Some relevant R functions are

- `prcomp` principal components
- `lm` linear regression
- `lda` linear discriminant analysis (in MASS package)
- `qda` linear discriminant analysis (in MASS package)
- `glm` logistic regression (generalized linear models)
- `glmnet` generalized linear models with lasso or elastic net regularization (in glmnet package)

The `glmnet` package also contains other useful functions.

The following slides illustrate some analyses of the “vowel data” using these functions.
The “Vowel Data”

11 classes in y and 10 features in x.1...x.10

Read in data and do some preliminary analysis

vtrain<-read.table(url,header=TRUE,sep=’,’,')
vtest<-read.table(url,header=TRUE,sep=’,’,')
# First we look at the data structure and organization in the training data:
dim(vtrain)
vtrain[1,]
ntrain<-dim(vtrain)[1]
k<-11

The dimensions are 528 by 12, and the first line is

row.names y  x.1  x.2  x.3  x.4  x.5  x.6  x.7  x.8  x.9  x.10
  1  1 -3.639  0.418 -0.67  1.779 -0.168  1.627 -0.388  0.529 -0.874 -0.814

Make sure the test data are similar

dim(vtest)
vtest[1,]
nittest<-dim(vtest)[1]

row.names y  x.1  x.2  x.3  x.4  x.5  x.6  x.7  x.8  x.9  x.10
  1  1 -1.149 -0.904 -1.988  0.739 -0.06  1.206  0.864  1.196 -0.3 -0.467
The “Vowel Data”

In the caption for Table 4.1, it is stated that “three account for 90% of the variance (via principal components analysis).”

This might lead us to believe that principal components analysis (PCA) is applied to the independent variables, or else to the full data set.

PCA determines the linear combination of variables ("direction") that accounts for the most total variation (proportion of the "generalized variance").

It does make any distinction between a dependent variable and other variables.

Linear discriminant analysis (LDA) does make this distinction, and so would be more appropriate for allocating a proportion of the variance in a classification problem.
I will first illustrate PCA on the vowel data, and then later do a LDA.

We ask how much of the total variation is accounted for by different linear combinations of the variables, without regard for which group the observations are in.

Consider a linear combination of the variables: \( l^T x \).

Compute \( S(l^T x) \) sample variance of \( l^T x \), for a given \( l \).

In \( d \) dimensions there are \( d \) orthogonal vectors like \( l \).

Find the vector \( l_1 \) that maximizes \( S(l^T x) \). then find next \( l_2 \) that is orthogonal to \( l_1 \), and so on.

These are the eigenvectors of \( X^T X \) corresponding to the eigenvalues in descending order.
Principal Component Analysis: Rotated Axes

The linear combinations effectively form a new set of coordinate axes.

The R function `prcomp` computes the principal components and, optionally, returns the data mapped onto the new coordinate axes.

```r
class attach(vtrain)
pc <- prcomp(vtrain[,3:12],retx=TRUE)
```

The new coordinates may be more useful in doing any modeling than are the original independent variables.

The reason is that these axes are orthogonal with respect to the structure of the data.
Principal Component Analysis: Proportion of “Total Variance”

Consider the proportions of the total $\sum_{i=1}^{d} S(l_i^T x)$ accounted for by the first $k$ of these:
$\sum_{i=1}^{k} S(l_i^T x) / \sum_{i=1}^{d} S(l_i^T x)$

Often we only use the linear combinations that account for a significant proportion of the total variation.

```r
tvar <- sum(pc$sdev^2)
for (k in 1:10) print(sum(pc$sdev[1:k]^2/tvar))
```

We get

0.3549357, 0.5517893, 0.7128186, 0.8062732, 0.8631287, 0.9095484, 0.9458702, 0.9739728, 0.9916787, 1

These numbers are not consistent with the caption for Table 4.1, where it is stated that “three account for 90% of the variance (via principal components analysis).”

We next consider use of linear regression on these data, and then later, we will come back to the issue of allocation proportions of variance.
Classifying the “Vowel Data”; Linear Regression

Form indicator matrix $Y$:

$Y <- \text{matrix}(\text{rep}(0, n\text{train}*k), nrow=n\text{train})$

for (i in 1:ntrain) $Y[i, v\text{train}\$y[i]] <- 1$

regfits <- lm($Y \sim x.1+x.2+x.3+x.4+x.5+x.6+x.7+x.8+x.9+x.10$, data=vtrain)

Now we want to get the predicted classes. To do this, we just get the predictions from each of the 11 regressions, and choose the largest one.

This works because of the way that I defined the class variables in the first place as 0's and 1's.

yhat <- numeric(ntrain)
yfits <- matrix(rep(0, ntrain*k), nrow=ntrain)
for (i in 1:ntrain) {
    yfits[i,] <- as.matrix(vtrain[i,3:12])%*%regfits$coef[2:11,] + regfits$coef[1,]
    yhat[i] <- which(yfits[i,] == max(yfits[i,]))
}

It should have been possible to get the predictions from each of the 11 regressions that I put in yfits by use of the R function predict, but this does not seem to work with multivariate regression, so I just formed them directly.
Error Rates for Linear Regression Classification of the “Vowel Data”

We first compute error rate for training data. This is just the proportion of the predicted classes that do not agree with the observed class.

\[
\text{sum}(yhat! = vtrain[,2]) / ntrain
\]

Now do same for test data

\[
yhatterst <- \text{numeric(ntest)}
yfitstest <- \text{matrix(rep(0, ntest * k), nrow = ntest)}
\]

\[
\text{for } (i \text{ in } 1: ntest) \{ \\
\quad \text{yfitstest}[i,] <- \text{as.matrix}(vtest[i,3:12]) \times \text{regfits$coef}[2:11,] + \text{regfits$coef}[1,] \\
\quad \text{yhattest}[i] <- \text{which}(\text{yfitstest}[i,] == \text{max}(\text{yfitstest}[i,])) \\
\}\n\]

\[
\text{sum}(\text{yhattest}[1:ntest] != vtest[,2]) / ntest
\]

We get 0.477 and 0.667, consistent with Table 4.1.
Classifying the “Vowel Data”; Linear Discriminant Analysis

We will now use linear discriminant analysis to set up discriminating functions between each of the groups and all other groups.

We will use the \texttt{lda} function in the \texttt{MASS} library.

\begin{verbatim}
library(MASS)
ldatrain<-lda(y~x.1+x.2+x.3+x.4+x.5+x.6+x.7+x.8+x.9+x.10, data=vtrain)
names(ldatrain)
\end{verbatim}

This function provides a number of useful statistics.

- \texttt{prior} are the prior probabilities, taken by default as the proportions in the data.
- \texttt{means} are the means of the independent variables in each class.
- \texttt{scaling} are the linear discriminant functions.
- \texttt{svd} are the square roots of the ratio of the between- and within-group variances for the linear discriminant variables.
- \texttt{predict} on an object of type \texttt{lda} provides the posterior probabilities.
Classifying the “Vowel Data”; Linear Discriminant Analysis

First of all, I’ll return to the question of percentage “of the variance” in Table 4.1.

Using LDA, we have treated the dependent variable (the class variable) specially, so this is more relevant.

The entries in $\text{svd}$ are the square roots of the ratio of the between-and within-group variances for the linear discriminant variables, so we might try that.
Linear Discriminant Analysis: Proportion of “Total Variance”

tvar <- sum(ldatrain$svd^2)
for (k in 1:10) print(sum(ldatrain$svd[1:k]^2/tvar))

We get

0.5616626, 0.9134936, 0.9580326, 0.9771749, 0.9878383
0.9961340, 0.9987125, 0.9997738, 0.9999154, 1

These numbers are not consistent with the caption for Table 4.1, where it is stated that “three account for 90% of the variance (via principal components analysis),” so I’m not sure what that statement means.
Classifying the “Vowel Data”; Linear Discriminant Analysis

We form a rotated dataset as the projections of the original data onto the linear discriminant functions.

\[
\text{rotdat} \leftarrow \text{as.matrix(vtrain[,3:12])}\%*\%\text{ldatrain}\$\text{scaling}
\]

\[
\text{plot(rotdat[,1],rotdat[,2],col=vtrain[,2],}
\]
\[
\text{xlab=}\text{"First Linear Discriminant for Training Data"},
\]
\[
\text{ylab=}\text{"Second Linear Discriminant for Training Data"},)
\]

We get the graph shown on the next page, which, except for the colors is the same as Figure 4.4.

Now we compute the error proportion for the training data

\[
\text{sum(predict(ldatrain,vtrain)$class!=vtrain[,2])/ntrain}
\]

We get 0.316, which agrees with the entry in Table 4.1.
Classifying the “Vowel Data”; Quadratic Discriminant Analysis

This follows the analysis shown for LDA.

library(MASS)
qdatrain<-qda(y~x.1+x.2+x.3+x.4+x.5+x.6+x.7+x.8+x.9+x.10, data=vtrain)

We can follow the same steps as on the previous slides.

predict on an object of type qda provides the posterior probabili-
ties.
Classifying the “Vowel Data”; Logistic Regression

Logistic regression is based on a binomial model.

Handle this similar to linear regression analysis. Exercise!