Package ‘biostat3’

November 5, 2019

Type Package

Title Utility Functions, Datasets and Extended Examples for Survival Analysis

Version 0.1.4

Date 2019-11-04

Author Annika Tillander [ctb],
Andreas Karlsson [aut],
Johan Zetterqvist [ctb],
Peter Strom [ctb],
Benedicte Delcoigne [ctb],
Mark Clements [aut, cre]

Maintainer Mark Clements <mark.clements@ki.se>

Description Utility functions, datasets and extended examples for survival analysis. This extends a range of other packages, some simple wrappers for time-to-event analyses, datasets, and extensive examples in HTML with R scripts. The package also supports the course Biostatistics III entitled "Survival analysis for epidemiologists in R".

Depends survival, R (>= 2.10), MASS

Imports muhaz, car, graphics, stats

Suggests bshazard, rstpm2, Epi, dplyr

License GPL (>= 2)

LazyData yes

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2019-11-05 21:50:02 UTC

R topics documented:

biostat3-package ........................................... 2
addIndicators ............................................. 3
brv ................................................. 3
Utility functions, datasets and extended examples for survival analysis. This extends a range of other packages, some simple wrappers for time-to-event analyses, datasets, and extensive examples in HTML with R scripts. The package also supports the course Biostatistics III entitled "Survival analysis for epidemiologists in R".

**Author(s)**

Annika Tillander [ctb], Andreas Karlsson [aut], Johan Zetterqvist [ctb], Peter Strom [ctb], Benedicte Delcoigne [ctb], Mark Clements [aut, cre]

Maintainer: Mark Clements <mark.clements@ki.se>

**Examples**

```r
plot(muhaz2(Surv(surv_mm, status == "Dead: cancer")~1, melanoma))
```
addIndicators

Utility to add indicators from a data-frame based on a formula.

Description

Column-bind a model matrix to the source data-frame

Usage

addIndicators(data, formula, drop.intercept = TRUE)

Arguments

data source data-frame or matrix.
formula model formula used to add columns.
drop.intercept logical as to whether to drop the column named '(Intercept)'.

Details

This function calls model.matrix, conditionally checks for and removes '(Intercept)', and binds with the original data-frame (or matrix).

Value

data-frame or matrix.

Examples

addIndicators(data.frame(f = c("a", "a", "b")), ~f+0)

brv Bereavement dataset

Description

Bereavement dataset

Usage

data("brv")
Format

A data frame with 399 observations on the following 11 variables.

- id a numeric vector the id of a subject
- couple a numeric vector for the id of a couple
- dob a Date for the date of birth
- doe a Date for the date of entry into study
- dox a Date for the date of exit from study
- disp a Date for the date of bereavement
- fail a numeric vector for status at study exit 0=alive 1=died
- group a numeric vector for Group
- disable a numeric vector for disability level
- health a numeric vector for perceived health status
- sex a numeric vector for sex 1=M 2=F

Examples

data(brv)
## maybe str(brv); plot(brv) ...

```
colon Colon cancer dataset
```

Description

Colon cancer dataset

Usage

data("colon")

Format

A data frame with 15564 observations on the following 18 variables.

- sex a factor with levels Male Female
- age a numeric vector
- stage a factor with levels Unknown Localised Regional Distant
- mmdx a numeric vector
- yydx a numeric vector
- surv_mm a numeric vector
- surv_yy a numeric vector
- status a factor with levels Alive Dead: cancer Dead: other Lost to follow-up
colon_sample

subsite a factor with levels Coecum and ascending Transverse Descending and sigmoid Other and NOS
year8594 a factor with levels Diagnosed 75–84 Diagnosed 85–94
agegrp a factor with levels 0–44 45–59 60–74 75+
dx a Date
exit a Date
id a numeric vector
ydx a numeric vector for continuous year of diagnosis
yexit a numeric vector for continuous year of exit
bdate a Date vector for year of birth
ybdate a numeric vector for continuous year of birth

Examples

data(colon)
## maybe str(colon) ; plot(colon) ...

---

Data frame with 35 observations on the following 9 variables.

sex a factor with levels Male Female
age a numeric vector
stage a factor with levels Unknown Localised Regional Distant
mmdx a numeric vector
yydx a numeric vector
surv_mm a numeric vector
surv_yy a numeric vector
status a factor with levels Alive Dead: cancer Dead: other Lost to follow-up
subsite a factor with levels Coecum and ascending Transverse Descending and sigmoid Other and NOS

Examples

data(colon_sample)
## maybe str(colon_sample) ; plot(colon_sample) ...
Smoothed hazard estimates for \texttt{coxph}

**Description**

Smoothed hazard estimates for \texttt{coxph}

**Usage**

\begin{verbatim}
coxphHaz(object, newdata, n.grid = 300, kernel = "epanechnikov", from, to, ...)
## S3 method for class 'coxphHaz'
print(x, digits=NULL, ...)
## S3 method for class 'coxphHaz'
plot(x, xlab="Time", ylab="Hazard", type="l", ...)
## S3 method for class 'coxphHazList'
plot(x, xlab="Time", ylab="Hazard", type="l",
     col=1:length(x), lty=1, legend.args=list(), ...)
## S3 method for class 'coxphHazList'
lines(x, ...)
\end{verbatim}

**Arguments**

- \texttt{object} \texttt{coxph} object
- \texttt{newdata} data-frame with covariates for prediction
- \texttt{n.grid} the number of grid values for which the hazard is calculated
- \texttt{kernel} the kernel used for smoothing
- \texttt{from} argument for density. Defaults to the minimum time.
- \texttt{to} argument for density. Defaults to the maximum time.
- \texttt{x} object
- \texttt{digits} argument passed to \texttt{print.density}
- \texttt{col} graphics argument
- \texttt{lty} graphics argument
- \texttt{xlab} graphics argument
- \texttt{ylab} graphics argument
- \texttt{type} graphics argument
- \texttt{legend.args} a list of options that are passed to the legend call. Defaults are \texttt{list(x="topright",legend=strata(attr(x,"newdata")),col=col,lty=lty)}.
- ... other arguments. For \texttt{coxphHaz}, these arguments are passed to \texttt{density}. For the \texttt{plot} and \texttt{lines} methods, these are passed to the relevant \texttt{plot, matplot} and \texttt{matlines} functions.

**Details**

Smooth hazard estimates from a Cox model using kernel smoothing of the Nelson-Aalen estimator.
The `coxphHaz` function returns either a class of type `c("coxphHaz","density")` when `newdata` has one row or, for multiple rows in `newdata`, a class of type `"coxphHazList"`, which is a list of type `c("coxphHaz","density")`.

**See Also**

`coxph`, `survfit`, `density`

**Examples**

```r
fit <- coxph(Surv(surv_mm/12,status="Dead: cancer")~agegrp, data=colon)
newdata <- data.frame(agegrp=levels(colon$agegrp))
haz <- suppressWarnings(coxphHaz(fit,newdata))
plot(haz, xlab="Time since diagnosis (years)")
legend("topright", legend=newdata$agegrp, lty=1:4, col=1:4, bty="n")
```

---

**diet**  
*Diet data set*

**Description**

Diet data set

**Usage**

`data("diet")`

**Format**

A data frame with 337 observations on the following 15 variables.

- `id` a numeric vector
- `chd` a numeric vector
- `y` a numeric vector
- `hieng` a factor with levels low high
- `energy` a numeric vector
- `job` a factor with levels driver conductor bank
- `month` a numeric vector
- `height` a numeric vector
- `weight` a numeric vector
- `doe` a Date for date of study entry
- `dox` a Date for date of study exit
- `dob` a Date for date of birth
- `yob` a numeric vector for continuous year of birth
- `yoe` a numeric vector for continuous year of entry
- `yox` a numeric vector for continuous year of exit
**eform**

Calculate the exponential form for coefficients and their confidence intervals using either profile likelihood-based or Wald-based confidence intervals.

**Description**

`irr` and `or` use `eform` with a different name for the estimator.

**Usage**

```r
eform(object, ...)  
## Default S3 method:  
eform(object, parm, level = 0.95, method = c("Delta","Profile"), name = "exp(beta)", ...)  
irr(..., name = "IRR")  
or(..., name = "OR")
```

**Arguments**

- `object` A fitted model object with `coef` and `confint` methods
- `parm` a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
- `level` the confidence level required
- `method` string to determine method to use the delta method (`stats::confint.default`), which assumes that the parameters are asymptotically normal, or profile likelihood-based confidence intervals (`MASS:::confint.gllm`), respectively.
- `name` name of the estimator.
- `...` arguments to pass from `irr` or `or` to `eform`.

**Value**

A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2 in

**Examples**

```r
## from example(glm)
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3, 1, 9); treatment <- gl(3, 3)
glm.D93 <- glm(counts ~ outcome + treatment, family = poisson())
eform(glm.D93)
eform(glm.D93, method="Profile")
```
Create cohort life table

Description
Create cohort life table.

Usage
lifetab(tis, ninit, nlost, nevent)

Arguments
tis a vector of end points of time intervals, whose length is 1 greater than nlost and nevent.
ninit the number of subjects initially entering the study.
nlost a vector of the number of individuals lost follow or withdrawn alive for whatever reason.
nevent a vector of the number of individuals who experienced the event

Details
This is a minor update of the lifetab function from the KMsurv package, where the start and stop times of the intervals are now included in the return value.

Value
A data.frame with the following columns:
tstart interval start time.
tstop interval end time.
nsubs the number of subject entering the intervals who have not experienced the event.
nlost the number of individuals lost follow or withdrawn alive for whatever reason.
nrisk the estimated number of individuals at risk of experiencing the event.
nevent the number of individuals who experienced the event.
surv the estimated survival function at the start of the intervals.
pdf the estimated probability density function at the midpoint of the intervals.
hazard the estimated hazard rate at the midpoint of the intervals.
se.surv the estimated standard deviation of survival at the beginning of the intervals.
se.pdf the estimated standard deviation of the probability density function at the midpoint of the intervals.
se.hazard the estimated standard deviation of the hazard function at the midpoint of the intervals.

The row.names are the intervals.
Author(s)

Jun Yan <jyan@stat.uconn.edu>

Examples

tis <- c(0, 2, 3, 5, 7, 11, 17, 25, 37, 53, NA)
nsubs <- c(927, 848, 774, 649, 565, 449, 296, 186, 112, 27)
nlost <- c(2, 3, 6, 9, 7, 5, 3, rep(0, 3))
nevent <- c(77, 71, 119, 75, 109, 148, 107, 74, 85, 27)

lifetab(tis, nsubs[1], nlost, nevent)

Description

Calculate a life table using the actuarial method using a formula and a data-frame with optional breaks.

Usage

lifetab2(formula, data, subset, breaks = NULL)

## S3 method for class 'lifetab2'
plot(x, y=NULL, ...)
## S3 method for class 'lifetab2'
lines(x, y=NULL, ...)

Arguments

formula  formula with the left-hand side being a Surv object, including a time and event indicator, and the right-hand side indicated stratification.

data  optional data.frame for the Surv object. If this is not provided, then the parent frame is used for the Surv object.

subset  optional subset statement

breaks  optional numeric vector of breaks. If this is not provided, then the unique time values from the Surv object are used together with Inf.

x  lifetab2 object

y  unused argument (part of the generic function)

...  other arguments

Details

See lifetab for details. This wrapper is meant to make life easier.

A copy of the lifetab function has been included in the biostat3 package to reduce dependencies.
**Value**

A data.frame as per `lifetab`.

**Author(s)**

Mark Clements for the wrapper.

**Examples**

```r
## we can use unique transformed times (colon_sample)
lifetab2(Surv(floor(surv_yy),status="Dead: cancer")~1, colon_sample)

## we can also use the breaks argument (colon)
lifetab2(Surv(surv_yy,status="Dead: cancer")~1, colon, breaks=0:10)
```

**Description**

Using results calculated by the `linearHypothesis` function in the `car` package, calculate a linear combination of regression parameters.

**Usage**

```
lincom(model, specification, level = 0.95, eform = FALSE, ...)
```

**Arguments**

- `model` regression model object (as per the `model` argument in `linearHypothesis`)
- `specification` specification of the linear combination. This is the same as a single component of the `hypothesis.matrix` argument in `linearHypothesis`.
- `level` the confidence level required
- `eform` logical for whether to exponentiate the confidence interval (default=FALSE)
- `...` other arguments to the `linearHypothesis` function.

**Details**

Multiple specifications of linear combinations are called individually.

**Value**

A matrix with columns including the estimate, a normal-based confidence interval, test statistic and p-values.

**See Also**

See Also `linearHypothesis`. 
Examples

```
fit <- glm(chd ~ hieng*job + offset(log(y)), data=diet, family=poisson)
lincom(fit, c("hienghigh+hienghigh:jobconductor",
            "hienghigh+hienghigh:jobbank"),
      eform=TRUE)
```

```
melanoma
Description

Melanoma cancer dataset

Usage

data("melanoma")

Format

A data frame with 7775 observations on the following 18 variables.

- `sex`: a factor with levels Male Female
- `age`: a numeric vector
- `stage`: a factor with levels Unknown Localised Regional Distant
- `mmdl`: a numeric vector
- `yydx`: a numeric vector
- `surv_mm`: a numeric vector
- `surv_yy`: a numeric vector
- `status`: a factor with levels Alive Dead: cancer Dead: other Lost to follow-up
- `subsite`: a factor with levels Head and Neck Trunk Limbs Multiple and NOS
- `year8594`: a factor with levels Diagnosed 75-84 Diagnosed 85-94
- `dx`: a Date
- `exit`: a Date
- `agegrp`: a factor with levels 0-44 45-59 60-74 75+
- `id`: a numeric vector
- `ydx`: a numeric vector for continuous year of diagnosis
- `yexit`: a numeric vector for continuous year of exit
- `bdate`: a Date vector for date of birth
- `ybdate`: a numeric vector for continuous year of birth

Examples

data(melanoma)
```
## maybe str(melanoma) ; plot(melanoma) ...
Formula wrapper for the `muhaz` function from the `muhaz` package.

Description

Formula wrapper for the `muhaz` function from the `muhaz` package.

Usage

```r
muhaz2(formula, data, subset, max.time, ...)  
## S3 method for class 'muhaz2'
plot(x, haz.scale=1, ylab="Hazard", ylim=NULL, log="", ...)  
## S3 method for class 'muhazList'
plot(x, lty=1:5, col=1:length(x), log="", legend.args=list(), ...)  
## S3 method for class 'muhaz2'
lines(x, ..., haz.scale = 1)  
## S3 method for class 'muhazList'
lines(x, lty=1, col=1:length(x), ...)  
## S3 method for class 'muhazList'
summary(object, ...)  
## S3 method for class 'muhazList'
as.data.frame(x, row.names, optional, ...)  
## S3 method for class 'muhaz'
as.data.frame(x, row.names, optional, ...)
```

Arguments

- `formula`: formula with the left-hand side being a `Surv` object, including a time and event indicator, and the right-hand side indicated stratification.
- `data`: optional data.frame for the `Surv` object. If this is not provided, then the parent frame is used for the `Surv` object.
- `subset`: subset predicate for the dataset
- `max.time`: maximum follow-up time for the hazards
- `ylab`: graphics argument for ylab (y-axis label)
- `lty`: graphics argument for line type
- `col`: graphics argument for line colour
- `legend.args`: a list of options that are passed to the legend call. Defaults are `list(x="topright",legend=names(x),col=col,lty=lty)`.
- `haz.scale`: scale for the hazard in the plot
- `row.names`: not currently used
- `object`: `muhazList` object
- `ylim`: graphics argument for the limits of the y axis
- `log`: graphics argument for a log transformation of the x or y axes
- `x`: `muhazList` or `muhaz` object
- `optional`: not currently used
- `...`: other arguments
Value

For a single strata, this is a muhaz object. For multiple strata, this is a muhazList object, which includes methods for

Examples

plot(muhaz2(Surv(surv_mm, status == "Dead: cancer")~1, melanoma))

poisson.ci

Exact Poisson confidence intervals.

Description

A wrapper for the poisson.test that allows for vector values.

Usage

poisson.ci(x, T = 1, conf.level = 0.95)

Arguments

x number of events.
T time base for event count.
conf.level confidence level for the returned confidence interval.

Details

This uses stats::poisson.test for the calculations.

Value

A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2 in % (by default 2.5% and 97.5%).

See Also

poisson.test

Examples

### These are paraphrased from data sets in the ISwR package

## SMR, Welsh Nickel workers
poisson.ci(137, 24.19893)

## eba1977, compare Fredericia to other three cities for ages 55-59
poisson.ci(c(11, 6+8+7), c(800, 1083+1050+878))
Description

popmort dataset, with population-based mortality rates

Usage

data("popmort")

Format

A data frame with 10600 observations on the following 5 variables.

- sex a numeric vector
- _year a numeric vector
- _age a numeric vector
- prob a numeric vector
- rate a numeric vector

Examples

data(pompert)
## maybe str(pompert); plot(pompert) ...

smoothHazard

Simple implementation for kernel density smoothing of the Nelson-Aalen estimator.

Description

Simple implementation for kernel density smoothing of the Nelson-Aalen estimator. Prefer muhaz for right censored data and bshazard for left truncated and right censored data.

Usage

smoothHazard(object, n.grid = 300, kernel = "epanechnikov",
              from = NULL, to = NULL, min.n.risk = 1, ...)
## S3 method for class 'smoothHazard'
plot(x, xlab = "Time", ylab = "Hazard", type = "l", ...)
survPHplot

Plot to assess non-proportionality

Description

Plot of log(time) versus -log(-log(survival)) to assess non-proportionality. A constant distance between curves suggest proportionality.

Usage

```
survPHplot(formula, data, subset, contrasts, weights, col = 1:5, lty = 1:5, pch = 19, xlab = "Time (log scale)", ylab = "-log(-log(Survival))", log = "x", legend.args = list(), ...)```

Arguments

- **formula**: either (i) formula with a Surv object on the left-hand-side and stratification covariates on the right-hand-side, or (ii) a survfit object
- **data**: data argument passed to survfit
- **subset**: subset argument passed to survfit
- **contrasts**: contrasts argument passed to survfit
- **weights**: weights argument passed to survfit
- **col**: colours of the curves passed to lines
- **lty**: line type of the curves passed to lines
- **pch**: pch for the curves passed to points
- **xlab**: xlab graphics argument passed to plot.default
- **ylab**: ylab graphics argument passed to plot.default
**survRate**

log graphics argument passed to `plot.default`  

legend.args list of arguments passed to `legend`. These arguments update the base arguments, which are `list(x="topright", legend=names(survfit$strata), col=col, lty=lty, pch=pch)`  

... Other arguments passed to `plot.default`

**Details**

The default plot is to use straight lines between the transformed survival values for each strata, rather than using steps.

**Value**

Primary purpose is for plotting (side effect). The return value is initial plot.

**Examples**

```r
survPHplot(Surv(surv_mm/12, status == "Dead: cancer") ~ year8594, 
data=colon, subset=(stage="Localised"), 
legend.args=list(bty="n")
```

---

**survRate**

`Describe rates`

**Description**

Describe rates using the `Surv` function.

**Usage**

`survRate(formula, data, subset, addvars = TRUE, ci=TRUE, ...)`

**Arguments**

- **formula**  
  formula with the left-hand-side being a `Surv` function and the right-hand-side being any stratification variables.

- **data**  
  source dataset

- **subset**  
  subset conditions for the source dataset

- **addvars**  
  logical for whether to add the stratification variables to the output (default=TRUE). This is useful for subsequent analysis.

- **ci**  
  logical for whether to calculate the confidence interval (default=TRUE).

- **...**  
  other arguments to the `poisson.test` function for calculation of the confidence intervals.
Value

data-frame with columns tstop, event, rate, lower and upper. Covariates are appended if addvar=TRUE.
Confidence intervals use stats::poisson.test.

Examples

```r
## incidence rates for CHD for low- or high-energy diets
survRate(Surv(y, chd) ~ hieng, data=diet)
```

utility utilities

Utility functions for the biostat3 package

Description

Utility functions for the biostat3 package.

Usage

```r
updateList(object, ...)
format_perc(probs, digits)
```

Arguments

- **object**: base object (list)
- **...**: arguments to update
- **probs**: probability to express as a percentage
- **digits**: number of significant digits

Details

Update the names in the base object list that are specified in the arguments to update.

Value

list

Examples

```r
updateList(list(a=1, b=2), a=10, c=30)
```
Description

Convert a Date vector to a numeric vector (either continuous or truncated).

Usage

```
year(date, trunc = FALSE, year.length = 365.24)
```

Arguments

- `date`: Date vector
- `trunc`: logical for whether to truncate the date to a whole year or consider the date as a double (default).
- `year.length`: assumed length of a year

Details

For the double calculation, we use (truncated year of Date) + (date - 1 Jan of Year)/year.length.

Value

numeric vector

Examples

```
c(year(as.Date("2001-07-01")), year(as.Date("2001-01-01"), trunc=TRUE))
```
Index

+Topic **datasets**
  - brv, 3
  - colon, 4
  - colon_sample, 5
  - diet, 7
  - melanoma, 12
  - popmort, 15

+Topic **manip**
  - lifetab, 9

+Topic **package**
  - biostat3-package, 2

+Topic **survival**
  - biostat3-package, 2
  - coxphHaz, 6
  - lifetab2, 10
  - muhaz2, 13
  - smoothHaz, 15
  - survPHplot, 16
  - survRate, 17

addIndicators, 3
as.data.frame.muhaz(muhaz2), 13
as.data.frame.muhazList(muhaz2), 13

biostat3(biostat3-package), 2
biostat3-package, 2
brv, 3

colon, 4, 5
colon_sample, 5
coxph, 7
coxphHaz, 6

density, 7
diet, 7
eform, 8

format_perc(utilities), 18
irr(eform), 8

lifetab, 9, 10, 11
lifetab2, 10
lincom, 11
linearHypothesis, 11
lines.coxphHazList(coxphHaz), 6
lines.lifetab2(lifetab2), 10
lines.muhaz2(muhaz2), 13
lines.muhazList(muhaz2), 13

melanoma, 12
muhaz, 13, 14
muhaz2, 13

or(eform), 8
plot.coxphHaz(coxphHaz), 6
plot.coxphHazList(coxphHaz), 6
plot.lifetab2(lifetab2), 10
plot.muhaz2(muhaz2), 13
plot.muhazList(muhaz2), 13
plot.smoothHaz(smoothHaz), 15
poisson.ci, 14
poisson.test, 14, 17
popmort, 15
print.coxphHaz(coxphHaz), 6
smoothHaz, 15
summary.muhazList(muhaz2), 13
Surv, 17
survfit, 7
survPHplot, 16
survRate, 17

updateList(utilities), 18
utilities, 18

year, 19