

# Package: nricens (via r-universe)

September 17, 2024

**Type** Package

**Title** NRI for Risk Prediction Models with Time to Event and Binary Response Data

**Version** 1.6

**Date** 2018-5-30

**Author** Eisuke Inoue

**Depends** survival

**Maintainer** Eisuke Inoue <eisuke.inoue@marianna-u.ac.jp>

**Description** Calculating the net reclassification improvement (NRI) for risk prediction models with time to event and binary data.

**License** GPL-2

**NeedsCompilation** no

**Date/Publication** 2018-05-30 10:40:23 UTC

**Repository** <https://eisuke-inoue.r-universe.dev>

**RemoteUrl** <https://github.com/cran/nricens>

**RemoteRef** HEAD

**RemoteSha** 9165fcfb4d715726e4aa1745eb9d412c23e7ce95

## Contents

nricens-package . . . . .	2
categorize . . . . .	3
get.risk.coxph . . . . .	4
get.risk.survreg . . . . .	4
get.surv.km . . . . .	5
get.uppdwn . . . . .	5
get.uppdwn.bin . . . . .	6
nribin . . . . .	6
nribin.count.main . . . . .	9
nricens . . . . .	10
nricens.ipw.main . . . . .	13
nricens.km.main . . . . .	13

**Index****15**

---

nricens-package	<i>R Functions to calculate NRI for comparing time to event and binary response models.</i>
-----------------	---

---

**Description**

This package provides the functions to estimate the net reclassification improvement (NRI) for competing risk prediction models with time to event and binary response data. The NRI for binary response models can be calculated by `nribin`, and that for time to event models can be calculated by `nricens`. The risk category based NRI and the risk difference based NRI are provided by these functions. Users can use several estimators for comparing time to event models. Confidence intervals are calculated by the percentile bootstrap method. In this version, several types of input data are allowed to improve user-friendliness and convenience.

**Details**

Package: nricens  
Type: Package  
Version: 1.6  
Date: 2018-5-30  
License: GPL-2

**Author(s)**

Eisuke Inoue <eisuke.inoue@marianna-u.ac.jp>

**References**

Pencina MJ, D'Agostino RB, Steyerberg EW. Extensions of net reclassification improvement calculations to measure usefulness of new biomarkers. *Statistics in Medicine* 2011.

Uno H, Tian L, Cai T, Kohane IS, Wei LJ. A unified inference procedure for a class of measures to assess improvement in risk prediction systems with survival data, *Statistics in Medicine* 2012.

Hsu CH, Taylor JMG. A robust weighted Kaplan-Meier approach for data with dependent censoring using linear combinations of prognostic covariates, *Statistics in Medicine* 2010.

**Examples**

```
## consider pbc dataset in survival package as an example
library(survival)
dat = pbc[1:312,]
dat$sex = ifelse(dat$sex=='f', 1, 0)
```

```

## prediciting the event of 'death'
time = dat$time
event = ifelse(dat$status==2, 1, 0)

## standard prediction model: age, bilirubin, and albumin
z.std = as.matrix(subset(dat, select = c(age, bili, albumin)))

## new prediction model: age, bilirubin, albumin, and protime
z.new = as.matrix(subset(dat, select = c(age, bili, albumin, protime)))

## fitting cox models
mstd = coxph(Surv(time, event) ~ ., data.frame(time, event, z.std), x=TRUE)
mnew = coxph(Surv(time, event) ~ ., data.frame(time, event, z.new), x=TRUE)

## Calculation of the risk category NRI at 2000 days
nricens mdl.std = mstd, mdl.new = mnew, t0 = 2000, cut = c(0.2, 0.4),
      niter = 10)

## Next, consider binary prediction models
library(survival)
dat = pbc[1:312,]
dat$sex = ifelse(dat$sex=='f', 1, 0)

## subjects censored before 2000 days are excluded
dat = dat[ dat$time > 2000 | (dat$time < 2000 & dat$status == 2), ]

## prediciting the event of 'death' before 2000 days
event = ifelse(dat$time < 2000 & dat$status == 2, 1, 0)

## standard prediction model: age, bilirubin, and albumin
z.std = as.matrix(subset(dat, select = c(age, bili, albumin)))

## new prediction model: age, bilirubin, albumin, and protime
z.new = as.matrix(subset(dat, select = c(age, bili, albumin, protime)))

## glm fit (logistic model)
mstd = glm(event ~ ., binomial(logit), data.frame(event, z.std), x=TRUE)
mnew = glm(event ~ ., binomial(logit), data.frame(event, z.new), x=TRUE)

## Calculation of risk difference NRI
nribin(mdl.std = mstd, mdl.new = mnew, cut = 0.02, niter = 0,
      updown = 'diff')

```

---

categorize

*Categorization of a continuous variable.*


---

### Description

Internally used function to categorize a continuous variable.

**Usage**

```
categorize(dat, threshold)
```

**Arguments**

dat	Numerical vector for a categorization.
threshold	Vector or scalar value(s) to determine the cutoff values for a categorization.

---

get.risk.coxph	<i>Calculate individual risks based on the Cox regression model.</i>
----------------	--

---

**Description**

This function estimates  $Pr\{T < t_0\}$  given covariates under the Cox model.

**Usage**

```
get.risk.coxph mdl, t0)
```

**Arguments**

mdl	coxph object.
t0	Scalar value indicating a time to determine event/non-event ( $t_0$ ).

---

get.risk.survreg	<i>Calculate individual risks based on the parametric survival regression model.</i>
------------------	--

---

**Description**

This function estimates  $Pr\{T < t_0\}$  given covariates under the parametric survival model.

**Usage**

```
get.risk.survreg(mdl, t0)
```

**Arguments**

mdl	survreg object.
t0	Scalar value indicating a time to determine event/non-event ( $t_0$ ).

---

get.surv.km	<i>Calculate a survival probability at a given time by the standard Kaplan-Meier estimator.</i>
-------------	---

---

**Description**

This function estimates  $Pr\{T > t_0\}$  by the Kaplan-Meier estimator.  $t_0$  should be given.

**Usage**

```
get.surv.km(time, event, t0, subs = NULL)
```

**Arguments**

time	Vector of follow up times.
event	Vector of event indicators, 1 for event of interest, 0 for censoring.
t0	Scalar value indicating a time to determine event/non-event ( $t_0$ ).
subs	Vector of logical values to determine which subjects to be used for a calculation. When this option is not specified (subs = NULL), all subjects are used for a calculation.

---

get.uppdown	<i>Determine UP and DOWN for the NRI calculation</i>
-------------	--

---

**Description**

Internally used function to determine subjects who belong to UP and DOWN.

**Usage**

```
get.uppdown(time, event, objs, flag.mdl, flag.prd, flag.rsk, t0, updown,
            cut, get.risk, msg = FALSE)
```

**Arguments**

time	Vector of follow up times.
event	Vector of event indicators, 1 for event of interest, 0 for censoring.
objs	List of data.
flag.mdl, flag.prd, flag.rsk	Logical values to determine the type of calculation based on the input data.
t0	Scalar value indicating a time to determine event/non-event.
updown	Character to specify the method to determine UP and DOWN.

cut	Scalar or vector to specify the cutoff value(s) of predicted risks for determining UP and DOWN. See also cut in nricens or nrabin.
get.risk	R function to calculate individual risks.
msg	Logical value to display computation process. Setting FALSE leads silent execution.

---

get.uppdown.bin	<i>Determine UP and DOWN for the NRI calculation</i>
-----------------	--

---

### Description

Internally used function to determine subjects who belong to UP and DOWN.

### Usage

```
get.uppdown.bin(event, objs, flag.mdl, flag.prd, flag.rsk, updown, cut,
                link, msg = FALSE)
```

### Arguments

event	Vector of event indicators, 1 for event of interest, 0 for non-event.
objs	List of data.
flag.mdl, flag.prd, flag.rsk	Logical values to determine the type of calculation based on the input data.
updown	Character to specify the method to determine UP and DOWN.
cut	Scalar or vector to specify the cutoff value(s) of predicted risks for determining UP and DOWN. See also cut in nricens or nrabin.
link	Character to specify the link function for a glm fitting.
msg	Logical value to display computation process. Setting FALSE leads silent execution.

---

nrabin	<i>NRI for binary models</i>
--------	------------------------------

---

### Description

This function estimates the NRI for competing risk prediction models with binary response variable. glm objects, predictors, and predicted risks can be used as input data for the calculation. The risk category based NRI and the risk difference based NRI can be calculated. The percentile bootstrap method is used for an interval estimation.

**Usage**

```
nrbin(event = NULL, mdl.std = NULL, mdl.new = NULL, z.std = NULL, z.new = NULL,
      p.std = NULL, p.new = NULL, updown = "category", cut = NULL,
      link = "logit", niter = 1000, alpha = 0.05, msg = TRUE)
```

**Arguments**

event	Vector of event indicators, 1 for event of interest, 0 for non-event.
mdl.std, mdl.new	glm objects corresponding to a standard and a new risk prediction models, respectively. Since predictors are extracted from these objects, x=TRUE is required for a glm fitting.
z.std, z.new	Matrix of predictors for a standard and a new risk prediction model, respectively. Neither factor nor character nor missing values are allowed.
p.std, p.new	Vector of predicted risks from a standard and a new prediction model, respectively.
updown	Character to specify the method to determine UP and DOWN. When "category" is specified (by default), the risk category based NRI is calculated. When "diff" is specified, the risk difference based NRI is calculated.
cut	Scalar or vector values to specify the cutoff value(s) of predicted risks for determining UP and DOWN. For the risk category based NRI (updown = "category"), this option corresponds to cutoff value(s) of risk categories. For the risk difference based NRI (updown = "diff"), this option corresponds to a cutoff value of a risk difference (only scalar is allowed).
niter	Scalar value to determine the number of bootstrap sampling. When 0 is specified, an interval estimation is skipped.
link	Character to specify a link function for a glm fitting.
alpha	1-alpha confidence intervals are calculated.
msg	Logical value to display computation process. Setting FALSE leads silent execution.

**Details**

Either one set of the following arguments should be specified for the NRI calculation: (mdl.std, mdl.new); (event, z.std, z.new); and (event, p.std, p.new).

In the first set of the argument, (mdl.std, mdl.new), fitted results are used for the NRI calculation. event, z.std, and z.new are extracted from fitted result objects. The variance of model parameters are accounted for an interval estimation of the NRI. When event is specified in arguments, those specified is used without extracting from glm object.

In the second set of the argument, (event, z.std, z.new), a standard and a new prediction models are fitted inside this function with specified link. The variance of model parameters are also accounted for an interval estimation of the NRI.

In the third set of the argument, (event, p.std, p.new), predicted risks are used. Since fit of prediction models are not conducted while in a bootstrap, this can be used for a validation study by an external data source or by a cross-validation.

For the risk category based NRI calculation, cutoff values of risk category can be specified by `cut`, which is a scalar for the case of two risk categories and is a vector for the case of more than two risk categories. UP and DOWN are determined by the movement in risk categories.

For the risk difference based NRI calculation, cutoff values of risk difference can also be specified by `cut`, where UP and DOWN are defined as  $p_{new} - p_{standard} > \delta$  and  $p_{standard} - p_{new} > \delta$ , respectively.  $p_{standard}$  and  $p_{new}$  are predicted individual risks from a standard and a new prediction model, respectively, and  $\delta$  corresponds to `cut`. The continuous NRI, which is the special version of the risk difference based NRI, can be calculated by specifying both `updown = "diff"` and `cut = 0`.

Interval estimation is based on the percentile bootstrap method.

## Value

Returns a list of the following items:

<code>nri</code>	Point and interval estimates of the NRI and its components.
<code>mdl.std, mdl.new</code>	Fitted <code>glm</code> objects corresponding to a standard and a new prediction model, respectively. These items are provided when prediction models are fitted inside this function. Otherwise <code>NULL</code> .
<code>z.std, z.new</code>	Predictors of a standard and a new prediction model, respectively. These items are provided when they are extracted from <code>mdl.std</code> and <code>mdl.new</code> . Otherwise <code>NULL</code> .
<code>p.std, p.new</code>	Predicted risks by a standard and a new prediction model, respectively.
<code>up, down</code>	Logical values to show subjects who belong to UP and DOWN, respectively.
<code>rtab, rtab.case, rtab.ctrl</code>	table objects corresponding to reclassification tables for all, case, and control subjects, respectively. These items are provided only when the risk category based NRI is calculated and <code>msg = TRUE</code> .
<code>bootstrapsample</code>	Results of each bootstrap sample.

## Examples

```
## here consider pbc dataset in survival package as an example
library(survival)
dat = pbc[1:312,]
dat$sex = ifelse(dat$sex=='f', 1, 0)

## subjects censored before 2000 days are excluded
dat = dat[ dat$time > 2000 | (dat$time < 2000 & dat$status == 2), ]

## predicting the event of 'death' before 2000 days
event = ifelse(dat$time < 2000 & dat$status == 2, 1, 0)

## standard prediction model: age, bilirubin, and albumin
z.std = as.matrix(subset(dat, select = c(age, bili, albumin)))

## new prediction model: age, bilirubin, albumin, and protime
z.new = as.matrix(subset(dat, select = c(age, bili, albumin, protime)))
```



```

## glm fit (logistic model)
mstd = glm(event ~ ., binomial(logit), data.frame(event, z.std), x=TRUE)
mnew = glm(event ~ ., binomial(logit), data.frame(event, z.new), x=TRUE)

## predicted risk
p.std = mstd$fitted.values
p.new = mnew$fitted.values

## Calculation of risk difference NRI using ('mdl.std', 'mdl.std').
nrabin(mdl.std = mstd, mdl.new = mnew, cut = 0.02, niter = 0,
       updown = 'diff')

## Calculation of risk difference NRI using ('event', 'z.std', 'z.new').
nrabin(event = event, z.std = z.std, z.new = z.new, cut = 0.02,
       niter = 0, updown = 'diff')

## Calculation of risk difference NRI using ('event', 'p.std', 'p.new').
nrabin(event = event, p.std = p.std, p.new = p.new, cut = 0.02,
       niter = 0, updown = 'diff')

## Calculation of risk category NRI using ('mdl.std', 'mdl.new').
nrabin(mdl.std = mstd, mdl.new = mnew, cut = c(0.2, 0.4),
       niter = 0, updown = 'category')

```

---

nrabin.count.main	<i>Estimate NRI by the counting method.</i>
-------------------	---

---

## Description

Internally used function by nrabin. In the comparison for binary models, it is possible to use directly when UP and DOWN subjects are known.

## Usage

```
nrabin.count.main(event, upp, dwn)
```

## Arguments

event	Vector of event indicators, 1 for event of interest, 0 for non-event.
upp, dwn	Vector of logical values to determine subjects who belong to UP and DOWN, respectively.

nricens

*NRI for time to event models***Description**

This function estimates the NRI for competing risk prediction models with time to event variable. `coxph` object, `survreg` object, predictors, and predicted risks can be used as input data for the calculation. The risk category based NRI and the risk difference based NRI can be calculated. Users can use several types of estimators to obtain point estimates of the NRI and its components. The percentile bootstrap method is used for an interval estimation.

**Usage**

```
nricens(time = NULL, event = NULL, mdl.std = NULL, mdl.new = NULL,
        z.std = NULL, z.new = NULL, p.std = NULL, p.new = NULL, t0 = NULL,
        updown = "category", cut = NULL, point.method = "km",
        niter = 1000, alpha = 0.05, msg = TRUE)
```

**Arguments**

<code>time</code>	Vector of observed follow up times, $X = \min(T, C)$ . $T$ is event time, and $C$ is censoring time.
<code>event</code>	Vector of event indicators, 1 for event of interest, 0 for censoring.
<code>mdl.std, mdl.new</code>	<code>coxph</code> or <code>survreg</code> objects corresponding to a standard and a new risk prediction model, respectively. Since predictors are extracted from these objects, <code>x=TRUE</code> is required when fitting a Cox or a parametric survival model.
<code>z.std, z.new</code>	Matrix of predictors for a standard and a new risk prediction model, respectively. Neither factor nor character nor missing values are allowed.
<code>p.std, p.new</code>	Vector of predicted risks from a standard and a new prediction model, respectively.
<code>t0</code>	Scalar value indicating a time to determine event/non-event.
<code>updown</code>	Character to specify the method to determine UP and DOWN. When "category" is specified (by default), the risk category based NRI is calculated. When "diff" is specified, the risk difference based NRI is calculated.
<code>cut</code>	Scalar or vector values to specify the cutoff value(s) of predicted risks for determining UP and DOWN. For the risk category based NRI ( <code>updown = "category"</code> ), this option corresponds to cutoff value(s) of risk categories. For the risk difference based NRI ( <code>updown = "diff"</code> ), this option corresponds to a cutoff value of a risk difference (only scalar is allowed).
<code>point.method</code>	Character to determine an estimator for a point estimation. When "km" is specified, the Kaplan-Meier (KM) based NRI estimator is used (Pencina et al., 2011). When "ipw" is specified, the inverse probability weighting (IPW) NRI estimator is used (Uno et al., 2012).

niter	Scalar value to determine the number of bootstrap sampling. When 0 is specified, an interval estimation is skipped.
alpha	1-alpha confidence interval is calculated.
msg	Logical value to display computation process. Setting FALSE leads silent execution.

## Details

Either one set of the following arguments should be specified for the NRI calculation: (mdl.std, mdl.new); (time, event, z.std, z.new); and (time, event, p.std, p.new).

In the first set of the argument, (mdl.std, mdl.new), fitted results by coxph or survreg are used for the NRI calculation. time, event, z.std, and z.new are extracted from fitted result objects. The variance of model parameters are accounted for an interval estimation of the NRI. When time and event are specified in arguments, those specified are used without extracting from coxph or survreg objects.

In the second set of the argument, (time, event, z.std, z.new), a standard and a new prediction models are fitted inside this function with time, event, z.std and z.new. The variance of model parameters are also accounted for an interval estimation of the NRI.

In the third set of the argument, (time, event, p.std, p.new), predicted risks are used. Since fit of prediction models are not conducted while in a bootstrap, this can be used for a validation study by an external data source or by a cross-validation.

For the risk category based NRI calculation, cutoff values of risk category can be specified by cut, which is a scalar for the case of two risk categories and is a vector for the case of more than two risk categories. UP and DOWN are determined by the movement in risk categories.

For the risk difference based NRI calculation, cutoff values of risk difference can also be specified by cut, where UP and DOWN are defined as  $p_{new} - p_{standard} > \delta$  and  $p_{standard} - p_{new} > \delta$ , respectively.  $p_{standard}$  and  $p_{new}$  are predicted individual risks from a standard and a new prediction model, respectively, and  $\delta$  corresponds to cut. The continuous NRI, which is the special version of the risk difference based NRI, can be calculated by specifying both updown = "diff" and cut = 0.

Interval estimation is based on the percentile bootstrap method.

## Value

Returns a list of the following items:

nri	Point and interval estimates of the NRI and its components.
mdl.std, mdl.new	Fitted coxph or survreg objects corresponding to a standard and a new prediction model, respectively. These items are provided when prediction models are fitted inside this function. Otherwise NULL.
z.std, z.new	Predictors of a standard and a new prediction model, respectively. These items are provided when they are extracted from mdl.std and mdl.new. Otherwise NULL.
p.std, p.new	Predicted risks by a standard and a new prediction model, respectively.
up, down	Logical values to show subjects who belong to UP and DOWN, respectively.

`rtab, rtab.case, rtab.ctrl`  
 table objects corresponding to reclassification tables for all, case, and control subjects, respectively. These items are provided only when the risk category based NRI is specified and `msg = TRUE`.

`bootstrapsample`  
 Results of each bootstrap sample.

## References

Pencina MJ, D'Agostino RB, Steyerberg EW. Extensions of net reclassification improvement calculations to measure usefulness of new biomarkers. *Statistics in Medicine* 2011.

Uno H, Tian L, Cai T, Kohane IS, Wei LJ. A unified inference procedure for a class of measures to assess improvement in risk prediction systems with survival data, *Statistics in Medicine* 2012.

Hsu CH, Taylor JMG. A robust weighted Kaplan-Meier approach for data with dependent censoring using linear combinations of prognostic covariates, *Statistics in Medicine* 2010.

## Examples

```
## here consider pbc dataset in survival package as an example
library(survival)
dat = pbc[1:312,]
dat$sex = ifelse(dat$sex=='f', 1, 0)

## predicting the event of 'death'
time = dat$time
event = ifelse(dat$status==2, 1, 0)

## standard prediction model: age, bilirubin, and albumin
z.std = as.matrix(subset(dat, select = c(age, bili, albumin)))

## new prediction model: age, bilirubin, albumin, and protime
z.new = as.matrix(subset(dat, select = c(age, bili, albumin, protime)))

## coxph fit
mstd = coxph(Surv(time,event) ~ ., data.frame(time,event,z.std), x=TRUE)
mnew = coxph(Surv(time,event) ~ ., data.frame(time,event,z.new), x=TRUE)

## predicted risk at t0=2000
p.std = get.risk.coxph(mstd, t0=2000)
p.new = get.risk.coxph(mnew, t0=2000)

## Calculation of risk category NRI
## by the KM estimator using ('mdl.std', 'mdl.std').
nricens(mdl.std = mstd, mdl.new = mnew, t0 = 2000, cut = c(0.2, 0.4),
        niter = 0)

## by the KM estimator using ('time', 'event', 'z.std', 'z.std').
nricens(time = time, event = event, z.std = z.std, z.new = z.new,
        t0 = 2000, cut = c(0.2, 0.4), niter = 0)

## by the KM estimator using ('time', 'event', 'p.std', 'p.std').
```

```

nricens(time = time, event = event, p.std = p.std, p.new = p.new,
        t0 = 2000, cut = c(0.2, 0.4), niter = 0)

## Calculation of risk difference NRI by the KM estimator
nricens(mdl.std = mstd, mdl.new = mnew, t0 = 2000, updown = 'diff',
        cut = 0.05, niter = 0)

## Calculation of risk difference NRI by the IPW estimator
nricens(mdl.std = mstd, mdl.new = mnew, t0 = 2000, updown = 'diff',
        cut = 0.05, point.method = 'ipw', niter = 0)

```

---

nricens.ipw.main      *Estimate NRI by the inverse probability weighting (IPW) method.*

---

### Description

Internally used function by nricens to provide the NRI estimator by the IPW method. In the comparison for time to event models, it is possible to use directly when UP and DOWN subjects are known.

### Usage

```
nricens.ipw.main(time, event, upp, dwn, t0)
```

### Arguments

time	Vector of follow up times.
event	Vector of event indicators, 1 for event of interest, 0 for censoring.
upp, dwn	Vector of logical values to determine subjects who belong to UP and DOWN, respectively.
t0	Scalar value indicating a time to determine event/non-event.

---

nricens.km.main      *Estimate NRI by the standard Kaplan-Meier method.*

---

### Description

Internally used function by nricens to provide the NRI estimator by the Kaplan-Meier(KM) method. In the comparison for time to event models, it is possible to use directly when UP and DOWN subjects are known.

### Usage

```
nricens.km.main(time, event, upp, dwn, t0)
```

**Arguments**

time	Vector of follow up times.
event	Vector of event indicators, 1 for event of interest, 0 for censoring.
upp, dwn	Vectors of logical values to determine subjects who belong to UP and DOWN, respectively.
t0	Scalar value indicating a time to determine event/non-event.

# Index

`categorize`, 3

`get.risk.coxph`, 4

`get.risk.survreg`, 4

`get.surv.km`, 5

`get.uppdwn`, 5

`get.uppdwn.bin`, 6

`nribin`, 6

`nribin.count.main`, 9

`nricens`, 10

`nricens-package`, 2

`nricens.ipw.main`, 13

`nricens.km.main`, 13