Package ‘ROC’

Type: Package

Title: Curve ROC analysis

Version: 0.1.0

Author: Juana Maria Vivo Molina, Manuel Franco Nicolás, Juan Pedro García Ortiz

Maintainer: Juan Pedro García Ortiz <juanpedro.garcia4@um.es >

Description: The package analyzes the Curve ROC, identificates it among different types of Curve ROC and calculates the area under de curve through the method that is most accuracy. More about what it does (maybe more than one line)

License: GPL-3

Encoding: UTF-8

LazyData: FALSE

Depends: golubEsets

import: golubEsets

BiocViews: Software, StatisticalMetho, Regression

RoxygenNote: 6.1.1

Suggests: knitr

VignetteBuilder: knitr

R topics documented:

points.curve *Function to calculate the coordinates (fpr,sen)*

Description

This function generate the sensibility and the specifity for each point creater between the lowest and highest value related to the cases once the model has been aplied. Points.curve needs 2 parameters: xsample and ysample. The first parameter correspond to the cases tags, while the seconde parameter is the value that the model has generated for each cases. The function generate as much points as cases there are in the vectors. Th points are generated through the mean between the ysample's lowest value and the next one, plus a point with the lowest value minus one, and another point with the highest value plus one.

Usage

```
points.curve(xsample, ysample)
```

Arguments

xsample	sample etiquetes
ysample	vector with the values of sensibility of a model

Format

An object of class NULL of length 0.

Value

return a matrix with a serie of coordenates that will be used to generate a ROC curve

Examples

```
# using example data from our package
data("intersticial")
intersticial <- intersiticial
sample <- sample(intersticial, dim(intersticial[, 1])/10)
dataset.boot<-intersticial[sample, ]
```

```
setsample<-dataset.boot[, 1]
datasample<-dataset.boot[, 2]
points <- points.curve(setsample, datasample)
```

TpAUC *Clasification of partial ROC curve*

Description

This function clasifies the ROC curve as valid or non valid using the specificity and sensibility that the function Points.curve has created. If is valid the function clasifies it as proper or non proper. Once is clasified, the function calculates the AUCp. A invalid ROC curve is once that is not growing. While and non proper is one that cross the diagonal line. This function has four parameters: two mandatories (dataset and datasample) and two optionals (low limit and up limit.). Dataset is a dataframe object generated by the model, this must be formed for two columns, the first one belong to the tags and the second to the values generated by the model. The datasample parameter is a vector in which is indicated the dataset's cases that are going to be tested. The limits parameters are two numeric values between 0 and 1. If these parameters are null the function take 0 and 1 as low.limit and up.limit respectivly. The function returns and object with four elements: AUCp without standarization, AUCp standarized, sensibilities and specifities.

Usage

```
TpAUC_function(bsdataset, bsdatasample, low.limit, up.limit)
```

Arguments

bsdataset	Dataframe of the complete information of the samples
bsdatasample	vector that contains the samples that are going to be analysed
Low.limit	Numeric value between 0 and 1
Up.limit	Numeric value between 0 and 1

Examples

```
#Using data from our package
data(Intersticial)
sample <- sample(intersticial, dim(intersticial[, 1])/10)
TpAUC <- TpAUC.function(intesticial, sample)
```

MpAUC *Clasification of partial ROC curve*

Description

This function calculates the AUCp introduced through the parameters. This function has four parameters: two mandatories (dataset and datasample) and two optionals (low limit and up limit.). Dataset is a dataframe object generated by the model, this must be formed for two columns, the first one belong to the tags and the second to the values generated by the model. The datasample parameter is a vector in which is indicated the dataset's cases that are going to be tested. The limits parameters are two numeric values between 0 and 1. If these parameters are null the function take 0 and 1 as low.limit and up.limit respectivly. The function returns and object with four elements: AUCp without standarization, AUCp standarized, sensibilities and specifities. Notice that if the AUCp is under the diagonal line the AUCp standarized will be NA.

Usage

```
MpAUC(bsdataset, bsdatasample, low.limit, up.limit)
```

Arguments

bsdataset	Dataframe of the complete information of the samples
bsdatasample	vector that contains the samples that are going to be analysed
Low.limit	Numeric value between 0 and 1
Up.limit	Numeric value between 0 and 1

Examples

```
#Using data from our package
data(Intersticial)
sample <- sample(intersticial, dim(intersticial[, 1])/10)
MpAUC.example <- MpAUC(intesticial, sample)
```