

Package ‘success’

June 19, 2024

Title Survival Control Charts Estimation Software

Version 1.1.0

Maintainer Daniel Gomon <dgstatsoft@gmail.com>

Description Quality control charts for survival outcomes.

Allows users to construct the Continuous Time Generalized Rapid Response CUSUM (CGR-CUSUM) <doi:10.1093/biostatistics/kxac041>, the Biswas & Kalbfleisch (2008) <doi:10.1002/sim.3216> CUSUM, the Bernoulli CUSUM and the risk-adjusted funnel plot for survival data <doi:10.1002/sim.1970>.

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, Rfast, R (>= 3.5.0)

Imports survival, ggrepel, Matrix, matrixcalc

Suggests gridExtra, knitr, rmarkdown, testthat (>= 3.1.10), covr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.2.3

NeedsCompilation yes

Author Daniel Gomon [aut, cre] (<<https://orcid.org/0000-0001-9011-3743>>),
Mirko Signorelli [ctb] (<<https://orcid.org/0000-0002-8102-3356>>)

Repository CRAN

Date/Publication 2024-06-19 14:10:02 UTC

Contents

arrival_rate	2
bernoulli_ARL	3
bernoulli_control_limit	6
bernoulli_cusum	8
bernoulli_RL_cdf	10
bk_control_limit	12
bk_cusum	15
breast	17
calc_risk	19
cgr_control_limit	20
cgr_cusum	22
exp_hazards	25
extract_hazard	26
funnel_plot	27
generate_units	29
gen_arriv_times	31
gen_surv_times	32
interactive_plot	33
parameter_assist	34
plot.cgrcusum	36
runlength	38
summary.funnelplot	39
surgerydat	40
weib_hazards	41
Index	42

arrival_rate	<i>Estimate arrival rate of a Poisson points process</i>
--------------	--

Description

In a Poisson point process, subjects arrive with exponentially distributed inter-arrival times with rate ψ . This function can be used to estimate the parameter ψ .

Usage

```
arrival_rate(data)
```

Arguments

data	A data.frame containing the following named column for each subject: entrytime: time of entry into study (numeric); If the data.frame also contains a column named unit, the arrival rate will be determined for each unit separately.
------	--

Value

A (named) vector containing the estimated arrival rate in the data, or for each unit in the data.

Author(s)

Daniel Gomon

Examples

```
arrival_rate(surgerydat)
```

 bernoulli_ARL

Average Run Length for Bernoulli CUSUM

Description

This function allows to estimate the Average Run Length (ARL) of the risk-adjusted Bernoulli CUSUM (see [bernoulli_cusum\(\)](#)) through a Markov Chain Approach (Brook & Evans(1972) & Steiner et al. (2000)) or exploiting the relationship with the Sequential Probability Ratio Test (Kemp (1971)). The function requires the specification of one of the following combinations of parameters as arguments to the function:

- glmmod & theta
- p_0 & theta
- p_0 & p_1

Average run length of lower-sided Bernoulli CUSUM charts can be determined by specifying $\theta < 0$.

Usage

```
bernoulli_ARL(h, n_grid, glmmod, theta, theta_true, p0, p1, method = c("MC",
  "SPRT"), smooth_prob = FALSE)
```

Arguments

h	Control limit for the Bernoulli CUSUM
n_grid	Number of state spaces used to discretize the outcome space (when method = "MC") or number of grid points used for trapezoidal integration (when method = "SPRT"). Increasing this number improves accuracy, but can also significantly increase computation time.
glmmod	Generalized linear regression model used for risk-adjustment as produced by the function glm() . Suggested: <code>glm(as.formula("(survtime <= followup) & (censord == 1) ~ covariates"), data = data).</code> Alternatively, a list containing the following elements:

formula: a `formula()` in the form \sim 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 θ value used to specify the odds ratio e^θ under the alternative hypothesis. If $\theta \geq 0$, the average run length for the upper one-sided Bernoulli CUSUM will be determined. If $\theta < 0$, the average run length for the lower one-sided CUSUM will be determined. Note that

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

theta_true The true log odds ratio θ , describing the true increase in failure rate from the null-hypothesis. Default = $\log(1)$, indicating no increase in failure rate.

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

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

method The method used to obtain the average run length. Either "MC" for Markov Chain or "SPRT" for SPRT methodology. Default = "MC".

smooth_prob Should the probability distribution of failure under the null distribution be smoothed? Useful for small samples. Can only be TRUE when glmmmod is supplied. Default = FALSE.

Details

The average run length of a CUSUM chart S_n is given by $E[\tau_n]$, where τ_n is defined as:

$$\tau_n = \inf\{n \geq 0 : S_n \geq h\}.$$

When method = "MC", the average run length will be determined by the Markov Chain approach described in Brook & Evans (1972), using the risk-adjustment correction proposed in Steiner et al. (2000). The idea is to discretize the domain $(0, h)$ into $n_{grid} - 1$ state spaces, with E_0 of width $w/2$ and $E_1, \dots, E_{n_{grid}-1}$ of width w , such that $E_{n_{grid}}$ is an absorbing state. This is done using the following steps:

- w is determined using the relationship $\frac{2h}{2t-1}$.
- Transition probabilities between the states are determined and 'transition matrix' R is constructed.
- The equation $(I - R)ARL = \mathbf{1}$ is solved to find the ARL starting from each of the states.

When method = "SPRT", the average run length will be determined by the relationship between the SPRT and CUSUM described in Kemp (1971), using the risk-adjustment correction proposed in Steiner et al. (2000). If N is the run length of a SPRT, $P(0)$ the probability of a SPRT terminating on the lower boundary of zero and R the run length of a CUSUM, then:

$$E[R] = \frac{E[N]}{1 - P(0)}.$$

$E[N]$ and $P(0)$ are completely determined by

$$G_n(z) = \int_0^h F(z-w) dG_{n-1}(w)$$

with $F(x)$ the cdf of the singletons W_n . The integral can be approximated using the generalized trapezoidal quadrature rule:

$$G_n(z) = \sum_{i=0}^{n_{grid}-1} \frac{F(z-x_{i+1}) + F(z-x_i)}{2} (G_{n-1}(x_{i+1}) - G_{n-1}(x_i))$$

Value

A list containing:

- **ARL_0**: A numeric value indicating the average run length in number of outcomes when starting from state E_0 .
- **ARL**: A data frame containing the average run length (#outcomes) depending on the state in which the process starts ($E_0, E_1, \dots, E_{n_{grid}-1}$)
start_val: Starting value of the CUSUM, corresponding to the discretized state spaces E_i ;
#outcomes: ARL for the CUSUM with initial value **start_val**;
- **R**: A transition probability matrix containing the transition probabilities between states E_0, \dots, E_{t-1} . $R_{i,j}$ is the transition probability from state i to state j .
- **h**: Value of the control limit.

The value of **ARL_0** will be printed to the console.

Author(s)

Daniel Gomon

References

- Brook, D., & Evans, D. A. (1972). An Approach to the Probability Distribution of CUSUM Run Length. *Biometrika*, 59(3), 539-549. doi:10.2307/2334805
- Steiner, S. H., Cook, R. J., Farewell, V. T., & Treasure, T. (2000). Monitoring surgical performance using risk-adjusted cumulative sum charts. *Biostatistics*, 1(4), 441-452. doi:10.1093/biostatistics/1.4.441
- Kemp, K. W. (1971). Formal Expressions which Can Be Applied to Cusum Charts. *Journal of the Royal Statistical Society. Series B (Methodological)*, 33(3), 331-360. doi:10.1111/j.2517-6161.1971.tb01521.x

See Also

[bernoulli_cusum](#), [bernoulli_control_limit](#)

Examples

```
#Determine a risk-adjustment model using a generalized linear model.
#Outcome (failure within 100 days) is regressed on the available covariates:
glmmodber <- glm((survtime <= 100) & (censorid == 1)~ age + sex + BMI,
  data = surgerydat, family = binomial(link = "logit"))
#Determine the Average Run Length in number of outcomes for
#control limit h = 2.5 with (0, h) divided into n_grid = 200 segments
ARL <- bernoulli_ARL(h = 2.5, n_grid = 200, glmmod = glmmodber, theta = log(2))
#Calculate ARL, but now exploiting connection between SPRT and CUSUM:
#n_grid now decides the accuracy of the Trapezoidal rule for integral approximation
ARLSPRT <- bernoulli_ARL(h = 2.5, n_grid = 200, glmmod = glmmodber,
  theta = log(2), method = "SPRT")
```

```
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: <code>glm(as.formula("(survtime <= followup) & (censorid == 1) ~ covariates"), data = data).</code> Alternatively, a list containing the following elements:

	<p>formula: a <code>formula()</code> in the form <code>~ covariates</code>;</p> <p>coefficients: a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in formula and colnames of data.</p>
baseline_data	<p>(optional): A <code>data.frame</code> used for covariate resampling with rows representing subjects and at least the following named columns:</p> <p>entrytime: time of entry into study (numeric);</p> <p>survtime: time from entry until event (numeric);</p> <p>sensorid: censoring indicator (0 = right censored, 1 = observed), (integer).</p> <p>and optionally additional covariates used for risk-adjustment. Can only be specified in combination with <code>coxphmod</code>.</p>
theta	<p>The θ value used to specify the odds ratio e^θ under the alternative hypothesis. If $\theta \geq 0$, 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). Note that</p> $p_1 = \frac{p_0 e^\theta}{1 - p_0 + p_0 e^\theta}.$
p0	The baseline failure probability at <code>entrytime + followup</code> for individuals.
p1	The alternative hypothesis failure probability at <code>entrytime + followup</code> 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 <code>parameter_assist()</code>

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
- p_0 & theta
- p_0 & p_1

Usage

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


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 <code>glm()</code> . Suggested: <code>glm(as.formula("(survtime <= followup) & (censorid == 1) ~ covariates"), data = data)</code> . Alternatively, a list containing the following elements: formula: a <code>formula()</code> in the form <code>~ covariates</code> ; coefficients: a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in formula and colnames of data.
theta	The θ value used to specify the odds ratio e^θ under the alternative hypothesis. If $\theta \geq 0$, 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). Note that
	$p_1 = \frac{p_0 e^\theta}{1 - p_0 + p_0 e^\theta}.$
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 <code>parameter_assist()</code>
twosided	(optional): Should a two-sided Bernoulli CUSUM be constructed? Default is FALSE.

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. In terms of the Odds Ratio:

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

For a risk-adjusted procedure (when `glmmod` is specified), a patient specific baseline failure probability p_{0i} is modelled using logistic regression first. 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](#)

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)
```

bernoulli_RL_cdf

Cumulative distribution function (cdf) of Run Length for Bernoulli CUSUM

Description

Calculate the cdf of the Run Length of the Bernoulli CUSUM, starting from initial value between 0 and h , using Markov Chain methodology.

Usage

```
bernoulli_RL_cdf(h, x, n_grid, glmmod, theta, theta_true, p0, p1,
  smooth_prob = FALSE, exact = TRUE)
```

Arguments

h	Control limit for the Bernoulli CUSUM
x	Quantile at which to evaluate the cdf.
n_grid	Number of state spaces used to discretize the outcome space (when method = "MC") or number of grid points used for trapezoidal integration (when method = "SPRT"). Increasing this number improves accuracy, but can also significantly increase computation time.
glmmod	Generalized linear regression model used for risk-adjustment as produced by the function <code>glm()</code> . Suggested: <code>glm(as.formula("(survtime <= followup) & (censord == 1) ~ covariates"), data = data)</code> . Alternatively, a list containing the following elements: formula: a <code>formula()</code> in the form <code>~ covariates</code> ; coefficients: a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in formula and colnames of data.
theta	The θ value used to specify the odds ratio e^θ under the alternative hypothesis. If $\theta \geq 0$, the average run length for the upper one-sided Bernoulli CUSUM will be determined. If $\theta < 0$, the average run length for the lower one-sided CUSUM will be determined. Note that $p_1 = \frac{p_0 e^\theta}{1 - p_0 + p_0 e^\theta}.$
theta_true	The true log odds ratio θ , describing the true increase in failure rate from the null-hypothesis. Default = <code>log(1)</code> , indicating no increase in failure rate.
p0	The baseline failure probability at <code>entrytime + followup</code> for individuals.
p1	The alternative hypothesis failure probability at <code>entrytime + followup</code> for individuals.
smooth_prob	Should the probability distribution of failure under the null distribution be smoothed? Useful for small samples. Can only be TRUE when <code>glmmod</code> is supplied. Default = FALSE.
exact	Should the cdf be determined exactly (TRUE), or approximately (FALSE)? The approximation works well for large x, and can cut computation time significantly. Default = TRUE.

Details

Let K denote the run length of the Bernoulli CUSUM with control limit h , then this function can be used to evaluate $P(K \leq x)$.

The formula on page 543 of Brook & Evans (1972) is used if `exact = TRUE`. When `exact = FALSE`, formula (3.9) on page 545 is used instead, approximating the transition matrix using its Jordan canonical form. This can save computation time considerably, but is not appropriate for small values of x .

Value

A list containing:

- Fr_0 : A numeric value indicating the probability of the run length being smaller than x .
- Fr : A data frame containing the cumulative distribution function of the run length depending on the state in which the process starts ($E_0, E_1, \dots, E_{n_{grid}-1}$)
 $start_val$: Starting value of the CUSUM, corresponding to the discretized state spaces E_i ;
 $P(K \leq x)$: Value of the cdf at x for the CUSUM with initial value $start_val$;
- R : A transition probability matrix containing the transition probabilities between states E_0, \dots, E_{t-1} .
 $R_{i,j}$ is the transition probability from state i to state j .

The value of ARL_0 will be printed to the console.

References

Brook, D., & Evans, D. A. (1972). An Approach to the Probability Distribution of CUSUM Run Length. *Biometrika*, 59(3), 539-549. doi:10.2307/2334805

Steiner, S. H., Cook, R. J., Farewell, V. T., & Treasure, T. (2000). Monitoring surgical performance using risk-adjusted cumulative sum charts. *Biostatistics*, 1(4), 441-452. doi:10.1093/biostatistics/1.4.441

Examples

```
#Determine a risk-adjustment model using a generalized linear model.
#Outcome (failure within 100 days) is regressed on the available covariates:
glmmodber <- glm((survtime <= 100) & (censord == 1) ~ age + sex + BMI,
                 data = surgerydat, family = binomial(link = "logit"))
#Determine probability of run length being less than 600
prob600 <- bernoulli_RL_cdf(h = 2.5, x = 600, n_grid = 200, glmmod = glmmodber, theta = log(2))
```

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 <code>parameter_assist()</code> .
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 $\theta \geq 0$, 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 <code>coxph()</code> . Suggested: <code>coxph(Surv(survtime, censorid) ~ covariates, data = data)</code> . Alternatively, a list with the following elements: formula: a <code>formula()</code> in the form <code>~ covariates</code> ; 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 <code>coxphmod</code> .
cbaseh	A function that returns the unadjusted cumulative baseline hazard $H_0(t)$. If <code>cbaseh</code> is missing but <code>coxphmod</code> has been specified as a survival object, this baseline hazard rate will be determined using the provided <code>coxphmod</code> .
inv_cbaseh	(optional): A function that returns the unadjusted inverse cumulative baseline hazard $H_0^{-1}(t)$. If <code>inv_cbaseh</code> is missing, it will be determined from <code>cbaseh</code> 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 <code>parameter_assist()</code>

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 α 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

- | | |
|-----------------------|--|
| <code>data</code> | A <code>data.frame</code> with rows representing subjects and the following named columns:
<code>entrytime</code> : time of entry into study (numeric);
<code>survtime</code> : time from entry until event (numeric);
<code>ensorid</code> : censoring indicator (0 = right censored, 1 = observed), (integer).
and optionally additional covariates used for risk-adjustment. |
| <code>theta</code> | The expected log-hazard ratio θ under the alternative hypothesis. If $\theta \geq 0$, 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). |
| <code>coxphmod</code> | A Cox proportional hazards regression model as produced by the function <code>coxph()</code> .
Suggested:
<code>coxph(Surv(survtime, censorid) ~ covariates, data = data)</code> .
Alternatively, a list with the following elements:
<code>formula</code> : a <code>formula()</code> in the form <code>~ covariates</code> ;
<code>coefficients</code> : a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in <code>formula</code> and <code>colnames</code> of <code>data</code> . |
| <code>cbaseh</code> | A function that returns the unadjusted cumulative baseline hazard $H_0(t)$. If <code>cbaseh</code> is missing but <code>coxphmod</code> has been specified as a survival object, this baseline hazard rate will be determined using the provided <code>coxphmod</code> . |

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 <code>parameter_assist()</code>

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

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

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: [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 <- 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)
```

breast

Survival after breast cancer surgery

Description

Data about patient survival after their breast cancer surgery procedure performed at one of the 15 units participating in a cancer treatment study. The data is based on a trial performed at the European Organisation for Research and Treatment of Cancer (EORTC).

Usage

```
breast
```

Format

A data.frame with 2663 rows and 11 variables:

entrytime Chronological time of entry of patient into study/time of surgery (numeric)

survtime Time from entry until failure of patient (numeric)

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

unit Unit number at which patient received treatment (integer)

var1-7 Covariates associated with patient (factor)

Source

Based on trial data from the European Organisation for Research and Treatment of Cancer, <https://www.eortc.org/>

Examples

```
#Determine the estimated arrival rate for all units in the data
arrival_rate(breast)

#Plot Quality Control charts for unit 11 in the study
library(survival)
phmodbreast <- coxph(Surv(survtime, censorid) ~ . - entrytime - unit ,
  data = breast)
glmmodbreast <- glm((survtime <= 36) & (censorid == 1) ~ . - entrytime - unit,
  data = breast, family = binomial(link = "logit"))

par(mfrow = c(1, 3))
p1 <- plot(cgr_cusum(data = subset(breast, unit == 11), coxphmod = phmodbreast)) +
  ggtitle("CGR-CUSUM")
p2 <- plot(bk_cusum(data = subset(breast, unit == 11), coxphmod = phmodbreast,
  theta = log(2))) + ggtitle("BK-CUSUM")
p3 <- plot(bernoulli_cusum(data = subset(breast, unit == 11), followup = 36,
  glmmod = glmmodbreast, theta = log(2))) + ggtitle("Bernoulli CUSUM")
p4 <- plot(funnel_plot(data = breast, glmmod = glmmodbreast, followup = 36 )) +
  ggtitle("Funnel plot")

if(require("gridExtra")){
  grid.arrange(p1, p2, p3, p4, nrow = 2)
}
```

calc_risk	<i>Calculate the Cox Proportional hazards relative risk associated with the covariates of subjects</i>
-----------	--

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 <code>coxph()</code> or a list containing: formula: a <code>formula()</code> in the form <code>~ covariates</code> ; 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 = log(6), 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 <code>parameter_assist()</code> .
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 <code>coxph()</code> . Suggested: <code>coxph(Surv(survtime, censorid) ~ covariates, data = baseline_data)</code> . Alternatively, a list with the following elements: formula: a <code>formula()</code> in the form <code>~ covariates</code> ; 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 <code>coxphmod</code> .
cbaseh	(optional): A function that returns the unadjusted cumulative baseline hazard $H_0(t)$. If <code>cbaseh</code> is missing but <code>coxphmod</code> has been specified as a survival object, this baseline hazard rate will be determined using the provided <code>coxphmod</code> .

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 $\log(6)$, meaning that at most a 6 times increase/decrease in the odds/hazard ratio is expected.
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. (2022). 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 = log(6))
```

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 <code>coxph()</code> . Suggested: <code>coxph(Surv(survtime, censorid) ~ covariates, data = data)</code> . Alternatively, a list with the following elements: formula: a <code>formula()</code> in the form <code>~ covariates</code> ; 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 <code>entrytime + C</code> .
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 <code>detectCores()</code> to check how many cores are available on your computer.
cmethod	Method to calculate chart values. One of the following: <ul style="list-style-type: none"> "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 <code>parameter_assist()</code>

`maxtheta` (optional): Maximum value the maximum likelihood estimate for parameter θ can take. If `detection = "lower"`, `-abs(theta)` will be the minimum value the maximum likelihood estimate for parameter θ can take. Default is `log(6)`, meaning that at most a 6 times increase/decrease in the odds/hazard ratio is expected.

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 .

When `maxtheta` is specified, the maximum likelihood estimate of θ is restricted to either `abs(maxtheta)` (upper sided CGR-CUSUM) or `-abs(maxtheta)` (lower sided CGR-CUSUM). Choosing this value smaller leads to smaller control limits and therefore quicker detection times, but can cause delays in detection if the true increase in failure rate is larger than the cut-off. The default of expecting at most a 6 times increase in hazard/odds ratio can therefore be the wrong choice for your intended application area. If unsure, the most conservative choice is to take `maxtheta = Inf`.

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 $e^{\theta t}$;
 - `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

References

Gomon, D. and Putter, H. and Nelissen, R. G. H. H. and van der Pas, S. (2022), CGR-CUSUM: A Continuous time Generalized Rapid Response Cumulative Sum chart, *Biostatistics* doi:10.1093/biostatistics/kxac041

Gomon D. and Fiocco M and Putter H and Signorelli M, SURvival Control Chart ESTimation Software in R: the success Package, *The R Journal*, vol. 15, no. 4, pp. 270–291, Dec. 2023, doi:10.32614/rj2023095

See Also

[plot.cgrcusum](#), [runlength.cgrcusum](#)

Other quality control charts: [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, mu = log(1))

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:

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

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

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

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

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

Value

Value of specified function at time t .

extract_hazard	<i>Extract (inverse) cumulative baseline hazard from Cox PH model</i>
----------------	---

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()`.

Details

The baseline hazard is extracted from the `coxph()` call using the `basehaz()` function. The baseline hazard function is then smoothed using `approxfun()` to obtain the linear interpolant. If required, the inverse baseline hazard is determined using root linear interpolation. For this, a function written by Zheyuan Li (see references) is used.

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

References

[Zheyuan Li: How to estimate x value from y value input after approxfun in R?](#) (accessed: 09/10/2023)

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, see Spiegelhalter (2005).

Usage

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

Arguments

<code>data</code>	A <code>data.frame</code> with rows representing subjects and the following named columns: <code>entrytime</code> : time of entry into study (numeric); <code>survtime</code> : time from entry until event (numeric); <code>ensorid</code> : censoring indicator (0 = right censored, 1 = observed), (integer); <code>unit</code> : integer or character indicating which unit (f.e. hospital) the observation belongs to. and optionally additional covariates used for risk-adjustment.
<code>ctime</code>	Construction time at which the funnel plot should be determined. Maximum possible time used when not specified.
<code>p0</code>	The baseline failure probability at <code>entrytime + followup</code> for individuals. If not specified, average failure proportion over whole data is used instead.
<code>glmmod</code>	A generalized linear regression model as produced by the function <code>glm()</code> . Recommended: <code>glm(as.formula("(survtime <= followup) & (ensorid == 1) ~ covariates"), data = data)</code> . Alternatively, a list with the following elements: <code>formula</code> : a <code>formula()</code> in the form <code>~ covariates</code> ; <code>coefficients</code> : a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in <code>formula</code> and <code>colnames</code> of <code>data</code> .
<code>followup</code>	The followup time for every individual. At what time after subject entry do we consider the outcome?
<code>predlim</code>	A vector of confidence levels for the prediction limits of interest. Default is <code>c(0.95, 0.99)</code> .
<code>assist</code>	(optional): Output of the function <code>parameter_assist()</code>

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;
 - `predlimels`: 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
- predlim: specified confidence level(s)
- p0: (Estimated) baseline failure probability

Author(s)

Daniel Gomon

References

Spiegelhalter D. J. (2005). Funnel plots for comparing institutional performance. *Statistics in medicine*, 24(8), 1185-1202. doi:10.1002/sim.1970

See Also

[plot.funnelplot](#), [summary.funnelplot](#)

Other quality control charts: [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 <code>inv_cbaseh</code> 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 <code>coxph()</code> or a list containing: formula: a <code>formula()</code> in the form <code>~ covariates</code> ; coefficients: a named vector specifying risk adjustment coefficients for covariates. Names must be the same as in formula and colnames of data. If both <code>cbaseh</code> and <code>inv_cbaseh</code> are missing, the hazard rate will be determined from <code>coxphmod</code> .
baseline_data	(optional): A <code>data.frame</code> used for covariate resampling with rows representing subjects and columns containing covariates to use for risk-adjustment. Should only be specified in combination with <code>coxphmod</code> .
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;
`ensorid`: 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 ψ 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	<i>Generate survival times</i>
----------------	--------------------------------

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

<code>invchaz</code>	the inverse cumulative (baseline) hazard rate to be used for generating survival times. Must take vector inputs!
<code>mu</code>	the true hazard ratio used to generate survival times.
<code>data</code>	an integer number of survival times to generate or (in combination with <code>coxphmod</code>): a <code>data.frame</code> containing subject covariates in named columns.
<code>coxphmod</code>	(optional) a cox proportional hazards regression model as produced by the function <code>coxph()</code> . Standard practice: <code>coxph(Surv(survtime, censorid) ~ covariates, data = data)</code> . Alternatively, a list with: <ul style="list-style-type: none"> • <code>\$formula</code> (<code>~ covariates</code>) • <code>\$coefficients</code> (named vector specifying risk adjustment coefficients for covariates - names must be the same as in <code>\$formula</code> and <code>colnames</code> of <code>data</code>).

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	<i>Plot a list of CUSUM charts (interactive)</i>
------------------	--

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 "Set2" color set from <code>palette()</code> 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, maxtheta = log(6))
```

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/odds ratio. In other words: the logarithm of the expected increase in the odds/hazard ratio. 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)
maxtheta	Maximum value the maximum likelihood estimate for parameter θ can take. If detection = "lower", $-\text{abs}(\theta)$ will be the minimum value the maximum likelihood estimate for parameter θ can take. Default is log(6), meaning that at most a 6 times increase/decrease in the odds/hazard ratio is expected.

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.
- `theta`: Expected increase in the odds/hazard ratio.
- `psi`: Estimated Poisson arrival rate in data.
- `time`: Time frame over which to restrict type I error.
- `alpha`: Desired level of type I error for control limit determination.
- `maxtheta`: Maximum expected increase/decrease in the odds/hazard ratio.

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.cgrcsum

Plot a quality control chart

Description

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

Usage

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

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

## S3 method for class 'funnelplot'
plot(x, percentage = TRUE, unit_label = TRUE,
```

```
    label_size = 3, col_fill = "blue", ...)  
  
## 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.
unit_label	Should unit labels be displayed next to detected units in the funnel plot? Default is TRUE.
label_size	Size of the labels when unit_label is TRUE. Default is 3.
col_fill	Single fill colour for the prediction intervals in the funnel plot. In any format that <code>col2rgb</code> accepts. Default is "blue".

Value

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

Methods (by class)

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

Author(s)

Daniel Gomon

See Also

[cgr_cusum](#), [bk_cusum](#), [bernoulli_cusum](#), [funnel_plot](#)

runlength	<i>Determine run length of a CUSUM chart</i>
-----------	--

Description

This function can be used to calculate the run length of a 'cgrcum', 'bkcum' or 'bercum' chart when using control limit h

Usage

```
runlength(chart, h)

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

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

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

Arguments

chart	A cgrcum, bkcum or bercum 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)

- `runlength(cgrcum)`: determines runlength of cgrcum object
- `runlength(bkcum)`: determines runlength of bkcum object
- `runlength(bercum)`: determines runlength of bercum 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)

- summary(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 $\text{age} = 0.003$, $\text{BMI} = 0.02$ and $\text{sexmale} = 0.2$ and exponential baseline hazard rate $h_0(t, \lambda = 0.01)e^{\mu}$. The increase in hazard rate is sampled from a normal distribution for all hospitals:

- $\theta \sim N(\log(1), sd = 0.4)$

This means that the average failure rate of hospitals in the data set should be baseline ($\theta = 0$), with some hospitals experiencing higher and lower failure rates. True failure rate can be found in the column `exptheta`.

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)

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

unit Hospital number at which patient received treatment (integer)

exptheta 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)

weib_hazards	<i>Weibull hazard, cumulative hazard and inverse cumulative hazard</i>
--------------	--

Description

Functions which return the hazard, cumulative hazard and inverse cumulative hazard at time t for a Weibull distribution with shape parameter λ , scale parameter θ and true hazard ratio μ .

Usage

```
haz_weib(t, lambda, theta, mu = log(1))
```

```
chaz_weib(t, lambda, theta, mu = log(1))
```

```
inv_chaz_weib(t, lambda, theta, mu = log(1))
```

Arguments

t	time of evaluation.
lambda	shape parameter λ
theta	scale parameter θ
mu	(optional) true excess hazard rate μ .

Details

The hazard function of a Weibull distribution is given by:

$$h(t|\lambda, \theta, \mu) = \frac{\lambda}{\theta} \left(\frac{t}{\theta}\right)^{\lambda-1} e^{\mu}$$

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

$$H(t|\lambda, \theta, \mu) = \left(\frac{t}{\theta}\right)^{\lambda} e^{\mu}$$

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

$$H^{-1}(t|\lambda, \theta, \mu) = \theta \left(\frac{t}{e^{\mu}}\right)^{1/\lambda}$$

Value

Value of specified function at time t .

Index

- * **average run length**
 - bernoulli_ARL, 3
- * **control limit simulation**
 - bernoulli_control_limit, 6
 - bk_control_limit, 12
 - cgr_control_limit, 20
- * **datasets**
 - breast, 17
 - surgerydat, 40
- * **quality control charts**
 - bk_cusum, 15
 - cgr_cusum, 22
 - funnel_plot, 27
- * **utils**
 - gen_arriv_times, 31
- approxfun(), 27
- arrival_rate, 2
- basehaz(), 27
- bernoulli_ARL, 3
- bernoulli_control_limit, 5, 6, 14, 22
- bernoulli_cusum, 5, 6, 8, 8, 34, 37
- bernoulli_cusum(), 3, 36
- bernoulli_RL_cdf, 10
- bk_control_limit, 8, 12, 22
- bk_cusum, 12, 14, 15, 25, 29, 34, 37
- breast, 17
- calc_risk, 19
- cgr_control_limit, 8, 14, 20
- cgr_cusum, 17, 20, 22, 22, 29, 34, 37
- cgr_cusum(), 36
- chaz_exp (exp_hazards), 25
- chaz_weib (weib_hazards), 41
- col2rgb, 37
- coxph, 27
- coxph(), 13, 15, 19, 20, 23, 26, 27, 30, 32, 36
- detectCores(), 21, 23
- exp_hazards, 25
- extract_hazard, 26
- formula(), 4, 7, 9, 11, 13, 15, 19, 20, 23, 28, 30
- funnel_plot, 17, 25, 27, 34, 37, 39
- funnel_plot(), 36
- gen_arriv_times, 31
- gen_surv_times, 32
- gen_surv_times(), 30
- generate_units, 29
- glm(), 3, 6, 9–11, 28, 36
- haz_exp (exp_hazards), 25
- haz_weib (weib_hazards), 41
- interactive_plot, 33
- inv_chaz_exp (exp_hazards), 25
- inv_chaz_weib (weib_hazards), 41
- palette(), 33
- parameter_assist, 34
- parameter_assist(), 6, 7, 9, 13, 16, 20, 21, 23, 28
- plot.bercusum, 10
- plot.bercusum (plot.cgrcusum), 36
- plot.bkcusum, 17
- plot.bkcusum (plot.cgrcusum), 36
- plot.cgrcusum, 25, 36
- plot.funnelplot, 29
- plot.funnelplot (plot.cgrcusum), 36
- runlength, 38
- runlength.bercusum, 10
- runlength.bkcusum, 17
- runlength.cgrcusum, 25
- summary.funnelplot, 29, 39
- surgerydat, 40

`uniroot()`, [30](#)

`weib_hazards`, [41](#)