DISCRIMINANT ANALYSIS – LDA and QDA

 # We'll try to predict the fuel ECO rating of automobiles.

```
> library(ISLR)   # This library contains datasets from our textbook (ISLR = name of our text). If this command returns
                  # an error, go to the top of the R command window, choose "Packages" → "Install package(s)…",
                  # choose a preferably US repository, and in the long list of packages, choose ISLR.
> attach(Auto)
> names(Auto)                          # List of variables in this dataset
[1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"       "acceleration"   "year"         "origin"       "name"
> summary(mpg)                         # ECO rating will be defined based on miles per gallon
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   9.00   17.00   22.75   23.45   29.00   46.60
                                       # Initiate a fuel consumption rating variable that will be treated as categorical
> ECO = rep("Fuel", length(mpg))
> ECO[mpg < 17] = "Heavy"
> ECO[mpg >= 17 & mpg < 22.75] = "OK"
> ECO[mpg >= 22.75 & mpg < 29] = "Economy"
> ECO[mpg >= 29] = "Excellent"

> table(ECO)                                  # We used sample quartiles of variable mpg to define these ratings,
ECO                                           # that's why we got four approximately equal groups.
      Economy  Excellent       Heavy          OK      # Now, we'll derive a classification rule, using other car characteristics
           93         103          92         104
```

Linear Discriminant Analysis

```
> library(MASS)                        # Package MASS ("Modern Applied Statistics with S") contains LDA and QDA

> lda( ECO ~ acceleration + year + horsepower + weight )          # The main command for LDA

Prior probabilities of groups:                  # These are sample proportions of the 4 groups, from our data
   Economy  Excellent      Heavy         OK
0.2372449 0.2627551 0.2346939 0.2653061

Group means:                                    # Multivariate group means are computed within each group
          acceleration     year horsepower   weight
Economy       16.33011 76.04301   87.82796 2537.387
Excellent     16.64757 78.93204   70.69903 2151.816
Heavy         13.23043 73.29348  158.20652 4151.380
OK            15.78462 75.37500  105.25962 3150.692

Coefficients of linear discriminants:                   # For our information only: these functions LD1-LD3
                     LD1          LD2          LD3              # are different from our linear discriminant
              functions.
acceleration -0.011123931  0.031857342 -0.249711185   # These printed coefficients determine the Fisher's
year         -0.193137397 -0.233122185  0.153228971   # linear discriminants LD1, LD2, LD3. The first one is
horsepower    0.009199232 -0.044693477 -0.050634817   # a linear function that achieves the maximal
weight        0.002222240  0.001371949  0.002151756   # separation of our four groups. LD2 is a linear
                                                              # function, orthogonal to LD1, that achieves the
Proportion of trace:                            # maximal separation among all linear functions orthogonal to LD1, etc.
   LD1    LD2    LD3                           # These functions are linear combinations of our linear discriminant functions.
0.9814 0.0128 0.0058                           # Their derivation is based on Linear Algebra. Here, LD1 captures 98% of differences
                                               # between the groups, LD2 adds 1% to that, and LD3 adds less than 1%.
```

Cross-validation

 # Option CV=TRUE is used for "leave one out" cross-validation; for each sampling unit, it gives its class assignment without
 # the current observation. This is a method of estimating the testing classifications rate instead of the training rate.

```
> lda.fit = lda( ECO ~ acceleration + year + horsepower + weight, CV=TRUE )
> table( ECO, lda.fit$class )

ECO          Economy Excellent Heavy OK
  Economy         61        20     0 12
  Excellent       15        86     0  2
  Heavy            0         0    78 14
  OK              22         1     8 73
```

 # The main diagonal shows correctly classified counts.

```
> mean( ECO == lda.fit$class )
[1] 0.7602041
```

 # Correct classification rate = proportion of correctly classified counts.

Prior probabilities of classes

 # We can also specify our own prior distribution; c(…,…) lists prior probabilities in the same order the classes are listed.

```
> lda.fit = lda( ECO ~ acceleration + year + horsepower + weight, prior=c(0.25,0.25,0.25,0.25), CV=TRUE )
> table( ECO, lda.fit$class )

ECO          Economy Excellent Heavy OK
  Economy         68        14     0 11
  Excellent       16        86     0  1
  Heavy            0         0    79 13
  OK              22         1     8 73

> mean( ECO == lda.fit$class )
[1] 0.7806122
```

 # The prior made an impact on our results, actually improving the rate

```
> lda.fit = lda( ECO ~ acceleration + year + horsepower + weight, prior=c(0.4,0.3,0.2,0.1), CV=TRUE )
> mean( ECO == lda.fit$class )
[1] 0.7219388
```

 # This prior (40% of cars are heavy consumers of fuel) is perhaps unrealistic.

Posterior probabilities of classes

```
> lda.fit$class[1:20]
 [1] Heavy   Heavy   Heavy   Heavy   Heavy   Heavy   Heavy   Heavy   Heavy   Heavy   Heavy
[12] Heavy   Heavy   Heavy   Economy OK      Economy Economy Economy Economy
Levels: Economy Excellent Heavy OK
```

 # We can see the class assignment for each car in our sample

```
> lda.fit$posterior[1:20, ]
        Economy    Excellent        Heavy           OK
1  3.337765e-03 1.138435e-06 8.845722e-01 1.120889e-01
2  1.060121e-04 1.353499e-08 9.947060e-01 5.187958e-03
3  2.468535e-03 8.412574e-07 9.509393e-01 4.659134e-02
4  3.578640e-03 1.264177e-06 9.371892e-01 5.923092e-02
```

 # R also computes all the posterior probabilities

 # Each line here contains $p_k(x) = P(Y=k \mid X=x)$,
 # the posterior probability for the corresponding
 # car (row) to belong to the given class (column)
 # The group (column) with the highest
 # posterior probability will be the Bayes decision,

```
5  3.074928e-03 1.231316e-06 9.175840e-01 7.933985e-02 # computed by LDA.
6  1.551166e-08 1.557705e-13 9.999864e-01 1.356390e-05
7  3.669641e-09 2.575480e-14 9.999973e-01 2.714342e-06
8  7.186054e-09 6.762158e-14 9.999950e-01 4.966257e-06
9  1.414734e-09 6.269786e-15 9.999988e-01 1.225826e-06
10 4.052546e-06 2.778368e-10 9.995869e-01 4.090868e-04
11 2.839730e-04 5.893738e-08 9.916736e-01 8.042372e-03
12 2.271364e-04 5.802140e-08 9.873464e-01 1.242643e-02
13 8.508487e-05 1.341557e-08 9.900835e-01 9.831387e-03
14 1.022746e-02 4.754893e-06 9.842674e-01 5.500354e-03
15 8.475534e-01 1.646497e-02 1.452560e-04 1.358364e-01
16 4.196166e-01 1.714934e-03 8.656601e-03 5.700118e-01
17 5.013899e-01 2.409147e-03 6.040770e-03 4.901602e-01
18 6.892842e-01 7.005241e-03 5.748319e-04 3.031358e-01
19 9.023775e-01 4.544958e-02 8.819585e-06 5.216407e-02
20 8.599324e-01 1.289021e-01 1.940575e-08 1.116549e-02

> rowSums(lda.fit$posterior[1:20,])     # These are discrete distributions, probabilities for each unit add to 1
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

Quadratic Discriminant Analysis

```
> qda.fit = qda(ECO ~ acceleration + year + horsepower + weight, prior=c(0.25,0.25,0.25,0.25), CV=TRUE )
> table( ECO, qda.fit$class )          # Similar commands

ECO          Economy Excellent Heavy OK
  Economy         68        14     0 11
  Excellent       13        89     0  1
  Heavy            0         0    79 13
  OK              24         0     9 71

> mean( ECO == qda.fit$class )
[1] 0.7831633                          # Here, QDA has a slightly better prediction power than LDA
```