# qbs124_hw5_gibran

Gibran Erlangga

5/3/2022

## Question 1

This homework relies of R functions mah and parsROC. Presentation matters. Use the following clean version of the data set

1. (15 points). Apply the LDA to construct the rule for identification of female.

(a) Using the double for loop over all predictors find the best two bone predictors that minimize the total theoretical misclassification error.

```
# initial
d=read.csv("Goldman.csv",stringsAsFactors=F)
nm=names(d[18:(ncol(d)-9)])
sex=as.numeric(as.vector(d[,3]))
```

```
## Warning: NAs introduced by coercion
```

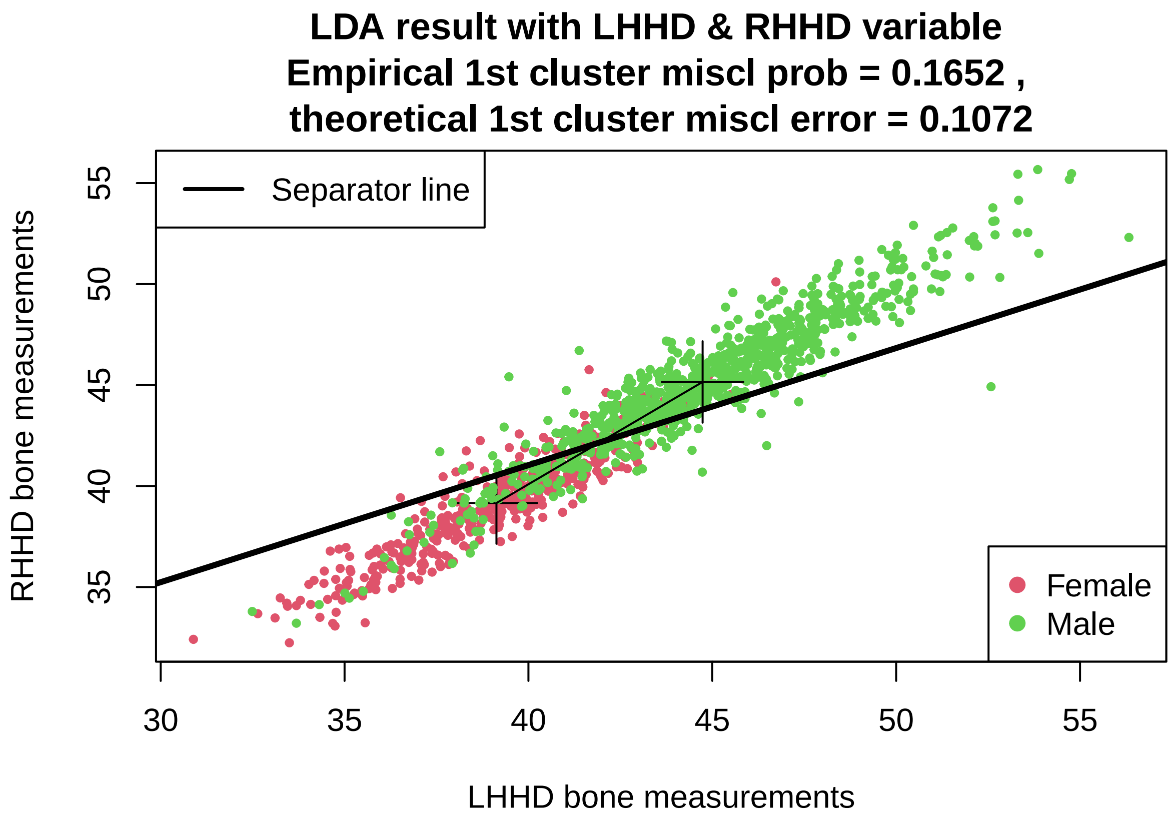
```
d=as.matrix(d[,18:(ncol(d)-9)])
d=d[!is.na(sex),]
sex=sex[!is.na(sex)]
nr=nrow(d);nc=ncol(d)

TheorErr_l=c()
var1=c()
var2=c()
for(ivar in 1:(nc-1))
{
  for (ivar2 in (ivar+1):nc){

    #the two bones
    x=d[,c(ivar,ivar2)]
    #bones and sex
    df=data.frame(cbind(x,sex))
      df=na.omit(df)
      y=as.matrix(df[,3])
    x=as.matrix(df[,c(1,2)])

    male=as.matrix(df[df['sex']==0,c(1,2)])
      female=as.matrix(df[df['sex']==1,c(1,2)])
    mu1=colMeans(male,na.rm = TRUE)
    mu2=colMeans(female,na.rm = TRUE)

    ro=cor(x[,1],x[,2])
    sd1=sd(male,na.rm = TRUE)
    sd2=sd(female,na.rm = TRUE)

    Omega=matrix(c(sd1^2,sd1*sd2*ro,sd1*sd2*ro,sd2^2),2,2)
    a=solve(Omega)%*%(mu1-mu2)
    delta2=t(mu1-mu2)%*%solve(Omega)%*%(mu1-mu2)
    TheorErr_l=append(TheorErr_l,pnorm(-sqrt(delta2)/2))
    var1=append(var1,ivar)
    var2=append(var2,ivar2)
  }
}
ibest=which.min(TheorErr_l)
bone1=colnames(d)[var1[ibest]]
bone2=colnames(d)[var2[ibest]]

print(paste("The best two bone predictors is",bone1," & ",bone2, " with score of ", TheorErr_l[which.mi
```

```
## [1] "The best two bone predictors isLHHD & RHHD with score of 0.0511778669619861"
```

(b) Display the points and the classification line (use red color for female and green for male). (c) Compute and display the minimum theoretical and empirical misclassification error.

```
first = cbind(subset(d, select = c(LHHD)), sex)
second = cbind(subset(d, select = c(RHHD)), sex)
co=chol(Omega)

# LHHD
first_female_data <- subset(first, sex==1, select = -c(sex))
first_male_data <- subset(first, sex==0, select = -c(sex))

# RHHD
second_female_data <- subset(second, sex==1, select = -c(sex))
second_male_data <- subset(second, sex==0, select = -c(sex))

mu1 = c(mean(first_female_data, na.rm=T), mean(second_female_data, na.rm=T))
mu2 = c(mean(first_male_data, na.rm=T), mean(second_male_data, na.rm=T))

p1 = na.omit(cbind(first_female_data,second_female_data))
p2 = na.omit(cbind(first_male_data,second_male_data))

sd1 = sd(colMeans(p1, na.rm=T))
sd2 = sd(colMeans(p2, na.rm=T))

n1 = nrow(p1)
n2 = nrow(p2)

p12=rbind(p1,p2)
plot(p12,type="n",xlab="LHHD bone measurements",
     ylab="RHHD bone measurements")
```

```
points(p1[,1],p1[,2],pch=19, cex=0.5, col=2)
points(mu1[1],mu1[2],pch=3,cex=4)
points(p2[,1],p2[,2],pch=19, cex=0.5, col=3)
points(mu2[1],mu2[2],pch=3,cex=4)
legend("bottomright", legend=c("Female", "Male"), col=c(2,3), pch=19)
legend("topleft", legend="Separator line", col=1, lty=1, lwd=2)
# separator line
z=seq(from=0,to=1,length=100)
lines(mu1[1]*z+mu2[1]*(1-z),mu1[2]*z+mu2[2]*(1-z))
a=solve(Omega)%*%(mu1-mu2)
x=seq(from=-100,to=100,length=100)
ma=(as.vector(mu1)+as.vector(mu2))/2
y=ma[2]-a[1]/a[2]*(x-ma[1])
lines(x,y,lwd=3)

i1=1:n1;i2=1:n2
ind=1:(n1+n2);memb=c(rep(1,n1),rep(2,n2))
classR=(p12-(rep(1,n1+n2)%*%t(ma)))%*%a
IFer=sum(classR<0 & memb==1)
EmpErr=IFer/n1
delta2=t(mu1-mu2)%*%solve(Omega)%*%(mu1-mu2)
TheorErr=pnorm(-sqrt(delta2)/2)
title(paste("LDA result with LHHD & RHHD variable","\nEmpirical 1st cluster miscl prob =",round(EmpErr,
```



## Question 2

(15 points). (a) Add to bone measurements quadratic and cross-product terms and repeat the analysis for the best predictor using logistic regression for identification of female.

```
## I commented the code for this question because it takes a long time to finish the double loop. I ran

# d=read.csv("Goldman.csv",stringsAsFactors=F)
# nm=names(d[18:(ncol(d)-9)])
# sex=as.numeric(as.vector(d[,3]))
# d=as.matrix(d[,18:(ncol(d)-9)])
# d=d[!is.na(sex),]
# sex=sex[!is.na(sex)]
# nr=nrow(d);nc=ncol(d)
# AUC = rep(0, nc)
#
#
# for (i in 1:nc) {
#   print(i)
#   temp = d[,i]
#   for (j in 1:nc) {
#     # quadratic -> i == j, cross -> i != j
#       ## cross-product
#     var = d[,i]*d[,j]
#     temp = cbind(temp, var)
#   }
#   x = temp
#   y=sex[!is.na(x)]
#   x=x[!is.na(x)]
#   ni=length(x)
#   n0=sum(1-y, na.rm=T);n1=sum(y, na.rm=T)
# # o=glm(y~x,family=binomial)
#   sod=sort(x)
#   fp0=0
#   for(k in 1:ni) {
#       sens=sum(x<sod[k]&y==1, na.rm=T)/n1
#       fp=sum(x<sod[k]&y==0, na.rm=T)/n0
#       if(k>1) AUC[i]=AUC[i]+sens*(fp-fp0)
#       fp0=fp
#   }
#   }

# AUC result
AUC = c(0.8372412, 0.9059148, 0.9082779, 0.8077102, 0.8002234, 0.8330184, 0.9150301, 0.9210204, 0.825325

AUC
```

```
##  [1] 0.8372412 0.9059148 0.9082779 0.8077102 0.8002234 0.8330184 0.9150301
##  [8] 0.9210204 0.8253250 0.8442765 0.8680080 0.7751884 0.8788605 0.8658543
## [15] 0.7886804 0.8941509 0.8334747 0.8386770 0.9317838 0.9113732 0.9065689
## [22] 0.8069891 0.8453707 0.8275305 0.8321896 0.9321473 0.9179277 0.9068206
## [29] 0.8214697 0.8473318 0.8082330 0.9150256 0.8557736 0.8749032 0.8090764
## [36] 0.9171456 0.8638610 0.8654107 0.6500123 0.7404787 0.7486713 0.8971095
## [43] 0.8855810
```

```
i=1:nc
ibest=i[AUC==max(AUC)]
v1.best=d[,ibest]
nm.best=nm[ibest]
nm.best
```

```
## [1] "RFEB"
```

```
paste("The best predictor is", nm.best)
```

```
## [1] "The best predictor is RFEB"
```

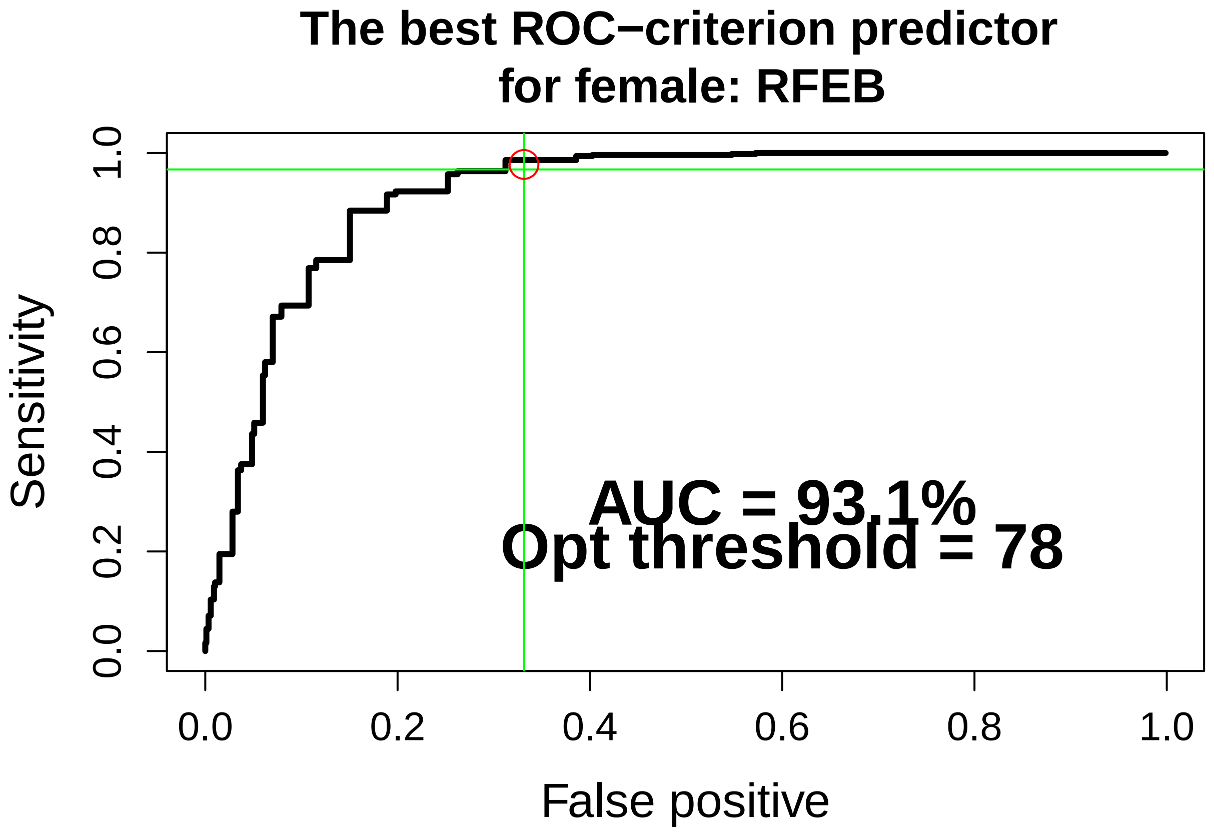
(b) Display the resulted ROC curve AUC and optimal threshold that minimizes the total misclassification error (display the value, the point on the curve and two segments parallel to sensitivity and false positive).

```
# calculate the optimum threshold
y=sex[!is.na(v1.best)]
x=v1.best[!is.na(v1.best)]
n0=sum(1-y, na.rm=T);n1=sum(y, na.rm=T)
ni=length(x)
o=glm(y~x,family=binomial)
print(summary(o))
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2634  -0.4493  -0.1688   0.5077   2.5199
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 32.67795    1.77624   18.40   <2e-16 ***
## x           -0.44485    0.02398  -18.55   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1796.36  on 1376  degrees of freedom
## Residual deviance:  938.97  on 1375  degrees of freedom
## AIC: 942.97
##
## Number of Fisher Scoring iterations: 6
```

```
sod=sort(x)
AUC.best=fp0=0
sens=fp=toter10=rep(0,ni)
```

```r
for(i in 1:ni)
{
    sens[i]=sum(x<sod[i]&y==1, na.rm=T)/n1
    fp[i]=sum(x<sod[i]&y==0, na.rm=T)/n0
    toter10[i]=(1-sens[i])+fp[i]
    if(i>1) AUC.best=AUC.best+sens[i]*(fp[i]-fp0)
    fp0=fp[i]
}

opt.thresh=unique(x[which(toter10==min(toter10, na.rm=TRUE))])
opt = opt.thresh[1]
fp10=sum(x<opt&y==0)/n0
sens10=sum(x<opt&y==0)/n1
d_plot = cbind(x, y)[y==1,][,'x']

# add plot to graph
par(mfrow=c(1,1),mar=c(4.5,4.5,4,1),cex.lab=1.5,cex.main=1.5,cex.axis=1.25)
plot(fp,sens,type="s",lwd=3,xlab="False positive",ylab="Sensitivity",
     main=paste("The best ROC-criterion predictor \n for female:",nm.best), cex=1)
text(.6,.3,paste("AUC = ",round(AUC.best*100,1),"%",sep=""),cex=2,font=2)
text(.6,.2,paste("Opt threshold = ",opt,sep=""),cex=2,font=2)
segments(-1,pnorm(opt, mean=mean(d_plot), sd=sd(d_plot))-0.01,2,
         pnorm(opt, mean=mean(d_plot), sd=sd(d_plot))-0.01, col="green")
segments(fp10,-1,fp10,2,col="green")
points(fp10,pnorm(opt, mean=mean(d_plot), sd=sd(d_plot)), pch=1,cex=2, col ="red")
```


The best ROC−criterion predictor
for female: RFEB
Sensitivity
AUC = 93.1%
Opt threshold = 78
False positive