Discriminant Analysis

Motivating examples:

1): Octopuses eaten by ling cod. Octopus beaks not digested; remain in cod stomach. Idea: make measurements on beaks to determine sex of eaten octopus — to determine where the different sexes live.

2): Digitized scan of handwritten number: 32×32 grid of pixels each either black or white. From vector of 256 numbers determine what number was written.

Goal: given observation $\mathbf{X}$ classify $\mathbf{X}$ as coming from 1 of several possible populations.

Two versions:

A) Clustering: you observe only $\mathbf{X}_1, \ldots, \mathbf{X}_n$ some (presumably) from each of several populations. You try to describe k populations such that each $\mathbf{X}_i$ appears to come from one of them.

(Take spectral measurements on many stars. Break measurements apart into some number of groups (to be determined from data), develope rule to classify new observation into group, estimate proportion of observations in each group. Eventually: develop subject matter theory explaining nature of groups.

B) Discrimination / classification: have training data. You are given samples $\mathbf{X}_{ij}, j = 1, \ldots, n_i$ for $i = 1, \ldots, k$ with k known. You develop a rule to classify a new observation into one of these k groups.

Octopus example: begin with beaks from octopuses of known sex.

Character recognition: begin with digits where you know what it was supposed to be.

Simplest case first. You don't need the training data because: new $\mathbf{X}$ comes from 1 of 2 possible densities: f_1 or f_2. (No unknown parameters; assume that f_1 and f_2 are known.)

Begin with Bayesian approach. Suppose *a priori* probability that $\mathbf{X}$ comes from f_1 is π.

Notation: A_i is event that $\mathbf{X}$ comes from population i.

Compute the posterior probability:

$$P(A_1|\mathbf{X})$$

Apply Bayes' theorem:

$$\begin{aligned} P(A_1|\mathbf{X}=x) &= \frac{P(\mathbf{X}=x, A_1)}{P(\mathbf{X}=x)} \\ &= \frac{P(\mathbf{X}=x|A_1)P(A_1)}{P(\mathbf{X}=x|A_1)P(A_1)+P(\mathbf{X}=x|A_2)P(A_2)} \\ &= \frac{f_1(x)\pi_1}{f_1(x)\pi_1+f_2(x)\pi_2} \end{aligned}$$

General case of k populations:

$$P(A_i|\mathbf{X} = x) = \frac{f_i(x)\pi_i}{\sum_j f_j(x)\pi_j}$$

Now: how would a Bayesian classify $\mathbf{X}$?

Answer: depends on consequences and costs.

Let C_{ij} be cost of classifying observation actually from population i as being from population j.

Possible procedure: corresponds to partion of sample space into events $B_i, i = 1, \ldots, k$. If $\mathbf{X} \in B_i$ classify into population i.

Find rule to minimize expected cost; minimize

$$\sum_{ij} \pi_i C_{ij} P_i(\mathbf{X} \in B_j)$$

Simplest case: $k = 2$. Take $C_{11} = C_{22} = 0$; no cost if you get it right.

Must minimize

$$\pi_1 C_{12} P_1(\mathbf{X} \in B_2) + (1 - \pi_1) C_{21} P_2(\mathbf{X} \in B_1)$$

Write as integral:

$$\int \Big[\pi_1 C_{12} f_1(x) 1_{B_2}(x) + \pi_2 C_{21} f_2(x) \Big\{1 - 1_{B_2}(x)\Big\}\Big] \, dx$$

Notice that we have two choices for each x: either put x into B_2 and get $\pi_1 C_{12} f_1(x)$ or put it in B_1 and get $\pi_2 C_{21} f_2(x)$. To minimize the integral put x in the one which has the smaller corresponding value.

That is:

$$B_1 = \{x | \pi_1 C_{12} f_1(x) > \pi_2 C_{21} f_2(x)\}.$$

For all errors equally bad:

$$B_1 = \{x | \frac{f_1(x)}{f_2(x)} > \frac{\pi_2}{\pi_1}\}.$$

Gereral k:

$$B_i = \{x | \pi_i f_i(x) = \max\{\pi_j f_j(x)\}\}$$

Example: Two normal populations:

$$f_i(x) = (2\pi)^{-p/2} \det \mathbf{\Sigma}_i^{-1/2} \exp\{-(x - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1} (x - \boldsymbol{\mu}_i)/2\}$$

Likelihood ratio is function of

$$(x - \boldsymbol{\mu}_1)^T \mathbf{\Sigma}_1^{-1} (x - \boldsymbol{\mu}_1)/2 - (x - \boldsymbol{\mu}_2)^T \mathbf{\Sigma}_2^{-1} (x - \boldsymbol{\mu}_2)/2$$

So: boundaries of B_i determined by solution of quadratic.

Jargon: **Mahalanobis** distance is

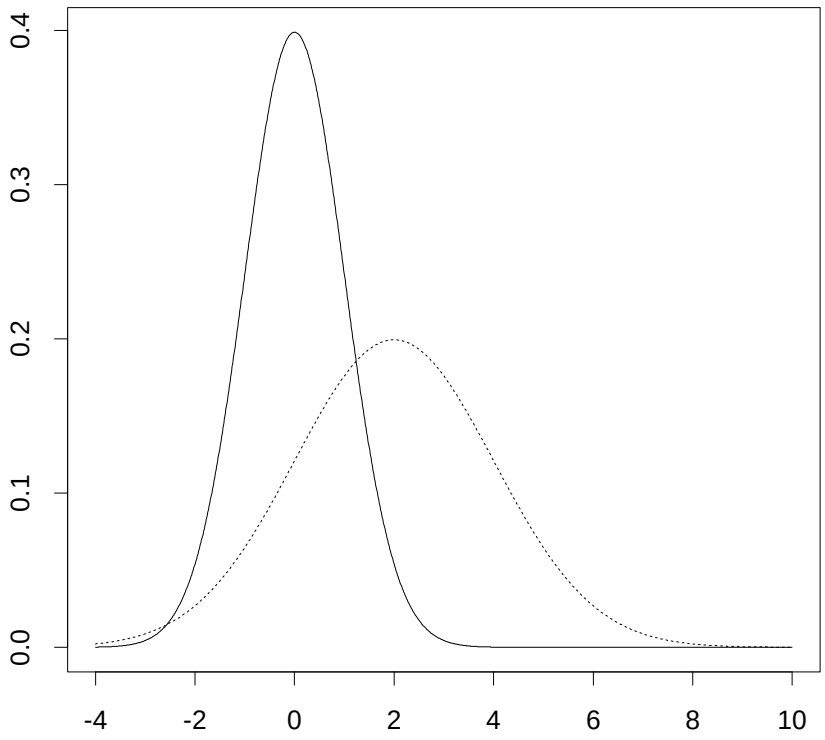
$$(x - \boldsymbol{\mu}_1)^T \mathbf{\Sigma}_1^{-1} (x - \boldsymbol{\mu}_1)$$

(from x to $\boldsymbol{\mu}_1$.)

One dimension:

$$x^2(\sigma_1^{-2} - \sigma_2^{-2}) - 2x(\mu_1/\sigma_1^2 - \mu_2/\sigma_2^2) = c$$

If $\sigma_1 < \sigma_2$ then B_1 is an interval and B_2 is rest.



Two dimensions some examples:

1) Both mean vectors 0. $\mathbf{\Sigma}_1 = \mathbf{I}$. $\mathbf{\Sigma}_2$ diagonal with entries 2 and 1/2.

Boundaries are now

$$x_1^2 - x_2^2/2 = c$$

Gives two curves. Special case $c = 0$ produces each region in form of two cones.

2) Same variances as in previous but now $\boldsymbol{\mu}_1 = (a, 0)$

Get

$$x_1^2 - 2ax_1 = x_2^2 + c$$

or

$$x_1 = a \pm \sqrt{x_2^2 + a^2 + c}$$

Usual rule: π_i taken all equal and all costs $C_{ij} = 1$ for $i \neq j$. So:

$$B_i = \{x | f_i(x) > f_j(x) \forall j \neq i\}$$

Minimizing total number of misclassifications.

Case of $k = 2$:

Notation:

$$\mathbf{D}_i = \mathbf{\Sigma}_i^{-1}$$
$$\ell = -2 \log \left\{ f_1(x) / f_2(x) \right\}$$

Difference in Mahalanobis distances becomes

$$x^T \mathbf{Q} x - 2x^T (\mathbf{D}_1 \boldsymbol{\mu}_1 - \mathbf{D}_2 \boldsymbol{\mu}_2) + \boldsymbol{\mu}_1^T \mathbf{D}_1 \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T \mathbf{D}_2 \boldsymbol{\mu}_2$$

where $\mathbf{Q} = \mathbf{D}_1 - \mathbf{D}_2$.

Rewrite as

$$(x-\boldsymbol{\theta})^T\mathbf{Q}(x-\boldsymbol{\theta})+k$$

where

$$\boldsymbol{\theta}=\mathbf{Q}^{-1}(\mathbf{D}_1\boldsymbol{\mu}_1-\mathbf{D}_2\boldsymbol{\mu}_2)$$

and

$$k=\boldsymbol{\theta}^T\mathbf{Q}\boldsymbol{\theta}+\boldsymbol{\mu}_1^T\mathbf{D}_1\boldsymbol{\mu}_1-\boldsymbol{\mu}_2^T\mathbf{D}_2\boldsymbol{\mu}_2$$

So boundaries have form

$$(x-\boldsymbol{\theta})^T\mathbf{Q}(x-\boldsymbol{\theta})=c-k$$

Shape depends on eigenvalues of $\mathbf{Q}$:

Both positive: ellipse

Both negative: ellipse

Opposite signs: hyperbola

Equal Covariances

When $\Sigma_1 = \Sigma_2$ the quadratic terms cancel and we get

$$\ell = (x-\boldsymbol{\mu}_1)^T\Sigma^{-1}(x-\boldsymbol{\mu}_1)-(x-\boldsymbol{\mu}_2)^T\Sigma^{-1}(x-\boldsymbol{\mu}_2)$$

which simplifies to

$$2x^T\Sigma^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)+\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1-\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2$$

This means the boundary between B_1 and B_2 is a hyperplane.

Misclassification rates: let P_i denote probability under f_i.

$$P_i(\text{assign X to } j) = \int_{B_j} f_i(x)dx$$

Do case $k = 2$ and $\Sigma_1 = \Sigma_2 = \Sigma$.

$$P_1(A_2) = P_1(\ell < 0)$$

But if X is from population 1 then

$$\ell \sim N(\delta^2, \sigma^2)$$

where

$$\begin{aligned}\delta^2 &= 2\boldsymbol{\mu}_1^T\Sigma^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 \\ &= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\end{aligned}$$

(squared Mahalanobis dist between means) and

$$\sigma^2 = 4(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T\Sigma^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) = 4\delta^2.$$

So:

$$P(\ell > 0) = P(N(0,1) < -\delta^2/\sigma) = \Phi(-\delta/2)$$

With data:

Framework: given training samples $\mathbf{X}_{ij}, j = 1 \ldots, n_i$ for $i = 1, \ldots, k$.

Replace $\boldsymbol{\Sigma}_i$ in all formulas by $\mathbf{S}_i$.

Replace $\boldsymbol{\mu}_i$ in all formulas by $\bar{\mathbf{X}}_i$.

If you conclude $\boldsymbol{\Sigma}_i$ all equal:

Estimate using

$$\mathbf{S} = \frac{\sum(n_i - 1)\mathbf{S}_i}{\sum(n_i - 1)}$$

Use Linear Discrimant analysis. Assign new $\mathbf{X}$ to group i which minimizes

$$(\mathbf{X} - \bar{\mathbf{X}}_i)^T \mathbf{S}^{-1} (\mathbf{X} - \bar{\mathbf{X}}_i)$$

For $k = 2$ rule is equivalent to assign to group 1 if

$$\mathbf{X}^T \mathbf{S}^{-1} (\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2) > \frac{1}{2} (\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2)^T \mathbf{S}^{-1} (\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2)$$

Notice linear boundaries

Drawing contours in discriminant analysis

Use Fisher's iris data.

Do quadratic discriminant analysis.

Use first two variables in iris data to do contouring.

Quadratic discriminant rule assigns x to that group i for which

$$(x - \bar{x}_i)^T S_i^{-1} (x - \bar{x}_i) + \log(|S_i|)$$

is minimized. I drew the contours as follows:

1. Make a grid of (x, y) pairs over the region where I wanted to do the plotting:

```
nd <- 500
xvals <- seq(2.8,8,length=nd)
yvals <- seq(1.8,4.5,length=nd)
xv <- rep(xvals,nd)
yv <- rep(yvals,rep(nd,nd))
```

2. Compute the means, covariances and logarithms of the determinants of the covariances for the three groups of data:

```
e1 <- var(diris[1:50,1:2])
e2 <- var(diris[51:100,1:2])
e3 <- var(diris[101:150,1:2])
c1 <- apply(diris[1:50,1:2],2,mean)
c2 <- apply(diris[51:100,1:2],2,mean)
c3 <- apply(diris[101:150,1:2],2,mean)
ld1<- log(prod(eigen(e1,
                symmetric=T)$values))
ld2<- log(prod(eigen(e2,
                symmetric=T)$values))
ld3<- log(prod(eigen(e3,
                symmetric=T)$values))
```

3. For each point on the grid compute the distance (adjusted by the log determinant) to each group center.

```
d1<-mahalanobis(cbind(xv,yv),c1,e1) +ld1
d2<- mahalanobis(cbind(xv,yv),c2,e2)+ld2
d3<- mahalanobis(cbind(xv,yv),c3,e3)+ld3
```

4. Build a matrix whose entry at i, j is the group to which (x_i, y_j) would be classified:

```
z<-rep(1,length(xv))
z[d2<d1] <- 2
z[d3<pmin(d1,d2)] <- 3
z <- matrix(z,nd,nd)
```

5. Use the contour plotting function to draw level curves between the levels 1, 2 and 3, and superimpose the points.

```
contour(xvals,yvals,z,
   levels=c(1.5,2.5),labex=0)
points(diris[1:50,1:2],pch=1)
points(diris[51:100,1:2],pch=2)
points(diris[101:150,1:2],pch=3)
```

Estimating Error Rates

Several ways if an estimated discriminant rule used.

Apparent Error Rate: Use the fitted rule to classify each observation in the training data set. Estimate

$$P_i(\text{Classify as } j)$$

using the number of i observations classified as j.

Severe underestimation likely.

Parametric Estimate of Error Rate: Compute the theoretical error rates of the known parameter classifiers; plug in estimates.

Underestimation likely.

Cross validation: Delete 1 observation from the data set. Estimate the parameters in the discriminant rule using all other data points. Classify the deleted observation. Use fraction of observations from population i classified as j to estimate error rates.

Forensic Glass Data

Six types of glass: Window Float, Window Non-float, Vehicle, Container Tableware, Vehicle Head Lamp

Eight Response variables: all percentages by weight of oxides of: Sodium, Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium and Iron.

Use Splus to do linear discriminant analysis:

```
glass <- read.table("fglass.dat",
                    header = T)
glass$type <- as.factor(glass$type)
```

Notice creation of class variable.

```
fit <- lm(as.matrix(glass[, 1:9]) ~ type)
r <- residuals(fit)
V <- 213*var(r)/208
FV <- fitted(fit)
Means <- FV[c(1, 71, 147, 164, 177, 186),  ]
for(i in 1:214) {
 for(j in 1:6)
  Distances[i,j]<-
     mahalanobis(glass[i,1:9],
        Means[j,  ], V)
}
```

Notice Splus version of glm.

```
> summary(fit)
Response: RI
            Df Sum ofSq   Mean Sq     F    Pr(F)
     type    5 0.000073 1.46295e-05 1.609 0.1590
Residuals 208 0.001891 9.09257e-06

Response: Na
            Df Sum of Sq  Mean Sq  F Value Pr(F)
     type    5  57.80464 11.56093 28.54802     0
Residuals 208  84.23257  0.40496

Response: Mg
            Df Sum of Sq  Mean Sq  F Value Pr(F)
     type    5  271.0954 54.21908 65.54452     0
Residuals 208  172.0597  0.82721

Response: Al
            Df Sum of Sq  Mean Sq  F Value Pr(F)
     type    5  24.53088 4.906177 35.72668     0
Residuals 208  28.56366 0.137325

Response: Si
            Df SumofSq MeanSq   F    Pr(F)
     type    5   8.024 1.6048 2.787 0.0185
Residuals 208 119.759 0.5758

Response: K
            Df SumofSq MeanSq   F    Pr(F)
     type    5  15.742 3.1484 8.748 1.507e-07
Residuals 208  74.858 0.3599
```

```
Response: Ca
            Df SumofSq MeanSq  F    Pr(F)
     type    5   28.76 5.7520 2.97 0.0130
Residuals 208  402.64 1.9358

Response: Ba
            Df Sum of Sq  Mean Sq F Value Pr(F)
     type    5  25.47176 5.094353 38.9746     0
Residuals 208  27.18759 0.130710

Response: Fe
            Df SumofSq  Mean Sq  F    Pr(F)
     type    5  0.1237 0.024744 2.71 0.0214
Residuals 208  1.8986 0.009128
```

Variable RI (Refractive Index) could well be dropped.

Now classify training data

```
Classes <- rep(1,214)
for(i in 1:214){
 jmin <- 1
 for(j in 2:6){
  if(Distances[i,j]
      < Distances[i,jmin])
        jmin <- j
}
Classes[i]<- jmin
Classmat <- matrix(0,7,7)
for( i in 1:214){
 Classmat[glass$type[i],Classes[i]] <-
 Classmat[glass$type[i],Classes[i]]+1}
```

```
 Classmat
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]   46   14   10    0    0    0    0
[2,]   16   41   12    4    3    0    0
[3,]    3    3   11    0    0    0    0
[4,]    0    0    0    0    0    0    0
[5,]    0    2    0   10    0    1    0
[6,]    1    1    0    0    7    0    0
[7,]    0    1    1    2    1   24    0
 Classmat <- Classmat[-4,-7]
 Classmat
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]   46   14   10    0    0    0
[2,]   16   41   12    4    3    0
[3,]    3    3   11    0    0    0
[4,]    0    2    0   10    0    1
[5,]    1    1    0    0    7    0
[6,]    0    1    1    2    1   24
```

The error rates for linear discriminant analysis are quite high.

Next do crossvalidation

```
CVClass <- rep(0,214)
for( i in 1:214){
 d <- as.matrix(glass[-i,1:9])
 tp <- as.numeric(glass[-i,10])
 mn <- matrix(0,7,9)
 for( j in c(1,2,3,5,6,7)){
 mn[j,] <- apply(d[tp==j,],2,mean)}
 mn <- mn[-4,]
 V <- var(residuals(lm(d~as.factor(tp))))
 V <- solve(V)
 dmin <- mahalanobis(glass[i,1:9],
              mn[1,],V,inverted=T)
 jmin <- 1
 for(j in 2:6)  {
  ds <- mahalanobis(glass[i,1:9],
           mn[j,],V,inverted=T)
  if( ds < dmin) { jmin<- j
      dmin <- ds
  }
 }
 CVClass[i] <- jmin
}
```

```
CVmat
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]   45   14   11    0    0    0
[2,]   17   37   12    6    3    1
[3,]    5    3    9    0    0    0
[4,]    0    0    0    0    0    0
[5,]    0    5    1    6    0    1
[6,]    1    1    0    0    6    1
[7,]    0    1    1    2    1   24
> CVmat <- CVmat[-4,]
> CVmat-Classmat
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]   -1    0    1    0    0    0
[2,]    1   -4    0    2    0    1
[3,]    2    0   -2    0    0    0
[4,]    0    3    1   -4    0    0
[5,]    0    0    0    0   -1    1
[6,]    0    0    0    0    0    0
```

Notice increased error rate using CV

Other methods:

Classification and Regression Trees

Neural Networks

Logistic Regression

Classification Tree Example

Use S function `tree` to classify the iris data.

First made variables from built-in data `iris`.

Here is my S code:

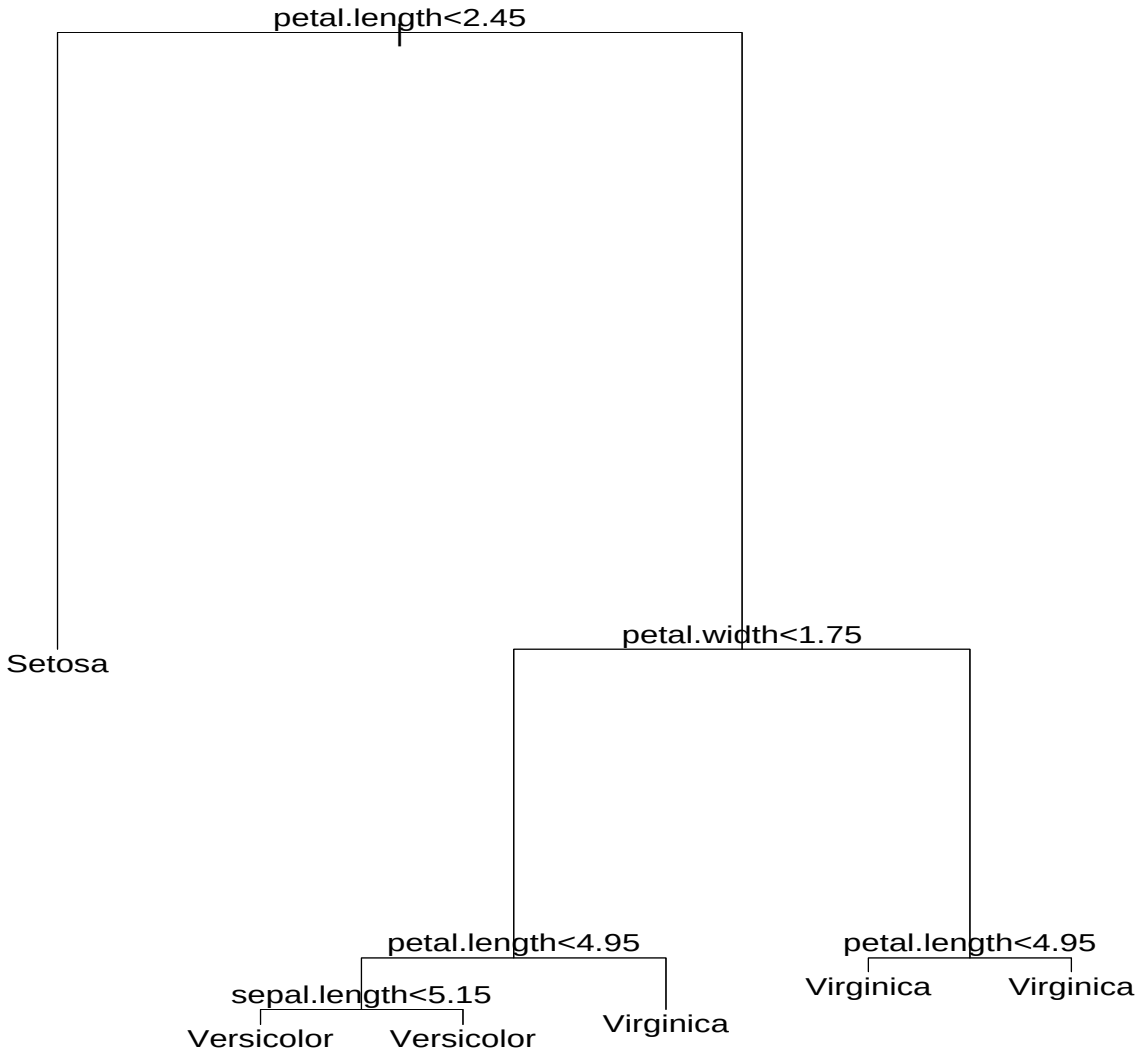
```
postscript("iristree.ps",horizontal=F)
variety <- as.category(c(rep("Setosa", 50),
 rep("Versicolor",50),rep("Virginica",50)))
iris.tree <- tree(variety ~ sepal.length +
 sepal.width + petal.length + petal.width)
plot(iris.tree)
text(iris.tree)
summary(iris.tree)
```

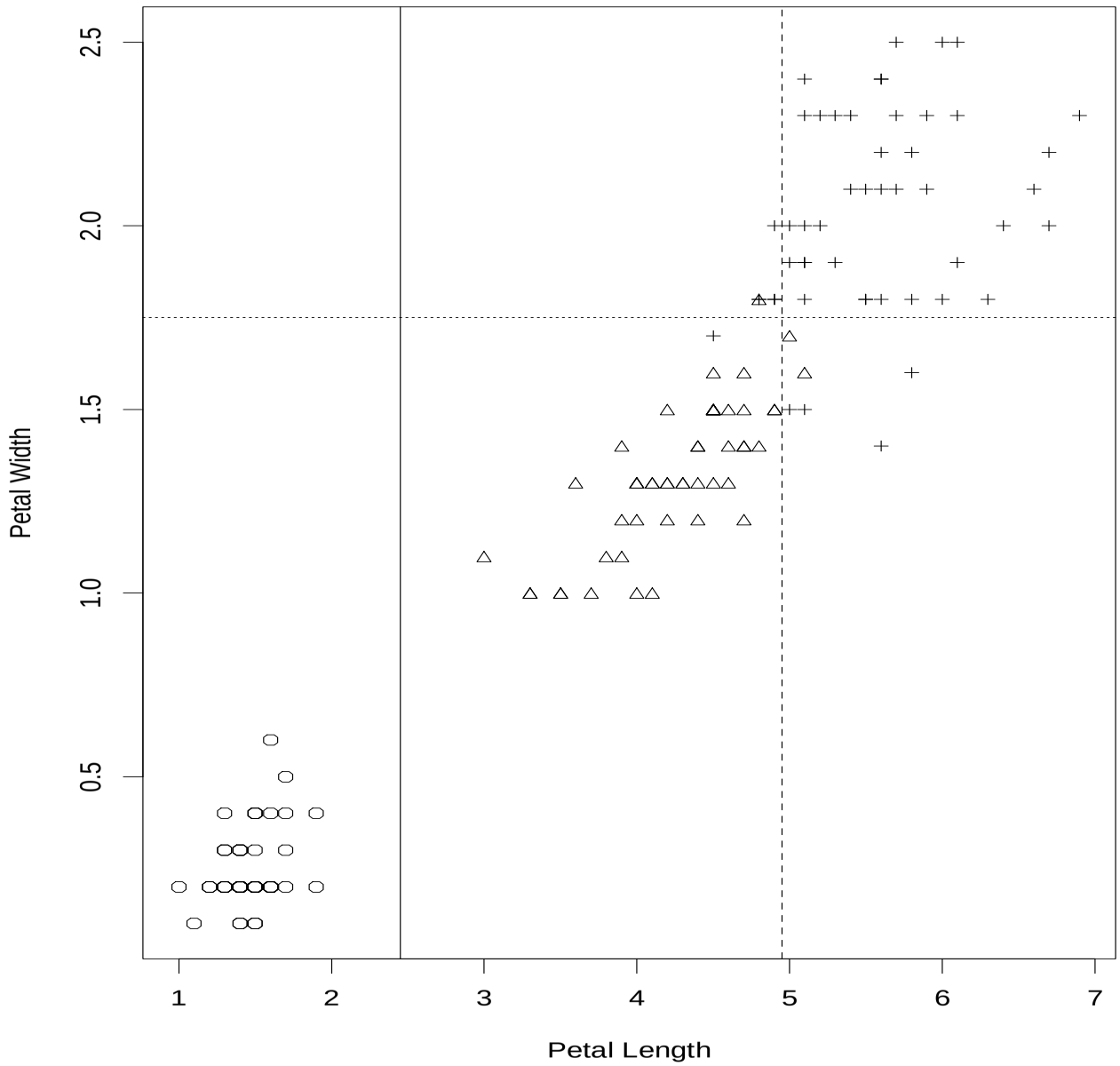
```
plot(petal.length,petal.width,
     xlab="Petal Length",
     ylab="Petal Width",type='n')
points(petal.length[1:50],
     petal.width[1:50], pch=1)
points(petal.length[51:100],
     petal.width[51:100], pch=2)
points(petal.length[101:150],
     petal.width[101:150], pch=3)
abline(v=2.45)
abline(h=1.75,lty=2)
abline(v=4.95,lty=3)

rt <- (petal.width < 1.75) & (variety > 1)
plot(petal.length[rt],sepal.length[rt],
  xlab="Petal Length",ylab="Sepal Length",
     type='n')
points(petal.length[rt & variety ==2],
     sepal.length[rt & variety ==2],pch=2)
points(petal.length[rt & variety ==3],
     sepal.length[rt & variety ==3],pch=3)
abline(v=4.95)
abline(h=5.15,lty=2)
dev.off()
```

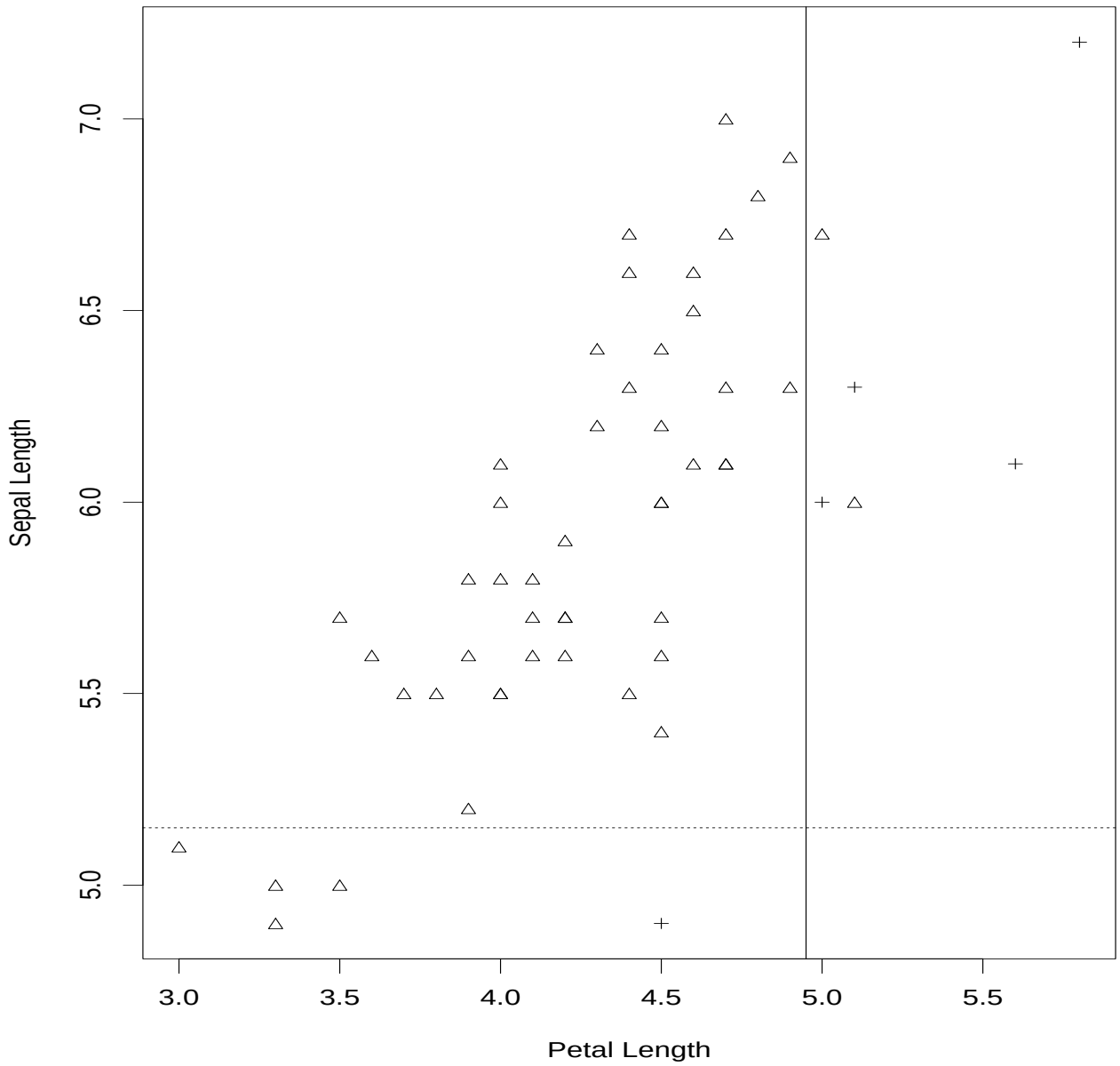
I create the categorical variable variety to hold the labels for the three types of Iris and then essentially regress this categorical variable on the covariates shown. The function `summary` produces:

```
Classification tree:
tree(formula = variety ~ sepal.length
    + sepal.width + petal.length
    + petal.width)
Variables actually used
          in tree construction:
 "petal.length" "petal.width" "sepal.length"
Number of terminal nodes:  6
Residual mean deviance:
        0.1253 = 18.05 / 144
Misclassification error
        rate: 0.02667 = 4 / 150
```

Deviance: fitting multinomial model to probability given flower comes from given species given the covariates. Model has X's as fixed covariates and the responses being the vector of 150 varieties. Though the model does not match the sampling scheme it often produces good classifiers.


petal.length<2.45
Setosa
petal.width<1.75
petal.length<4.95
sepal.length<5.15
Versicolor
Versicolor
Virginica
petal.length<4.95
Virginica
Virginica


Petal Width
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7
Petal Length


Sepal Length
7.0
6.5
6.0
5.5
5.0
3.0
3.5
4.0
4.5
5.0
5.5
Petal Length