Package ‘longROC’

October 13, 2022

Type   Package
Title   Time-Dependent Prognostic Accuracy with Multiply Evaluated Bio
        Markers or Scores
Version 1.0
Date   2017-11-18
Author Alessio Farcomeni
Maintainer Alessio Farcomeni <alessio.farcomeni@uniroma1.it>
Description Time-dependent Receiver Operating Characteristic curves, Area Un-
        der the Curve, and Net Reclassification Indexes for repeated measures. It is based on meth-
License GPL (>= 2)
Depends R (>= 3.1.2)
Imports survival
Encoding UTF-8
LazyData true
NeedsCompilation no
Repository CRAN
Date/Publication 2017-11-20 12:17:47 UTC

**R topics documented:**

- auc
- bootstrap
- bootstrap.nri
- bootstrap.s
- maxauc
- maxauc.s
- nri
- plotAUC
- plotAUC.s
- plotROC
Description

Compute area under the ROC curve

Usage

```
 auc(ss)
```

Arguments

- **ss**: Matrix with two columns (1-specificities, sensitivities). It can be simply the output of `roc` function.

Details

Area under the ROC curve.

Value

A scalar with the AUC.

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References


See Also

```
roc, bootstrap, maxauc
```

Examples

# parameters
n=100
tt=3
Tmax=10
u=1.5
s=2
vtimes=c(0,1,2,5)

# generate data

ngrid=5000
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
  sa=sample(ngrid/6,1)
  vals=sample(3,1)-1
  X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(vtimes))
S2=matrix(NA,n,length(vtimes))
S2[,1]=X2[,1]
for(j in 2:length(vtimes)) {
  tm=which.min(abs(ts-vtimes[j]))
  S2[,j]=X2[,tm]
}
cens=runif(n)
ripart=1-exp(-0.01*apply(exp(X2),1,cumsum)*ts/1:ngrid)

Ti=rep(NA,n)
for(i in 1:n) {
  Ti[i]=ts[which.min(abs(ripart[i]-cens[i]))]
}
cens=runif(n,0,Tmax*2)
delta=ifelse(cens>Ti,1,0)
Ti[cens<Ti]=cens[cens<Ti]

##
## an important marker

ro=roc(S2,Ti,delta,u,tt,s,vtimes)
auc(ro)

## an unrelated marker

ro=roc(S1,Ti,delta,u,tt,s,vtimes)
Bootstrapping AUC

Description

Bootstrap the AUC for significance testing and confidence interval calculation

Usage

\texttt{butstrap(X, etime, status, u=NULL, tt, s, vtimes, auc1, B=50, fc=NULL)}

Arguments

- \texttt{X}: \( n \) by \( S \) matrix of longitudinal score/biomarker for \( i \)-th subject at \( j \)-th occasion (NA if unmeasured)
- \texttt{etime}: \( n \) vector with follow-up times
- \texttt{status}: \( n \) vector with event indicators
- \texttt{u}: Lower limit for evaluation of sensitivity and specificity. Defaults to \( \text{vtimes}[s] \) (see below)
- \texttt{tt}: Upper limit (time-horizon) for evaluation of sensitivity and specificity.
- \texttt{s}: Scalar number of measurements/visits to use for each subject. \( s \leq S \)
- \texttt{vtimes}: \( S \) vector with visit times
- \texttt{auc1}: AUC for the original data set
- \texttt{B}: Number of bootstrap replicates. Defaults to 50
- \texttt{fc}: Events are defined as \( fc = 1 \). Defaults to \( \text{SI}(cup X(t_j)>cutoff) \)

Details

This function can be used to resample the AUC. The resulting p-value is obtained after assumption that the resampled AUC is Gaussian. Non-parametric confidence interval is obtained as the 2.5 and 97.5 confidence interval is simply given by a Gaussian approximation.

Value

A list with the following elements:

- \texttt{p.value}: (Parametric) p-value for \( H_0: \text{AUC}=0.5 \)
- \texttt{se}: Standard deviation of the AUC replicates
- \texttt{ci.np}: Non-parametric 95\% confidence interval for AUC
- \texttt{ci.par}: Parametric 95\% confidence interval for AUC
Author(s)
Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

See Also
roc, auc, maxauc

Examples

```r
# parameters
n=100
tt=3
Tmax=10
u=1.5
s=2
vtimes=c(0,1,2,5)

# generate data
ngrid=5000
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
  sa=sample(ngrid/6,1)
  vals=sample(3,1)-1
  X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(vtimes))
S2=matrix(NA,n,length(vtimes))
S2[,1]=X2[,1]
for(j in 2:length(vtimes)) {
  tm=which.min(abs(ts-vtimes[j]))
  S2[,j]=X2[,tm]
}
cens=runif(n)
ripart=1-exp(-0.01*apply(exp(X2),1,cumsum)*ts/1:ngrid)
Ti=rep(NA,n)
for(i in 1:n) {
  Ti[i]=ts[which.min(abs(ripart[,i]-cens[i]))]
}
```
```r
cens = runif(n, 0, Tmax*2)
delta = ifelse(cens > Ti, 1, 0)
Ti[cens < Ti] = cens[cens < Ti]

## an unimportant marker
ro = roc(S1, Ti, delta, u, tt, s, vtimes)
but = bootstrap(S1, Ti, delta, u, tt, s, vtimes, ro)
```

---

**butstrap.nri**  
*Bootsraping NRI*

**Description**

Bootstrap the AUC for significance testing and confidence interval calculation.

**Usage**

```r
butstrap.nri(risk1, risk2, etime, status, u, tt, nri1, wh, B = 1000)
```

**Arguments**

- `risk1`: Baseline risk measurements
- `risk2`: Enhanced risk measurements
- `etime`: n vector with follow-up times
- `status`: n vector with event indicators
- `u`: Lower limit for evaluation of sensitivity and specificity
- `tt`: Upper limit (time-horizon) for evaluation of sensitivity and specificity.
- `nri1`: NRI for the original data set
- `wh`: Which NRI to bootstrap? `wh = 1` 1/2NRI, `wh = 2` NRI for events, `wh = 3` NRI for non-events
- `B`: Number of bootstrap replicates. Defaults to `1000`

**Details**

This function can be used to resample the NRI. The resulting p-value is obtained after assumption that the resampled NRI is Gaussian. Non-parametric confidence interval is obtained as the 2.5 and 97.5 confidence interval is simply given by a Gaussian approximation.

**Value**

A list with the following elements:

- `p.value`: (Parametric) p-value for H0: NRI = 0
- `se`: Standard deviation of the NRI replicates
- `ci.np`: Non-parametric 95% confidence interval for NRI
- `ci.par`: Parametric 95% confidence interval for NRI
Author(s)
Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

See Also
nri

Examples
```r
# parameters
n=25
tt=3
Tmax=10
u=1.5
s=2
vtimes=c(0,1,2,5)

# generate data

ngrid=1000
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
  sa=sample(ngrid/6,1)
  vals=sample(3,1)-1
  X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(vtimes))
S2=matrix(NA,n,length(vtimes))
S2[,1]=X2[,1]
for(j in 2:length(vtimes)) {
  tm=which.min(abs(ts-vtimes[j]))
  S2[,j]=X2[,tm]
  cens=runif(n)
  ripart=1-exp(-0.01*apply(exp(X2),1,cumsum)*ts/1:ngrid)
  Ti=rep(NA,n)
  for(i in 1:n) {
    Ti[i]=ts[which.min(abs(ripart[,i]-cens[i]))]
  }
}
```
cens=runif(n,0,Tmax*2)
delta=ifelse(cens>Ti,1,0)
Ti[cens<Ti]=cens[cens<Ti]

risk1=apply(S1[,1:s],1,sum)
risk1=(risk1-min(risk1))/(max(risk1)-min(risk1))
risk2=apply(S2[,1:s],1,sum)
risk2=(risk2-min(risk2))/(max(risk2)-min(risk2))
butstrap.nri(risk1,risk2,Ti,delta,u,tt,nri(risk1,risk2,Ti,delta,u,tt)$nri,wh=1,B=500)

butstrap.s

---

**Description**

Bootstrap the AUC for significance testing and confidence interval calculation

**Usage**

`butstrap.s(X, etime, status, u=NULL, tt, s, vtimes, auc1, B=50, fc=NULL)`

**Arguments**

- **X**: n by S matrix of longitudinal score/biomarker for i-th subject at j-th occasion (NA if unmeasured)
- **etime**: n vector with follow-up times
- **status**: n vector with event indicators
- **u**: Lower limit for evaluation of sensitivity and specificity. Defaults to `max(vtimes[s])` (see below)
- **tt**: Upper limit (time-horizon) for evaluation of sensitivity and specificity.
- **s**: n vector of number of measurements/visits to use for each subject. all(s<=S)
- **vtimes**: S vector with visit times
- **auc1**: AUC for the original data set
- **B**: Number of bootstrap replicates. Defaults to 50
- **fc**: Events are defined as fc = 1. Defaults to `Sil(cup X(t_j)>cutoff)`

**Details**

This function can be used to resample the AUC. The resulting p-value is obtained after assumption that the resampled AUC is Gaussian. Non-parametric confidence interval is obtained as the 2.5 and 97.5 confidence interval is simply given by a Gaussian approximation.

**Value**

A list with the following elements:
Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References


See Also

*roc, auc, maxauc*

Examples

```r
# parameters
n=100
tt=3
Tmax=10
u=1.5
s=sample(3,n,replace=TRUE)
vtimes=c(0,1,2,5)

# generate data

ngrid=5000
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
  sa=sample(ngrid/6,1)
  vals=sample(3,1)-1
  X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(vtimes))
S2=matrix(NA,n,length(vtimes))
S2[,1]=X2[,1]
for(j in 2:length(vtimes)) {
  tm=which.min(abs(ts-vtimes[j]))
  S2[,j]=X2[,tm]
}
cens=runif(n)
```
ripart = 1 - exp(-0.01 * apply(exp(X2), 1, cumsum) * ts / ngrid)

Ti = rep(NA, n)
for (i in 1:n) {
  Ti[i] = ts[which.min(abs(ripart[, i] - cens[i]))]
}

cens = runif(n, 0, Tmax*2)
delta = ifelse(cens > Ti, 1, 0)
Ti[cens < Ti] = cens[cens < Ti]

## an unimportant marker
ro = roc.s(S1, Ti, delta, u, tt, s, vtimes)
but = butstrap.s(S1, Ti, delta, u, tt, s, vtimes, ro)

---

maxauc

**Optimal Score**

**Description**

Compute optimal score for AUC

**Usage**

maxauc(X, etime, status, u = NULL, tt, s, vtimes, fc = NULL)

**Arguments**

- **X**: p by n by S array of longitudinal scores/biomarkers for i-th subject at j-th occasion (NA if unmeasured)
- **etime**: n vector with follow-up times
- **status**: n vector with event indicators
- **u**: Lower limit for evaluation of sensitivity and specificity. Defaults to vtimes[s] (see below)
- **tt**: Upper limit (time-horizon) for evaluation of sensitivity and specificity.
- **s**: Scalar number of measurements/visits to use for each subject. s<=S
- **vtimes**: S vector with visit times
- **fc**: Events are defined as fc = 1. Defaults to $I(\cup X(t_j)>cutoff)$

**Details**

This function can be used to find an optimal linear combination of p scores/biomarkers repeatedly measured over time. The resulting score is optimal as it maximizes the AUC among all possible linear combinations. The first biomarker in array X plays a special role, as by default its coefficient is unitary.
Value

A list with the following elements:

<table>
<thead>
<tr>
<th>beta</th>
<th>Beta coefficients for the optimal score</th>
</tr>
</thead>
<tbody>
<tr>
<td>score</td>
<td>Optimal score</td>
</tr>
</tbody>
</table>

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References


See Also

`auc`, `butstrap`, `maxauc`

Examples

```r
# parameters
n=25
tt=3
Tmax=10
u=1.5
s=2
vtimes=c(0,1,2,5)

# generate data
ngrid=500
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
  sa=sample(ngrid/6,1)
  vals=sample(3,1)-1
  X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(vtimes))
S2=matrix(NA,n,length(vtimes))
S2[,1]=X2[,1]
for(j in 2:length(vtimes)) {
  tm=which.min(abs(ts-vtimes[j]))
  S2[,j]=X2[,tm]
}`
cens=runif(n)
ripart=1-exp(-0.01*apply(exp(X2),1,cumsum)*ts/1:ngrid)

Ti=rep(NA,n)
for(i in 1:n) {
  Ti[i]=ts[which.min(abs(ripart[,i]-cens[i]))]
}
cens=runif(n,0,Tmax*2)
delta=ifelse(cens>Ti,1,0)
Ti[cens<Ti]=cens[cens<Ti]

##
X=array(NA,c(2,nrow(S1),ncol(S1))))
X[1,]=round(S2) #fewer different values, quicker computation
X[2,]=S1
sc=maxauc(X,Ti,delta,u,tt,s,vtimes)
# beta coefficients
sc$beta

# final score (X[1,]+X[2,]*sc$beta[1]+...+X[p,]*sc$beta[p-1])
sc$score

maxauc.s

---

Optimal Score

**maxauc.s**

---

**Description**

Compute optimal score for AUC

**Usage**

```r
maxauc.s(X, etime, status, u=NULL, tt, s, vtimes, fc=NULL)
```

**Arguments**

- **X**: n by S matrix of longitudinal score/biomarker for i-th subject at j-th occasion (NA if unmeasured)
- **etime**: n vector with follow-up times
- **status**: n vector with event indicators
- **u**: Lower limit for evaluation of sensitivity and specificity. Defaults to `max(vtimes[s])` (see below)
maxauc.s

upper limit (time-horizon) for evaluation of sensitivity and specificity.
s n vector of number of measurements/visits to use for each subject. all(s<=S)
vtimes S vector with visit times
fc Events are defined as fc = 1. Defaults to $I(cup X(t_j)>cutoff)$

Details
This function can be used to find an optimal linear combination of p scores/biomarkers repeatedly measured over time. The resulting score is optimal as it maximizes the AUC among all possible linear combinations. The first biomarker in array X plays a special role, as by default its coefficient is unitary.

Value
A list with the following elements:

beta Beta coefficients for the optimal score
score Optimal score

Author(s)
Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

See Also
auc, bootstrap, maxauc

Examples

# parameters
n=20
tt=3
Tmax=10
u=1.5
s=sample(3,n,replace=TRUE)
vtimes=c(0,1,2,5)

# generate data

ngrid=500
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {

sa = sample(ngrid/6, 1)
vals = sample(3, 1)
X2[i, 1:sa[1]] = vals[1] + X2[i, 1:sa[1]]
}

S1 = matrix(sample(4, n, replace=TRUE), n, length(vtimes))
S2 = matrix(NA, n, length(vtimes))
S2[, 1] = X2[, 1]
for(j in 2:length(vtimes)) {
  tm = which.min(abs(ts - vtimes[j]))
  S2[, j] = X2[, tm]
}
cens = runif(n)
ripart = 1 - exp(-0.01 * apply(exp(X2), 1, cumsum) * ts / 1:ngrid)

Ti = rep(NA, n)
for(i in 1:n) {
  Ti[i] = ts[which.min(abs(ripart[, i] - cens[i]))]
}
cens = runif(n, 0, Tmax * 2)
delta = ifelse(cens > Ti, 1, 0)
Ti[cens < Ti] = cens[cens < Ti]

##
X = array(NA, c(2, nrow(S1), ncol(S1)))
X[1, , ] = round(S2)  # fewer different values, quicker computation
X[2, , ] = S1
sc = maxauc.s(X, Ti, delta, u, tt, s, vtimes)

# beta coefficients
sc$beta

# final score (X[1, , ] * X[2, , ] * sc$beta[1] + ... + X[p, , ] * sc$beta[p-1])
sc$score

---

### nri

**NRI**

**Description**

Compute NRI
Usage

nri(risk1, risk2, etime, status, u, tt)

Arguments

risk1  Baseline risk measures
risk2  Enhanced risk measures
etime  n vector with follow-up times
status n vector with event indicators
u     Lower limit for evaluation of sensitivity and specificity.
tt    Upper limit (time-horizon) for evaluation of sensitivity and specificity.

Details

This function gives the continuous NRI to compare two risk measures.

Value

A list with the following elements:

  nri  1/2 NRI
  nri.events  NRI for events
  nri.nonevents  NRI for non-events

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References


See Also

bootstrap.nri

Examples

# parameters
n=100
tt=3
Tmax=10
u=1.5
s=2
vtimes=c(0,1,2,5)
# generate data

grid=5000

Ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
  sa=sample(ngrid/6,1)
  vals=sample(3,1)-1
  X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(vtimes))
S2=matrix(NA,n,length(vtimes))
S2[,1]=X2[,1]

for(j in 2:length(vtimes)) {
  tm=which.min(abs(ts-vtimes[j]))
  S2[,j]=X2[,tm]
}
cens=runif(n)
ripart=1-exp(-0.01*apply(exp(X2),1,cumsum)*ts/ngrid)

Ti=rep(NA,n)
for(i in 1:n) {
  Ti[i]=ts[which.min(abs(ripart[i]-cens[i]))]
}
cens=runif(n,0,Tmax*2)
delta=ifelse(cens>Ti,1,0)
Ti[cens<Ti]=cens[cens<Ti]

risk1=apply(S1[,1:s],1,sum)
risk1=(risk1-min(risk1))/(max(risk1)-min(risk1))
risk2=apply(S2[,1:s],1,sum)
risk2=(risk2-min(risk2))/(max(risk2)-min(risk2))
nri(risk1,risk2,Ti,delta,u,tt)
### Arguments

- **X**: n by S matrix of longitudinal score/biomarker for i-th subject at j-th occasion (NA if unmeasured)
- **etime**: n vector with follow-up times
- **status**: n vector with event indicators
- **u**: Lower limit for evaluation of sensitivity and specificity. Defaults to `vtimes[s]` (see below)
- **tt**: A vector of upper limits (time-horizons) for evaluation of sensitivity and specificity.
- **s**: Scalar number of measurements/visits to use for each subject. s<=S
- **vtimes**: S vector with visit times
- **fc**: Events are defined as fc = 1. Defaults to $\text{SI}(\text{cup} X(t_j)>\text{cutoff})$
- **plot**: Do we plot the AUCs? Defaults to `TRUE`

### Details

Area under the ROC curve is computed for each value of the vector `tt`. The resulting vector is returned. If `plot=TRUE` (which is the default) also a plot of `tt` vs AUC is displayed.

### Value

A vector with AUCs

### Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

### References


### See Also

`roc`, `butstrap`, `auc`

### Examples

```r
# parameters
n=25
tt=3
Tmax=10
u=1.5
s=2
vtimes=c(0,1,2,5)
```
# generate data

ngrid=1000
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
  sa=sample(ngrid/6,1)
  vals=sample(3,1)-1
  X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(vtimes))
S2=matrix(NA,n,length(vtimes))
S2[,1]=X2[,1]
for(j in 2:length(vtimes)) {
  tm=which.min(abs(ts-vtimes[j]))
  S2[,j]=X2[,tm]
}
cens=runif(n)
ripart=1-exp(-0.01*apply(exp(X2),1,cumsum)*ts/1:ngrid)

Ti=rep(NA,n)
for(i in 1:n) {
  Ti[i]=ts[which.min(abs(ripart[,i]-cens[i]))]
}
cens=runif(n,0,Tmax*2)
delta=ifelse(cens>Ti,1,0)
Ti[cens<Ti]=cens[cens<Ti]

##
## an important marker

aucs=plotAUC(S2,Ti,delta,u,seq(2,5,length=5),s,vtimes)

---

**plotAUC.s**

*AUC as a function of time*

---

**Description**

Compute area under the ROC curve for several values of the time horizon

**Usage**

plotAUC.s(X, etime, status, u=NULL, tt, s, vtimes, fc=NULL, plot=TRUE)
Arguments

X  n by S matrix of longitudinal score/biomarker for i-th subject at j-th occasion (NA if unmeasured)
etime  n vector with follow-up times
status  n vector with event indicators
u  Lower limit for evaluation of sensitivity and specificity. Defaults to vtimes[s] (see below)
tt  A vector of upper limits (time-horizons) for evaluation of sensitivity and specificity.
s  n vector of measurements/visits to use for each subject. all(s<=S)
vtimes  S vector with visit times
fc  Events are defined as fc = 1. Defaults to $I(cup X(t_j)>cutoff)$
plot  Do we plot the AUCs? Defaults to TRUE

Details

Area under the ROC curve is computed for each value of the vector tt. The resulting vector is returned. If plot=TRUE (which is the default) also a plot of tt vs AUC is displayed.

Value

A vector with AUCs

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References


See Also

roc.s, bootstrap.s, auc

Examples

# parameters
n=25
tt=3
Tmax=10
u=1.5
s=sample(3,n,replace=TRUE)
vtimes=c(0,1,2,5)
# generate data

```r
ngrid=1000
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
  sa=sample(ngrid/6,1)
  vals=sample(3,1)-1
  X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(vtimes))
S2=matrix(NA,n,length(vtimes))
S2[,1]=X2[,1]
for(j in 2:length(vtimes)) {
  tm=which.min(abs(ts-vtimes[j]))
  S2[,j]=X2[,tm]
}
cens=runif(n)
ripart=1-exp(-0.01*apply(exp(X2),1,cumsum)*ts/1:ngrid)
Ti=rep(NA,n)
for(i in 1:n) {
  Ti[i]=ts[which.min(abs(ripart[,i]-cens[i]))]
}
cens=runif(n,0,Tmax*2)
delta=ifelse(cens>Ti,1,0)
Ti[cens<Ti]=cens[cens<Ti]
```

```r
##
## an important marker
```

```r
aucs=plotAUC.s(S2,Ti,delta,u,seq(2,5,length=5),s,vtimes)
```

---

**plotROC**  

**Plot ROC**

---

**Description**

Plot the ROC curve

**Usage**

```r
plotROC(ro, add=FALSE, col=NULL)
```
Arguments

ro Matrix with two columns (1-specificities, sensitivities). It can be simply the output of roc function
add If FALSE (default) creates a new plot, otherwise adds to the existing one
col Colour for the ROC curve (defaults to red)

Details

Plots the area under the ROC curve.

Value

A plot or a new line in an open plot.

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References


See Also

roc, roc.s, auc

Examples

# parameters
n=100
tt=3
Tmax=10
u=1.5
s=2
vtimes=c(0,1,2,5)

# generate data

ngrid=5000
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
  sa=sample(ngrid/6,1)
  vals=sample(3,1)-1
  X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}

# plot ROC
plotROC(ro, add=TRUE, col="red")
roc

### Description

Compute ROC curve

### Usage

```r
roc(X, etime, status, u=NULL, tt, s, vtimes, fc=NULL)
```

### Arguments

- **X**: n by S matrix of longitudinal score/biomarker for i-th subject at j-th occasion (NA if unmeasured)
- **etime**: n vector with follow-up times
status n vector with event indicators
u Lower limit for evaluation of sensitivity and specificity. Defaults to \texttt{vtimes[s]}
(see below)
tt Upper limit (time-horizon) for evaluation of sensitivity and specificity.
s Scalar number of measurements/visits to use for each subject. s<=S
vtimes S vector with visit times
fc Events are defined as fc = 1. Defaults to $\{\text{cup} \ X(t_j)>cutoff\}$

Details

ROC curve is defined as the curve given by (1-specificities, sensitivities). Here these are obtained for a time-dependent multiply-measured marker are defined as

\[
Se(t,c,s,u) = \Pr(f_c(X(t_1),X(t_2),...,X(t_s_i))|u <= T <= t),
\]

and

\[
Sp(t,c,s,u) = 1-\Pr(f_c(X(t_1),X(t_2),...,X(t_s_i)) | T > t)
\]

for some fixed $f_c$, where c is a cutoff. The default for $f_c$ is that a positive diagnosis is given as soon as any measurement among the $s$ considered is above the threshold.

Value

A matrix with the following columns:

<table>
<thead>
<tr>
<th>1-spec</th>
<th>1-Specificities</th>
</tr>
</thead>
<tbody>
<tr>
<td>sens</td>
<td>Sensitivities</td>
</tr>
</tbody>
</table>

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References


See Also

\texttt{auc, bootstrap, maxauc}

Examples

# parameters
n=100
tt=3
Tmax=10
u=1.5
s=2
times=c(0,1,2,5)

# generate data

grid=5000
ts=seq(0,Tmax,length=grid)
X2=matrix(rnorm(n*grid,0,0.1),n,grid)
for(i in 1:n) {
sa=sample(grid/6,1)
vals=sample(1,1)
X2[i,1:sa]=vals+X2[i,1:sa]
X2[i,(sa+1):grid]=vals+sample(c(-2,2),1)+X2[i,(sa+1):grid]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(times))
S2=matrix(NA,n,length(times))

S2[,1]=X2[,1]
for(j in 2:length(times)) {
  tm=which.min(abs(ts-times[j]))
  S2[,j]=X2[,tm]
}
cens=runif(n)
ripart=1-exp(-0.01*apply(exp(X2),1,cumsum)*ts/1:grid)

ti=rep(NA,n)
for(i in 1:n) {
  ti[i]=ts[which.min(abs(ripart[,i]-cens[i]))]
}
cens=runif(n,0,Tmax*2)
delta=ifelse(cens>ti,1,0)

ti[cens<ti]=cens[cens<ti]

##
## an important marker

ro=roc(S2,ti,delta,u,tt,s,times)
plot(ro,type="l",col="red")
abline(a=0,b=1)

## an unrelated marker

ro=roc(S1,ti,delta,u,tt,s,times)
plot(ro,type="l",col="red")
abline(a=0,b=1)
**Description**

Compute ROC curve

**Usage**

roc.s(X, etime, status, u=NULL, tt, s, vtimes, fc=NULL)

**Arguments**

- **X**: n by S matrix of longitudinal score/biomarker for i-th subject at j-th occasion (NA if unmeasured)
- **etime**: n vector with follow-up times
- **status**: n vector with event indicators
- **u**: Lower limit for evaluation of sensitivity and specificity. Defaults to max(vtimes[s]) (see below)
- **tt**: Upper limit (time-horizon) for evaluation of sensitivity and specificity.
- **s**: n vector of measurements/visits to use for each subject. all(s<=S)
- **vtimes**: S vector with visit times
- **fc**: Events are defined as fc = 1. Defaults to $I(\cup X(t_j)>cutoff)$

**Details**

ROC curve is defined as the curve given by (1-specificities, sensitivities). Here these are obtained for a time-dependent multiply-measured marker are defined as

$$Se(t,c,s,u) = Pr(f_c(X(t_1),X(t_2),...,X(t_{s_i}))| u <= T <= t),$$

and

$$Sp(t,c,s,u) = 1-Pr(f_c(X(t_1),X(t_2),...,X(t_{s_i})) | T > t)$$

for some fixed f_c, where c is a cutoff. The default for f_c is that a positive diagnosis is given as soon as any measurement among the s considered is above the threshold.

**Value**

A matrix with the following columns:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1-spec</td>
<td>1-Specificities</td>
</tr>
<tr>
<td>sens</td>
<td>Sensitivities</td>
</tr>
</tbody>
</table>
Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References


See Also

`auc`, `butstrap`, `maxauc`

Examples

```r
# parameters
n=100
tt=3
Tmax=10
u=1.5
s=sample(3,n,replace=TRUE)
vtimes=c(0,1,2,5)

# generate data

ngrid=5000
ts=seq(0,Tmax,length=ngrid)
X2=matrix(rnorm(n*ngrid,0,0.1),n,ngrid)
for(i in 1:n) {
sa=sample(ngrid/6,1)
vals=sample(3,1)-1
X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
}
S1=matrix(sample(4,n,replace=TRUE),n,length(vtimes))
S2=matrix(NA,n,length(vtimes))
S2[,1]=X2[,1]
for(j in 2:length(vtimes)) {
tm=which.min(abs(ts-vtimes[j]))
S2[,j]=X2[,tm]
}
cens=runif(n)
ripart=1-exp(-0.01*apply(exp(X2),1,cumsum)*ts/1:ngrid)

Ti=rep(NA,n)
for(i in 1:n) {
Ti[i]=ts[which.min(abs(ripart[,i]-cens[i]))]
}
```
cens=runif(n,0,Tmax*2)
delta=ifelse(cens>Ti,1,0)
Ti[cens<Ti]=cens[cens<Ti]

##
## an important marker

ro=roc.s(S2,Ti,delta,u,tt,s,vtimes)
plot(ro,type="l",col="red")
abline(a=0,b=1)

## an unrelated marker

ro=roc.s(S1,Ti,delta,u,tt,s,vtimes)
plot(ro,type="l",col="red")
abline(a=0,b=1)

---

sensspec

### Sensitivity and Specificity

**Description**

Compute sensitivity and specificity

**Usage**

```r
sensspec(X, etime, status, u=NULL, tt, s, vtimes, cutoff=0, fc=NULL)
```

**Arguments**

- **X**: \( n \) by \( S \) matrix of longitudinal score/biomarker for \( i \)-th subject at \( j \)-th occasion (NA if unmeasured)
- **etime**: \( n \) vector with follow-up times
- **status**: \( n \) vector with event indicators
- **u**: Lower limit for evaluation of sensitivity and specificity. Defaults to \( \text{vtimes}[s] \) (see below)
- **tt**: Upper limit (time-horizon) for evaluation of sensitivity and specificity.
- **s**: Scalar number of measurements/visits to use for each subject. \( s \leq S \)
- **vtimes**: \( S \) vector with visit times
- **cutoff**: cutoff for defining events. Defaults to 0
- **fc**: Events are defined as \( fc = 1 \). Defaults to \( \text{SI}(cup X(t_j) > \text{cutoff})$
Details
Sensitivity and specificities for a time-dependent multiply-measured marker are defined as
\[ Se(t,c,s,u) = Pr(f_c(X(t_1),X(t_2),\ldots,X(t_{s_i})) | u \leq T \leq t), \]
and
\[ Sp(t,c,s,u) = 1 - Pr(f_c(X(t_1),X(t_2),\ldots,X(t_{s_i})) | T > t) \]
for some fixed \( f_c \), where \( c \) is a cutoff. The default for \( f_c \) is that a positive diagnosis is given as soon as any measurement among the \( s \) considered is above the threshold.

Value
A vector with the following elements:

- **sens**: Sensitivity at the cutoff
- **spec**: Specificity at the cutoff

Author(s)
Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

See Also
roc, auc, bootstrap, maxauc

---

sensspec.s  
*Sensitivity and Specificity*

Description
Compute sensitivity and specificity

Usage
sensspec.s(X, etime, status, u=NULL, tt, s, vtimes, cutoff=0, fc=NULL)
Arguments

- **X**: n by S matrix of longitudinal score/biomarker for i-th subject at j-th occasion (NA if unmeasured)
- **etime**: n vector with follow-up times
- **status**: n vector with event indicators
- **u**: Lower limit for evaluation of sensitivity and specificity. Defaults to \( \max(\text{vtimes}[s]) \) (see below)
- **tt**: Upper limit (time-horizon) for evaluation of sensitivity and specificity.
- **s**: n vector of measurements/visits to use for each subject. all(s<=S)
- **vtimes**: S vector with visit times
- **cutoff**: cutoff for defining events. Defaults to \( \emptyset \)
- **fc**: Events are defined as \( f_c = 1 \). Defaults to \( \{ \cup \{ X(t_j) > \text{cutoff} \} \} \)

Details

Sensitivity and specificities for a time-dependent multiply-measured marker are defined as

\[ Se(t,c,s,u) = \Pr(f_c(X(t_1),X(t_2),...,X(t_s_i)) | u \leq T \leq t), \]

and

\[ Sp(t,c,s,u) = 1-\Pr(f_c(X(t_1),X(t_2),...,X(t_s_i)) \mid T > t) \]

for some fixed \( f_c \), where \( c \) is a cutoff. The default for \( f_c \) is that a positive diagnosis is given as soon as any measurement among the \( s \) considered is above the threshold.

Value

A vector with the following elements:

- **sens**: Sensitivity at the cutoff
- **spec**: Specificity at the cutoff

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References


See Also

*roc, auc, bootstrap, maxauc*
Index

auc, 2, 5, 9, 11, 13, 17, 19, 21, 23, 26, 28, 29

butstrap, 2, 4, 11, 13, 17, 23, 26, 28, 29
butstrap.nri, 6, 15
butstrap.s, 8, 19

maxauc, 2, 5, 9, 10, 11, 13, 23, 26, 28, 29
maxauc.s, 12

nri, 7, 14

plotAUC, 16
plotAUC.s, 18
plotROC, 20

roc, 2, 5, 9, 17, 21, 22, 28, 29
roc., 25
roc.s, 19, 21
roc.s (roc.), 25

sensspec, 27
sensspec.s, 28