Package ‘success’

April 15, 2022

Title Survival Control Charts Estimation Software

Version 0.1.0

Maintainer Daniel Gomon <d.gomon@math.leidenuniv.nl>

Description Quality control charts for survival outcomes.
Allows users to construct the Continuous Time Generalized
Rapid Response CUSUM (CGR-CUSUM), the Biswas & Kalbfleisch (2008)
<doi:10.1002/sim.3216> CUSUM, the Bernoulli CUSUM and the
risk-adjusted funnel plot for survival data. These procedures can be
used to monitor survival processes for a change in the failure rate.

License GPL (>= 3)

URL https://github.com/d-gomon/success

Depends ggplot2, pbapply, plotly, RColorBrewer, R (>= 3.5.0)

Imports survival

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

NeedsCompilation no

Author Daniel Gomon [aut, cre] (<https://orcid.org/0000-0001-9011-3743>),
Mirko Signorelli [ctb] (<https://orcid.org/0000-0001-9011-3743>)

Repository CRAN

Date/Publication 2022-04-15 10:40:02 UTC

R topics documented:

bernoulli_control_limit

Determine control limits for the Bernoulli CUSUM by simulation

Description

This function can be used to determine control limits for the Bernoulli CUSUM (bernoulli_cusum) procedure by restricting the type I error alpha of the procedure over time.

Usage

```
bernoulli_control_limit(time, alpha = 0.05, followup, psi, n_sim = 200,
  glmmod, baseline_data, theta, p0, p1, h_precision = 0.01, seed = 1041996,
  pb = FALSE, assist)
```

Arguments

time	A numeric value over which the type I error alpha must be restricted.
alpha	A proportion between 0 and 1 indicating the required maximal type I error.
followup	The value of the follow-up time to be used to determine event time. Event time will be equal to entrytime + followup for each subject.
psi	A numeric value indicating the estimated Poisson arrival rate of subjects at their respective units. Can be determined using parameter_assist().
n_sim	An integer value indicating the amount of units to generate for the determination of the control limit. Larger values yield more precise control limits, but increase computation times. Default is 200.

`glmmod` Generalized linear regression model used for risk-adjustment as produced by the function `glm()`. Suggested:
`glm(as.formula("(survtime <= followup) & (censorid == 1) ~ covariates"),data = data)`.
Alternatively, a list containing the following elements:

`formula`**:** a `formula()` in the form `~ covariates`;

`coefficients`**:** a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in `formula` and colnames of `data`.

`baseline_data` (optional): A `data.frame` used for covariate resampling with rows representing subjects and at least the following named columns:

`entrytime`**:** time of entry into study (numeric);

`survtime`**:** time from entry until event (numeric);

`censorid`**:** censoring indicator (0 = right censored, 1 = observed), (integer).

and optionally additional covariates used for risk-adjustment. Can only be specified in combination with `coxphmod`.

`theta` The $theta$ value used to specify the odds ratio e^θ under the alternative hypothesis. Note that

$$p_1 = \frac{p_0 e^\theta}{(1-p_0)(1+p_0 e^\theta)}.$$

`p0` The baseline failure probability at `entrytime + followup` for individuals.

`p1` The alternative hypothesis failure probability at `entrytime + followup` for individuals.

`h_precision` (optional): A numerical value indicating how precisely the control limit should be determined. By default, control limits will be determined up to 2 significant digits.

`seed` (optional): A numeric seed for survival time generation. Default is 01041996 (my birthday).

`pb` (optional): A boolean indicating whether a progress bar should be shown. Default is `FALSE`.

`assist` (optional): Output of the function `parameter_assist()`

Details

This function performs 3 steps to determine a suitable control limit.

- Step 1: Generates `n_sim` in-control units (failure rate as baseline). If `data` is provided, subject covariates are resampled from the data set.
- Step 2: Determines chart values for all simulated units.
- Step 3: Determines control limits such that at most a proportion `alpha` of all units cross the control limit.

The generated data as well as the charts are also returned in the output.

Value

A list containing three components:

- `call`: the call used to obtain output;
- `charts`: A list of length `n_sim` containing the constructed charts;
- `data`: A `data.frame` containing the in-control generated data.
- `h`: Determined value of the control limit.

Author(s)

Daniel Gomon

See Also

`bernoulli_cusum`

Other control limit simulation: `bk_control_limit()`, `cgr_control_limit()`

Examples

```
#We consider patient outcomes 100 days after their entry into the study.
followup <- 100

#Determine a risk-adjustment model using a generalized linear model.
#Outcome (failure within 100 days) is regressed on the available covariates:
exprfitber <- as.formula("(survtime <= followup) & (censorid == 1)~ age + sex + BMI")
glmmodber <- glm(exprfitber, data = surgerydat, family = binomial(link = "logit"))

#Determine control limit restricting type I error to 0.1 over 500 days
#using the risk-adjusted glm constructed on the baseline data.
a <- bernoulli_control_limit(time = 500, alpha = 0.1, followup = followup,
 psi = 0.5, n_sim = 10, theta = log(2), glmmod = glmmodber, baseline_data = surgerydat)

print(a$h)
```

`bernoulli_cusum` *Risk-adjusted Bernoulli CUSUM*

Description

This function can be used to construct a risk-adjusted Bernoulli CUSUM chart for survival data. It requires the specification of one of the following combinations of parameters as arguments to the function:

- `glmmod` & `theta`
- `p0` & `theta`
- `p0` & `p1`

Usage

```
bernoulli_cusum(data, followup, glmmod, theta, p0, p1, h, stoptime, assist)
```

Arguments

data
: A `data.frame` containing the following named columns for each subject:

  `entrytime`**:** time of entry into study (numeric);

  `survtime`**:** time from entry until event (numeric);

  `censorid`**:** censoring indicator (0 = right censored, 1 = observed) (integer);

  and optionally additional covariates used for risk-adjustment.

followup
: The value of the follow-up time to be used to determine event time. Event time will be equal to `entrytime + followup` for each subject.

glmmod
: Generalized linear regression model used for risk-adjustment as produced by the function `glm()`. Suggested:
  `glm(as.formula("(survtime <= followup) & (censorid == 1) ~ covariates"),data = data)`.
  Alternatively, a list containing the following elements:

  `formula`**:** a `formula()` in the form `~ covariates`;

  `coefficients`**:** a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in `formula` and colnames of `data`.

theta
: The $theta$ value used to specify the odds ratio e^θ under the alternative hypothesis. Note that

$$p_1 = \frac{p_0 e^\theta}{(1-p_0)(1+p_0 e^\theta)}.$$

p0
: The baseline failure probability at `entrytime + followup` for individuals.

p1
: The alternative hypothesis failure probability at `entrytime + followup` for individuals.

h
: (optional): Control limit to be used for the procedure.

stoptime
: (optional): Time after which the value of the chart should no longer be determined.

assist
: (optional): Output of the function `parameter_assist()`

Details

The Bernoulli CUSUM chart is given by

$$S_n = \max(0, S_{n-1} + W_n),$$

where

$$W_n = X_n \ln\left(\frac{p_1(1-p_0)}{p_0(1-p_1)}\right) + \ln\left(\frac{1-p_1}{1-p_0}\right)$$

and X_n is the outcome of the n-th (chronological) subject in the data. Instead of the standard practice of displaying patient numbering on the x-axis, the time of outcome is displayed.

Value

An object of class `bercusum` containing:

- `CUSUM`: A `data.frame` containing the following named columns:
  `time`**:** times at which chart is constructed;
  `value`**:** value of the chart at corresponding times;
  `numobs`**:** number of observations at corresponding times.
- `call`: the call used to obtain output;
- `glmmod`: coefficients of the `glm()` used for risk-adjustment, if specified;
- `stopind`: indicator for whether the chart was stopped by the control limit.

Author(s)

Daniel Gomon

See Also

`plot.bercusum`, `runlength.bercusum`

Other quality control charts: `bk_cusum()`, `cgr_cusum()`, `funnel_plot()`

Examples

```
#We consider patient outcomes 100 days after their entry into the study.
followup <- 100
#Determine a risk-adjustment model using a generalized linear model.
#Outcome (failure within 100 days) is regressed on the available covariates:
exprfitber <- as.formula(”(survtime <= followup) & (censorid == 1)~ age + sex + BMI”)
glmmodber <- glm(exprfitber, data = surgerydat, family = binomial(link = ”logit”))
#Construct the Bernoulli CUSUM on the 1st hospital in the data set.
bercus <- bernoulli_cusum(data = subset(surgerydat, unit == 1), glmmod = glmmodber,
 followup = followup, theta = log(2))
#Plot the Bernoulli CUSUM
plot(bercus)
```

`bk_control_limit` *Determine control limits for BK-CUSUM by simulation*

Description

This function can be used to determine control limits for the BK-CUSUM (`bk_cusum`) procedure by restricting the type I error `alpha` of the procedure over `time`.

Usage

```
bk_control_limit(time, alpha = 0.05, psi, n_sim = 200, theta, coxphmod,
  baseline_data, cbaseh, inv_cbaseh, interval = c(0, 9e+12),
  h_precision = 0.01, seed = 1041996, pb = FALSE, chartpb = FALSE,
  assist)
```

Arguments

`time` A numeric value over which the type I error `alpha` must be restricted.

`alpha` A proportion between 0 and 1 indicating the required maximal type I error. Default is 0.05.

`psi` A numeric value indicating the estimated Poisson arrival rate of subjects at their respective units. Can be determined using `parameter_assist()`.

`n_sim` An integer value indicating the amount of units to generate for the determination of the control limit. Larger values yield more precise control limits, but increase computation times. Default is 200.

`theta` The expected log-hazard ratio θ under the alternative hypothesis. If $\theta >= 0$, the chart will try to detect an increase in hazard ratio (upper one-sided). If $\theta < 0$, the chart will look for a decrease in hazard ratio (lower one-sided).

`coxphmod` A Cox proportional hazards regression model as produced by the function `coxph()`. Suggested:
`coxph(Surv(survtime,censorid) ~ covariates,data = data)`.
Alternatively, a list with the following elements:

- `formula`**:** a `formula()` in the form `~ covariates`;
- `coefficients`**:** a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in `formula` and colnames of `data`.

`baseline_data` (optional): A `data.frame` used for covariate resampling with rows representing subjects and at least the following named columns:

- `entrytime`**:** time of entry into study (numeric);
- `survtime`**:** time from entry until event (numeric);
- `censorid`**:** censoring indicator (0 = right censored, 1 = observed), (integer).

and optionally additional covariates used for risk-adjustment. Can only be specified in combination with `coxphmod`.

`cbaseh` A function that returns the unadjusted cumulative baseline hazard $H_0(t)$. If `cbaseh` is missing but `coxphmod` has been specified as a survival object, this baseline hazard rate will be determined using the provided `coxphmod`.

`inv_cbaseh` (optional): A function that returns the unadjusted inverse cumulative baseline hazard $H_0^{-1}(t)$. If `inv_cbaseh` is missing, it will be determined from `cbaseh` numerically.

`interval` (optional): Interval in which survival times should be solved for numerically.

`h_precision` (optional): A numerical value indicating how precisely the control limit should be determined. By default, control limits will be determined up to 2 significant digits.

`seed` (optional): A numeric seed for survival time generation. Default is 01041996 (my birthday).

`pb` (optional): A boolean indicating whether a progress bar should be shown. Default is `FALSE`.

`chartpb` (optional): A boolean indicating whether progress bars should be displayed for the constructions of the charts. Default is `FALSE`.

`assist` (optional): Output of the function `parameter_assist()`

Details

This function performs 3 steps to determine a suitable control limit.

- Step 1: Generates `n_sim` in-control units (failure rate as baseline). If `data` is provided, subject covariates are resampled from the data set.
- Step 2: Determines chart values for all simulated units.
- Step 3: Determines control limits such that at most a proportion `alpha` of all units cross the control limit.

The generated data as well as the charts are also returned in the output.

Value

A list containing three components:

- `call`: the call used to obtain output;
- `charts`: A list of length `n_sim` containing the constructed charts;
- `data`: A `data.frame` containing the in-control generated data.
- `h`: Determined value of the control limit.
- `achieved_alpha`: Achieved type I error on the sample of `n_sim` simulated units.

Author(s)

Daniel Gomon

See Also

`bk_cusum`

Other control limit simulation: `bernoulli_control_limit()`, `cgr_control_limit()`

Examples

```
require(survival)

#Determine a cox proportional hazards model for risk-adjustment
exprfit <- as.formula("Surv(survtime, censorid) ~ age + sex + BMI")
tcoxmod <- coxph(exprfit, data= surgerydat)

#Determine a control limit restricting type I error to 0.1 over 500 days
#with specified cumulative hazard function without risk-adjustment
a <- bk_control_limit(time = 500, alpha = 0.1, theta = log(2),
cbaseh = function(t) chaz_exp(t, lambda = 0.02),
inv_cbaseh = function(t) inv_chaz_exp(t, lambda = 0.02), psi = 0.5,
n_sim = 10)

#Determine a control limit restricting type I error to 0.1 over 500 days
#using the risk-adjusted cumulative hazard determined using coxph()
b <- bk_control_limit(time = 500, alpha = 0.1, theta = log(2),
coxphmod = tcoxmod, psi = 0.5, n_sim = 10)
```

```
print(a$h)
print(b$h)
```

---

bk_cusum *Continuous time BK-CUSUM*

Description

This function implements the BK-CUSUM procedure based on the Biswas & Kalbfleisch (2008) CUSUM. To construct the Biswas & Kalbfleisch (2008) CUSUM, set `C = 1` (years) or `C = 365` (days). For detection purposes, it is sufficient to determine the value of the chart at the times of failure. This can be achieved by leaving `ctimes` unspecified. The function requires the specification of `theta` and has two vital parameters, at least one of which must be specified:

- `coxphmod`: Cox proportional hazards model to be used for risk-adjustment. If `cbaseh` is not specified, it will be determined from `coxphmod` numerically.
- `cbaseh`: The cumulative baseline hazard rate to use for chart construction. If specified with `coxphmod` missing, no risk-adjustment will be performed

Usage

```
bk_cusum(data, theta, coxphmod, cbaseh, ctimes, h, stoptime, C,
  twosided = FALSE, pb = FALSE, assist)
```

Arguments

`data` A `data.frame` with rows representing subjects and the following named columns:

`entrytime`: time of entry into study (numeric);

`survtime`: time from entry until event (numeric);

`censorid`: censoring indicator (0 = right censored, 1 = observed), (integer).

and optionally additional covariates used for risk-adjustment.

`theta` The expected log-hazard ratio θ under the alternative hypothesis. If $\theta >= 0$, the chart will try to detect an increase in hazard ratio (upper one-sided). If $\theta < 0$, the chart will look for a decrease in hazard ratio (lower one-sided).

`coxphmod` A Cox proportional hazards regression model as produced by the function `coxph()`. Suggested:
`coxph(Surv(survtime,censorid) ~ covariates,data = data)`.
Alternatively, a list with the following elements:

`formula`: a `formula()` in the form `~ covariates`;

`coefficients`: a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in `formula` and colnames of `data`.

`cbaseh` A function that returns the unadjusted cumulative baseline hazard $H_0(t)$. If `cbaseh` is missing but `coxphmod` has been specified as a survival object, this baseline hazard rate will be determined using the provided `coxphmod`.

`ctimes` (optional): Vector of construction times at which the value of the chart should be determined. When not specified, the chart is constructed at all failure times.

`h` (optional): Value of the control limit. The chart will only be constructed until the value of the control limit has been reached or surpassed.

`stoptime` (optional): Time after which the value of the chart should no longer be determined. Default = max(failure time). Useful when ctimes has not been specified.

`C` (optional): A numeric value indicating how long after entering the study patients should no longer influence the value of the chart. This is equivalent to right-censoring every observation at time `entrytime` + C.

`twosided` (optional): A boolean indicating whether a two-sided CUSUM should be constructed. If `TRUE`, 2 CUSUM charts are determined. One to check for an increase of e^θ and one for a decrease of $e^{-\theta}$ in the hazard rate w.r.t. the baseline hazard. Default is `FALSE`.

`pb` (optional): A boolean indicating whether a progress bar should be shown. Default is `FALSE`.

`assist` (optional): Output of the function `parameter_assist()`

Details

The BK-CUSUM can be used to test the alternative hypothesis of an instant change of fixed size e^θ in the subject specific hazard rate from $h_i(t)$ to $h_i(t)e^\theta$. The parameter `C` can be used to ignore the contributions of subjects, C time units after their entry into the study. The BK-CUSUM is constructed as

$$G(t) = \max_{0 \leq k \leq t} \left(\theta N(k,t) - \left(e^\theta - 1\right) \Lambda(k,t)\right),$$

where θ is the log expected hazard ratio,

$$N(k,t) = N(t) - N(k)$$

with $N(t)$ the counting process of all failures at time t, and

$$\Lambda(k,t) = \Lambda(t) - \Lambda(k)$$

with $\Lambda(t)$ the summed cumulative intensity of all subjects at time t.

Value

An object of class `bkcusum` containing:

- `BK`: a `data.frame` containing the following named columns:

  `time`: times at which chart is constructed;

  `value`: value of the chart at corresponding times.

- `stopind`: indicator for whether the chart was stopped by the control limit;
- `call`: the call used to obtain output;
- `h`: Specified value for the control limit.

Author(s)

Daniel Gomon

References

Biswas P. and Kalbfleisch J.D. (2008), A risk-adjusted CUSUM in continuous time based on the Cox Model, Statistics in medicine 27, 3452-3452. doi: 10.1002/sim.3216

See Also

plot.bkcusum, runlength.bkcusum

Other quality control charts: bernoulli_cusum(), cgr_cusum(), funnel_plot()

Examples

```
require(survival)
#Select only the data of the first hospital in the surgerydat data set
tdat <- subset(surgerydat, unit == 1)

#We know that the cumulative baseline hazard in the data set is
#Exponential(0.01). If you don't know the cumulative baseline, we suggest
#leaving the cbaseh argument empty and determining a coxphmod (see help)
tcbaseh <- function(t) chaz_exp(t, lambda = 0.01)

#Determine a risk-adjustment model using a Cox proportional hazards model.
#Outcome (survival) is regressed on the available covariates:
exprfit <- as.formula("Surv(survtime, censorid) ~ age + sex + BMI")
tcoxmod <- coxph(exprfit, data= surgerydat)

#Determine the values of the chart
bk <- bk_cusum(data = tdat, theta = log(2), coxphmod = tcoxmod, cbaseh = tcbaseh, pb = TRUE)
#plot the BK-CUSUM (exact hazard)
plot(bk)

#Alternatively, cbaseh can be left empty when specifying coxphmod through coxph()
bk_cox <- bk_cusum(data = tdat, theta = log(2), coxphmod = tcoxmod, pb = TRUE)
#plot the BK-CUSUM (estimated hazard from coxph)
plot(bk_cox)
```

---

calc_risk *Calculate the Cox Proportional hazards relative risk associated with the covariates of subjects*

Description

This function can be used to calculate the change in the relative risk of a subject pertaining to their covariates under a specified Cox proportional hazards model.

Usage

```
calc_risk(data, coxphmod = NULL)
```

Arguments

`data`	A data frame containing the covariates to be used for risk-adjustment as named columns.
`coxphmod`	(optional): A cox proportional hazards model generated using `coxph()` or a list containing: `formula`: a `formula()` in the form ~ `covariates`; `coefficients`: a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in `formula` and colnames of `data`.

Details

The subject-specific relative risk is $e^{\beta Z_i}$, where β is a vector of regression coefficients and Z_i a vector of covariates for subject i.

Value

A vector of `nrow(data)` specifying the increased/decreased risk of failure for each subject.

Author(s)

Daniel Gomon

Examples

```
#Small example data
crdat <- data.frame(age = rnorm(10, 40, 5), BMI = rnorm(10, 24, 3))
#Example risk-adjustment list (can also specify coxphmod)
crlist <- list(formula = as.formula("~age + BMI"), coefficients = c("age"= 0.02, "BMI"= 0.009))
#Calculate the increase or decrease of the relative risk for the subjects
#in crdat.
calc_risk(crdat, crlist)
```

`cgr_control_limit` *Determine control limits for CGR-CUSUM by simulation*

Description

This function can be used to determine control limits for the CGR-CUSUM (`cgr_cusum`) procedure by restricting the type I error `alpha` of the procedure over `time`.

Usage

```
cgr_control_limit(time, alpha = 0.05, psi, n_sim = 20, coxphmod,
  baseline_data, cbaseh, inv_cbaseh, interval = c(0, 9e+12),
  h_precision = 0.01, ncores = 1, seed = 1041996, pb = FALSE,
  chartpb = FALSE, detection = "upper", maxtheta = Inf, assist)
```

Arguments

time
: A numeric value over which the type I error alpha must be restricted.

alpha
: A proportion between 0 and 1 indicating the required maximal type I error. Default is 0.05.

psi
: A numeric value indicating the estimated Poisson arrival rate of subjects at their respective units. Can be determined using parameter_assist().

n_sim
: An integer value indicating the amount of units to generate for the determination of the control limit. Larger values yield more precise control limits, but greatly increase computation times. Default is 20.

coxphmod
: (optional): A cox proportional hazards regression model as produced by the function coxph(). Suggested:
coxph(Surv(survtime,censorid) ~ covariates,data = baseline_data).
Alternatively, a list with the following elements:

formula: a formula() in the form ~ covariates;

coefficients: a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in formula and colnames of data.

baseline_data
: (optional): A data.frame used for covariate resampling with rows representing subjects and at least the following named columns:

entrytime: time of entry into study (numeric);

survtime: time from entry until event (numeric);

censorid: censoring indicator (0 = right censored, 1 = observed), (integer).

and optionally additional covariates used for risk-adjustment. Can only be specified in combination with coxphmod.

cbaseh
: (optional): A function that returns the unadjusted cumulative baseline hazard $H_0(t)$. If cbaseh is missing but coxphmod has been specified as a survival object, this baseline hazard rate will be determined using the provided coxphmod.

inv_cbaseh
: (optional): A function that returns the unadjusted inverse cumulative baseline hazard $H_0^{-1}(t)$. If inv_cbaseh is missing, it will be determined from cbaseh numerically.

interval
: (optional): Interval in which survival times should be solved for numerically.

h_precision
: (optional): A numerical value indicating how precisely the control limit should be determined. By default, control limits will be determined up to 2 significant digits.

ncores
: (optional): Number of cores to use to parallelize the computation of the CGR-CUSUM charts. If ncores = 1 (default), no parallelization is done. You can use detectCores() to check how many cores are available on your computer.

`seed` (optional): A numeric seed for survival time generation. Default = my birthday.

`pb` (optional): A boolean indicating whether a progress bar should be shown. Default is `FALSE`.

`chartpb` (optional): A boolean indicating whether progress bars should be displayed for the constructions of the charts. Default is `FALSE`.

`detection` Should the control limit be determined for an `"upper"` or `"lower"` CGR-CUSUM? Default is `"upper"`.

`maxtheta` (optional): Maximum value of maximum likelihood estimate for parameter θ. Default is `Inf`.

`assist` (optional): Output of the function `parameter_assist()`

Details

This function performs 3 steps to determine a suitable control limit.

- Step 1: Generates `n_sim` in-control units (failure rate as baseline). If `data` is provided, subject covariates are resampled from the data set.
- Step 2: Determines chart values for all simulated units.
- Step 3: Determines control limits such that at most a proportion `alpha` of all units cross the control limit.

The generated data as well as the charts are also returned in the output.

Value

A list containing three components:

- `call`: the call used to obtain output;
- `charts`: A list of length `n_sim` containing the constructed charts;
- `data`: A `data.frame` containing the in-control generated data.
- `h`: Determined value of the control limit.
- `achieved_alpha`: Achieved type I error on the sample of `n_sim` simulated units.

Author(s)

Daniel Gomon

See Also

`cgr_cusum`

Other control limit simulation: `bernoulli_control_limit()`, `bk_control_limit()`

Examples

```
require(survival)

#Determine a cox proportional hazards model for risk-adjustment
exprfit <- as.formula("Surv(survtime, censorid) ~ age + sex + BMI")
tcoxmod <- coxph(exprfit, data= surgerydat)

#Determine a control limit restricting type I error to 0.1 over 500 days
#with specified cumulative hazard function without risk-adjustment
a <- cgr_control_limit(time = 500, alpha = 0.1, cbaseh = function(t) chaz_exp(t, lambda = 0.02),
inv_cbaseh = function(t) inv_chaz_exp(t, lambda = 0.02), psi = 0.5, n_sim = 10)


#Determine a control limit restricting type I error to 0.1 over 500 days
#using the risk-adjusted cumulative hazard found using coxph()
b <- cgr_control_limit(time = 500, alpha = 0.1, coxphmod = tcoxmod, psi = 0.5, n_sim = 10,
baseline_data = subset(surgerydat, unit == 1))

print(a$h)
print(b$h)
```

cgr_cusum *Continuous time Generalized Rapid response CUSUM (CGR-CUSUM)*

Description

This function performs the CGR-CUSUM procedure described in Gomon et al. (in review). For detection purposes, it suffices to determine the value of the chart at the times of failure. This can be achieved by leaving `ctimes` unspecified. The function has two vital parameters, at least one of which must be specified:

- `coxphmod`: Cox proportional hazards model to be used for risk-adjustment. If `cbaseh` is not specified, it will be determined from `coxphmod` numerically.
- `cbaseh`: The cumulative baseline hazard rate to use for chart construction. If specified with `coxphmod` missing, no risk-adjustment will be performed

Usage

```
cgr_cusum(data, coxphmod, cbaseh, ctimes, h, stoptime, C, pb = FALSE,
  ncores = 1, cmethod = "memory", dependencies, detection = "upper",
  assist, maxtheta = Inf)
```

Arguments

`data` A `data.frame` with rows representing subjects and the following named columns:

`entrytime`**:** time of entry into study (numeric);

`survtime`**:** time from entry until event (numeric);

`censorid`**:** censoring indicator (0 = right censored, 1 = observed), (integer).

and optionally additional covariates used for risk-adjustment.

`coxphmod` A Cox proportional hazards regression model as produced by the function `coxph()`. Suggested:
`coxph(Surv(survtime,censorid) ~ covariates,data = data)`.
Alternatively, a list with the following elements:

`formula`**:** a `formula()` in the form `~ covariates`;

`coefficients`**:** a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in `formula` and colnames of `data`.

`cbaseh` A function that returns the unadjusted cumulative baseline hazard $H_0(t)$. If `cbaseh` is missing but `coxphmod` has been specified as a survival object, this baseline hazard rate will be determined using the provided `coxphmod`.

`ctimes` (optional): Vector of construction times at which the value of the chart should be determined. When not specified, the chart is constructed at all failure times.

`h` (optional): Value of the control limit. The chart will only be constructed until the value of the control limit has been reached or surpassed.

`stoptime` (optional): Time after which the value of the chart should no longer be determined. Default = max(failure time). Useful when ctimes has not been specified.

`C` (optional): A numeric value indicating how long after entering the study patients should no longer influence the value of the chart. This is equivalent to right-censoring every observation at time `entrytime` + C.

`pb` (optional): A boolean indicating whether a progress bar should be shown. Default is `FALSE`.

`ncores` number of cores to use to parallelize the computation of the CGR-CUSUM chart. If ncores = 1 (default), no parallelization is done. You can use `detectCores()` to check how many cores are available on your computer.

`cmethod` Method to calculate chart values. One of the following:

- `"memory"` (default): matrix formulation of the problem (faster for high volume/long time construction)
- `"CPU"` calculates the value of the CGR-CUSUM for every time point from scratch. Recommended for small data volume (lower initialization time).

`dependencies` (optional): When `ncores > 1`, specify a list of variables/functions/other dependencies to be exported to the core clusters for parallel computation.

`detection` Should an `"upper"` or `"lower"` CGR-CUSUM be constructed. Upper CUSUMs can be used to monitor for an increase in the failure rate, while lower CUSUMs can be used to monitor for a decrease in the failure rate.

`assist` (optional): Output of the function `parameter_assist()`

`maxtheta` (optional): Maximum value of maximum likelihood estimate for parameter θ. Default is `Inf`.

Details

The CGR-CUSUM can be used to test for a change of unknown positive fixed size θ in the subject-specific hazard rate from $h_i(t)$ to $h_i(t)e^\theta$ starting from some unknown subject ν. The starting time of the first subject who had an increase in failure rate as well as the estimated increase in the hazard rate are shown in the output. The CGR-CUSUM is determined as

$$\max_{1 \leq \nu \leq n} \left(\hat{\theta}_{\geq \nu}(t) N_{\geq \nu}(t) - \left(\exp\left(\hat{\theta}_{\geq \nu}(t) \right) - 1 \right) \Lambda_{\geq \nu}(t) \right),$$

where

$$N(\geq \nu)(t) = \sum_{i \geq \nu} N_i(t),$$

with $N_i(t)$ the counting process for the failure at time t of subject i and

$$\Lambda_{\geq \nu}(t) = \sum_{i \geq \nu} \Lambda_i(t),$$

where $\Lambda_i(t)$ is the cumulative intensity of subject i at time t.

Value

An object of class "cgrcusum" containing:

- `CGR`: a `data.frame` with named columns:

  `time`: times at which chart is constructed;

  `value`: value of the chart at corresponding times;

  `exp_theta_t`: value of MLE e^{θ_t};

  `S_nu` time from which patients are considered for constructing the chart.
- `call`: the call used to obtain output;
- `stopind`: indicator for whether the chart was stopped by the control limit;
- `h`: Specified value for the control limit.

Author(s)

Daniel Gomon

See Also

`plot.cgrcusum`, `runlength.cgrcusum`

Other quality control charts: `bernoulli_cusum()`, `bk_cusum()`, `funnel_plot()`

Examples

```
require(survival)
#Select only the data of the first year of the first hospital in the surgerydat data set
tdat <- subset(surgerydat, unit == 1 & entrytime < 365)

#We know that the cumulative baseline hazard in the data set is
#Exponential(0.01). If you don't know the cumulative baseline, we suggest
```

```
#leaving the cbaseh argument empty and determining a coxphmod (see help)
tcbaseh <- function(t) chaz_exp(t, lambda = 0.01)

#Determine a risk-adjustment model using a Cox proportional hazards model.
#Outcome (survival) is regressed on the available covariates:
exprfit <- as.formula("Surv(survtime, censorid) ~ age + sex + BMI")
tcoxmod <- coxph(exprfit, data= surgerydat)

#Determine the values of the chart
cgr <- cgr_cusum(data = tdat, coxphmod = tcoxmod, cbaseh = tcbaseh, pb = TRUE)
#plot the CGR-CUSUM (exact hazard)
plot(cgr)

#Alternatively, cbaseh can be left empty when specifying coxphmod through coxph()
cgr_cox <- cgr_cusum(data = tdat, coxphmod = tcoxmod, pb = TRUE)
#plot the CGR-CUSUM (estimated hazard from coxph)
plot(cgr_cox)
```

exp_hazards *Exponential hazard, cumulative hazard and inverse cumulative hazard*

Description

Functions which return the hazard, cumulative hazard and inverse cumulative hazard at time t for an exponential distribution with parameter λ and true hazard ratio μ.

Usage

```
haz_exp(t, lambda)

chaz_exp(t, lambda, mu = log(1))

inv_chaz_exp(t, lambda, mu = log(1))
```

Arguments

t	time of evaluation.
lambda	parameter of the exponential distribution.
mu	(optional) true excess hazard rate μ.

Details

The hazard function of an exponential distribution is given by:

$$h(\lambda) = \lambda$$

The cumulative hazard (with true hazard ratio μ) is given by:

$$H(\lambda, \mu) = \lambda t e^{\mu}$$

The inverse cumulative hazard (with true hazard ratio μ) by:

$$H^{-1}(\lambda, \mu) = \frac{t}{\lambda e^{\mu}}$$

Value

Value of specified function at time t.

`extract_hazard` *Extract (inverse) cumulative baseline hazard from Cox PH model*

Description

Extracts a function which returns the (inverse) cumulative baseline hazard from a `coxph()` call.

Usage

```
extract_hazard(coxphmod)
```

Arguments

`coxphmod`	A call to `coxph()`.

Value

A list containing:

- `cbaseh`: A function which returns the cumulative baseline hazard at specified time;
- `inv_cbaseh`: A function which returns the inverse cumulative baseline hazard at specified time.
- `max_time`: maximal time at which `cbaseh` is known;
- `max_haz`: value of maximal hazard (at maximum time).

Author(s)

Daniel Gomon

See Also

`coxph`

Examples

```
require(survival)
exprfit <- as.formula("Surv(survtime, censorid) ~ age + sex + BMI")
tcoxmod <- coxph(exprfit, data= surgerydat)
tcox_hazard_fcts <- extract_hazard(tcoxmod)
```

`funnel_plot` *Risk-adjusted funnel plot*

Description

This function allows to construct a risk-adjusted funnel plot for comparing survival proportion between units.

Usage

```
funnel_plot(data, ctime, p0, glmmod, followup, conflev = c(0.95, 0.99), assist)
```

Arguments

`data` A `data.frame` with rows representing subjects and the following named columns:

- `entrytime`**:** time of entry into study (numeric);
- `survtime`**:** time from entry until event (numeric);
- `censorid`**:** censoring indicator (0 = right censored, 1 = observed), (integer);
- `unit`**:** integer or character indicating which unit (f.e. hospital) the observation belongs to.

and optionally additional covariates used for risk-adjustment.

`ctime` Construction time at which the funnel plot should be determined. Maximum possible time used when not specified.

`p0` The baseline failure probability at `entrytime` + `followup` for individuals. If not specified, average failure proportion over whole data is used instead.

`glmmod` A generalized linear regression model as produced by the function `glm()`. Recommended:
`glm(as.formula("(survtime <= followup) & (censorid == 1) ~ covariates"),data = data)`.
Alternatively, a list with the following elements:

- `formula`**:** a `formula()` in the form ~ `covariates`;
- `coefficients`**:** a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in `formula` and colnames of `data`.

`followup` The followup time for every individual. At what time after subject entry do we consider the outcome?

`conflev` A vector of confidence levels of interest. Default is c(0.95, 0.99).

`assist` (optional): Output of the function `parameter_assist()`

Value

An object of class "funnelplot" containing:

- `data`: A `data.frame` containing:

  `unit`**:** unit number/name;

`observed`: observed number of failures at unit;

`expected`: expected (risk-adjusted) number of failures at unit;

`numtotal` total number of individuals considered at this unit;

`p`: (risk-adjusted) proportion of failure at unit;

`conflevels`: worse/in-control/better performance than expected at specified confidence levels.

- `call`: the call used to obtain output
- `plotdata`: data used for plotting confidence intervals
- `conflev`: specified confidence level(s)
- `p0`: (Estimated) baseline failure probability

Author(s)

Daniel Gomon

See Also

`plot.funnelplot`, `summary.funnelplot`

Other quality control charts: `bernoulli_cusum()`, `bk_cusum()`, `cgr_cusum()`

Examples

```
#Determine a risk-adjustment model using a generalized linear model.
#Outcome (survival in first 100 days) is regressed on the available covariates:
exprfitfunnel <- as.formula("(survtime <= 100) & (censorid == 1)~ age + sex + BMI")
glmmodfun <- glm(exprfitfunnel, data = surgerydat, family = binomial(link = "logit"))
#Determine the necessary values to produce a funnel plot
funnel <- funnel_plot(data = surgerydat, ctime = 3*365, glmmod = glmmodfun, followup = 100)
#Produce a funnel plot!
plot(funnel)
## Not run:
require(plotly)
#Create an interactive plot!
ggplotly(plot(funnel))

## End(Not run)
```

`generate_units` *Generate units with specified failure rate*

Description

Generate `n_sim` units with subjects arriving according to a Poisson process with rate `psi` until `time`. Failure rate is determined either from `cbaseh` or `inv_cbaseh`, or from specified `coxphmod`. Covariates will be resampled from `baseline_data` if specified.

Usage

```
generate_units(time, psi, n_sim = 20, cbaseh, inv_cbaseh, coxphmod = NULL,
  baseline_data, interval = c(0, 9e+12), mu = 0)
```

Arguments

time
: A numeric value indicating until what time from the start of the study subjects can arrive.

psi
: Poisson arrival rate for subjects.

n_sim
: An integer indicating how many units should be generated. Default is 20.

cbaseh
: A function returning the cumulative baseline hazard at each relevant time point. Will be numerically inverted to generate failure times. If the inverse cumulative baseline hazard function is available, please specify inv_cbaseh instead.

inv_cbaseh
: A function returning the inverse cumulative baseline hazard at each relevant time point.

coxphmod
: (optional): A cox proportional hazards model generated using coxph() or a list containing:

  formula: a formula() in the form ~ covariates;

  coefficients: a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in formula and colnames of data.

  If both cbaseh and inv_cbaseh are missing, the hazard rate will be determined from coxphmod.

baseline_data
: (optional): A data.frame used for covariate resampling with rows representing subjects and columns containing covariates to use for risk-adjustment. Should only be specified in combination with coxphmod.

interval
: (optional) A numeric vector of length 2 indicating in which range of values the failure times of subjects should be determined. By default, failure times will be restricted between 0 and 9e12.

mu
: (optional) The increased log hazard ratio at the generated units with respect to the specified baseline hazard rate. Default is log(1) = 0.

Details

In a Poisson arrival process, inter-arrival times are exponentially distributed with parameter psi. If cbaseh is specified, the inverse baseline hazard will be determined using uniroot(). The times of failure are then determined using gen_surv_times().

Value

A data.frame with rows representing subjects and the following named columns:

entrytime: time of subject entry into the study;

survtime: survival time of subject;

censorid: censoring indicator, 0 = censored, 1 = observed;

unit: unit number;

`expmu`**:** exponent of the log hazard ratio used to generate survival times;

`psival`**:** arrival rate at unit;

`covariates`**:** covariates resampled from `baseline_data`.

Author(s)

Daniel Gomon

Examples

```
require(survival)
#Fit a Cox model
exprfit <- as.formula("Surv(survtime, censorid) ~ age + sex + BMI")
tcoxmod <- coxph(exprfit, data= surgerydat)

#Generate 30 hospitals with on average 2 patients per day arriving
#according to the Cox model determined above, with resampling from the
#original data set. The hazard rate at the hospitals is twice the baseline
#hazard.
generate_units(time = 50, psi = 2, n_sim = 30, coxphmod = tcoxmod,
baseline_data = surgerydat, mu = log(2))
```

`gen_arriv_times` *Generate arrival times according to a Poisson point process*

Description

This function can be used to generate arrival times for a Poisson point process with rate psi up until time t.

Usage

```
gen_arriv_times(psi, t)
```

Arguments

`psi`	rate of the arrival process.
`t`	time until which arrivals should be generated.

Details

Exponential(ψ) interarrival times.

Value

A vector of arrival times up until time t.

Author(s)

Daniel Gomon

Examples

```
set.seed(123)
gen_arriv_times(psi = 0.3, t = 5)
gen_arriv_times(psi = 0.3, t = 20)
```

---

gen_surv_times *Generate survival times*

Description

Generate survival times according to hazard rate $h(t)\exp(\mu)$ with $h(t)$ the hazard rate associated with the specified inverse cumulative hazard rate `invchaz` and μ the specified true hazard ratio `mu`. See Bender et al. (2005).

Usage

```
gen_surv_times(invchaz, mu = log(1), data, coxphmod = NULL)
```

Arguments

`invchaz`	the inverse cumulative (baseline) hazard rate to be used for generating survival times. Must take vector inputs!
`mu`	the true hazard ratio used to generate survival times.
`data`	an integer number of survival times to generate or (in combination with coxphmod): a `data.frame` containing subject covariates in named columns.
`coxphmod`	(optional) a cox proportional hazards regression model as produced by the function `coxph()`. Standard practice: `coxph(Surv(survtime,censorid) ~ covariates,data = data)`. Alternatively, a list with: • $formula (~ covariates) • $coefficients (named vector specifying risk adjustment coefficients for covariates - names must be the same as in $formula and colnames of `data`).

Details

Sometimes it is desirable to generate survival times from an increased hazard rate

$$h(t, \mu) = h_0(t)e^{\mu}$$

with h_0 the baseline hazard rate. We call e^{μ} the true hazard ratio.

Value

A vector of survival times from subject entry time.

Author(s)

Daniel Gomon

References

Bender, R., Augustin, T., & Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. Statistics in Medicine, 24(11), 1713-1723. doi: 10.1002/sim.2059

Examples

```
set.seed(123)
gen_surv_times(invchaz = function(t) inv_chaz_exp(t, lambda = 0.01), data = 5)
```

`interactive_plot` *Plot a list of CUSUM charts (interactive)*

Description

Create an interactive plot visualizing a combination of control charts which can be created using this package.

Usage

```
interactive_plot(x, unit_names, scale = FALSE, group_by = c("none", "unit",
  "type"), highlight = FALSE, manual_colors = c(), ...)
```

Arguments

`x` A list with each item containing one cumulative sum chart.

`unit_names` The unit names to be displayed in the interactive plot. Must be of equal length as `x`.

`scale` Should charts be scaled with respect to their control limit/ maximum value? Default is `FALSE`.

`group_by` Character indicating how to group the CUSUM charts in the plot. Possible values are `c("none","unit","type")`. Default is `"none"`.

`highlight` Should charts be highlighted on hover? Default is `FALSE`.

`manual_colors` A character vector specifying which colors to use for the units in the data. By default, the "Dark2" color set from `brewer.pal()` will be used.

`...` Further plotting parameters

Value

An interactive plot will be produced in the current graphics device. For more information on the possibilities for interaction, see https://plotly.com/r/.

See Also

cgr_cusum, bk_cusum, bernoulli_cusum, funnel_plot

Examples

```
require(survival)
#Extract data to construct CUSUM charts on
tdat <- subset(surgerydat, unit == 1 & entrytime < 365)
tdat2 <- subset(surgerydat, unit == 2 & entrytime < 365)

#Determine model parameters
followup <- 100
tcbaseh <- function(t) chaz_exp(t, lambda = 0.01)
exprfit <- as.formula("Surv(survtime, censorid) ~ age + sex + BMI")
tcoxmod <- coxph(exprfit, data= surgerydat)
exprfitber <- as.formula("(survtime <= followup) & (censorid == 1)~ age + sex + BMI")
glmmodber <- glm(exprfitber, data = surgerydat, family = binomial(link = "logit"))


#Construct the charts
cgr <- cgr_cusum(data = tdat, coxphmod = tcoxmod, cbaseh = tcbaseh, pb = TRUE)
cgr$h <- 8.29
bk <- bk_cusum(data = tdat, theta = log(2), coxphmod = tcoxmod, cbaseh = tcbaseh, pb = TRUE)
bk$h <- 6.23
bercus <- bernoulli_cusum(data = subset(surgerydat, unit == 1), glmmod = glmmodber,
followup = followup, theta = log(2))
bercus$h <- 3.36
bk2 <- bk_cusum(data = tdat2, theta = log(2), coxphmod = tcoxmod, cbaseh = tcbaseh, pb = TRUE)
bk2$h <- 6.23

#Create the plot
interactive_plot(list(cgr, bk, bercus, bk2), unit_names =
c("hosp1", "hosp1", "hosp1", "hosp2"))
```

---

parameter_assist *Assist users in parameter selection*

Description

This function can be used to determine some of the vital parameters used to construct control charts in this package.

Usage

```
parameter_assist(baseline_data, data, formula, followup, theta = log(2),
  time, alpha = 0.05)
```

Arguments

`baseline_data` A `data.frame` for determining a baseline performance metric. Rows should represent subjects and the following named columns should be present:

`entrytime`: time of entry into study (numeric);

`survtime`: time from entry until event (numeric);

`censorid`: censoring indicator (0 = right censored, 1 = observed), (integer).

and optionally additional covariates used for risk-adjustment.

`data` A `data.frame` with data on which the user wants to construct quality control charts. Rows should represent subjects and the following named columns should be present:

`entrytime`: time of entry into study (numeric);

`survtime`: time from entry until event (numeric);

`censorid`: censoring indicator (0 = right censored, 1 = observed), (integer).

and optionally additional covariates used for risk-adjustment.

`formula` A formula with right-hand side (RHS) indicating the form in which the covariates should be used for the Cox and GLM regression models. LHS of the formula will be ignored, and can be left empty.

`followup` (optional): The value of the follow-up time to be used to determine event time. Event time will be equal to `entrytime` + `followup` for each subject.

`theta` The value of the expected log-hazard ratio. In other words: the logarithm of the expected increase in the odds of failure/failure rate. Default is log(2) (detecting a doubling of the odds/failure rate).

`time` Timeframe over which the type I error of the control chart should be limited. Should be in the same unit as `survtime` in `data`. If left unspecified, the maximum entrytime in `baseline_data` is taken. (numeric)

`alpha` Required maximal type I error (between 0 and 1) of the procedure over the timeframe specified in `time`. Default is 0.05. (numeric)

Details

Depending on the specified arguments, the function will return parameters. If `covariate_names` is not specified, the returned risk-adjustment models will be trivial. If `formula` is not specified but `covariate_names` are, the function assumes the simplest form for the regression model (cov1 + cov2 + ...). If `followup` is not specified, no `glmmod` will be determined

Value

A list of parameters to feed to quality control charts in this package

- call: The call used to obtain output.

- data: The data used in the call to the function.
- baseline_data: The baseline_data used in the call to the function
- glmmod: A `glm()` model which can be fed to the `funnel_plot()` and `bernoulli_cusum()` functions.
- coxphmod: A `coxph()` model which can be fed to the `cgr_cusum()` and `cgr_cusum()` functions.
- psi: Estimated Poisson arrival rate in `data`.

Author(s)

Daniel Gomon

Examples

```
require(survival)

#Minimal example - no risk-adjustment
pars_min <- parameter_assist(baseline_data = surgerydat,
data = subset(surgerydat, unit == 1))

#Specifying all parameters
pars <- parameter_assist(baseline_data = surgerydat,
data = subset(surgerydat, unit == 1),
formula = formula("survtime ~ age + sex + BMI"), followup = 100)
```

`plot.cgrcusum` *Plot a quality control chart*

Description

Plot a `cgrcusum`, `bkcusum`, `bercusum` or `funnelplot` chart, or a list containing a combination of `'bercusum'`, `'bkcusum'` and `'cgrcusum'` charts.

Usage

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

## S3 method for class 'bkcusum'
plot(x, h, ...)

## S3 method for class 'funnelplot'
plot(x, percentage = TRUE, ...)

## S3 method for class 'bercusum'
plot(x, h = x$h, ...)
```

Arguments

`x`	Chart to plot
`h`	Control limit to display for `cgrcusum`, `bkcusum` or `bercusum`
`...`	Further plotting parameters
`percentage`	Should output be shown in percentages? Default is `TRUE`.

Value

A plot of the associated chart is displayed in the current graphics device.

Methods (by class)

- `cgrcusum`: Plot a CGR-CUSUM
- `bkcusum`: Plot a BK-CUSUM
- `funnelplot`: Display a funnel plot
- `bercusum`: Plot a Bernoulli CUSUM

Author(s)

Daniel Gomon

See Also

`cgr_cusum`, `bk_cusum`, `bernoulli_cusum`, `funnel_plot`

`runlength` *Determine run length of a CUSUM chart*

Description

This function can be used to calculate the run length of a 'cgrcusum', 'bkcusum' or 'bercusum' chart when using control limit h

Usage

```
runlength(chart, h)

## S3 method for class 'cgrcusum'
runlength(chart, h, ...)

## S3 method for class 'bkcusum'
runlength(chart, h, ...)

## S3 method for class 'bercusum'
runlength(chart, h, ...)
```

Arguments

`chart`	A `cgrcusum`, `bkcusum` or `bercusum` chart.
`h`	Control limit h to be used when determining the run length
`...`	Other parameters

Value

The run length of the chart with the given control limit.

Methods (by class)

- `cgrcusum`: determines runlength of `cgrcusum` object
- `bkcusum`: determines runlength of `bkcusum` object
- `bercusum`: determines runlength of `bercusum` object

Author(s)

Daniel Gomon

Examples

```
exprfitber <- as.formula("(survtime <= 100) & (censorid == 1) ~ age + sex + BMI")
glmmodber <- glm(exprfitber, data = surgerydat, family = binomial(link = "logit"))
bercus <- bernoulli_cusum(data = subset(surgerydat, unit == 14), glmmod = glmmodber,
                   followup = 100, theta = log(2))
#Determine the run length of the above Bernoulli CUSUM when using a control limit
#of h = 1.
runlength(bercus, h = 1)
```

`summary.funnelplot` *Summarizes S3 objects in this package.*

Description

Prints a summary of the `funnelplot` object.

Usage

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

Arguments

`object`	S3 object to summarize
`...`	extra parameters

Value

A `data.frame` with:

- `unit`: unit number/identifier;
- `observed`: the observed amount of failures at respective unit;
- `expected`: the expected amount of failures at respective unit, given that the unit is performing at target;
- `numtotal`: total number of subjects at respective unit;
- `p`: estimated probability of failure at unit;
- `'0.xx'`: better/normal/worse proportion of failure at specified confidence levels.

Methods (by class)

- `funnelplot`: summarize instances detected by the `funnelplot` object

See Also

`funnel_plot`

`surgerydat` *Simulated data set with data of surgery procedures performed at multiple hospitals.*

Description

Data about patients and their surgery procedure from 45 simulated hospitals with patient arrivals in the first 400 days after the start of the study.
Patient survival times were determined using a risk-adjusted Cox proportional hazards model with coefficients age = 0.003, BMI = 0.02 and sexmale = 0.2 and exponential baseline hazard rate $h_0(t, \lambda = 0.01)e^\mu$. Some hospitals have an increased failure rate:

- Hospitals 1-15: $\mu \sim N(log(0.5), sd = 0.25)$
- Hospitals 16-30: $\mu \sim N(log(1), sd = 0.25)$
- Hospitals 31-45: $\mu \sim N(log(2), sd = 0.25)$

which means that the hazard rate at hospitals 31-45 is twice higher than exponential(λ) and twice smaller at hospitals 1-15.
The arrival rate ψ of patients at a hospital differs. The arrival rates are:

- Hospitals 1-5 & 16-20: 0.5 patients per day (small hospitals)
- Hospitals 6-10 & 21-25: 1 patient per day (medium sized hospitals)
- Hospitals 11-15 & 26-30: 1.5 patients per day (large hospitals)

These are then respectively small, medium and large hospitals.

Usage

```
surgerydat
```

Format

A `data.frame` with 12010 rows and 9 variables:

entrytime Time of entry of patient into study (numeric)

survtime Time from entry until failure of patient (numeric)

censorid Censoring indicator (0 - right censored, 1 - observed) (integer)

hosp_num Hospital number at which patient received treatment (integer)

expmu True excess hazard used for generating patient survival (numeric)

psival Poisson arrival rate at hospital which the patient was at (numeric)

age Age of the patient (numeric)

sex Sex of the patient (factor)

BMI Body mass index of the patient (numeric)

Index