Package 'SvyNom'

October 12, 2022

Type Package

Title Nomograms for Right-Censored Outcomes from Survey Designs

Version 1.2

Description Builds, evaluates and validates a nomogram with survey data and right-censored outcomes. As described in Capanu (2015) <doi:10.18637/jss.v064.c01>, the package contains functions to create the nomogram, validate it using bootstrap, as well as produce the calibration plots.

License GPL-2

URL https://github.com/MSKCC-Epi-Bio/SvyNom,
https://mskcc-epi-bio.github.io/SvyNom/

BugReports https://github.com/MSKCC-Epi-Bio/SvyNom/issues

Imports Hmisc, rms, survey, survival

Encoding UTF-8

LazyLoad yes

Suggests spelling

Language en-US

NeedsCompilation no

Author Mithat Gonen [aut, cre],
Marinela Capanu [aut]

Maintainer Mithat Gonen <gonenm@mskcc.org>

Repository CRAN

Date/Publication 2022-04-28 23:10:11 UTC

R topics documented:

SvyNom-package   *Nomograms for right-censored outcomes with complex survey data*

Description

Builds, evaluates and validates a nomogram with survey data and right-censored outcomes.

Details

Package:	SvyNom
Type:	Package
Version:	1.1
Date:	2015-01-06
License:	GPL-2
LazyLoad:	yes

There are three functions for the user svycox.nomogram, svycox.validate, svycox.calibrate

Author(s)

Marinela Capanu and Mithat Gonen
Maintainer: gonenm@mskcc.org

References

Marinela Capanu, Mithat Gonen (2015). Building a Nomogram for Survey-Weighted Cox Models Using R. Journal of Statistical Software, Code Snippets, 64(1), 1-17. URL https://www.jstatsoft.org/v64/c01/.

---

noNA   *Gastric cancer case-control study*

Description

Example dataset for the SvyNom package.

Usage

```
data(noNA)
```

Author(s)

Mithat Gonen, Marinela Capanu

Examples

```
data(noNA)
```

---

svycox.calibrate	*Checking the calibration of a nomogram for a survey-weighted Cox model*

Description

Checks the calibration of a nomogram for a survey-weighted Cox model.

Usage

```
svycox.calibrate(.nom, .timept = .nom$pred.at, .ngroup = 5)
```

Arguments

.nom	a nomogram object from svycox.nomogram
.timept	the time point at which calibration will take place; defaults to the time value of the prediction axis in the nomogram
.ngroup	number of groups to be formed for validation purposes

Value

returns a matrix of calibration values and plots them

Author(s)

Mithat Gonen, Marinela Capanu

References

Marinela Capanu, Mithat Gonen (2015). Building a Nomogram for Survey-Weighted Cox Models Using R. Journal of Statistical Software, Code Snippets, 64(1), 1-17. URL https://www.jstatsoft.org/v64/c01/.

Examples

```
library(survey)
library(rms)
data(noNA)
dd=datadist(noNA)
options(datadist="dd")

dstr2=svydesign(id=~1, strata=~group, prob=~inv_weight, fpc=~ssize, data=noNA)

mynom=svycox.nomogram(.design=dstr2, .model=Surv(survival,surv_cens)~ECOG+liver_only+Alb+Hb+Age+
Differentiation+Gt_1_m1site+lymph_only, .data=noNA, pred.at=24, fun.lab="Prob of 2 Yr OS")

svycox.calibrate(mynom)
```

---

svycox.nomogram *Builds a nomogram for a survey-weighted Cox model*

Description

Builds a nomogram for a survey-weighted Cox model.

Usage

```
svycox.nomogram(.design, .model, .data, pred.at, fun.lab)
```

Arguments

.design	represents a survey design object obtained with the package "survey"
.model	indicates a Cox model specification
.data	contains the data on which the model is to be fit (can not contain NAs)
pred.at	specifies the time point at which the nomogram prediction axis will be drawn
fun.lab	designate the label of the prediction axis

Details

In addition to the inputs, this function expects the following: 1) the input dataset (.data) cannot contain NAs. You can accomplish this using the na.omit function. See example. 2) datadist must be set. See examples and the documentation for the rms package. 3) survey design must have been saved in .design All of these requirements are explained in Capanu & Gonen (2015) in detail

Value

A list including elements

nomog	A nomogram object
preds	predicted values from the model

In addition to what is listed below, the design and the fitted survey weighted Cox model (svy.cox), as well as the timepoint at which the nomogram prediction axis will be drawn (pred.at) are stored

Author(s)

Mithat Gonen, Marinela Capanu

References

Marinela Capanu, Mithat Gonen (2015). Building a Nomogram for Survey-Weighted Cox Models Using R. Journal of Statistical Software, Code Snippets, 64(1), 1-17. URL https://www.jstatsoft.org/v64/c01/.

Examples

```
library(survey)
library(rms)
data(noNA)
dd=datadist(noNA)
options(datadist="dd")
dstr2=svydesign(id=~1, strata=~group, prob=~inv_weight, fpc=~ssize, data=noNA)
mynom=svycox.nomogram(.design=dstr2, .model=Surv(survival,surv_cens)~ECOG+liver_only+Alb+Hb+Age+
Differentiation+Gt_1_m1site+lymph_only, .data=noNA, pred.at=24, fun.lab="Prob of 2 Yr OS")
plot(mynom$nomog)
```

---

svycox.validate   *Validating a nomogram for a survey-weighted Cox model*

Description

Validates a nomogram for a survey-weighted Cox model using bootstrap.

Usage

```
svycox.validate(.boot.index, .nom, .data)
```

Arguments

.boot.index	a matrix of bootstrap sample indicators with the number of rows the same as the number of rows in the data on which the nomogram was created and the number of columns being the number of bootstrap samples
.nom	a nomogram object returned from svycox.nomogram
.data	contains the dataset on which the validation will take place

Details

Note that generating the bootstrap sample is design dependent and it is not part of the function. The user has to generate the bootstrap samples consistent with the design used. An example of how the bootstrap sample was generated for the dataset is presented in the reference below.

Value

prints the estimated optimism and returns the vector of optimism values for each bootstrap sample which can be used to summarize the validation with the measure of choice

References

Marinela Capanu, Mithat Gonen (2015). Building a Nomogram for Survey-Weighted Cox Models Using R. Journal of Statistical Software, Code Snippets, 64(1), 1-17. URL https://www.jstatsoft.org/v64/c01/.

Examples

```
bootit=200
library(survey)
library(rms)
data(noNA)
dd=datadist(noNA)
options(datadist="dd")
dstr2=svydesign(id=~1, strata=~group, prob=~inv_weight, fpc=~ssize, data=noNA)
mynom=svycox.nomogram(.design=dstr2, .model=Surv(survival,surv_cens)~ECOG+liver_only+Alb+Hb+Age+
Differentiation+Gt_1_m1site+lymph_only, .data=noNA, pred.at=24, fun.lab="Prob of 2 Yr OS")

cases=which(noNA$group=="long")
controls=which(noNA$group=="<24")
boot.index=matrix(NA,nrow(noNA),bootit)
for(i in 1:bootit){
boot.index[,i]=c(sample(cases,replace=TRUE),sample(controls,replace=TRUE))
}
myval=svycox.validate(boot.index,mynom,noNA)
```

Index