If you read this file as R code, make sure you have installed knitr in order to correctly run the code and view results in a html doc (ie with Knit HTML in RStudio)

# Measuring predictive performance and screening with ROC in R (ROCR, pROC)

author: Pedro.Concejero at Telefónica Investigación y Desarrollo (tid.es) date: 27th February 2014

gRupo R madRid <http://r-es.org/Grupo+de+Inter%C3%A9s+Local+de+Madrid+-+GIL+Madrid&structure=Comunidad>

# Very special thanks to

* Luis Mariano Esteban from Universidad Zaragoza
* My colleagues Paulo, Goyo, David & Fernando from Telefónica I+D
* gRupo R madRid participants

# What is ROC

ROC means Receiver Operating Characteristic (also Curve)

**Don't google 'ROC'**, you will go to pharmaceutical company (or to Redeeming Our Communities)

**Google 'ROC curve'**

A good source: wikipedia

<http://en.wikipedia.org/wiki/Receiver_operating_characteristic>

(y en español) <http://es.wikipedia.org/wiki/Curva_ROC>

# ROC libraries in R

Install the two libraries we will use throughout the presentation

¡CAVEAT!

ROCR (all uppercase)

pROC (first letter is lowercase)

sqldf and gmodels also useful later

library(ROCR)

## Loading required package: gplots  
##   
## Attaching package: 'gplots'  
##   
## The following object is masked from 'package:stats':  
##   
## lowess

library(pROC)

## Warning: package 'pROC' was built under R version 3.1.3

## Type 'citation("pROC")' for a citation.  
##   
## Attaching package: 'pROC'  
##   
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(sqldf)

## Loading required package: gsubfn  
## Loading required package: proto  
## Loading required package: RSQLite  
## Loading required package: DBI

library(gmodels)

## Warning: package 'gmodels' was built under R version 3.1.3

##   
## Attaching package: 'gmodels'  
##   
## The following object is masked from 'package:pROC':  
##   
## ci

ROCR site <http://rocr.bioinf.mpi-sb.mpg.de/>

pROC site <http://web.expasy.org/pROC/>

# PSA data

Data from Etzioni, Pepe et al. (1999): Incorporating the Time Dimension in Receiver Operating Characteristic Curves: A Case Study of Prostate Cancer. Med Decis Making 1999; 19:242-251

PSA = "Prostate Specific Antigen"

<http://labs.fhcrc.org/pepe/dabs/datasets.html>

# PUT YOUR WORKING DIRECTORY HERE!  
  
setwd("C:/Users/pedroc/Desktop/madRid-R/ROC")  
  
datos <- read.table(file = "http://research.fhcrc.org/content/dam/stripe/diagnostic-biomarkers-statistical-center/files/psa2b.csv",  
 sep = ",",  
 header = TRUE)

# PSA data

<http://labs.fhcrc.org/pepe/book/data/psa2b_desc.txt>

d - diagnostic | fpsa - free PSA | tpsa - total PSA

str(datos)

## 'data.frame': 683 obs. of 6 variables:  
## $ id : int 1 2 2 2 3 3 3 3 3 3 ...  
## $ d : int 1 1 1 1 0 0 0 0 0 0 ...  
## $ t : num -4.482 -4.498 -1.336 -0.356 -3.381 ...  
## $ fpsa: num 3.525 1.104 2.405 2.428 0.226 ...  
## $ tpsa: num 14.82 5.54 8.15 10.71 0.94 ...  
## $ age : num 67.6 70.2 73.3 74.3 55 ...

# PSA data

Apart from that we have repeated PSA measures in time -t- and age at the moment of test. But we are not going to use the time dimension for our ROC -will use simplest model: obtain the first PSA measure but you must know diagnostic might be done at the end of the process

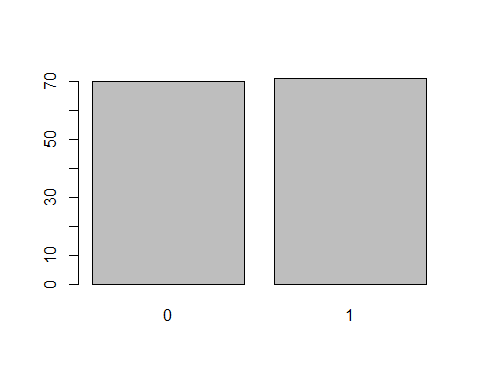
datos.originales <- datos  
  
datos <- sqldf("select id, d, min(t), fpsa, tpsa, age from 'datos.originales'  
 group by id")

## Loading required package: tcltk

# PSA data

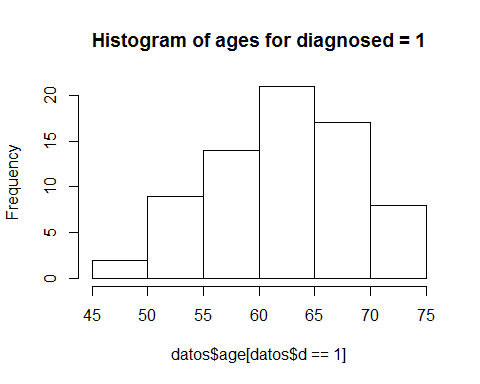
This study is an age-matched case-control design: each diagnosed case was assigned a control matched to case by date of birth. There are 70 of each group. One of the main advantages of ROC: it is applicable to any study **independently of base rate**. In this case: 50%.

barplot(table(datos$d))

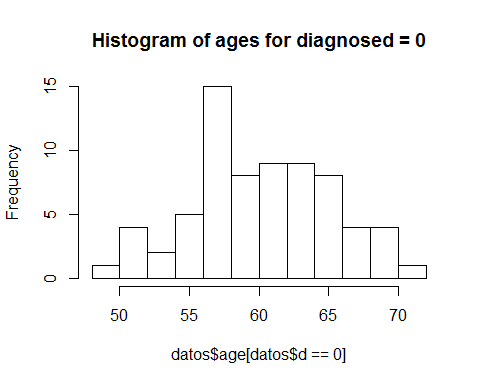


# PSA data

# Diagnosed = 1  
hist(datos$age[datos$d == 1],  
 main = "Histogram of ages for diagnosed = 1")

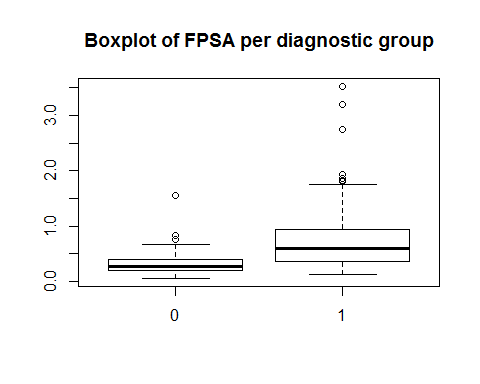
 \*\*\*

# Diagnosed = 0  
hist(datos$age[datos$d == 0],  
 main = "Histogram of ages for diagnosed = 0")

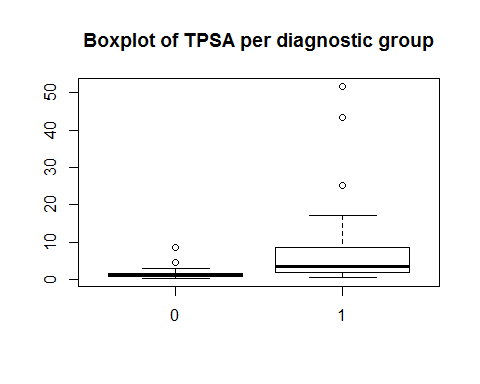


# PSA data

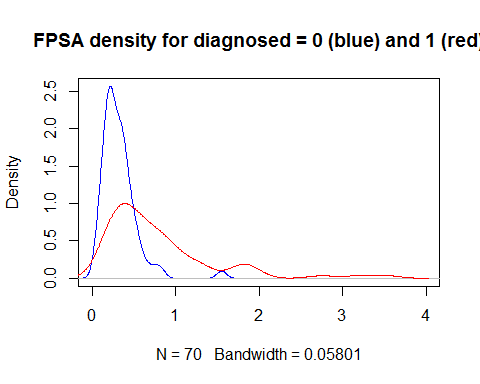
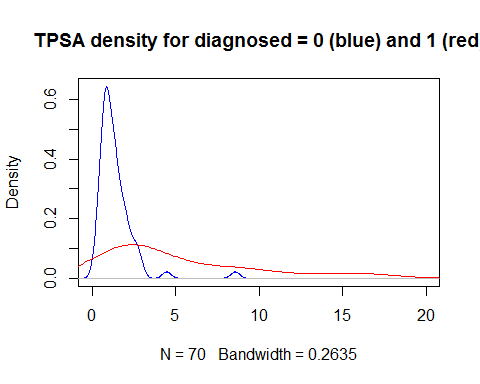
boxplot(datos$fpsa ~ datos$d,  
 main = "Boxplot of FPSA per diagnostic group")



boxplot(datos$tpsa ~ datos$d,  
 main = "Boxplot of TPSA per diagnostic group")

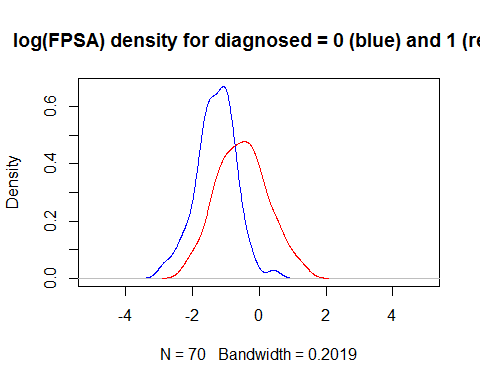
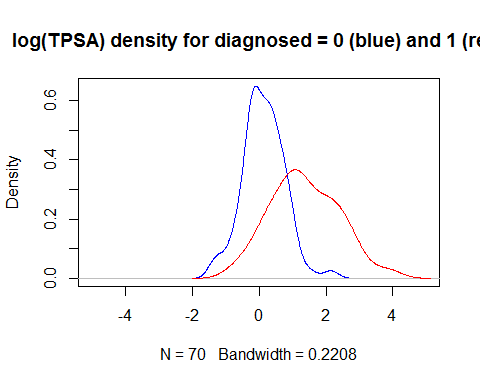


# PSA data

 \*\*\* 

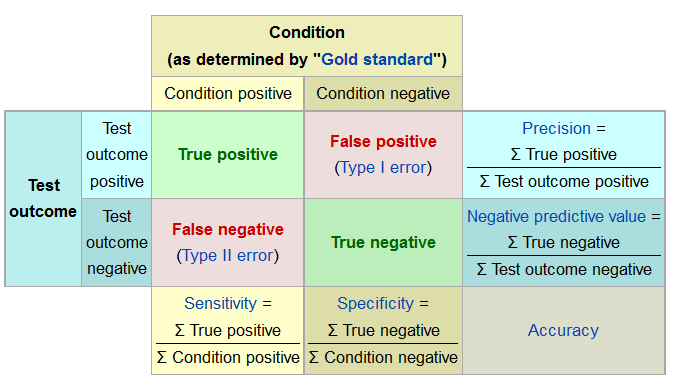
# PSA data

More usually, we will use a log transformation

 \*\*\* 

# The 2 x 2 table for each cutoff point

Thanks english wikipedia! <http://en.wikipedia.org/wiki/Receiver_operating_characteristic#Basic_concept>



# TPSA cutoff point 2

Performance at different cutoff points. Let's see 2 for tpsa

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Col Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 141   
##   
##   
## | datos$cond   
## datos$scree | a-cond pos | b-cond neg | Row Total |   
## -------------|------------|------------|------------|  
## a-test pos | 52 | 12 | 64 |   
## | 0.732 | 0.171 | |   
## -------------|------------|------------|------------|  
## b-test neg | 19 | 58 | 77 |   
## | 0.268 | 0.829 | |   
## -------------|------------|------------|------------|  
## Column Total | 71 | 70 | 141 |   
## | 0.504 | 0.496 | |   
## -------------|------------|------------|------------|  
##   
##

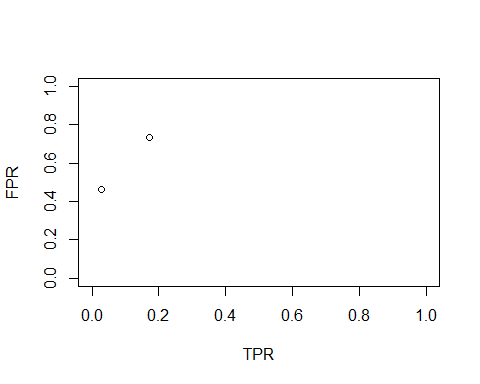
# TPSA cutoff point 4

Performance at different cutoff points. Let's see 4 for tpsa

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Col Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 141   
##   
##   
## | datos$cond   
## datos$scree | a-cond pos | b-cond neg | Row Total |   
## -------------|------------|------------|------------|  
## a-test pos | 33 | 2 | 35 |   
## | 0.465 | 0.029 | |   
## -------------|------------|------------|------------|  
## b-test neg | 38 | 68 | 106 |   
## | 0.535 | 0.971 | |   
## -------------|------------|------------|------------|  
## Column Total | 71 | 70 | 141 |   
## | 0.504 | 0.496 | |   
## -------------|------------|------------|------------|  
##   
##

# Plot ROC with ROCR

ROC is **a set of points** in the square space (0, 1) x (0, 1) where each point is the pair (FPR: FALSE Positives Ratio - x axis, TPR: TRUE Positives Ratio - y axis)



# ROC with ROCR - example participant's id

prediction creates the predictor + criterion object

performance creates the object with measures **from prediction object**

Let's use this for the id - just in case it predicts anything

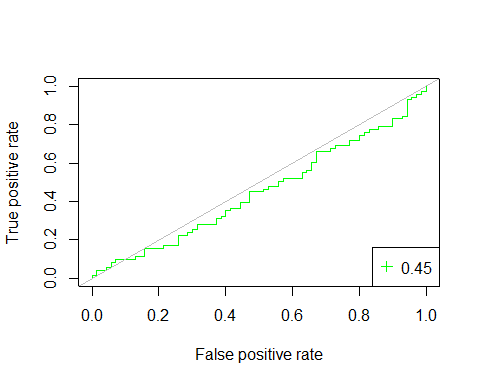
pred.z.01 <- prediction(datos$id, datos$d)  
  
# uso: performance(prediction\_object, "tpr", "fpr") creates the object with performance metrics  
# TPR: True Positive Ratio  
# FPR: False Positive Ratio  
  
perf.z.01 <- performance(pred.z.01, "tpr", "fpr")

# ROC for participant's id

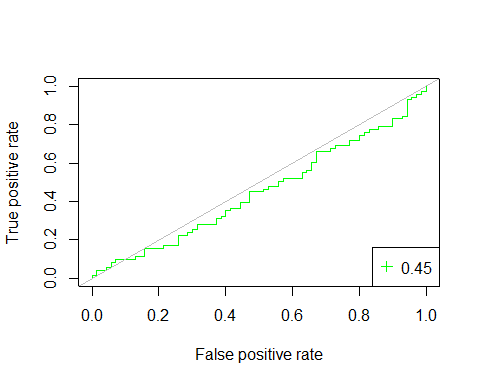
We also plot the null predictive performance (when TPR = FPR; or diagonal in ROC space).

AUC = Area Under Curve (0.5 = null predictive perf. 1 = perfect predictive perf)

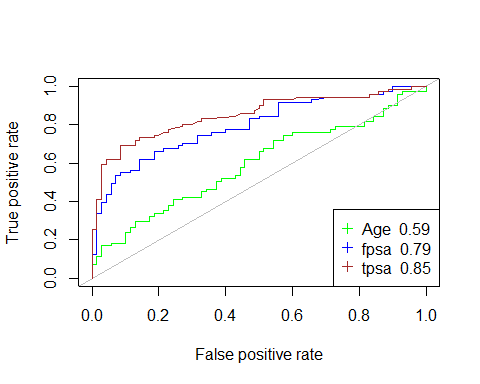
plot.new()  
plot(perf.z.01, col = "green")   
abline(0, 1,   
 col = "grey")  
  
auc.z.01 <- performance(pred.z.01, "auc")  
  
legend("bottomright",   
 paste(round(as.numeric(auc.z.01@y.values), digits = 2)),   
 col = c("green"),  
 pch = c(3))

 ROC for participant's id ========================================================

AUC = 0.45 -> null predictive perf. (as could be expected)



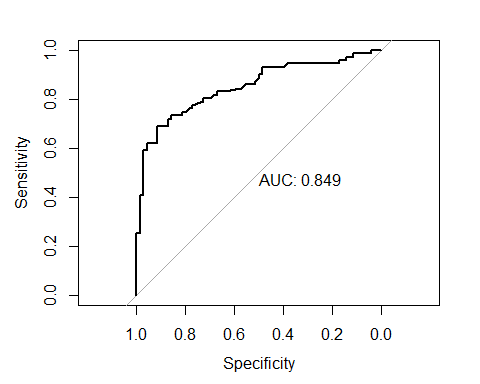
# ROC for fpsa, tpsa and age



# ROC with pROC

Far easier!

pROC::plot.roc(datos$d, datos$tpsa,  
 print.auc = TRUE)



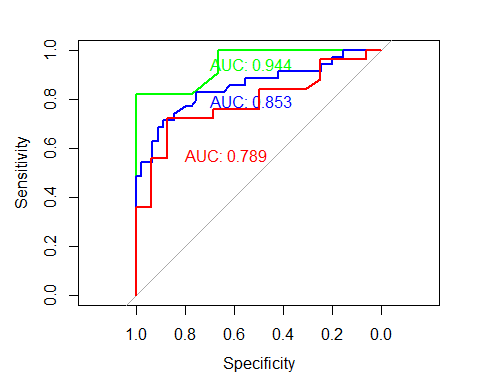
##   
## Call:  
## plot.roc.default(x = datos$d, predictor = datos$tpsa, print.auc = TRUE)  
##   
## Data: datos$tpsa in 70 controls (datos$d 0) < 71 cases (datos$d 1).  
## Area under the curve: 0.849

# ROC with pROC - with age groups

green 45-55 blue 55-65 red >65

##   
## Call:  
## plot.roc.default(x = datos$d[datos$age\_gr == "(45,55]"], predictor = datos$tpsa[datos$age\_gr == "(45,55]"], print.auc = TRUE, col = "green", print.auc.col = "green", print.auc.y = 0.97, print.auc.x = 0.7)  
##   
## Data: datos$tpsa[datos$age\_gr == "(45,55]"] in 9 controls (datos$d[datos$age\_gr == "(45,55]"] 0) < 11 cases (datos$d[datos$age\_gr == "(45,55]"] 1).  
## Area under the curve: 0.944

##   
## Call:  
## plot.roc.default(x = datos$d[datos$age\_gr == "(55,65]"], predictor = datos$tpsa[datos$age\_gr == "(55,65]"], print.auc = TRUE, col = "blue", add = TRUE, print.auc.col = "blue", print.auc.y = 0.82, print.auc.x = 0.7)  
##   
## Data: datos$tpsa[datos$age\_gr == "(55,65]"] in 45 controls (datos$d[datos$age\_gr == "(55,65]"] 0) < 35 cases (datos$d[datos$age\_gr == "(55,65]"] 1).  
## Area under the curve: 0.853



##   
## Call:  
## plot.roc.default(x = datos$d[datos$age\_gr == "(65,100]"], predictor = datos$tpsa[datos$age\_gr == "(65,100]"], print.auc = TRUE, col = "red", add = TRUE, print.auc.col = "red", print.auc.y = 0.6, print.auc.x = 0.8)  
##   
## Data: datos$tpsa[datos$age\_gr == "(65,100]"] in 16 controls (datos$d[datos$age\_gr == "(65,100]"] 0) < 25 cases (datos$d[datos$age\_gr == "(65,100]"] 1).  
## Area under the curve: 0.789

# CONCLUSIONS

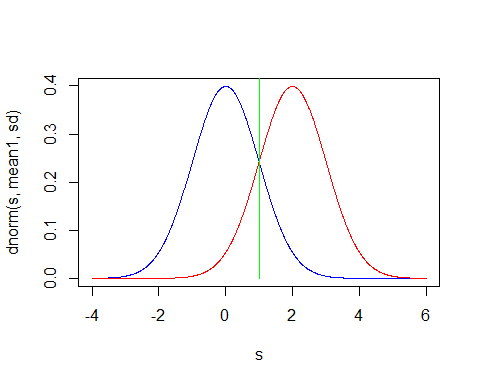
* ROC is standard in medicine - huge experience in using this for screening and also for diagnostic performance of many types of tests
* We have seen one direct measure (tPSA), but you can measure predictive performance of any output of predictive models though you need a binary criterion
* Do not make a confusion with AUC -it is no percentage!! It is an abstract measure of performance
* You can use statistical contrasts to test differences between predictors, as well as predictors versus null predictive performance
* Make your choice pROC is more recent, many useful functions, ROCR maybe makes fancier plots
* R is **absolutely** the best option to make ROC analysis

# APPENDIX - save workspace

save.image("roc\_psa\_wkspace.rdata")

# APPENDIX - Origins in Signal Theory

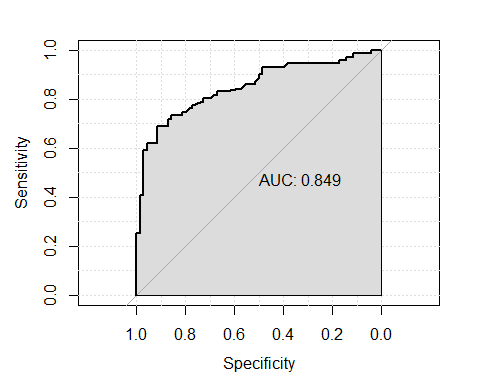
Two normal distributions, same sd different mean



# APPENDIX - ROC with pROC - many useful options

auc.polygon

pROC::plot.roc(datos$d, datos$tpsa,  
 print.auc = TRUE,  
 grid = TRUE,  
# partial.auc = c(100, 90),  
 auc.polygon = TRUE)



##   
## Call:  
## plot.roc.default(x = datos$d, predictor = datos$tpsa, print.auc = TRUE, grid = TRUE, auc.polygon = TRUE)  
##   
## Data: datos$tpsa in 70 controls (datos$d 0) < 71 cases (datos$d 1).  
## Area under the curve: 0.849