Tutorial 3

Chapter 11: Discrimination and Classification

Exercise 5: Example 11.11 page 635

The admission officer of a business school has used an "index" of undergraduate grade point average (GPA) and graduate management aptitude test (GMAT) scores to help decide which applicants should be admitted to the school's graduate programs.

$x_1 = GPA, x_2 = GMAT$ values for groups of recent applicants who have been categorized as $\boldsymbol{\pi_1}: Admit$; $\boldsymbol{\pi_2}: Do\ not\ Admit\ and\ \boldsymbol{\pi_3}: Borderline$ The data pictured are listed in Table 11.6.

a) Plot the data .

b) Assuming that the covariance matrices Σ_i are the same for all three bivariate normal populations. Construct sample Squared Distances $d_i^2(x)$ with $P1 = P2 = P3 = \frac{1}{3}$. Using the rule given by (11-54), classify the new observation $x_0 = (3.21, 497)$ into population π1, π2, or π3.

[Or Q: construct the linear discriminate score $\hat{d}_i(x)$, rule given by (11-51)]

c) Assuming that the populations are bivariate normal, construct the quadratic discriminate scores $\hat{d}_i^{\,Q}(x_0)$ with $P1 = P2 = P3 = \frac{1}{3}$. Using the rule given by (11-47), classify the new observation $x_0 = (3.21, 497)$ into population π1, π2, or π3. Compare the results in parts (b) and (c). Which approach do you prefer? Explain
[(11-47) Consider covariance matrix are unequal and unknown].

d) Using the linear discriminant function from part (b), construct the "confusion matrix" by classifying the given observations. Calculate the **APER** .

e) Using the linear discriminant function from part (b), Estimate the actual error rate (cross-validation).

NOTE: [(d, e) Using the linear discriminant scores from Part (b), to Calculate the **APER** and **E(AER).** (To calculate the latter, you should use Lachen-bruch's holdout procedure.]

Solution:

b) sample Squared Distances

Assign x to the population $\boldsymbol{\pi}_i$ for which $-\frac{1}{2}\boldsymbol{D}_i^2(\boldsymbol{x_0}) + \ln \boldsymbol{p}_i$ is largest .

$$D_i^2(x) = (x - \bar{x}_i)' S_{pooled}^{-1} (x - \bar{x}_i)$$

With $\mathbf{x}_0' = [3.21, 497]$, the sample squared distances are

$$D_1^2(\mathbf{x}_0) = (\mathbf{x}_0 - \bar{\mathbf{x}}_1)'\mathbf{S}_{\text{pooled}}^{-1}(\mathbf{x}_0 - \bar{\mathbf{x}}_1)$$

$$= [3.21 - 3.40, \quad 497 - 561.23]\begin{bmatrix} 28.6096 & .0158 \\ .0158 & .0003 \end{bmatrix}\begin{bmatrix} 3.21 - 3.40 \\ 497 - 561.23 \end{bmatrix}$$

$$= 2.58$$

$$D_2^2(\mathbf{x}_0) = (\mathbf{x}_0 - \bar{\mathbf{x}}_2)'\mathbf{S}_{\text{pooled}}^{-1}(\mathbf{x}_0 - \bar{\mathbf{x}}_2) = 17.10$$

$$D_3^2(\mathbf{x}_0) = (\mathbf{x}_0 - \bar{\mathbf{x}}_3)'\mathbf{S}_{\text{pooled}}^{-1}(\mathbf{x}_0 - \bar{\mathbf{x}}_3) = 2.47$$

Since the distance from $\mathbf{x}_0' = [3.21, 497]$ to the group mean $\bar{\mathbf{x}}_3$ is smallest, we assign this applicant to π_3, borderline. ■

Or by use linear discriminate score

$$\hat{d}_i(x_0) = \bar{x}_i' S_{\text{Pooled}}^{-1} x_0 - \frac{1}{2}\bar{x}_i' S_{\text{Pooled}}^{-1} \bar{x}_i + \ln p_i \qquad (11-51)$$

$$\hat{d}_1(x_0) = 205.0467; \quad \hat{d}_2(x_0) = 197.8666; \quad \hat{d}_3(x_0) = \mathbf{205.1496}$$

Therefore, assign $x_0' = [3.21,\ 497]$ to Borderline population (third).

c) quadratic discriminate scores $\hat{d}_i^{\,Q}(x_0)$

$$\hat{\boldsymbol{d}}_i^{\,Q}(\boldsymbol{X_0}) = -\frac{1}{2}\ln|\boldsymbol{S_i}| - \frac{1}{2}(\boldsymbol{X_0} - \bar{\boldsymbol{X}}_i)'\,\boldsymbol{S_i^{-1}}(\boldsymbol{X_0} - \bar{\boldsymbol{X}}_i) + \ln \boldsymbol{P_i}$$

$$\hat{d}_1^{\,Q}(X_0) = \mathbf{-4.625117}; \qquad \hat{d}_2^{\,Q}(X_0) = -12.139287; \qquad \hat{d}_3^{\,Q}(X_0) = -6.934398$$

Therefore, assign $x_0' = [3.21,\ 497]$ to Admit population (first).

$$APER = 0.0353 \ ; \ E(APER) = 0.0470$$

d) The confusion matrix is :

	fitted		
truth	admit	border	notadmit
admit	27	4	0
border	1	25	0
notadmit	0	2	26

Apparent Error Rate: $APER = \frac{4+2+1}{31+28+26} = 0.0823$

e) leave-one-out cross validation for linear discriminant analysis

confusion matrix is :

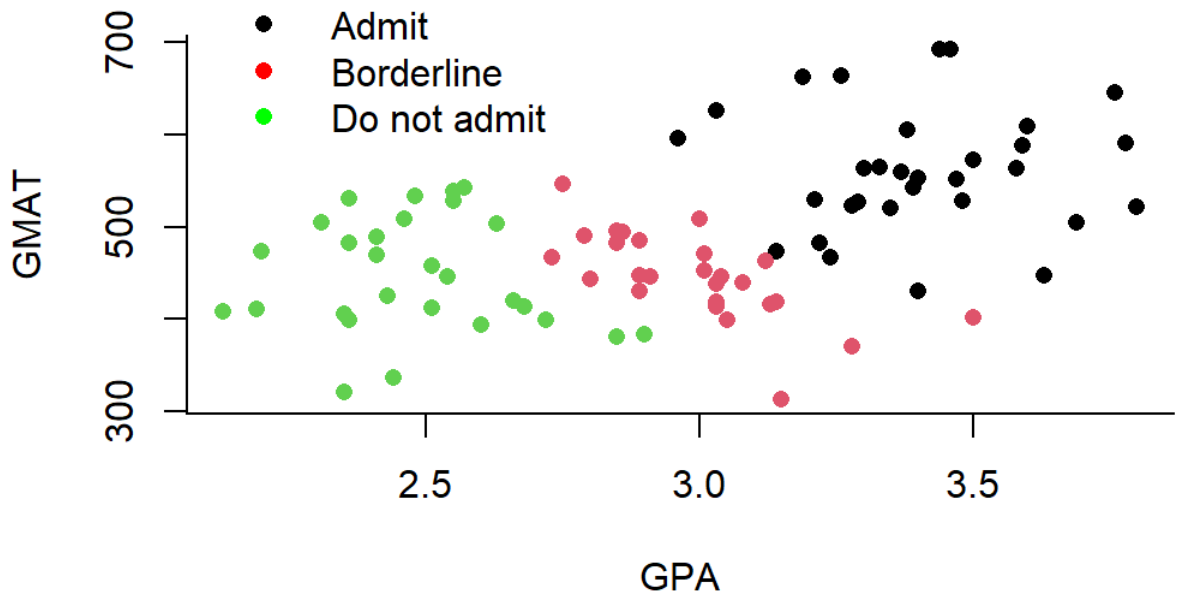
	fitted		
truth	admit	border	notadmit
admit	27	5	0
border	1	24	1
notadmit	0	2	26

$$E(APER) = \frac{5+1+1+2}{31+28+26} = \frac{9}{85} = \ 0.1059$$

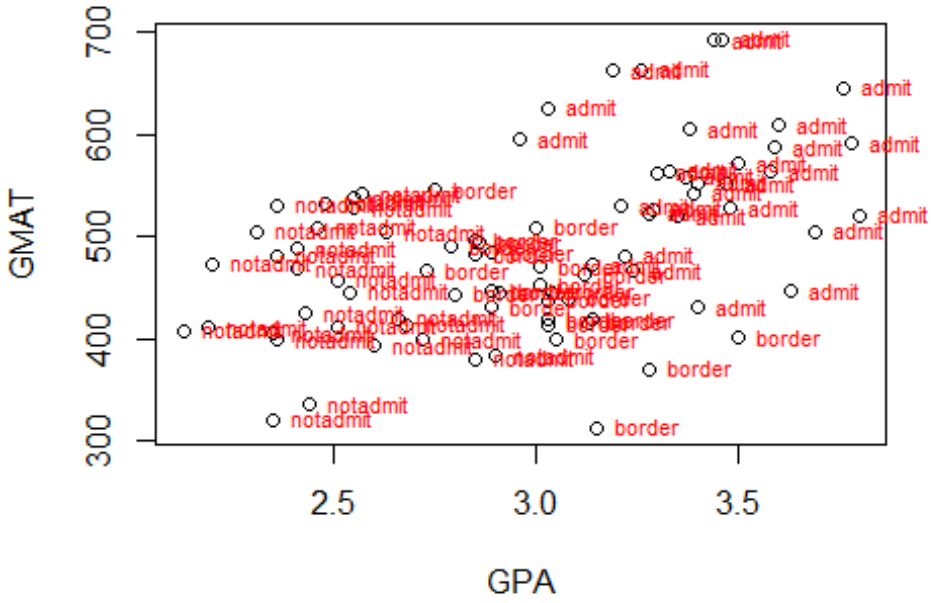
R code

```
##Exercise 5 (Example 11.11 page 635)
rm(list=ls())
admit=read.table(choose.files()) #T11-6.DAT
colnames(admit)=c("GPA","GMAT","Status")
admit$Status[admit$Status==1]<-"admit"
 admit$Status[admit$Status==2]<-"notadmit"
  admit$Status[admit$Status==3]<-"border"
admit[,3]<- as.factor(admit[,3])

## part a: plot the three groups in different colors
par(xpd=TRUE) #to enable things to be drawn outside the plot region.
plot(admit$GPA,admit$GMAT,xlab="GPA",ylab="GMAT",col=(admit$Status),pch=16
,bty='L')
#Add Legends to Plots
legend("topleft",inset= c(0.01,-0.15),legend = c("Admit","Borderline","Do
not admit"),col=c("black","red","green") ,pch=16,bty='n')
```




```
#pch:symbol to use. #bty:determined the type of box.
#### or plot by use
plot(admit$GPA,admit$GMAT,xlab="GPA",ylab="GMAT")
text(admit$GPA,admit$GMAT,labels =admit$Status,cex=0.7,pos=4,col="red")
```



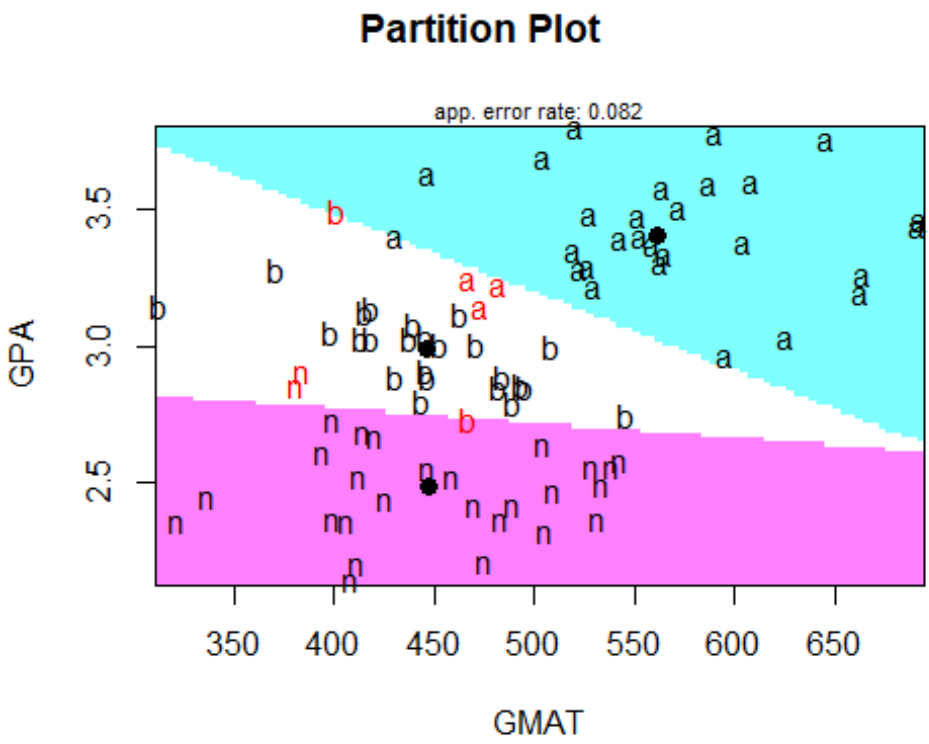

```
## part b: Linear Discriminate Analysis (lda)
 #install.packages("MASS") #for LDA and QDA functions
 #install.packages("klaR") #for partition plot function
library("MASS")
library("klaR")
LDA<- lda(Status ~ GPA+GMAT,data=admit,prior=c(1/3,1/3,1/3))
predict(LDA,newdata=data.frame(GPA=3.21,GMAT=497))

$class
[1] border
Levels: admit border notadmit

$posterior
      admit    border     notadmit
1 0.4741136 0.5255253 0.0003610973

$x
       LD1       LD2
1 1.359243  0.2879257

#assign new observation to borderline.
##LDA Partition Plots
partimat(Status ~ GPA+GMAT,method="lda",prec=100,data=admit)
```




```
# part d:Look at accuracy on the training data and construct confusion mat
rix
lda_fitted<-predict(LDA,newdata = admit)
(lda_table<- table(truth=admit$Status,fitted =lda_fitted$class))
```

```
          fitted
truth      admit border notadmit
  admit       27      4        0
  border       1     25        0
  notadmit     0      2       26
```

```
 n=sum(lda_table)
(APER<-(n-sum(diag(lda_table)))/n)
```

```
[1] 0.08235294
```

```
# Posterior probabilities of a categorize.
fit<-cbind.data.frame(admit,predict=lda_fitted$class,lda_fitted$posterior)
head(fit)
```

```
   GPA GMAT Status predict     admit       border     notadmit
1 2.96  596  admit   admit 0.5857197 4.093133e-01 4.966975e-03
2 3.14  473  admit  border 0.1201761 8.778041e-01 2.019771e-03
3 3.22  482  admit  border 0.3653729 6.342036e-01 4.235319e-04
4 3.29  527  admit   admit 0.8958479 1.041344e-01 1.762110e-05
5 3.69  505  admit   admit 0.9988036 1.196375e-03 7.041976e-10
6 3.46  693  admit   admit 0.9999820 1.796525e-05 6.953949e-11
```

```
## part e:leave-one-out cross validation for linear discriminant analysis.
# cannot run the predict function using the object with CV = TRUE.
lda_cv<-lda(Status~.,data=admit , CV=TRUE,prior=c(1/3,1/3,1/3))
```

```r
# confusion matrix of cross validation
(lda_table_cv<-table(truth=admit$Status, fitted=lda_cv$class))
```

```
          fitted
truth      admit border notadmit
  admit       26      5        0
  border       1     24        1
  notadmit     0      2       26
```

```r
(EAPER<-(n-sum(diag(lda_table_cv)))/n)
```

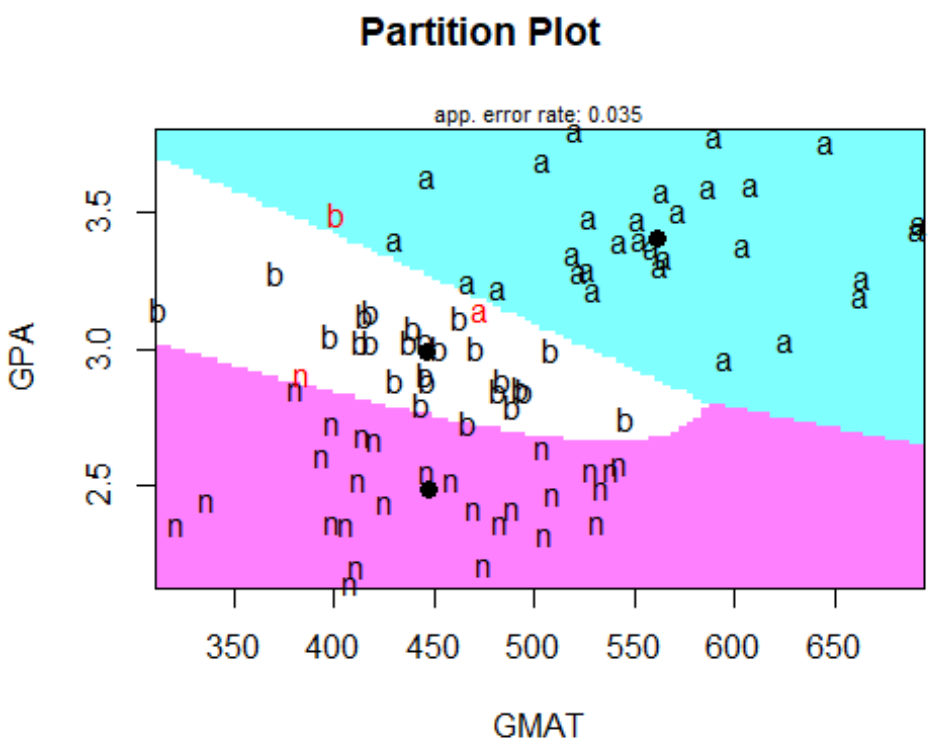
```
[1] 0.1058824
```

```r
## part c: The Quadratic Discriminant Analysis (QDA)
QDA<-qda(Status ~.,data=admit,prior=c(1/3,1/3,1/3),method="mle")
predict(QDA,newdata=data.frame(GPA=3.21,GMAT=497))
```

```
$class
[1] admit
Levels: admit border notadmit

$posterior
      admit     border     notadmit
1 0.9190847 0.08053886 0.0003764731
```

```r
partimat(Status ~ GPA+GMAT,method="qda",prec=100,data=admit)
```




```r
#Look at accuracy on the training data and construct confusion matrix.
qda_fitted<-predict(QDA,newdata = admit)
(qda_table<- table(truth=admit$Status,fitted =qda_fitted$class))
```

```
          fitted
truth      admit border notadmit
  admit       30      1        0
  border       1     25        0
  notadmit     0      1       27
```

```
n=sum(qda_table)
(APER.qda<-(n-sum(diag(qda_table)))/n)
```

```
[1] 0.03529412
```

```
#cross validation for QDA.
qda_cv<-qda(Status~.,data=admit , CV=TRUE,prior=c(1/3,1/3,1/3))
# confusion matrix of cross validation
(qda_table_cv<-table(truth=admit$Status, fitted=qda_cv$class))
```

```
          fitted
truth      admit border notadmit
  admit       30      1        0
  border       1     24        1
  notadmit     0      1       27
```

```
(EAPER.qda<-(n-sum(diag(qda_table_cv)))/n)
```

```
[1] 0.04705882
```

```
## part b: Construct sample Squared Distances or linear discriminate score
table(admit$Status)
```

```
   admit   border notadmit
      31       26       28
```

```
n1<-31
n2<-28
n3<-26

xbar1<-colMeans(admit[admit$Status== "admit",-3])
xbar2<-colMeans(admit[admit$Status=="notadmit" ,-3])
xbar3<-colMeans(admit[admit$Status=="border",-3])
xbar<- colMeans(admit[,-3])
S1<-var(admit[admit$Status=="admit",-3])
S2<- var(admit[admit$Status=="notadmit",-3])
S3<-var(admit[admit$Status=="border",-3])
(Sp <-((n1-1)*S1+(n2-1)*S2+(n3-1) * S3)/(n1+n2+n3-3))
```

```
             GPA        GMAT
GPA   0.03606795   -2.018759
GMAT -2.01875915 3655.901121
```

```
x0=c(3.21,497)

#sample Squared Distances
p<- 1/3
Dsq1=-0.5*t(x0-xbar1)%*%solve(Sp)%*%(x0-xbar1)+log(p)
Dsq2=-0.5*t(x0-xbar2)%*%solve(Sp)%*%(x0-xbar2)+log(p)
```

```
Dsq3=-0.5*t(x0-xbar3)%*%solve(Sp)%*%(x0-xbar3)+log(p)
c(Dsq1,Dsq2,Dsq3) #the observation is assign to borderline

[1] -2.415125 -9.595180 -2.312174

# posterior probability
p1<-exp(Dsq1)/(exp(Dsq1)+exp(Dsq2)+exp(Dsq3))
p2<-exp(Dsq2)/(exp(Dsq1)+exp(Dsq2)+exp(Dsq3))
p3<-exp(Dsq3)/(exp(Dsq1)+exp(Dsq2)+exp(Dsq3))
c(p1,p2,p3)

[1] 0.4741136282 0.0003610973 0.5255252745

#or linear discriminate score
d1<- t(xbar1)%*%solve(Sp)%*%x0-.5*t(xbar1)%*%solve(Sp)%*%xbar1+log(p)
d2<- t(xbar2)%*%solve(Sp)%*%x0-.5*t(xbar2)%*%solve(Sp)%*%xbar2+log(p)
d3<- t(xbar3)%*%solve(Sp)%*%x0-.5*t(xbar3)%*%solve(Sp)%*%xbar3+log(p)
c(d1,d2,d3) #the observation is assign to borderline

[1] 205.0467 197.8666 205.1496

##part c: Construct Quadratic discriminate score.
qd1= -0.5*log(det(S1))-.5*t(x0-xbar1)%*%solve(S1)%*%(x0-xbar1)+log(p)
qd2= -0.5*log(det(S2))-.5*t(x0-xbar2)%*%solve(S2)%*%(x0-xbar2)+log(p)
qd3= -0.5*log(det(S3))-.5*t(x0-xbar3)%*%solve(S3)%*%(x0-xbar3)+log(p)
c(qd1 ,qd2 ,qd3) #the observation is assign to Admit

[1]  -4.625117 -12.139287  -6.934398
```

Exercise 6: Example 11.13 page

Example 11.13 (Calculating Fisher's sample discriminants for three populations) Consider the observations on $p = 2$ variables from $g = 3$ populations given in Example 11.10. Assuming that the populations have a common covariance matrix Σ, let us obtain the Fisher discriminants. The data are

$$\pi_1 (n_1 = 3) \qquad \pi_2 (n_2 = 3) \qquad \pi_3 (n_3 = 3)$$

$$\mathbf{X}_1 = \begin{bmatrix} -2 & 5 \\ 0 & 3 \\ -1 & 1 \end{bmatrix}; \quad \mathbf{X}_2 = \begin{bmatrix} 0 & 6 \\ 2 & 4 \\ 1 & 2 \end{bmatrix}; \quad \mathbf{X}_3 = \begin{bmatrix} 1 & -2 \\ 0 & 0 \\ -1 & -4 \end{bmatrix}$$

$$\bar{\mathbf{x}}_1 = \begin{bmatrix} -1 \\ 3 \end{bmatrix}; \quad \bar{\mathbf{x}}_2 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}; \quad \bar{\mathbf{x}}_3 = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$$

$$\bar{\mathbf{x}} = \begin{bmatrix} 0 \\ \frac{5}{3} \end{bmatrix}; \quad \mathbf{B} = \sum_{i=1}^{3} (\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})' = \begin{bmatrix} 2 & 1 \\ 1 & 62/3 \end{bmatrix}$$

$$\mathbf{W} = \sum_{i=1}^{3} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)' = (n_1 + n_2 + n_3 - 3)\mathbf{S}_{\text{pooled}} = \begin{bmatrix} 6 & -2 \\ -2 & 24 \end{bmatrix}$$

R code

```
## Example 11.13
library("MASS")
data<-read.table(choose.files(),header = TRUE) #EX13.11.txt
data$Y<-as.factor(data$Y)
LDA1 <- lda(Y ~.,data=data)
LDA1

Call:
lda(Y ~ ., data = data)

Prior probabilities of groups:
        1         2         3
0.3333333 0.3333333 0.3333333

Group means:
  X1 X2
1 -1  3
2  1  4
3  0 -2

Coefficients of linear discriminants:
          LD1        LD2
X1 -0.3856092 -0.9380176
X2 -0.4945830  0.1119397

Proportion of trace:
   LD1    LD2
0.7602 0.2398

## variance
S1<-var(data[data$Y=="1",-3])
S2<-var(data[data$Y=="2",-3])
S3<-var(data[data$Y=="3",-3])
table(data$Y)


1 2 3
3 3 3

n1=3 ;n2=3 ;n3=3
#or use
n1<-nrow(data[data$Y=="1",])
SP<- ((n1-1)*S1+(n2-1)*S2+(n3-1)*S3 )/(n1+n2+n3-3)
round(SP,digits = 3)

       X1     X2
X1  1.000 -0.333
X2 -0.333  4.000

# mean
(m<-aggregate(.~Y,data,mean))

  Y X1 X2
1 1 -1  3
```

```
2 2  1  4
3 3  0 -2

(m<-t(LDA1$means))

    1 2  3
X1 -1 1  0
X2  3 4 -2

(Xbar<-colMeans(data[,-3])) #overall mean

      X1       X2
0.000000 1.666667
```