Package ‘ggDCA’

September 6, 2020

Title Calculate and Plot Decision Curve

Version 1.1

Description Diagnostic and prognostic models are typically evaluated with
measures of accuracy that do not address clinical consequences.
Decision-analytic techniques allow assessment of clinical outcomes but
often require collection of additional information and may be cumbersome to
apply to models that yield a continuous result.
Decision curve analysis is a suitable method for evaluating
alternative diagnostic and prognostic strategies that has advantages over
other commonly used measures and techniques.
This method was described by Andrew J. Vickers (2006) <doi:10.1177/0272989X06295361>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports do, set, rms (>= 6.0.1), base.rms, survival (>= 3.1-12)

Depends R (>= 2.10), ggplot2

NeedsCompilation no

Author Jing Zhang [aut, cre],
Zhi Jin [aut]

Maintainer Jing Zhang <zj391120@163.com>

Repository CRAN

Date/Publication 2020-09-06 16:10:02 UTC

R topics documented:

AUDC *Area under Decision Curve*

Description

Area under Decision Curve

Usage

```
AUDC(x)
```

Arguments

x	results of dca() function

Value

Area under decision curves for each model.

dca *Calculate Decision Curve Data*

Description

Calculate Decision Curve Data

Usage

```
dca(...)

## S3 method for class 'coxph'
dca(
  ...,
  model.names = do::get_names(...),
  test.harm = 0,
  new.data = NULL,
  times = "median"
)

## S3 method for class 'cph'
dca(
  ...,
  model.names = do::get_names(...),
  test.harm = 0,
  new.data = NULL,
  times = "median"
)

## S3 method for class 'glm'
dca(..., model.names = do::get_names(...), test.harm = 0, new.data = NULL)

## S3 method for class 'lrm'
dca(..., model.names = do::get_names(...), test.harm = 0, new.data = NULL)
```

Arguments

`...`	one or more results of logistic or cox regression
`model.names`	names for models
`test.harm`	test harm, default is 0
`new.data`	new data for validation
`times`	times for cox regresion, default is 'median'

Value

a dataframe contains thresholds, TPR: true positive rate, FPR: false positive rate, NB: net benefit, model: model names.

References

Vickers, A. J., & Elkin, E. B. (2006). Decision Curve Analysis: A Novel Method for Evaluating Prediction Models. Medical Decision Making, 26(6), 565–574. https://doi.org/10.1177/0272989X06295361

`ggplot.rFP.p100` *Plot for decision curve*

Description

Plot for decision curve

Plot for decision curve

Plot Decision Curve

Usage

```
## S3 method for class 'rFP.p100'
ggplot(
  data,
  mapping,
  color = TRUE,
  linetype = TRUE,
  lwd = 1.05,
  ...,
  environment = parent.frame()
)

## S3 method for class 'dca.cph'
ggplot(
  data,
  mapping,
  color = TRUE,
  linetype = TRUE,
  lwd = 1.05,
  ...,
  environment = parent.frame()
)

## S3 method for class 'dca.lrm'
ggplot(
  data,
  mapping,
  color = TRUE,
  linetype = TRUE,
  lwd = 1.05,
  ...,
  environment = parent.frame()
)
```

Arguments

`data`	results of dca() function
`mapping`	ignore
`color`	logical, whether models will be classified by color
`linetype`	logical, whether models will be classified by line type
`lwd`	line width
`...`	ignore
`environment`	ignore

Value

a ggplot2 picture

Examples

```
library(ggDCA)
library(rms)

######## logistic regression

model1 <- lrm(status~ANLN,LIRI)
```

```
d <- dca(model1,model.names = 'ANLN')
ggplot(d)



model2 <- lrm(status~ANLN+CENPA,LIRI)
d <- dca(model2,model.names = 'ANLN+CENPA')
ggplot(d)


model3 <- lrm(status~ANLN+CENPA+GPR182,LIRI)
d <- dca(model3,model.names = 'ANLN+CENPA+GPR182')
ggplot(d)

model4 <- lrm(status~ANLN+CENPA+GPR182+BCO2,LIRI)
d <- dca(model4,model.names = 'ANLN+CENPA+GPR182+BCO2')
ggplot(d)


d <- dca(model1,model2,model3,model4,
         model.names = c('ANLN',
                         'ANLN+CENPA',
                         'ANLN+CENPA+GPR182',
                         'ANLN+CENPA+GPR182+BCO2'))
ggplot(d,
       linetype = FALSE,
       color = c('blue','green','black','red','gray','gray'))


##########  cox regression

# evaluate at median time

model1 <- coxph(Surv(time,status)~ANLN,LIRI)
d <- dca(model1,model.names = 'ANLN')
ggplot(d)

model2 <- coxph(Surv(time,status)~ANLN+CENPA,LIRI)
d <- dca(model2,model.names = 'ANLN+CENPA')
ggplot(d)


model3 <- coxph(Surv(time,status)~ANLN+CENPA+GPR182,LIRI)
d <- dca(model3,model.names = 'ANLN+CENPA+GPR182')
ggplot(d)

model4 <- coxph(Surv(time,status)~ANLN+CENPA+GPR182+BCO2,LIRI)
d <- dca(model4,model.names = 'ANLN+CENPA+GPR182+BCO2')
ggplot(d)


d <- dca(model1,model2,model3,model4,
         model.names = c('ANLN',
                         'ANLN+CENPA',
                         'ANLN+CENPA+GPR182',
                         'ANLN+CENPA+GPR182+BCO2'))
ggplot(d,
       linetype = FALSE,
       color = c('blue','green','black','red','gray','gray'))



# evaluate at different times

qt <- quantile(LIRI$time,c(0.25,0.5,0.75))
qt=round(qt,2)
model1 <- coxph(Surv(time,status)~ANLN,LIRI)
d <- dca(model1,
         model.names = 'ANLN',
         times = qt)
ggplot(d)

model2 <- coxph(Surv(time,status)~ANLN+CENPA,LIRI)
d <- dca(model2,
         model.names = 'ANLN+CENPA',
         times = qt)
ggplot(d)


model3 <- coxph(Surv(time,status)~ANLN+CENPA+GPR182,LIRI)
d <- dca(model3,
         model.names = 'ANLN+CENPA+GPR182',
         times = qt)
ggplot(d)

model4 <- coxph(Surv(time,status)~ANLN+CENPA+GPR182+BCO2,LIRI)
d <- dca(model4,
         model.names = 'ANLN+CENPA+GPR182+BCO2',
         times = qt)
ggplot(d)


d <- dca(model1,model2,model3,model4,
         model.names = c('ANLN',
                         'ANLN+CENPA',
                         'ANLN+CENPA+GPR182',
                         'ANLN+CENPA+GPR182+BCO2'),
         times = qt)
ggplot(d)
```

LIRI *ICGC Liver Data from Japan*

Description

This data is a liver cancer data from Japan Data released in ICGC database (Link). It cantains time, event and four genes.

Usage

```
data(LIRI)
```

Format

An object of class `data.frame` with 232 rows and 6 columns.

Examples

```
data(LIRI)
```

`range` *Ranges for net benefit*

Description

Ranges for net benefit

Usage

```
## S3 method for class 'dca.lrm'
range(...)

## S3 method for class 'dca.cph'
range(...)
```

Arguments

`...`	results of dca() function

Value

a dataframe contains the minium and maximum of net benefit for each model `rFP.p100` *Calculate reduction in false positive count*

Description

Calculate reduction in false positive count

Usage

```
rFP.p100(x)
```

Arguments

`x`	result of dca() function

Value

a dataframe contains thresholds, rFP.100: reduction in false positive count per 100 patients.

Index