Module 7: Discrimination And Classification

This module covers discrimination analysis that provides one of the supervised learning techniques to construct optimal rule for classification.

Classify 150 renal patients measured by $\boldsymbol{x} = (x_1, \ldots, x_p)$ variables into either "progressive" or "non-progressive".

- Discrimination Analysis (Separation Analysis): To describe (graphically or algebraically) the differential features (e.g. biomarkers, patient's demographics etc.) of data from several known populations (e.g. Progressive and non-progressive). Technically, to find "discriminants" whose numerical values are such that the populations are separated as much as possible.

- Classification Analysis (Allocation Analysis): To develop a rule that enables us to allocate data cases (e.g. patients) into two or more labeled classes (e.g. progressive and non-progressive).

- In practice, these two tasks often overlap.

SEPARATION OF TWO POPULATIONS

Two populations:

- Population π_1: $f_1(\boldsymbol{x})$
- Population π_2: $f_2(\boldsymbol{x})$

The sample space is Ω.

Classification rule:

- R_1: the set of $\boldsymbol{x}$ for subjects being classified as π_1
- R_2: the set of $\boldsymbol{x}$ for subjects being classified as π_2

where $R_1 \cup R_2 = \Omega$.

Measures of classification accuracy:

- Conditional probabilities:

$$
\begin{aligned}
p(2|1) =& \text{Probabilty of classifying a subject as } \pi_2 \text{ when it is from } \pi_1 \\
=& P(\boldsymbol{x} \in R_2|\pi_1) = \int_{R_2} f_1(\boldsymbol{x})d\boldsymbol{x} \\
p(1|2) =& \text{Probabilty of classifying a subject as } \pi_1 \text{ when it is from } \pi_2 \\
=& P(\boldsymbol{x} \in R_1|\pi_2) = \int_{R_1} f_2(\boldsymbol{x})d\boldsymbol{x}
\end{aligned}
$$

- Marginal probabilities of the accuracy:

  Given the prior probabilities $p_1 = P(\pi_1)$ and $p_2 = P(\pi_2)$:

$$
P(\text{misclassifying a subject as } \pi_1) = p_2 p(1|2)
$$

$$
P(\text{misclassifying a subject as } \pi_2) = p_1 p(2|1)
$$

- Total probability of misclassification (TPM):

$$
\begin{aligned}
TPM &= p(1|2)p_2 + p(2|1)p_1 \\
&= p_2 \int_{R_1} f_2(\boldsymbol{x}) d\boldsymbol{x} + p_1 \int_{R_2} f_1(\boldsymbol{x}) d\boldsymbol{x}
\end{aligned}
$$

- Expected cost of misclassification (ECM)

  If misclassification cost is

  - Cost of misclassifying a subject as π_1 when it is actually from π_2: $c(1|2)$
  - Cost of misclassifying a subject as π_2 when it is actually from π_1: $c(2|1)$
  - Implicitly $c(1|1) = c(2|2) = 0$.

then

$$
\begin{aligned}
\text{ECM} &= c(1|2)p(1|2)p_2 + c(2|1)p(2|1)p_1 \\
&= c(1|2)p_2 \int_{R_1} f_2(\boldsymbol{x})d\boldsymbol{x} + c(2|1)p_1 \int_{R_2} f_1(\boldsymbol{x})d\boldsymbol{x} \\
&= \int_{R_1} c(1|2)f_2(\boldsymbol{x})p_2 d\boldsymbol{x} + \int_{R_2} c(2|1)f_1(\boldsymbol{x})p_1 d\boldsymbol{x} \\
&= \int_{R_1} \left[c(1|2)f_2(\boldsymbol{x})p_2 - c(2|1)f_1(\boldsymbol{x})p_1\right] d\boldsymbol{x} + c(2|1)p_1
\end{aligned}
$$

Question: How to find a classification rule R_1 (or R_2) that minimizes ECM (or TPM)?

Theorem 1 (Optimal classification rule) *R_1 minimizes ECM if*

$$R_1 = \{\boldsymbol{x} : c(2|1)f_1(\boldsymbol{x})p_1 \geq c(1|2)f_2(\boldsymbol{x})p_2\} = \left\{\boldsymbol{x} : \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} \geq \frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1}\right\}$$

Some special cases:

- *If* $\frac{p_1}{p_2} = 1$ *(a subject from the two populations with equal probabilities), then*

$$R_1 = \left\{ \boldsymbol{x} : \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} \geq \frac{c(1|2)}{c(2|1)} \right\}$$

- *If* $\frac{c(1|2)}{c(2|1)} = 1$ *(the costs of the two types of misclassification are equal), then*

$$R_1 = \left\{ \boldsymbol{x} : \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} \geq \frac{p_2}{p_1} \right\}$$

- *If* $\frac{c(1|2)}{c(2|1)} = \frac{p_2}{p_1} = 1$*, then*

$$R_1 = \left\{ \boldsymbol{x} : \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} \geq 1 \right\}$$

Remark 1

- The optimal classification only involves the ratios of the cost and the prior probabilities.
- The result can be extended to more than two populations.

Proof 1 If there is another classification rule given by R_1^* and R_2^* such that $R_1^* \cup R_2^* = \Omega$ and they are different from R_1 and R_2, the associated ECM is

$$ECM^* = \int_{R_1^*} [c(1|2)f_2(\boldsymbol{x})p_2 - c(2|1)f_1(\boldsymbol{x})p_1]\, d\boldsymbol{x} + c(2|1)p_1$$

Compare the two ECM's:

$$
\begin{aligned}
ECM - ECM^* &= \int_{R_1} \left[c(1|2) f_2(\boldsymbol{x}) p_2 - c(2|1) f_1(\boldsymbol{x}) p_1 \right] d\boldsymbol{x} - \\
&\quad \int_{R_1^*} \left[c(1|2) f_2(\boldsymbol{x}) p_2 - c(2|1) f_1(\boldsymbol{x}) p_1 \right] d\boldsymbol{x} \\
&= \int_{R_1 \cap (\overline{R_1 \cap R_1^*})} + \int_{R_1 \cap R_1^*} - \int_{R_1^* \cap (\overline{R_1 \cap R_1^*})} - \int_{R_1 \cap R_1^*} \\
&= \int_{R_1 \cap (\overline{R_1 \cap R_1^*})} - \int_{R_1^* \cap (\overline{R_1 \cap R_1^*})} \leq 0
\end{aligned}
$$

Discrimination of Two Normal Populations

Given that

$$f_1(\boldsymbol{x}) \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_2), \qquad f_2(\boldsymbol{x}) \sim N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

we can express R_1 in a more meaningful form.

Case 1: $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$

Theorem 2 (Discriminating two normal populations with equal covariances) *The R_1 that minimizes ECM is*

$$\begin{aligned} R_1 &= \left\{ \boldsymbol{x} : \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} \geq \frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1} \right\} \\ &= \left\{ \boldsymbol{x} : (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - \frac{1}{2} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) \geq \log \left(\frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1} \right) \right\} \end{aligned}$$

Equivalently, we can also do

$$\boldsymbol{\mu}_1'\boldsymbol{\Sigma}^{-1}\boldsymbol{x}-\frac{1}{2}\boldsymbol{\mu}_1'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_2) \geq \boldsymbol{\mu}_2'\boldsymbol{\Sigma}^{-1}\boldsymbol{x}-\frac{1}{2}\boldsymbol{\mu}_2'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_2)+\log\left(\frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1}\right).$$

When $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, and $\boldsymbol{\Sigma}$ are unknown, the sample version of R_1 is

$$R_1 = \left\{\boldsymbol{x} : (\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)'\boldsymbol{S}_{pooled}^{-1}\boldsymbol{x} - \frac{1}{2}(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)'\boldsymbol{S}_{pooled}^{-1}(\overline{\boldsymbol{x}}_1 + \overline{\boldsymbol{x}}_2) \geq \log\left(\frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1}\right)\right\}$$

where

$$\boldsymbol{S}_{pooled} = \frac{1}{n_1 + n_2 - 2}\left[(n_1 - 1)\boldsymbol{S}_1 + (n_2 - 1)\boldsymbol{S}_2\right]$$

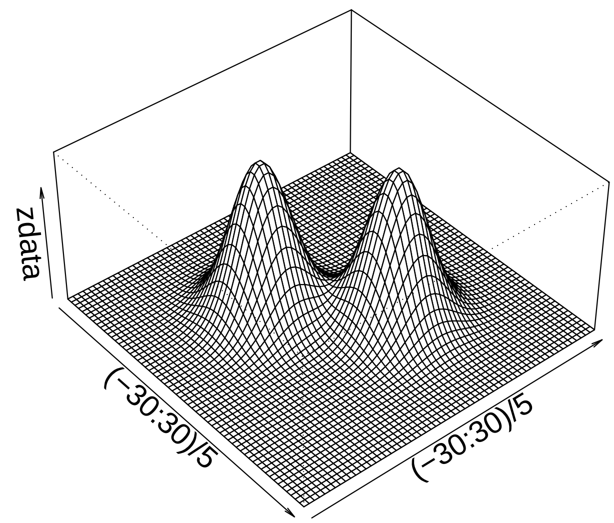
Proof 2

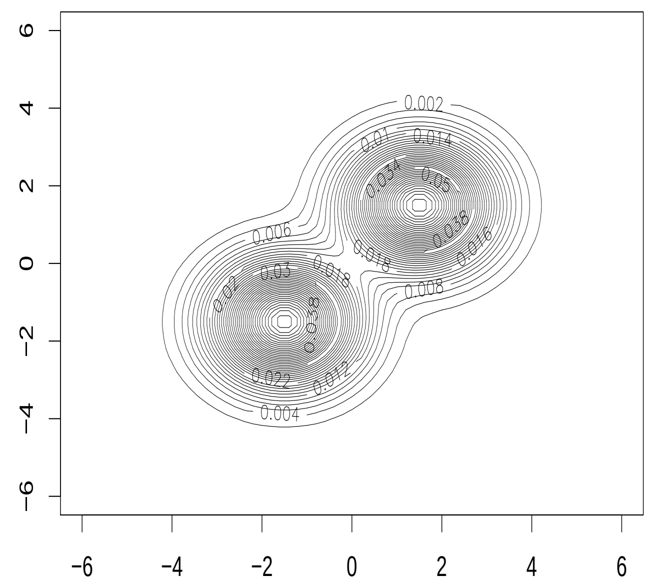
$$\begin{aligned}
\frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} &= \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)\right] \\
&= \exp\left[-\frac{1}{2}tr\left\{(\boldsymbol{x}-\boldsymbol{\mu}_1)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right\} + \frac{1}{2}tr\left\{(\boldsymbol{x}-\boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)\right\}\right] \\
&= \exp\left[-\frac{1}{2}tr\left\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)(\boldsymbol{x}-\boldsymbol{\mu}_1)'\right\} + \frac{1}{2}tr\left\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)(\boldsymbol{x}-\boldsymbol{\mu}_2)'\right\}\right] \\
&= \exp\left[tr\left(\boldsymbol{\Sigma}^{-1}\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)(\boldsymbol{x}-\boldsymbol{\mu}_1)' + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)(\boldsymbol{x}-\boldsymbol{\mu}_2)'\right\}\right)\right] \\
&= \exp\left[tr\left(\boldsymbol{\Sigma}^{-1}\left\{\boldsymbol{x}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_1)' - \frac{1}{2}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)'\right\}\right)\right] \\
&= \exp\left[(\boldsymbol{\mu}_1-\boldsymbol{\mu}_1)'\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_2)\right]
\end{aligned}$$

Example 1 (Graphical representation of R_1) *We run*

```
> zxy <- cbind(x=rep((-30:30)/5, rep(61, 61)), y = rep((-30:30)/5, 61))
```

```
> zdata <- matrix(.5*dmvnorm(zxy, mean = c(-1.5, -1.5),
  sigma = rbind(c(1, 0), c(0, 1))) + .5*dmvnorm(zxy, mean = c(1.5, 1.5),
  sigma = rbind(c(1, 0), c(0, 1))), ncol = 61, byrow = T)
> persp((-30:30)/5, (-30:30)/5, zdata, theta = 50, phi = 40, r = 10, expand = .5, ltheta = 50, lphi = 40)
> contour((-30:30)/5, (-30:30)/5, zdata, nlevels = 30)
```





0.002
0.01
0.014
0.034
0.05
0.038
0.016
0.006
0.018
0.008
0.03
0.018
0.02
0.038
0.022
0.012
0.004
-6
-4
-2
0
2
4
6

Remark 2

- R_1 is determined by a linear function of $\boldsymbol{x}$

- Let $\boldsymbol{a} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$. If $c(1|2) = c(2|1)$ and $p_1 = p_2$, then

$$R_1 = \left\{ \boldsymbol{x} : \boldsymbol{a}'\boldsymbol{x} - \frac{1}{2}\boldsymbol{a}'(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) \geq \log \left(\frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1} \right) \right\}$$
$$= \left\{ \boldsymbol{x} : \boldsymbol{a}'\boldsymbol{x} \geq \boldsymbol{a}' \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{2} \right\} = \left\{ y : y \geq \boldsymbol{a}' \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{2} \right\}$$

where $y = \boldsymbol{a}'\boldsymbol{x}$. In this case, if $\boldsymbol{x} \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$, then

$$p(2|1) = P\left(y < \boldsymbol{a}' \ \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{2} \middle| \, y \sim N(\boldsymbol{a}'\boldsymbol{\mu}_1, \boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}) \right)$$
$$= P\left(z \leq -\frac{1}{2}\sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)} \right) = \Phi\left(-\frac{1}{2}\Delta \right)$$

Similarly $p(1|2) = \Phi\left(-\frac{1}{2}\Delta\right)$.

  - Both are decreasing when Δ increases.
  - When $\Delta = 0$, $p(1|2) = p(2|1) = 0.5$.

- The sample version is

$$R_1 = \left\{ \boldsymbol{x} : \boldsymbol{a}'\boldsymbol{x} - \frac{1}{2}\boldsymbol{a}'(\overline{\boldsymbol{x}}_1 + \overline{\boldsymbol{x}}_2) \geq \log\left(\frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1} \right) \right\}$$

where $\boldsymbol{a} = \boldsymbol{S}_{pooled}^{-1}(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)$.

Case 2: $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$

Theorem 3 (Discriminating two normal populations with unequal covariances) *The R_1 that minimizes ECM is*

$$\begin{aligned} R_1 &= \left\{ \boldsymbol{x} : \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_2) \right] \left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|} \right)^{-1/2} \right. \\ &\quad \left. \geq \frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1} \right\} \\ &= \left\{ \boldsymbol{x} : -\frac{1}{2}\boldsymbol{x}'(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1})\boldsymbol{x} + (\boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\mu}_2\boldsymbol{\Sigma}_2^{-1})\boldsymbol{x} - k \geq \log\left(\frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1} \right) \right\} \end{aligned}$$

where the constant

$$k = \frac{1}{2}\log\left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|}\right) + \frac{1}{2}(\boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2'\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2)$$

The sample version is

$$R_1 = \left\{\boldsymbol{x} : -\frac{1}{2}\boldsymbol{x}'(\boldsymbol{S}_1^{-1} - \boldsymbol{S}_2^{-1})\boldsymbol{x} + (\overline{\boldsymbol{x}}_1'\boldsymbol{S}_1^{-1} - \overline{\boldsymbol{x}}_2\boldsymbol{S}_2^{-1})\boldsymbol{x} - k \geq \log\left(\frac{c(1|2)}{c(2|1)}\frac{p_2}{p_1}\right)\right\}$$

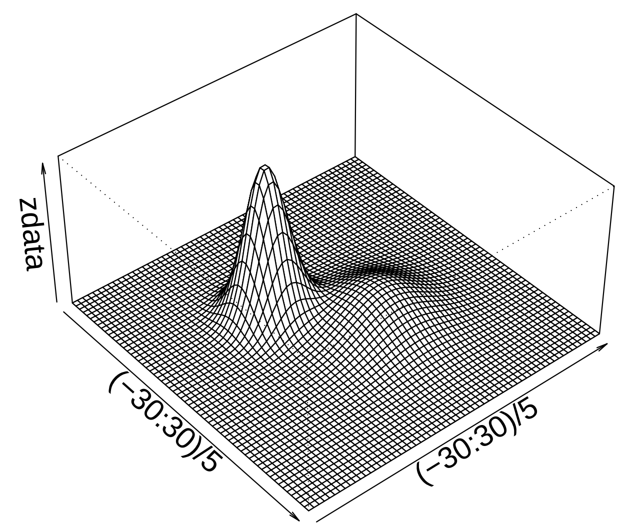
Remark 3

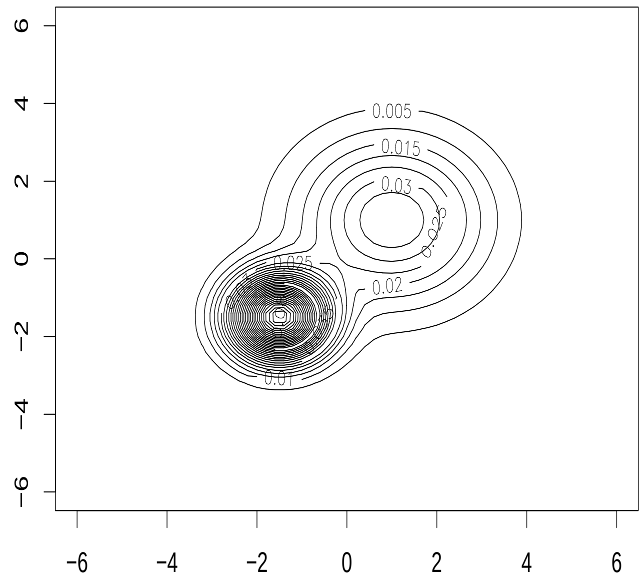
- R_1 is determined by a quadratic curve instead of a straight line.
- The discrimination rule may be more sensitive to the normal assumption.

Example 2 (Graphical representation of R_1 with unequal covariances)
We run

```
> zxy <- cbind(x=rep((-30:30)/5, rep(61, 61)), y = rep((-30:30)/5, 61))
> zdata <- matrix(.5*dmvnorm(zxy, mean = c(-1.5, -1.5),
           sigma = rbind(c(.5, 0), c(0, .5))) + .5*dmvnorm(zxy, mean = c(1, 1),
```

```
          sigma = rbind(c(2, 0), c(0, 2))), ncol = 61, byrow = T)
> persp((-30:30)/5, (-30:30)/5, zdata, theta = 50, phi = 40, r = 10, expand = .5, ltheta = 50, lphi = 40)
> contour((-30:30)/5, (-30:30)/5, zdata, nlevels = 30)
```





Example 3 (Example 11.8) *We assume equal prior probabilities, equal costs, and equal covariance structure.*

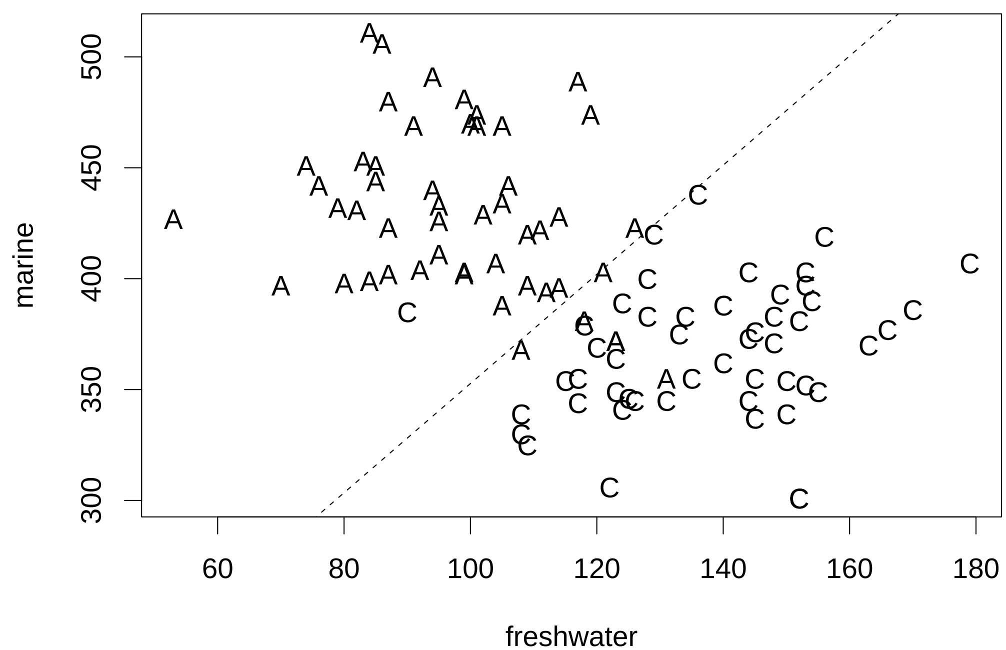
```
> t11.2 <- read.table("T11-2.DAT", header = F, col.names = c("group", "gender", "freshwater", "marine"))
> t11.2$group <- factor(t11.2$group, labels = c("Alaskan", "Canadian"))
> t11.2$gender <- factor(t11.2$gender, labels = c("female", "male"))
> t11.2

       group gender freshwater marine
```

```
  1  Alaskan   male        108    368
  2  Alaskan female        131    355
  3  Alaskan female        105    469
  4  Alaskan   male         86    506
  5  Alaskan female         99    402
  6  Alaskan   male         87    423
  7  Alaskan female         94    440
  8  Alaskan   male        117    489
  9  Alaskan   male         79    432
 10  Alaskan female         99    403
 ... ...

> zmu1 <- colMeans(t11.2[t11.2$group=="Alaskan", 3:4])
> zmu1

freshwater     marine
     98.38     429.66

> zmu2 <- colMeans(t11.2[t11.2$group=="Canadian", 3:4])
> zmu2

freshwater     marine
    137.46     366.62

> zs1 <- var(t11.2[t11.2$group=="Alaskan", 3:4])
> zs1

           freshwater    marine
freshwater   260.6078 -188.0927
marine      -188.0927 1399.0861

> zs2 <- var(t11.2[t11.2$group=="Canadian", 3:4])
> zs2

           freshwater   marine
```

```
freshwater   326.0902 133.5049
marine       133.5049 893.2608

> zs <- (49*zs1 + 49*zs2)/98
> zs

           freshwater     marine
freshwater  293.34898  -27.29388
marine      -27.29388 1146.17347

> za <- solve(zs) %*% (zmu1 - zmu2)
> za

                  [,1]
freshwater -0.12838726
marine      0.05194311

> t(za) %*% (zmu1 + zmu2)/2

         [,1]
[1,] 5.541204
```

Thus

$$R_1 = \left\{ \boldsymbol{x} : \boldsymbol{a}'\boldsymbol{x} \geq \boldsymbol{a}' \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{2} \right\} = \{ \boldsymbol{x} : -0.128x_1 + .052x_2 \geq 5.541 \}$$


marine
freshwater
300
350
400
450
500
60
80
100
120
140
160
180

```
> zres <- (as.matrix(t11.2[, 3:4]) %*% za) >= (t(za) %*% (zmu1 + zmu2)/2)[1, 1]
> table(zres, t11.2$group)

zres    Alaskan Canadian
  FALSE       6       49
  TRUE       44        1
```

To class a new Salmon with the first-year freshwater growth of 100in and the first-year marine growth of 400in

```
> c(100, 400) %*% za > 5.541

     [,1]
[1,] TRUE
```

R has functions `lda` and `qda` and S+ has a function `discrim` for this discrimination analysis

```
> z <- discrim(group ~ freshwater + marine, data = t11.2)
> z

Call:
discrim(group ~ freshwater + marine, data = t11.2)

Group means:
         freshwater marine  N Priors
 Alaskan      98.38 429.66 50    0.5
Canadian     137.46 366.62 50    0.5

Covariance Structure: homoscedastic
           freshwater    marine
freshwater   293.3490   -27.294
    marine             1146.173

Constants:
   Alaskan  Canadian
```

```
 -101.3765 -95.83531

Linear Coefficients:
             Alaskan  Canadian
freshwater 0.3710689 0.4994562
    marine 0.3837010 0.3317579
```

The output contains the three group means $\bar{\boldsymbol{x}}_1$, $\bar{\boldsymbol{x}}_2$, *and the common covariance matrix* $\boldsymbol{S}_{pooled}$. *It also contains the two linear discriminant functions* $\hat{d}_1(\boldsymbol{x})$, $\hat{d}_2(\boldsymbol{x})$ *where*

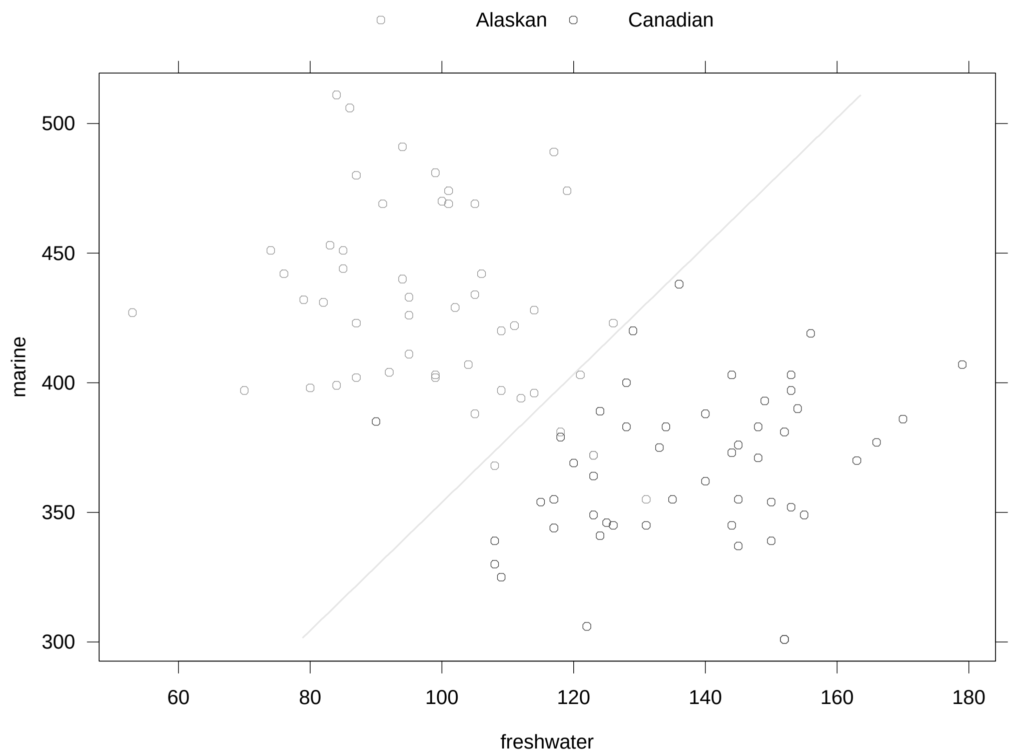
$$\hat{d}_1(\boldsymbol{x}) = \boldsymbol{\mu}_1' \boldsymbol{S}_{pooled}^{-1} \boldsymbol{x} - \frac{1}{2} \boldsymbol{\mu}_1' \boldsymbol{S}_{pooled}^{-1} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) = -101.3765 + 0.3710689 x_1 + 0.3837010 x_2$$

$$\hat{d}_2(\boldsymbol{x}) = \boldsymbol{\mu}_2' \boldsymbol{S}_{pooled}^{-1} \boldsymbol{x} - \frac{1}{2} \boldsymbol{\mu}_2' \boldsymbol{S}_{pooled}^{-1} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) = -95.83531 + 0.4994562 x_1 + 0.3317579 x_2$$

The set of $\boldsymbol{x}$ *that will be classified as population* π_1 *is given by*

$$R_1 = \{\boldsymbol{x} : \hat{d}_1(\boldsymbol{x}) > \hat{d}_2(\boldsymbol{x})\}$$

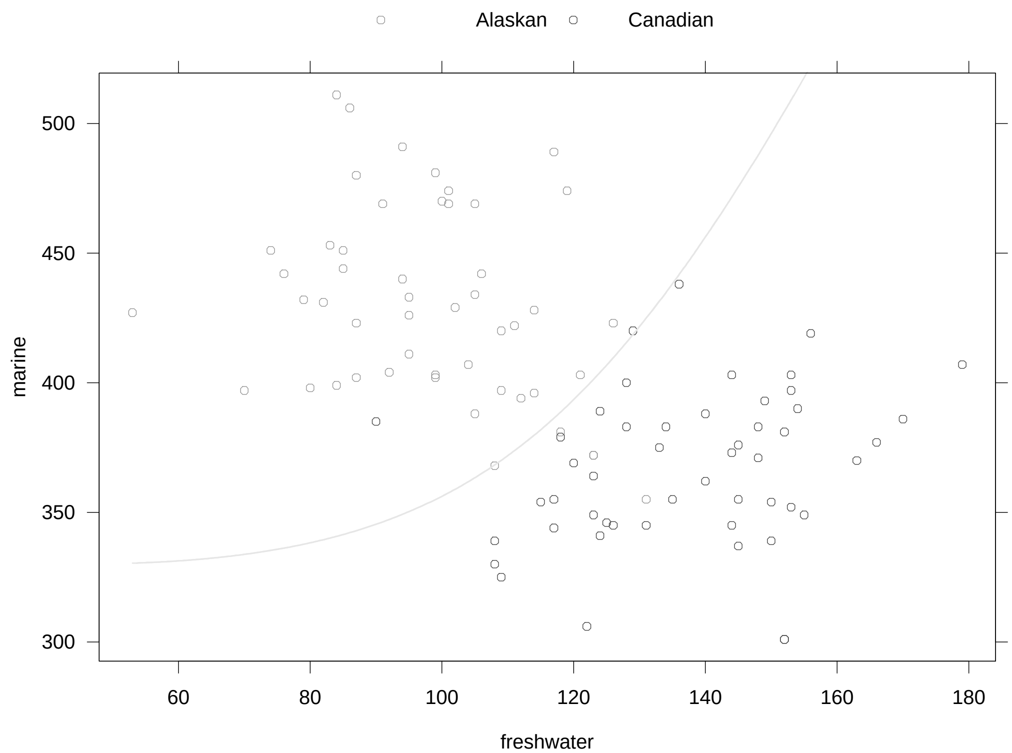
```
> plot(z)
```



If we do not make the equal covariance assumption, we can obtain the quadratic discrimination function:

```
> z <- discrim(group ~ marine + freshwater, data = t11.2, family = Classical("heteroscedastic"))
> z

Call:
discrim(group ~ marine + freshwater, data = t11.2, family = Classical(
"heteroscedastic"))

Group means:
         marine freshwater  N Priors
```

```
 Alaskan 429.66      98.38 50    0.5
Canadian 366.62     137.46 50    0.5

Covariance Structure: heteroscedastic
Group: Alaskan
              marine freshwater
    marine  1399.086  -188.0927
freshwater            260.6078
Group: Canadian
             marine freshwater
    marine 893.2608   133.5049
freshwater            326.0902

Constants:
  Alaskan  Canadian
 -124.823 -93.34938
Linear Coefficients:
             Alaskan  Canadian
    marine 0.3963058 0.3700709
freshwater 0.6635344 0.2700287

Quadratic coefficents:
group: Alaskan
                  marine   freshwater
    marine -0.0003957791 -0.000285652
freshwater               -0.002124760
group: Canadian
                  marine   freshwater
    marine -0.0005962301  0.000244103
freshwater               -0.001633257

> plot(z)
```



R has a function `lda` *in MASS library that can also perform above discrimination analysis*

```
> z <- lda(group ~ freshwater + marine, data = t11.2)
> z

Call:
lda(group ~ freshwater + marine, data = t11.2)

Prior probabilities of groups:
 Alaskan Canadian
     0.5      0.5
```

```
Group means:
         freshwater marine
Alaskan       98.38 429.66
Canadian     137.46 366.62

Coefficients of linear discriminants:
                   LD1
freshwater  0.04458572
marine     -0.01803856
```

(The coefficents from `lda` *can be obtained from the coefficents from* `discrim` *divided by the square root of* $\boldsymbol{a}'\boldsymbol{S}_{pooled}\boldsymbol{a}$ *to ensure* $\boldsymbol{a}'\boldsymbol{x}$ *has the variance 1. The signs are opposite.)*

```
> t(za) %*% zs %*%za
         [,1]
[1,] 8.291868
> za/sqrt(8.292)
                  [,1]
freshwater -0.04458536
marine      0.01803841
> predict(z, newdata = data.frame(freshwater = 100, marine = 400))

$class
[1] Alaskan
Levels: Alaskan Canadian

$posterior
    Alaskan   Canadian
1 0.9166222 0.08337776
$x
         LD1
1 -0.8325277
```

EVALUATING SAMPLE CLASSIFICATION FUNCTIONS

1. AER

   Given a sample classification rule $\widehat{R}_1$, the actual error rate (AER) is

$$\text{AER} = TPM(\widehat{R}_1) = p_2 \int_{\widehat{R}_1} f_2(\boldsymbol{x})d\boldsymbol{x} + p_1 \int_{\widehat{R}_2} f_1(\boldsymbol{x})d\boldsymbol{x}$$

   Problem: $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are unknown.

2. APER

   Let

$$\begin{aligned} n_1 &= \text{number of subjects in } \pi_1 \\ n_{1M} &= \text{number of subjects in } \pi_1 \text{ misclassified as } \pi_2 \\ n_2 &= \text{number of subjects in } \pi_2 \\ n_{2M} &= \text{number of subjects in } \pi_2 \text{ misclassified as } \pi_1 \end{aligned}$$

Then

$$\text{Apparent error rate (APER)} = \frac{n_{1M} + n_{2M}}{n_1 + n_2}$$

Example 4 (APER) *In the previous example:*

$$APER = \frac{6+1}{50+50} = .07$$

Pros: It is easy and does not require any parametric assumptions

Cons: APER may underestimate AER.

3. Modified APER (cross validation)
   (a) Randomly split data into a training sample and a validation sample
   (b) Construct classification rule/function from the training sample
   (c) Compute APER from the validation sample

   Pros:

- It does not depend on any parametric assumptions
- It does not underestimate AER

Cons:

- It requires large samples
- The classification function evaluated is not the classification function of interest.

4. "Holdout" procedure (jackknife procedure)

   - Omit one subject (holdout subject) from π_1 and construct the classification function based on the $n_1 - 1$ subjects (training dataset)
   - Classify the holdout subject using the classification function in above step
   - Repeat above two steps for all subjects in π_1 and denote the number of holdout subjects in π_1 that are misclassified to π_2 as $n_{1M}^{(H)}$.

- Repeat above steps for subjects in π_2 and obtain $n_{2M}^{(H)}$.
- Estimate AER by

$$\frac{n_{1M}^{(H)} + n_{2M}^{(H)}}{n_1 + n_2}$$

Example 5 (Holdout procedure) *Following is based on an R function:*

```
> z <- discrim(group ~ freshwater + marine, data = t11.2)
> z2 <- crossvalidate(z)
> z2

      groups   Alaskan  Canadian
  1 Canadian 0.3948409 0.6051591
  2 Canadian 0.0117575 0.9882425
  3  Alaskan 0.9949318 0.0050682
  4  Alaskan 0.9999539 0.0000461
  5  Alaskan 0.9302454 0.0697546
 ... ...
```

Then table the memberships:

```
> table(t11.2$group, z2$groups)

         Alaskan Canadian
 Alaskan      44        6
Canadian       1       49
```

Incidentally it is the same as the apparent error rate.

If we do not assume equal covariances, APER can be found from:

```
> z <- discrim(group ~ marine + freshwater, data = t11.2, family = Classical("heteroscedastic"))
> table(crossvalidate(z)$group, t11.2$group)

         Alaskan Canadian
 Alaskan      45        3
Canadian       5       47
```

R steps are given as follows

```
> z <- lda(group ~ freshwater + marine, data = t11.2, CV = T)
> table(z$class, t11.2$group)

           Alaskan Canadian
  Alaskan       44        1
  Canadian       6       49
```

For a quadratic discrimination function,

```
> z <- qda(group ~ freshwater + marine, data = t11.2, CV = T)
> table(z$class, t11.2$group)

           Alaskan Canadian
  Alaskan       45        3
  Canadian       5       47
```

Classification with Several Populations

Populations:	$\pi_1, \pi_2, \ldots, \pi_g$
Population distributions:	$f_1(\boldsymbol{x}), \ldots, f_g(\boldsymbol{x})$
Prior probabilities:	$p_1, p_2, \ldots, p_g$
Classification rule:	$R_1, R_2, \ldots, R_g$
Probabilties of misclassification:	$p(k\|i) = \int_{R_k} f_i(\boldsymbol{x}) d\boldsymbol{x}$ with $\sum_{k=1}^{g} p(k\|i) = 1$
Costs of misclassification:	$c(k\|i)$ with $c(i\|i) = 0$

Expected cost of misclassifying subjects from π_i to π_k, $k \neq i$:

$$ECM(i) = \sum_{k \neq i} p(k|i) c(k|i)$$

Expected cost of misclassification under current classification rule:

$$ECM = \sum_{i=1}^{g} p_i ECM(i) = \sum_{i=1}^{g} p_i \left(\sum_{k \neq i} p(k|i)c(k|i) \right)$$
$$= \sum_{i=1}^{g} p_i \left(\sum_{k \neq i} c(k|i) \int_{R_k} f_i(\boldsymbol{x}) d\boldsymbol{x} \right)$$

Theorem 4 (Optimal classification rule for several populations)
$R_1, R_2, \ldots, R_g$ *minimizes ECM if*

$$R_k = \{\boldsymbol{x} : \ell_k(\boldsymbol{x}) < \ell_i(\boldsymbol{x}), i = 1, \ldots, g, i \neq k\}$$

where

$$\ell_k(\boldsymbol{x}) = \sum_{j=1, j \neq k}^{g} p_j c(k|j) f_j(\boldsymbol{x}),$$ *(average loss of misclassifying subjects into* π_k*)*

Special cases:

1. *If $g = 2$, then*

$$R_1 = \{\boldsymbol{x} : \ell_1(\boldsymbol{x}) < \ell_2(\boldsymbol{x})\} = \{\boldsymbol{x} : p_2 c(1|2) f_2(\boldsymbol{x}) < p_1 c(2|1) f_1(\boldsymbol{x})\}$$
$$= \left\{\boldsymbol{x} : \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} > \frac{p_2 c(1|2)}{p_1 c(2|1)}\right\}$$

and

$$R_2 = \left\{\boldsymbol{x} : \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} < \frac{p_2 c(1|2)}{p_1 c(2|1)}\right\}$$

2. *If $g = 3$ and* $c(1|2) = c(1|3) = c(2|1) = c(2|3) = c(3|1) = c(3|2) = 1$, *then we really compare among mixtures of the other two populations. See the diagram on the board.*

3. *If $c(k|i) = 1$ for all $k \neq i$, then*

$$R_k = \left\{ \boldsymbol{x} : \sum_{j=1, j \neq k}^{g} p_j f_j(\boldsymbol{x}) < \sum_{j=1, j \neq i}^{g} p_j f_j(\boldsymbol{x}), i = 1, \ldots, g, i \neq k \right\}$$

$$= \left\{ \boldsymbol{x} : \begin{array}{cc} p_1 f_1(\boldsymbol{x}) < p_k f_k(\boldsymbol{x}) & i = 1 \\ \vdots & i \neq k \\ p_g f_g(\boldsymbol{x}) < p_k f_k(\boldsymbol{x}) & i = g \end{array} \right\}$$

$$= \{ \boldsymbol{x} : p_k f_k(\boldsymbol{x}) > p_i f_i(\boldsymbol{x}), i \neq k \}$$

Note that the posterior probability $p(k|\boldsymbol{x}) \propto p_k f_k(\boldsymbol{x})$.

4. *If $f_i(\boldsymbol{x}) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, and $c(k|i) = 1$ for $k \neq i$, then*

$$R_k = \{ \boldsymbol{x} : p_k f_k(\boldsymbol{x}) > p_i f_i(\boldsymbol{x}), i \neq k \}$$

$$= \left\{ \boldsymbol{x} : d_k^Q(\boldsymbol{x}) > d_i^Q(\boldsymbol{x}), i \neq k \right\}$$

where

$$d_i^Q(\boldsymbol{x}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) + \log p_i,$$

(a quadratic function of $\boldsymbol{x}$)

If $\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_g$, then $R_k = \{\boldsymbol{x} : d_k(\boldsymbol{x}) > d_i(\boldsymbol{x}), i \neq k\}$

$$d_i(\boldsymbol{x}) = \boldsymbol{\mu}_i'\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_i'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \log p_i, \text{ (a linear function of } \boldsymbol{x}\text{)}$$

The sample version replaces $\boldsymbol{\mu}_i$ with $\overline{\boldsymbol{x}}_i$, $\boldsymbol{\Sigma}_i$ with $\boldsymbol{S}_i$, and $\boldsymbol{\Sigma}$ with $\boldsymbol{S}_{pooled}$.

If $p_1 = \cdots = p_g$ with $\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_g$, then

$$d_i(\boldsymbol{x}) = -\frac{1}{2}D_i^2(\boldsymbol{x}), \quad \text{Distance between } \boldsymbol{x} \text{ and } \overline{\boldsymbol{x}}_i$$

and R_k implies assigning $\boldsymbol{x}$ to the closest population.

Example 6 (Example 11.10) *Form a data.frame object:*

```
> e11.10 <- data.frame(x1 = c(-2, 0, -1, 0, 2, 1, 1, 0, -1),
            x2 = c(5, 3, 1, 6, 4, 2, -2, 0, -4),
            group = factor(c(rep(1, 3), rep(2, 3), rep(3, 3))))
> e11.10

  x1 x2 group
1 -2  5     1
2  0  3     1
3 -1  1     1
4  0  6     2
5  2  4     2
6  1  2     2
7  1 -2     3
8  0  0     3
9 -1 -4     3
```

Assuming a common covariance matrix, but unequal prior probabilities for the groups:

```
> z <- discrim(group ~ x1 + x2, data = e11.10, prior = c(.25, .25, .5))
> z

Call:
discrim(structure(.Data = group ~ x1 + x2, class = "formula"), data = e11.10,
prior = c(0.25, 0.25, 0.5))
Group means:
  x1 x2 N Priors
1 -1  3 3   0.25
2  1  4 3   0.25
3  0 -2 3   0.50
Covariance Structure: homoscedastic
           x1        x2
```

```
x1  1.0000000 -0.333333
x2            4.000000
Constants:
        1        2         3
 -2.80058 -4.30058 -1.207433
Linear Coefficients:
            1        2          3
x1 -0.7714286 1.371429 -0.1714286
x2  0.6857143 1.114286 -0.5142857
```

The output contains the three group means $\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \bar{\boldsymbol{x}}_3$*, and the common covariance matrix* $\boldsymbol{S}_{pooled}$*. It also contains the three linear discriminant functions* $\hat{d}_1(\boldsymbol{x})$*,* $\hat{d}_2(\boldsymbol{x})$*,* $\hat{d}_3(\boldsymbol{x})$ *where*

$$\hat{d}_1(\boldsymbol{x}) = -2.80058 - 0.7714286x_1 + 0.6857143x_2$$

$$\hat{d}_2(\boldsymbol{x}) = -4.30058 + 1.371429x_1 + 1.114286x_2$$

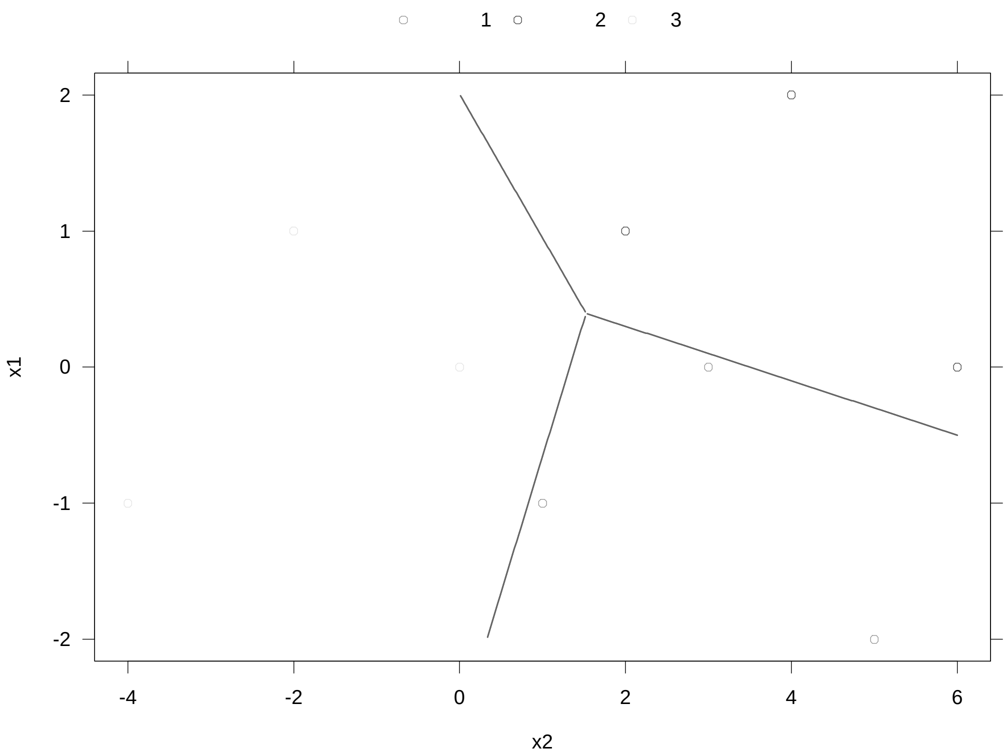
$$\hat{d}_3(\boldsymbol{x}) = -1.207433 - 0.1714286x_1 - 0.5142857x_2$$

The set of $\boldsymbol{x}$ *that will be classified as population* π_k *is given by*

$$R_k = \{\boldsymbol{x} : \hat{d}_k(\boldsymbol{x}) > \hat{d}_i(\boldsymbol{x}), \text{ for } i \neq k\}$$

To view the partition of the sample space:

```
> plot(z)
```



To allocate a new object $\boldsymbol{x}_0 = (-2, -1)$*, use*

```
> predict(z, newdata = data.frame(x1 = -2, x2 = -1))

  groups        X1        X2        X3
1      3 0.1688845 0.0003379 0.8307776
```

Instead of reporting the values of $\hat{d}_1(\boldsymbol{x}_0)$, $\hat{d}_2(\boldsymbol{x}_0)$, $\hat{d}_3(\boldsymbol{x}_0)$, `predict` *reports the posterior probabilities of* $\boldsymbol{x}_0$ *being in the three groups. To obtain the values of* $\hat{d}_1(\boldsymbol{x}_0)$, $\hat{d}_2(\boldsymbol{x}_0)$, $\hat{d}_3(\boldsymbol{x}_0)$*, do the following:*

```
> c(-2, -1) %*% coef(z)$linear.coefficients + coef(z)$constants
```

```
            1         2        3
[1,] -1.943437 -8.157723 -0.35029
```

Fisher's Linear Discriminant Analysis

Fisher's idea: Linearly transform high dimensional variables $\boldsymbol{x}$ into a univariate y

Linear transformation:

$$y = \boldsymbol{a}'\boldsymbol{x}$$

Question: How to choose $\boldsymbol{a}$ and determine R_1 based on y?

Given $\boldsymbol{a}$:

- population π_1: $\boldsymbol{x}_{11}, \ldots, \boldsymbol{x}_{1n_1} \xrightarrow{\boldsymbol{a}} y_{11}, \ldots, y_{1n_1} \longrightarrow \overline{\boldsymbol{y}}_1$
- population π_2: $\boldsymbol{x}_{21}, \ldots, \boldsymbol{x}_{2n_2} \xrightarrow{\boldsymbol{a}} y_{21}, \ldots, y_{2n_2} \longrightarrow \overline{\boldsymbol{y}}_2$

The separation of the two populations can be measured by

$$\frac{|\overline{\boldsymbol{y}}_1 - \overline{\boldsymbol{y}}_2|}{s_y}, \qquad s_y^2 \text{ : pooled variance of } y'_{ij}s$$

We first find $\boldsymbol{a}$ that maximizes the separation of the two populations.

Theorem 5 (Fisher's linear discriminant function) *The linear coefficient $\boldsymbol{a}$ that maximizes the separation is*

$$\widehat{\boldsymbol{a}} = \boldsymbol{S}_{pooled}^{-1}(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)$$

The maximum separation is

$$\left.\frac{|\overline{\boldsymbol{y}}_1 - \overline{\boldsymbol{y}}_2|}{s_y}\right|_{\boldsymbol{a}=\widehat{\boldsymbol{a}}} = (\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)'\boldsymbol{S}_{pooled}^{-1}(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2) = D^2$$

It is the sample generalized squared distance between the two sample means. Assuming the variances of the two populations are equal, the optimal classification rule is

$$R_1 = \left\{\boldsymbol{x} : \boldsymbol{a}'\boldsymbol{x} \geq \frac{1}{2}\boldsymbol{a}'(\overline{\boldsymbol{x}}_1 + \overline{\boldsymbol{x}}_2)\right\}$$

Proof 3 Maximizing $\frac{|\overline{\boldsymbol{y}}_1 - \overline{\boldsymbol{y}}_2|}{s_y}$ is equivalent to maximizing $\frac{(\overline{\boldsymbol{y}}_1 - \overline{\boldsymbol{y}}_2)^2}{s_y^2}$.

Following the matrix maximization result:

$$\max_{\boldsymbol{a}} \frac{(\overline{\boldsymbol{y}}_1 - \overline{\boldsymbol{y}}_2)^2}{s_y^2} = \max_{\boldsymbol{a}} \frac{(\boldsymbol{a}'\overline{\boldsymbol{x}}_1 - \boldsymbol{a}'\overline{\boldsymbol{x}}_2)^2}{\boldsymbol{a}'\boldsymbol{S}_{pooled}\boldsymbol{a}} = \max_{\boldsymbol{a}} \frac{(\boldsymbol{a}'\boldsymbol{d})^2}{\boldsymbol{a}'\boldsymbol{S}_{pooled}\boldsymbol{a}},$$

(subject to $\boldsymbol{a}'\boldsymbol{S}_{pooled}\boldsymbol{a} = 1$)

$$= \boldsymbol{d}'\boldsymbol{S}_{pooled}^{-1}\boldsymbol{d} = (\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)'\boldsymbol{S}_{pooled}^{-1}(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2) = D^2$$

and $\widehat{\boldsymbol{a}} = \boldsymbol{S}_{pooled}^{-1}\boldsymbol{d} = \boldsymbol{S}_{pooled}^{-1}(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)$.

Remark 4 • Fisher's linear discriminant function is the same as the best discriminant function under equal covariances multivariate normal distributions.

- D^2 is equivalent Hotelling's T^2 test statistic for testing H_0: $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$.

Example 7 (Using LDA) *See a previous example*

Question: How to generalize the idea to more than two populations?

Suppose that there are g populations:

$$\pi_1 : \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1 \Longrightarrow \boldsymbol{x}_{11}, \boldsymbol{x}_{12}, \ldots, \boldsymbol{x}_{1n_1} \Longrightarrow \overline{\boldsymbol{x}}_1, \boldsymbol{S}_1$$

$$\ldots \qquad \ldots \qquad \ldots$$

$$\pi_g : \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g \Longrightarrow \boldsymbol{x}_{g1}, \boldsymbol{x}_{g2}, \ldots, \boldsymbol{x}_{gn_g} \Longrightarrow \overline{\boldsymbol{x}}_g, \boldsymbol{S}_g$$

with $\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_g = \boldsymbol{\Sigma}$.

Consider $y = \boldsymbol{a}'\boldsymbol{x}$:

$$\pi_1 : \mu_{1y} = \boldsymbol{a}'\boldsymbol{\mu}_1, \sigma_y^2 = \boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}, \Longrightarrow y_{11}, y_{12}, \ldots, y_{1n_1}$$

$$\text{with } y_{1j} = \boldsymbol{a}'\boldsymbol{x}_{1j} \Longrightarrow \overline{y}_1 = \boldsymbol{a}'\overline{\boldsymbol{x}}_1, s_1^2 = \boldsymbol{a}'\boldsymbol{S}_1\boldsymbol{a}$$

$$\ldots \qquad \ldots \qquad \ldots$$

$$\pi_g : \mu_{gy} = \boldsymbol{a}'\boldsymbol{\mu}_g, \sigma_y^2 = \boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}, \Longrightarrow y_{g1}, y_{g2}, \ldots, y_{gn_g}$$

$$\text{with } y_{gj} = \boldsymbol{a}'\boldsymbol{x}_{gj} \Longrightarrow \overline{y}_g = \boldsymbol{a}'\overline{\boldsymbol{x}}_g, s_g^2 = \boldsymbol{a}'\boldsymbol{S}_g\boldsymbol{a}$$

We want to choose $\boldsymbol{a}$ so that the separation of $\pi_1, \pi_2, \ldots, \pi_g$ populations is maximized.

The idea is similar to ANOVA: Let $\overline{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{g} \boldsymbol{\mu}_i$ and $\overline{\mu}_y = \frac{1}{n}\sum_{i=1}^{g} \mu_{iy}$. Then find $\boldsymbol{a}$ such that

$$\max_{\boldsymbol{a}} \frac{\text{Variation between populations of } \pi_1, \pi_2, \ldots, \pi_g}{\text{Variation within populations of } \pi_1, \pi_2, \ldots, \pi_g}$$

$$= \max_{\boldsymbol{a}} \frac{\sum_{i=1}^{g}(\mu_{iy} - \overline{\mu}_y)^2}{\boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}} = \max_{\boldsymbol{a}} \frac{\sum_{i=1}^{g}(\boldsymbol{a}'\boldsymbol{\mu}_i - \boldsymbol{a}'\overline{\boldsymbol{\mu}})^2}{\boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}}$$

$$= \max_{\boldsymbol{a}} \frac{\boldsymbol{a}'\left[\sum_{i=1}^{g}(\boldsymbol{\mu}_i - \overline{\boldsymbol{\mu}})(\boldsymbol{\mu}_i - \overline{\boldsymbol{\mu}})\right]\boldsymbol{a}}{\boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}} = \max_{\boldsymbol{a}} \frac{\boldsymbol{a}'\boldsymbol{B}\boldsymbol{a}}{\boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}}, \qquad \text{(subject to } \boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a} = 1\text{)}$$

The sample version: Let

$$\overline{y} = \frac{\sum_{i=1}^{g}\sum_{j=1}^{n_i} y_{ij}}{\sum_{i=1}^{g} n_i} = \frac{1}{\sum_{i=1}^{g} n_i}\sum_{i=1}^{g} n_i \overline{y}_i, \quad s^2 = \frac{1}{n_1+\cdots+n_g-g}\sum_{i=1}^{g}(n_i-1)s_i^2$$

$$\overline{\boldsymbol{x}}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\boldsymbol{x}_i, \quad \overline{\boldsymbol{x}} = \frac{\sum_{i=1}^{g}\sum_{j=1}^{n_i}\boldsymbol{x}_{ij}}{\sum_{i=1}^{g} n_i} = \frac{1}{\sum_{i=1}^{g} n_i}\sum_{i=1}^{g} n_i \overline{\boldsymbol{x}}_i$$

$$\boldsymbol{B} = \sum_{i=1}^{g} n_i(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})'$$

$$\boldsymbol{W} = \sum_{i=1}^{g}(n_i-1)\boldsymbol{S}_i \text{ and } \widehat{\boldsymbol{\Sigma}} = \frac{\boldsymbol{W}}{n_1+\cdots+n_g-g} = \frac{\boldsymbol{W}}{n-g} = \boldsymbol{S}_{pooled}$$

then finding $\boldsymbol{a}$ is equivalent to

$$\max_{\boldsymbol{a}} \frac{\sum_{i=1}^{g} n_i(\overline{y}_i - \overline{y})^2}{s^2} = \max_{\boldsymbol{a}} \frac{\boldsymbol{a}'\boldsymbol{B}\boldsymbol{a}}{\boldsymbol{a}'\widehat{\boldsymbol{\Sigma}}\boldsymbol{a}} \Leftrightarrow \max_{\boldsymbol{a}} \frac{\boldsymbol{a}'\boldsymbol{B}\boldsymbol{a}}{\boldsymbol{a}'\boldsymbol{W}\boldsymbol{a}}, \quad \text{(subject to } \boldsymbol{a}'\boldsymbol{S}\boldsymbol{a} = 1)$$

Theorem 6 (Fisher's linear discriminants for several populations) *Let $\lambda_1, \lambda_2, \ldots, \lambda_s$ denote the $s \le \min(g-1, p)$ nonzero eigenvalues of $\boldsymbol{W}^{-1}\boldsymbol{B}$ and $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_s$ be the corresponding eigenvectors (scaled so that $\boldsymbol{e}_i' \boldsymbol{S}_{pooled} \boldsymbol{e}_i = 1$). Then $\boldsymbol{a} = \boldsymbol{e}_1$ maximizes the ratio*

$$\frac{\boldsymbol{a}'\boldsymbol{B}\boldsymbol{a}}{\boldsymbol{a}'\boldsymbol{W}\boldsymbol{a}}$$

We also call

$$\begin{aligned}
\boldsymbol{e}_1'\boldsymbol{x} &: \textit{the sample first discriminant} \\
\boldsymbol{e}_2'\boldsymbol{x} &: \textit{the sample second discriminant} \\
&\cdots \\
\boldsymbol{e}_s'\boldsymbol{x} &: \textit{the sample } s\textit{th discriminant}
\end{aligned}$$

Matrix Result 1 (Quadratic forms for points on the unit sphere) *Let $\boldsymbol{B}$ be a postive definite matrix with eigenvalues $\lambda_1 \ge \cdots \ge \lambda_p \ge 0$ and* *associated eigenvectors* $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_p$*. Then*

$$\max_{\substack{\boldsymbol{x}\neq\boldsymbol{0} \\ \boldsymbol{x}'\boldsymbol{x}=1}} \frac{\boldsymbol{x}'\boldsymbol{B}\boldsymbol{x}}{\boldsymbol{x}'\boldsymbol{x}} = \lambda_1 \qquad \text{(attained when } \boldsymbol{x} = \boldsymbol{e}_1\text{)}$$

$$\min_{\substack{\boldsymbol{x}\neq\boldsymbol{0} \\ \boldsymbol{x}'\boldsymbol{x}=1}} \frac{\boldsymbol{x}'\boldsymbol{B}\boldsymbol{x}}{\boldsymbol{x}'\boldsymbol{x}} = \lambda_p \qquad \text{(attained when } \boldsymbol{x} = \boldsymbol{e}_p\text{)}$$

Moreover,

$$\max_{\substack{\boldsymbol{x}\perp\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_k \\ \boldsymbol{x}'\boldsymbol{x}=1}} \frac{\boldsymbol{x}'\boldsymbol{B}\boldsymbol{x}}{\boldsymbol{x}'\boldsymbol{x}} = \lambda_{k+1} \qquad \text{(attained when } \boldsymbol{x} = \boldsymbol{e}_{k+1}\text{), k=1,2,\ldots,p-1}$$

Proof 4 The spectral decomposition of $\boldsymbol{W} = \boldsymbol{P}'\boldsymbol{\Lambda}\boldsymbol{P} = \boldsymbol{P}'\boldsymbol{\Lambda}^{1/2}\boldsymbol{\Lambda}^{1/2}\boldsymbol{P}$. Then

$$\boldsymbol{W}^{1/2} = \boldsymbol{P}'\boldsymbol{\Lambda}^{1/2}\boldsymbol{P}$$

Let $\boldsymbol{u} = \boldsymbol{W}^{1/2}\boldsymbol{a}$. The problem becomes

$$\max_{\boldsymbol{u}} \frac{\boldsymbol{u}'\boldsymbol{W}^{-1/2}\boldsymbol{B}\boldsymbol{W}^{-1/2}\boldsymbol{u}}{\boldsymbol{u}'\boldsymbol{u}}$$

Let λ_1 be the largest eigenvalue of $\boldsymbol{W}^{-1/2}\boldsymbol{B}\boldsymbol{W}^{-1/2}$ and associated eigenvector is $\boldsymbol{e}_1$. Then $\boldsymbol{u} = \boldsymbol{e}_1$ maximizes the ratio above. Thus

$$\boldsymbol{a} = \boldsymbol{W}^{-1/2}\boldsymbol{u} = \boldsymbol{W}^{-1/2}\boldsymbol{e}_1$$

Since $\boldsymbol{W}^{-1/2}\boldsymbol{B}\boldsymbol{W}^{-1/2}\boldsymbol{e} = \lambda\boldsymbol{e}$

$$\boldsymbol{W}^{-1/2}\boldsymbol{W}^{-1/2}\boldsymbol{B}\boldsymbol{W}^{-1/2}\boldsymbol{e} = \lambda\boldsymbol{W}^{-1/2}\boldsymbol{e}$$

Therefore λ is also the eigenvalue of $\boldsymbol{W}^{-1}\boldsymbol{B}$ and the corresponding eigenvector is $\boldsymbol{W}^{-1/2}\boldsymbol{e} = \boldsymbol{a}$.

Since $\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}, \ldots, \overline{\boldsymbol{x}}_g - \overline{\boldsymbol{x}}$ is in $q \leq g-1$ subspace of the p dimensional space, if $\boldsymbol{e}$ is orthogonal to any of $\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}$, then $\boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{e} = \boldsymbol{0} = 0\boldsymbol{e}$. Thus 0 is the eigenvalue of $\boldsymbol{W}^{-1}\boldsymbol{B}$ and there will be $p-q$ eigenvectors for the

0 eigenvalue. It implies there will be q or fewer nonzero eigenvalues. Therefore $s \le \min(p, g-1)$.

It is easy to see that the sample variance of the projects of y_{ij} onto $\boldsymbol{a}_1$ is

$$\frac{1}{n_1 + \cdots + n_g - g} \sum_{i=1}^{g} \sum_{j=1}^{n_i} (\boldsymbol{a}_1' \boldsymbol{x}_{ij} - \boldsymbol{a}_1' \overline{\boldsymbol{x}}_i)^2 = \boldsymbol{a}_1' \boldsymbol{S}_{pooled} \boldsymbol{a}_1 = 1$$

For $\boldsymbol{a}_2 = \boldsymbol{W}^{-1/2} \boldsymbol{e}_2$, the sample covariance between the projects of y_{ij} onto $\boldsymbol{a}_1$ and those onto $\boldsymbol{a}_2$ is

$$\frac{1}{n_1 + \cdots + n_g - g} \sum_{i=1}^{g} \sum_{j=1}^{n_i} (\boldsymbol{a}_1' \boldsymbol{x}_{ij} - \boldsymbol{a}_1' \overline{\boldsymbol{x}}_i)(\boldsymbol{a}_2' \boldsymbol{x}_{ij} - \boldsymbol{a}_2' \overline{\boldsymbol{x}}_i) = \boldsymbol{a}_1' \boldsymbol{S}_{pooled} \boldsymbol{a}_2 = 0$$

Generally

$$\boldsymbol{a}_i' \boldsymbol{S}_{pooled} \boldsymbol{a}_k = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases}$$

However, above is not true if $\boldsymbol{S}_{pooled}$ is replaced with $\boldsymbol{S}_i$.

Remark 5 *Because* $s \leq \min(g-1, p)$*, there is no loss of information for discriminantion by plotting in two dimensions if*

Number of variables	*Number of populations*	*Number of discriminants*
Any p	$g = 2$	$r = 1$
Any p	$g = 3$	$r = 2$
$p = 2$	*Any* g	$r = 2$

Question: How to construct classification rule based on $\boldsymbol{a}_i$?

Theorem 7 (Classification rules based on Fisher's discriminants) *Consider $r \le s$ discriminants $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_r$. Classify $\boldsymbol{x}$ to π_k if*

$$\sum_{j=1}^{r}[\boldsymbol{a}_j'(\boldsymbol{x}-\overline{\boldsymbol{x}}_k)]^2 \le \sum_{j=1}^{r}[\boldsymbol{a}_j'(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)]^2 \quad \textit{for all } i \ne k$$

Proof 5 Let $\boldsymbol{E} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_p)$ where $\boldsymbol{e}_i$ is the eigenvector of $\boldsymbol{W}^{-1/2}\boldsymbol{B}\boldsymbol{W}^{-1/2}$. The generalized square distance between $\boldsymbol{x}$ and $\overline{\boldsymbol{x}}_i$ is

$$(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)'\boldsymbol{S}_{pooled}^{-1}(\boldsymbol{x}-\overline{\boldsymbol{x}}_i) = (n-g)(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)'\boldsymbol{W}^{-1}(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)$$

$$= (n-g)(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)'\boldsymbol{W}^{-1/2}\boldsymbol{W}^{-1/2}(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)$$

$$= (n-g)(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)'\boldsymbol{W}^{-1/2}\boldsymbol{E}\boldsymbol{E}'\boldsymbol{W}^{-1/2}(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)$$

$$= (n-g)(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)'\left(\sum_{j=1}^{p}\boldsymbol{a}_j\boldsymbol{a}_j'\right)(\boldsymbol{x}-\overline{\boldsymbol{x}}_i) = (n-g)\sum_{j=1}^{p}\left[\boldsymbol{a}_j'(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)\right]^2$$

Therefore $\sum_{j=1}^{p}\left[\boldsymbol{a}_j'(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)\right]^2$ measures the generalized square distance between $\boldsymbol{x}$ and $\overline{\boldsymbol{x}}_i$. For those $\boldsymbol{a}_j = \boldsymbol{W}^{-1/2}\boldsymbol{e}_j$ where $\boldsymbol{e}_j$ is an eigenvector corresponding to the zero eigenvalue of $\boldsymbol{W}^{-1/2}\boldsymbol{B}\boldsymbol{W}^{-1/2}$,

$$\boldsymbol{e}_j \perp \overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}} \text{ and } \boldsymbol{e}_j \perp \overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}} \Longrightarrow \boldsymbol{e}_j \perp \overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}} - (\overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}}) = \overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}_k$$

$$\Longrightarrow \boldsymbol{e}_j'(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}_k) = 0 \Longrightarrow \boldsymbol{a}_j'(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}_k) = 0 \Longrightarrow \boldsymbol{a}_j'\overline{\boldsymbol{x}}_i = \boldsymbol{a}_j'\overline{\boldsymbol{x}}_k$$

Therefore the last $p - s$ summands

$$\sum_{j=s+1}^{p}\left[\boldsymbol{a}_j'(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)\right]^2 \text{ is a constant with respect to } i$$

Therefore we only consider

$$\sum_{j=1}^{r}\left[\boldsymbol{a}_j'(\boldsymbol{x}-\overline{\boldsymbol{x}}_i)\right]^2$$

for $r \le s$.

Remark 6 *When using the discriminant functions, subjects are classified to populations based on Euclidean distances.*

Question: What is the practical meaning of λ_j?

Consider the separation of the g populations in the direction of the jth discriminant $\boldsymbol{a}_j$, weighted by the sample sizes:

$$\sum_{i=1}^{g} n_i[\boldsymbol{a}_j'(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})]^2 = \sum_{i=1}^{g} n_i \boldsymbol{a}_j'(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})'\boldsymbol{a}_j$$

$$= \sum_{i=1}^{g} n_i \boldsymbol{e}_j \boldsymbol{W}^{-1/2}(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})'\boldsymbol{W}^{-1/2}\boldsymbol{e}_j = \boldsymbol{e}_j'\boldsymbol{W}^{-1/2}\boldsymbol{B}\boldsymbol{W}^{-1/2}\boldsymbol{e}_j = \lambda_j$$

Therefore λ_j measures the squared distances between population means and the overall mean after they are projected onto $\boldsymbol{a}_j$.

Further the overall separation of the g populations can be measured by

$$\Delta_S^2 = \sum_{i=1}^{g} n_i(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})' \boldsymbol{S}_{pooled}^{-1} (\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})$$

$$= (n-g) \sum_{i=1}^{g} n_i(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})' \boldsymbol{W}^{-1} (\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})$$

$$= (n-g) \sum_{i=1}^{g} n_i(\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})' \boldsymbol{W}^{-1/2} \boldsymbol{E}\boldsymbol{E}' \boldsymbol{W}^{-1/2} (\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})$$

$$= (n-g) \sum_{i=1}^{g} n_i \sum_{j=1}^{p} [\boldsymbol{a}_j' (\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})]^2$$

$$= (n-g) \sum_{j=1}^{p} \sum_{i=1}^{g} n_i [\boldsymbol{a}_j' (\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}})]^2 = (n-g) \sum_{j=1}^{p} \lambda_j$$

Therefore the first few eigenvalues contribute more to the separation of the populations than the last few eigenvalues.

Example 8 (Example 11.13) Compute by “hand”

Get overall mean

```
> zm <- colMeans(e11.10[, 1:2])
```

Get group means

```
> zmi <- by(e11.10[, 1:2], e11.10$group, colMeans)
```

Get $\boldsymbol{B}$

```
> zb <- matrix(apply(sapply(zmi, function(x, mu)3*(x-mu) %*% t(x-mu), zm), 1, sum), 2)
```

Get $\boldsymbol{W}$

```
> zw <- matrix(apply(sapply(by(e11.10[, 1:2], e11.10$group, function(x)2*var(x)), function(x)x), 1, sum), 2)
```

Find the eigen values of $\boldsymbol{W}^{-1}\boldsymbol{B}$:

```
> z <- eigen(solve(zw) %*% zb)

> z$values/sum(z$values)

[1] 0.7602082 0.2397918
```

Get the coefficient of the first discriminant:

```
> z$vectors[, 1]/sqrt(z$vectors[, 1] %*% (zw/6) %*% z$vectors[, 1])

[1] -0.3856092 -0.4945830
```

Get the coefficient of the second discriminant:

```
> z$vectors[, 2]/sqrt(z$vectors[, 2] %*% (zw/6) %*% z$vectors[, 2])
```

```
[1] -0.9380176  0.1119397
```

Discrimination analysis by `lda`:

```
> z <- lda(group ~ x1 +x2, data = e11.10)
> z

Call:
lda.formula(group ~ x1 + x2, data = e11.10)

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 
```

The output contains the two linear discriminant functions. They differ from those in the text only by signs. The proportion of trace provides proportions of eigenvalues:

$$\frac{\lambda_i}{\sum_{i=1}^{s} \lambda_i}$$

To allocate a new point $\boldsymbol{x}_0 = (1, 3)$:

```
> predict(z, newdata = data.frame(x1 = 1, x2 = 3))

$class:
[1] 2
$class: Levels:
[1] "1" "2" "3"
$posterior:
          1         2          3
1 0.1249667 0.8597303 0.01530298
$x:
        LD1       LD2
1 -1.045053 0.7887646
```

It contains the population number it is allocated and also the values of two discriminant functions (which are different from those values in the text for the same reason as above).

If only the first discriminant is used for classification:

```
> predict(z, newdata = data.frame(x1 = 1, x2 = 3), dimen = 1)

$class
[1] 2
Levels: 1 2 3
$posterior
         1         2          3
1 0.451675 0.5381006 0.01022434
$x
        LD1
1 -1.045053
```

The performance of the LDA method can be examined easily with `CV` argument of `lda` function.

Regression Methods for Discrimination

Multivariate linear regression method Define

$$Y_{ik} = \begin{cases} 1 & \text{if subject } i \text{ is in } \pi_k \\ 0 & \text{Otherwise} \end{cases}$$

We expand $\boldsymbol{X}_{n\times p}$ into $\boldsymbol{X}_{n\times(p+1)}$ to include a column of 1's for intercept terms. The multivariate regression model is

$$Y_{ik} = \boldsymbol{x}_i'\boldsymbol{\beta}_{(k)} + \boldsymbol{\epsilon}_{ik}, \quad i = 1, \ldots, n, \quad k = 1, \ldots, g$$

We fit a multivariate linear regression model to describe the relationship between the response variable $\boldsymbol{Y}_{n\times g}$ and $\boldsymbol{X}_{n\times p}$.

$$\boldsymbol{Y}_{n\times g} = \boldsymbol{X}_{n\times(p+1)}\boldsymbol{b}_{(p+1)\times g} + \boldsymbol{\epsilon}_{n\times g}$$

The least square estimate of $\boldsymbol{\beta}$

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$$

Or for population k, the LS estimate is

$$\widehat{\boldsymbol{\beta}}_{(k)} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}_{(k)}$$

Classification rule: a given subject with $\boldsymbol{x}_0$ is classified into population k if

$$(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_g) = \widehat{\boldsymbol{y}}_{1\times g} = \boldsymbol{x}_0'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$$

and

$$\hat{y}_k \geq \hat{y}_i, \quad i = 1, \ldots, g, i \neq k$$

because

$$\hat{y}_k = E(Y_k|\boldsymbol{X} = \boldsymbol{x}) = P(Y_k = 1|\boldsymbol{X})$$

Remark 7 • It can be verified that $\sum_{i=1}^{g} \hat{y}_i = 1$ (for example, check it with $p = 1$). That is, the subject will be classified into one of the g populations. However, $\hat{y}_i$ may bot always be between 0 and 1.

- The linear regression model may not work well because of the rigid nature of linear regression models. Some populations may be completely missed/masked, leading to large AER.

Example 9 (Multivariate linear regression for discrimination analysis) *We run*

```
> z <- lm(cbind(e11.10$group==1, e11.10$group==2, e11.10$group==3) ~ x1 + x2, data = e11.10)
> z

Call:
lm(formula = cbind(e11.10$group == 1, e11.10$group == 2, e11.10$group == 3)
~ x1 + x2, data = e11.10)

Coefficients:
             [,1]      [,2]      [,3]
(Intercept)   0.25089   0.20239   0.54672
x1           -0.25412   0.24345   0.01067
x2            0.04947   0.07856  -0.12803
```

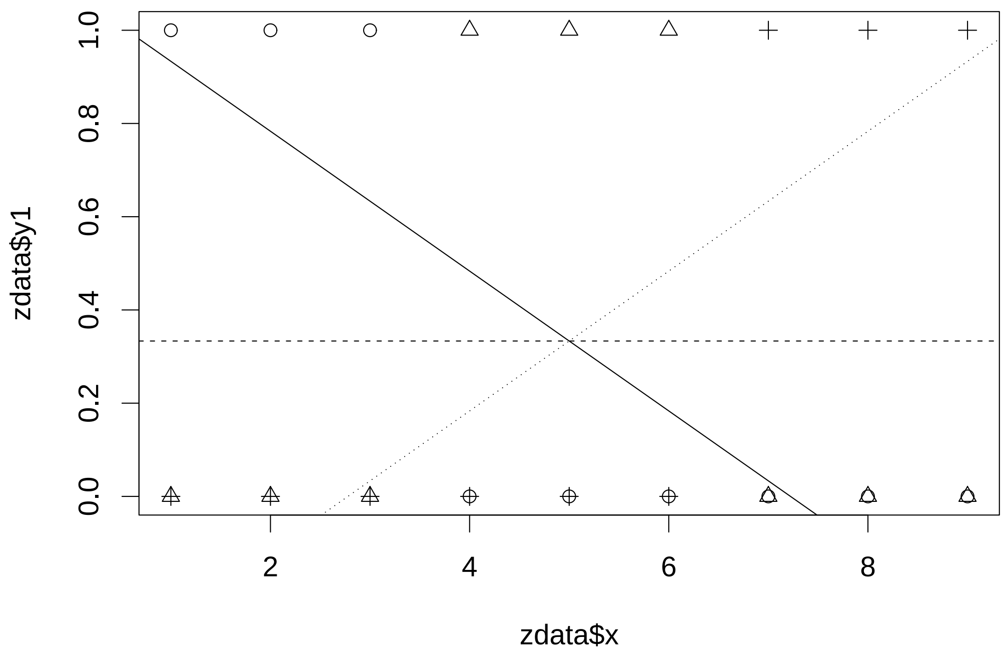
```
> predict(z, newdata = data.frame(x1 = 1, x2 = 3))

       [,1]     [,2]      [,3]
1 0.1451665 0.681539 0.1732945

> predict(z, newdata = data.frame(x1 = 1, x2 = 5))

       [,1]     [,2]        [,3]
1 0.2440996 0.838668 -0.08276754
```

Example 10 (Example of masking) *We run*

```
> zdata <- data.frame(x = 1:9, y1 = c(1, 1, 1, rep(0, 6)),
  y2 = c(0, 0, 0, 1, 1, 1, 0, 0, 0), y3 = c(rep(0, 6), 1, 1, 1))
> zdata

  x y1 y2 y3
1 1  1  0  0
2 2  1  0  0
3 3  1  0  0
4 4  0  1  0
5 5  0  1  0
6 6  0  1  0
7 7  0  0  1
8 8  0  0  1
9 9  0  0  1

> plot(zdata$x, zdata$y1)
> points(zdata$x, zdata$y2, pch = 2)
> points(zdata$x, zdata$y3, pch = 3)
> z <- lm(cbind(y1, y2, y3) ~ x, data = zdata)
> coef(z)
```

```
                    y1           y2         y3
(Intercept)  1.083333 3.333333e-01 -0.4166667
x           -0.150000 3.124828e-18  0.1500000
```

```
> abline(a = coef(z)[1, 1], b = coef(z)[2, 1])
> abline(a = coef(z)[1, 2], b = coef(z)[2, 2], lty = 2)
> abline(a = coef(z)[1, 3], b = coef(z)[2, 3], lty = 3)
```



Logistic regression If $g = 2$, the group/population indicator is a binary variable and a logistic model can be fit to the data:

$$\log\left(\frac{P(Y_{i2}=1|\boldsymbol{x}_i)}{P(Y_{i1}=1|\boldsymbol{x}_i)}\right) = \log\left(\frac{P(Y_{i2}=1|\boldsymbol{x}_i)}{1-P(Y_{i2}=1|\boldsymbol{x}_i)}\right) = \text{logit}[P(Y_{i2}=1|\boldsymbol{x}_i)] = \boldsymbol{x}_i'\boldsymbol{\beta}$$

Under this model

$$P(Y_{i1} = 1|\boldsymbol{x}_i) = \frac{1}{1 + \exp(\boldsymbol{x}_i'\boldsymbol{\beta})}$$

$$P(Y_{i2} = 1|\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{x}_i'\boldsymbol{\beta})}{1 + \exp(\boldsymbol{x}_i'\boldsymbol{\beta})}$$

Example 11 (Logistic regression) *We run*

```
> z <- glm((group=="Canadian") ~ freshwater + marine, data = t11.2, family = binomial())
> z

Call:  glm(formula = (group == "Canadian") ~ freshwater + marine,
family = binomial(), data = t11.2)

Coefficients:
(Intercept)   freshwater       marine
    3.92484      0.12605     -0.04854

Degrees of Freedom: 99 Total (i.e. Null);  97 Residual
Null Deviance:     138.6
Residual Deviance: 38.79  AIC: 44.79

> predict(z, newdata = data.frame(freshwater = 100, marine = 400), type = "response")

[1] 0.05276334
```

Multinomial regression The multinomial log-linear model is an extension of the logistic model for $g > 2$. The model is given as follows

$$\log\left(\frac{P(Y_{ik} = 1|\boldsymbol{x}_i)}{P(Y_{i1} = 1|\boldsymbol{x}_i)}\right) = \boldsymbol{x}_i'\boldsymbol{\beta}_{(k)}, \quad i = 1, \ldots, n, \quad k = 2, \ldots, g$$

or

$$P(Y_{ik} = 1|\boldsymbol{x}_i) = \exp(\boldsymbol{x}_i'\boldsymbol{\beta}_{(k)})P(Y_{i1} = 1|\boldsymbol{x}_i)$$

It can be shown easily that

$$P(Y_{i1} = 1|\boldsymbol{x}_i) = \frac{1}{1 + \sum_{j=2}^{g} \exp(\boldsymbol{x}_i'\boldsymbol{\beta}_{(j)})}$$

$$P(Y_{ik} = 1|\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{x}_i'\boldsymbol{\beta}_{(k)})}{1 + \sum_{j=2}^{g} \exp(\boldsymbol{x}_i'\boldsymbol{\beta}_{(j)})}, \quad k = 2, \ldots, g$$

Remark 8 In the classical classification function under the multivariate normal distributiona with equal covariances, we also obtain a log-linear model:

$$\log\left(\frac{P(Y_{ik}=1|\boldsymbol{x}_i)}{P(Y_{i1}=1|\boldsymbol{x}_i)}\right) = \log\left(\frac{f_k(\boldsymbol{x}_i)p_k}{f_1(\boldsymbol{x}_i)p_1}\right) = \log\frac{f_k(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)} + \log\frac{p_k}{p_1}$$

$$= \log\frac{p_k}{p_1} - \frac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_1)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_1) + \boldsymbol{x}_i'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_1)$$

$$= \log\frac{p_k}{p_1} - \frac{1}{2}(\overline{\boldsymbol{x}}_k + \overline{\boldsymbol{x}}_1)'\boldsymbol{S}_{pooled}^{-1}(\overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}}_1) + \boldsymbol{x}_i'\boldsymbol{S}_{pooled}^{-1}(\overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}}_1)$$

There are some important differences in parameter estimation in the two approaches:

- The classical discrimination method is a full parametric method and it depends on the marginal distribution of $\boldsymbol{x}$, which is a mixture distribution

$$\sum_{k=1}^{g} p_k f_k(\boldsymbol{x}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma})$$

- Logistic/multinomial regression model ignores the marginal distribution of $\boldsymbol{x}$ and estimates the parameters based on the conditional likelihood — the multinomial likelihood with $P(Y_{ik} = 1|\boldsymbol{x})$.

- If the multivariate normal assumption is true, the logistic/multinomial method may lose efficiency (about 30% in a worse case, or 30% more data to do as well). The LDA method based on multivariate normal assumption can also use the information about marginal distribution from a subject without a class label.

- Logistic/multinomial method is more robust to outliers or deviation from the multivariate normal assumption. It is safer to use when the normality assumption is a question.

Example 12 (Multinomial regression) Fitting a multinomial log-linear model using `multinom` in the library `nnet`:

```
> library(nnet)
> z <- multinom(group ~ x1 + x2, data = e11.10)
> z

Call:
```

```
multinom(formula = group ~ x1 + x2, data = e11.10)

Coefficients:
  (Intercept)        x1         x2
2   -31.69858 28.368659   6.978336
3    11.97077  9.701029 -19.411615

Residual Deviance: 0.0001800467
AIC: 12.00018

> predict(z, newdata = data.frame(x1 = 1, x2 = 5))

[1] 2
Levels: 1 2 3

> predict(z, newdata = data.frame(x1 = -2, x2 = 1))

[1] 1
Levels: 1 2 3

> predict(z, newdata = data.frame(x1 = -2, x2 = 1), type = "probs")

           1            2            3
1.000000e+00 4.201730e-36 2.199249e-12
```

A summary Comparing to the classical LDA methods introduced before:

- Pros:

  - easy to fit
  - may easily accommodate different types of variables, such as qualitattive variables
  - model diagnostic methods available
- Cons: may not work well in some situations.

k-Nearest Neighbor Classification

This is a complete nonparametric method.

Classify a subject with $\boldsymbol{x}_0$ according to the following steps:

- Locate k training points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$ closest in distance to $\boldsymbol{x}_0$
- Classify $\boldsymbol{x}_0$ using majority vote among the k neighbors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$.

Some technical notes:

- k Neighbors are determined using the distance $d(\boldsymbol{x}_i, \boldsymbol{x}_0)$. The distance can be
  - Euclidean distance: $d(\boldsymbol{x}_i, \boldsymbol{x}_0) = \sqrt{(\boldsymbol{x} - \boldsymbol{x}_0)'(\boldsymbol{x} - \boldsymbol{x}_0)}$
  - absolute distance: $d(\boldsymbol{x}_i, \boldsymbol{x}_0) = |\boldsymbol{x} - \boldsymbol{x}_0|'\mathbf{1}$ (city-block distance)
- Ties are broken at random.
  - The number majority votes are the same in at least two populations

- The members of k-nearest neighbors

- Variables may be standardized to have mean zero and variance 1.
- For $k = 1$, one gets the simple nearest neighbor method with maximal local technique
- For $k \to n$ a global majority vote of the whole training set results: That is, always classify a new subject into the most frequent population.

Pros and cons:

- Simple, low bias
- Large variation

The classifier can be evaluated using the jackknife procedure.

The k-nearest neighbor classification based on Euclidean distances can be done in R with library `class`

Example 13 (Classification with k-nearest neighbors) *We run*

```
> library(class)
> knn(t11.2[, c("freshwater", "marine")], data.frame(freshwater = 100, marine = 400), t11.2$group, k = 1)

[1] Alaskan
Levels: Alaskan Canadian

> knn(t11.2[, c("freshwater", "marine")], data.frame(freshwater = 100, marine = 400), t11.2$group, k = 3)

[1] Alaskan
Levels: Alaskan Canadian

> knn(t11.2[, c("freshwater", "marine")], data.frame(freshwater = 100, marine = 400), t11.2$group, k = 9)

[1] Alaskan
Levels: Alaskan Canadian
```

To evaluate AER of this classifier with jackknife procedure:

```
> table(knn.cv(t11.2[, c("freshwater", "marine")], t11.2$group, k = 9), t11.2$group)

           Alaskan Canadian
  Alaskan       46        3
  Canadian       4       47
```

This result is quite compariable with that from `lda`

Final Words About Discrimination Analysis

Other techiques

- CART: Classification and regression trees
- Neural networks
- Bayesian belief networks
- Projection pursuit

Variable selection

- Number of variables
- Which variables
- Variable transformation, linear or non-linear (for example, instead of quadratic discrimination analysis based on X_1 and X_2, you‘ may do

linear discrimination analysis based on $X_1, X_2, X_1X_2, X_1^2, X_2^2$).

Example 14 (Iris data) *We run*

```
> t11.5 <- read.table("T11-5.DAT", header = F, col.names = c("SL", "SW", "PL", "PW", "group"))
> t11.5$group <- factor(t11.5$group, labels = c("setosa", "versicolor", "virginica"))
> z <- lda(group ~ SL + SW + PL + PW, data = t11.5, CV = T)
> table(t11.5$group, z$class)

             setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          1        49
```

The error rate is $3/150 = .02$. If we use a single variable PW:

```
> z <- lda(group ~ PW, data = t11.5, CV = T)
> table(t11.5$group, z$class)

             setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          4        46
```

The error rate is $6/150 = .04$.

```
> z <- princomp(t11.5[, 1:4])
> z

Call:
princomp(x = t11.5[, 1:4])
```

```
Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4
2.0494032 0.4909714 0.2787259 0.1538707
 4  variables and  150 observations.
```

```
> zz <- lda(group ~ Comp.1, data = data.frame(predict(z),
        group = t11.5$group), CV = T)
> table(t11.5$group, zz$class)

            setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         46         4
  virginica       0          6        44

> zz <- lda(group ~ Comp.1 + Comp.2, data = data.frame(predict(z),
        group = t11.5$group), CV = T)
> table(t11.5$group, zz$class)

            setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         47         3
  virginica       0          4        46

> zz <- lda(group ~ Comp.1 + Comp.2 + Comp.3, data = data.frame(predict(z),
        group = t11.5$group), CV = T)
> table(t11.5$group, zz$class)

            setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          0        50
```

The AER is $2/150 = 0.013$.