qbs124_hw6_gibran

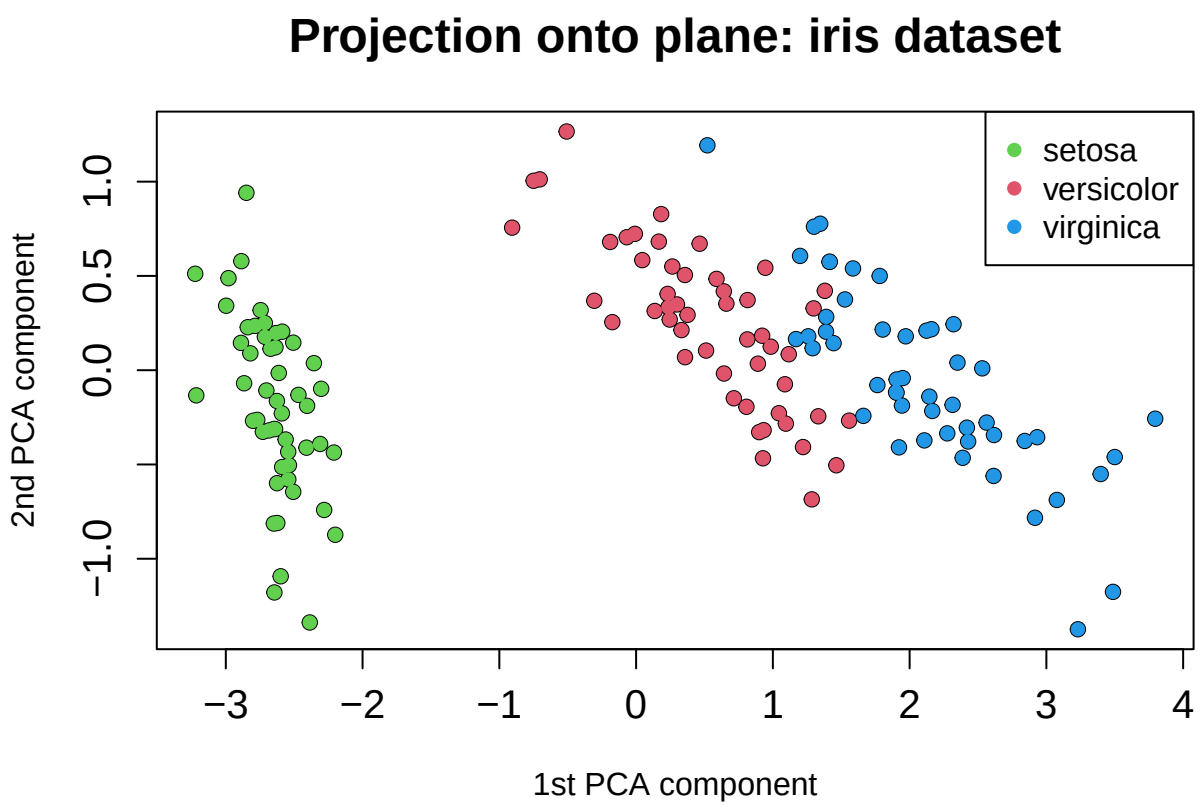
Gibran Erlangga

5/6/2022

Question 1

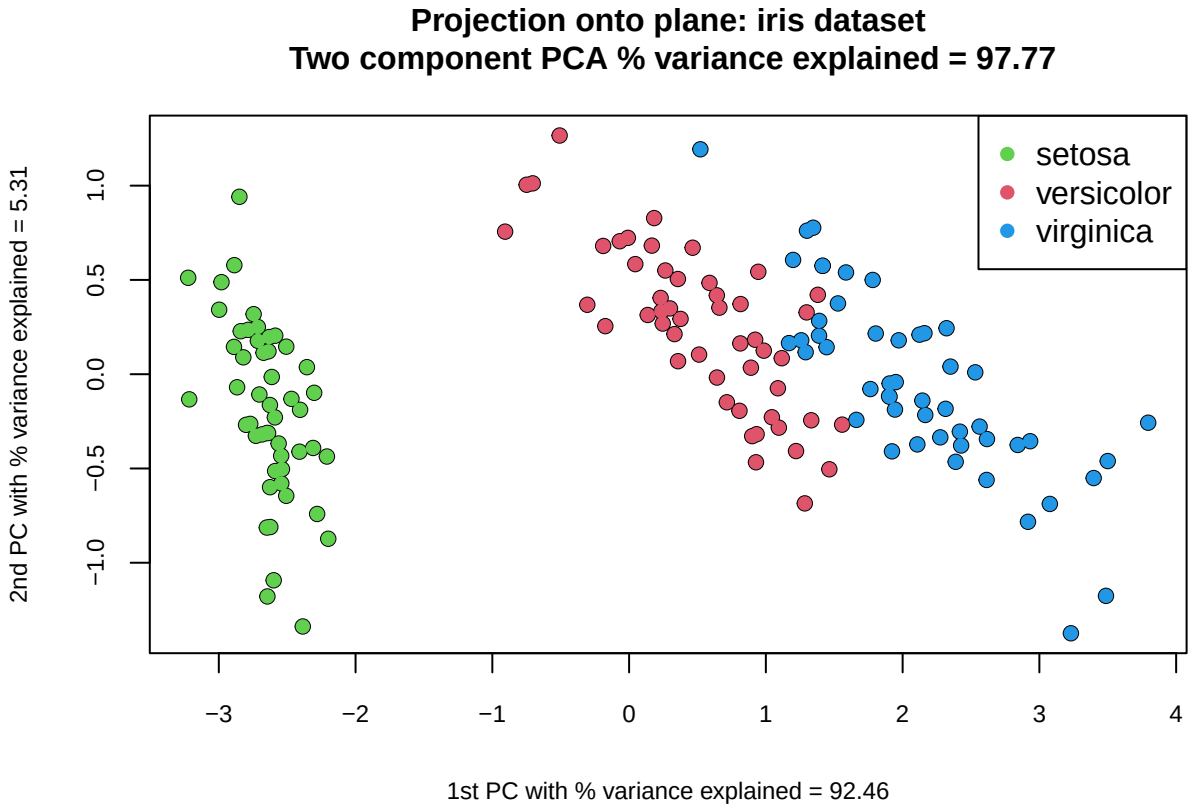
(20 points). (a) Apply PCA to project the iris data onto the plane, display and color each point.

```
#import data
species <- subset(iris, select = c(Species))

# get index for each type
idx_setosa <- which(species == "setosa")
idx_versicolor <- which(species == "versicolor")
idx_virginica <- which(species == "virginica")

X <- subset(iris, select = -c(Species))
X <- data.matrix(X)
n=nrow(X);m=ncol(X)

#========================== Matrix centering and computation of Z'Z
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=1.5,cex.axis=1.25)
a=eigen(tZZ,symmetric=T)$vectors[,1:2]
Z=X-rep(1,n)%*%t(colMeans(X))
proj=Z%*%a

plot(proj,xlab="1st PCA component",ylab="2nd PCA component")
title("Projection onto plane: iris dataset")
points(proj[idx_setosa,],col=3,pch=16)
points(proj[idx_versicolor,],col=2,pch=16)
points(proj[idx_virginica,],col=4,pch=16)
legend("topright", legend = c("setosa", "versicolor", "virginica"), pch = c(16, 16, 16), col = c(3,2,4))
```



(b) Compute and display the proportion of variance explained by individual components and two components together on the top of the graph.

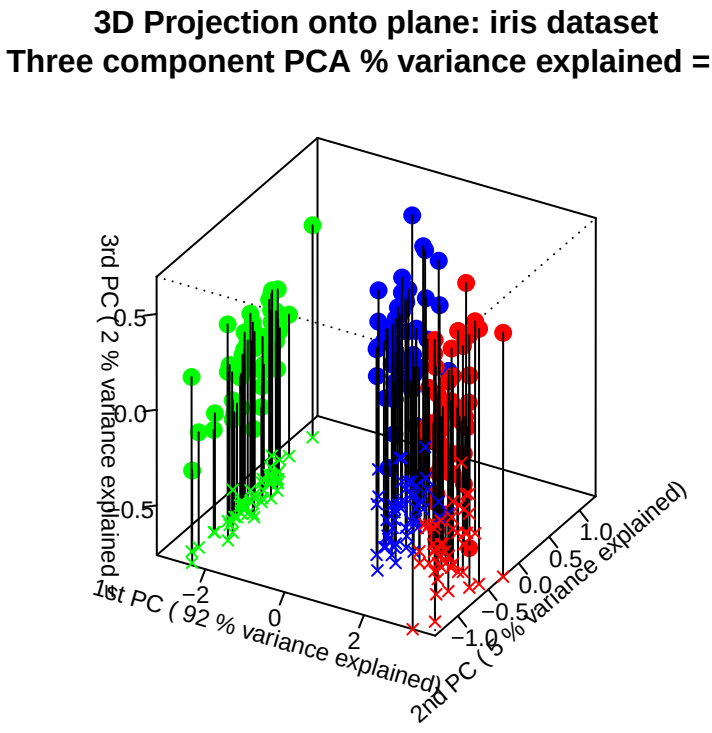
```
par(mfrow=c(1,1),mar=c(4.5,4.5,4,1),cex.lab=.75,cex.main=1,cex.axis=.75)
# a=eigen(tZZ,symmetric=T)$vectors[,1:2]
# Z=X-rep(1,n)%*%t(colMeans(X))
# proj=Z%*%a
L12=eigen(tZZ,symmetric=T)$values
var1.expl=L12[1]/sum(L12)*100
var2.expl=L12[2]/sum(L12)*100
var12.expl=(L12[1]+L12[2])/sum(L12)*100
txt1=paste("1st 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(proj,xlab=txt1,ylab=txt2)
title(paste("Projection onto plane: iris dataset\n",txt12))
points(proj[idx_setosa,],col=3,pch=16)
points(proj[idx_versicolor,],col=2,pch=16)
points(proj[idx_virginica,],col=4,pch=16)
legend("topright", legend = c("setosa", "versicolor", "virginica"), pch = c(16, 16, 16), col = c(3,2,4))
```



(c) Repeat the same tasks for the 3D projection (use theta=30 and phi=30).

```
nm=names(iris)
flnames=as.character(iris[,5])
uflnames=unique(flnames);
nfl=length(uflnames)
cl=rep(0,nrow(iris))
for(i in 1:nfl) cl[flnames==uflnames[i]]=i
allcl=c("green","blue","red")

a=eigen(tZZ,symmetric=T)$vectors[,1:3]
Z=X-rep(1,n)%*%t(colMeans(X))
proj=Z%*%a

L123=eigen(tZZ,symmetric=T)$values
var1.expl=L123[1]/sum(L123)*100
var2.expl=L123[2]/sum(L123)*100
var3.expl=L123[3]/sum(L123)*100
var123.expl=(L123[1]+L123[2]+L123[3])/sum(L123)*100
txt1=paste("1st PC (", round(var1.expl), "% variance explained)")
txt2=paste("2nd PC (", round(var2.expl), "% variance explained)")
txt3=paste("3rd PC (", round(var3.expl), "% variance explained =")
txt123=paste("Three component PCA % variance explained =",round(var123.expl))

nmo = c(txt1, txt2,txt3)
```

```
X = proj
par(mfrow=c(1,1),mar=c(4.5,4.5,4,1),cex.lab=.75,cex.main=1,cex.axis=.75)
op=persp(x=range(X[,1]),y=range(X[,2]),z=matrix(ncol=2,nrow=2),
         zlim=range(X[,3]),xlab=nmo[1],ylab=nmo[2],zlab=nmo[3],
         theta=30,phi=30,r=1000,
         main=paste("3D Projection onto plane: iris dataset\n",txt123),
         ticktype="detailed")
  p3=trans3d(x=X[,1], y=X[,2], z=X[,3], pmat=op)
  points(p3,pch=16,cex=1.25,col=allcl[cl])
  n=length(X[,1])
  p2=trans3d(x=X[,1], y=X[,2], z=rep(min(X[,3]),n), pmat=op)
  segments(p3$x,p3$y,p2$x,p2$y)
  points(p2,pch=4,cex=.75,col=allcl[cl])
```

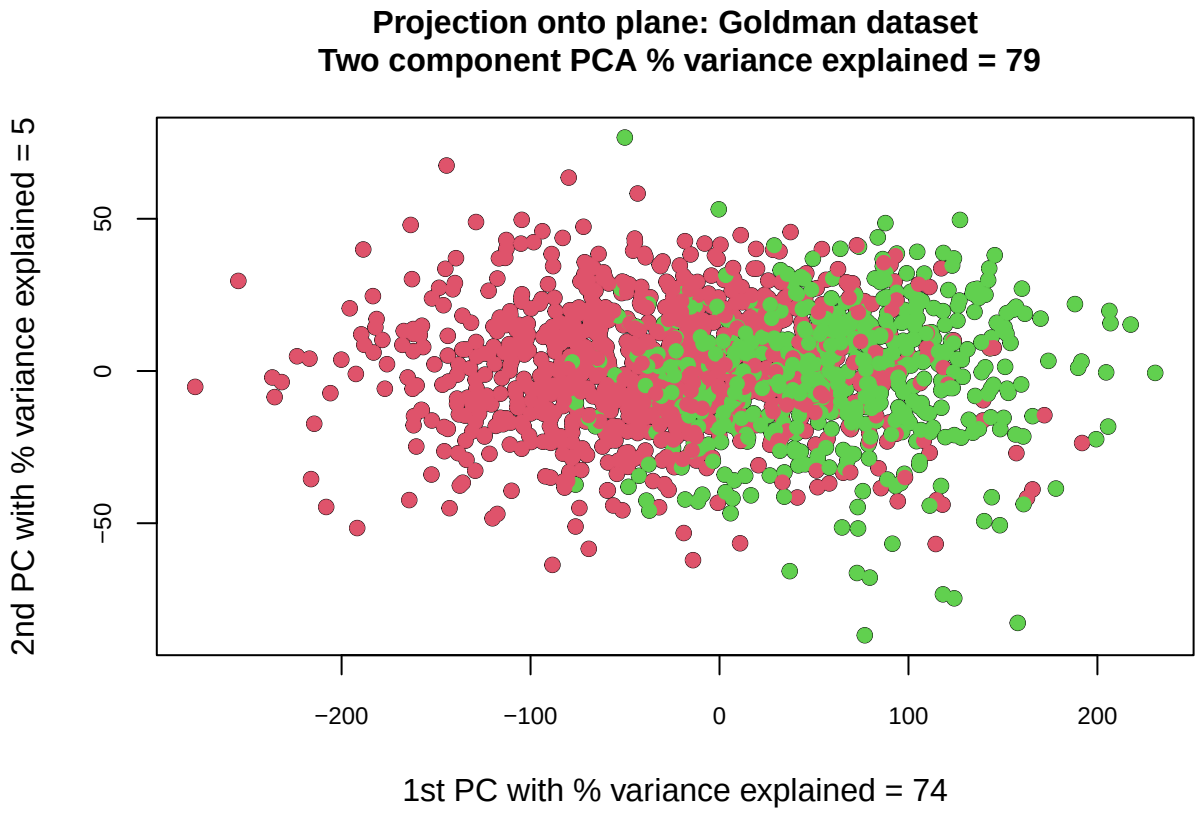


Question 2

(20 points). (a) Project Goldman.imputed.csv data onto plane. (b) Use red to color male and green to color female.

```
X=read.csv("Goldman.imputed.csv")
sex=as.numeric(as.vector(X[,1]))
n_sex = length(sex)
X <- data.matrix(subset(X, select = -female))
n=nrow(X); m=ncol(X)
```

```
# create sex color
sex_clr = rep(0, n_sex)
for (i in 1:n_sex) {
  if (sex[i] == 1) {
    sex_clr[i] = 3
  }
  else {
    sex_clr[i] = 2
  }
}

Z=X
for(j in 1:m)
{
    xj=X[,j]
    Z[,j]=xj-mean(xj)
}
tZZ=t(Z)%*%Z

a=eigen(tZZ,symmetric=T)$vectors[,1:2]
Z=X-rep(1,n)%*%t(colMeans(X))
proj=Z%*%a
L12=eigen(tZZ,symmetric=T)$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))
txt2=paste("2nd PC with % variance explained =",round(var2.expl))
txt12=paste("Two component PCA % variance explained =",round(var12.expl))

par(mfrow=c(1,1),mar=c(4.5,4.5,4,1),cex.lab=1,cex.main=1,cex.axis=.75)
plot(proj,xlab=txt1,ylab=txt2)
points(proj,col=sex_clr,pch=19, lwd=.5)
title(paste("Projection onto plane: Goldman dataset\n",txt12))
```



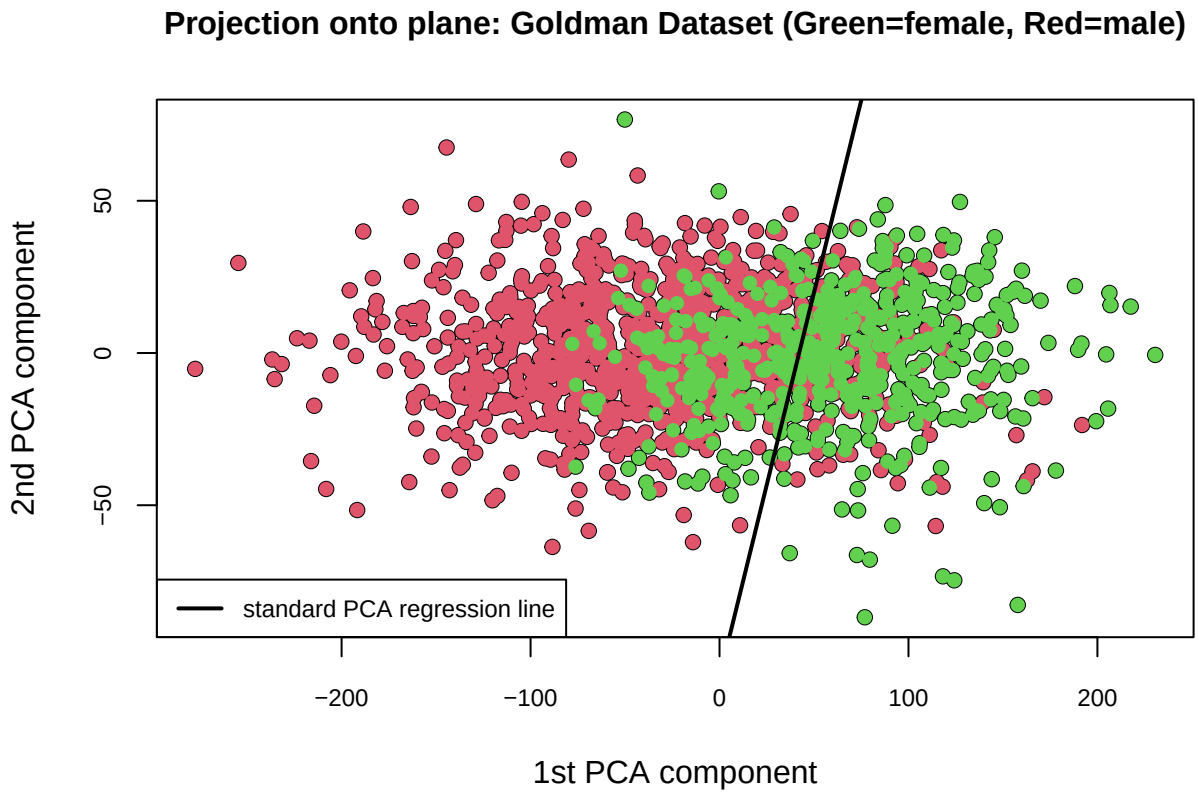
(c) Compute and display the non-optimized/standard PCA logistic regression separation line.

```
#setup
X=read.csv("Goldman.imputed.csv")
sex=as.numeric(as.vector(X[,1]))
n_sex = length(sex)
col_names = names(subset(X, select = -female))
X <- data.matrix(subset(X, select = -female))

# compute eigenvectors
W=var(X,use="pairwise.complete.obs")
eigenW=eigen(W,sym=T)
p.max=eigenW$vectors[,1]
lambda.max=eigenW$values[1]

# compute principal components
a=eigenW$vectors[,1:2]
Z=X-rep(1,nrow(X))%*%t(colMeans(X))
proj=Z%*%a
ni = length(proj[,1])

# plot PCA projection
par(mfrow=c(1,1),mar=c(4.5,4.5,4,1),cex.lab=1,cex.main=1,cex.axis=.75)
plot(proj,xlab="1st PCA component",ylab="2nd PCA component")
title("Projection onto plane: Goldman Dataset (Green=female, Red=male)")
```

```
points(proj[sex==0,],col=2,pch=16)
points(proj[sex==1,],col=3,pch=16)

# plot standard PCA line
y=sex
oLOG.R=glm(y~proj,family=binomial)
a=coef(oLOG.R)
x=seq(from=-300,to=300,length=1000)
proj.y=-(a[1]+a[2]*x)/a[3]
lines(x,proj.y,lwd=2)

# sort c and loop over all values, get c with minimum total error for optimized PCA
lp=oLOG.R$linear.predictors
threshold=seq(min(lp),max(lp),length.out=ni)
total_error=rep(0,ni)
n0 = sum(sex == 0); n1 = sum(sex == 1)
for (i in 1:ni){
 total_error[i]=sum(lp<threshold[i] & sex==1)/n1+sum(lp>threshold[i] & sex==0)/n0
 }
best.threshold=threshold[which.min(total_error)]
print(paste("the best threshold c is",round(best.threshold,2)))
```

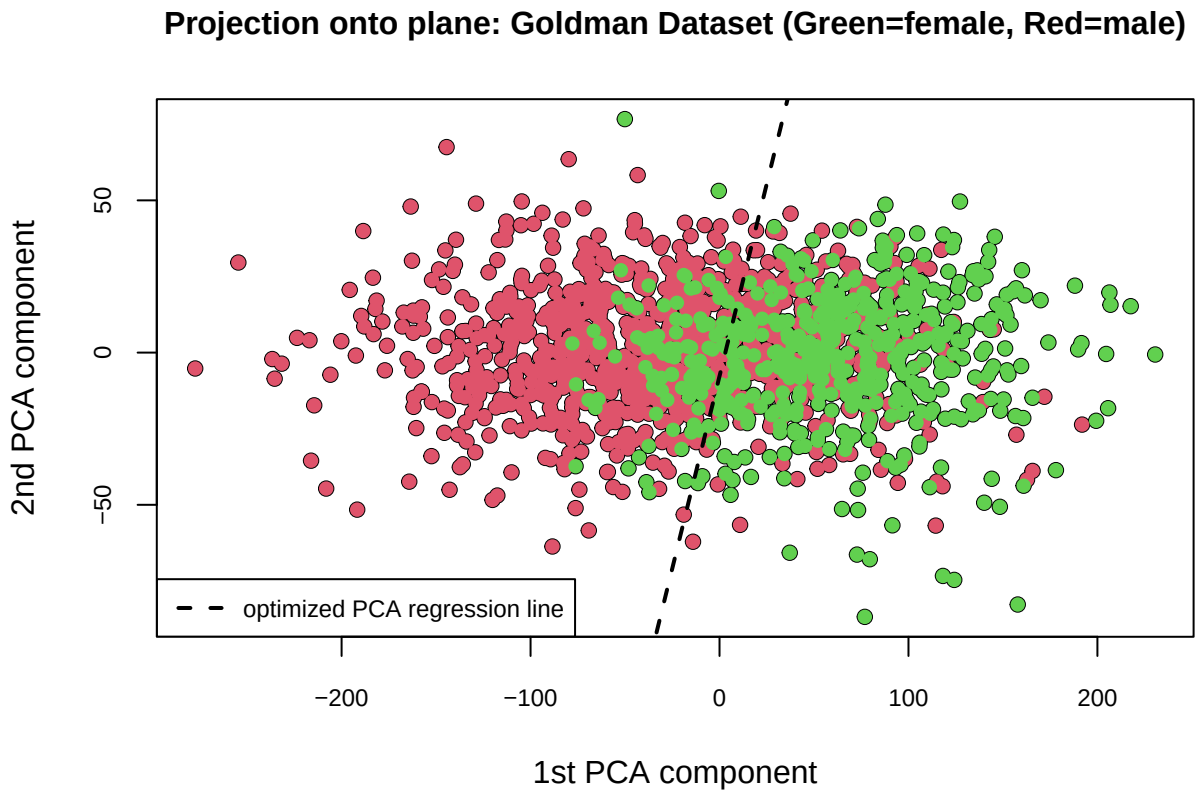
```
## [1] "the best threshold c is -0.86"
```

```
legend("bottomleft", legend = c("standard PCA regression line"), lty = c(1), col = c(1), lwd=c(2), cex=
```



(d) Compute and display the optimized PCA logistic regression line with the threshold that minimizes the total misclassification error.

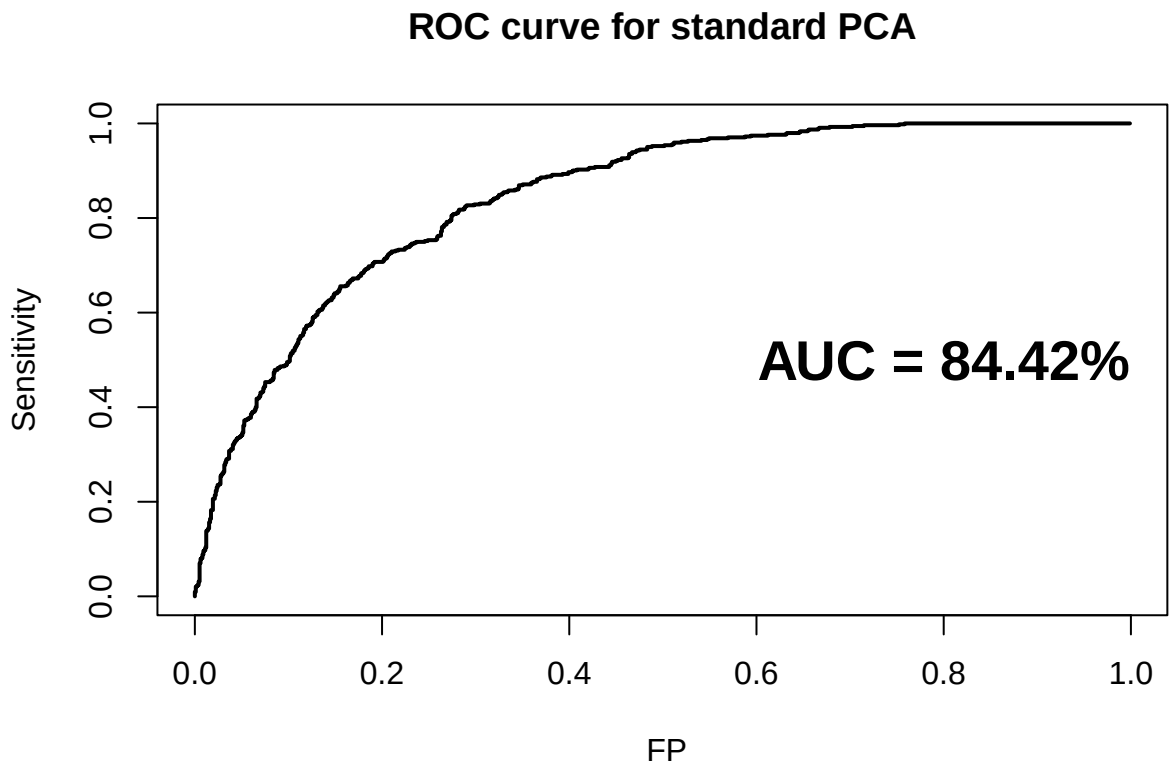
```
par(mfrow=c(1,1),mar=c(4.5,4.5,4,1),cex.lab=1,cex.main=1,cex.axis=.75)
plot(proj,xlab="1st PCA component",ylab="2nd PCA component")
title("Projection onto plane: Goldman Dataset (Green=female, Red=male)")
points(proj[sex==0,],col=2,pch=16)
points(proj[sex==1,],col=3,pch=16)

x=seq(from=-300,to=300,length=1000)
proj.y=(best.threshold-(a[1]+a[2]*x))/a[3]
lines(x,proj.y,lwd=2, lty=2,col=1)
legend("bottomleft", legend = c("optimized PCA regression line"), lty = c(2), col = c(1), lwd=c(2), cex
```



(e) Display the two ROC curves along with the respective AUCs.

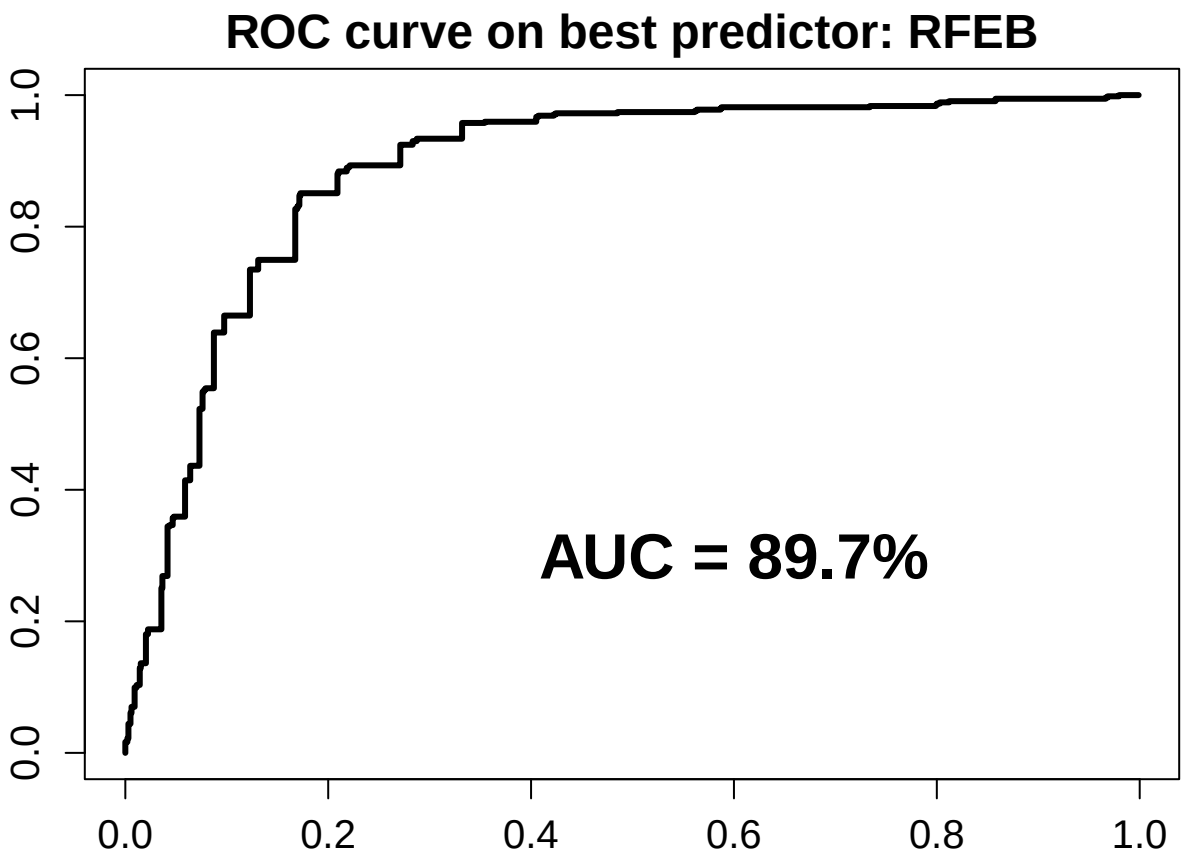
```
# ROC curve for standard PCA
logit = glm(sex~proj, family=binomial)
lin_pred = logit$linear.predictors
sod = sort(lin_pred)
ni = length(proj[,1])
AUC = 0
sens = fp = rep(0, ni)
toter = rep(NA, ni)
n1 = sum(sex == 1)
```

```r
n0 = sum(sex == 0)
for (i in 1:ni) {
  sens[i]=sum(lin_pred > sod[i] & sex==1)/n1
  fp[i]=sum(lin_pred > sod[i] & sex==0)/n0
  if(i>1) AUC=AUC+sens[i]*(fp[i-1]-fp[i])
}

optimal_threshold = sod[toter == min(toter)]
optimal_sens = sens[toter == min(toter)]
optimal_fp = fp[toter == min(toter)]

plot(fp, sens, type='s', lwd=2, xlab= ' FP', ylab='Sensitivity', main='ROC curve for standard PCA')
lines(x=c(-1, optimal_fp, optimal_fp), y = c(optimal_sens, optimal_sens, -1), col='red')
text(.8,.5,paste("AUC = ",round(AUC*100,2),"%",sep=""),cex=1.75,font=2)
```




```r
# ROC curve using 1 variable (best predictor)
nc=ncol(X)
AUC=rep(0,nc)
for(ivar in 1:nc) {
  x=X[,ivar]
  y=sex
  ni=length(x)
  n0=sum(1-y);n1=sum(y)
  sod=sort(x)
  fp0=0
```

```
  for(i in 1:ni) {
    sens=sum(x<sod[i]&y==1)/n1
    fp=sum(x<sod[i]&y==0)/n0
    if(i>1) AUC[ivar]=AUC[ivar]+sens*(fp-fp0)
    fp0=fp
  }
}

i=1:nc
ibest=i[AUC==max(AUC)]
v1.best=X[,ibest];nm.best=col_names[ibest]
x=v1.best
n0=sum(1-y);n1=sum(y)
ni=length(x)
o=glm(y~x,family=binomial)
sod=sort(x)
AUC.best=fp0=0
sens=fp=rep(0,ni)
for(i in 1:ni)
{
sens[i]=sum(x<sod[i]&y==1)/n1
fp[i]=sum(x<sod[i]&y==0)/n0
if(i>1) AUC.best=AUC.best+sens[i]*(fp[i]-fp0)
fp0=fp[i]
}
par(mfrow=c(1,1),mar=c(2,2,2,2),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("ROC curve on best predictor:", nm.best))
text(.6,.3,paste("AUC = ",round(AUC.best*100,1),"%",sep=""),cex=2,font=2)
```



(f) Explain why the PCA logistic regression yields a worse result than just using one RFEB variable.

Answer: PCA carries all information from the predictors available in the Goldman dataset, including the features that do not bring useful information to predict sex. This irrelevant information can be considered as a "noise" to the model, as opposed to only use the best predictor (RFEB). Therefore, RFEB has better AUC compared to the PCA because it has less noise.