Package 'hrIPW'

April 13, 2020

Title Hazard Ratio Estimation using Cox Model Weighted by the
Estimated Propensity Score

Version 0.1.3

Description Estimates the log hazard ratio associated with a binary exposure us-
ing a Cox PH model weighted by the propensity score.
Propensity model is estimated using a simple logistic regression.
Variance estimation takes into account the propensity score estimation step with the method pro-
posed by Hajage et al. (2018) <doi:10.1002/bimj.201700330>.
Both the average treatment effect on the overall (ATE) or the treated (ATT) population can be es-
timated.
For the ATE estimation, both unstabilized and stabilized weights can be used.

Depends R (>= 3.3)

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports survival

Suggests RISCA, boot

NeedsCompilation no

Author David Hajage [aut, cre]

Maintainer David Hajage <david.hajage@aphp.fr>

Repository CRAN

Date/Publication 2020-04-13 13:50:02 UTC

R topics documented:

---

hrData   *A simulated data set*

Description

A simulated cohort with 9 confounders (X1 to X9), generated with the algorithm described in Hajage *et al.* (2018) doi: 10.1002/bimj.201700330.

Usage

```
data(hrData)
```

Format

An object of class "data.frame".

References

Closed-form variance estimator for weighted propensity score estimators with survival outcome. Submitted to Statistics in Medicine (2017).

Examples

```
data(hrData, package = "hrIPW")
hrIPW(hrData, time = "time", status = "status", exposure = "Trt",
      variables = paste("X", 1:9, sep = ""), wtype = "ATE-stab")
```

---

hrIPW   *Hazard ratio estimation using Cox model weighted by the propensity*
                                score

Description

This function allows estimating the log hazard ratio associated with a binary exposure using a Cox PH model weighted by the propensity score. Propensity model is estimated using a simple logistic regression. Variance estimation takes into account the propensity score estimation step with the method proposed by Hajage *et al.* (2018) doi: 10.1002/bimj.201700330. Both the average treatment effect on the overall (ATE) or the treated (ATT) population can be estimated. For ATE, both unstabilized and stabilized weights can be used. Ties are handled through the Breslow approximation.

Usage

```
hrIPW(data, time, status, exposure, variables, wtype, alpha = 0.05)
```

Arguments

`data`	The data.frame to be analyzed.
`time`	A character string indicating the name of the follow up times column.
`status`	A character string indicating the name of the failure column (0=right censored, 1=event).
`exposure`	A character string indicating the name of the exposure column (0=unexposed, 1=exposed).
`variables`	A character vector indicating the names of the confounders to be accounted for in the propensity model. Only numeric variables could be handled, so factors, characters and other eligible classes should be expanded into dummy/indicator variables.
`wtype`	A character string indicating the type of weights that should be used ('ATE-unstab' for unstabilized ATE weights, 'ATE-stab' for stabilized ATE weights, or 'ATT' for ATT weights)
`alpha`	A numeric value indicating the (bilateral) alpha level used for the computation of the confidence interval

Value

A list with the following elements:

- coefficient: the log-HR associated with the exposure
- std: the standard error
- ciinf and cisup: lower and upper limits of the (1-alpha) condidence interval
- p.value: the p-value

References

Closed-form variance estimator for weighted propensity score estimators with survival outcome. Submitted to Statistics in Medicine (2017).

Austin PC. Variance estimation when using inverse probability of treatment weighting (IPTW) with survival analysis. Statistics in medicine, 30;35(30):5642-5655, 2016. <doi: 10.1002/sim.7084>

Lin DY, Wei LJ. The Robust Inference for the Cox Proportional Hazards Model. Journal of the American Statistical Association 84(408):1074-1078, 1989. <doi: 10.1080/01621459.1989.10478874>

Examples

```
## Using a simulated cohort
data(hrData, package = "hrIPW")
hrIPW(hrData, time = "time", status = "status", exposure = "Trt",
      variables = paste("X", 1:9, sep = ""), wtype = "ATE-stab")

# Standard error could be compared with the (robust) Lin's standard error
# which does not take into account the propensity score estimation step:
modT <- glm(Trt ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9, data = hrData, family = "binomial")
probT <- predict(modT, type = "response")
```

```
hrData$w <- 1/ifelse(hrData$Trt == 1, probT, 1 - probT)
library(survival)
coxph(Surv(time, status) ~ Trt + cluster(id), data = hrData, method = "breslow", weights = w)

# or with the bootstrap-based standard-error (see Austin 2016):
## Not run:
f.boot <- function(data, i, wtype) {
    df <- data[i, ]
  modT <- glm(Trt ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9, data = df, family = "binomial")
    probT <- predict(modT, type = "response")
    df$w <- 1/ifelse(df$Trt == 1, probT, 1 - probT)

    return(coxph(Surv(time, status) ~ Trt, data = df, weights = w)$coef)
}

library(boot); set.seed(1234)
rcoefs <- boot(data = hrData, statistic = f.boot, R = 500)$t
sd(rcoefs)

## End(Not run)

## Using the DIVAT data base (package IPWsurvival, to be installed)
data(dataDIVAT2, package = "RISCA")
hrIPW(data = dataDIVAT2, time = "times", status = "failures", exposure = "ecd",
      variables = c("age", "hla", "retransplant"), wtype = "ATE-unstab")

# Standard error could be compared with the (robust) Lin's standard error
# which does not take into account the propensity score estimation step:
modT <- glm(ecd ~ age + hla + retransplant, data = dataDIVAT2, family = "binomial")
probT <- predict(modT, type = "response")
dataDIVAT2$w <- 1/ifelse(dataDIVAT2$ecd == 1, probT, 1 - probT)
dataDIVAT2$id <- 1:nrow(dataDIVAT2)
coxph(Surv(times, failures) ~ ecd + cluster(id), data = dataDIVAT2, method = "breslow", weights = w)

# or with the bootstrap-based estimation (see Austin 2016):
## Not run:
f.boot <- function(data, i, wtype) {
    df <- data[i, ]
    modT <- glm(ecd ~ age + hla + retransplant, data = df, family = "binomial")
    probT <- predict(modT, type = "response")
    df$w <- 1/ifelse(df$ecd == 1, probT, 1 - probT)

    return(coxph(Surv(times, failures) ~ ecd, data = df, weights = w)$coef)
}

set.seed(123)
rcoefs <- boot(data = dataDIVAT2, statistic = f.boot, R = 200)$t
sd(rcoefs)

## End(Not run)
```

Index