qbs124_midterm_gibran

Gibran Erlangga

5/13/2022

Immunological measurements, in particular measurements of T-cells, can confirm COVID-19 weeks after the symptoms. To reduce the heterogeneity of the diagnosis the biomedical data scientist incorporates other person-specific variables. File midCOVID.csv contains T-cell count on the % scale (TcellValue), the binary variable GotCovid (GotCovid=0 if the person felt sick but his/her test did not confirm COVID and GotCovid=1 if COVID has been confirmed), Age, Gender (Gender=1: man, Gender=0: woman), BMI and Week as time in weeks the measurements of T-cell counts have been taken after the symptoms among 958 individuals with the purpose of detection of an uninfected individual.

Note: Questions 1 and 2 require par(mfrow=c(1,2),mar=c(4.5,4.5,4,1),cex.lab=1.5,cex.main=1.5). Each question is worth 10 points, the graphics presentation matters.

```r
# import library
suppressMessages(library(tidyverse))

# import data
data <- read.csv('midCOVID.csv')
paste("data dimension (# of rows, # of cols):", dim(data)[1], ",", dim(data)[2])
```

```
## [1] "data dimension (# of rows, # of cols): 958 , 6"
```

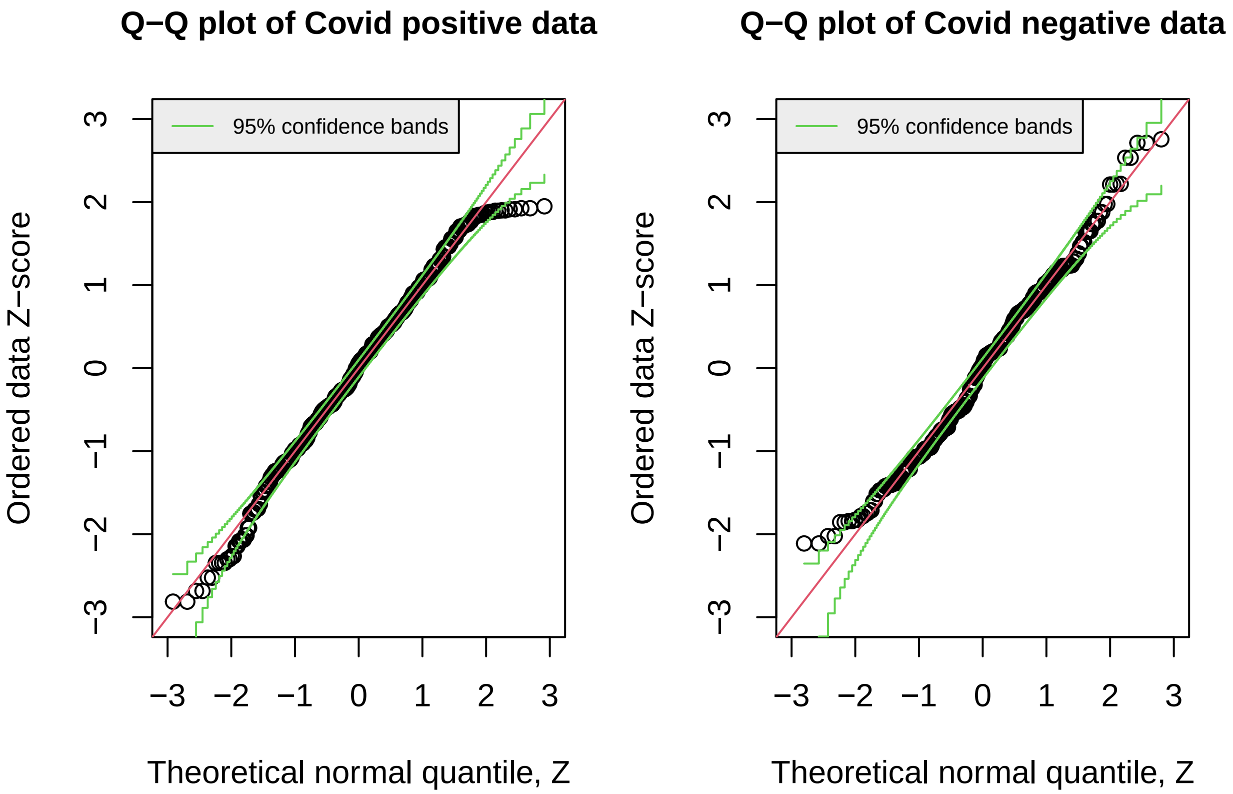
```r
head(data, 2)
```

```
##   TcellValue GotCovid Age Gender BMI Week
## 1      22.73        0  32      1  15   14
## 2      55.21        1  48      0  15    7
```

Question 1

1. Use q-q plots with CBs to confirm that TcellValue in two groups, according to GotCovid, follow normal distribution. Use the R function qqCB.r (provided).

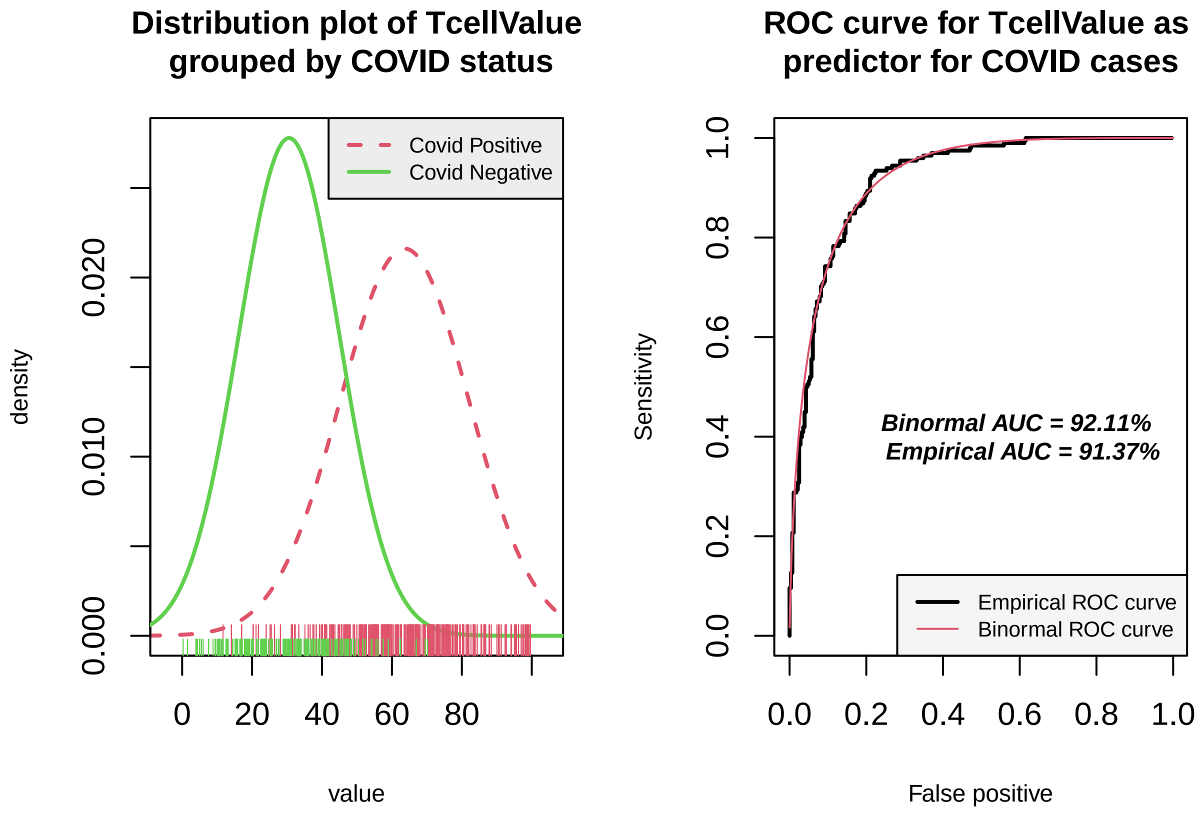
```r
# get TcellValue column for each group
covid_pos_tcell <- data.matrix(data %>%
                                 filter(GotCovid == 1) %>%
                                 select(TcellValue))
covid_neg_tcell <- data.matrix(data %>%
                                 filter(GotCovid == 0) %>%
                                 select(TcellValue))

# covid positive dataset
Y=sort(covid_pos_tcell)
n_p=length(Y)
ii_p=1:n_p;thq_p=qnorm((1:n_p)/n_p)
Z_p=(Y-mean(Y))/sd(Y)
qn_p=qnorm((1:n_p)/n_p)

# covid negative dataset
Y=sort(covid_neg_tcell)
n_n=length(Y)
ii_n=1:n_n;thq_n=qnorm((1:n_n)/n_n)
Z_n=(Y-mean(Y))/sd(Y)
qn_n=qnorm((1:n_n)/n_n)

lambda = 0.95

# plot
par(mfrow=c(1,2),mar=c(4.5,4.5,4,1),cex.lab=1,cex.main=1)
plot(qn_p,Z_p,xlim=c(-3,3),ylim=c(-3,3),
     xlab="Theoretical normal quantile, Z",
     ylab="Ordered data Z-score",
     main="Q-Q plot of Covid positive data")
segments(-5,-5,5,5,col=2)
upB=qnorm(qbeta(.5+lambda/2,shape1=ii_p,shape2=n_p-ii_p+1))
lowB=qnorm(qbeta(.5-lambda/2,shape1=ii_p,shape2=n_p-ii_p+1))
lines(thq_p,upB,type="s",col=3)
lines(thq_p,lowB,type="s",col=3)
legend("topleft", legend="95% confidence bands", col=3, lty=1, lwd=1, cex=.7,
       bg="gray93")

plot(qn_n,Z_n,xlim=c(-3,3),ylim=c(-3,3),
     xlab="Theoretical normal quantile, Z",
     ylab="Ordered data Z-score",
     main="Q-Q plot of Covid negative data")
segments(-5,-5,5,5,col=2)
upB=qnorm(qbeta(.5+lambda/2,shape1=ii_n,shape2=n_n-ii_n+1))
lowB=qnorm(qbeta(.5-lambda/2,shape1=ii_n,shape2=n_n-ii_n+1))
lines(thq_n,upB,type="s",col=3)
lines(thq_n,lowB,type="s",col=3)
legend("topleft", legend="95% confidence bands", col=3, lty=1, lwd=1, cex=.7,
       bg="gray93")
```



2. Plot TcellValue data for the two groups using rug with different colors and the respective normal densities using dnorm command. At the graph at right plot the two ROC curves, one empirical and another binormal to demonstrate how the T-cell count can be used to detect uninfected individuals; show the respective values of AUC.

```
# plot 2 graphs
par(mfrow=c(1,2),mar=c(4.5,4.5,4,1),cex.lab=.75,cex.main=1)

## left graph: distribution graph of TcellValue between 0 and 1 in GotCovid
muY=mean(covid_pos_tcell);sdY=sd(covid_pos_tcell)
muX=mean(covid_neg_tcell);sdX=sd(covid_neg_tcell)
Lr=c(range(covid_neg_tcell)[1]-5,range(covid_pos_tcell)[2]+5)
x=seq(from=Lr[1]-5,to=Lr[2]+5,length=200)
dY=dnorm(x,mean=muY,sd=sdY);dX=dnorm(x,mean=muX,sd=sdX)
matplot(x,cbind(dY,dX),xlim=Lr,lty=c(2,1),type="l",col=c(2,3),lwd=2,
        xlab="value",
        ylab="density",
        main="Distribution plot of TcellValue\n grouped by COVID status")
rug(covid_pos_tcell,lty=1,ticksize=.075, col=2)
rug(covid_neg_tcell,lty=1,ticksize=.04, col=3)
legend("topright",c("Covid Positive","Covid Negative"),lty=c(2,1),lwd=2,
       bg="gray93",cex=0.7, col = c(2,3))

# right graph: plot two ROC curves (empirical and binormal)
covid_status = data$GotCovid
tcell=data$TcellValue
n = length(covid_status)
```

```r
x=seq(from=0,to=200,length=200)

tcell_0 = sort(tcell[covid_status == 0])
n0 = length(tcell_0)
m0=mean(tcell_0);s0=sd(tcell_0)

tcell_1 = sort(tcell[covid_status == 1])
n1=length(tcell_1)
m1=mean(tcell_1);s1=sd(tcell_1)

tcell = sort(tcell)
sens=fp=toter10=rep(NA,n)
AUC=toter=0

# get sensitivity, false positive, total error and AUC values on every
# point from 1 to n
for(i in 1:n) {
  sens[i]= sum(tcell_0 < tcell[i])/n0
  fp[i]=sum(tcell_1 < tcell[i])/n1
  if(i>1) AUC=AUC+(fp[i]-fp[i-1])*sens[i]
}

plot(fp,sens,type="l",lwd=2,xlab="False positive",ylab="Sensitivity",
     main="ROC curve for TcellValue as \npredictor for COVID cases")
lines(pnorm(x,mean=m1,sd=s1),pnorm(x,mean=m0,sd=s0), col=2, lwd=1)
AUC.th=pnorm((m1-m0)/sqrt(s0^2+s1^2))
text(.6,.4,paste("Binormal AUC = ",round(100*AUC.th,2),
                 "% \n Empirical AUC = ",round(100*AUC,2), "%", sep=""),
     cex=.75,font=4)
legend("bottomright",c("Empirical ROC curve","Binormal ROC curve"),
       lty=1,lwd=c(2,1),col=c(1,2),cex=.7,bg="gray96")
```

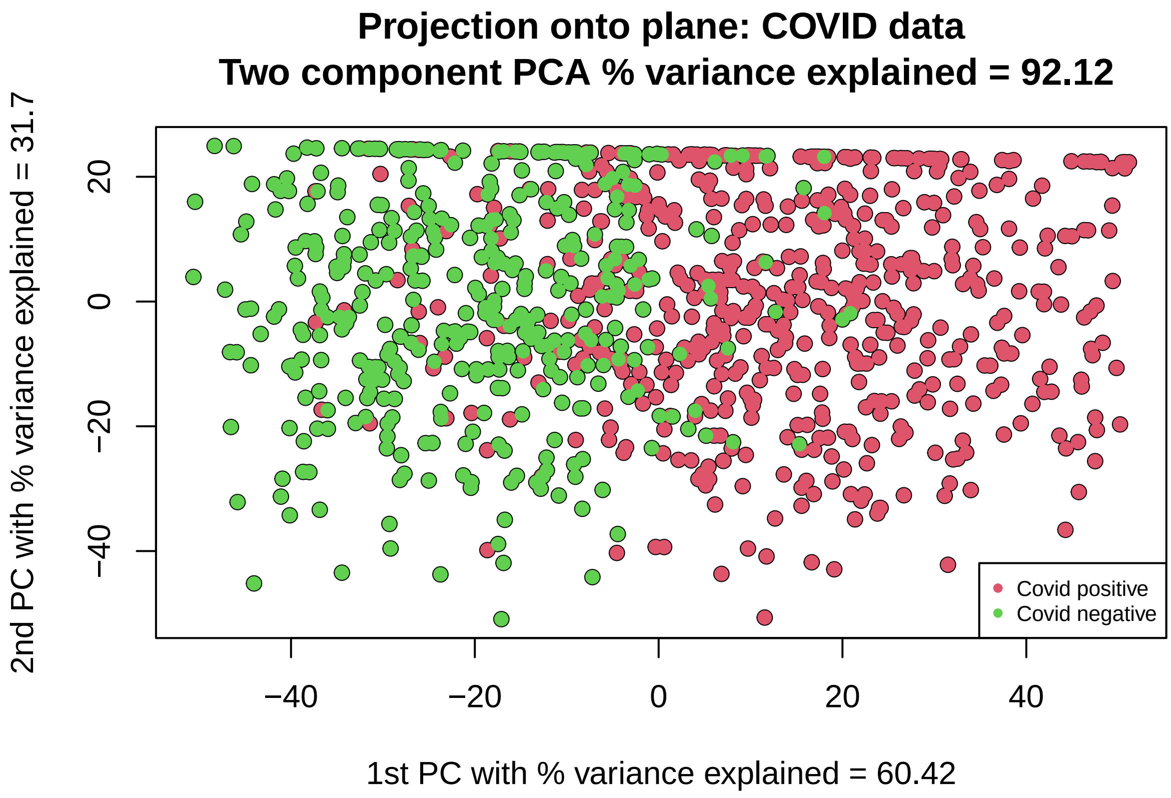


```
# store variables under different name for future uses
fp_tcell = fp; sens_tcell = sens
AUC_tcell = AUC
m1_tcell = m1
m0_tcell = m0
x_tcell = x
s1_tcell = s1
s0_tcell = s0
AUC.th_tcell = AUC.th
```

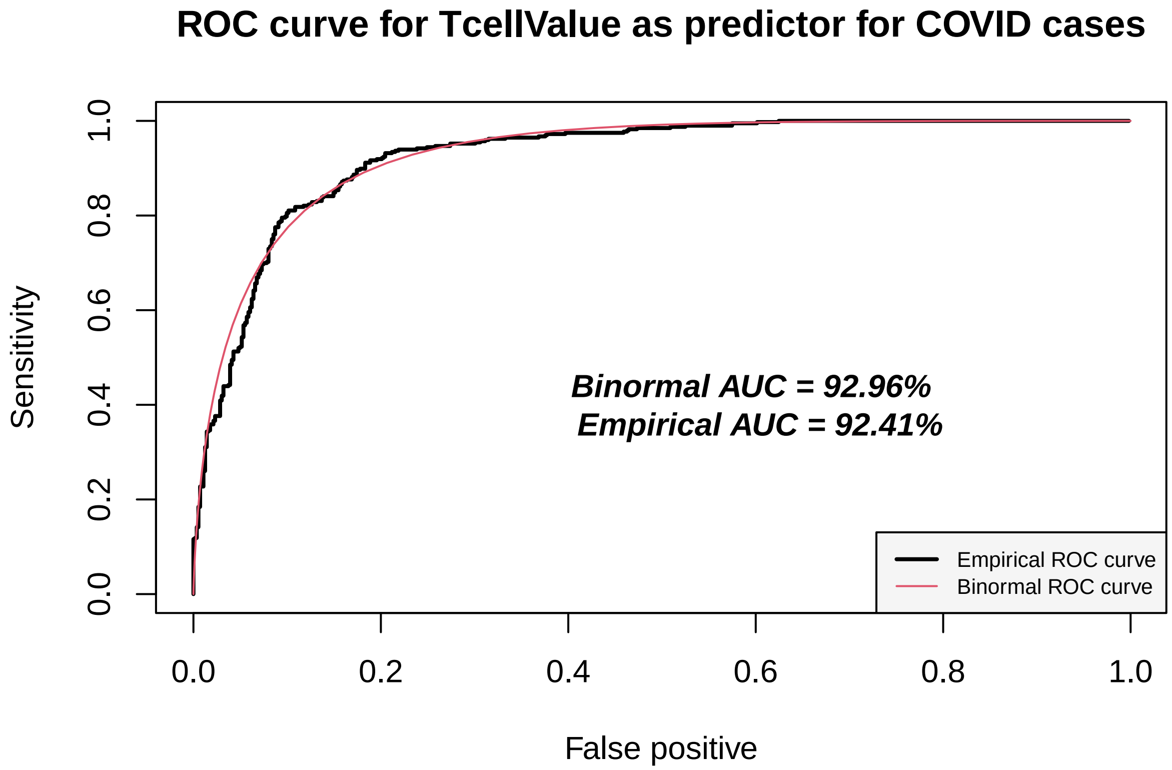
3. Project the personal data onto the plane using PCA and use different colors to display the COVID-19 status. Show % variance explained for each PCA component and the total on the top of the graph. At right, display the empirical and binormal ROC curve using the first two PCA components as linear predictors; show the AUC values on the graph.

```
# plot 2 graphs (no need to be side-by-side)
## first graph: PCA components along with its variance explained %
covid_status <- subset(data, select = c(GotCovid))

# get index for each type
idx_covid_pos <- which(covid_status == 1)
idx_covid_neg <- which(covid_status == 0)

X <- subset(data, select = -c(GotCovid))
X <- data.matrix(X)
```

```
n=nrow(X);m=ncol(X)
Z=X
for(j in 1:m) {
    xj=X[,j]
    Z[,j]=xj-mean(xj)
}

tZZ=t(Z)%*%Z

# par(mfrow=c(1,1),mar=c(4.5,4.5,4,1),cex.lab=1,cex.main=1.25,cex.axis=1.25)
a=eigen(tZZ,symmetric=T)$vectors[,1:2]
Z=X-rep(1,n)%*%t(colMeans(X))
proj=Z%*%a

# compute % variance explained by 1st and 2nd principle components
L12=eigen(tZZ,symmetric=T)$values
var1.expl=L12[1]/sum(L12)*100
var2.expl=L12[2]/sum(L12)*100
var12.expl=(L12[1]+L12[2])/sum(L12)*100
txt1=paste("1st PC with % variance explained =",round(var1.expl,2))
txt2=paste("2nd PC with % variance explained =",round(var2.expl,2))
txt12=paste("Two component PCA % variance explained =",round(var12.expl,2))

# plot
plot(proj,xlab=txt1,ylab=txt2, cex=1)
title(paste("Projection onto plane: COVID data\n",txt12), cex=.75)
points(proj[idx_covid_pos,],col=2,pch=16, cex=1)
points(proj[idx_covid_neg,],col=3,pch=16, cex=1)
legend("bottomright", legend = c("Covid positive", "Covid negative"),
       col=c(2,3), pch = 16, cex=.65)
```



```
# second graph: empirical & binormal ROC curve using PCA as linear
# predictors; show AUC score
y=data$GotCovid
n = length(y)
x=seq(from=-200,to=100,length=1500)

logit = glm(y~proj, family=binomial)
lin_pred = logit$linear.predictors

lin_pred_0 = sort(lin_pred[y == 0])
n0 = length(lin_pred_0)
m0=mean(lin_pred_0);s0=sd(lin_pred_0)

lin_pred_1 = sort(lin_pred[y == 1])
n1=length(lin_pred_1)
m1=mean(lin_pred_1);s1=sd(lin_pred_1)

sod = sort(lin_pred)
sens=fp=toter10=rep(NA,n)
AUC=toter=0

# get sensitivity, false positive, total error and AUC values on every
# point from 1 to n
for(i in 1:n) {
  sens[i]= sum(lin_pred_0 < sod[i])/n0
  fp[i]=sum(lin_pred_1 < sod[i])/n1
  if(i>1) AUC=AUC+(fp[i]-fp[i-1])*sens[i]
}

plot(fp,sens,type="l",lwd=2,xlab="False positive",ylab="Sensitivity",
     main="ROC curve for TcellValue as predictor for COVID cases")
lines(pnorm(x,mean=m1,sd=s1),pnorm(x,mean=m0,sd=s0), col=2, lwd=1)
AUC.th=pnorm((m1-m0)/sqrt(s0^2+s1^2))
text(.6,.4,paste("Binormal AUC = ",round(100*AUC.th,2),
                 "% \n Empirical AUC = ",round(100*AUC,2), "%", sep=""),
     cex=1,font=4)
legend("bottomright",c("Empirical ROC curve","Binormal ROC curve"),lty=1,
       lwd=c(2,1),col=c(1,2),cex=.7,bg="gray96")
```

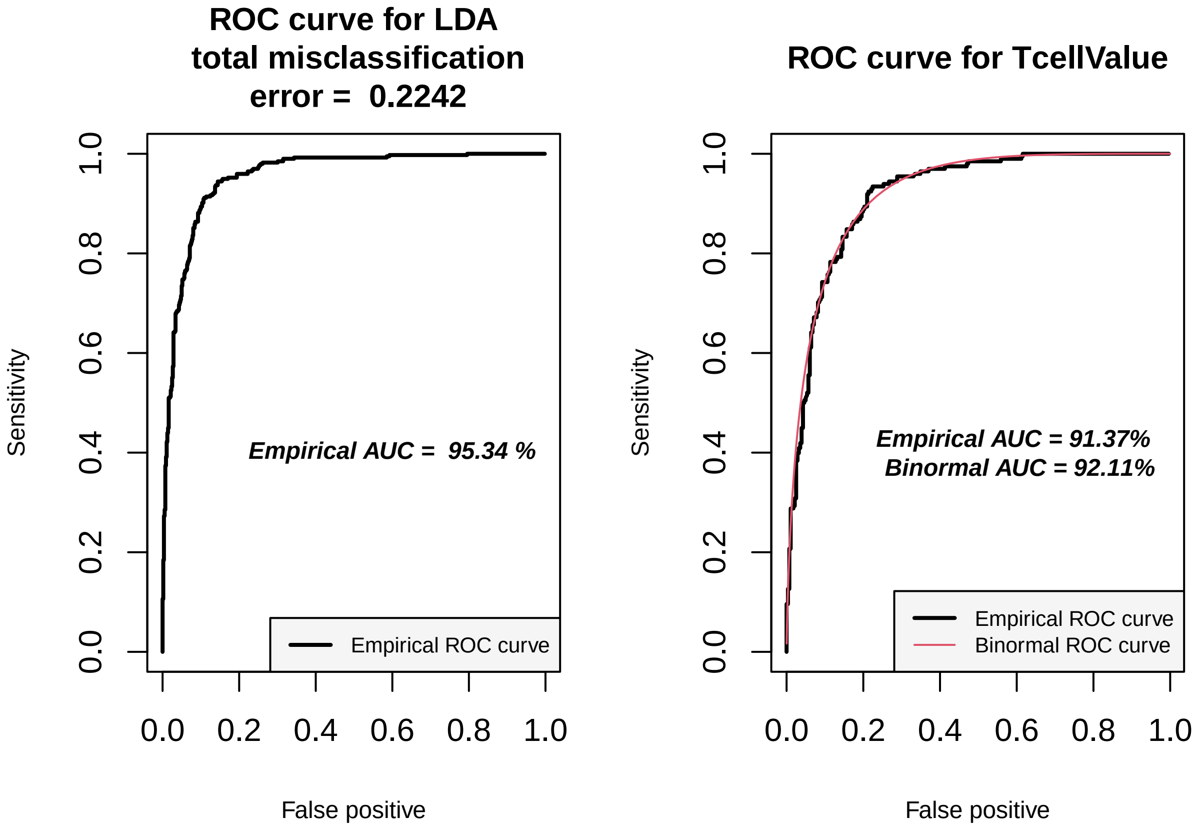


4. Apply linear discriminant analysis to differentiate the groups using all person information. Display the empirical ROC curve using the LDA rule $(y-\mu)^1 a < c$ where c is the threshold. Compute and display the value of the total LDA theoretical misclassification error. Display the ROC curve derived in #2 for comparison and show the empirical AUC value on the graph. Does addition of individual-specific variables improve COVID detection?

```
## plot 1 graph
## empirical ROC curve using stated LDA rule with c as threshold;
## display total LDA theoretical misclassification error

## first graph: PCA components along with its variance explained %
y <- subset(data, select = c(GotCovid))
X <- subset(data, select = -c(GotCovid))
```

```
X <- data.matrix(X)
n = nrow(X)

par(mfrow=c(1,2),mar=c(4.5,4.5,4,1),cex.lab=.75,cex.main=1)

X0=X[y==0,];n0=nrow(X0)
mu0=colMeans(X0); OM0=var(X0)
X1=X[y==1,];n1=nrow(X1)
mu1=colMeans(X1); OM1=var(X1)
Omega=((n0-1)*OM0+(n1-1)*OM1)/(n0+n1-2)
a=solve(Omega)%*%(mu1-mu0)
mu=colMeans(X)
classR=as.vector((X-rep(1,n)%*%t(mu))%*%a)

classR_0 = sort(classR[y == 0])
n0 = length(classR_0)
m0=mean(classR_0);s0=sd(classR_0)

classR_1 = sort(classR[y == 1])
n1=length(classR_1)
m1=mean(classR_1);s1=sd(classR_1)

s.classR=sort(classR)
sens=fp=rep(0,n)

AUC=0
for(i in 1:n) {
  sens[i] = sum(classR_0 < s.classR[i])/n0
    fp[i]= sum(classR_1 < s.classR[i])/n1
    if(i>1) AUC=AUC+(fp[i]-fp[i-1])*sens[i]
}

# total misclassification error
delta2=t(mu0-mu1)%*%solve(Omega)%*%(mu0-mu1)
totmisl=2*pnorm(-.5*sqrt(delta2))

## plot ROC curve for LDA
plot(fp, sens, type="l", lwd=2,xlab="False positive",ylab="Sensitivity",
     main=paste("ROC curve for LDA\n total misclassification\n error = ",
                round(totmisl, 4)),
     lty=1, col=1)
text(.6,.4, paste("Empirical AUC = ", round(100*AUC, 2),"%"),
     cex=.75,font=4)
legend("bottomright",c("Empirical ROC curve"),
       lty=1,lwd=c(2),col=c(1),cex=.7,bg="gray96")

## plot ROC curve from #2
plot(fp_tcell,sens_tcell,type="l",lwd=2,
     xlab="False positive",
     ylab="Sensitivity",
     main="ROC curve for TcellValue")
lines(pnorm(x_tcell,mean=m1_tcell,sd=s1_tcell),
      pnorm(x_tcell,mean=m0_tcell,sd=s0_tcell), col=2, lwd=1)
```

```
AUC.th=pnorm((m1-m0)/sqrt(s0^2+s1^2))
text(.6,.4,paste("Empirical AUC = ",round(100*AUC_tcell,2),
                 "% \n Binormal AUC = ",round(100*AUC.th_tcell,2), "%", sep=""),
     cex=.75,font=4)
legend("bottomright",c("Empirical ROC curve","Binormal ROC curve"),
       lty=1,lwd=c(2,1),col=c(1,2),cex=.7,bg="gray96")
```



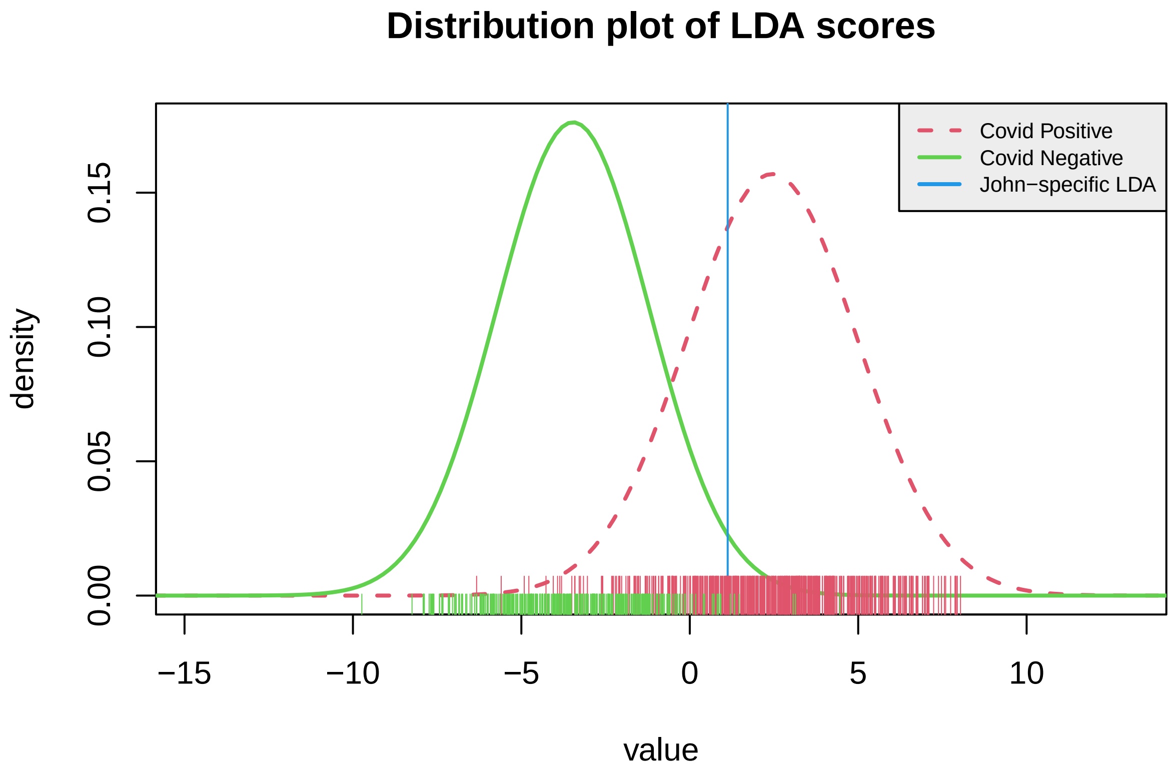
Does addition of individual-specific variables improve COVID detection?

Answer: Addition of individual-specific variables ($TcellValue, Age, Gender, BMI, Week$) does **improve** the COVID detection ability compared to single-variable predictor ($TcellValue$) for separating the two classes. The AUC score improved from 91.37% to 95.34%.

5. The T-cell count of 56 years old John (BMI=19) measured 7 weeks after he felt sick was 45%. Estimate the probability that he got COVID-19 using LDA (use theoretical probability for the estimate). Before computing this probability display LDA scores along with the normal pdfs as in #2 and display John-specific LDA score as a vertical bar. Report this probability on the graph.

```
# Store John information as vector (TcellValue, Age, Gender, BMI, Week)
john_stats <- c(45, 56, 1, 19, 7)

# LDA misclassification rule based on week 5 lecture notes page 1, Theorem 2
mu_avg = (mu0+mu1)/2
score = t(john_stats-mu_avg)%*%a
```

```r
## distribution graph of LDA scores
muY=mean(classR_1);sdY=sd(classR_1)
muX=mean(classR_0);sdX=sd(classR_0)
Lr=c(range(classR_0)[1]-5,range(classR_1)[2]+5)
x=seq(from=Lr[1]-5,to=Lr[2]+5,length=200)
dY=dnorm(x,mean=muY,sd=sdY);dX=dnorm(x,mean=muX,sd=sdX)

matplot(x,cbind(dY,dX),xlim=Lr,lty=c(2,1),type="l",col=c(2,3),lwd=2,
        xlab="value",
        ylab="density",
        main="Distribution plot of LDA scores")
abline(v=score, col=4)
rug(classR_1,lty=1,ticksize=.075, col=2)
rug(classR_0,lty=1,ticksize=.04, col=3)
legend("topright",c("Covid Positive","Covid Negative", "John-specific LDA"),lty=c(2,1,1),lwd=2,
       bg="gray93",cex=0.7, col = c(2,3,4))
```



Answer:

The LDA score for John is 1.125, which tells us that based on the LDA misclassification rule, if the score is more than 0 then it belongs to COVID negative group.Therefore, John is predicted to be COVID negative.