

# Package ‘hrcomprisk’

January 21, 2020

**Type** Package

**Title** Nonparametric Assessment Between Competing Risks Hazard Ratios

**Version** 0.1.1

**Author** Daniel Antiporta <dantiporta@jhu.edu>;  
Matthew Matheson <mmathes4@jhu.edu>;  
Derek Ng <dng@jhu.edu>;  
Alvaro Munoz <amunoz@jhu.edu>

**Maintainer** Daniel Antiporta <dantiporta@jhu.edu>

**Description** Nonparametric cumulative-incidence based estimation of the ratios of sub-hazard ratios to cause-specific hazard ratios using the approach from Ng et al. (2020).

**Depends** R (>= 3.0.0)

**Imports** survival

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.0.2

**URL** <https://github.com/AntiportaD/hrcomprisk>

**BugReports** <https://github.com/AntiportaD/hrcomprisk/issues>

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-01-21 20:10:02 UTC

## R topics documented:

bootCRCumInc . . . . .	2
CRCumInc . . . . .	3
datcheck . . . . .	5

dat_ckid . . . . .	5
hrcomprisk . . . . .	6
npcrest . . . . .	8
plotCIF . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

bootCRCumInc	<i>Bootstrap for Ratios of Hazard Ratios</i>
--------------	--

---

## Description

Bootstrap 95% Confidence Intervals limits for estimated Ratios of sHR/csHR.

## Usage

```
bootCRCumInc(
  df,
  exit,
  event,
  exposure,
  entry = NULL,
  weights = NULL,
  ipwvars = NULL,
  rep = 0,
  print.attr = T,
  seed = 54321
)
```

## Arguments

df	A data frame containing, at a minimum, exit, event, and exposure.
exit	Name of the column in df containing times of event or censoring.
event	Name of the column in df containing codes for censoring (0) and event types (1-4). Analysis of more than 4 competing events is not supported by this function.
exposure	Name of the column in df containing a binary (0/1) exposure variable for stratification.
entry	Name of the column in df containing late entry times.
weights	Name of the column in df containing user-supplied weights. If ipwvars is utilized, this argument is ignored.
ipwvars	A vector of names of columns in 'df' containing predictor variables for building a propensity score model for exposure and creating standardized inverse probability weights using this model. Overrides the weights argument.
rep	Number of replicates for bootstrapping if confidence intervals for the sHR/csHR estimate are desired. See more details on bootstrapping below.
print.attr	A logical indicator for whether results should be returned in console.
seed	A seed number start for the bootstrap estimation.

**Value**

A data frame with the 95% confidence interval limits (upper and lower) for Sub-hazard ratio/Cause-specific hazard ratio for each event:

**R1.lower** Lower limit of the 95%CI of the Sub-hazard ratio/Cause-specific hazard ratio for event 1 at time t

**R1.upper** Upper limit of the 95%CI of the Sub-hazard ratio/Cause-specific hazard ratio for event 1 at time t

**R2.lower** Lower limit of the 95%CI of the Sub-hazard ratio/Cause-specific hazard ratio for event 2 at time t

**R2.upper** Upper limit of the 95%CI of the Sub-hazard ratio/Cause-specific hazard ratio for event 2 at time t

**Examples**

```
#data from the package
data <- hrcomprisk::dat_ckid
#Obtain the 95%CI by bootstrapping
ciCIF<-bootCRCumInc(df=data, exit=exit, event=event, exposure=b1nb0, rep=10, print.attr=TRUE)
```

---

 CRCumInc

---

*Estimation of Cumulative Incidence of competing events*


---

**Description**

Estimation of Cumulative Incidence Functions (CIF) of competing events.

This function is based on the CIF estimated by the survival package.

**Usage**

```
CRCumInc(
  df,
  time,
  event,
  exposed,
  entry = NULL,
  weights = NULL,
  ipwvars = NULL,
  print.attr = T
)
```

**Arguments**

<code>df</code>	A data frame containing, at a minimum, exit, event, and exposure.
<code>time</code>	Name of the column in <code>df</code> containing times of event or censoring.
<code>event</code>	Name of the column in <code>df</code> containing codes for censoring (0) and event types (1-4). Analysis of more than 4 competing events is not supported by this function.
<code>exposed</code>	Name of the column in <code>df</code> containing a binary (0/1) exposure variable for stratification.
<code>entry</code>	Name of the column in <code>df</code> containing late entry times.
<code>weights</code>	Name of the column in <code>df</code> containing user-supplied weights. If <code>ipwvars</code> is utilized, this argument is ignored.
<code>ipwvars</code>	A vector of names of columns in ‘ <code>df</code> ’ containing predictor variables for building a propensity score model for exposure and creating standardized inverse probability weights using this model. Overrides the <code>weights</code> argument.
<code>print.attr</code>	A logical indicator for whether results should be returned in console.

**Value**

A data frame with the following columns:

**event** Type of event that occurs at the given time.

**exposure** Exposure group in which the event happens.

**time** Time of the event.

**CIoinc\_comp** Value of the unexposed (denoted by “o”) composite cumulative incidence at the given time.

**CIxinc\_comp** Value of the exposed (denoted by “x”) composite cumulative incidence at the given time.

**CIoinc\_1** Value of the unexposed cumulative incidence of event 1 at the given time.

**CIxinc\_1** Value of the exposed cumulative incidence of event 1 at the given time.

**R\_1** Sub-hazard ratio/Cause-specific hazard ratio for event 1.

**R\_2** Sub-hazard ratio/Cause-specific hazard ratio for event 2.

**Examples**

```
#data from the package
data <- hrcomprisk::dat_ckid
#Estimate the Cumulative Incidence Functions and Ratios of sHR and csHR
mydat.CIF<-CRCumInc(df=data, time=exit, event=event, exposed=b1nb0, print.attr=TRUE)
```

---

datcheck	<i>Data Check for Wrapper Function</i>
----------	--

---

**Description**

A function to ensure that the data frame fulfills the relevant variable content and type requirements.

**Usage**

```
datcheck(df, qexit, qevent, qexposure, qentry, qweights, qipwvars, eoi = -1)
```

**Arguments**

df	A data frame containing, at a minimum, exit, event, and exposure.
qexit	Name of the column in df containing times of event or censoring.
qevent	Name of the column in df containing codes for censoring (0) and event types (1-4). Analysis of more than 4 competing events is not supported by this function.
qexposure	Name of the column in df containing a binary (0/1) exposure variable for stratification.
qentry	Name of the column in df containing late entry times.
qweights	Name of the column in df containing user-supplied weights. If ipwvars is utilized, this argument is ignored.
qipwvars	A vector of names of columns in 'df' containing predictor variables for building a propensity score model for exposure and creating standardized inverse probability weights using this model. Overrides the weights argument.
eoi	Event number for the event of interest, useful when more than two events exist. If utilized, only two cumulative incidence curves will be plotted: one for the event of interest, and one for the composite of all competing events. Each event will still have its SHR/cSHR ratio plotted.

**Value**

Check dataset

---

dat_ckid	<i>CKID dataset</i>
----------	---------------------

---

**Description**

A dataset containing time, socioeconomic and outcome variables of 626 subjects from the Chronic Kidney Disease in Children (CKiD) Study.

**Usage**

dat\_ckid

**Format**

A data frame with 626 rows and 13 variables:

**b1nb0** Binary indicator for race: black=1, non-black=0

**entry** Years since onset of chronic kidney disease at entry into study

**event** Renal replacement therapy indicator: 0=none, 1=dialysis, 2=transplant

**exit** Years since onset of chronic kidney disease at event/censoring time

**foodassist** Binary indicator for use of food assistance

**inckd** Years in study (=exit-entry)

**incomegt75** Household income > \$75,000 per year

**incomelt30** Household income < \$30,000 per year

**lps** Binary indicator of low birth weight, premature birth, or small for gestational age

**male1fe0** Binary indicator for sex: male=1, female=0

**matedultcoll** Maternal education less than college

**privatemd** Binary indicator for private doctor

**public** Binary indicator for public insurances

**Source**

<https://statepi.jhsph.edu/ckid/ckid.html>

---

hrcomprisk

*Nonparametric Assessment Between Competing Risks Hazards*

---

**Description**

Estimate nonparametric cumulative-incidence based estimation of the ratios of sub-hazard ratios to cause-specific hazard ratios from Ng, Antiporta, Matheson and Muñoz (2019)[1] to compare sub-hazard ratio (a la Fine and Gray; sHR) and cause-specific hazard ratio (csHR) approaches.

**Details**

While doing either analysis individually involves parametric or semi-parametric estimation, because of the fact that the derivatives of the cumulative incidences involved in the quantities cancel out when their ratio is considered, this ratio can be characterized completely using nonparametric estimates of the event-specific cumulative incidences. This provides a useful diagnostic when both analyses are performed as well as a method for estimating the sub-hazard ratios in a way that is valid and free of tethering assumptions characterized by Muñoz et al. [2].

**1. Bootstrapped confidence intervals for the sHR/csHR quantities**

If a positive number of bootstrap replicates is requested via the `rep` argument, the program will calculate and provide pointwise percentile-based bootstrap confidence intervals for the sHR/csHR ratios. The bootstrapping process uses two loops. In the first loop, `rep` bootstrap samples are taken stratified by exposure (so each sample has the same exposure prevalence as the original data) and all event-specific cumulative incidences are calculated and stored for each of them. In the second loop, for each event time (i.e., each change in any one of the cumulative incidence functions), the 2.5th and 97.5th percentiles of the bootstrap estimates of the `rep` sHR/csHR ratios are stored as the lower and upper confidence limits. These are not directly returned to the user, but are used in the plotting of the sHR/csHR ratios.

## 2. User-supplied vs. program-generated weights

If confidence intervals are not desired, the user can supply a column of weights (e.g. inverse probability weights from a model predicting exposure) which will be used in the estimation of the cumulative incidences and the sHR/csHR ratios derived from them. However, the use of bootstrapping for calculation of confidence intervals as described above necessitates that such a model be refit for each bootstrap sample, generating new weights for new estimates of all quantities. If this is desired, the user should include all the predictor variables as columns in the data frame so that the appropriate model can be fit automatically using the `ipwvars` argument. If this method is used, the program uses a logistic regression model to calculate probability of exposure, stabilizes the resulting weights to the sample size, and winsorizes weights that fall outside  $\pm 4$  standard deviations on the log scale. Using this method can increase computation time as the model must be refit on each bootstrap replicate.

## 3. Use of the nonparametric sHR/csHR ratio for calculation of sHR estimates

Simultaneous estimation of all subhazard ratios and cause-specific hazard ratios is often fraught with problematic results due to incompatible modeling assumptions (e.g., not all ratios can be proportional) and the tethering inherent in subhazard analysis of multiple events as described by Muñoz et al. [2]. Cause-specific hazard ratios are not subject to such tethering constraints, and will be admissible regardless of whether the model is misspecified. As such, the output of this function – valid nonparametric estimates of the sHR/csHR ratio – can be combined with (i.e., multiplied by) cause-specific hazard ratio estimates (e.g., from a proportional or loglinear cause-specific hazards model) to produce subhazard ratio estimates which do not violate the principles of tethering.

## Functions

The `hrcomprisk` package provides 3 main functions and a wrapper function:

- `npcrest` : Main wrapper function.
- `CRCumInc` : Estimation of Cumulative Incidence Functions (CIF) of competing events.
- `plotCIF` : Plot Cumulative Incidence and Ratio of sHR/csHR.
- `bootCRCumInc` : Bootstrap 95

## Author(s)

**Maintainer:** Daniel Antiporta <<dantiporta@jhu.edu>>

### Authors:

- Daniel Antiporta
- Matthew Matheson

- Derek Ng
- Alvaro Muñoz

## References

1. Ng D, Antiporta DA, Matheson M, Munoz A. Nonparametric assessment of differences between competing risks hazard ratios: application to racial differences in pediatric chronic kidney disease progression. *Clinical Epidemiology*, 2020 (in press)
2. Muñoz A, Abraham AG, Matheson M, Wada N. In: Risk Assessment and Evaluation of Predictions. Lee MLT, Gail M, Pfeiffer R, Satten G, Cai T, Gandy A, editor. New York: Springer; 2013. Non-proportionality of hazards in the competing risks framework; pp. 3–22. [[Google Scholar](#)]

## See Also

Useful links:

<https://github.com/AntiportaD/hrcomprisk>

## Examples

```
#data from the package - See fuctions for specific examples
data <- hrcomprisk::dat_ckid
```

---

npcrest

*Nonparametric Estimation of the Ratios of Hazard Ratios*

---

## Description

A comprehensive wrapper function for implementing the competing risks diagnostic of Ng, Antiporta, Matheson and Muñoz (2019) to compare sub-hazard ratio (a la Fine and Gray; sHR) and cause-specific hazard ratio (csHR) approaches. While doing either analysis individually involves parametric or semi-parametric estimation, because of the fact that the derivatives of the cumulative incidences involved in the quantities cancel out when their ratio is considered, this ratio can be characterized completely using nonparametric estimates of the event-specific cumulative incidences. This provides a useful diagnostic when both analyses are performed as well as a method for estimating the sub-hazard ratios in a way that is valid and free of tethering assumptions characterized by Muñoz et al. (2013).

This function calls `datcheck`, `CRCumInc`, `bootCRCumInc` (if confidence intervals are requested), and `plotCIF`, all of which are also included in the `hrcomprisk` package. These functions should generally not be utilized directly.

## Usage

```
npcrest(
  df,
  exit,
  event,
```



```

    exposure,
    entry = NULL,
    weights = NULL,
    ipwvars = NULL,
    maxtime = Inf,
    rep = NULL,
    eoi = -1,
    print.attr = T
)

```

## Arguments

<code>df</code>	A data frame containing, at a minimum, exit, event, and exposure.
<code>exit</code>	Name of the column in <code>df</code> containing times of event or censoring.
<code>event</code>	Name of the column in <code>df</code> containing codes for censoring (0) and event types (1-4). Analysis of more than 4 competing events is not supported by this function.
<code>exposure</code>	Name of the column in <code>df</code> containing a binary (0/1) exposure variable for stratification.
<code>entry</code>	Name of the column in <code>df</code> containing late entry times.
<code>weights</code>	Name of the column in <code>df</code> containing user-supplied weights. If <code>ipwvars</code> is utilized, this argument is ignored.
<code>ipwvars</code>	A vector of names of columns in ‘ <code>df</code> ’ containing predictor variables for building a propensity score model for exposure and creating standardized inverse probability weights using this model. Overrides the <code>weights</code> argument.
<code>maxtime</code>	Largest time to display on the x-axis of all output plots. As data can become sparse and thus more widely variable at times get large, this argument may be used to restrict plots to a range of the data that is discerned to be more accurate and reliable.
<code>rep</code>	Number of replicates for bootstrapping if confidence intervals for the sHR/csHR estimate are desired. See more details on bootstrapping below.
<code>eoi</code>	Event number for the event of interest, useful when more than two events exist. If utilized, only two cumulative incidence curves will be plotted: one for the event of interest, and one for the composite of all competing events. Each event will still have its sHR/csHR ratio plotted.
<code>print.attr</code>	A logical indicator for whether results should be returned in console.

## Details

### 1. Bootstrapped confidence intervals for the sHR/csHR quantities

If a positive number of bootstrap replicates is requested via the `rep` argument, the program will calculate and provide pointwise percentile-based bootstrap confidence intervals for the sHR/csHR ratios. The bootstrapping process uses two loops. In the first loop, `rep` bootstrap samples are taken stratified by exposure (so each sample has the same exposure prevalence as the original data) and all event-specific cumulative incidences are calculated and stored for each of them. In the second loop, for each event time (i.e., each change in any one of the cumulative incidence functions), the 2.5th and 97.5th percentiles of the bootstrap estimates of the `rep` sHR/csHR ratios are stored as the

lower and upper confidence limits. These are not directly returned to the user, but are used in the plotting of the sHR/csHR ratios.

## 2. User-supplied vs. program-generated weights

If confidence intervals are not desired, the user can supply a column of weights (e.g. inverse probability weights from a model predicting exposure) which will be used in the estimation of the cumulative incidences and the sHR/csHR ratios derived from them. However, the use of bootstrapping for calculation of confidence intervals as described above necessitates that such a model be refit for each bootstrap sample, generating new weights for new estimates of all quantities. If this is desired, the user should include all the predictor variables as columns in the data frame so that the appropriate model can be fit automatically using the `ipwvars` argument. If this method is used, the program uses a logistic regression model to calculate probability of exposure, stabilizes the resulting weights to the sample size, and winsorizes weights that fall outside  $\pm 4$  standard deviations on the log scale. Using this method can increase computation time as the model must be refit on each bootstrap replicate.

## 3. Use of the nonparametric sHR/csHR ratio for calculation of sHR estimates

Simultaneous estimation of all subhazard ratios and cause-specific hazard ratios is often fraught with problematic results due to incompatible modeling assumptions (e.g., not all ratios can be proportional) and the tethering inherent in subhazard analysis of multiple events as described by Muñoz et al. (2013). Cause-specific hazard ratios are not subject to such tethering constraints, and will be admissible regardless of whether the model is misspecified. As such, the output of this function – valid nonparametric estimates of the sHR/csHR ratio – can be combined with (i.e., multiplied by) cause-specific hazard ratio estimates (e.g., from a proportional or loglinear cause-specific hazards model) to produce subhazard ratio estimates which do not violate the principles of tethering. See below for an example of how to implement this.

### Value

An object containing the plotted figures (`$plots`) and a data frame (`$cuminc`) with the following columns:

**event** Type of event that occurs at the given time.

**exposure** Exposure group in which the event happens.

**time** Time of the event.

**CIoinc\_comp** Value of the unexposed (denoted by “o”) composite cumulative incidence at the given time.

**CIxinc\_comp** Value of the exposed (denoted by “x”) composite cumulative incidence at the given time.

**CIoinc\_1** Value of the unexposed cumulative incidence of event 1 at the given time.

**CIxinc\_1** Value of the exposed cumulative incidence of event 1 at the given time.

**R\_1** Sub-hazard ratio/Cause-specific hazard ratio for event 1.

**R\_2** Sub-hazard ratio/Cause-specific hazard ratio for event 2.

## References

1. Ng D, Antiporta DA, Matheson M, Munoz A. Nonparametric assessment of differences between competing risks hazard ratios: application to racial differences in pediatric chronic kidney disease progression. (Clinical Epidemiology, 2019-in print)
2. Muñoz A, Abraham AG, Matheson M, Wada N. In: Risk Assessment and Evaluation of Predictions. Lee MLT, Gail M, Pfeiffer R, Satten G, Cai T, Gandy A, editor. New York: Springer; 2013. Non-proportionality of hazards in the competing risks framework; pp. 3–22. [Google Scholar]([https://link.springer.com/chapter/10.1007/978-1-4614-8981-8\\_1](https://link.springer.com/chapter/10.1007/978-1-4614-8981-8_1))

## Examples

```
#data from the package
data <- hrcomprisk::dat_ckid
#Using the wrapper function
npcrest(df=data, exit=exit, event=event, exposure=b1nb0, rep=10, maxtime=20, print.attr=TRUE)
```

---

plotCIF

*Plot Incidence and Ratio of sHR/csHR*

---

## Description

Plot Cumulative Incidence and Ratio of sHR/csHR.

## Usage

```
plotCIF(cifobj, maxtime = Inf, ci = NULL, eoi = -1)
```

## Arguments

cifobj	A dataframe containing the Cumulative Incidence of each competing event by exposure group.
maxtime	Largest time to display on the x-axis of all output plots.
ci	A dataframe containing the 95% CI for each ratio of csHR/sHR.
eoi	Event number for the event of interest, useful when more than two events exist.

## Value

A large list containing 2 figures:

**\$plot1** Plot the cumulative incidence of the composite event and of each event by exposure group.

**\$plot2** Plot the ratio of Sub-hazard ratio and Cause-specific hazard ratio for each event  $i$  ( $R_i$ ).

**Examples**

```
#data from the package
data <- hrcomprisk::dat_ckid
#Estimate the Cumulative Incidence Functions and Ratios of sHR and csHR
mydat.CIF<-CRCumInc(df=data, time=exit, event=event, exposed=b1nb0, print.attr=FALSE)
#Plot the CIs and Ratios estimated
plots<-plotCIF(cifobj=mydat.CIF, maxtime = 20, eoi = 1)
```

# Index

\*Topic **datasets**

dat\_ckid, [5](#)

bootCRCumInc, [2](#), [7](#)

CRCumInc, [3](#), [7](#)

dat\_ckid, [5](#)

datcheck, [5](#)

hrcomprisk, [6](#)

npcrest, [7](#), [8](#)

plotCIF, [7](#), [11](#)