

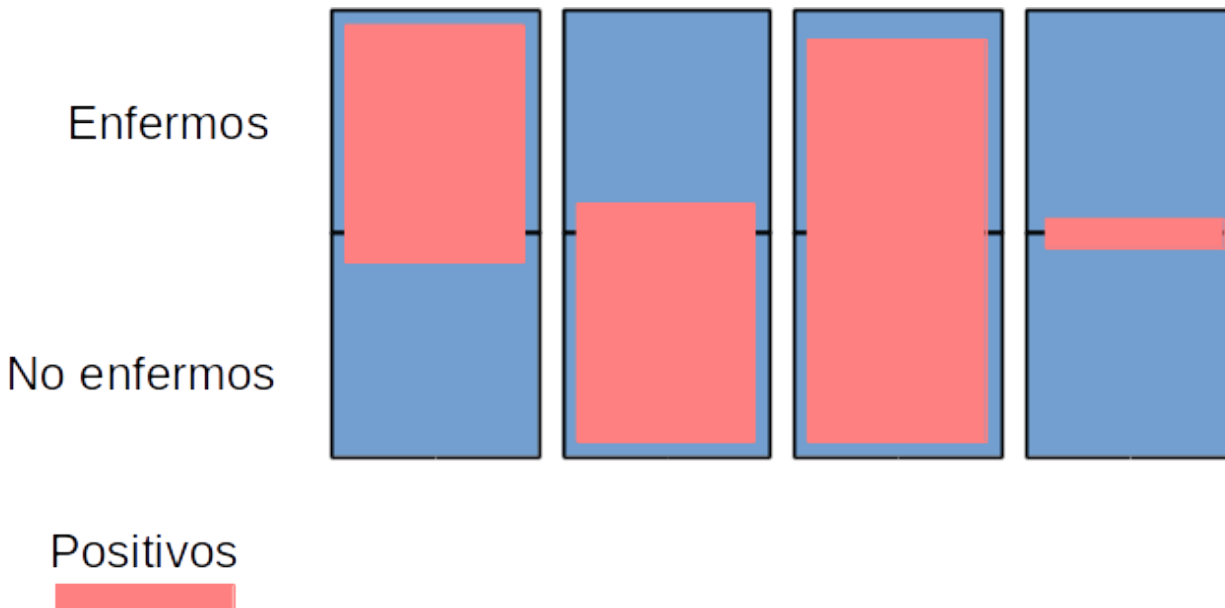
CURVAS ROC

007R team

Abril 2016

Curvas ROC en análisis de pruebas diagnósticas.

En análisis ROC se trabaja con los conceptos de *sensibilidad* y *especificidad*.



Tenemos dos grupos de personas: enfermos y no enfermos. Con todos se ensaya una prueba diagnóstica (p.e. se mide el nivel de glucosa en sangre como indicador de diabetes). Los pacientes que resulten positivo los llamamos **P** y los que resulten negativos **N**.

- La sensibilidad es el cociente entre (P)ositivos / (E)nfermos, es decir la proporción de positivos en la prueba diagnóstica que realmente están enfermos (positivos verdaderos TP).
- La especificidad mide la proporción de negativos entre los no enfermos, (N)egativos / (N)o (E)nfermos. Verdaderos negativos (TN).

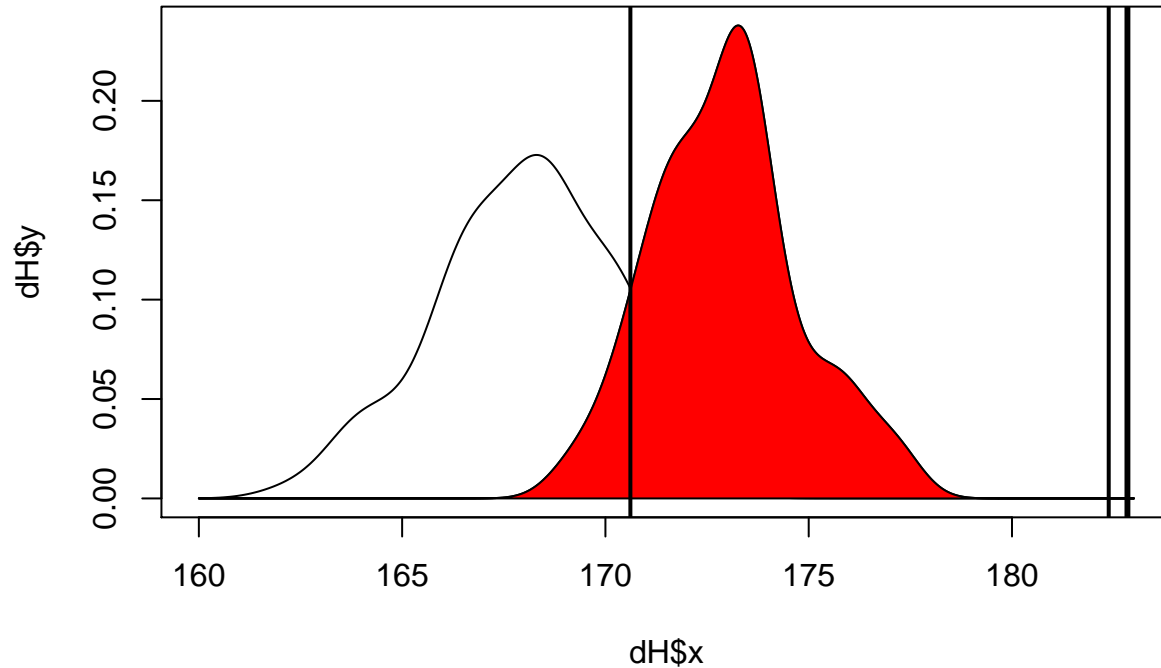
Interesa una prueba diagnóstica con alta sensibilidad, es decir que la mayoría de pacientes enfermos den positivo para esa prueba. También interesa que la especificidad sea alta, es decir que la mayoría de pacientes no enfermos den negativo.

```
altH<-rnorm(100, 173, 2)
altM<-rnorm(100, 168, 2)
dH<-density(altH, from=160, to=183, n=1000)
dM<-density(altM, from=160, to=183, n=1000)
mydf<-data.frame(dH$x, dM$x, dH$y, dM$y)
incr<-0
intersX<-dH$x[as.logical(abs(diff(dH$y < dM$y))))]+incr
#intersY<-dH$y[as.logical(abs(diff(dH$y < dM$y))))]
plot(dH$x, dH$y, ylim= range(0, pmax(dH$y,dM$y)), xlim=range(dH$x), type="l")
```

```

lines(dM$x, dM$y)
#polygon(dH$x, pmin(dH$y, dM$y), col="red")
fin<-length(dH$x[(dH$x <= intersX)])
polygon(c(dH$x[1:fin], dH$x[fin]), c(dH$y[1:fin], 0), col="red")
FP<-dH$x[(dH$x > intersX)]
polygon(c(dH$x[fin:length(dH$x)], dH$x[fin]), c(dM$y[fin:length(dH$x)], 0), col="green")
abline(v=intersX, lwd=2)

```



```

# Altura como predictor
mydata<-data.frame(c(altM, altH), as.factor(rep(c(1,2), times=1, each=100)))
colnames(mydata)<-c("altura", "genero")
library(ROCR)

```

```
## Loading required package: gplots
```

```
##
```

```
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
```

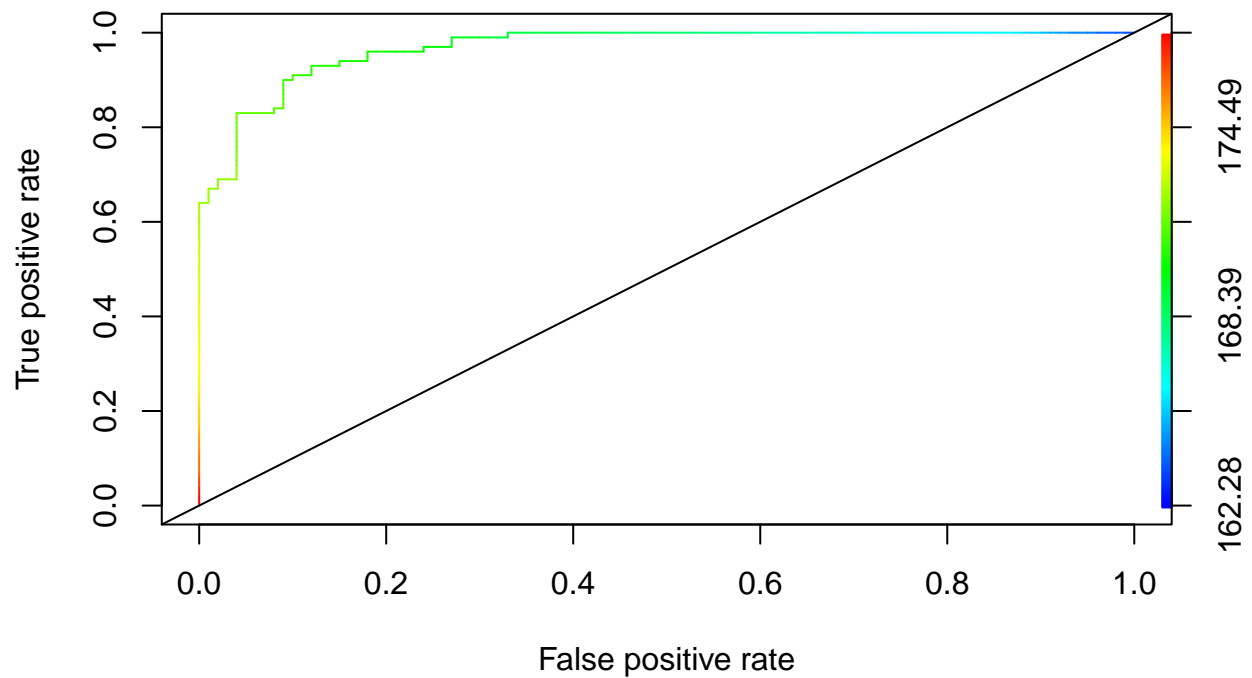
```
##
```

```
## lowess
```

```

pred<-prediction(mydata$altura, mydata$genero)
perf<-performance(pred, measure = "tpr", x.measure= "fpr")
plot(perf, colorize=TRUE)
abline(a=0, b=1)

```

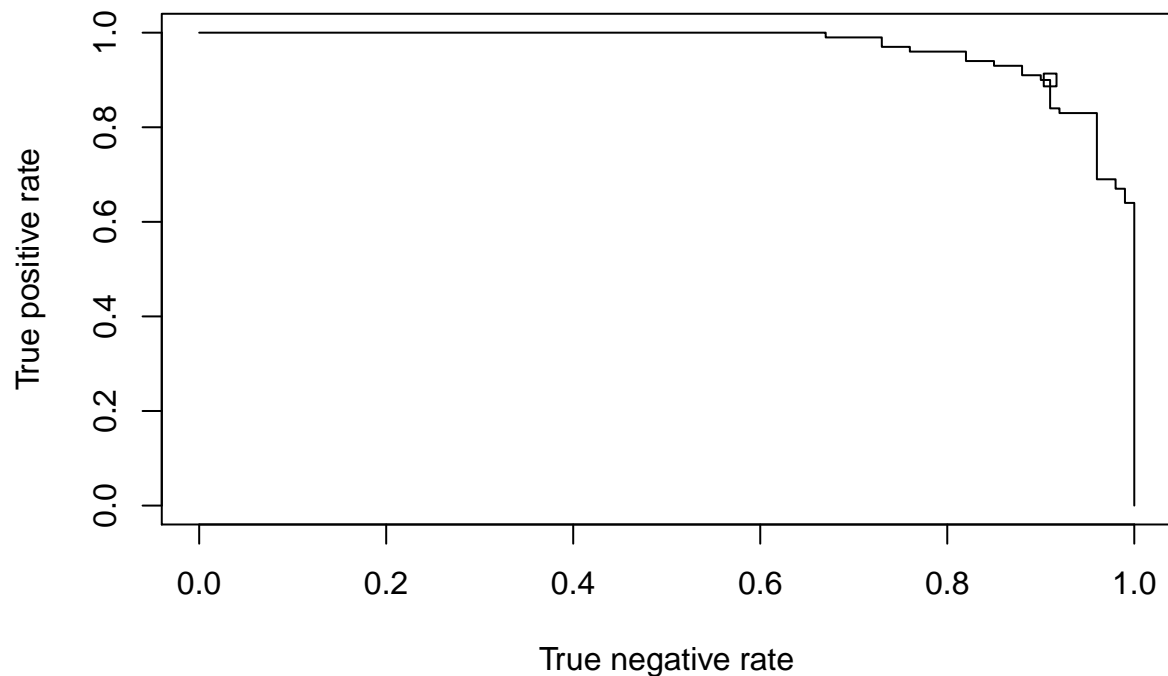


```
AUC<-performance(pred, measure = "auc")
AUCaltura<-AUC@y.values
```

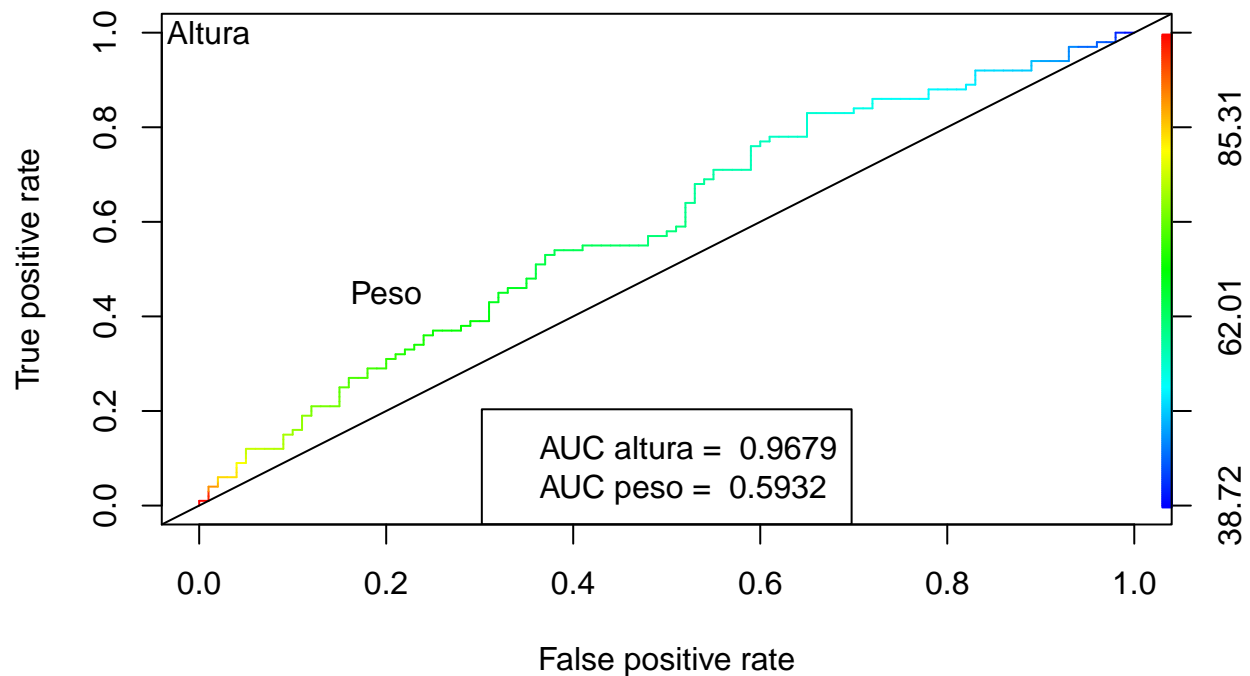
```
# calcular el optimo cutoff
perf<-performance(pred, measure = "tpr", x.measure = "tnr")
plot(perf)
perf@alpha.values[[1]][which.max(perf@x.values[[1]]+perf@y.values[[1]])]
```

```
## [1] 170.8173
```

```
xpoint<-perf@x.values[[1]][which.max(perf@x.values[[1]]+perf@y.values[[1]])]
ypoint<-perf@y.values[[1]][which.max(perf@x.values[[1]]+perf@y.values[[1]])]
points(xpoint,ypoint, type="p", pch=22)
```



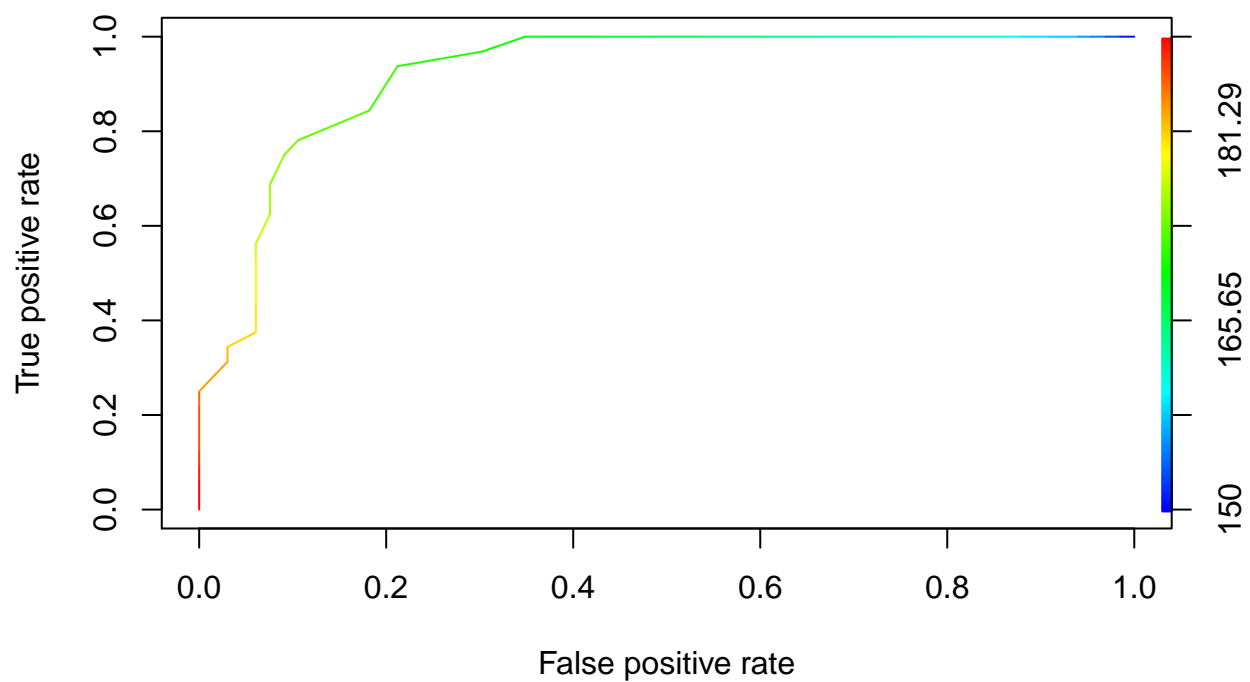
```
#peso como predictor
IMC = rnorm(100, 22, 4)
pesoH <- IMC*(altH/100)^2
pesoM <- IMC*(altM/100)^2
indH <- pesoH
indM <- pesoM
mydata2<-data.frame(c(altM, altH), c(indM, indH), as.factor(rep(c(1,2), times=1, each=100)))
colnames(mydata2)<-c("altura", "ind", "genero")
pred2<-prediction(mydata2$ind, mydata2$genero)
perf2<-performance(pred2, measure = "tpr", x.measure= "fpr")
plot(perf2, colorize=TRUE, add=FALSE)
abline(a=0, b=1)
AUC<-performance(pred2, measure = "auc")
AUCpeso<-AUC@y.values
leg1<-paste("AUC altura = ", AUCaltura)
leg2<-paste("AUC peso = ", AUCpeso)
legend("bottom", c(leg1, leg2))
text(0.01, 1, "Altura")
text(0.2, 0.45, "Peso")
```



```
# Datos reales
biom<-read.table("http://ares.inf.um.es/00RTeam/datos/biom2003.dat")
head(biom)
```

```
##  Grupo Peso Altura Pie Hombros Brazos Caderas Sexo Ojos Tipo
## 1      1   60   163  37    41    68    95    1    1    2
## 2      1   52   166  37    37    70    87    1    2    1
## 3      1   61   172  39    39    69    91    1    2    1
## 4      1   73   181  43    50    78   101    2    2    1
## 5      1   53   172  39    39    72    89    1    1    1
## 6      1   63   169  40    37    66    96    1    2    1
```

```
predReal<-prediction(biom$Altura, biom$Sexo)
perfReal<-performance(predReal, measure = "tpr", x.measure = "fpr")
plot(perfReal, colorize=TRUE, add=FALSE)
```



[Ver este enlace.](#)