Package 'drcte'

May 8, 2023

Date 2023-04-19

Type Package

Title Statistical Approaches for Time-to-Event Data in Agriculture

Version 1.0.30

Depends R (>= 4.0.0), drc

Imports methods, plyr, nor1mix, graphics, stats, mclust, survival, sandwich, lmtest, dplyr, multcomp, tidyr, MASS

Maintainer Andrea Onofri <andrea.onofri@unipg.it>

Description A specific and comprehensive framework for the analyses of time-to-event data in agriculture. Fit non-parametric and parametric time-to-event models. Compare time-to-event curves for different experimental groups. Plots and other displays. It is particularly tailored to the analyses of data from germination and emergence assays. The methods are described in Onofri et al. (2020) ``A unified framework for the analysis of germination, emergence, and other time-to-event data in weed science"", Weed Science, 70, 259-271 <doi:10.1017/wsc.2022.8>.

URL https://www.statforbiology.com

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 6.1.1

NeedsCompilation no

Author Andrea Onofri [aut, cre]

Repository CRAN

Date/Publication 2023-05-08 18:10:02 UTC

R topics documented:

`compCDF` *Compare time-to-event curves*

Description

This function is used to compare time-to-event curves for different experimental treatments. The apprach depends on the type of time-to-event curves, i.e., parametric, NPMLE and KDE (see details)

Usage

```
compCDF(obj, scores = c("wmw", "logrank1", "logrank2"),
B = 199, type = c("naive", "permutation"), units = NULL, upperl,
lowerl, display = TRUE)
```

Arguments

`obj`	A time-to-event model fitted with 'drmte' (drcte object)
`scores`	This argument is only meaningful for NPMLEs of time-to-event curves and it specifies the type of log-rank statistic to compare the curves. Possible values are 'wmw' (Wilcoxon-Mann-Whitney) , 'logrank1' (Sun's scores), 'logrank2' (Finkelstein's scores) and it defaults to 'wmw'. See details in Fay and Shaw, 2010.
`B`	Number of permutations for permutation tests
`type`	For parametric models, comparisons are based on a Likelihood Ratio Test (LRT). The argument type specifies whether P-levels should be calculated by using a 'naive' chi-square approximation, or a 'permutation' approach. It defaults to 'naive' and it is neglected for NPMLE and KDE fits, which always use the permutation approach.

units	A vector for the clustering units (e.g. Petri dishes or other containers for seed germination/emergence assays). When type = 'permutation' and 'units' is given, the permutation approach is performed at the units level and not at the individuals level.
upperl	Only for parametric models: a numeric vector of upper limits for all parameters in the model (the default corresponds to infinity for all parameters).
lowerl	Only for parametric models: a numeric vector of lower limits for all parameters in the model (the default corresponds to minus infinity for all parameters).
display	logical. If TRUE, results are displayed, otherwise, they are not (useful in simulations).

Details

This function is used to compare time-to-event curves fitted with the function 'drmte()' in the 'drcte' package, where treatment factors have been included by using the 'curveid' argument. The type of test statistic depends on the fitted model, whether it is a parametric time-to-event model or a NPMLE or a KDE. With parametric time-to-event models a Likelihood Ratio test is used, with NPMLE, weighted log-rank statistics are used, while for KDEs, a Cramér-von Mises type distance among curves is used. In all cases, this function is used to determine the p-level for the null (i.e. the curves are not significantly different from each other). WARNING: a permutation based approach may not necessarily work with parametric and KDE models; indeed, the permuted sample can be such that a model cannot be fitted into it. We are working a solution for these cases, but, in the meantime, we recommend that the user resorts to the naive inference approach whenever a permutation based approach does not work.

Value

method	The test statistic used for the comparison
scores	The individual scores for the statistic (when applicable)
val0	The observed value for the test statistic
vali	The observed values of the test statistic for the permutational samples
pval	The P-value for the LRT statistic, by using the chi square approximation
pvalb	The P-value for the test statistic, by using permutation methods
U	The score values for each group (only for Weighted Log-rank methods)
N	Sample size for each group (only for Weighted Log-rank methods)

Note

Some code is taken from the 'interval' package and from the 'binnednp' package (citations below).

Author(s)

Andrea Onofri

References

Bolker, BM (2008) Ecological models and data in R. Princeton University Press

Fay MP, Shaw PA (2010) Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The interval R Package. J Stat Softw 36:1–34

Barreiro-Ures D, Francisco-Fernández M, Cao R, Fraguela BB, Doallo R, González-Andújar JL, Reyes M (2019) Analysis of interval-grouped data in weed science: The binnednp Rcpp package. Ecol Evol 9:10903–10915

Examples

```
data(verbascum)

modVerb <- drmte(nSeeds ~ timeBef + timeAf, curveid = Species,
              fct = NPMLE(), data = verbascum)
compCDF(modVerb, units = verbascum$Dish)
```

`drmte` *Fitting time-to-event models for seed science*

Description

A general platform for the analysis of seed germination/emergence and other time-to-event data in agriculture.

Usage

```
drmte(formula, curveid, pmodels, data = NULL, subset, fct,
start, na.action = na.omit, logDose = NULL, type = "event",
control = drmteControl(), lowerl = NULL, upperl = NULL, separate = FALSE,
pshifts = NULL, varcov = NULL)
```

Arguments

`formula`	a symbolic description of the model to be fit. I must be in the form 'count start + end', where count is the number of events observed in the interval between 'start' and 'end'. Other variables can be specified after the 'start' and 'end', as needed to fit, e.g. hydro-time or thermal-time models.
`curveid`	a numeric vector or factor containing the grouping of the data.
`pmodels`	a data frame with a many columns as there are parameters in the non-linear function. Or a list containing a formula for each parameter in the nonlinear function.
`data`	an optional data frame containing the variables in the model.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.

fct	a list with three or more elements specifying the non-linear function, the accompanying self starter function, the names of the parameter in the non-linear function and, optionally, the first and second derivatives as well as information used for calculation of ED values. Currently available functions include, among others, the four- and five-parameter log-logistic models LL.4, LL.5 and the Weibull model W1.4. Use getMeanFunctions for a full list.
type	a character string specifying the distribution of the data (parameter estimation will depend on the assumed distribution as different log likelihood functions will be used). The default is "event", implying a multinomial distribution.
start	an optional numeric vector containing starting values for all mean parameters in the model. Overrules any self starter function.
na.action	a function for treating mising values ('NA's). Default is na.omit.
logDose	a numeric value or NULL. If log doses value are provided the base of the logarithm should be specified (exp(1) for the natural logarithm and 10 for 10-logarithm).
control	a list of arguments controlling constrained optimisation (zero as boundary), maximum number of iteration in the optimisation, relative tolerance in the optimisation, warnings issued during the optimisation.
lowerl	a numeric vector of lower limits for all parameters in the model (the default corresponds to minus infinity for all parameters).
upperl	a numeric vector of upper limits for all parameters in the model (the default corresponds to plus infinity for all parameters).
separate	logical value indicating whether curves should be fit separately (independent of each other).
pshifts	a matrix of constants to be added to the matrix of parameters. Default is no shift for all parameters.
varcov	an optional user-defined known variance-covariance matrix for the responses. Default is the identity matrix (NULL), corresponding to independent response values with a common standard deviation, which will be estimated from the data.

Details

This function relies on the general optimiser function optim for the minimisation of negative log likelihood function. The control arguments are specified using the function drmc.

Setting lowerl and/or upperl automatically invokes constrained optimisation.

The columns of a data frame argument to pmodels are automatically converted into factors. This does not happen if a list is specified.

Value

An object of class 'drcte' and 'drc'.

Author(s)

Andrea Onofri

Examples

```
data(verbascum)
modVerb <- drmte(nSeeds ~ timeBef + timeAf, curveid = Species,
            fct = NPMLE(), data = verbascum)
```

`drmteControl` *Sets control arguments*

Description

Set control arguments in the control argument in the function 'drmte()'.

Usage

```
drmteControl(constr = FALSE, errorm = TRUE, maxIt = 500, method="BFGS",
noMessage = FALSE, relTol = 1e-07, rmNA=FALSE, useD = FALSE,
trace = FALSE, otrace = FALSE, warnVal = -1, dscaleThres = 1e-15, rscaleThres = 1e-15,
conCheck = TRUE)
```

Arguments

`constr`	logical. If TRUE optimisation is constrained, only yielding non-negative parameters.
`errorm`	logical specifying whether failed convergence in `drm` should result in an error or only a warning.
`maxIt`	numeric. The maximum number of iterations in the optimisation procedure.
`method`	character string. The method used in the optimisation procedure. See `optim` for available methods.
`noMessage`	logical, specifying whether or not messages should be displayed.
`relTol`	numeric. The relative tolerance in the optimisation procedure.
`rmNA`	logical. Should NAs be removed from sum of squares used for estimation? Default is FALSE (not removed).
`useD`	logical. If TRUE derivatives are used for estimation (if available).
`trace`	logical. If TRUE the trace from `optim` is displayed.
`otrace`	logical. If TRUE the output from `optim` is displayed.
`warnVal`	numeric. If equal to 0 then the warnings are stored and displayed at the end. See under 'warn' in `options`. The default results in suppression of warnings.
`dscaleThres`	numeric value specifying the threshold for dose scaling.
`rscaleThres`	numeric value specifying the threshold for response scaling.
`conCheck`	logical, switching on/off handling of control measurements.

Value

A list with 8 components, one for each of the above arguments.

Note

none, yet.

Author(s)

Christian Ritz

Examples

```
### Displaying the default settings
drmteControl()
```

`ED.drcte` *Estimating the quantiles for time-to-event models*

Description

The function `ED` estimates the expected time to obtain a certain fraction (percentage) of individuals with events, for one or more time-to-event curves. With 'type = "relative"' the above fraction is regarded as relative to the maximum observed fraction. This function is included for compatibility with the 'drc' package, but the use of the 'quantile()' function is recommended, instead.

Usage

```
  ## S3 method for class 'drcte'
ED(object, respLev, interval = c("none", "delta", "boot"),
  clevel = NULL, level = ifelse(!(interval == "none"), 0.95, NULL),
  reference = c("control", "upper"), type = c("relative", "absolute"),
  lref, uref, bound = TRUE, od = FALSE, vcov. = vcov, display = TRUE,
  pool = TRUE, logBase = NULL, multcomp = FALSE,
  intType = "confidence", rate = FALSE, B = 200,
  seed = 1234, units = NULL, ...)
```

Arguments

`object`	an object of class 'drcte'.
`respLev`	a numeric vector containing the response levels. For 'type = "absolute"' 'respLev' must be included between 0 and 1, while for 'type = "relative"', 'respLev' expresses the percentage of maximum response and must be included between 0 and 100%.
`interval`	character string specifying the type of confidence intervals to be supplied. The default is "none". See Details below for more explanation.
`clevel`	character string specifying the curve id in case on estimates for a specific curve or compound is requested. By default estimates are shown for all curves.
`level`	numeric. The level for the confidence intervals. The default is 0.95.
`reference`	character string. Is the upper limit or the control level the reference?

type	character string. Whether the specified response levels are absolute or relative (default).
lref	numeric value specifying the lower limit to serve as reference.
uref	numeric value specifying the upper limit to serve as reference (e.g., 100%).
bound	logical. If TRUE only ED values between 0 and 100% are allowed. FALSE is useful for hormesis models.
od	logical. If TRUE adjustment for over-dispersion is used.
vcov.	function providing the variance-covariance matrix. vcov is the default, but sandwich is also an option (for obtaining robust standard errors).
display	logical. If TRUE results are displayed. Otherwise they are not (useful in simulations).
pool	logical. If TRUE curves are pooled. Otherwise they are not. This argument only works for models with independently fitted curves as specified in drm.
logBase	numeric. The base of the logarithm in case logarithm transformed dose values are used.
multcomp	logical to switch on output for use with the package multcomp (which needs to be activated first). Default is FALSE (corresponding to the original output).
intType	string specifying the type of interval to use with the predict method in case the type of confidence interval chosen with the argument "type" is "inverse regression."
rate	Logical: if FALSE quantiles for time-to-events are reported (default), if TRUE, rates (inverse of times) are reported.
B	Number of bootstrap resamples for bootstrap based inference
seed	seed for sampling methods
units	The name of a variable, coding for the experimental units, within which the observational units are clustered (e.g., Pretri dishes in germination assays)
...	see the details section below.

Details

There are several options for calculating confidence intervals through the argument interval. The option "none" (default) does not provide confidence intervals. The option "delta" results in asymptotical Wald-type confidence intervals (using the delta method and the normal distribution) and it is only available for parametric time-to-event models. The option "boot" provides bootraspped confidence intervals and it is only available for NPMLE or KDE curves; it may be rather slow. For (KDE models, the additional arguments lower and upper may be supplied. These arguments specify the lower and upper limits of the bisection method used to find the ED values. The lower and upper limits need to be smaller/larger than the EDx level to be calculated. The default limits are 0 and 1

Value

An invisible matrix containing the shown matrix with two or more columns, containing the estimates and the corresponding estimated standard errors and possibly lower and upper confidence limits. Or, alternatively, a list with elements that may be plugged directly into parm in the package *multcomp* (in case the argument multcomp is TRUE).

Author(s)

Andrea Onofri

See Also

The 'quantile()' function

Examples

```
start <- c(0, 6, 7, 10, 13, 16, 22, 23)
end <- c(6, 7, 10, 13, 16, 22, 23, Inf)
count <- c(3, 9, 3, 8, 1, 1, 5, 5)

# Fitting a non-parametric time-to-event model
mod <- drmte(count ~ start + end, fct = NPMLE())

# Getting the times required to reach three different fractions
# of individuals with events (i.e.: 0.1, 0.3 and 0.5)
ED(mod, respLev = c(0.10, 0.30, 0.50), type = "absolute")

# Getting the times required to reach three 10, 30 and 50%
# of the maximum fraction of individuals with events
ED(mod, respLev = c(10, 30, 50), type = "relative")
```

KDE *Kernel estimator for the cumulative distribution function*

Description

This function provides the kernel estimator for the distribution function for interval-grouped data.

Usage

```
KDE(bw = c("AMISE", "boot"))
KDE.fun(x, start, end, count, h)
```

Arguments

x	for time-to-event data, the predictor variable is time
start	for each assessment interval, this is the start time
end	for each assessment interval, this is the end time
count	for each assessment interval, this is the number of events
h	bandwidth value
bw	Method for bandwidth selection

Details

The function 'KDE.fun()' provides the kernel estimator for the distribution function, as modified for interval-grouped data. The equation is:

$$F(x) = \sum w_i * pnorm(\frac{x - t_i}{h})$$

where 'w_i' is the observed proportion of individuals getting the event in the i-th interval, 't_i' is the mid-point of the i-th time interval, 'x' is the time and 'h' is the bandwidth. The function KDE() is meant to be used with the 'drmte()' function, to estimate the optimal bandwidth, according to a bootstrap selector (Barreiro-Ures et al., 2019).

Value

The 'KDE.fun()' functions returns the CDF, for given values of time. The KDE() function returns a list containing the nonlinear function and other facilities, which are internally used by the 'drmte()' function.

Author(s)

Andrea Onofri. Codes for the estimation of optimal bandwidth have been largely taken from the 'binnednp' package (Barreiro-Ures et al., 2019).

References

Barreiro-Ures, D., Francisco-Fernández, M., Cao, R., Fraguela, B.B., Doallo, R., González-Andújar, J.L., Reyes, M., 2019. Analysis of interval-grouped data in weed science: The binnednp Rcpp package. Ecol Evol 9, 10903–10915.

Cao, R., Francisco-Fernández, M., Anand, A., Bastida, F., González-Andújar, J.L., 2013. Modeling Bromus diandrus Seedling Emergence Using Nonparametric Estimation. Journal of Agricultural, Biological, and Environmental Statistics 18, 64–86.

Cao, R., Francisco-Fernández, M., Anand, A., Bastida, F., González-Andújar, J.L., 2011. Computing statistical indices for hydrothermal times using weed emergence data. The Journal of Agricultural Science 149, 701–712.

Gonzalez-Andujar, J.L., Francisco-Fernandez, M., Cao, R., Reyes, M., Urbano, J.M., Forcella, F., Bastida, F., 2016. A comparative study between nonlinear regression and nonparametric approaches for modelling Phalaris paradoxa seedling emergence. Weed Research 56, 367–376.

Examples

```
data(chickweed)
mod <- drmte(count ~ start + end, data=chickweed,
             fct = KDE())
summary(mod)
plot(mod, ylim = c(0, 0.25), log = "")
```

`loglogistic` *Log-logistic distribution of germination times within a seed lot*

Description

This function provides the truncated log-logistic cumulative distribution of germination times, to be used to fit time-to-event methods to the time-course of seed germination for a seed lot.

Usage

```
loglogistic(fixed = c(NA, NA, NA), names = c("b", "d", "e"))
loglogisticSurv(fixed = c(NA, NA, NA), names = c("b", "d", "e"))
```

Arguments

`fixed`	numeric vector. Specifies which parameters are fixed and at what value they are fixed. NAs for parameter that are not fixed. It is often used to fix the 'd' parameter to 1, which provides the usual log-logistic distribution, with no ungerminated fraction.
`names`	a vector of character strings giving the names of the parameters. The default is reasonable.

Details

The log-logistic distribution of germination times is parameterised as:

$$P(t) = \frac{d}{(1 + \exp(-b(\log(t) - \log(e))))}$$

where 't' is the time and 'P(t)' is the proportion of germinated seeds at time = t. LogLogisticSurv is parameterised as usual in survival analysis:

$$P(t) = \frac{d}{(1 + \exp(-(\log(t) - e)/exp(b)))}$$

i.e. with 'e' on a log-scale and a log-link on 'b'. With no cured fraction (d = 1), loglogisticSurv gives same results as 'survereg' in the 'survival package', with 'dist = "loglogistic"'.

Value

The value returned is a list containing the nonlinear function, the self starter function and the parameter names.

Author(s)

Andrea Onofri

References

Kleinbaum, D.G., Klein, M., 2005. Survival analysis. Springer Science, USA.

Examples

```
library(drcte)
data(chickweed)
modb <- drmte(count ~ start + end,
              data=chickweed,
              fct = loglogistic())
```

`lognormal` *Log-normal distribution of times to an event*

Description

This function provides the truncated log-normal cumulative distribution of event times, to be used to fit time-to-event methods to the time-course of seed germination for a seed lot.

Usage

```
lognormal(fixed = c(NA, NA, NA), names = c("b", "d", "e"))
lognormalSurv(fixed = c(NA, NA, NA), names = c("b", "d", "e"))
```

Arguments

`fixed`	numeric vector. Specifies which parameters are fixed and at what value they are fixed. NAs for parameter that are not fixed. It is often used to fix the 'd' parameter to 1, which provides the usual log-logistic distribution, with no ungerminated fraction.
`names`	a vector of character strings giving the names of the parameters. The default is reasonable.

Details

The log-normal distribution of germination times is parameterised as:

$$P(t) = d * pnorm(b * (log(t + 0.000001) - log(d)))$$

where 't' is the time and 'P(t)' is the proportion of germinated seeds at time = t. LogNormalSurv is parameterised as usual in survival analysis:

$$P(t) = d * pnorm(b * (log(t + 0.000001) - d))$$

i.e. with 'e' on a log-scale and a log-link on 'b'. With no cured fraction (d = 1), logNormalSurv gives same results as 'survereg' in the 'survival package', with 'dist = "lognormal"'.

Value

The value returned is a list containing the nonlinear function, the self starter function and the parameter names.

Author(s)

Andrea Onofri

References

Kleinbaum, D.G., Klein, M., 2005. Survival analysis. Springer Science, USA.

Examples

```
library(drcte)
data(chickweed)
modb <- drmte(count ~ start + end,
              data=chickweed,
              fct = lognormal())
```

---

lotus *Germination assay with Lotus ornithopodioides*

Description

Seeds of L. ornithopodioides were harvested in three stages after anthesis and the time course of their germination was evaluated in Petri dishes. The dataset is organised as required by time-to-event analyses with the drc package.

Usage

```
data("lotus")
```

Format

A data frame with 192 observations on the following 7 variables.

`Stage` a character vector coding for the maturation stage

`Dish` a numeric vector coding for the Petri Dish

`timeBef` a numeric vector: beginning of scoring interval

`timeAf` a numeric vector: end of scoring interval

`count` a numeric vector: count of germinated seeds in each scoring interval

`nCum` a numeric vector: cumulative count of germinated seeds

`propCum` a numeric vector: cumulative proportions of germinated seeds

Details

There were 25 seeds per Petri dish. Seeds of Lotus ornithopodioides were harvested at the following stages (A) Green fleshy pods with brilliant gree sedds (27 Days After Anthesis; DAA), (B) green pergamenous pods with brown soft seeds (35 DAA) and (C) green pergamenous pods with brown seeds and moderately hard (41 DAA).

Source

Data are kindly provided by Fabio Gresta (University of Reggio Calabria, Italy).

References

Gresta, F., Avola, G., Onofri, A., Anastasi, U., Cristaudo, A., 2011. When Does Hard Coat Impose Dormancy in Legume Seeds? Lotus and Scorpiurus Case Study. Crop Science 51, 1739–1747.

Examples

```
data(lotus)
library(drcte)
#A constraint was put on the second curve (Stage B), in order to avoid an estimate higher than 1.
mod <- drmte(count ~ timeBef + timeAf, data=lotus, fct=LL.3(),
           curveid=Stage, upperl = c(NA, 1, NA))
summary(mod)
```

---

lotusCum *Germination assay with Lotus ornithopodioides*

Description

Seeds of L. ornithopodioides were harvested in three stages after anthesis and the time course of their germination was evaluated in Petri dishes. This is the dataset as required for nonlinear regression analysis and needs to be transformed into a form that is suitable for time-to-event analysis in R.

Usage

```
data("lotusCum")
```

Format

A data frame with 180 observations on the following 6 variables.

`Stage` a factor with levels `A` `B` `C`

`Dish` a numeric vector: code for Petri dishes

`Time` a numeric vector: inspection time in days

`nSeeds` a numeric vector: number of germinated seeds at each time

`nCum` a numeric vector: cumulative number of germinated seeds at each time

`Prop` a numeric vector: cumulative proportion of germinated seeds at each time

Details

There were 25 seeds per Petri dish. The columns represent the characteristics of each Petri dish. The columns from 1 to 15 represent the number of germinated seeds counted in each assessment time. Seeds of Lotus ornithopodioides were harvested at the following stages (A) Green fleshy pods with brilliant gree sedds (27 Days After Anthesis; DAA), (B) green pergamenous pods with brown soft seeds (35 DAA) and (C) green pergamenous pods with brown seeds and moderately hard (41 DAA).

Source

Data are kindly provided by Fabio Gresta (University of Reggio Calabria, Italy).

References

Gresta, F., Avola, G., Onofri, A., Anastasi, U., Cristaudo, A., 2011. When Does Hard Coat Impose Dormancy in Legume Seeds? Lotus and Scorpiurus Case Study. Crop Science 51, 1739–1747.

Examples

```
data(lotusCum)
data(lotusCum)
dataset_sd <- decumulate_te(lotusCum,
                            resp = nCum,
                            treat_cols = Stage,
                            monitimes = Time,
                            units = Dish,
                            n.subjects = rep(25, 12),
                            type = "count")

dataset_sd <- decumulate_te(lotusCum,
                            resp = Prop,
                            treat_cols = "Stage",
                            monitimes = Time,
                            units = Dish,
                            n.subjects = rep(25, 12),
                            type = "proportion")
```

---

lotusOr *Germination assay with Lotus ornithopodioides*

Description

Seeds of L. ornithopodioides were harvested in three stages after anthesis and the time course of their germination was evaluated in Petri dishes. This is the original dataset, as collected by the technicians and needs to be transformed into a form that is suitable for time-to-event analysis in R.

Usage

```
data("lotusOr")
```

Format

A data frame with 12 observations on the following 17 variables.

`Stage` a factor with levels `A` `B` `C`

`Dish` a numeric vector: code of Petri dish

'`1`' a numeric vector: counts at day 1

'`2`' a numeric vector: counts at day 2

'`3`' a numeric vector: counts at day 3

'`4`' a numeric vector: counts at day 4

'`5`' a numeric vector: counts at day 5

'`6`' a numeric vector: counts at day 6

'`7`' a numeric vector: counts at day 7

'`8`' a numeric vector: counts at day 8

'`9`' a numeric vector: counts at day 9

'`10`' a numeric vector: counts at day 10

'`11`' a numeric vector: counts at day 11

'`12`' a numeric vector: counts at day 12

'`13`' a numeric vector: counts at day 13

'`14`' a numeric vector: counts at day 14

'`15`' a numeric vector: counts at day 15

Details

Every line of data represents a Petri dish. There were 25 seeds per Petri dish. The columns represent the characteristics of each Petri dish. The columns from 1 to 15 represent the number of germinated seeds counted in each assessment time. Seeds of Lotus ornithopodioides were harvested at the following stages (A) Green fleshy pods with brilliant gree sedds (27 Days After Anthesis; DAA), (B) green pergamenous pods with brown soft seeds (35 DAA) and (C) green pergamenous pods with brown seeds and moderately hard (41 DAA).

Source

Data are kindly provided by Fabio Gresta (University of Reggio Calabria, Italy).

References

Gresta, F., Avola, G., Onofri, A., Anastasi, U., Cristaudo, A., 2011. When Does Hard Coat Impose Dormancy in Legume Seeds? Lotus and Scorpiurus Case Study. Crop Science 51, 1739–1747.

Examples

```
data(lotusOr)
datasetG <- melt_te(lotusOr, count_cols = 3:17, treat_cols = Stage,
            monitimes = 1:15, n.subjects = rep(25,12))
head(datasetG, 16)
```

NPMLE *NPMLE for time-to-event data*

Description

This function provides the non-parametric maximum likelihood estimator for time-to-event data for the distribution function for interval-grouped data.

Usage

```
NPMLE()
```

Arguments

This function has no arguments

Details

The function 'KDE()' provides the NPMLE for the cumulative distribution function of time-to-event data. It uses the algorithms proposed in the 'interval' package (Fay and Shaw, 2010)

Value

The 'KDE()' function returns a list containing the nonlinear function and other facilities, which are internally used by the 'drmte()' function.

Author(s)

Andrea Onofri. Codes for the estimation have been largely taken from the 'interval' package (Fay and Shaw, 2010).

References

Michael P. Fay, Pamela A. Shaw (2010). Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The interval R Package. Journal of Statistical Software, 36(2), 1-34. URL https://www.jstatsoft.org/v36/i02/.

Examples

```
data(chickweed)
mod <- drmte(count ~ start + end, data=chickweed,
             fct = NPMLE())
summary(mod)
plot(mod, ylim = c(0, 0.25), log = "")
```

plot.drcte *Plotting fitted time-to-event models*

Description

plot displays fitted curves and observations in the same plot window, distinguishing between curves by different plot symbols and line types.

Usage

```
  ## S3 method for class 'drcte'
plot(x, ..., add = FALSE, level = NULL, shading = TRUE,
  type = "all",
  npmle.type = c("interpolation", "midpoint", "right", "left", "none"),
  npmle.points = FALSE, kde.points = TRUE,
  broken = FALSE, bp, bcontrol = NULL, conName = NULL, axes = TRUE,
  gridsize = 100, log = "", xtsty, xttrim = TRUE,
  xt = NULL, xtlab = NULL, xlab, xlim,
  yt = NULL, ytlab = NULL, ylab, ylim,
  cex, cex.axis = 1, col = FALSE, lty, pch,
  legend, legendText, legendPos, cex.legend = 1,
  normal = FALSE, normRef = 1, confidence.level = 0.95)
```

Arguments

x	an object of class 'drcte'.
...	additional graphical arguments. For instance, use lwd=2 or lwd=3 to increase the width of plot symbols.
add	logical. If TRUE then add to already existing plot.
level	vector of character strings. To plot only the curves specified by their identification number.
shading	For NPMLE, it prints shading on the graphs for the points where the likelihood value is not unique. Defaults to TRUE
type	it has been left for analogy with the plot method for 'drc' objects, but it is neglected in the case of 'drcte' objects.
npmle.type	the NPMLE of the cumulative density function is only specified at the end of each inspection interval, while it is not unique within each interval. This argument specifies how the CDF increases within each interval: possible values are "interpolation" (it is assumed that the CDF increases progressively), "left" (the CDF increases at the beginning of each interval), "right" (the CDF increases at the end of each interval; it is very common in survival analysis) and "midpoint" (the CDF increases in the middle of each interval; it is very common in survival analysis). This argument is neglected with parametric and KDE fits.

npmle.points	If F, with NPMLE fits, plotting the NPMLEs at the end of each interval as symbols is suppressed. Not implemented, yet.
kde.points	If F, with KDE fits, plotting the NPMLE at the end of each interval is suppressed
broken	logical. If TRUE the x axis is broken provided this axis is logarithmic (using functionality in the CRAN package 'plotrix').
bp	numeric value specifying the break point below which the dose is zero (the amount of stretching on the dose axis above zero in order to create the visual illusion of a logarithmic scale *including* 0). The default is the base-10 value corresponding to the rounded value of the minimum of the log10 values of all positive dose values. This argument is only working for logarithmic dose axes.
bcontrol	a list with components `factor`, `style` and `width`. Controlling the appearance of the break (in case `broken` is `TRUE`). The component `factor` is the distance from the control to the break as a multiple of the value of `bp` (default is 2). The component `style` can take the values: `gap`, `slash` or `zigzag`. The component `width` is the width of the break symbol (default is 0.02).
conName	character string. Name on x axis for dose zero. Default is '"0"'.
axes	logical indicating whether both axes should be drawn on the plot.
gridsize	numeric. Number of points in the grid used for plotting the fitted curves.
log	a character string which contains '"x"' if the x axis is to be logarithmic, '"y"' if the y axis is to be logarithmic and '"xy"' or '"yx"' if both axes are to be logarithmic. The default is "x". The empty string "" yields the original axes.
xtsty	a character string specifying the dose axis style for arrangement of tick marks. By default ("base10") For a logarithmic axis by default only base 10 tick marks are shown ("base10"). Otherwise sensible equidistantly located tick marks are shown ("standard"), relying on `axTicks`.
xttrim	logical specifying if the number of tick marks should be trimmed in case too many tick marks are initially determined.
xt	a numeric vector containing the positions of the tick marks on the x axis.
xtlab	a vector containing the tick marks on the x axis.
xlab	an optional label for the x axis.
xlim	a numeric vector of length two, containing the lower and upper limit for the x axis.
yt	a numeric vector, containing the positions of the tick marks on the y axis.
ytlab	a vector containing the tick marks on the y axis.
ylab	an optional label for the y axis.
ylim	a numeric vector of length two, containing the lower and upper limit for the y axis.
cex	numeric or numeric vector specifying the size of plotting symbols and text (see `par` for details).
cex.axis	numeric value specifying the magnification to be used for axis annotation relative to the current setting of cex.
col	either logical or a vector of colours. If TRUE default colours are used. If FALSE (default) no colours are used.

legend	logical. If TRUE a legend is displayed.
legendText	a character string or vector of character strings specifying the legend text (the position of the upper right corner of the legend box).
legendPos	numeric vector of length 2 giving the position of the legend.
cex.legend	numeric specifying the legend text size.
lty	a numeric vector specifying the line types.
pch	a vector of plotting characters or symbols (see points).
normal	logical. If TRUE the plot of the normalized data and fitted curves are shown (for details see Weimer et al. (2012) for details).
normRef	numeric specifying the reference for the normalization (default is 1).
confidence.level	confidence level for error bars. Defaults to 0.95.

Details

The plot method for 'drcte' objects inherits from the plot method for 'drc' objects and adds functionalities for nonparametric time-to-event fits. For parametric time-to-event models, the fitted curve is presented, together with symbols, corresponding to the NPMLE estimator of the cumulative distribution function at the end of each observation interval. For NPMLE fits, the cumulative CDF at the end of each interval is presented and the way that the density increases during the interval is specified through the argument 'npmle.type'. For KDE fits, the cumulative density is presented as a line, while the observed NPMLE of the CDF at the end of each interval can be either presented as symbols or suppressed. For all other information, please consult ?plot.drc. This method does not work for models with environmental covariates.

Value

An invisible data frame with the values used for plotting the fitted curves. The first column contains the dose values, and the following columns (one for each curve) contain the fitted response values.

Author(s)

Andrea Onofri

References

Weimer, M., Jiang, X., Ponta, O., Stanzel, S., Freyberger, A., Kopp-Schneider, A. (2012) The impact of data transformations on concentration-response modeling. *Toxicology Letters*, **213**, 292–298.

Examples

```
library(drcte)
data(verbascum)
mod <- drmte(nSeeds ~ timeBef + timeAf, fct = NPMLE(),
                data = verbascum, curveid = Species)
plot(mod)
```

plotData *Data for plotting fitted time-to-event models*

Description

plotData returns the data for plotting a regression model with other plotting packages, such as ggplot2

Usage

```
plotData(x, xlim, confidence.level = 0.95, gridsize = 100,
type = c("average", "all", "bars", "none", "obs", "confidence"),
npmle.type = c("interpolation", "midpoint", "right", "left","none"))
```

Arguments

x	an object of class 'drcte' or 'drc'.
xlim	as in the basic plot method: the x-range for plotting
confidence.level	For confidence intervals
gridsize	the gridsize for predictions
type	it has been left for analogy with the plot method for 'drc' objects, but it is neglected in the case of 'drcte' objects.
npmle.type	the NPMLE of the cumulative density function is only specified at the end of each inspection interval, while it is not unique within each interval. This argument specifies how the CDF increases within each interval: possible values are "interpolation" (it is assumed that the CDF increases progressively), "left" (the CDF increases at the beginning of each interval), "right" (the CDF increases at the end of each interval; it is very common in survival analysis) and "midpoint" (the CDF increases in the middle of each interval; it is very common in survival analysis). This argument is neglected with parametric and KDE fits.

Details

The plotData method for 'drcte' objects is used to get the data for plotting with ggplot2.

Value

A list with two elements. The first element (plotPoints) contains the data for plotting the observed values, the second element (plotFits) contains the data for plotting curves.

Author(s)

Andrea Onofri

Examples

```
library(drcte)
data(verbascum)
mod <- drmte(nSeeds ~ timeBef + timeAf, fct = LL.3(),
                 data = verbascum, curveid = Species)
plotData(mod)
```

`predict.drcte` *Prediction*

Description

Predicting probabilities for models of class 'drcte'.

Usage

```
  ## S3 method for class 'drcte'
predict(object, newdata, se.fit = FALSE,
  interval = FALSE,
  level = 0.95, na.action = na.pass,
  npmle.type = c("interpolation", "left", "right", "midpoint"),
  robust = FALSE, units = NULL, B = 200, ...)
  ## S3 method for class 'list'
predict(object, newdata, coefs, vcov. = NULL, ...)
```

Arguments

`object`	an object of class 'drcte' or a list representing a time-to-event model (e.g.: LL.3(), LN.3()...)
`newdata`	A data frame in which to look for variables with which to predict. The first variable, must always be the time at which predictions are sought; if necessary, other covariates can be added in succeeding columns, in the same order as they are specified by the fitted model. For models where time is the only predictor, 'newdata' can also be provided as a vector.
`se.fit`	logical. If TRUE standard errors are provided.
`interval`	logical. If TRUE confidence intervals are provided.
`level`	Confidence level.
`na.action`	function determining what should be done with missing values in 'newdata'. The default is to predict 'NA'.
`npmle.type`	character string relating to the type of prediction for NPMLE. For interval censored observations there is not a unique MLE for a specific time, and, therefore, predictions can be obtained by three methods: "interpolation" (default; predictions are sought on the line connecting the two points bounding the non-unique MLE interval), "left" (taking the left side of the non-unique MLE interval) or "right" (taking the right side of the non-unique MLE interval. The argument is neglected for parametric or kernel based time-to-event models.

robust	a logical value. If TRUE, robust confidence interval are provided (see below for detail)
units	a vector coding for randomisation units or other clustering structures. It causes a cluster robust standard error to be provided.
B	Number of bootstrap resamples for bootstrap based inference
coefs	The coefficients of a parametric fit
vcov.	Variance-covariance matrix
...	further arguments passed to or from other methods.

Details

With parametric models, 'naive' (asymptotic) standard errors are obtained by using the 'delta method, in association to the second derivative of the likelihood function. For seed germination and emergence assays, the observational units are the individual seeds and, most often, they are clustered within randomisation units (e.g., Petri dishes or other types of containers), to which the experimental treatments are allocated. For these and other cases of clustered data, cluster-robust SEs can be obtained by using the so-called 'sandwich estimator' (Carroll et al. 1998), which has proven reliable for clustered survival data (Yu and Peng 2008). Robust SEs are obtained by exploiting the facilities provided in the 'sandwich' package (Zeileis et al. 2020).

Value

A vector (when 'newdata' is missing and 'se.fit' = 'interval' = FALSE) or a data.frame with as many rows as in 'newdata' (or as in the original dataset, when 'newdata' is missing) and a set of columns containing predictors, predictions, standard errors, lower and upper limits of confidence intervals.

Author(s)

Andrea Onofri, borrowing code from Christian Ritz

References

Carroll RJ, Wang S, Simpson DG, Stromberg AJ, Ruppert D (1998) The sandwich (robust covariance matrix) estimator. Technical report, Department of statistics. A&M University. Texas (USA)

Onofri, A., Mesgaran, M., & Ritz, C. (2022). A unified framework for the analysis of germination, emergence, and other time-to-event data in weed science. Weed Science, 1-13. doi:10.1017/wsc.2022.8

Yu B, Peng Y (2008) Mixture cure models for multivariate survival data. Comput Stat Data Anal 52:1524–1532

Zeileis A, Koell S, Graham N (2020) Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R. J Stat Softw 95

Examples

```
library(drcte)
data(chickweed)
modb <- drmte(count ~ start + end,
              data=chickweed,
```

```
               fct = LL.3())
predict(modb, se.fit = TRUE, interval = TRUE,
newdata = data.frame(time = c(1, 10, 100, 250)))
```

---

quantile.drcte *Estimating quantiles for time-to-event models*

Description

This function estimates the quantiles for time-to-events data, either for the whole sample (restricted = FALSE) or only for the fraction of individuals with events (restricted = TRUE). The quantiles for rates (the inverse of times) are also available, by using the 'rate = TRUE' option. The quantiles represent the time/rate to reach a given fraction of individuals with event

Usage

```
  ## S3 method for class 'drcte'
quantile(x, probs, restricted = FALSE, rate = FALSE,
                           interval = FALSE,
                           level = ifelse(!(interval == "none"), 0.95, NULL),
                           robust = FALSE, B = 999, units = NULL,
                           display = TRUE, ...)
```

Arguments

x	an object of class 'drcte'.
probs	a vector with the sought probabilities; the values must be included between 0 and 1.
restricted	a logical value: if FALSE, the quantiles are calculated for the whole sample (the default); if TRUE, only the fraction of individuals with events is considered.
rate	a logical value: if FALSE quantiles for time-to-events are reported (default), if TRUE, rates (inverse of times) are reported.
interval	a logical value: if TRUE, confidence intervals are supplied. The default is FALSE. See Details below for more explanation.
level	numeric. The level for the confidence intervals. The default is 0.95.
robust	a logical value: if TRUE, robust standard error and confidence intervals are reported. The default is FALSE.
B	Number of bootstrap resamples for bootstrap based inference
units	a vector coding for randomisation units or other clustering structures. It causes a cluster robust standard error to be provided.
display	a logical value: if TRUE (default), the results are displayed at the end of calculations.
...	further arguments passed to or from other methods. In particular, it is used to pass environmental covariates, wherever necessary

Details

Quantiles for parametric models are calculated by using the inverse of the cumulative probability function. For NPMLE fits, it is assumed that events are evenly spread within each Turnbull interval and the quantiles are the values corresponding to the abscissas of the points where the horizontal line starting from the selected percentile crosses the time-to-event line. For KDE fits, quantiles are calculated from the time-to-event curve by the bisection method, which can be rather slow. If requested, confidence intervals are calculated, according to the type of time-to-event model. For parametric models, standard errors are calculated by using the delta method with Wald-type confidence intervals (normal distribution). If robust = TRUE, a sandwich estimator of the variance-covariance matrix is used, which is in its cluster-robust form, if a 'units' variable is provided, containing the coding fro Petri dishes, other grouping structures. For NPMLE fits, bootrapped confidence intervals are provided, based on B resamples (default to 1000); if the 'units' argument is provided, the cluster-robust version of the bootstrap is used. For KDE fits, standard errors and confidence intervals are not yet provided.

Value

An invisible matrix containing the shown matrix with two or more columns, containing the estimates and the corresponding estimated standard errors and possibly lower and upper confidence limits.

Author(s)

Andrea Onofri

See Also

The 'ED()' function.

Examples

```
start <- c(0, 6, 7, 10, 13, 16, 22, 23)
end <- c(6, 7, 10, 13, 16, 22, 23, Inf)
count <- c(3, 9, 3, 8, 1, 1, 5, 5)

# Fitting a non-parametric time-to-event model
mod <- drmte(count ~ start + end, fct = NPMLE())

# Getting the times required to reach three different fractions
# of individuals with events (i.e.: 0.1, 0.3 and 0.5), either including
# the whole lot of individuals, or only the individuals with events
quantile(mod, probs = c(0.10, 0.30, 0.50))
quantile(mod, probs = c(0.10, 0.30, 0.50), restricted = TRUE)
```

reshape_te *Reshaping time-to-event datasets*

Description

This is a set of functions that can be used to transform time-to-event datasets into a form that is amenable for data analyses with the 'drc' and 'drcte' packages

Usage

```
melt_te(data = NULL, count_cols, treat_cols, monitimes,
n.subjects = NULL, grouped = TRUE)
decumulate_te(data = NULL, resp, treat_cols, monitimes,
units, n.subjects, type = c("count", "proportion"))
group_te(data)
ungroup_te(data, counts)
```

Arguments

data	a data frame to be reshaped. In this data frame, each row represents a container (e.g., a Petri dish) and each column represents a piece of information about that container, including the counts of germinated seeds at each inspection time.
count_cols	a numeric vector, specifying the positions (not names) of the columns containing the counts in 'data'
treat_cols	a vector, specifying the positions/names of the columns containing the treatment levels in 'data'
monitimes	a numeric vector of monitoring times. Must be of same length as 'count_cols'
n.subjects	a numeric vector listing the number of viable seeds for each container (e.g., Petri dish). It may be either of length one (if it is the same for all randomisation units) or it must have same length as the number of rows in 'data' and 'treat_cols'. With the function 'melt_te()', it is also possible to pass a reference to the name of a variable in 'data', containing the number of individuals. If missing, it is assumed that all individuals experienced the event and there is no 'cured' fraction.
grouped	logical: specifying whether the output should be in LONG GROUPED or LONG UNGROUPED format
counts	the name of a numeric vector listing the counts of events in each interval. It should be available in data
resp	For 'decumulate_te(): the response variable, i.e. a vector of cumulative counts/proportions
units	For 'decumulate_te(): a label for Petri dishes or other containers
type	For 'decumulate_te(): character string, it is "count" or "proportion", depending on what it is given in the argument 'resp'

Details

Details are given in https://www.statforbiology.com/2021/stat_drcte_3-reshapingdata/

Value

Returns a data frame

Author(s)

Andrea Onofri

References

See https://www.statforbiology.com/2021/stat_drcte_3-reshapingdata/

Examples

```
# Transform a dataset from WIDE to LONG GROUPED
library(drcte)
data(lotusOr)
datasetG <- melt_te(lotusOr, count_cols = 3:17, treat_cols = Stage,
            monitimes = 1:15, n.subjects = rep(25,12))
head(datasetG, 16)

# Transform a dataset from WIDE to LONG UNGROUPED
datasetU <- melt_te(lotusOr, count_cols = 3:17, treat_cols = 1,
                monitimes = 1:15, n.subjects = rep(25,12),
                grouped = FALSE)
head(datasetU, 16)

# From LONG GROUPED to LONG UNGROUPED
datasetU2 <- ungroup_te(datasetG, count)[,-c(5, 6)]
head(datasetU2, 16)

# From LONG UNGROUPED to LONG GROUPED
datasetG2 <- group_te(datasetU)
head(datasetG2, 16)

# Chunk 7
# Decumulate a dataset with cumulative counts
data(lotusCum)
dataset_sd <- decumulate_te(lotusCum,
                            resp = nCum,
                            treat_cols = Stage,
                            monitimes = Time,
                            units = Dish,
                            n.subjects = rep(25, 12),
                            type = "count")

dataset_sd <- decumulate_te(lotusCum,
                            resp = Prop,
                            treat_cols = "Stage",
                            monitimes = Time,
                            units = Dish,
                            n.subjects = rep(25, 12),
                            type = "proportion")
```

Service functions *Service functions*

Description

The functions documented here are internally used by the package and they are not to be used elsewhere

Usage

```
NLSLL.3(predictor, b, d, ED50)
```

Arguments

predictor	response variable
b	shepe parameter
d	truncation parameter
ED50	location parameter

summary.drcte *Summarising time-to-event fits*

Description

'summary' returns a comprehensive summary for model parameters from a parametric time-to-event fit object of class 'drcte'. For non-parametric models (NPLME), this function reports the set of Turnbull's interval and related masses, while for KDEs it reports the AMISE or bootstrap bandwidth value.

Usage

```
  ## S3 method for class 'drcte'
summary(object, robust = FALSE, units = NULL,
  type = c("sandwich", "bootstrap", "jackknife"), ...)
```

Arguments

object	an object of class 'drcte'.
robust	Logical: if TRUE, robust sandwich standard errors are printed instead of the asymptotic formula. Defaults to TRUE if 'units' or 'type' is given. It is neglected for non-parametric time-to-event models
units	Optional vector that identifies groups of subjects, used in computing a cluster robust standard error. Like model variables, this is searched for in the dataset pointed to by the data argument. It is neglected for non-parametric time-to-event models.
type	Type of robust standard error. It defaults to 'sandwich' that is the only available method at moment. Bootstrap and jackknife standard errors are under study. It is neglected for non-parametric time-to-event models
...	additional arguments.

Details

With parametric models, 'naive' (asymptotic) standard errors are obtained from the second derivative of the likelihood function. For seed germination and emergence assays, the observational units are the individual seeds and, most often, they are not independent from one another, as they are clustered within randomisation units (e.g., Petri dishes or other types of containers), to which the experimental treatments are allocated. For these and other cases of clustered data, cluster-robust SEs can be obtained by using the so-called 'sandwich estimator' (Carroll et al. 1998), which has proven reliable for clustered survival data (Yu and Peng 2008). Robust SEs are obtained by exploiting the facilities provided in the 'sandwich' package (Zeileis et al. 2020).

Value

A list of summary statistics that includes parameter estimates and estimated standard errors.

Author(s)

Andrea Onofri, Christian Ritz

References

Carroll RJ, Wang S, Simpson DG, Stromberg AJ, Ruppert D (1998) The sandwich (robust covariance matrix) estimator. Technical report, Department of statistics. A&M University. Texas (USA)

Onofri, A., Mesgaran, M., & Ritz, C. (2022). A unified framework for the analysis of germination, emergence, and other time-to-event data in weed science. Weed Science, 1-13. doi:10.1017/wsc.2022.8

Yu B, Peng Y (2008) Mixture cure models for multivariate survival data. Comput Stat Data Anal 52:1524–1532

Zeileis A, Koell S, Graham N (2020) Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R. J Stat Softw 95

verbascum *Germination time-course for three species of the Verbascum genus*

Description

This dataset relates to a germination assay with three species of Verbascum, four replicated Petri dishes and 25 seeds per Petri dish (12 dishes in all). Inspections were made daily for 15 days.

Usage

```
data("verbascum")
```

Format

A data frame with 192 observations on the following 6 variables.

`Dish` a numeric vector with the coding for Petri dishes

`Species` a factor with the name of the species

`timeBef` a numeric vector with the start date for the each assessment interval

`timeAf` a numeric vector with the end date for the each assessment interval. It contains the value Inf (Infinity), that indicates the final interval of time, after the final assessment time

`nSeeds` a numeric vector with the number of germinated seeds in each interval. Relating to final interval of time, from the last assessment date to Infinity, 'nSeeds' contains the final number of ungerminated seeds

`nCum` a numeric vector with the cumulative number of germinated seeds in each interval of time

Source

Data are kindly provided by Antonietta Cristaudo (University of Catania, Italy).

References

Catara, S., Cristaudo, A., Gualtieri, A., Galesi, R., Impelluso, C., Onofri, A., 2016. Threshold temperatures for seed germination in nine species of Verbascum (Scrophulariaceae). Seed Science Research 26, 30–46.

Examples

```
data(verbascum)
head(verbascum)
```

Index