

# Package ‘expirest’

June 1, 2022

**Type** Package

**Title** Expiry Estimation Procedures

**Version** 0.1.3

**Description** The Australian Regulatory Guidelines for Prescription Medicines (ARGPM), guidance on “Stability testing for prescription medicines”  
<<https://www.tga.gov.au/stability-testing-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)  
<<https://database.ich.org/sites/default/files/Q1E%20Guideline.pdf>>.

**License** GPL (>= 2)

**URL** <https://github.com/piusdahinden/expirest>

**BugReports** <https://github.com/piusdahinden/expirest/issues>

**Depends** R (>= 3.5)

**Imports** ggplot2

**Suggests** testthat

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.0

**NeedsCompilation** no

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

**Maintainer** Pius Dahinden <[pius.dahinden@tillotts.com](mailto:pius.dahinden@tillotts.com)>

**Repository** CRAN

**Date/Publication** 2022-06-01 07:40:04 UTC

## R topics documented:

exp1 . . . . .	2
exp2 . . . . .	3
exp3 . . . . .	4
exp4 . . . . .	5
expirest_osle . . . . .	5
expirest_wisle . . . . .	11
plot.plot_expirest_osle . . . . .	18
plot.plot_expirest_wisle . . . . .	19
plot_expirest_osle . . . . .	20
plot_expirest_wisle . . . . .	23
print.expirest_osle . . . . .	26
print.expirest_wisle . . . . .	28
print.plot_expirest_osle . . . . .	30
print.plot_expirest_wisle . . . . .	32
summary.expirest_osle . . . . .	33
summary.expirest_wisle . . . . .	35
<b>Index</b>	<b>39</b>

---

exp1	<i>Stability potency data of five batches</i>
------	---

---

### 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 (batches 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.  
<https://www.ivtnetwork.com/sites/default/files/LinearRegression.pdf>

**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.  
<https://www.ivtnetwork.com/sites/default/files/LinearRegression.pdf>

**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.

<https://www.ivtnetwork.com/sites/default/files/LinearRegression.pdf>

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

```

**Arguments**

<code>data</code>	A data frame with the columns specified by <code>response_vbl</code> , <code>time_vbl</code> and <code>batch_vbl</code> .
<code>response_vbl</code>	A character string specifying the response variable name that must be a column of data.
<code>time_vbl</code>	A character string specifying the time variable name that must be a column of data.
<code>batch_vbl</code>	A character string specifying the column in data with the grouping information (i.e. a factorial variable) for the differentiation of the observations of the different batches.
<code>sl</code>	A numeric value or a numeric vector of length 2 specifying the specification limit or limits. If a vector is provided it must be of the form <code>c(lower limit, upper limit)</code> .
<code>sl_sf</code>	A positive integer or a vector of positive integers specifying the number of “significant figures” (sf) of <code>sl</code> . It must have the same length as <code>sl</code> .
<code>srch_range</code>	A vector of length 2 specifying the end-points of the (time) range that is supposed to contain the shelf life or retest period.
<code>alpha</code>	A numeric value between 0 and 1 specifying the significance level for the calculation of confidence or prediction intervals. The default is 0.05.
<code>alpha_pool</code>	A numeric value between 0 and 1 specifying 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.
<code>xform</code>	A vector of two character strings specifying the transformation of the response and the time variable. The default is “no” transformation, i.e. <code>c("no", "no")</code> ,

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 <code>xform</code> 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 $\leq 0$ are log or square root transformed. The default is <code>c(0, 0)</code> .
sf_option	A character string specifying if the limits ( <code>rl</code> or <code>sl</code> ) should be regarded as "tight" or "loose", i.e. either "tight" or "loose", respectively. The option "tight" means that the limits are rounded to the specified number of significant figures specified by the parameters <code>rl_sf</code> and <code>sl_sf</code> . In case of the option "loose" the limits are rounded to the specified number of significant figures ( $n$ ), followed by the subtraction of 1 from the $n^{\text{th}}$ digit and addition of 5 to the $(n + 1)^{\text{th}}$ digit if <code>ivl_side</code> is "lower", or followed by the addition of 4 to the $(n + 1)^{\text{th}}$ digit if <code>ivl_side</code> is "upper".
ivl	A character string of either "confidence" or "prediction", i.e. specifying the type of interval of interest. The default is "confidence".
ivl_type	A character string specifying 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 specifying if the "upper" or the "lower" limit is the relevant limit, i.e. either "upper" or "lower", respectively. The default is "lower".
...	Additional named or unnamed arguments passed on to <code>uniroot()</code> .

## 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.*" With respect to the number of batches to be included in the analysis it says that "*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 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 <code>alpha</code> , <code>alpha.pool</code> , <code>ivl</code> , <code>ivl.type</code> and <code>ivl.side</code> .
Variables	A list of the variable names, i.e. the original names of <code>batch_vbl</code> , <code>time_vbl</code> and <code>response_vbl</code> and, if applicable, of the transformed variables.
Model.Type	A list of five elements specifying which model, based on the ANCOVA analysis, suits best. The first element ( <code>type.spec</code> ) is a numeric vector of length 2 specifying the best model accepted at the significance level of 0.25. 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 ( <code>type.acronym</code> ) is an acronym representing the first item.
Models	A list of four elements named <code>cics</code> , <code>dics</code> , <code>dids</code> and <code>individual</code> . The first three elements contain the ‘lm’ objects of the “common intercept / common slope” ( <code>cics</code> ), “different intercept / common slope” ( <code>dics</code> ) and “different intercept / different slope” ( <code>dids</code> ) models. The fourth element is a list of the ‘lm’ objects of the models obtained from fitting the data of each batch individually.
AIC	A numeric named vector of the Akaike Information Criterion (AIC) values of each of the three fitted models.
BIC	A numeric named vector of the Bayesian Information Criterion (BIC) values of each of the three fitted models.
<code>wc.icpt</code>	A numeric value of the worst case intercept.
<code>wc.batch</code>	A numeric value of the worst case batch. In case of the <code>dids</code> model type, the estimation is done using the models obtained from fitting the data of each batch individually.
Limits	A list of all limits.
Intercepts	A list of the intercepts of the three fitted models.
POI	A numeric named vector of the POI values of each of the three fitted models. In case of the <code>dids</code> model type, the estimation is done using the models obtained from fitting the data of each batch individually.

### 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  $\alpha_{\text{pool}} = 0.25$  has 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 is not of practical relevance because the corresponding model is missing an effect. If the slopes are significantly different, there is no point comparing intercepts.



These requirements can be checked by aid of an “ANalysis of COVAriance” (ANCOVA) including the batch variable as 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 ( $\alpha_{\text{pool}} = 0.25$ , Type I error) is used to increase the power of the test to detect cases where the data should not be pooled. Setting  $\alpha_{\text{pool}} = 0.25$  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).

<https://www.ich.org/page/quality-guidelines>

## See Also

[expirest\\_wisle](#), [find\\_poi](#), [uniroot](#), [lm](#), [AIC](#), [BIC](#).

## Examples

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

# Successful estimations
res1 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
               response_vbl = "Potency", time_vbl = "Month",
               batch_vbl = "Batch", sl = 95, sl_sf = 3, srch_range = c(0, 500))
res1$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
# 26.22410 24.80030 23.34184
```

```

res2 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3, srch_range = c(0, 500))
res2$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
# 28.12518 22.47939 15.96453

# Moisture stability data (% (w/w)) of three batches of a drug product obtained
# over a 24 months period:
str(exp3)

# 'data.frame': 33 obs. of  3 variables:
# $ Batch   : Factor w/ 3 levels "b1","b2","b3": 1 1 1 1 1 1 1 1 1 1 ...
# $ Month   : num  0 1 2 3 3 6 6 12 12 24 ...
# $ Moisture: num  2.2 1.7 3.32 2.76 2.43 ...

# Unsuccessful estimations
# Intervals are wider than the specification limits (no intersection)
res3 <-
  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")
res3$POI

# (Expected) results in res3$POI
# cics dics dids
#  NA   NA   NA

# Estimation may also fail because of an inappropriate 'srch-range' setting.
res4 <-
  expirest_osle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
                response_vbl = "Potency", time_vbl = "Month",
                batch_vbl = "Batch", sl = 95, sl_sf = 3, srch_range = c(0, 5))
res4$POI

# (Expected) results in res4$POI
# cics dics dids
#  NA   NA   NA

```

---

expirest_wisle	<i>What-if (approach for) shelf life estimation (wisle)</i>
----------------	---

---

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

### Arguments

<code>data</code>	A data frame with the columns specified by <code>response_vbl</code> , <code>time_vbl</code> and <code>batch_vbl</code> .
<code>response_vbl</code>	A character string specifying the response variable name that must be a column of data.
<code>time_vbl</code>	A character string specifying the time variable name that must be a column of data.
<code>batch_vbl</code>	A character string specifying the column in data with the grouping information (i.e. a factorial variable) for the differentiation of the observations of the different batches.

<code>r1</code>	A numeric value or a numeric vector specifying the release specification limit(s) for which the corresponding expiry should be estimated.
<code>r1_sf</code>	A positive integer or a vector of positive integers specifying the number of “significant figures” (sf) of <code>r1</code> . It must have the same length as <code>r1</code> .
<code>s1</code>	A numeric value or a numeric vector of length 2 specifying the specification limit or limits. If a vector is provided it must be of the form <code>c(lower limit, upper limit)</code> .
<code>s1_sf</code>	A positive integer or a vector of positive integers specifying the number of “significant figures” (sf) of <code>s1</code> . It must have the same length as <code>s1</code> .
<code>srch_range</code>	A vector of length 2 specifying the end-points of the (time) range that is supposed to contain the shelf life or retest period.
<code>alpha</code>	A numeric value between 0 and 1 specifying the significance level for the calculation of confidence or prediction intervals. The default is <code>0.05</code> .
<code>alpha_pool</code>	A numeric value between 0 and 1 specifying the type I error rate for the test of the poolability of the batches. The default is <code>0.25</code> , following ICH Q1E guideline. The value should not be changed unless supported by well-founded reasons.
<code>xform</code>	A vector of two character strings specifying the transformation of the response and the time variable. The default is “no” transformation, i.e. <code>c("no", "no")</code> , 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).
<code>shift</code>	A vector of two values which will be added to the variables $x$ and/or $y$ before they are transformed as specified by the <code>xform</code> 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 $\leq 0$ are log or square root transformed. The default is <code>c(0, 0)</code> .
<code>sf_option</code>	A character string specifying if the limits ( <code>r1</code> or <code>s1</code> ) should be regarded as “tight” or “loose”, i.e. either “tight” or “loose”, respectively. The option “tight” means that the limits are rounded to the specified number of significant figures specified by the parameters <code>r1_sf</code> and <code>s1_sf</code> . In case of the option “loose” the limits are rounded to the specified number of significant figures ( $n$ ), followed by the subtraction of 1 from the $n^{th}$ digit and addition of 5 to the $(n + 1)^{th}$ digit if <code>ivl_side</code> is “lower”, or followed by the addition of 4 to the $(n + 1)^{th}$ digit if <code>ivl_side</code> is “upper”.
<code>ivl</code>	A character string of either “confidence” or “prediction”, i.e. specifying the type of interval of interest. The default is “confidence”.
<code>ivl_type</code>	A character string specifying 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”.
<code>ivl_side</code>	A character string specifying if the “upper” or the “lower” limit is the relevant limit, i.e. either “upper” or “lower”, respectively. The default is “lower”.
<code>...</code>	Additional named or unnamed arguments passed on to <code>uniroot()</code> .

## 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 <code>alpha</code> , <code>alpha.pool</code> , <code>ivl</code> , <code>ivl.type</code> and <code>ivl.side</code> .

Variables	A list of the variable names, i.e. the original names of <code>batch_vb1</code> , <code>time_vb1</code> and <code>response_vb1</code> and, if applicable, of the transformed variables.
Model.Type	A list of five elements specifying which model, based on the ANCOVA analysis, suits best. The first element ( <code>type.spec</code> ) is a numeric vector of length 2 specifying the best model accepted at the significance level of 0.25. 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”.
Models	A list of four elements named <code>cics</code> , <code>dics</code> , <code>dids</code> and <code>individual</code> . The first three elements contain the ‘lm’ objects of the “common intercept / common slope” ( <code>cics</code> ), “different intercept / common slope” ( <code>dics</code> ) and “different intercept / different slope” ( <code>dids</code> ) models. The fourth element is a list of the ‘lm’ objects of the models obtained from fitting the data of each batch individually.
AIC	A numeric named vector of the Akaike Information Criterion (AIC) values of each of the three fitted models.
BIC	A numeric named vector of the Bayesian Information Criterion (BIC) values of each of the three fitted models.
<code>wc.icpt</code>	A data frame of the worst case intercepts of each of the three fitted models.
<code>wc.batch</code>	A list of numeric value(s) of the worst case batch(es) per model type. In case of the <code>dids</code> model type, the estimation is done using the models obtained from fitting the data of each batch individually.
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. In case of the <code>dids</code> model type, the estimation of the POI values is done using the models obtained from fitting the data of each batch individually.

The POI data frame has the following columns:

<code>Intercept.cics</code>	The intercept of the worst case batch for the <code>cics</code> model.
<code>Intercept.dics</code>	The intercept of the worst case batch for the <code>dics</code> model.
<code>Intercept.dids</code>	The intercept of the worst case batch for the <code>dids</code> model.
<code>Delta.cics</code>	Absolute difference between the release and and the shelf life specification for the <code>cics</code> model.
<code>Delta.dics</code>	Absolute difference between the release and and the shelf life specification for the <code>dics</code> model.
<code>Delta.dids</code>	Absolute difference between the release and and the shelf life specification for the <code>dids</code> model.
<code>WCSL.cics</code>	WCSL for the <code>cics</code> model.
<code>WCSL.dics</code>	WCSL for the <code>dics</code> model.
<code>WCSL.dids</code>	WCSL for the <code>dids</code> model.
<code>Exp.Spec</code>	The (expiry) specification, i.e. the specification which is relevant for the determination of the expiry.
<code>Rel.Spec</code>	The calculated release specification.

Shelf.Life.cics	The estimated shelf life for the cics model.
Shelf.Life.dics	The estimated shelf life for the dics model.
Shelf.Life.dids	The estimated shelf life for the dids model.
POI.Model.cics	The POI of the cics model.
POI.Model.dics	The POI of the dics model.
POI.Model.dids	The POI of the dids model.

### 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  $\alpha_{\text{pool}} = 0.25$  has 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 is not of practical relevance because the corresponding model is missing an effect. If the slopes are significantly different, there is no point comparing intercepts.

These requirements can be checked by aid of an “ANalysis of COVAriance” (ANCOVA) including the batch variable as 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 ( $\alpha_{\text{pool}} = 0.25$ , Type I error) is used to increase the power of the test to detect cases where the data should not be pooled. Setting  $\alpha_{\text{pool}} = 0.25$  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

<https://www.tga.gov.au/stability-testing-prescription-medicines>

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).

<https://www.ich.org/page/quality-guidelines>

**See Also**

[expirest\\_osle](#), [find\\_poi](#), [uniroot](#), [lm](#), [AIC](#), [BIC](#).

**Examples**

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

# Successful estimations
res1 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
    response_vbl = "Potency", time_vbl = "Month",
    batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
    sl_sf = 3, srch_range = c(0, 500))
res1$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 Delta.cics Delta.dics
# 100.5669 100.3638 100.2491 3 3
# Delta.dids WCSL.cics WCSL.dics WCSL.dids Exp.Spec.Report Exp.Spec
# 3 97.56688 97.36375 97.24914 95 94.95
# Rel.Spec.Report Rel.Spec Shelf.Life.cics Shelf.Life.dics Shelf.Life.dids
# 98 97.95 14.07398 13.23176 13.34561
# POI.Model.cics POI.Model.dics POI.Model.dids
# 26.2241 24.8003 23.34184

res2 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b4", "b5", "b8"), ],
    response_vbl = "Potency", time_vbl = "Month",
    batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
    sl_sf = 3, srch_range = c(0, 500))
res2$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 Delta.cics Delta.dics
#      101.5498      100.4882      101.2594          3          3
# Delta.dids WCSL.cics WCSL.dics WCSL.dids Exp.Spec.Report Exp.Spec
#           3 98.54976 97.48822 98.25938          95 94.95
# Rel.Spec.Report Rel.Spec Shelf.Life.cics Shelf.Life.dics Shelf.Life.dids
#           98 97.95          13.03332          11.42141          7.619661
# POI.Model.cics POI.Model.dics POI.Model.dids
#      28.12518      22.47939      15.96453

# Unsuccessful estimation
res3 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
    response_vbl = "Potency", time_vbl = "Month",
    batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
    sl_sf = 3, srch_range = c(0, 500), alpha = 1E-16)
res3$POI

# (Expected) results in res3$POI
# Intercept.cics Intercept.dics Intercept.dids Delta.cics Delta.dics
#      100.5669          NA          NA          3          NA
# Delta.dids WCSL.cics WCSL.dics WCSL.dids Exp.Spec.Report Exp.Spec
#           NA 97.56688          NA          NA          95 94.95
# Rel.Spec.Report Rel.Spec Shelf.Life.cics Shelf.Life.dics Shelf.Life.dids
#           98 97.95          NA          NA          NA          NA
# POI.Model.cics POI.Model.dics POI.Model.dids
#      14.42089      2.480635          NA

# Estimation may also fail because of an inappropriate 'srch-range' setting.
res4 <-
  expirest_wisle(data = exp1[exp1$Batch %in% c("b2", "b5", "b7"), ],
    response_vbl = "Potency", time_vbl = "Month",
    batch_vbl = "Batch", rl = 98, rl_sf = 3, sl = 95,
    sl_sf = 3, srch_range = c(0, 5))
res4$POI

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

```

---

`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

<code>x</code>	An object of class <code>'plot_expirest_osle'</code> returned by the <a href="#">plot_expirest_osle()</a> function.
<code>...</code>	Further arguments passed to or from other methods or arguments that can be passed down to the <a href="#">formatC()</a> 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

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug  
# product obtained over a 24 months period:  
str(exp1)  
  
# 'data.frame': 53 obs. of 3 variables:  
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 ...  
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...  
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...  
  
# 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))

# 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.
gg1 <- plot_expirest_osle(
  model = res1, show_grouping = "no", response_vbl_unit = "%",
  y_range = c(93, 107), x_range = NULL, plot_option = "full",
  ci_app = "line")
gg2 <- plot(gg1)

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

```

---

```
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

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

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

# 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))

# 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.
gg1 <- plot_expirest_wisle(
  model = res1, rl_index = 1, show_grouping = "no",
  response_vbl_unit = "%", y_range = c(93, 107), x_range = NULL,
  scenario = "standard", plot_option = "full", ci_app = "line")
gg2 <- plot(gg1)

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

---

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,
  y_range,
```

```

  x_range = NULL,
  plot_option = "full",
  ci_app = "line"
)

```

## Arguments

model	An ‘expirest_osle’ object, i.e. a list returned by the <code>expirest_osle()</code> function.
show_grouping	A character string specifying if the grouping of the data should be taken into account (“yes”) or not (“no”), i.e. if the results of the most appropriate model should be shown or the results from the marginal analysis ignoring the grouping (being equivalent with the <i>common intercept / common slope case</i> ). The default is “yes”.
response_vbl_unit	A character string specifying the unit associated with the response variable. The default is NULL.
y_range	A numeric vector of the form <code>c(min, max)</code> specifying the range of the response variable to be plotted.
x_range	A numeric vector of the form <code>c(min, max)</code> specifying 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.
plot_option	A character string of either “full” or “lean”, i.e. specifying if all the 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”, specifying 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 <code>model</code> 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](#).

### Examples

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

# 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))

# 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, show_grouping = "no", response_vbl_unit = "%",
  y_range = c(93, 107), x_range = NULL, plot_option = "full",
  ci_app = "line")
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))
}

# Illustration of the grouping
gg2 <- plot_expirest_osle(
  model = res1, show_grouping = "yes", response_vbl_unit = "%",
```

```

  y_range = c(93, 107), x_range = c(0, 35), plot_option = "full",
  ci_app = "line")
gg2

# 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))

gg3 <- plot_expirest_osle(
  model = res2, show_grouping = "yes", response_vbl_unit = "%",
  y_range = c(83, 107), x_range = c(0, 43), plot_option = "full",
  ci_app = "ribbon")
gg3

```

---

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,
  y_range,
  x_range = NULL,
  scenario = "standard",
  plot_option = "full",
  ci_app = "line"
)

```

## Arguments

- |               |   |
|---------------|---|
| model         | An 'expirest_wisle' object, i.e. a list returned by the <code>expirest_wisle()</code> function.   |
| rl_index      | A positive integer specifying which of the release limit values that have been handed over to <code>expirest_wisle()</code> should be displayed. The default value is 1.  |
| show_grouping | A character string specifying if the grouping of the data should be taken into account ("yes") or not ("no"), i.e. if the results of the most appropriate model should be shown or the results from the marginal analysis ignoring the grouping (being equivalent with the <i>common intercept / common slope case</i> ). The default is "yes". |

response_vbl_unit	A character string specifying the unit associated with the response variable. The default is NULL.
y_range	A numeric vector of the form $c(\min, \max)$ specifying the range of the response variable to be plotted.
x_range	A numeric vector of the form $c(\min, \max)$ specifying 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.
scenario	A character string specifying 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”.
plot_option	A character string of either “full”, “lean1”, “lean2”, “basic1” and “basic2”, specifying if all the 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”, specifying 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.
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.
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](#).

**Examples**

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

# 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", r1 = 98, r1_sf = 3, sl = 95,
                sl_sf = 3, srch_range = c(0, 500))

# 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, r1_index = 1, show_grouping = "no",
  response_vbl_unit = "%", y_range = c(93, 107), x_range = NULL,
  scenario = "standard", plot_option = "full", ci_app = "line")
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))
}

# Illustration of the grouping
gg2 <- plot_expirest_wisle(
  model = res1, r1_index = 1, show_grouping = "yes",
  response_vbl_unit = "%", y_range = c(93, 107), x_range = NULL,
  scenario = "standard", plot_option = "full", ci_app = "line")
gg2

# 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", r1 = 98, r1_sf = 3, sl = 95,
sl_sf = 3, srch_range = c(0, 500))

gg3 <- plot_expirest_wisle(
  model = res2, r1_index = 1, show_grouping = "yes",
  response_vbl_unit = "%", y_range = c(89, 107), x_range = c(0, 26),
  scenario = "standard", plot_option = "full", ci_app = "ribbon")
gg3

```

---

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

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

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

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

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
# Worst case batch: NA
#
# Estimated shelf life for cics model: 26.2241

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
# Worst case batch: b5
#
# Estimated shelf life for dics model: 23.60194
```

```

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
# Worst case batch: b8
#
# Estimated shelf life for dids model: 15.96453

```

---

```
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

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

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

# 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))

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))

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))

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(s): 100.5669
# Worst case batch(es): NA
#
# Estimated shelf life (lives) for cics model:
# SL RL wisle osle
# cics 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
# (i.e. following ARGPM guidance).

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(s): 100.82
# Worst case batch(es): b5
#
# Estimated shelf life (lives) for dics model:
#   SL RL   wisle   osle
# cics 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
# (i.e. following ARGPM guidance).

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(s): 101.2594
# Worst case batch(es): b8
#
# Estimated shelf life (lives) for dids model:
#   SL RL   wisle   osle
# cics 95 98 7.619661 15.96453
#
# 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
# (i.e. following ARGPM guidance).
```

---

`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**

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

# 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))

# 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.
```

```

gg1 <- plot_expirest_osle(
  model = res1, show_grouping = "no", response_vbl_unit = "%",
  y_range = c(93, 107), x_range = NULL, plot_option = "full",
  ci_app = "line")
gg2 <- print(gg1)

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

```

---

```
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

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

```

# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

# 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", r1 = 98, r1_sf = 3, sl = 95,
                sl_sf = 3, srch_range = c(0, 500))

# 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.
gg1 <- plot_expirest_wisle(
  model = res1, r1_index = 1, show_grouping = "no",
  response_vbl_unit = "%", y_range = c(93, 107), x_range = NULL,
  scenario = "standard", plot_option = "full", ci_app = "line")
gg2 <- print(gg1)

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

```

---

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 <code>expirest_osle()</code> function.
...	Further arguments passed to or from other methods or arguments that can be passed down to the <code>formatC()</code> 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

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

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

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
# Worst case batch: NA
#
# Estimated shelf life for cics model: 26.2241

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
# Worst case batch: b5
#
# Estimated shelf life for dics model: 23.60194

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
# Worst case batch: b8
#
# Estimated shelf life for dids model: 15.96453
```

---

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 <code>expirest_wisle()</code> function.
...	Further arguments passed to or from other methods or arguments that can be passed down to the <code>formatC()</code> 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**

```
# Potency stability data (in % of label claim (LC)) of five batches of a drug
# product obtained over a 24 months period:
str(exp1)

# 'data.frame': 53 obs. of 3 variables:
# $ Batch : Factor w/ 6 levels "b2","b3","b4",...: 1 1 1 1 1 1 1 1 1 1 ...
# $ Month : num 0 1 3 3 6 6 12 12 24 24 ...
# $ Potency: num 101 101.3 99.8 99.2 99.5 ...

# 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", r1 = 98, r1_sf = 3, sl = 95,
    sl_sf = 3, srch_range = c(0, 500))
```

```
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))

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))

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(s): 100.5669
# Worst case batch(es): NA
#
# Estimated shelf life (lives) for cics model:
#   SL RL   wisle   osle
# cics 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
# (i.e. following ARGPM guidance).

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(s): 100.82
# Worst case batch(es): b5
#
# Estimated shelf life (lives) for dics model:
#   SL RL   wisle   osle
# cics 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
# (i.e. following ARGPM guidance).
```

```
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(s): 101.2594
# Worst case batch(es): b8
#
# Estimated shelf life (lives) for dids model:
#   SL RL   wisle   osle
# cics 95 98 7.619661 15.96453
#
# 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
# (i.e. following ARGPM guidance).
```

# Index

## \* datasets

- exp1, 2
- exp2, 3
- exp3, 4
- exp4, 5

AIC, 9, 16

BIC, 9, 16

exp1, 2

exp2, 3

exp3, 4

exp4, 5

expirest\_osle, 5, 16, 18, 20–22, 25–27, 29,  
31, 33, 34, 36

expirest\_wisle, 9, 11, 20, 22–25, 27–29, 32,  
34, 36

find\_poi, 9, 16

formatC, 18, 19, 26–29, 31–34, 36

ggplot, 18–21, 24, 31, 32

lm, 9, 16

methods, 18, 20, 27, 29, 31, 32, 34, 36

plot.plot\_expirest\_osle, 18

plot.plot\_expirest\_wisle, 19

plot\_expirest\_osle, 18, 20, 21, 31

plot\_expirest\_wisle, 19, 20, 23, 24, 32

print.expirest\_osle, 26

print.expirest\_wisle, 28

print.plot\_expirest\_osle, 30

print.plot\_expirest\_wisle, 32

summary.expirest\_osle, 33

summary.expirest\_wisle, 35

uniroot, 9, 16