LDA: Banknotes

Gustavo Cepparo and Milica Cudina

We consider another **banknote** data set. It exists in the `MixGHD` library.

```
#install.packages("MixGHD")
library(MixGHD)
```

```
## Loading required package: MASS
```

The data set `banknote` is used to predict the `Status` of Swiss Franc bills based on measurements of the bills.

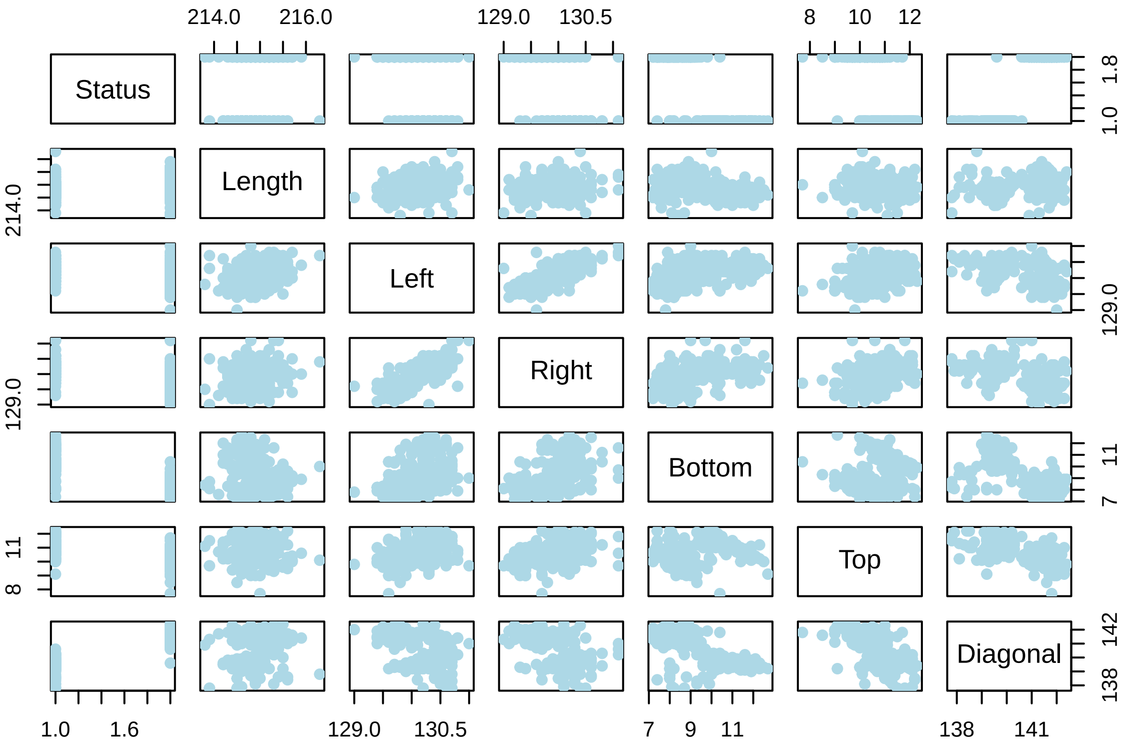
```
data(banknote)
names(banknote)
```

```
## [1] "Status"   "Length"   "Left"     "Right"    "Bottom"   "Top"      "Diagonal"
```

```
attach(banknote)
data<-banknote
```

We can start with visualization (as usual).

```
pairs(data,
      col="lightblue", pch=19)
```



Our aim is to perform the linear discriminant analysis here. First, we will partition our data into training and testing.

```r
set.seed(1)
ind.train <- sample(nrow(data), floor(nrow(data)*0.6))
training<-data[ind.train,]
testing<-data[-ind.train,]
```

My goal with this example is to ultimately demonstrate how to plot a partition in the plane. So, I will “arbitrarily” choose two explanatory variables: `Top` and `Diagonal`.

```r
library(MASS)
linear <- lda(Status~Top+Diagonal, data=training)
summary(linear)
```

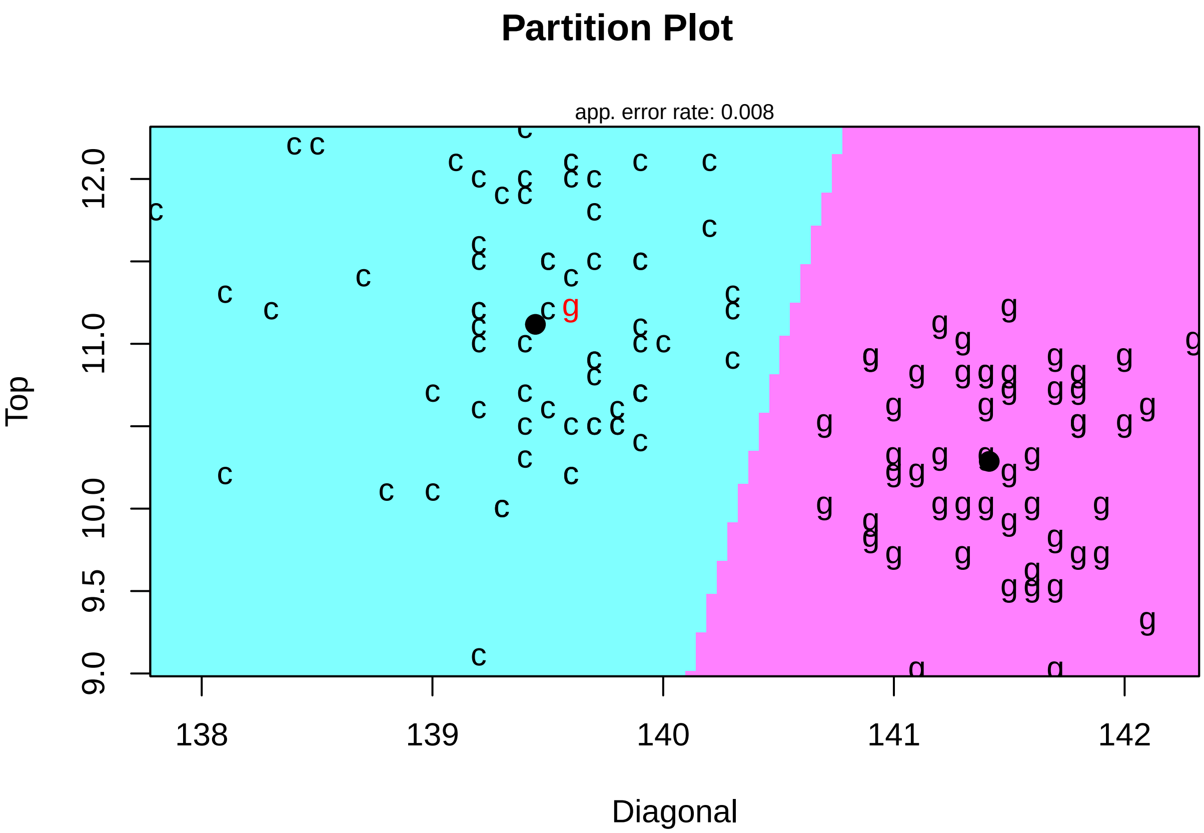
```
##         Length Class  Mode
## prior   2      -none- numeric
## counts  2      -none- numeric
## means   4      -none- numeric
## scaling 2      -none- numeric
## lev     2      -none- character
## svd     1      -none- numeric
## N       1      -none- numeric
## call    3      -none- call
## terms   3      terms  call
## xlevels 0      -none- list
```

```r
linear
```

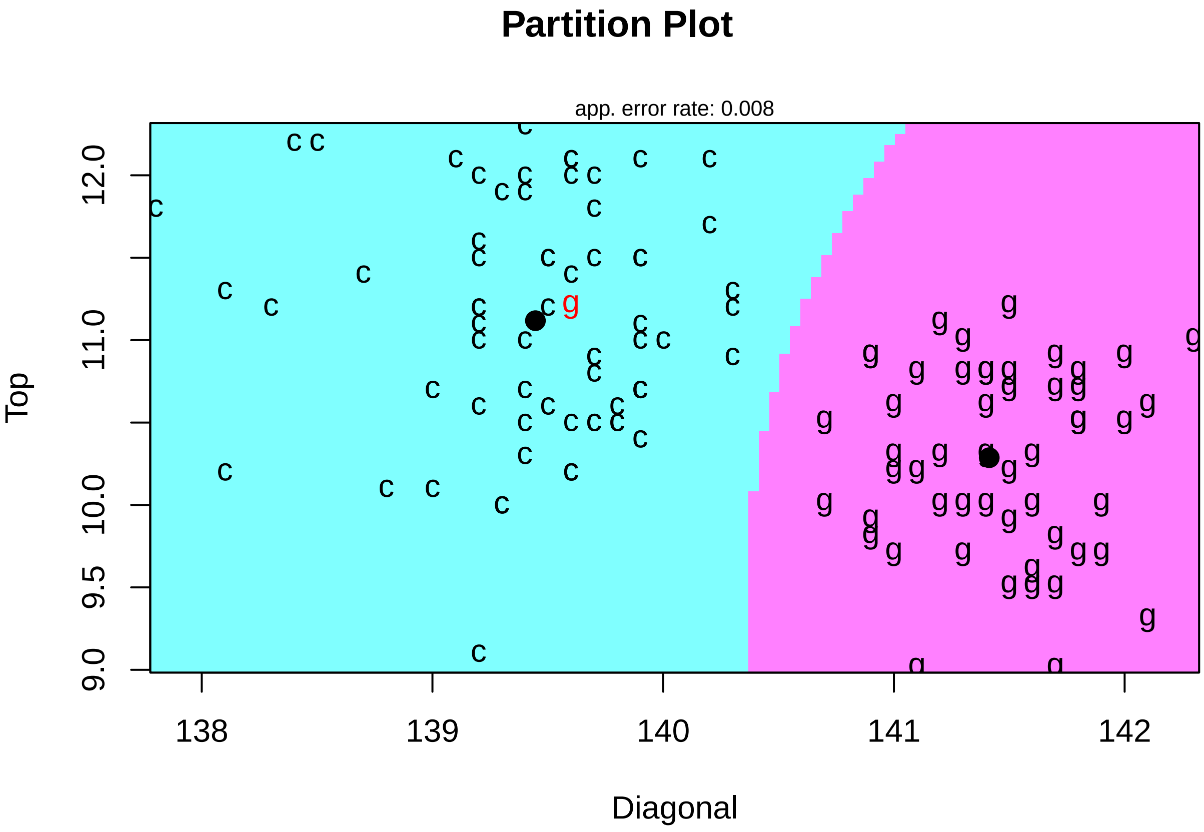
```
## Call:
## lda(Status ~ Top + Diagonal, data = training)
##
## Prior probabilities of groups:
## counterfeit     genuine
##   0.5583333   0.4416667
##
## Group means:
##                   Top Diagonal
## counterfeit 11.11791 139.4463
## genuine     10.28679 141.4132
##
## Coefficients of linear discriminants:
##                  LD1
## Top      -0.3911736
## Diagonal  1.9162568
```

Now, we try to create a partition plot.

```r
#install.packages("klaR")
library(klaR)
partimat(Status~Top+Diagonal, data = training, method = "lda")
```



```r
partimat(Status~Top+Diagonal, data = training, method = "qda")
```



Finally, let's consider the confusion matrix and accuracy, both on the training set and the testing set.

```
#the training set
p.tr <- predict(linear, training)$class
tab.tr <- table(Predicted = p.tr, Actual = training$Status)
tab.tr
```

```
##              Actual
## Predicted     counterfeit genuine
##   counterfeit          67       1
##   genuine               0      52
```

```
sum(diag(tab.tr))/sum(tab.tr)
```

```
## [1] 0.9916667
```

```
#this is the error rate indicated in the label on the plot above
1-sum(diag(tab.tr))/sum(tab.tr)
```

```
## [1] 0.008333333
```

```
#the testing set
p.te <- predict(linear, testing)$class
tab.te <- table(Predicted = p.te, Actual = testing$Status)
tab.te
```

```
##              Actual
## Predicted     counterfeit genuine
##   counterfeit          33       0
##   genuine               0      47
```

```
sum(diag(tab.te))/sum(tab.te)
```

```
## [1] 1
```