Package: exceedProb (via r-universe)

October 8, 2024

Version 0.0.2

Date 2019-12-08

Title Confidence Intervals for Exceedance Probability

Description Computes confidence intervals for the exceedance probability of normally distributed estimators. Currently supports general linear models, with a beta function for Cox models. Please see Segal (2019) <doi:10.1080/00031305.2019.1678521> for more information.

Depends R (>= 3.1)

Imports Rcpp (>= 1.0.2), survival (>= 2.44), boot (>= 1.3)

LinkingTo Rcpp, BH

Author Brian D. Segal [aut, cre]

Maintainer Brian D. Segal <segal.brian@gmail.com>

License GPL (>= 3)

URL https://github.com/bdsegal/exceedProb

LazyData true

RoxygenNote 6.1.1

Repository https://bdsegal.r-universe.dev

RemoteUrl https://github.com/bdsegal/exceedprob

RemoteRef HEAD

RemoteSha 7c5639b7b7f1cff852196c6be4e023bd332ed6b3

Contents

CoxEP *Point estimate of the exceedance probability for Cox model parameters*

Description

This function obtains point estimates for the exceedance probability for Cox model parameters.

Usage

```
CoxEP(data, surv_formula, j, cutoff, n, m, lower_tail)
```

Arguments

data	survival data (data.frame)
surv_formula	Survival formula
j	Index of parameter for which the exceedance probability is obtained
cutoff	Cutoff values (scalar or vector)
n	Number of observations in the original study
m	Number of observations in the replication study (defaults to n if NULL)
lower_tail	If TRUE, reports lower tail probabilities

Value

point (scalar or vector) Point estimate of exceedance probability

exceedProb *Confidence intervals for the exceedance probability*

Description

This function obtains confidence intervals for exceedance probability

Usage

```
exceedProb(cutoff, theta_hat, sd_hat, alpha, d, n, m, interval = c(-100,
  100), lower_tail = FALSE)
```

Arguments

cutoff	Cutoff values (scalar or vector)
theta_hat	Point estimate for the parameter of interest
sd_hat	Estimated standard deviation for the parameter of interest (Note: not the standard error)
alpha	Significance level
d	Number of parameters in the general linear model
n	Number of observations in the initial study
m	Number of observations in the replication study
interval	Interval within which to search for roots
lower_tail	If TRUE, reports lower tail probabilities

Value

ep Exceedance probability with confidence intervals

Examples

```
library(exceedProb)

# Sample mean ----------------------------------------------------
n <- 100
x <- rnorm(n = n)

theta_hat <- mean(x)
sd_hat <- sd(x)

cutoff <- seq(from = theta_hat - 0.5, to = theta_hat + 0.5, by = 0.1)

exceedProb(cutoff = cutoff,
           theta_hat = theta_hat,
           sd_hat = sd_hat,
           alpha = 0.05,
           d = 1,
           n = n,
           m = n)

# Linear regression ----------------------------------------------
n <- 100
beta <- c(1, 2)
x <-runif(n = n, min = 0, max = 10)
y <- rnorm(n = n, mean = cbind(1, x) %*% beta, sd = 1)

j <- 2
fit <- lm(y ~ x)
theta_hat <- coef(fit)[j]
sd_hat <- sqrt(n * vcov(fit)[j, j])

cutoff <- seq(from = theta_hat - 0.5, to = theta_hat + 0.5, by = 0.1)
```

```
exceedProb(cutoff = cutoff,
           theta_hat = theta_hat,
           sd_hat = sd_hat,
           alpha = 0.05,
           d = length(beta),
           n = n,
           m = n)
```

`exceedProbCoxBoot` *Bootstrap confidence intervals for the exceedance probability of Cox model parameters*

Description

This function obtains nonparametric bootstrap percentile confidence for Cox model parameters. Beta version.

Usage

```
exceedProbCoxBoot(data, cox_fit, j, alpha, R, cutoff = NULL, m = NULL,
  lower_tail = FALSE, sim = "model")
```

Arguments

`data`	survival data (data.frame)
`cox_fit`	(coxph.object) A fitted Cox model
`j`	Index of parameter for which the exceedance probability is obtained
`alpha`	Significance level
`R`	Number of bootstrap resamples
`cutoff`	Cutoff values (scalar or vector if supplied, otherwise set to +/- 0.5 of theta_hat)
`m`	Number of observations in the replication study (defaults to n if NULL)
`lower_tail`	If TRUE, reports lower tail probabilities; otherwise reports upper tail probabilities
`sim`	type of simulation, input to boot::censboot

Value

ep Exceedance probability with confidence intervals

Examples

```
library(exceedProb)
library(survival)

# Cox model ----------------------------------------------------

# Simulate exponential data
n <- 50
baseline_hazard <- 1
theta <- 0.4
p_censor <- 0.3
prop_tx <- 0.5

tx_indicator = rbinom(n = n, size = 1, prob = prop_tx)
event_rate <- baseline_hazard * exp(theta * tx_indicator)
censor_rate <- event_rate * p_censor / (1 - p_censor)

event_time <- rexp(n = n, rate = event_rate)
censor_time <- rexp(n = n, rate = censor_rate)
time <- pmin(event_time, censor_time)
event <- time == event_time
surv_data = data.frame(time = time, event = event, group = tx_indicator)

# Fit Cox model and get bootstrap percentile confidence intervals for the exceedance probability
# with model-based resampling (see documentation for boot::censboot)
cox_fit <- coxph(Surv(time, event) ~ group, data = surv_data)
ep <- exceedProbCoxBoot(data = surv_data,
                        cox_fit = cox_fit,
                        j = 1,
                        alpha = 0.05,
                        R = 500)

# Plot results
with(ep, plot(cutoff, point, type = "l"))
with(ep, lines(cutoff, lower, lty = 2))
with(ep, lines(cutoff, upper, lty = 2))
```

`getDeltaCI` *Confidence intervals for noncentrality parameter of t-distribution*

Description

This function obtains confidence intervals for the non-centrality parameter of a t-distribution.

Usage

```
getDeltaCI(test_stat, alpha, d, n, interval)
```

Arguments

test_stat	Test statistics
alpha	Significance level
d	Number of parameters in general linear model
n	Number of observations in initial study
interval	Interval within which to search for roots

Value

ep Exceedance probability with confidence intervals (vector if cutoff is scalar and matrix otherwise)

---

pnct *t-distribution with Boost*

Description

This function returns the cdf of a noncentral t-distribution. It is more accurate than stats::pt() for large ncp

Usage

```
pnct(x, df, ncp)
```

Arguments

x	Test statistic
df	Degrees of freedom
ncp	Noncentrality parameter

Value

Cumulative probability

tRoot *This function is used to find the root for a t-distribution pivotal quantity*

Description

This function returns the difference between the lower tail probability of a non-central t-distribution and a confidence level q where the t-distribution has df degrees of freedom and non-centrality parameter delta.

Usage

```
tRoot(delta, test_stat, df, conf_level)
```

Arguments

`delta`	Non-centrality parameter
`test_stat`	Test statistic at which to evaluate the t-distribution
`df`	Degrees of freedom
`conf_level`	Confidence level (usually alpha/2 or 1-alpha/2)

Value

dif Difference between t-distribution quantile and confidence level

Index