35459 Multivariate Statistics

Week 10 (approx) Seminar - Discrimination and Classification I

The goal of *discrimination* is to determine a rule that can divide the sample into two ore more groups based on the value of a variable of interest. *Classification* is then used to predict new values for the variable of interest using the discrimination model.

Swiss Bank Notes

The data in Notes.csv contain various characteristics of 100 genuine and 100 counterfeit Swiss bank notes. The characteristics include:

- Length of the bank note
- Height of the bank note, measured on the left
- Height of the bank note, measured on the right
- Distance of inner frame to the lower border
- Distance of inner frame to the upper border
- Length of the diagonal

Observations 1-100 are the genuine bank notes and the other 100 observations are the counterfeit bank notes. Construct a discriminant model that can be used to classify new notes. How well does this model perform?

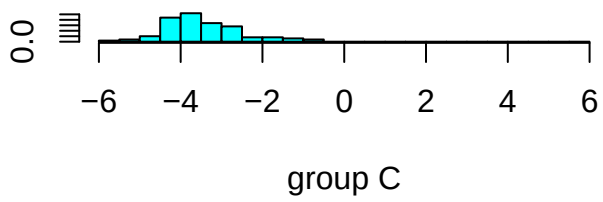
```
library(MASS)
library(ggplot2)
library(klaR)
```

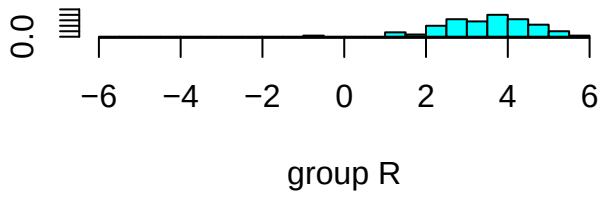
```
notes_data<-read.csv("C:/Documents/Notes.csv") notes_data$Group<-
substr(notes_data$Status,1,1)
notes_lda<-lda(Group ~ Length+Height+Height.1+Inner.Frame+Inner.Frame.1
               +Diagonal,data=notes_data)
notes_lda

## Call:
## lda(Group ~ Length + Height + Height.1 + Inner.Frame + Inner.Frame.1 +
##     Diagonal, data = notes_data)
##
## Prior probabilities of groups:
##   C   R
## 0.5 0.5
```

```
##
## Group means:
##   Length Height Height.1 Inner.Frame Inner.Frame.1 Diagonal
## C  214.8  130.3    130.2      10.530         11.13    139.4
## R  215.0  129.9    129.7       8.305         10.17    141.5
##
## Coefficients of linear discriminants:
##                      LD1
## Length          0.005011
## Height          0.832433
## Height.1       -0.848993
## Inner.Frame    -1.117336
## Inner.Frame.1 -1.178884
## Diagonal        1.556521

plot(notes_lda)
```



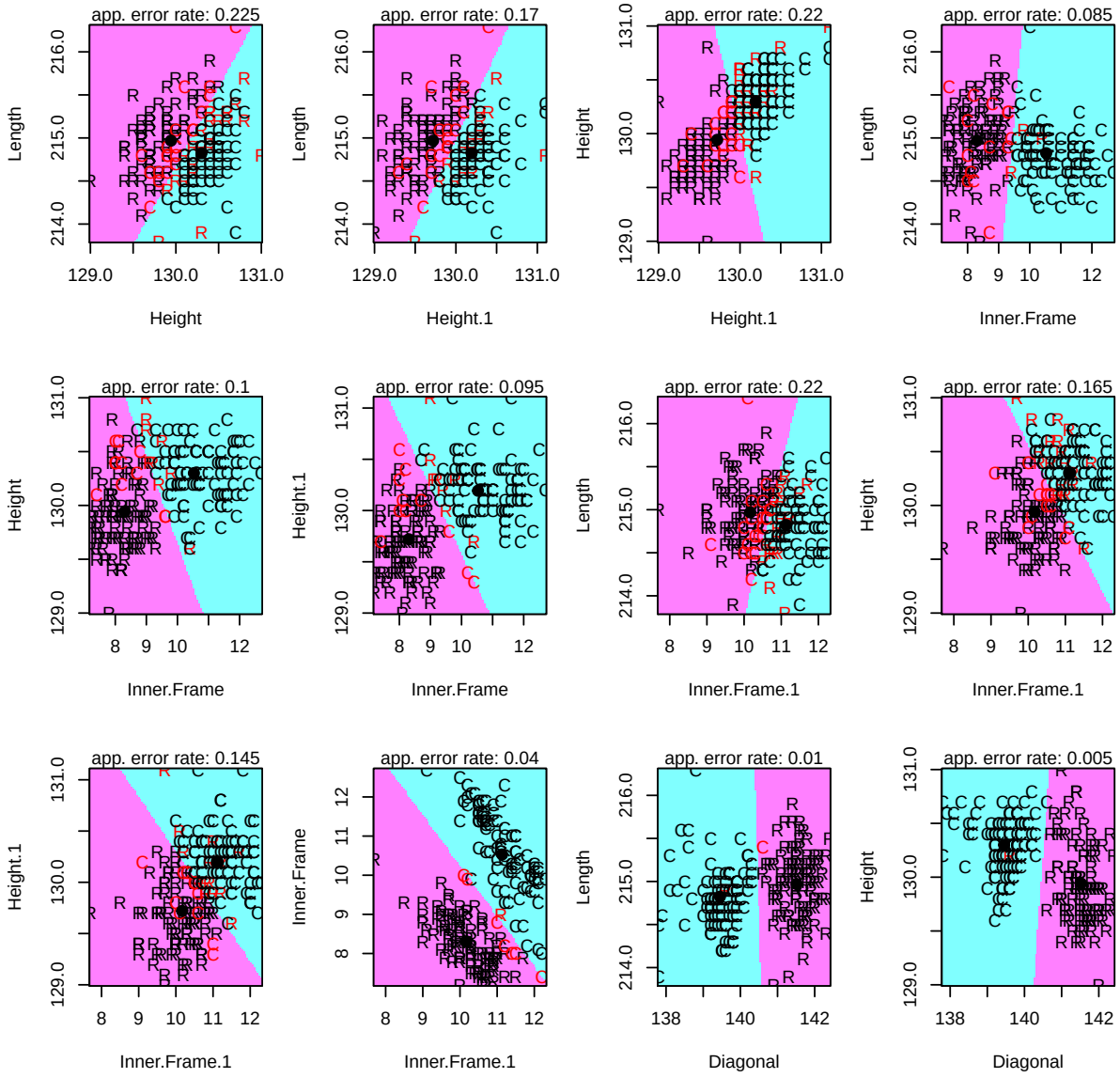


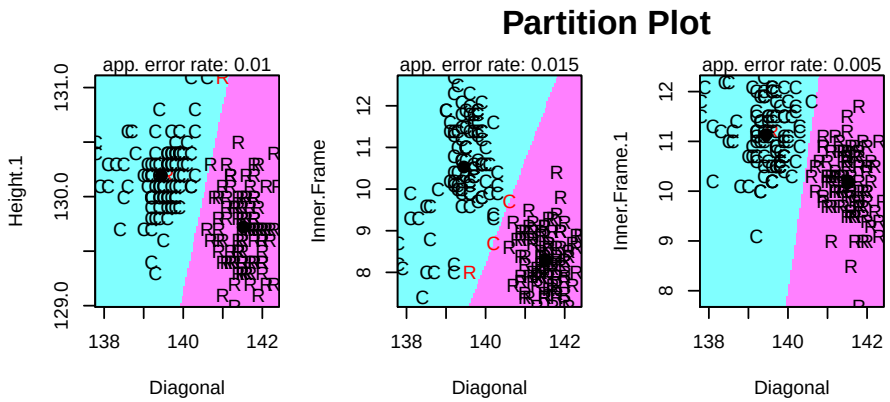

```
notes_data$lda_predict<-predict(notes_lda,notes_data[,1:6])$class
table(notes_data$Group,notes_data$lda_predict)

##
##       C   R
##   C 100   0
##   R   1  99
```

So AER=1/200=0.005.

```
partimat(as.factor(Group) ~ Length+Height+Height.1+Inner.Frame+Inner.Frame.1
                +Diagonal,data=notes_data,method="lda")
```






```
notes_qda<-qda(Group ~ Length+Height+Height.1+Inner.Frame+Inner.Frame.1
               +Diagonal,data=notes_data)
notes_qda

## Call:
## qda(Group ~ Length + Height + Height.1 + Inner.Frame + Inner.Frame.1 +
##     Diagonal, data = notes_data)
##
## Prior probabilities of groups:
##   C   R
## 0.5 0.5
##
## Group means:
```

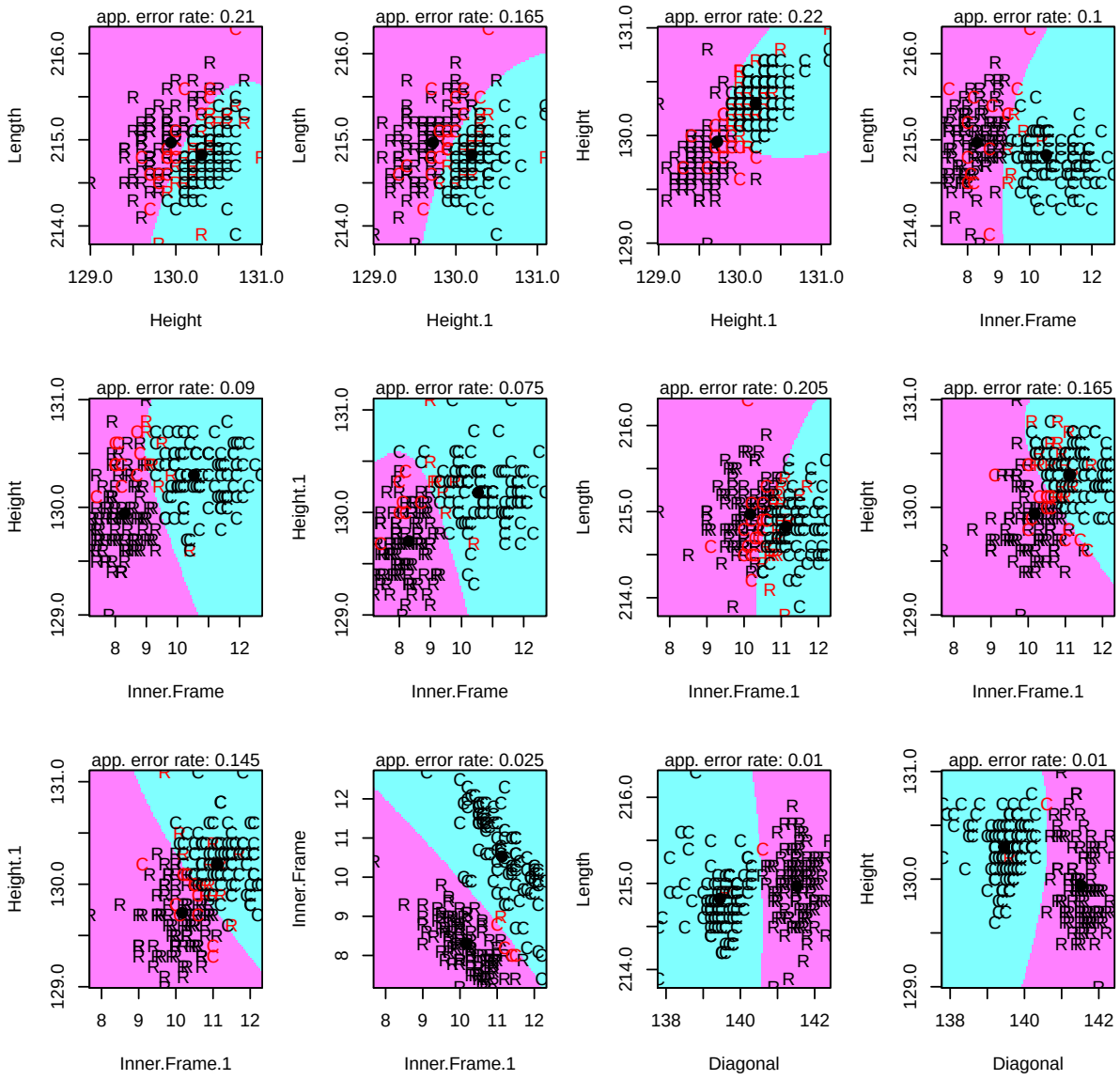
```
##   Length Height Height.1 Inner.Frame Inner.Frame.1 Diagonal
## C  214.8  130.3    130.2      10.530         11.13    139.4
## R  215.0  129.9    129.7       8.305         10.17    141.5
```

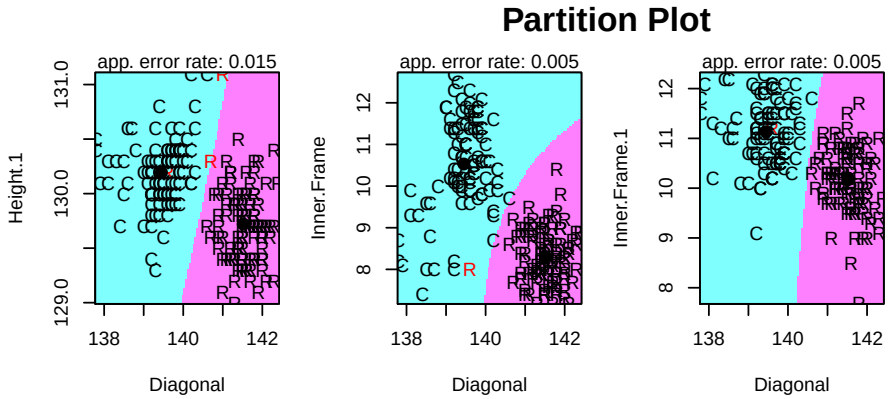
```
notes_data$qda_predict<-predict(notes_qda,notes_data[,1:6])$class
table(notes_data$Group,notes_data$qda_predict)

##
##       C   R
##   C 100   0
##   R   1  99
```

Again, AER=1/200=0.005.

```
partimat(as.factor(Group) ~ Length+Height+Height.1+Inner.Frame+Inner.Frame.1
              +Diagonal,data=notes_data,method="qda")
```


app. error rate: 0.21
Length
Height
app. error rate: 0.165
Length
Height.1
app. error rate: 0.22
Height
Height.1
app. error rate: 0.1
Length
Inner.Frame
app. error rate: 0.09
Height
Inner.Frame
app. error rate: 0.075
Height.1
Inner.Frame
app. error rate: 0.205
Length
Inner.Frame.1
app. error rate: 0.165
Height
Inner.Frame.1
app. error rate: 0.145
Height.1
Inner.Frame.1
app. error rate: 0.025
Inner.Frame
Inner.Frame.1
app. error rate: 0.01
Length
Diagonal
app. error rate: 0.01
Height
Diagonal



It is good practice to divide the data that we have into two groups:

- Training Data: the data used to build the model
- Test Data: the data used to determine whether the model can predict using observations that the model wasn't built on

The purpose of having these separate data sets is to gain some insight about whether the model is *overfitting* the data. That is, the model does very well in predicting observations that were used to build the model, but poorly when new data are introduced.

We can refit the linear and quadratic models using the first 80 observations for each category as the training data and the remaining 20 observations as the test data.

```
notes_data_training<-notes_data[c(1:80,101:180),]
notes_data_test<-notes_data[c(81:100,181:200),]
notes_lda<-lda(Group ~ Length+Height+Height.1+Inner.Frame+Inner.Frame.1
               +Diagonal,data=notes_data_training)
notes_lda
```

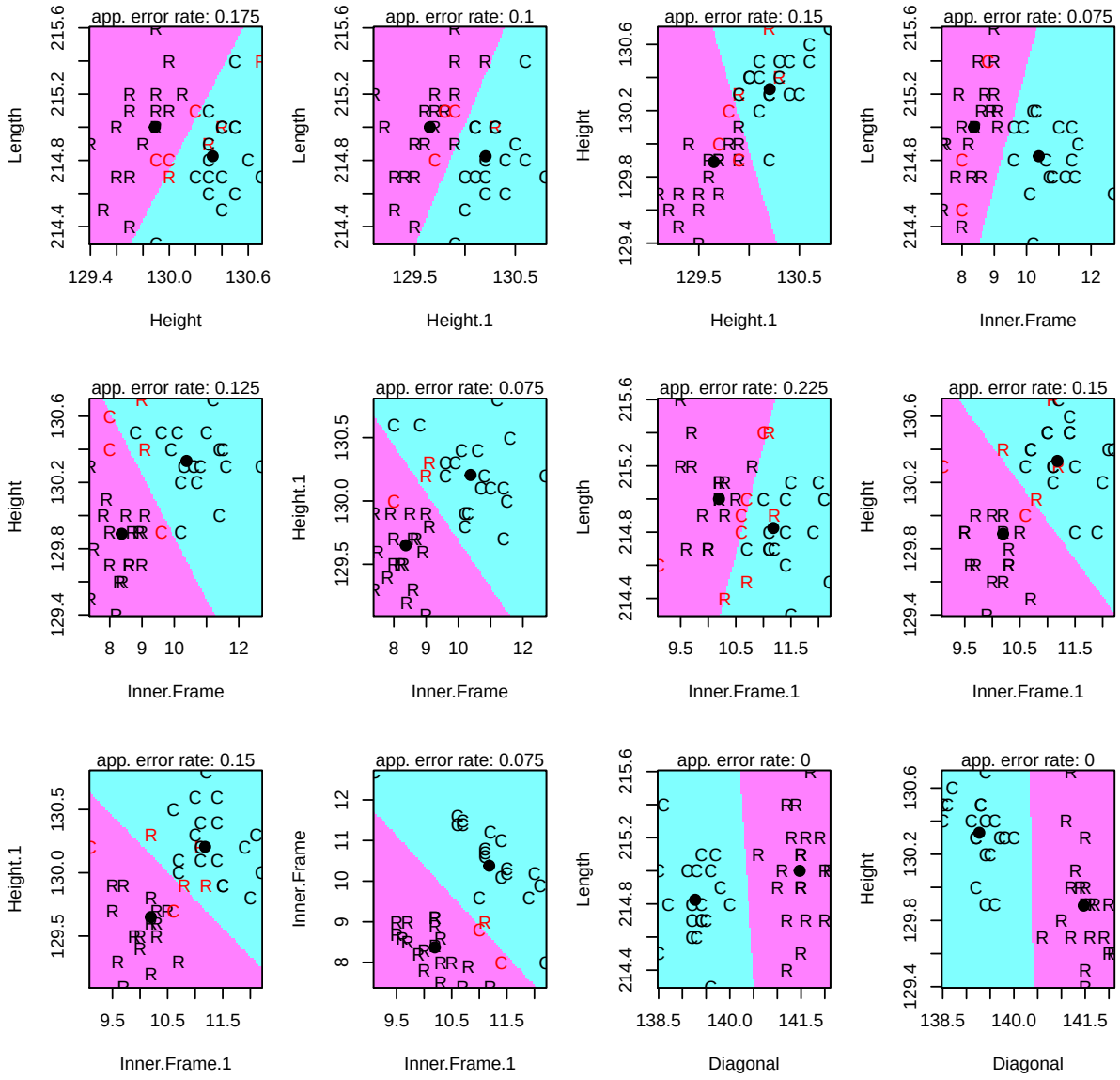
```
## Call:
## lda(Group ~ Length + Height + Height.1 + Inner.Frame + Inner.Frame.1 +
##     Diagonal, data = notes_data_training)
##
## Prior probabilities of groups:
##   C   R
## 0.5 0.5
##
## Group means:
##   Length Height Height.1 Inner.Frame Inner.Frame.1 Diagonal
## C  214.8  130.3    130.2      10.568         11.12    139.5
## R  215.0  130.0    129.7       8.289         10.16    141.5
##
## Coefficients of linear discriminants:
##                     LD1
## Length        -0.08273
## Height         0.73717
## Height.1      -0.76727
## Inner.Frame   -1.11703
## Inner.Frame.1 -1.15583
## Diagonal       1.42627
```

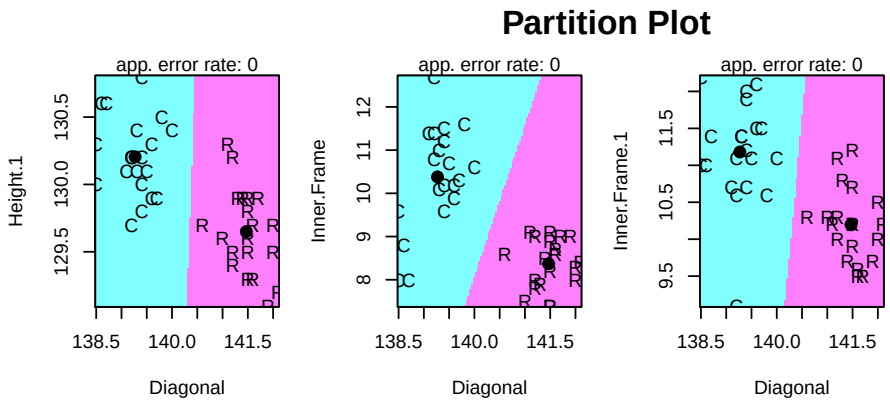
```
notes_data_test$lda_predict<-predict(notes_lda,notes_data_test[,1:6])$class
table(notes_data_test$Group,notes_data_test$lda_predict)
```

```
##
##      C  R
##   C 20  0
##   R  0 20
```

In this case, AER=0/40=0

```
partimat(as.factor(Group) ~ Length+Height+Height.1+Inner.Frame+Inner.Frame.1
               +Diagonal,data=notes_data_test,method="lda")
```


app. error rate: 0.175
Length
Height
app. error rate: 0.1
Length
Height.1
app. error rate: 0.15
Height
Height.1
app. error rate: 0.075
Length
Inner.Frame
app. error rate: 0.125
Height
Inner.Frame
app. error rate: 0.075
Height.1
Inner.Frame
app. error rate: 0.225
Length
Inner.Frame.1
app. error rate: 0.15
Height
Inner.Frame.1
app. error rate: 0.15
Height.1
Inner.Frame.1
app. error rate: 0.075
Inner.Frame
Inner.Frame.1
app. error rate: 0
Length
Diagonal
app. error rate: 0
Height
Diagonal




```
notes_qda<-qda(Group ~ Length+Height+Height.1+Inner.Frame+Inner.Frame.1
               +Diagonal,data=notes_data_training)
notes_qda

## Call:
## qda(Group ~ Length + Height + Height.1 + Inner.Frame + Inner.Frame.1 +
##     Diagonal, data = notes_data_training)
##
## Prior probabilities of groups:
##   C   R
## 0.5 0.5
##
## Group means:
```

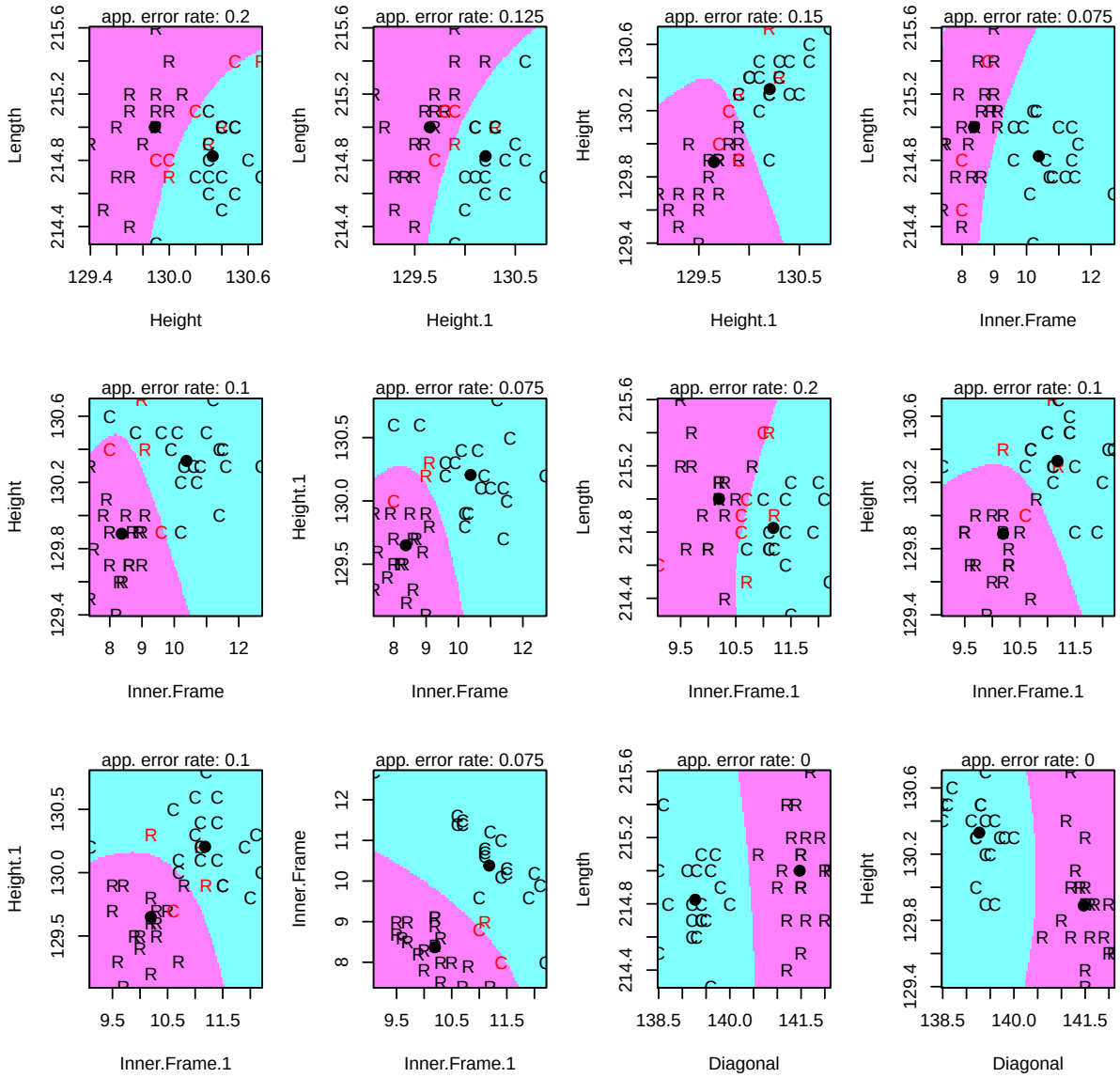
```
##   Length Height Height.1 Inner.Frame Inner.Frame.1 Diagonal
## C  214.8  130.3    130.2      10.568         11.12    139.5
## R  215.0  130.0    129.7       8.289         10.16    141.5
```

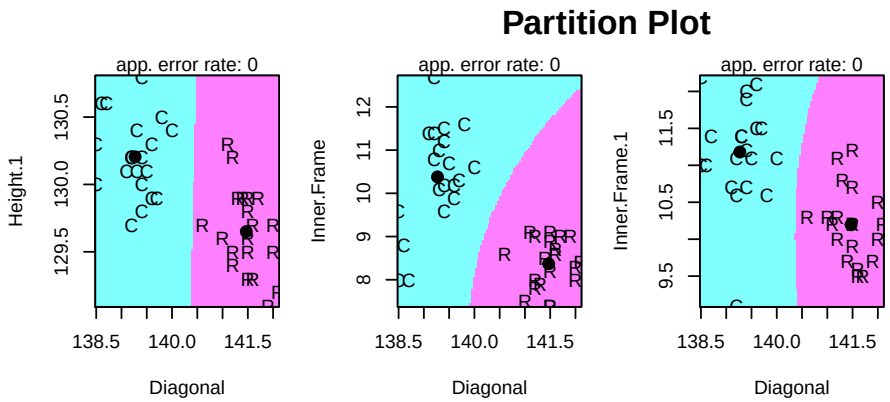
```
notes_data_test$qda_predict<-predict(notes_qda,notes_data_test[,1:6])$class
table(notes_data_test$Group,notes_data_test$qda_predict)

##
##      C  R
##   C 20  0
##   R  0 20
```

In this case, AER=0/40=40.

```
partimat(as.factor(Group) ~ Length+Height+Height.1+Inner.Frame+Inner.Frame.1
              +Diagonal,data=notes_data_test,method="qda")
```


app. error rate: 0.2
Length
Height
app. error rate: 0.125
Length
Height.1
app. error rate: 0.15
Height
Height.1
app. error rate: 0.075
Length
Inner.Frame
app. error rate: 0.1
Height
Inner.Frame
app. error rate: 0.075
Height.1
Inner.Frame
app. error rate: 0.2
Length
Inner.Frame.1
app. error rate: 0.1
Height
Inner.Frame.1
app. error rate: 0.1
Height.1
Inner.Frame.1
app. error rate: 0.075
Inner.Frame
Inner.Frame.1
app. error rate: 0
Length
Diagonal
app. error rate: 0
Height
Diagonal



Prior Probabilities

Suppose that we know that we are much more likely to come across a real note than a counterfeit note. Then we would like to use this to weight the likelihood of each outcome. Suppose that 95% of all notes are real.

```
notes_lda<-lda(Group ~ Length+Height+Height.1+Inner.Frame+Inner.Frame.1
               +Diagonal,data=notes_data_training,prior=c(0.95,0.05))
notes_lda

## Call:
## lda(Group ~ Length + Height + Height.1 + Inner.Frame + Inner.Frame.1 +
```

```
##     Diagonal, data = notes_data_training, prior = c(0.95, 0.05))
##
## Prior probabilities of groups:
##    C    R
## 0.95 0.05
##
## Group means:
##   Length Height Height.1 Inner.Frame Inner.Frame.1 Diagonal
## C  214.8  130.3    130.2      10.568         11.12    139.5
## R  215.0  130.0    129.7       8.289         10.16    141.5
##
## Coefficients of linear discriminants:
##                    LD1
## Length        -0.08273
## Height         0.73717
## Height.1      -0.76727
## Inner.Frame   -1.11703
## Inner.Frame.1 -1.15583
## Diagonal       1.42627
```

```
notes_data_test$lda_predict<-predict(notes_lda,notes_data_test[,1:6])$class
table(notes_data_test$Group,notes_data_test$lda_predict)

##
##      C  R
##   C 20  0
##   R  0 20
```

In this case, AER=0/40=40.

In-class exercise: Wine Data

The wine data set contains data collected from three different cultivars of wine, measuring 13 different variables describing the concentration of chemicals that can be found in the wine as well as other properties of the wine. The variables are:

- Cult: Cultivar
- Alc: Alcohol
- MalAcid: Malic acid
- Ash: Ash
- AshAlk: Alcalinity of ash
- Mag: Magnesium
- TotPhen: Total phenols
- Flav: Flavanoids
- NonFlav: Nonflavanoid phenols
- Proant: Proanthocyanins
- Color: Color intensity
- Hue: Hue
- OD280OD315: OD280/OD315 of diluted wines
- Proline: Proline

Use linear and quadratic discriminant analysis to develop a rule that discriminates between the three cultivars.

In-class exercise: Credit Card Scoring

The data in creditcard.csv contain information obtained from credit card applications. For the purpose of confidentiality, the names and values of the attributes of the data set have been coded. These dat have been obtained from here: `https://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval)`

The variables are as follows:

- A1: Categorical variable (0,1)

- A2: Continuous variable
- A3: Continuous variable
- A4: Categorical variable (1,2,3)
- A5: Categorical variable (1,2,3,4,5,6,7,8,9,10,11,12,13,14)
- A6: Categorical variable (1,2,3,4,5,6,7,8,9)
- A7: Continuous variable
- A8: Categorical variable (0,1)
- A9: Categorical variable (0,1)
- A10: Continuous variable
- A11: Categorical variable (0,1)
- A12: Categorical variable (1,2,3)
- A13: Continuous variable
- A14: Continuous variable
- A15: Prediction variable (1,2)

Use variables A1–A14 to develop a classification rule for predicting A15.

Exercises

Construct linear and quadratic classification rules for

- the bankrupcy data in Table 11.4
- the iris data in Table 11.5
- the admission data in Table 11.6

of Johnson and Wichern. Experiment with different proportions of training and test data set sizes (as a percentage of observations in each group), and determine the ability of the models built on the training data to predict for the test data.