC)) of three batches of a drug product obtained over a 24 months period. A total of $n = 24$ independent measurements are available (corresponding to data shown in Table XI in LeBlond et al. (2011)).

Usage

```
data(exp2)
```

Format

A data frame with 48 observations and 3 variables:

Batch Factor with levels `b4`, `b5` and `b8`

Month Numeric representing the time points of testing from the start (0 months) to the end (24 months) of the study.

Related Numeric of the measured related substance levels in %LC

Source

See reference: Example data sets shown in Table XI.

References

LeBlond, D., Griffith, D. and Aubuchon, K. Linear Regression 102: Stability Shelf Life Estimation Using Analysis of Covariance. *J Valid Technol* (2011) **17**(3): 47-68.

Examples

```
str(exp2)
```

exp3 *Stability moisture data of three batches*

Description

A data set containing the moisture stability data (% (w/w)) of three batches of a drug product obtained over a 24 months period. A total of $n = 33$ independent measurements are available (corresponding to data shown in Table XIII in LeBlond et al. (2011)).

Usage

```
data(exp3)
```

Format

A data frame with 33 observations and 3 variables:

Batch Factor with levels `b1`, `b2` and `b3`

Month Numeric representing the time points of testing from the start (0 months) to the end (24 months) of the study.

Moisture Numeric of the measured moisture levels %(w/w)

Source

See reference: Example data sets shown in Table XIII.

References

LeBlond, D., Griffith, D. and Aubuchon, K. Linear Regression 102: Stability Shelf Life Estimation Using Analysis of Covariance. *J Valid Technol* (2011) **17**(3): 47-68.

Examples

```
str(exp3)
```

exp4 *Stability data of 4 batches*

Description

A data set containing the concentration data (mg/kg) of four batches obtained over a 24 months period of a drug product. A total of $n = 36$ independent measurements are available (corresponding to the data set *Reliability/Stability.jmp* in *JMP(R) 12 Reliability and Survival Methods* manual).

Usage

```
data(exp4)
```

Format

A data frame with 36 observations and 3 variables:

Batch Factor with levels `1_11`, `2_12`, `3_13` and `4_14`.

Month Numeric representing the time points of testing from the start (0 months) to the end (24 months) of the study.

Conc Numeric of the measured concentrations in mg/kg.

Source

See reference: Example data set (Stability.jmp) used in chapter *Stability Analysis*, p. 174-176.

References

SAS Institute Inc. 2015. *JMP(R) 12 Reliability and Survival Methods*. Cary, NC: SAS Institute Inc.

Examples

```
str(exp4)
```

`expirest_osle` *Ordinary shelf life estimation (osle)*

Description

Based on a linear regression model fitted to a stability data set the function `expirest_osle()` estimates the shelf life, or retest period, following the ICH Q1E guideline. The abbreviation "osle" stands for "ordinary shelf life estimation".

Usage

```
expirest_osle(
  data,
  response_vbl,
  time_vbl,
  batch_vbl,
  sl,
  sl_sf,
  srch_range,
  alpha = 0.05,
  alpha_pool = 0.25,
  xform = c("no", "no"),
  shift = c(0, 0),
```

```
  sf_option = "tight",
  ivl = "confidence",
  ivl_type = "one.sided",
  ivl_side = "lower",
  ...
)
```

Arguments

data	A data frame with the columns specified by response_vbl, time_vbl and batch_vbl.
response_vbl	A character string that specifies the response variable name that must be a column of data.
time_vbl	A character string that specifies the time variable name that must be a column of data.
batch_vbl	A character string that specifies the column in data with the grouping information (i.e. a factorial variable) for the differentiation of the observations of the various batches.
sl	A numeric value or a numeric vector of length 2 that specifies the specification limit or limits. If a vector is provided it must be of the form c(lower limit, upper limit).
sl_sf	A positive integer or a vector of positive integers that specifies the number of "significant figures" (sf) of sl. It must have the same length as sl.
srch_range	A vector of length 2 that specifies the end-points of the (time) range that is supposed to contain the shelf life or retest period.
alpha	A numeric value between 0 and 1 that specifies the significance level for the calculation of confidence or prediction intervals. The default is 0.05.
alpha_pool	A numeric value between 0 and 1 that specifies the type I error rate for the test of the poolability of the batches. The default is 0.25, following ICH Q1E guideline. The value should not be changed unless supported by well-founded reasons.
xform	A vector of two character strings that specifies the transformation of the response and the time variable. The default is "no" transformation, i.e. c("no", "no"), where the first element specifies the transformation of the x variable and the second element the transformation of the y variable. Valid alternatives for x and/or y variable transformation are "log" (natural logarithm), "sqrt" (square root) and "sq" (square).
shift	A vector of two values which will be added to the variables x and/or y before they are transformed as specified by the xform parameter, where the first element will be added to the x variable and the second element to the y variable. The purpose is to prevent an undefined state which could arise when variables with values of ≤ 0 are log or square root transformed. The default is c(0, 0).
sf_option	A character string that specifies if the limits (rl or sl) should be regarded as "tight" or "loose", i.e. either "tight" or "loose", respectively. The default is "tight". The option "tight" means that the limits are rounded to the number

of significant digits specified by the parameters `rl_sf` and `sl_sf`. If `sf_option = "loose"`, four on the order of the last significant digit + 1 is added if `ivl_side = "upper"` or five on the order of the last significant digit + 1 is subtracted if `ivl_side = "upper"`.

`ivl` A character string of either `"confidence"` or `"prediction"` that specifies the type of interval of interest. The default is `"confidence"`.

`ivl_type` A character string that specifies if a "one sided" or a "two sided" interval should be calculated, i.e. either `"one.sided"` or `"two.sided"`, respectively. The default is `"one.sided"`.

`ivl_side` A character string that specifies if the specification limit, given that the limit has only one side, is an "upper" or a "lower" bound, i.e. it is specified as either `"upper"` or `"lower"`, respectively. The default is `"lower"`. If the specification has two boundaries, then this parameter specifies the preferred side. If no side is preferred over the other, `"both"` can be used.

`...` Additional named or unnamed arguments passed on to `uniroot()`.

Details

According to ICH Q1E guideline, "*an appropriate approach to retest period or shelf life estimation is to analyse a quantitative attribute by determining the earliest time at which the 95 percent confidence limit for the mean intersects the proposed acceptance criterion*" (in this package, this point is called the "point of intersection" (POI)). Furthermore, it says that "*for an attribute known to increase with time, the upper one-sided 95 percent confidence limit should be compared to the acceptance criterion. For an attribute that can either increase or decrease, or whose direction of change is not known, two-sided 95 percent confidence limits should be calculated and compared to the upper and lower acceptance criteria.*" The approach can be used to estimate the retest period or shelf life for a single batch or for multiple batches. According to the guideline, "*for a drug substance or for a drug product available in a single strength and a single container size and/or fill, the retest period or shelf life is generally estimated based on the stability data from a minimum of three batches.*"

Before performing the retest period or shelf life estimation with results from multiple batches, the most suitable model should be determined. It should particularly be verified if data of all test batches are poolable or not. Details on this are described in section "Checking batch poolability" below.

Value

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

`Data` Data frame of the original data including new columns with transformed variables, if applicable.

`Parameters` A list of the parameters with the elements `alpha`, `alpha.pool`, `ivl`, `ivl.type` and `ivl.side`.

`Variables` A list of the variable names, i.e. the original names of `batch_vbl`, `time_vbl` and `response_vbl` and, if applicable, of the transformed variables.

`Model.Type` A list of two elements that specifies which model, based on the ANCOVA analysis, suits best. The first element (`type.spec`) is a numeric vector of length 2 that specifies the best model accepted at the significance level specified by

alpha.pool. The first number represents the decision on the intercept and the second on the slope, where 1 stands for "common" and 2 stands for "different". The second element (type.acronym) is an acronym representing the first item.

Models — A list of four elements named cics, dics, dids.pmse and dids. The first three elements contain the 'lm' objects of the "common intercept / common slope" (cics), "different intercept / common slope" (dics) and "different intercept / different slope" (dids) models. The fourth element is a list of the 'lm' objects that is obtained from fitting a regression model to the data of each level of the categorical variable separately. The cics, dics and dids.pmse elements are NA if data of only a single batch is available.

AIC — A numeric named vector of the Akaike Information Criterion (AIC) values of the cics, dics and dids.pmse models.

BIC — A numeric named vector of the Bayesian Information Criterion (BIC) values of each of the cics, dics and dids.pmse models.

wc.icpt — A numeric named vector of the worst case intercepts. The information about which limit the corresponding confidence interval crosses is stored in the attribute named side.

wc.batch — A numeric named vector of the batches with the worst case intercepts. The information about which limit the corresponding confidence interval crosses is stored in the attribute named side.

Limits — A list of all limits.

Intercepts — A list of the intercepts of all models.

All.POI — A list of two elements named lower and upper that contain either NA or lists of the POI values of all models.

POI — A numeric named vector of the POI values of the worst case batches of each model. The information about which limit the corresponding confidence interval crosses is stored in the attribute named side.

Checking batch poolability

According to ICH Q1E guideline, construction of the 95% confidence interval on the basis of the combined data of all test batches is allowed only if it has been confirmed by aid of a statistical test whether the regression lines from the different batches have a common slope and a common intercept. A significance level of alpha_pool = 0.25 should to be used for both batch-related terms, and the test of the slopes has to precede the test of the intercepts. From these tests, three possible models may be appropriate, i.e.

- a *common intercept / common slope* model (cics),
- a *different intercept / common slope* model (dics) or
- a *different intercept / different slope* model (dids).

The *common intercept / different slope* model (cids) is of limited practical relevance because the corresponding model is missing an effect. When slopes exhibit significant differences, comparing intercepts becomes inconsequential. Moreover, while initial levels of different batches in a manufacturing process might be relatively well-controlled, it is improbable that these levels are identical.

Consequently, if the model probabilities associated with the intercepts and slopes suggest the appropriateness of the cids model, the decision is taken in favour of a dids model. The dids model has individual intercepts and individual slopes, and the calculation of confidence intervals is based on the corresponding individual mean square errors. The *different intercept / different slope* model where the mean square error is pooled across batches is reported as dids.pmse.

These requirements can be checked by aid of an "ANalysis of COVAriance" (ANCOVA) including the batch variable as main effect and as $batch \times time$ interaction term. The full ANCOVA model simultaneously tests all the effects, and non-significant effects can be identified and removed for fitting of the final regression model that is used for the estimation of the shelf life or retest period.

The significance level (`alpha_pool = 0.25`, Type I error) is used to increase the power of the test to detect cases where the data should not be pooled. Setting `alpha_pool = 0.25` decreases the probability of incorrectly concluding that stability data from multiple batches can be pooled. On the other hand, though, it increases the probability of using a single batch to determine expiry when pooling batches would be more appropriate.

References

International Council for Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human (ICH), Harmonised Tripartite Guideline, Evaluation of Stability Data Q1E, step 4, February 2003 (CPMP/ICH/420/02).

See Also

`expirest_wisle`, `plot_expirest_wisle`, `uniroot`, `lm`, `AIC`, `BIC`.

Examples

```
# Successful estimations
# A model with common intercepts / common slopes (cics)
res1 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")
res1$Model.Type
res1$POI

# Expected results in res1$Model.Type
# $type.spec
# common.icpt  common.slp
#           1           1
#
# $type.acronym
# [1] "cics"

# Expected results in res1$POI
#     cics      dics dids.pmse      dids
# 26.22410  24.80030  23.66724  23.34184
# attr(,"side")
#     cics      dics dids.pmse      dids
```

```
#  "lower"   "lower"   "lower"   "lower"

# The parameter settings sf_option = "loose" and ivl_side = "lower" (the
# default setting of ivl_side) cause the specification limit of 95.0
# (sl_sf = 3, i.e. 3 significant digits) to be reduced by 0.05, i.e. the
# actual specification limit is 94.95.

# A model with different intercepts / different slopes (dids)
res2 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")
res2$Model.Type
res2$POI

# Expected results in res2$Model.Type
# $type.spec
# common.icpt  common.slp
#           0           0
#
# $type.acronym
# [1] "dids"

# Expected results in res2$POI
#     cics      dics dids.pmse      dids
# 28.12518  22.47939  15.72348  15.96453
# attr(,"side")
#     cics      dics dids.pmse      dids
#  "lower"   "lower"   "lower"   "lower"

# Estimation where it is not know a priori which limit is crossed first, i.e.
# letting the estimation be done for both specification limits.
res3 <-
  expirest_osle(
    data = exp3, response_vbl = "Moisture", time_vbl = "Month",
    batch_vbl = "Batch", sl = c(1.5, 3.5), sl_sf = c(2, 2),
    srch_range = c(0, 500), sf_option = "loose", ivl = "confidence",
    ivl_type = "two.sided", ivl_side = "both")
res3$Model.Type
res3$POI

# Expected results in res3$Model.Type
# $type.spec
# common.icpt  common.slp
#           1           1
#
# $type.acronym
# [1] "cics"

# Expected results in res3$POI
#     cics      dics dids.pmse      dids
# 46.85172  41.84802  22.41808  22.50966
```

```
# attr(,"side")
#     cics      dics dids.pmse      dids
#  "upper"   "upper"   "upper"   "lower"

# The parameter settings sf_option = "loose" and ivl_side = "both" (the
# default setting of ivl_side) cause the specification limits of 1.5 and 3.5
# (sl_sf = 2, i.e. 2 significant digits) to be reduced by 0.05 and increased
# by 0.04, respectively, i.e. the actual specification limits are 1.45 and
# 3.54, respectively.

# Analysis with a single batch
res4 <-
  expirest_osle(
    data = exp3[exp3$Batch == "b1", ], response_vbl = "Moisture",
    time_vbl = "Month", batch_vbl = "Batch", sl = c(1.5, 3.5),
    sl_sf = c(2, 2), srch_range = c(0, 500), sf_option = "tight",
    ivl = "confidence", ivl_type = "two.sided", ivl_side = "both")

# Since only one batch is involved there is no model type. Nevertheless, the
# result is reported under the dids model name.
res4$Model.Type
res4$POI

# Expected results in res4$Model.Type
# $type.spec
# common.icpt  common.slp
#         NA           NA
#
# $type.acronym
# [1] "n.a."

# Expected results in res4$POI
# cics      dics dids.pmse      dids
#   NA        NA        NA  21.42596
# attr(,"side")
# cics      dics dids.pmse      dids
# "NA"      "NA"      "NA"   "lower"

# Unsuccessful estimations
## Not run:
  # Intervals are wider than the specification limits (no intersection).
  res5 <-
    expirest_osle(
      data = exp3, response_vbl = "Moisture", time_vbl = "Month",
      batch_vbl = "Batch", sl = 1.5, sl_sf = 2, srch_range = c(0, 500),
      sf_option = "tight", ivl = "prediction", ivl_type = "two.sided",
      ivl_side = "lower")
  res5$POI

  # (Expected) results in res5$POI
  #    cics      dics dids.pmse      dids
  #     NA        NA        NA        NA
  # attr(,"side")
```

```
  #    cics      dics dids.pmse      dids
  # "lower"   "lower"   "lower"   "lower"

  # Estimation may also fail because of an inappropriate 'srch_range' setting.
  res6 <-
    expirest_osle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                  response_vbl = "Potency", time_vbl = "Month",
                  batch_vbl = "Batch", sl = 95, sl_sf = 3,
                  srch_range = c(0, 5))
  res6$POI

  # (Expected) results in res6$POI
  #    cics      dics dids.pmse      dids
  #     NA        NA        NA        NA
  # attr(,"side")
  #    cics      dics dids.pmse      dids
  # "lower"   "lower"   "lower"   "lower"

## End(Not run)
```

`expirest_wisle` *What-if (approach for) shelf life estimation (wisle)*

Description

Based on a linear regression model fitted to a stability data set the function `expirest_wisle()` estimates the shelf life, or retest period, for the specified release and specification limit following the ARGPM guidance "Stability testing for prescription medicines". The abbreviation "wisle" stands for "what-if shelf life estimation" (because it estimates the shelf life ("what") for a given release limit ("if")).

Usage

```
expirest_wisle(
  data,
  response_vbl,
  time_vbl,
  batch_vbl,
  rl,
  rl_sf,
  sl,
  sl_sf,
  srch_range,
  alpha = 0.05,
  alpha_pool = 0.25,
  xform = c("no", "no"),
  shift = c(0, 0),
  sf_option = "tight",
```

```
  ivl = "confidence",
  ivl_type = "one.sided",
  ivl_side = "lower",
  ...
)
```

Arguments

`data`	A data frame with the columns specified by `response_vbl`, `time_vbl` and `batch_vbl`.
`response_vbl`	A character string that specifies the response variable name that must be a column of `data`.
`time_vbl`	A character string that specifies the time variable name that must be a column of `data`.
`batch_vbl`	A character string that specifies the column in `data` with the grouping information (i.e. a factorial variable) for the differentiation of the observations of the various batches.
`rl`	A numeric value or a numeric vector that specifies the release specification limit(s) for which the corresponding expiry should be estimated.
`rl_sf`	A positive integer or a vector of positive integers that specifies the number of "significant figures" (sf) of `rl`. It must have the same length as `rl`.
`sl`	A numeric value or a numeric vector of length `2` that specifies the specification limit or limits. If a vector is provided it must be of the form `c(lower limit, upper limit)`.
`sl_sf`	A positive integer or a vector of positive integers that specifies the number of "significant figures" (sf) of `sl`. It must have the same length as `sl`.
`srch_range`	A vector of length `2` that specifies the end-points of the (time) range that is supposed to contain the shelf life or retest period.
`alpha`	A numeric value between 0 and 1 that specifies the significance level for the calculation of confidence or prediction intervals. The default is `0.05`.
`alpha_pool`	A numeric value between 0 and 1 that specifies the type I error rate for the test of the poolability of the batches. The default is `0.25`, following ICH Q1E guideline. The value should not be changed unless supported by well-founded reasons.
`xform`	A vector of two character strings that specifies the transformation of the response and the time variable. The default is "no" transformation, i.e. `c("no", "no")`, where the first element specifies the transformation of the x variable and the second element the transformation of the y variable. Valid alternatives for x and/or y variable transformation are `"log"` (natural logarithm), `"sqrt"` (square root) and `"sq"` (square).
`shift`	A vector of two values which will be added to the variables x and/or y before they are transformed as specified by the `xform` parameter, where the first element will be added to the x variable and the second element to the y variable. The purpose is to prevent an undefined state which could arise when variables with values of ≤ 0 are log or square root transformed. The default is `c(0, 0)`.

`sf_option` A character string that specifies if the limits (`rl` or `sl`) should be regarded as "tight" or "loose", i.e. either `"tight"` or `"loose"`, respectively. The default is `"tight"`. The option `"tight"` means that the limits are rounded to the number of significant digits specified by the parameters `rl_sf` and `sl_sf`. If `sf_option = "loose"`, four on the order of the last significant digit + 1 is added if `ivl_side = "upper"` or five on the order of the last significant digit + 1 is subtracted if `ivl_side = "upper"`.

`ivl` A character string of either `"confidence"` or `"prediction"` that specifies the type of interval of interest. The default is `"confidence"`.

`ivl_type` A character string that specifies if a "one sided" or a "two sided" interval should be calculated, i.e. either `"one.sided"` or `"two.sided"`, respectively. The default is `"one.sided"`.

`ivl_side` A character string that specifies if the "upper" or the "lower" limit is the relevant limit, i.e. either `"upper"` or `"lower"`, respectively. The default is `"lower"`. Since this parameter additionally specifies the relationship of `rl` with `sl`, i.e. which of the two sides of `sl` the `rl` is compared to, only either either `"upper"` or `"lower"` is possible. In this respect, the usage of `ivl_side` differs from its usage in the `expirest_osle()` function where `ivl_side` in addition can be `"both"`.

`...` Additional named or unnamed arguments passed on to `uniroot()`.

Details

For the shelf life estimation for submissions in Australia the Australian Regulatory Guidelines for Prescription Medicines (ARGPM), i.e. the guidance "Stability testing for prescription medicines", is binding. In chapter 14.3.1, "Predicting shelf life from stability data", it is described how the estimation should be done. It recommends to take the worst case situation at batch release into account. The following examples are listed:

1a) For medicine that has a lower Assay release limit of 95 per cent and a lower assay expiry limit of 90 per cent, the maximum decrease in assay allowed over the shelf-life is 5 per cent.

2a) Similarly, for a medicine that has a release limit for an individual impurity of 0.2 per cent and an expiry limit of 0.5 per cent, the maximum increase in the impurity allowed over the shelf life is 0.3 per cent.

1b) For the same medicine, if the actual release assay result is 101 per cent, then the shelf life should be determined at the time the medicine (or confidence interval of the regression line) decreases by 5 per cent and reaches 96 per cent, rather than the expiry limit (90 per cent). This takes into account the possibility of batches being released at the lower release limit (i.e. 95 per cent) and ensures they will comply with the expiry limit throughout the shelf life.

2b) Similarly, if the actual impurity content is 0.1 per cent at batch release, the shelf life should be determined as the time the 95 per cent confidence interval reaches 0.4 per cent (i.e. increases by 0.3 per cent).

Consequently, it is necessary to define

- a release limit and
- an expiry limit or shelf life limit (usually the specification limit).

If both of these limits were the same the shelf life would be zero, i.e. no change allowed.

For the estimation of the shelf life or expiry limit it is necessary to find the point where the upper or lower 95% confidence interval limit of the linear model fitted to the data (by aid of `lm`) intersects the "worst case scenario limit" (WCSL), which is defined by the ARGPM guidance "Stability testing for prescription medicines" as the intercept $\pm$ difference between the specification limit and the release limit, where this differences is added $(+)$ if the upper limit is relevant or it is subtracted $(-)$ if the lower limit is relevant. In this package, this point is called the "point of intersection" or "point of interest", abbreviated POI.

Before performing the retest period or shelf life estimation the most suitable model should be determined. It should particularly be verified if data of all test batches are poolable or not. Details on this are described in section "Checking batch poolability" below.

Value

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

`Data`	Data frame of the original data including new columns with transformed variables, if applicable.
`Parameters`	A list of the parameters with the elements `alpha`, `alpha.pool`, `ivl`, `ivl.type` and `ivl.side`.
`Variables`	A list of the variable names, i.e. the original names of `batch_vbl`, `time_vbl` and `response_vbl` and, if applicable, of the transformed variables.
`Model.Type`	A list of two elements that specifies which model, based on the ANCOVA analysis, suits best. The first element (`type.spec`) is a numeric vector of length 2 that specifies the best model accepted at the significance level specified by `alpha.pool`. The first number represents the decision on the intercept and the second on the slope, where `1` stands for "common" and `2` stands for "different". The second element (`type.acronym`) is an acronym representing the first item.
`Models`	A list of four elements named `cics`, `dics`, `dids.pmse` and `dids`. The first three elements contain the '`lm`' objects of the "common intercept / common slope" (`cics`), "different intercept / common slope" (`dics`) and "different intercept / different slope" (`dids`) models. The fourth element is a list of the '`lm`' objects that is obtained from fitting a regression model to the data of each level of the categorical variable separately. The `cics`, `dics` and `dids.pmse` elements are `NA` if data of only a single batch is available.
`AIC`	A numeric named vector of the Akaike Information Criterion (AIC) values of the `cics`, `dics` and `dids.pmse` models.
`BIC`	A numeric named vector of the Bayesian Information Criterion (BIC) values of each of the `cics`, `dics` and `dids.pmse` models.
`wc.icpt`	A data frame of the worst case intercepts of each of the four fitted models.
`wc.batch`	A list of numeric value(s) of the worst case batch(es) per model type.
`Limits`	A list of all limits.
`POI`	A data frame of the intercepts, the differences between release and shelf life limits, the WCSLs, the expiry and release specification limits, the shelf lives and POI values.

Structure of the POI data frame:

Intercept.cics	The intercept of the worst case batch of the cics model.
Intercept.dics	The intercept of the worst case batch of the dics model.
Intercept.dids.pmse	The intercept of the worst case batch of the dids model with pooled mean square error (pmse).
Intercept.dids	The intercept of the worst case batch of the dids model obtained by fitting individual models to the data of each batch.
Delta.cics	Absolute difference between the release and and the shelf life specification of the cics model.
Delta.dics	Absolute difference between the release and and the shelf life specification of the dics model.
Delta.dids.pmse	Absolute difference between the release and and the shelf life specification of the dids model with pooled mean square error (pmse).
Delta.dids	Absolute difference between the release and and the shelf life specification of the dids model obtained by fitting individual models to the data of each batch.
WCSL.cics	WCSL of the cics model.
WCSL.dics	WCSL of the dics model.
WCSL.dids.pmse	WCSL of the dids model with pooled mean square error (pmse).
WCSL.dids	WCSL of the dids model obtained by fitting individual models to the data of each batch.
Exp.Spec	The (expiry) specification, i.e. the specification which is relevant for the determination of the expiry.
Rel.Spec	The calculated release specification.
Shelf.Life.cics	The estimated shelf life of the cics model.
Shelf.Life.dics	The estimated shelf life of the dics model.
Shelf.Life.dids.pmse	The estimated shelf life of the dids model with pooled mean square error (pmse).
Shelf.Life.dids	The estimated shelf life of the dids model obtained by fitting individual models to the data of each batch.
POI.Model.cics	The POI of the cics model.
POI.Model.dics	The POI of the dics model.
POI.Model.dids.pmse	The POI of the dids model with pooled mean square error (pmse).
POI.Model.dids	The POI of the dids model obtained by fitting individual models to the data of each batch.

Checking batch poolability

According to ICH Q1E guideline, construction of the 95% confidence interval on the basis of the combined data of all test batches is allowed only if it has been confirmed by aid of a statistical test whether the regression lines from the different batches have a common slope and a common intercept. A significance level of `alpha_pool = 0.25` should to be used for both batch-related terms, and the test of the slopes has to precede the test of the intercepts. From these tests, three possible models may be appropriate, i.e.

- a *common intercept / common slope* model (cics),
- a *different intercept / common slope* model (dics) or
- a *different intercept / different slope* model (dids).

The *common intercept / different slope* model (cids) is of limited practical relevance because the corresponding model is missing an effect. When slopes exhibit significant differences, comparing intercepts becomes inconsequential. Moreover, while initial levels of different batches in a manufacturing process might be relatively well-controlled, it is improbable that these levels are identical. Consequently, if the model probabilities associated with the intercepts and slopes suggest the appropriateness of the cids model, the decision is taken in favour of a dids model. The dids model has individual intercepts and individual slopes, and the calculation of confidence intervals is based on the corresponding individual mean square errors. The *different intercept / different slope* model where the mean square error is pooled across batches is reported as dids.pmse.

These requirements can be checked by aid of an "ANalysis of COVAriance" (ANCOVA) including the batch variable as main effect and as $batch \times time$ interaction term. The full ANCOVA model simultaneously tests all the effects, and non-significant effects can be identified and removed for fitting of the final regression model that is used for the estimation of the shelf life or retest period.

The significance level (`alpha_pool = 0.25`, Type I error) is used to increase the power of the test to detect cases where the data should not be pooled. Setting `alpha_pool = 0.25` decreases the probability of incorrectly concluding that stability data from multiple batches can be pooled. On the other hand, though, it increases the probability of using a single batch to determine expiry when pooling batches would be more appropriate.

References

Therapeutic Goods Administration (TGA) of the Department of Health of the Australian Government, Australian Regulatory Guidelines for Prescription Medicines (ARGPM), Stability testing for prescription medicines, Version 1.1, March 2017

International Council for Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human (ICH), Harmonised Tripartite Guideline, Evaluation of Stability Data Q1E, step 4, February 2003 (CPMP/ICH/420/02).

See Also

`expirest_osle`, `plot_expirest_wisle`, `uniroot`, `lm`, `AIC`, `BIC`.

Examples

```
# Successful estimations
# A model with common intercepts / common slopes (cics)
```

```
res1 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")
res1$Model.Type
res1$POI

# Expected results in res1$Model.Type
# $type.spec
# common.icpt  common.slp
#           1           1
#
# $type.acronym
# [1] "cics"

# (Expected) results in res1$POI
# Intercept.cics Intercept.dics Intercept.dids Intercept.dids.pmse Delta.cics
#       100.5669       100.3638       100.2491            100.2491          3
# Delta.dics Delta.dids Delta.dids.pmse WCSL.cics WCSL.dics WCSL.dids
#          3          3               3  97.56688  97.36375  97.24914
# WCSL.dids.pmse Exp.Spec.Report Exp.Spec Rel.Spec.Report Rel.Spec
#       97.24914              95    94.95              98    97.95
# Shelf.Life.cics Shelf.Life.dics Shelf.Life.dids Shelf.Life.dids.pmse
#        14.07398        13.23176        13.34561             13.80695
# POI.Model.cics POI.Model.dics POI.Model.dids POI.Model.dids.pmse
#        26.2241        24.8003       23.34184            23.66724

# The parameter settings sf_option = "loose" and ivl_side = "lower" (the
# default setting of ivl_side) cause the specification limit of 95.0
# (sl_sf = 3, i.e. 3 significant digits) to be reduced by 0.05, i.e. the
# actual specification limit is 94.95. Analogously, the release limit of 98.0
# is reduced to 97.95.

# A model with different intercepts / different slopes (dids)
res2 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")
res2$Model.Type
res2$POI

# Expected results in res2$Model.Type
# $type.spec
# common.icpt  common.slp
#           0           0
#
# $type.acronym
# [1] "dids"

# (Expected) results in res2$POI
# Intercept.cics Intercept.dics Intercept.dids Intercept.dids.pmse Delta.cics
```

```
#       101.5498       100.4882       101.2594            101.2594          3
# Delta.dics Delta.dids Delta.dids.pmse WCSL.cics WCSL.dics WCSL.dids
#          3          3               3  98.54976  97.48822  98.25938
# WCSL.dids.pmse Exp.Spec.Report Exp.Spec Rel.Spec.Report Rel.Spec
#       98.25938              95    94.95              98    97.95
# Shelf.Life.cics Shelf.Life.dics Shelf.Life.dids Shelf.Life.dids.pmse
#        13.03332        11.42141        7.619661             7.483223
# POI.Model.cics POI.Model.dics POI.Model.dids POI.Model.dids.pmse
#       28.12518       22.47939       15.96453            15.72348

# Analysis with a single batch (i.e. the worst case batch of res2 above)
res3 <-
  expirest_wisle(data = exp1[exp1$Batch == "b8", ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")
res3$Model.Type
res3$POI

# Since only one batch is involved there is no model type. Nevertheless, the
# result is reported under the dids model name.

# Expected results in res3$Model.Type
# $type.spec
# common.icpt  common.slp
#         NA           NA
#
# $type.acronym
# [1] "n.a."

# (Expected) results in res3$POI
# Intercept.cics Intercept.dics Intercept.dids Intercept.dids.pmse Delta.cics
#             NA             NA       101.2594                  NA         NA
# Delta.dics Delta.dids Delta.dids.pmse WCSL.cics WCSL.dics WCSL.dids
#         NA          3              NA        NA        NA  98.25938
# WCSL.dids.pmse Exp.Spec.Report Exp.Spec Rel.Spec.Report Rel.Spec
#             NA              95    94.95              98    97.95
# Shelf.Life.cics Shelf.Life.dics Shelf.Life.dids Shelf.Life.dids.pmse
#              NA              NA        7.619661                  NA
# POI.Model.cics POI.Model.dics POI.Model.dids POI.Model.dids.pmse
#             NA             NA       15.96453                  NA

# Unsuccessful estimation
## Not run:
  # The interval does not cross the limit (i.e. not within srch_range).
  res4 <-
    expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                   response_vbl = "Potency", time_vbl = "Month",
                   batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 105,
                   sl_sf = 4, srch_range = c(0, 500), sf_option = "loose",
                   ivl_side = "upper")
  res4$POI
```

```
  # (Expected) results in res3$POI
  # Intercept.cics Intercept.dics Intercept.dids Intercept.dids.pmse Delta.cics
  #       100.5669             NA             NA                  NA          3
  # Delta.dics Delta.dids Delta.dids.pmse WCSL.cics WCSL.dics WCSL.dids
  #         NA         NA              NA  107.5669        NA        NA
  # WCSL.dids.pmse Exp.Spec.Report Exp.Spec Rel.Spec.Report Rel.Spec
  #             NA              105   105.04              98    98.04
  # Shelf.Life.cics Shelf.Life.dics Shelf.Life.dids Shelf.Life.dids.pmse
  #              NA              NA              NA                   NA
  # POI.Model.cics POI.Model.dics POI.Model.dids POI.Model.dids.pmse
  #             NA             NA             NA                  NA

  # Estimation may also fail because of an inappropriate 'srch_range' setting.
  res5 <-
    expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                   response_vbl = "Potency", time_vbl = "Month",
                   batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                   sl_sf = 3, srch_range = c(0, 5), sf_option = "loose")
  res5$POI

  # (Expected) results in res4$POI
  # Intercept.cics Intercept.dics Intercept.dids Intercept.dids.pmse Delta.cics
  #       100.5669             NA             NA                  NA          3
  # Delta.dics Delta.dids Delta.dids.pmse WCSL.cics WCSL.dics WCSL.dids
  #         NA         NA              NA  97.56688        NA        NA
  # WCSL.dids.pmse Exp.Spec.Report Exp.Spec Rel.Spec.Report Rel.Spec
  #             NA              95    94.95              98    97.95
  # Shelf.Life.cics Shelf.Life.dics Shelf.Life.dids Shelf.Life.dids.pmse
  #              NA              NA              NA                   NA
  # POI.Model.cics POI.Model.dics POI.Model.dids POI.Model.dids.pmse
  #             NA             NA             NA                  NA

## End(Not run)
```

`plot.plot_expirest_osle` *Plot illustrating the shelf life estimation (osle)*

Description

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

Usage

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

Arguments

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

Details

The element Graph of the 'plot_expirest_osle' object that is returned by the function plot_expirest_osle() 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_expirest_osle' object passed to the x parameter is returned invisibly.

See Also

expirest_osle, plot_expirest_osle, ggplot(), methods.

Examples

```
# Performing an "ordinary shelf life estimation" (osle)
res1 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")

# The 'expirest_osle' object can be passed on to the plot_expirest_osle()
# function. This function does not produce any output but returns a
# 'plot_expirest_osle' object.
## Not run:
  gg1 <- plot_expirest_osle(
    model = res1, response_vbl_unit = "%", x_range = NULL, y_range = c(93, 105),
    mtbs = "verified", plot_option = "full", ci_app = "line")
  gg2 <- plot(gg1)

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

## End(Not run)
```

plot.plot_expirest_wisle *Plot illustrating the what-if shelf life estimation (wisle)*

Description

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

Usage

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

Arguments

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

Details

The element `Graph` of the '`plot_expirest_wisle`' object that is returned by the function `plot_expirest_wisle()` 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_expirest_wisle`' object passed to the `x` parameter is returned invisibly.

See Also

`expirest_wisle`, `plot_expirest_wisle`, `ggplot()`, `methods`.

Examples

```
# Performing a "what-if (approach for) shelf life estimation" (wisle)
res1 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")

# The 'expirest_wisle' object can be passed on to the plot_expirest_wisle()
# function. This function does not produce any output but returns a
# 'plot_expirest_wisle' object.
## Not run:
```

```
  gg1 <- plot_expirest_wisle(
    model = res1, rl_index = 1, response_vbl_unit = "%", x_range = NULL,
    y_range = c(93, 105), scenario = "standard", mtbs = "verified",
    plot_option = "full", ci_app = "line")
  gg2 <- plot(gg1)

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

## End(Not run)
```

`plot_expirest_osle` *Illustrating the ordinary shelf life estimate (osle)*

Description

The function `plot_expirest_osle()` makes a graphical display of the shelf life estimate done by the `expirest_osle()` function.

Usage

```
plot_expirest_osle(
  model,
  show_grouping = "yes",
  response_vbl_unit = NULL,
  x_range = NULL,
  y_range = NULL,
  mtbs = "verified",
  plot_option = "full",
  ci_app = "line"
)
```

Arguments

`model`	An 'expirest_osle' object, i.e. a list returned by the `expirest_osle()` function.
`show_grouping`	'r lifecycle::badge("deprecated")' 'show_grouping = \"yes\" or \"no\"' is no longer supported. Use the `mtbs` parameter instead which allows choosing a specific model, i.e. also the *common intercept / common slope case* model which was the default model when `show_grouping` was `"no"`.
`response_vbl_unit`	A character string that specifies the unit associated with the response variable. The default is `NULL`.
`x_range`	A numeric vector of the form `c(min, max)` that specifies the range of the time variable to be plotted. The default is `NULL` and the x range is calculated automatically on the basis of the estimated shelf life.

y_range	A numeric vector of the form c(min, max) that specifies the range of the response variable to be plotted. The default is NULL and the y range is calculated automatically on the basis of the time course of the response.
mtbs	A character string that specifies the "model to be shown", i.e. either verified, which is the default, or one of cics, dics, dids or dids.pmse. The verified model is the model that was identified through the poolability check. It is thus also one of the possible optional models. The dids model represents the case where a separate model is fitted to the data of each individual batch while the dids.pmse model is the interaction model which includes the $batch$ variable as main effect and in the interaction term with the $time$ variable ($batch \times time$), i.e. a model where the mean square error is pooled across batches.
plot_option	A character string of either "full" or "lean" that specifies if additional information should be put out on the plot (option "full") or only basic information (option "lean"), i.e. the data points, the fitted regression line with the confidence interval, the specification limit(s) and the estimated shelf life limit(s). The default is "full".
ci_app	A character string of either "line" or "ribbon", that specifies the appearance of the confidence interval, i.e. if the limits should be plotted as lines (option "line") or as a shaded ribbon (option "ribbon"). The default is "line".

Details

The function plot_expirest_osle() uses the data and the information about the linear model that was used for the estimation of the shelf life by aid of the expirest_osle() function. It plots a graph of the time course of a parameter, a linear regression line fitted to the data and the associated confidence or prediction interval. In addition, it shows features of the shelf life estimation.

For plotting, the ggplot() function from the 'ggplot2' package is used. The various arguments can be used to control the appearance of the plot. The 'ggplot2' object of the generated plot is contained in the Graph element of the list that is returned by plot_expirest_osle() and can be used to modify the appearance of the graph.

Value

An object of class 'plot_expirest_osle' is returned invisibly consisting of the following elements:

Model	The 'expirest_osle' object that was passed via the model argument.
Expiry	A data frame of type expiry.
Graph	A 'ggplot2' object for the graphical display.
Prediction	A data frame of the predicted values.
text	A data frame of the text elements on the plot.
hlines	A data frame of the horizontal line elements on the plot.
vlines	A data frame of the vertical line elements on the plot.

See Also

expirest_osle, expirest_wisle, plot_expirest_wisle.

Examples

```
# Start by making an "ordinary shelf life estimation" (osle).
res1 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")

# Pass the 'expirest_osle' object on to the plot_expirest_osle() function.
# This function does not produce any output. It returns a 'plot_expirest_osle'
# object that is essentially an 'expirest_osle' object augmented by a 'ggplot'
# object.
gg1 <- plot_expirest_osle(
  model = res1, response_vbl_unit = "%", x_range = NULL, y_range = c(93, 105),
  mtbs = "verified", plot_option = "full", ci_app = "line")
## Not run:
  gg1

  # Since the element gg1$Graph is a 'ggplot' object it can be used for further
  # manipulation by aid of 'ggplot2' functions.
  if (requireNamespace("ggplot2")) {
    library(ggplot2)

    gg1$Graph + labs(title = "Ordinary Shelf Life Estimation (OSLE)",
                     x = "Time [months]", y = "Potency [% LC]") +
      scale_x_continuous(limits = c(-1, 31), breaks = seq(0, 30, 6))
  }

## End(Not run)

# Repeat this for a different intercept / different slope (dids) model.
res2 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")

gg2 <- plot_expirest_osle(
  model = res2, response_vbl_unit = "%", x_range = c(0, 43),
  y_range = c(83, 107), mtbs = "verified", plot_option = "full",
  ci_app = "ribbon")
## Not run:
  gg2

## End(Not run)

# In case of different intercept / different slope models, individually fit
# linear models are shown by default, i.e. with the 'mtbs' parameter set
# as "verified". To get the different intercept / different slope model
# displayed where the mean square error is pooled across batches, i.e. the
# dids.pmse model, the 'mtbs' parameter has to be set accordingly.
```

```
gg3 <- plot_expirest_osle(
  model = res2, response_vbl_unit = "%", x_range = c(0, 43),
  y_range = c(83, 107), mtbs = "dids.pmse", plot_option = "full",
  ci_app = "ribbon")
## Not run:
  gg3

## End(Not run)
```

`plot_expirest_wisle` *Illustrating the what-if (approach for) shelf life estimate (wisle)*

Description

The function `plot_expirest_wisle()` makes a graphical display of the shelf life estimate done by the `expirest_wisle()` function.

Usage

```
plot_expirest_wisle(
  model,
  rl_index = 1,
  show_grouping = "yes",
  response_vbl_unit = NULL,
  x_range = NULL,
  y_range = NULL,
  scenario = "standard",
  mtbs = "verified",
  plot_option = "full",
  ci_app = "line"
)
```

Arguments

`model`	An 'expirest_wisle' object, i.e. a list returned by the `expirest_wisle()` function.
`rl_index`	A positive integer that specifies which of the release limit values that have been handed over to `expirest_wisle()` should be displayed. The default value is `1`.
`show_grouping`	'r lifecycle::badge("deprecated")' 'show_grouping = \"yes\" or \"no\"' is no longer supported. Use the `mtbs` parameter instead which allows choosing a specific model, i.e. also the *common intercept / common slope case* model which was the default model when `show_grouping` was `"no"`.
`response_vbl_unit`	A character string that specifies the unit associated with the response variable. The default is `NULL`.

x_range	A numeric vector of the form c(min, max) that specifies the range of the time variable to be plotted. The default is NULL and the x range is calculated automatically on the basis of the estimated shelf life.
y_range	A numeric vector of the form c(min, max) that specifies the range of the response variable to be plotted. The default is NULL and the y range is calculated automatically on the basis of the time course of the response.
scenario	A character string that specifies if the plot should be extended (with respect to the x axis) up to the "standard scenario" ("standard") or up to the "worst case scenario" ("worst"). The default is "standard".
mtbs	A character string that specifies the "model to be shown", i.e. either verified, which is the default, or one of cics, dics, dids or dids.pmse. The verified model is the model that was identified through the poolability check. It is thus also one of the possible optional models. The dids model represents the case where a separate model is fitted to the data of each individual batch while the dids.pmse model is the interaction model which includes the $batch$ variable as main effect and in the interaction term with the $time$ variable ($batch \times time$), i.e. a model where the mean square error is pooled across batches.
plot_option	A character string of either "full", "lean1", "lean2", "basic1" and "basic2" that specifies if additional information should be shown in the plot (option "full") or only basic information (options "lean" and "basic"). Full means the data points, the fitted regression line with the confidence or prediction interval, the specification limit(s) and the estimated shelf life. The default is "full".
ci_app	A character string of either "line" or "ribbon", that specifies the appearance of the confidence interval, i.e. if the limits should be plotted as lines (option "line") or as a shaded ribbon (option "ribbon"). The default is "line".

Details

The function plot_expirest_wisle() uses the data and the information about the linear model that was used for the estimation of the shelf life by aid of the expirest_wisle() function. It plots a graph of the time course of a parameter, a linear regression line fitted to the data and the associated confidence or prediction interval. In addition, it shows features of the worst case scenario shelf life estimation.

For plotting, the ggplot() function from the 'ggplot2' package is used. The various arguments can be used to control the appearance of the plot. The 'ggplot2' object of the generated plot is contained in the Graph element of the list that is returned by plot_expirest_wisle() and can be used to modify the appearance of the graph.

Value

An object of class 'plot_expirest_wisle' is returned invisibly consisting of the following elements:

Model	The 'expirest_wisle' object that was passed via the model argument.
Expiery	A data frame of type expiry.
Graph	A 'ggplot2' object for the graphical display.
Prediction	A data frame of the predicted values.

text	A data frame of the text elements on the plot.
hlines	A data frame of the horizontal line elements on the plot.
vlines	A data frame of the vertical line elements on the plot.
segments	A data frame of segment line elements on the plot.
arrow	A data frame of arrow elements on the plot.

See Also

expirest_wisle, expirest_osle, plot_expirest_osle.

Examples

```
# Start by making a "what-if (approach for) shelf life estimation" (wisle)
res1 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")

# Pass the 'expirest_wisle' object on to the plot_expirest_wisle() function.
# This function does not produce any output. It returns a 'plot_expirest_wisle'
# object that is essentially an 'expirest_wisle' object augmented by a 'ggplot'
# object.
gg1 <- plot_expirest_wisle(
  model = res1, rl_index = 1, response_vbl_unit = "%", x_range = NULL,
  y_range = c(93, 105), scenario = "standard", mtbs = "verified",
  plot_option = "full", ci_app = "line")

## Not run:
  gg1

  # Since the element gg1$Graph is a 'ggplot' object it can be used for further
  # manipulation by aid of 'ggplot2' functions.
  if (requireNamespace("ggplot2")) {
    library(ggplot2)

    gg1$Graph + labs(title = "What-if Shelf Life Estimation (WISLE)",
                     x = "Time [months]", y = "Potency [% LC]") +
      scale_x_continuous(limits = c(-5, 31), breaks = seq(0, 30, 6))
  }

## End(Not run)

# Repeat this for a different intercept / different slope (dids) model.
res2 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")

gg2 <- plot_expirest_wisle(
```

```
  model = res2, rl_index = 1, response_vbl_unit = "%", x_range = c(0, 26),
  y_range = c(89, 107),  scenario = "standard",  mtbs = "verified",
  plot_option = "full", ci_app = "ribbon")
## Not run:
  gg2

## End(Not run)

# In case of different intercept / different slope models, individually fit
# linear models are shown by default, i.e. with the 'mtbs' parameter set
# as "verified". To get the different intercept / different slope model
# displayed where the mean square error is pooled across batches, i.e. the
# dids.pmse model, the 'mtbs' parameter has to be set accordingly.

gg3 <- plot_expirest_wisle(
  model = res2, rl_index = 1, response_vbl_unit = "%", x_range = c(0, 26),
  y_range = c(89, 107),  scenario = "standard",  mtbs = "dids.pmse",
  plot_option = "full", ci_app = "ribbon")
## Not run:
  gg3

## End(Not run)
```

`print.expirest_osle` *Print a summary of the shelf life estimation (osle)*

Description

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

Usage

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

Arguments

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

Details

The function `expirest_osle()` estimates the shelf life, or retest period, following the ICH Q1E guideline. By default, batch poolability is checked as recommended by the guideline at a significance level of 0.25. Other levels can be used, although not recommended, by changing the default of the `alpha_pool` parameter. Three possible models may be appropriate, i.e.

- a *common intercept / common slope* model (cics),
- a *different intercept / common slope* model (dics) or
- a *different intercept / different slope* model (dids).

The worst case intercept is the intercept of the batch whose confidence limit is the first crossing the acceptance limit. As in case of the `cics` model type all batches have a common intercept and a common confidence interval, all batches can be regarded as equally worst case. In case of the `dids` model type, shelf life estimation is done using the models obtained from fitting the data of each batch individually.

Value

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

See Also

`expirest_osle`, `expirest_wisle`, `formatC`, `methods`.

Examples

```
# Fit models of different type
res1 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")
res2 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b3", "b4", "b5"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")
res3 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")

# The parameter settings sf_option = "loose" and ivl_side = "lower" (the
# default setting of ivl_side) cause the specification limit of 95.0
# (sl_sf = 3, i.e. 3 significant digits) to be reduced by 0.05, i.e. the
# actual specification limit is 94.95.

## Not run:
  res1
  # Expected output of print(res1)
  # Summary of shelf life estimation following the ICH Q1E guideline
  #
  # The best model accepted at a significance level of 0.25 has
  # Common intercepts and Common slopes (acronym: cics).
  #
  # Worst case intercept: 100.5669 (Potency)
  # Worst case batch: NA
```

```
# Estimated shelf life for cics model:  26.2241 (Month)
#
# Worst case intercepts, POIs and batches of all models
# (Including information about the side where the confidence
#   interval crosses the specification boundary):
#           Intercept      POI  Side Batch
# cics       100.5669  26.2241 lower    NA
# dics       100.3638  24.8003 lower    b2
# dids.pmse  100.7819 23.66724 lower    b5
# dids       100.7819 23.34184 lower    b5

res2
# Expected output of print(res2)
# Summary of shelf life estimation following the ICH Q1E guideline
#
# The best model accepted at a significance level of 0.25 has
# Different intercepts and Common slopes (acronym: dics).
#
# Worst case intercept:   100.82 (Potency)
# Worst case batch: b5
# Estimated shelf life for dics model: 23.60194 (Month)
#
# Worst case intercepts, POIs and batches of all models
# (Including information about the side where the confidence
#   interval crosses the specification boundary):
#           Intercept      POI  Side Batch
# cics       102.0513 29.18093 lower    NA
# dics         100.82 23.60194 lower    b5
# dids.pmse  100.7819 22.49726 lower    b5
# dids       102.3841 23.26251 lower    b3

res3
# Expected output of print(res3)
# Summary of shelf life estimation following the ICH Q1E guideline
#
# The best model accepted at a significance level of 0.25 has
# Different intercepts and Different slopes (acronym: dids).
#
# Worst case intercept: 101.2594 (Potency)
# Worst case batch: b8
# Estimated shelf life for dids model: 15.96453 (Month)
#
# Worst case intercepts, POIs and batches of all models
# (Including information about the side where the confidence
#   interval crosses the specification boundary):
#           Intercept      POI  Side Batch
# cics       101.5498 28.12518 lower    NA
# dics       100.4882 22.47939 lower    b8
# dids.pmse  101.2594 15.72348 lower    b8
# dids       101.2594 15.96453 lower    b8

## End(Not run)
```

print.expirest_wisle *Print a summary of the what-if shelf life estimation (wisle)*

Description

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

Usage

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

Arguments

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

Details

The function `expirest_wisle()` estimates the expiry for the specified release and specification limit following the ARGPM guidance "Stability testing for prescription medicines". By default, batch poolability is checked as recommended by the ICH Q1E guideline at a significance level of 0.25. Other levels can be used, although not recommended, by changing the default of the `alpha_pool` parameter. Three possible models may be appropriate, i.e.

- a *common intercept / common slope* model (cics),
- a *different intercept / common slope* model (dics) or
- a *different intercept / different slope* model (dids).

The worst case intercept is the intercept of the batch whose confidence limit is the first crossing the acceptance limit. As in case of the `cics` model type all batches have a common intercept and a common confidence interval, all batches can be regarded as equally worst case. In case of the `dids` model type, shelf life estimation is done using the models obtained from fitting the data of each batch individually. In addition to the shelf life estimated according to the ARGPM also the estimate according to ICH Q1E is shown.

Value

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

See Also

`expirest_wisle`, `expirest_osle`, `formatC`, `methods`.

Examples

```
# Fit models of different type
res1 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")

res2 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b3", "b4", "b5"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")
res3 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")

# The parameter settings sf_option = "loose" and ivl_side = "lower" (the
# default setting of ivl_side) cause the specification limit of 95.0
# (sl_sf = 3, i.e. 3 significant digits) to be reduced by 0.05, i.e. the
# actual specification limit is 94.95.

## Not run:
  res1
  # Expected output of print(res1)
  # Summary of shelf life estimation following the ARGPM
  # guidance "Stability testing for prescription medicines"
  #
  # The best model accepted at a significance level of 0.25 has
  # Common intercepts and Common slopes (acronym: cics).
  #
  # Worst case intercept and batch:
  #   RL Batch Intercept
  # 1 98    NA  100.5669
  #
  # Estimated shelf lives for the cics model:
  #   SL RL    wisle    osle
  # 1 95 98 14.07398 26.2241
  #
  # Abbreviations:
  #   ARGPM: Australian Regulatory Guidelines for Prescription Medicines;
  #   ICH: International Council for Harmonisation;
  #   osle: Ordinary shelf life estimation (i.e. following the ICH guidance);
  #   RL: Release Limit;
  #   SL: Specification Limit;
  #   wisle: What-if (approach for) shelf life estimation (see ARGPM guidance).

  res2
  # Expected output of print(res2)
  # Summary of shelf life estimation following the ARGPM
```

```
# guidance "Stability testing for prescription medicines"
#
# The best model accepted at a significance level of 0.25 has
# Different intercepts and Common slopes (acronym: dics).
#
# Worst case intercept and batch:
#   RL Batch Intercept
# 1 98    b5    100.82
#
# Estimated shelf lives for the dics model:
#   SL RL    wisle     osle
# 1 95 98 11.40993 23.60194
#
# Abbreviations:
#   ARGPM: Australian Regulatory Guidelines for Prescription Medicines;
#   ICH: International Council for Harmonisation;
#   osle: Ordinary shelf life estimation (i.e. following the ICH guidance);
#   RL: Release Limit;
#   SL: Specification Limit;
#   wisle: What-if (approach for) shelf life estimation (see ARGPM guidance).

res3
# Expected output of print(res3)
# Summary of shelf life estimation following the ARGPM
#   guidance "Stability testing for prescription medicines"
#
# The best model accepted at a significance level of 0.25 has
#  Different intercepts and Different slopes (acronym: dids).
#
# Worst case intercept and batch:
#   RL Batch Intercept
# 1 98    b8  101.2594
#
# Estimated shelf lives for the dids model (for information, the results of
#   the model fitted with pooled mean square error (pmse) are also shown:
#   SL RL    wisle wisle (pmse)     osle osle (pmse)
# 1 95 98 7.619661     7.483223 15.96453    15.72348
#
# Abbreviations:
#   ARGPM: Australian Regulatory Guidelines for Prescription Medicines;
#   ICH: International Council for Harmonisation;
#   osle: Ordinary shelf life estimation (i.e. following the ICH guidance);
#   pmse: Pooled mean square error;
#   RL: Release Limit;
#   SL: Specification Limit;
#   wisle: What-if (approach for) shelf life estimation (see ARGPM guidance).

## End(Not run)
```

print.plot_expirest_osle *Print a plot illustrating the shelf life estimation (osle)*

Description

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

Usage

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

Arguments

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

Details

The element Graph of the 'plot_expirest_osle' object that is returned by the function plot_expirest_osle() 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_expirest_osle' object passed to the x parameter is returned invisibly.

See Also

expirest_osle, plot_expirest_osle, ggplot(), methods.

Examples

```
# Performing an "ordinary shelf life estimation" (osle)
res1 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")

# The 'expirest_osle' object can be passed on to the plot_expirest_osle()
# function. This function does not produce any output but returns a
# 'plot_expirest_osle' object.
## Not run:
  gg1 <- plot_expirest_osle(
    model = res1, response_vbl_unit = "%", x_range = NULL, y_range = c(93, 105),
```

```
  mtbs = "verified", plot_option = "full", ci_app = "line")
gg2 <- print(gg1)

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

## End(Not run)
```

`print.plot_expirest_wisle` *Print a plot illustrating the what-if shelf life estimation (wisle)*

Description

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

Usage

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

Arguments

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

Details

The element `Graph` of the '`plot_expirest_wisle`' object that is returned by the function `plot_expirest_wisle()` 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_expirest_wisle`' object passed to the `x` parameter is returned invisibly.

See Also

`expirest_wisle`, `plot_expirest_wisle`, `ggplot()`, `methods`.

Examples

```
# Performing a "what-if (approach for) shelf life estimation" (wisle)
res1 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")

# The 'expirest_wisle' object can be passed on to the plot_expirest_wisle()
# function. This function does not produce any output but returns a
# 'plot_expirest_wisle' object.
## Not run:
  gg1 <- plot_expirest_wisle(
    model = res1, rl_index = 1, response_vbl_unit = "%", x_range = NULL,
    y_range = c(93, 105), scenario = "standard", mtbs = "verified",
    plot_option = "full", ci_app = "line")
  gg2 <- print(gg1)

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

## End(Not run)
```

---

summary.expirest_osle *Summary of the shelf life estimation (osle)*

Description

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

Usage

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

Arguments

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

Details

The function `expirest_osle()` estimates the shelf life, or retest period, following the ICH Q1E guideline. By default, batch poolability is checked as recommended by the guideline at a significance level of 0.25. Other levels can be used, although not recommended, by changing the default of the `alpha_pool` parameter. Three possible models may be appropriate, i.e.

- a *common intercept / common slope* model (cics),
- a *different intercept / common slope* model (dics) or
- a *different intercept / different slope* model (dids).

The worst case intercept is the intercept of the batch whose confidence limit is the first crossing the acceptance limit. As in case of the `cics` model type all batches have a common intercept and a common confidence interval, all batches can be regarded as equally worst case. In case of the `dids` model type, shelf life estimation is done using the models obtained from fitting the data of each batch individually.

Value

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

See Also

`expirest_osle`, `expirest_wisle`, `formatC`, `methods`.

Examples

```
# Fit models of different type
res1 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")
res2 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b3", "b4", "b5"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")
res3 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3,
                srch_range = c(0, 500), sf_option = "loose")

# The parameter settings sf_option = "loose" and ivl_side = "lower" (the
# default setting of ivl_side) cause the specification limit of 95.0
# (sl_sf = 3, i.e. 3 significant digits) to be reduced by 0.05, i.e. the
# actual specification limit is 94.95.

## Not run:
  summary(res1)
  # Expected output of summary(res1)
  # Summary of shelf life estimation following the ICH Q1E guideline
  #
  # The best model accepted at a significance level of 0.25 has
  # Common intercepts and Common slopes (acronym: cics).
  #
  # Worst case intercept: 100.5669 (Potency)
  # Worst case batch: NA
```

```
# Estimated shelf life for cics model:  26.2241 (Month)
#
# Worst case intercepts, POIs and batches of all models
# (Including information about the side where the confidence
#   interval crosses the specification boundary):
#           Intercept      POI  Side Batch
# cics       100.5669  26.2241 lower    NA
# dics       100.3638  24.8003 lower    b2
# dids.pmse  100.7819 23.66724 lower    b5
# dids       100.7819 23.34184 lower    b5

summary(res2)
# Expected output of summary(res2)
# Summary of shelf life estimation following the ICH Q1E guideline
#
# The best model accepted at a significance level of 0.25 has
# Different intercepts and Common slopes (acronym: dics).
#
# Worst case intercept:   100.82 (Potency)
# Worst case batch: b5
# Estimated shelf life for dics model: 23.60194 (Month)
#
# Worst case intercepts, POIs and batches of all models
# (Including information about the side where the confidence
#   interval crosses the specification boundary):
#           Intercept      POI  Side Batch
# cics       102.0513 29.18093 lower    NA
# dics         100.82 23.60194 lower    b5
# dids.pmse  100.7819 22.49726 lower    b5
# dids       102.3841 23.26251 lower    b3

summary(res3)
# Expected output of summary(res3)
# Summary of shelf life estimation following the ICH Q1E guideline
#
# The best model accepted at a significance level of 0.25 has
# Different intercepts and Different slopes (acronym: dids).
#
# Worst case intercept: 101.2594 (Potency)
# Worst case batch: b8
# Estimated shelf life for dids model: 15.96453 (Month)
#
# Worst case intercepts, POIs and batches of all models
# (Including information about the side where the confidence
#   interval crosses the specification boundary):
#           Intercept      POI  Side Batch
# cics       101.5498 28.12518 lower    NA
# dics       100.4882 22.47939 lower    b8
# dids.pmse  101.2594 15.72348 lower    b8
# dids       101.2594 15.96453 lower    b8

## End(Not run)
```

summary.expirest_wisle

Summary of the what-if shelf life estimation (wisle)

Description

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

Usage

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

Arguments

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

Details

The function `expirest_wisle()` estimates the expiry for the specified release and specification limit following the ARGPM guidance "Stability testing for prescription medicines". By default, batch poolability is checked as recommended by the ICH Q1E guideline at a significance level of 0.25. Other levels can be used, although not recommended, by changing the default of the `alpha_pool` parameter. Three possible models may be appropriate, i.e.

- a *common intercept / common slope* model (cics),
- a *different intercept / common slope* model (dics) or
- a *different intercept / different slope* model (dids).

The worst case intercept is the intercept of the batch whose confidence limit is the first crossing the acceptance limit. As in case of the `cics` model type all batches have a common intercept and a common confidence interval, all batches can be regarded as equally worst case. In case of the `dids` model type, shelf life estimation is done using the models obtained from fitting the data of each batch individually. In addition to the shelf life estimated according to the ARGPM also the estimate according to ICH Q1E is shown.

Value

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

See Also

`expirest_wisle`, `expirest_osle`, `formatC`, `methods`.

Examples

```
# Fit models of different type
res1 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")

res2 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b3", "b4", "b5"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")
res3 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
                 response_vbl = "Potency", time_vbl = "Month",
                 batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
                 sl_sf = 3, srch_range = c(0, 500), sf_option = "loose")

# The parameter settings sf_option = "loose" and ivl_side = "lower" (the
# default setting of ivl_side) cause the specification limit of 95.0
# (sl_sf = 3, i.e. 3 significant digits) to be reduced by 0.05, i.e. the
# actual specification limit is 94.95.

## Not run:
  summary(res1)
  # Expected output of summary(res1)
  # Summary of shelf life estimation following the ARGPM
  # guidance "Stability testing for prescription medicines"
  #
  # The best model accepted at a significance level of 0.25 has
  # Common intercepts and Common slopes (acronym: cics).
  #
  # Worst case intercept and batch:
  #   RL Batch Intercept
  # 1 98    NA  100.5669
  #
  # Estimated shelf lives for the cics model:
  #   SL RL    wisle    osle
  # 1 95 98 14.07398 26.2241
  #
  # Abbreviations:
  #   ARGPM: Australian Regulatory Guidelines for Prescription Medicines;
  #   ICH: International Council for Harmonisation;
  #   osle: Ordinary shelf life estimation (i.e. following the ICH guidance);
  #   RL: Release Limit;
  #   SL: Specification Limit;
  #   wisle: What-if (approach for) shelf life estimation (see ARGPM guidance).

  summary(res2)
  # Expected output of summary(res2)
  # Summary of shelf life estimation following the ARGPM
```

```
# guidance "Stability testing for prescription medicines"
#
# The best model accepted at a significance level of 0.25 has
# Different intercepts and Common slopes (acronym: dics).
#
# Worst case intercept and batch:
#   RL Batch Intercept
# 1 98    b5    100.82
#
# Estimated shelf lives for the dics model:
#   SL RL    wisle     osle
# 1 95 98 11.40993 23.60194
#
# Abbreviations:
#   ARGPM: Australian Regulatory Guidelines for Prescription Medicines;
#   ICH: International Council for Harmonisation;
#   osle: Ordinary shelf life estimation (i.e. following the ICH guidance);
#   RL: Release Limit;
#   SL: Specification Limit;
#   wisle: What-if (approach for) shelf life estimation (see ARGPM guidance).

summary(res3)
# Expected output of summary(res3)
# Summary of shelf life estimation following the ARGPM
#   guidance "Stability testing for prescription medicines"
#
# The best model accepted at a significance level of 0.25 has
#  Different intercepts and Different slopes (acronym: dids).
#
# Worst case intercept and batch:
#   RL Batch Intercept
# 1 98    b8  101.2594
#
# Estimated shelf lives for the dids model (for information, the results of
#   the model fitted with pooled mean square error (pmse) are also shown:
#   SL RL    wisle wisle (pmse)     osle osle (pmse)
# 1 95 98 7.619661     7.483223 15.96453    15.72348
#
# Abbreviations:
#   ARGPM: Australian Regulatory Guidelines for Prescription Medicines;
#   ICH: International Council for Harmonisation;
#   osle: Ordinary shelf life estimation (i.e. following the ICH guidance);
#   pmse: Pooled mean square error;
#   RL: Release Limit;
#   SL: Specification Limit;
#   wisle: What-if (approach for) shelf life estimation (see ARGPM guidance).

## End(Not run)
```

Index