Package ‘nomogramFormula’

January 28, 2020

Type Package

Title Calculate Total Points and Probabilities for Nomogram

Version 1.2.0.0

Description A nomogram, which can be carried out in 'rms' package, provides a graphical explanation of a prediction process. However, it is not very easy to draw straight lines, read points and probabilities accurately. Even, it is hard for users to calculate total points and probabilities for all subjects.

This package provides formula_rd() and formula_lp() functions to fit the formula of total points with raw data and linear predictors respectively by polynomial regression. Function points_cal() will help you calculate the total points. prob_cal() can be used to calculate the probabilities after lrm(), cph() or psm() regression.

For more complex condition, interaction or restricted cubic spine, TotalPoints.rms() can be used.

Author Jing Zhang, Zhi Jin

Maintainer Jing Zhang<zj391120@163.com>

License GPL-3

Encoding UTF-8

LazyData true

Imports rms, do, Hmisc

RoxygenNote 6.1.1

URL https://github.com/yikeshu0611/nomogramFormula

BugReports https://github.com/yikeshu0611/nomogramFormula/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2020-01-28 13:10:02 UTC
Explore the Formula of Total Points and Linear Predictors

**Description**

Explore the formula of total points and linear predictors by the best power.

**Usage**

`formula_lp(nomogram, power, digits = 6)`

**Arguments**

- `nomogram`: results of `nomogram()` function in `rms` package
- `power`: power can be automatically selected based on all R2 equal 1
- `digits`: default is 6

**Value**

`formula` is the formula of total points and linear predictors. `test` is the R2 and RMSE which are used to test the fitted points. `diff` is difference between nomogram points and fitted points.

**Examples**

```r
library(rms) # needed for nomogram
set.seed(2018)
n <- 2019
age <- rnorm(n, 60, 20)
sex <- factor(sample(c("female", "male"), n, TRUE))
sex <- as.numeric(sex)
weight <- sample(50:100, n, replace = TRUE)
time <- sample(50:800, n, replace = TRUE)
units(time) = "day"
death <- sample(c(1, 0, 0), n, replace = TRUE)
df <- data.frame(time, death, age, sex, weight)
ddist <- datadist(df)
oldoption <- options(datadist = 'ddist')
f <- cph(formula(Surv(time, death) ~ sex + age + weight), data = df,
         x = TRUE, y = TRUE, surv = TRUE, time.inc = 3)
```
surv <- Survival(f)
nomo <- nomogram(f, 
lp=TRUE, 
fun=list(function(x) surv(365,x), 
function(x) surv(365*2,x)), 
funlabel=c("1-Year Survival Prob", 
"2-Year Survival Prob"))
options(oldoption)
formula_lp(nomogram = nomo)
formula_lp(nomogram = nomo, power = 1)
formula_lp(nomogram = nomo, power = 3, digits=6)

---

**Explore the Formula of Total Points and Raw Data**

**Description**

Explore the formula of total points and raw data by the best power.

**Usage**

`formula_rd(nomogram, power, digits = 6)`

**Arguments**

- `nomogram`: results of `nomogram()` function in `rms` package
- `power`: power can be automatically selected based on all R2 equal 1
- `digits`: default is 6

**Value**

`formula` is the formula of total points and raw data. `test` is the R2 and RMSE which are used to test the fitted points. `diff` is difference between nomogram points and fitted points

**Examples**

```r
library(rms) # needed for nomogram
set.seed(2018)
n <-2019
age <- rnorm(n,60,20)
sex <- factor(sample(c('female','male'),n,TRUE))
sex <- as.numeric(sex)
weight <- sample(50:100,n,replace = TRUE)
time <- sample(50:800,n,replace = TRUE)
units(time)="day"
death <- sample(c(1,0,0),n,replace = TRUE)
df <- data.frame(time,death,age,sex,weight)
ddist <- datadist(df)
oldoption <- options(datadist='ddist')
```

```r
f <- cph(formula(Surv(time,death)~sex+age+weight),data=df,
x=TRUE,y=TRUE,surv=TRUE,time.inc=3)
surv <- Survival(f)
nomo <- nomogram(f,
   lp=TRUE,
   fun=list(function(x) surv(365,x),
              function(x) surv(365*2,x)),
   funlabel=c("1-Year Survival Prob",
             "2-Year Survival Prob"))
options(oldoption)
formula_rd(nomogram = nomo)
formula_rd(nomogram = nomo,power = 1)
formula_rd(nomogram = nomo,power = 3,digits=6)
```

---

### points_cal

**Calculate Total Points**

**Description**

Calculate total points.

**Usage**

```r
points_cal(formula, rd, lp, digits = 6)
```

**Arguments**

- `formula`: the formula of total points with raw data or linear predictors
- `rd`: raw data, which cannot have missing values
- `lp`: linear predictors
- `digits`: default is 6

**Value**

total Points

**Examples**

```r
library(rms) # needed for nomogram
set.seed(2018)
n <-2019
age <- rnorm(n,60,20)
sex <- factor(sample(c('female','male'),n,TRUE))
sex <- as.numeric(sex)
weight <- sample(50:100,n,replace = TRUE)
time <- sample(50:800,n,replace = TRUE)
units(time)="day"
death <- sample(c(1,0,0),n,replace = TRUE)
df <- data.frame(time,death,age,sex,weight)
```
ddist <- datadist(df)
oldoption <- options(datadist='ddist')
f <- cph(formula(Surv(time,death)~sex+age+weight),data=df, x=TRUE,y=TRUE,surv=TRUE,time.inc=3)
surv <- Survival(f)
nomo <- nomogram(f,
lp=TRUE,
fun=list(function(x) surv(365,x),
function(x) surv(365*2,x)),
funlabel=c("1-Year Survival Prob",
"2-Year Survival Prob"))
options(oldoption)
#get the formula by the best power using formula_lp
results <- formula_lp(nomo)
points_cal(formula = results$formula,lp=f$linear.predictors)

#get the formula by the best power using formula_rd
results <- formula_rd(nomogram = nomo)
points_cal(formula = results$formula,rd=df)

prob_cal

Calculate Probabilities

Description
Use Survival() function from 'rms' pacakge to calculate probabilities after lrm(), cph() or psm() regression. If you want to calculate lrm() probabilities, please leave linear.predictors be TRUE and times be missing. If you want to calculate cph() probabilites, please leave both linear.predictors and surv be TRUE.

Usage
prob_cal(reg, times, q, lp)

Arguments
reg regression results after lrm(), cph() or psm() in 'rms' package.
times if you want to calculate probabilities for lrm() function, please left times missing.
q quantile, for example 0.5
lp linear predictors

Value
linear predictors and probabilities as a dataframe
Examples

```r
set.seed(2018)
n <- 2019
age <- rnorm(n, 60, 20)
sex <- factor(sample(c("female", "male"), n, TRUE))
sex <- as.numeric(sex)
weight <- sample(50:100, n, replace = TRUE)
time <- sample(50:800, n, replace = TRUE)
units(time) = "day"
death <- sample(c(1, 0, 0), n, replace = TRUE)
df <- data.frame(time, death, age, sex, weight)

library(rms) # needed for lrm(), cph() and psm()
ddist <- datadist(df)
oldoption <- options(datadist = 'ddist')

# lrm() function
f <- lrm(death ~ sex + age + weight, data = df, 
          linear.predictors = TRUE)
head(prob_cal(reg = f))

# cph() function
f <- cph(Surv(time, death) ~ sex + age + weight, data = df, 
          linear.predictors = TRUE, surv = TRUE)
head(prob_cal(reg = f, times = c(365, 365*2)))

# psm() function
f <- psm(Surv(time, death) ~ sex + age + weight, data = df)
head(prob_cal(reg = f, times = c(365, 365*2)))
```

---

**TotalPoints.rms**

_Calculate Total Points for nomogram Picture_

**Description**

Compared with points_cal() command, TotalPoints.rms() is suit for more complexed condition. Since this command is based on formula from `rms` package, it may be also more accurate. However, formula for each variable can not be calculated.

**Usage**

```r
TotalPoints.rms(rd, fit, nom, kint = NULL)
```

**Arguments**

- **rd** raw data
- **fit** regression result in `rma` package
- **nom** nomoram() command result
- **kint** number of intercept. Default is to use fit$interceptRef if it exists, or 1.
Value

A dataframe contains raw data and total points.

Examples

```r
library(rms)
set.seed(17)

d <- data.frame(age = rnorm(n, 50, 10),
  blood.pressure = rnorm(n, 120, 15),
  cholesterol = rnorm(n, 200, 25),
  sex = factor(sample(c('female', 'male'), n, TRUE)))

d <- upData(d,
  L = .4*(sex=='male') + .045*(age-50) +
    (log(cholesterol - 10)-5.2)*(-2*(sex=='female') + 2*(sex=='male')),
  y = ifelse(runif(n) < plogis(L), 1, 0))

ddist <- datadist(d); options(datadist='ddist')

f <- lrm(y ~ lsp(age,50) + sex * rcs(cholesterol, 4) + blood.pressure,
  data=d)

nom <- nomogram(f)

TotalPoints.rms(rd = d, fit = f, nom = nom)
```
Index

formula_lp, 2
formula_rd, 3
points_cal, 4
prob_cal, 5
TotalPoints.rms, 6