

# Package: biostat3 (via r-universe)

September 12, 2024

**Type** Package

**Title** Utility Functions, Datasets and Extended Examples for Survival Analysis

**Version** 0.2.1

**Date** 2024-08-13

**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 (>= 3.5), MASS, methods

**Imports** graphics, stats

**Suggests** car, bshazard, rstpm2, Epi, dplyr, ggplot2, muhaz

**License** GPL (>=2)

**LazyData** yes

**LazyLoad** yes

**Repository** <https://mclements.r-universe.dev>

**RemoteUrl** <https://github.com/mclements/biostat3>

**RemoteRef** HEAD

**RemoteSha** a8220a55a68b007f411a6a51b789e37a8b94901a

## Contents

|                  |   |
|------------------|---|
| biostat3-package | 2 |
| addIndicators    | 3 |
| brv              | 3 |
| colon            | 4 |
| colon_sample     | 5 |
| coxphHaz         | 6 |
| diet             | 7 |

|                      |           |
|----------------------|-----------|
| eform . . . . .      | 8         |
| lifetab . . . . .    | 9         |
| lifetab2 . . . . .   | 11        |
| lincom . . . . .     | 12        |
| melanoma . . . . .   | 13        |
| muhaz2 . . . . .     | 14        |
| poisson.ci . . . . . | 15        |
| popmort . . . . .    | 16        |
| smoothHaz . . . . .  | 16        |
| survPHplot . . . . . | 17        |
| survRate . . . . .   | 18        |
| utilities . . . . .  | 19        |
| year . . . . .       | 20        |
| <b>Index</b>         | <b>21</b> |

---

|                  |                                                                                |
|------------------|--------------------------------------------------------------------------------|
| biostat3-package | <i>Utility Functions, Datasets and Extended Examples for Survival Analysis</i> |
|------------------|--------------------------------------------------------------------------------|

---

## 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".

## 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

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

---

|               |                                                                        |
|---------------|------------------------------------------------------------------------|
| addIndicators | <i>Utility to add indicators from a data-frame based on a formula.</i> |
|---------------|------------------------------------------------------------------------|

---

**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 | <i>Bereavement dataset</i> |
|-----|----------------------------|

---

**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  
 dosp 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  
 disab 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

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

### Examples

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

---

|              |                                                                         |
|--------------|-------------------------------------------------------------------------|
| colon_sample | <i>Sample from the <a href="#">colon</a> dataset used for teaching.</i> |
|--------------|-------------------------------------------------------------------------|

---

### Description

Sample from the [colon](#) dataset used for teaching.

### Usage

```
data("colon_sample")
```

### Format

A 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) ...
```

---

 coxphHaz

*Smoothed hazard estimates for coxph*


---

### Description

Smoothed hazard estimates for coxph

### Usage

```
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, ...)
## S3 method for class 'coxphHaz'
as.data.frame(x, row.names=NULL, optional=FALSE, level=0.95, ...)
## S3 method for class 'coxphHazList'
as.data.frame(x, row.names=NULL, optional=FALSE, ...)
```

### Arguments

|           |                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------|
| object    | coxph object                                                                                        |
| newdata   | data-frame with covariates for prediction                                                           |
| n.grid    | the number of grid values for which the hazard is calculated                                        |
| kernel    | the kernel used for smoothing                                                                       |
| from      | argument for density. Defaults to the minimum time.                                                 |
| to        | argument for density. Defaults to the maximum time.                                                 |
| x         | object                                                                                              |
| digits    | argument passed to print.density                                                                    |
| col       | graphics argument                                                                                   |
| lty       | graphics argument                                                                                   |
| xlab      | graphics argument                                                                                   |
| ylab      | graphics argument                                                                                   |
| type      | graphics argument                                                                                   |
| level     | level for confidence intervals (default=0.95)                                                       |
| row.names | NULL or a character vector giving the row names for the data frame. Missing values are not allowed. |

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| optional    | logical. If TRUE, setting row names and converting column names (to syntactic names: see <code>make.names</code> ) is optional. Note that all of R's base package <code>as.data.frame()</code> methods use <code>optional</code> only for column names treatment, basically with the meaning of <code>data.frame(*, check.names = !optional)</code> . See also the <code>make.names</code> argument of the <code>matrix</code> method. |
| legend.args | a list of options that are passed to the legend call. Defaults are <code>list(x="topright", legend=strata(atts</code>                                                                                                                                                                                                                                                                                                                  |
| ...         | other arguments. For <code>coxphHaz</code> , these arguments are passed to <code>density</code> . For the <code>plot</code> and <code>lines</code> methods, these are passed to the relevant <code>plot</code> , <code>matplot</code> and <code>matlines</code> functions.                                                                                                                                                             |

## Details

Smooth hazard estimates from a Cox model using kernel smoothing of the Nelson-Aalen estimator.

## Value

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

```
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)")
```

---

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

**Examples**

```
data(diet)
## maybe str(diet) ; plot(diet) ...
```

---

|       |                                                                                                                                                                 |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eform | <i>Calculate the exponential form for coefficients and their confidence intervals using either profile likelihood-based or Wald-based confidence intervals.</i> |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**Description**

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

**Usage**

```
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.glm), 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  $(1-\text{level})/2$  and  $1 - (1-\text{level})/2$  in

**Examples**

```
## 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")
```

---

|         |                                 |
|---------|---------------------------------|
| lifetab | <i>Create cohort life table</i> |
|---------|---------------------------------|

---

**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:

|                        |                                                                                                        |
|------------------------|--------------------------------------------------------------------------------------------------------|
| <code>tstart</code>    | interval start time.                                                                                   |
| <code>tstop</code>     | interval end time.                                                                                     |
| <code>nsubs</code>     | the number of subject entering the intervals who have not experienced the event.                       |
| <code>nlost</code>     | the number of individuals lost follow or withdrawn alive for whatever reason.                          |
| <code>nrisk</code>     | the estimated number of individuals at risk of experiencing the event.                                 |
| <code>nevent</code>    | the number of individuals who experienced the event.                                                   |
| <code>surv</code>      | the estimated survival function at the start of the intervals.                                         |
| <code>pdf</code>       | the estimated probability density function at the midpoint of the intervals.                           |
| <code>hazard</code>    | the estimated hazard rate at the midpoint of the intervals.                                            |
| <code>se.surv</code>   | the estimated standard deviation of survival at the beginning of the intervals.                        |
| <code>se.pdf</code>    | the estimated standard deviation of the probability density function at the midpoint of the intervals. |
| <code>se.hazard</code> | 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)
```

---

`lifetab2`*Formula wrapper for `lifetab` from the `KMsurv` package.*

---

### 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

|                      |                                                                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>formula</code> | formula with the left-hand side being a <code>Surv</code> object, including a time and event indicator, and the right-hand side indicated stratification.           |
| <code>data</code>    | optional <code>data.frame</code> for the <code>Surv</code> object. If this is not provided, then the parent frame is used for the <code>Surv</code> object.         |
| <code>subset</code>  | optional subset statement                                                                                                                                           |
| <code>breaks</code>  | optional numeric vector of breaks. If this is not provided, then the unique time values from the <code>Surv</code> object are used together with <code>Inf</code> . |
| <code>x</code>       | <code>lifetab2</code> object                                                                                                                                        |
| <code>y</code>       | unused argument (part of the generic function)                                                                                                                      |
| <code>...</code>     | 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**

```
## 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)
```

---

lincom

*Linear combination of regression parameters.*


---

**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**

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

*Melanoma cancer dataset*

---

## 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

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 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

## Examples

```
data(melanoma)
## maybe str(melanoma) ; plot(melanoma) ...
```

muhaz2

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

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

**Usage**

```

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**

|                          |                                                                                                                                                             |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>formula</code>     | formula with the left-hand side being a <code>Surv</code> object, including a time and event indicator, and the right-hand side indicated stratification.   |
| <code>data</code>        | optional <code>data.frame</code> for the <code>Surv</code> object. If this is not provided, then the parent frame is used for the <code>Surv</code> object. |
| <code>subset</code>      | subset predicate for the dataset                                                                                                                            |
| <code>max.time</code>    | maximum follow-up time for the hazards                                                                                                                      |
| <code>ylab</code>        | graphics argument for <code>ylab</code> (y-axis label)                                                                                                      |
| <code>lty</code>         | graphics argument for line type                                                                                                                             |
| <code>col</code>         | graphics argument for line colour                                                                                                                           |
| <code>legend.args</code> | a list of options that are passed to the legend call. Defaults are <code>list(x="topright", legend=names(x), c</code>                                       |
| <code>haz.scale</code>   | scale for the hazard in the plot                                                                                                                            |
| <code>row.names</code>   | not currently used                                                                                                                                          |
| <code>object</code>      | <code>muhazList</code> object                                                                                                                               |
| <code>ylim</code>        | graphics argument for the limits of the y axis                                                                                                              |
| <code>log</code>         | graphics argument for a log transformation of the x or y axes                                                                                               |
| <code>x</code>           | <code>muhazList</code> or <code>muhaz</code> object                                                                                                         |
| <code>optional</code>    | not currently used                                                                                                                                          |
| <code>...</code>         | 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  $(1-\text{level})/2$  and  $1 - (1-\text{level})/2$  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))
```

---

|         |                                                               |
|---------|---------------------------------------------------------------|
| popmort | <i>popmort dataset, with population-based mortality rates</i> |
|---------|---------------------------------------------------------------|

---

**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(popmort)
## maybe str(popmort) ; plot(popmort) ...
```

---

|           |                                                                                          |
|-----------|------------------------------------------------------------------------------------------|
| smoothHaz | <i>Simple implementation for kernel density smoothing of the Nelson-Aalen estimator.</i> |
|-----------|------------------------------------------------------------------------------------------|

---

**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**

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

**Arguments**

|            |                                          |
|------------|------------------------------------------|
| object     | survfit object                           |
| n.grid     | number of grid points; passed to density |
| kernel     | kernel used; passed to density           |
| from       | left boundary; passed to density         |
| to         | right boundary; passed to density        |
| min.n.risk | minimum number at risk                   |
| x          | object of class smoothHaz                |
| xlab       | graphics argument                        |
| ylab       | graphics argument                        |
| type       | graphics argument                        |
| ...        | Other arguments                          |

---

 survPHplot

*Plot to assess non-proportionality*


---

**Description**

Plot of  $\log(\text{time})$  versus  $-\log(-\log(\text{survival}))$  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 co-variates 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                                                                                               |

|             |                                                                                                                                                                       |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| log         | 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

```
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 <a href="#">Surv</a> 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 <a href="#">poisson.test</a> 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**

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

---

utilities

*Utility functions for the biostat3 package*

---

**Description**

Utility functions for the biostat3 package.

**Usage**

```
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**

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

---

`year`*Convert a Date vector to a numeric vector*

---

**Description**

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

**Usage**

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

**Arguments**

|                          |                                                                                                      |
|--------------------------|------------------------------------------------------------------------------------------------------|
| <code>date</code>        | Date vector                                                                                          |
| <code>trunc</code>       | logical for whether to truncate the date to a whole year or consider the date as a double (default). |
| <code>year.length</code> | 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

- \* **datasets**
  - brv, 3
  - colon, 4
  - colon\_sample, 5
  - diet, 7
  - melanoma, 13
  - popmort, 16
- \* **manip**
  - lifetab, 9
- \* **package**
  - biostat3-package, 2
- \* **survival**
  - biostat3-package, 2
  - coxphHaz, 6
  - lifetab2, 11
  - muhaz2, 14
  - smoothHaz, 16
  - survPHplot, 17
  - survRate, 18
- addIndicators, 3
- as.data.frame.coxphHaz (coxphHaz), 6
- as.data.frame.coxphHazList (coxphHaz), 6
- as.data.frame.muhaz (muhaz2), 14
- as.data.frame.muhazList (muhaz2), 14
- 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), 19
- irr (eform), 8
- lifetab, 9, 11
- lifetab2, 11
- lincom, 12
- lines.coxphHazList (coxphHaz), 6
- lines.lifetab2 (lifetab2), 11
- lines.muhaz2 (muhaz2), 14
- lines.muhazList (muhaz2), 14
- melanoma, 13
- muhaz, 14, 15
- muhaz2, 14
- or (eform), 8
- plot.coxphHaz (coxphHaz), 6
- plot.coxphHazList (coxphHaz), 6
- plot.lifetab2 (lifetab2), 11
- plot.muhaz2 (muhaz2), 14
- plot.muhazList (muhaz2), 14
- plot.smoothHaz (smoothHaz), 16
- poisson.ci, 15
- poisson.test, 15, 18
- popmort, 16
- print.coxphHaz (coxphHaz), 6
- smoothHaz, 16
- summary.muhazList (muhaz2), 14
- Surv, 18
- survfit, 7
- survPHplot, 17
- survRate, 18
- updateList (utilities), 19
- utilities, 19
- year, 20