#Solutions HW7
require("ggplot2")
require("MASS")
require("car")
require('mvtnorm')

#Problem 1:
#1.14
#The table is store on the course webpage as sclerosis.dat.
col.names = c("x1", "x2", "x3", "x4", "x5", "Group"))

#NOTE: Reading in the table gives 6 columns,
#however the last column does not match the data set.
#This set has no ID. Instead the last column identifies
#sclerosis/nonsclerosis. Always check your data structures!

summary(d)
summary(as.factor(d$Group))
head(d)

#a) Scatterplot matrix:
#Plotting a scatter plot matrix can be done in using "pairs"
pairs(d)

#Since we only want x2 and x4 in the diagram:
head(d[,c("x2", "x4")])
pairs(d[,c("x2", "x4")])

#Normal scatter plot:
plot(d$x2,d$x4)

#b) Compute Xbar, Sn, R arrays for both groups:
d.nonscler<-d[which(d$Group == '0'),]
d.scler<-d[which(d$Group == 1),]
head(d.nonscler)
head(d.scler)

rbind(mean(d.nonscler[,1:5]),
mean(d.scler[,1:5]))

cov(d.nonscler[,1:5])
cov(d.nonscler[,1:5])

cor(d.scler[,1:5])
cor(d.scler[,1:5])

#Problem 1:
#11.23 a)
#qplot() is a wrapper for ggplot that require comparitively little code:
#qplot is smart: all you need is "sample=datavector"
#requires ggplot2 package: http://had.co.nz/ggplot2/

qplot()
# QQ-plots for sclerosis set
qplot(sample=d.scler$x1);
qplot(sample=d.scler$x2);
qplot(sample=d.scler$x3);
qplot(sample=d.scler$x4);
qplot(sample=d.scler$x5);

# QQ-plots for non-sclerosis set
qplot(sample=d.nonscler$x1)
qplot(sample=d.nonscler$x2)
qplot(sample=d.nonscler$x3)
qplot(sample=d.nonscler$x4)
qplot(sample=d.nonscler$x5)

#variables x3 and x5 appear to violate the normality assumption.
#log transforms does not appear to help much:
qplot(sample=log(d.scler$x3))
qplot(sample=log(d.nonscler$x3))
qplot(sample=log(d.scler$x5))
qplot(sample=log(d.nonscler$x5))

d.scler
#Power transformation seem to do better:
#requires library("car")
Box.Cox<-powerTransform(d.scler[,1:5],family="bctrans")
#Some values are zero! Need a power transformation that works
#when data is not strictly positive!

#slight variation Yeo-Johnson does!
powers<-powerTransform(d.scler[,1:5],family="yjPower")
trans.scler<-yjPower(d.scler[,1:5],coef(powers,round=TRUE))

qplot(sample=trans.scler$x1)
qplot(sample=trans.scler$x2)
qplot(sample=trans.scler$x3)
qplot(sample=trans.scler$x4)
qplot(sample=trans.scler$x5)

powers<-powerTransform(d.nonscler[,1:5],family="yjPower")
trans.nonscler<-yjPower(d.nonscler[,1:5],coef(powers,round=TRUE))

qplot(sample=trans.nonscler$x1)
qplot(sample=trans.nonscler$x2)
qplot(sample=trans.nonscler$x3)
qplot(sample=trans.nonscler$x4)
qplot(sample=trans.nonscler$x5)

#11.23 b)
#Ignoring the potential differences between the covariance matrices
#(a difference illustrated using a BoxMTest from Dr. Maitra's page,
we can use Fishers linear discriminant function, which in R is

```r
# see lda overall
lda(Group ~ ., data=d)

# get coefficients of lin. disc. func.
lda.coefs<-lda(Group ~ ., data=d)[[4]][1:5]

# From the coefficients we see that x5, x3 and possibly x4 are
# the most important variables.

# A classification rule:
# Following the discussion on page 582 of the book,
# specifically the choice to assign an observation
# X0 to either the sclerotic or non-sclerotic group
# as described in part (c) of the box:

classify11.7<-function(X0,d.1,d.0){
p1<-ncol(d.1); p0<-ncol(d.0)
n1<-nrow(d.1); n0<-nrow(d.0);
Sp.ish<-(n1-1)*cov(d.1)+(n0-1)*cov(d.0) # proportional to Spool
res<-(t(mean(d.1)-mean(d.0))%*%ginv(Sp.ish)%*%X0 -
     .5*(mean(d.1)-mean(d.0))%*%ginv(Sp.ish)%*%(mean(d.1)+mean(d.0)))
return(paste("Belongs to", c(paste("d",as.numeric(res>0),sep="")),sep=" "))
}

classify11.7(mean(d.scler),d.1=d.nonscler,d.0=d.scler)
classify11.7(mean(d.nonscler),d.1=d.nonscler,d.0=d.scler)
classify11.7(mean(d.scler)-mean(d.nonscler),d.1=d.nonscler,d.0=d.scler)

# 11.23 c)
# This is easy enough using lda()

# Predicting new observations:
t1<-table(d$Group,predict(lda(Group ~ ., data=d,prior=c(1/2,1/2)))$class)
APER<-(t1[1,2]+t1[2,1])/sum(t1)
APER

# We get the expected actual error rate using cross validation:
t2<- table(d$Group,lda(Group ~ ., data=d,prior=c(1/2,1/2),CV=TRUE)$class)
EAER<-(t2[1,2]+t2[2,1])/sum(t2)
EAER

# A difference of about 23% of the expected value.
(APER-EAER)/EAER
### Problem 2

#### 11.27 a)
```r
head(iris)
qplot(Sepal.Width,Petal.Width,data=iris)
qplot(Sepal.Width,Petal.Width,data=iris,color=Species,shape=Species)
```
Each group looks roughly elliptical in this arrangement, so bivariate normal is not unreasonable.

#### 11.27 b)
We can use a multivariate pillai trace to determine effects of species on the mean, as we did in previous homeworks (5?)
```r
fit.species<-lm(cbind(iris$Sepal.Width,iris$Petal.Width)~iris$Species)
Manova(fit.species)
```
The p-value from the Pillai test is so small, that we reject the null hypothesis.
Thus, there is statistical evidence that the means are not the same for each species of plant.

We can test the assumption of identical covariance matrices via Dr. Maitra:
```r
source('http://www.public.iastate.edu/~maitra/stat501/Rcode/BoxMTest.R')
BoxMTest(cbind(iris$Sepal.Width,iris$Petal.Width),iris$Species)
```
The small p-value indicates that we must sadly reject this assumption too.

#### 11.27 c)
```r
new.x<-c(3.5,1.75)
```
Following 11-51 and 11-52, we construct:
```r
d.1<-iris[1:50,]
d.2<-iris[51:100,]
d.3<-iris[101:150,]

test_x_in_group<-function(x,d,p=1/3){
  return(-.5*log(det(var(d)))-
         .5*t(x-mean(d))%*%ginv(var(d))%*%(x-mean(d))+log(p))
}
test_x_in_group(new.x,d.1[,c(2,4)])
test_x_in_group(new.x,d.2[,c(2,4)])
test_x_in_group(new.x,d.3[,c(2,4)])
```
The maximum score goes to veriscolor. So new.x is a veriscolor.

#### 11.27 d)
```r
Spooled<-49*(cov(d.2[,c(2,4)])+cov(d.1[,c(2,4)])+ cov(d.3[,c(2,4)])) / 147

```
Modifying the function above, we see that:
testXin_pool<-function(x,d,p=1/3,sig=Spooled){
  return(
    t(mean(d))%*%ginv(sig)%*%x -.5*t(mean(d))%*%ginv(sig)%*%mean(d)+log(p))
}
testXin_pool(new.x,d.1[,c(2,4)])
testXin_pool(new.x,d.2[,c(2,4)])
testXin_pool(new.x,d.3[,c(2,4)])

#Once again, we would classify this flower as a variscolor.

#11.27 e)  
#Consider:  
rule1156<-function(x,d1,d2,sig=Spooled){
  return( t(mean(d1)-mean(d2))%*%ginv(Spooled)%*%x -
    .5*t(mean(d1)-mean(d2))%*%ginv(Spooled)%*%(mean(d1)+mean(d2)))
}
rule1156(new.x,d.1[,c(2,4)],d.2[,c(2,4)])
rule1156(new.x,d.1[,c(2,4)],d.3[,c(2,4)])
rule1156(new.x,d.2[,c(2,4)],d.3[,c(2,4)])

#results: choose d2 over d1,
#          choose d3 over d1,
#          choose d2 over d3

#We have once again determined that the flower is a variscolor.

#LDA partitions:  
library(klaR)
x<-c("Xnew")
partimat(Species~Petal.Width+Sepal.Width,method='lda',data=data.new)
#Verfinica on top

#11.27 f)  
#Similar to a previous problem:  

#Predicting new observations:  

iris.lda<-lda(iris$Species ~ iris[,2]+iris[,4],prior=c(1/3,1/3,1/3))
APER<-mean(iris$Species != predict(iris.lda)$class)
APER

iris.cv<-lda(iris$Species ~ iris[,2]+iris[,4],prior=c(1/3,1/3,1/3),CV=TRUE)
EAER<-mean(iris$Species != iris.cv$class)
EAER

#A difference of about 23% of the expected value.
(APER-EAER)/EAER
#over estimated by about 17%
#11.28

#11.28 a)

```r
head(iris)
iris$Y1 <- iris$Sepal.Length/iris$Sepal.Width
iris$Y2 <- iris$Petal.Length/iris$Petal.Width
qplot(log(Y1), log(Y2), data = iris, color = Species)
```

#There could be a bivariate normal distribution on this data.

#11.28 b) AND c)

#We can do all of this in a few quick steps:

```r
lda.column <- lda(iris$Species ~ iris[, 1], prior = c(1, 1, 1)/3)
table(iris$Species, predict(lda.column)$class)
```

```r
lda.column <- lda(iris$Species ~ log(iris[, 2]), prior = c(1, 1, 1)/3)
table(iris$Species, predict(lda.column)$class)
```

#Repeating these will give us the information required.

#To summarize:

```
#   AP   EA
# logy1 0.327 0.327
# logy2 0.327 0.327
# both  0.173 0.180
```

#11.28 d) It would seem to be true if we consider both sets of shape at a time.

#Problem 3

#11.24 a)

#a) Scatterplot matrix:

```r
cash.money <- read.table("http://www.public.iastate.edu/~maitra/stat501/datasets/finance.dat", col.names = c("x1", "x2", "x3", "x4", "pop"))
cash.money$pop <- as.factor(cash.money$pop)
head(cash.money)
pairs(cash.money)
```

```
#most of the data is ordinal - x1 vs x2 seems to transformable into bivariate normal, as does, though more diffuse, x3 vs x4, x3 vs x2, x3 vs x1 and x4 vs x1
```

#11.24 b)

```r
bankrupt <- cash.money[1:21, 1:2]
banking <- cash.money[1:21, 2]
means <- rbind(mean(bankrupt), mean(banking))
s.ing <- var(banking);
s.rupt <- var(bankrupt)
```
#11.24 c)  

We can use qda to accomplish this; however, from scratch Rcode is not very different than previous tests.


```r
b.qda <- qda(cash.money$pop ~ cash.money$x1 + cash.money$x2, data = cash.money, prior = c(.5, .5), CV = F)
b.qda
dataframe(k <- -1/2*log(det(s.b)/det(s.n)) + 1/2*(t(means[1,]) %*% solve(s.b) %*% means[1,]
          - t(means[2,]) %*% solve(s.n) %*% means[2,]))
rule <- -1/2*diag((x) %*% (solve(s.b)-solve(s.n)) %*% t(x)) +
as.vector((t(means[1,]) %*% solve(s.b)-t(means[2,]) %*% solve(s.n)) %*% t(x)) - rep(k, 46, nrow=46)
```

#11.24 d)  

```r
b.qda <- qda(cash.money$pop ~ cash.money$x1 + cash.money$x2, data = cash.money, prior = c(.5, .5), CV = F)
table(cash.money$pop, predict(b.qda)$class)
APER <- mean(cash.money$pop != predict(b.qda)$class)
b.cvqda <- qda(cash.money$pop ~ cash.money$x1 + cash.money$x2, data = cash.money, prior = c(.5, .5), CV = T)
table(cash.money$pop, b.cvqda$class)
EAER <- mean(cash.money$pop != b.cvqda$class)
c(APER, EAER)
```

#11.24e)

```r
b.qda <- qda(cash.money$pop ~ cash.money$x1 + cash.money$x2, data = cash.money, prior = c(.95, .05), CV = F)
table(cash.money$pop, predict(b.qda)$class)
APER <- mean(cash.money$pop != predict(b.qda)$class)
b.cvqda <- qda(cash.money$pop ~ cash.money$x1 + cash.money$x2, data = cash.money, prior = c(.95, .05), CV = T)
table(cash.money$pop, b.cvqda$class)
EAER <- mean(cash.money$pop != b.cvqda$class)
c(APER, EAER)
```

#APER and EAER are larger than in #11.24d

#11.24f)

```r
Spool <- 20*s.rupt + 24*s.ing
Spool <- Spool/(44)
scores <- t(means[1,] - means[2,]) %*% solve(Spool) %*% t(cash.money[,1:2])
m <- -1/2*(t(means[1,] - means[2,]) %*% solve(Spool) %*% (means[1,] + means[2,]))
m <- rep(m, nrow(cash.money))
scores[scores > m] <- 0
scores[scores != 0] <- 1
scores <- as.vector(scores)
mean(cash.money[, 5] != scores)
bartlett.test(cash.money[, 1:2], cash.money$POP)
```
head(cash.money)
# g, h are repetition of the first part of this problem.
# One useful way to do this is to write a function that handles these calculations in one sweep,
# such as the one below (slightly modified from, and with thanks to, a student):
Sweep1124<-function(data1, data2, class=pop, largedataset=bankruptcy, classvar=pop, basicprior=c(.5,.5)){
  mean1<-mean(data1);  S1<-var(data1)
  mean2<-mean(data1);  S2<-var(data1)
  r1.qda <-qda(largedataset$classvar ~ ., data = rbind(data1,data2), prior = basicprior, CV = F)
  r1.cvqda<-qda(largedataset$classvar ~ ., data = rbind(data1,data2), prior = basicprior, CV = T)
  qda.resultr1 <- mean(largedataset$classvar!= predict(r1.qda)$class)
  cvda.resultr1 <- mean(largedataset$classvar!= r1.cvqda$class)
  r2.qda  <-qda(largedataset$classvar ~ ., data = rbind(data1,data2), prior = basicprior, CV = F)
  r2.cvqda<-qda(largedataset$classvar ~ ., data = rbind(data1,data2), prior = basicprior, CV = T)
  qda.resultr2 <- mean(largedataset$classvar!= predict(r2.qda)$class)
  cvda.resultr2 <- mean(largedataset$classvar!= r2.cvqda$class)
  r3.lda <- lda(largedataset$classvar ~ ., data = rbind(data1,data2), prior = basicprior, CV = F)
  Fisher.coef <- coef(b.lda)
  APER.f <- mean(largedataset$classvar!= predict(b.lda)$class)
  return(list(xbar1 = mean1,
              S1 = S1,
              xbar2 = mean2,
              S2 = S2,
              APER.d = r1.qda, EAER.d = r1.cvqda,
              APER.e = r2.qda, EAER.e = r2.cvqda,
              Fisher.coef = Fisher.coef, APER.f = APER))
}

#11.25

#11.25 a)
                     header=FALSE, col.names = c(paste("X","",1:4, sep=""), "is.finance"))
finance$is.finance<- as.factor(finance$is.finance)
johnson<-finance[,c(1,3)]
means<-apply(johnson, 2, function(u) tapply(u, finance$is.finance, mean))
s.bj<-var(johnson[finance$is.finance==0,])
s.nj<-var(johnson[finance$is.finance==1,])
spool<-(20*s.bj + 24*s.nj)/(44)
fisherj <- t(means[1,] - means[2,])%*%solve(spool)
m <- 1/2*t(means[1,] - means[2,])%*%solve(spool)%*(means[1,] + means[2,])
m <- rep(m,nrow(finance))
scoresj<-t(fisherj%*%t(johnson))
scoresj[scoresj > m]<-0
scoresj[scoresj != 0]<-1
mean(finance$is.finance != scoresj)

plot(johnson, main="Discriminant Line and Data", xlab="x1", ylab="x3", col = as.numeric(finance$is.finance)+3, pch=20)
abline(a0,a1)

#11.25 b)
johnson.16.lda<-lda(finance$is.finance[-16] ~ ., data=johnson[16,], prior=c(1,1)/2)
mean(finance$is.finance[-16] != predict(johnson.16.lda)$class)
johnson.13.lda<-lda(finance$is.finance[-13] ~ ., data=johnson.13, prior=c(1,1)/2)
mean(finance$is.finance[-13] != predict(johnson.13.lda)$class)
c1<- -fisher.16[1]/fisher.16[2]
c0<- m.16/fisher.16[2]
e1<- -fisher.13[1]/fisher.13[2]
e0<- m.13/fisher.13[2]
abline(c0,c1, col="red")
abline(e0,e1, col="purple")
legend(locator(1),legend=c("All","w/o 16","w/o 13"),lty=c(1,1,1),col=c("black", "red", "purple"))

#11.26 a)
bankrupt<- read.table("http://www.public.iastate.edu/~maitra/stat501/datasets/bankruptcy.dat", header=FALSE, col.names = c(paste("X",1:4, sep=""), "is.bankrupt"))
bankrupt$is.bankrupt<- as.factor(bankrupt$is.bankrupt)
#Dot diagrams were one of the TeachingDemos:
library("TeachingDemos")
fit.lm<-lm(as.numeric(is.bankrupt) ~ X3,data=bankrupt)
fit.lm
b0<-fit.lm$fitted[bankrupt$is.bankrupt==0]
b1<-fit.lm$fitted[bankrupt$is.bankrupt==1]
par(mfrow=c(1,2))
dots(int.fit)
dots(slope.fit)
table(bankrupt$is.bankrupt, as.numeric(fit.lm$fitted >= 0.5))
mean(bankrupt$is.bankrupt != as.numeric(fit.lm$fitted >= 0.5))

# Table
#  0  1
# 0 19  2
# 1  4  21

# APER = 6/46

#11.26 b)
# The APER in problem a is kind of large -> may not be best cut off.
# Data sorts better here, maybe one misclassification:

# Table
#  0  1
# 0 19  2
# 1  6  21

# APER = 8/46

#11.26 c)
# Logistic regression leads to much better results:

# Table
#  0  1
# 0 18  3
# 1  1  24

#11.30
oil <- read.table("http://www.public.iastate.edu/~maitra/stat501/datasets/crude-oil.dat",
                   col.names = c("x1", "x2", "x3", "x4", "x5", "zone"))

# In order to discuss the distributions, we need the variance processes first:
# This can be done again using Dr. Maitra's code:
source("http://www.public.iastate.edu/~maitra/stat501/Rcode/BoxMTest.R")

X<-oil[,1:5]
zone<-oil[,6]
BoxMTest(X,zone)

# The p-value is small -> reject H0 -> evidence of a difference.

# We can test normality using the energy test:
library(energy)
mvnorm.etest(X)

# The p-value is small -> reject H0 -> evidence of a difference.

# With both assumptions in doubt, it appears we may not have a normality
# structure we are hoping for.
# variable 3 may be the worst one?
qplot(sample=oil$x3)

# Because of the normality issues, we choose the more robust quadratic discriminant procedure:

oil.qda <- qda(oil$zone ~ ., data = oil[, -6],
               prior = c(1/3, 1/3, 1/3), CV = FALSE)

dq <- NULL
p <- c(1, 1, 1)/3
for (i in 1:nlevels(oil$zone)){
  dq <- cbind(dq, dmvnorm(x = cbind(oil$x1, oil$x2,
                                 oil$x3, oil$x4,
                                 oil$x5),
               mean = oil.qda$means[i,],
               sigma = solve(oil.qda$scaling[i, ,],
               log = TRUE) + log (p[i]))}

This gives us the TPM rule.

# 11.30 b)
E(AER) is best calculated using qda as before:

oil.cv<-qda(oil$zone~.,data=oil[,1:5],prior=c(1,1,1)/3,CV=TRUE)
table(oil$zone, oil.cv$class)

<table>
<thead>
<tr>
<th></th>
<th>SubMul</th>
<th>Upper</th>
<th>Wilhelm</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubMul</td>
<td>8</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Upper</td>
<td>2</td>
<td>36</td>
<td>0</td>
</tr>
<tr>
<td>Wilhelm</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

This leads to E(AER) of: .196

# 11.30 c)
# We can attempt to remove some of the normality problems by looking at correcting the marginals:
Consider the power test, as in problem 1:

# Power transformation seem to do better:
# requires library("car")

Box.Cox<-powerTransform(oil[,1:5],family="bctrans")  # Some values are zero! Need a power transformation that works # when data is not strictly positive!
#slight variation Yeo-Johnson does!
powers<-powerTransform(oil[,1:5],family="yjPower")
trans.oil<-yjPower(oil[,1:5],coef(powers,round=TRUE))
summary(trans.oil)

#Repeating the process for the transformed data is exactly the same.

#11.34:
cereals<-read.table("http://www.public.iastate.edu/~maitra/stat501/datasets/cereals.dat",
  col.names = c("ID", "Maker", paste("x", 1:8, sep = ""), "group"))
head(cereals)

#Function that teaches us how to sort the cereals:
cereals.lda <- lda(as.factor(cereals$Maker) ~., data = cereals[, 1:8],
  prior = c(1/3, 1/3, 1/3),
  CV = FALSE)

n1 <- sum(cereals$Maker=="G")
n2 <- sum(cereals$Maker=="K")
n3 <- sum(cereals$Maker=="Q")
S1 <- var(cereals[cereals$Maker=="G",3:10])
S2 <- var(cereals[cereals$Maker=="K",3:10])
S3 <- var(cereals[cereals$Maker=="Q",3:10])
S.pool <- ((n1-1)*S1 + (n2-1)*S2 + (n3-1)*S3)/(n1 + n2 + n3 -3)
t(solve(S.pool)%*%t(cereals.lda$means[3:10]))

cereals.cvlda <- lda(as.factor(cereals$Maker) ~., data = cereals[, 3:10],
  prior = c(1/3, 1/3, 1/3),
  CV = T)
table(cereals$Maker, cereals.cvlda$class)
mean(cereals$Maker!= cereals.cvlda$class)

E(AER) is pretty high: 40% of Quakers cereals were sorted poorly,
Further there appears to be some 20% of observations for both Kelloggs and
General Mills that are missorted.

#Quaker makes low calorie cereiels,
qplot(x1,x6,data=cereals,shape=Maker,size=x3,color=Maker,
  xlab="calories",ylab="Carbohydrates",
  geom='path',group=Maker
)

qplot(x4,x6,data=cereals,shape=Maker,size=x3,color=Maker,
  xlab="sodium",ylab="Carbohydrates"
)