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 Under the Curve, and Net Reclassification Indexes for repeated measures. It is based on methods in Barbati and Farcomeni (2017) <doi:10.1007/s10260-017-0410-2>.

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 *AUC*

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

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods & Applications*, in press

See Also

roc, *butstrap*, *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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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)
```

```
auc(ro)
```

`butstrap` *Bootstrapping AUC*

Description

Boostrap the AUC for significance testing and confidence interval calculation

Usage

```
butstrap(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 `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
`auc1`	AUC for the original data set
`B`	Number of bootstrap replicates. Defaults to `50`
`fc`	Events are defined as fc = 1. Defaults to $I(cup X(t_j)>cutoff)$

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:

`p.value`	(Parametric) p-value for H0: AUC=0.5
`se`	Standard deviation of the AUC replicates
`ci.np`	Non-parametric 95% confidence interval for AUC
`ci.par`	Parametric 95% confidence interval for AUC

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

roc, *auc*, *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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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 unimportant marker

ro=roc(S1,Ti,delta,u,tt,s,vtimes)
but=butstrap(S1,Ti,delta,u,tt,s,vtimes,ro)
```

`butstrap.nri` *Bootstrapping NRI*

Description

Boostrap the AUC for significance testing and confidence interval calculation

Usage

```
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 boostrap? `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

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

nri

Examples

```
# 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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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` *Bootstrapping AUC*

Description

Boostrap 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 $I(cup X(t_j)>cutoff)$

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:

p.value	(Parametric) p-value for H0: AUC=0.5
se	Standard deviation of the AUC replicates
ci.np	Non-parametric 95% confidence interval for AUC
ci.par	Parametric 95% confidence interval for AUC

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

roc, *auc*, *maxauc*

Examples

```
# 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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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 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:

`beta`	Beta coefficients for the optimal score
`score`	Optimal score

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

auc, *butstrap*, *maxauc*

Examples

```
# 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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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*

Description

Compute optimal score for AUC

Usage

```
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)

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
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

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

auc, *butstrap*, *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)-1
X2[i,1:sa[1]]=vals[1]+X2[i,1:sa[1]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

butstrap.nri

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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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))
nri(risk1,risk2,Ti,delta,u,tt)
```

---

plotAUC *AUC as a function of time*

Description

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

Usage

```
plotAUC(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	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 $I(cup X(t_j)>cutoff)$
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

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods & Applications*, in press

See Also

roc, *butstrap*, *auc*

Examples

```
# 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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods & Applications*, in press

See Also

roc.s, *butstrap.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

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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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

---

plotROC *Plot ROC*

Description

Plot the ROC curve

Usage

```
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

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods & Applications*, in press

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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}
```

```
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)
plotROC(ro)

## an unrelated marker

ro=roc(S1,Ti,delta,u,tt,s,vtimes)
plotROC(ro)
```

`roc` *ROC curve*

Description

Compute ROC curve

Usage

```
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 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

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:

1-spec	1-Specificities
sens	Sensitivities

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

auc, *butstrap*, *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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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)
plot(ro,type="l",col="red")
abline(a=0,b=1)

## an unrelated marker

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

roc. *ROC curve*

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:

1-spec	1-Specificities
sens	Sensitivities

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

auc, *butstrap*, *maxauc*

Examples

```
# 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]]
X2[i,(sa[1]+1):ngrid]=vals[1]+sample(c(-2,2),1)+X2[i,(sa[1]+1):ngrid]
}

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

```
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 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
cutoff	cutoff for defininings events. Defaults to 0
fc	Events are defined as fc = 1. Defaults to $I(cup X(t_j)>cutoff)$

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 <= 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 vector with the following elements:

`sens`	Sensitivity at the cutoff
`spec`	Specificity at the cutoff

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

roc, *auc*, *butstrap*, *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(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 defininining events. Defaults to 0
fc	Events are defined as fc = 1. Defaults to $I(cup X(t_j)>cutoff)$

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 <= 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 vector with the following elements:

sens	Sensitivity at the cutoff
spec	Specificity at the cutoff

Author(s)

Alessio Farcomeni <alessio.farcomeni@uniroma1.it>

References

Barbati, G. and Farcomeni, A. (2017) Prognostic assessment of repeatedly measured time-dependent biomarkers, with application to dilated cardiomuopathy, *Statistical Methods \& Applications*, in press

See Also

roc, *auc*, *butstrap*, *maxauc*

Index