Discriminant Analysis

Linear Discriminant Analysis

The main steps of LDA include:

- **Compute Class Statistics**
- Calculate the mean vector for each class (centroid of class data points).
- Compute the within-class scatter matrix, which measures how data points scatter within each class.
- Compute the between-class scatter matrix, which measures how the class centroids are spread relative to each other.
- **Find Linear Discriminants**
  - Solve an optimization problem to find a projection that maximizes the ratio of the between-class scatter to within-class scatter.
- **Project Data**
  - Project the original dataset onto the linear discriminants (a lower-dimensional space). In this space, classes are more separable.
- **Classification**
  - Use a simple classifier, such as computing the distance to class means, for predicting the class of a new observation.
- perform LDA on the Smarket data
- In R, we fit an LDA model using the lda() function, which is part of the MASS library

```r
library(MASS)
library(tidyverse)
library(ISLR2)
```

```r
training_data <- Smarket[Smarket$Year < 2005, ]
testing_data <- Smarket[Smarket$Year == 2005, ]
```

```r
attach(Smarket)

lda.fit <- lda(Direction ~ Lag1 + Lag2, data = training_data)
lda.fit
```

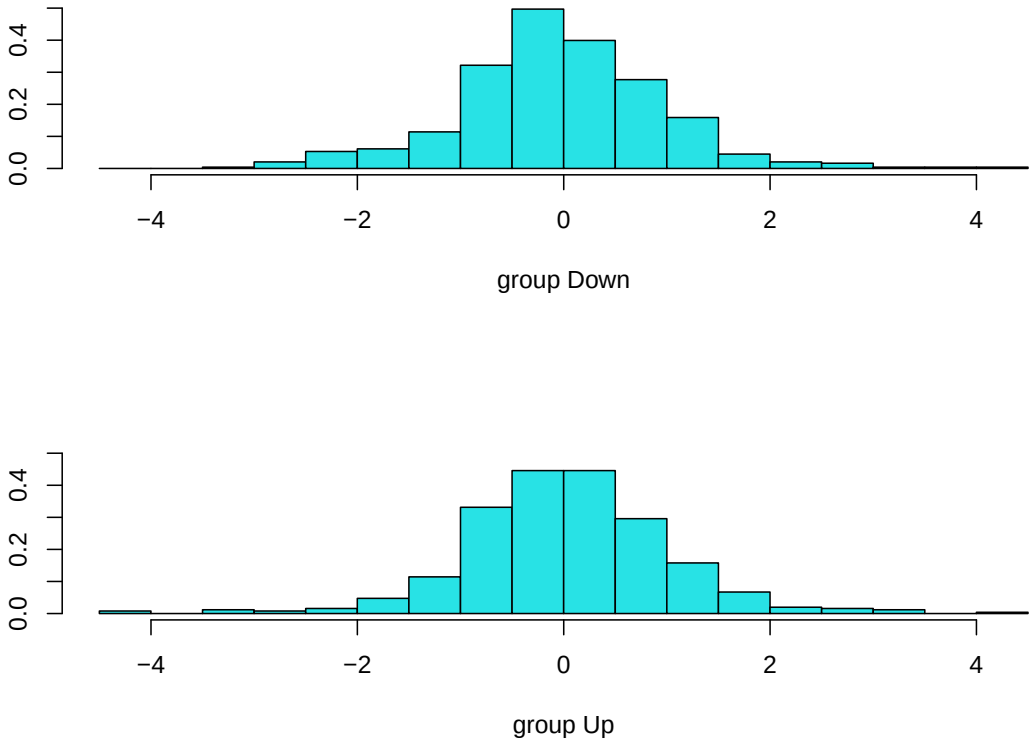
```
Call:
lda(Direction ~ Lag1 + Lag2, data = training_data)

Prior probabilities of groups:
    Down       Up
0.491984 0.508016

Group means:
            Lag1        Lag2
Down  0.04279022  0.03389409
Up   -0.03954635 -0.03132544

Coefficients of linear discriminants:
            LD1
Lag1 -0.6420190
Lag2 -0.5135293
```

Interpretation of LDA model

- The LDA output indicates that $\mathbf{1 = 0.492,\ 2 = 0.508}$ (Prior Probabilities)
- Approximately 49.2% of the training observations belong to the "Down" class, while 50.8% belong to the "Up" class. These priors are used in classification to reflect the proportion of each class in the data.
- **Group Means** : These are the **centroids** (average values) of the predictors Lag1 and Lag2 for each class ("Down" and "Up").
- When the market moves "Down," the mean of Lag1 is approximately 0.0428, and the mean of Lag2 is approximately 0.0339.
- When the market moves "Up," the mean of Lag1 is approximately -0.0395, and the mean of Lag2 is approximately -0.0313. -This indicates that the "Down" and "Up" classes have distinct centers in the feature space, which LDA uses for classification.

- **Coefficients of Linear Discriminants**
- These are the coefficients of the linear discriminant function (LD1), which is used to compute a linear combination of the predictors (Lag1 and Lag2) to separate the classes.
- The negative coefficients suggest that higher values of Lag1 and Lag2 are associated with the "Down" class, while lower values are associated with the "Up" class.

```
plot(lda.fit)
```



- **predict()** function returns 3 elements
  - class : contains LDA's predictions about the movement of the market
  - posterior : matrix whose k th column contains the posterior probability that the corresponding observation belongs to the th class. Posterior probabilities are calculated using Bayes' theorem.
  - x : contains the linear discriminants. These scores are projections of the data onto the linear discriminants, which are combinations of predictor variables that maximize class separation.

```r
lda.pred <- predict(lda.fit, testing_data)
names(lda.pred)
```

```
[1] "class"     "posterior" "x"
```

```r
lda.pred$class[1:5]
```

```
[1] Up Up Up Up Up
Levels: Down Up
```

```r
# 3 rows all columns
lda.pred$posterior[1:4,]
```

```
          Down        Up
999  0.4901792 0.5098208
1000 0.4792185 0.5207815
1001 0.4668185 0.5331815
1002 0.4740011 0.5259989
```

```r
lda.pred$x[1:3]
```

```
[1] 0.08293096 0.59114102 1.16723063
```

```r
lda.class <- lda.pred$class
table(lda.class, testing_data$Direction)
```

```
lda.class Down  Up
     Down   35  35
     Up     76 106
```

```r
mean(lda.class == testing_data$Direction)
```

```
[1] 0.5595238
```

```r
sum(lda.pred$posterior[, 1] >= 0.5)
```

```
[1] 70
```

```r
sum(lda.pred$posterior[, 1] < 0.5)
```

```
[1] 182
```

```r
# set the threshold for 0.9
sum(lda.pred$posterior[, 1] > .9)
```

```
[1] 0
```

Quadratic Discriminant Analysis

- QDA is implemented in R using the qda() function, which is also part of the MASS library.
- The **output** contains :
  - prior probabilities : reflect the likelihood of each class before considering the predictors.
  - group means : The mean values of each predictor for each class.
  - NOT contain coefficients of the linear discriminants -
    - Unlike LDA, QDA does not output coefficients of linear discriminants because QDA models the decision boundary as a quadratic function of the predictors.
    - This flexibility allows QDA to capture non-linear relationships between the predictors and the class labels.

```r
qda.fit <- qda(Direction ~ Lag1 + Lag2, data = training_data)
qda.fit
```

```
Call:
qda(Direction ~ Lag1 + Lag2, data = training_data)

Prior probabilities of groups:
    Down       Up
0.491984 0.508016

Group means:
```

```
            Lag1        Lag2
Down  0.04279022  0.03389409
Up   -0.03954635 -0.03132544
```

```
qda.class <- predict(qda.fit, testing_data)$class
table(qda.class, testing_data$Direction)
```

```
qda.class Down  Up
     Down   30  20
     Up     81 121
```

```
mean(qda.class == testing_data$Direction)
```

```
[1] 0.5992063
```

- the QDA predictions are accurate almost 60% of the time for testing set. This is good for stock market data.