

lab1

Erik Tedhamre

8 December 2018

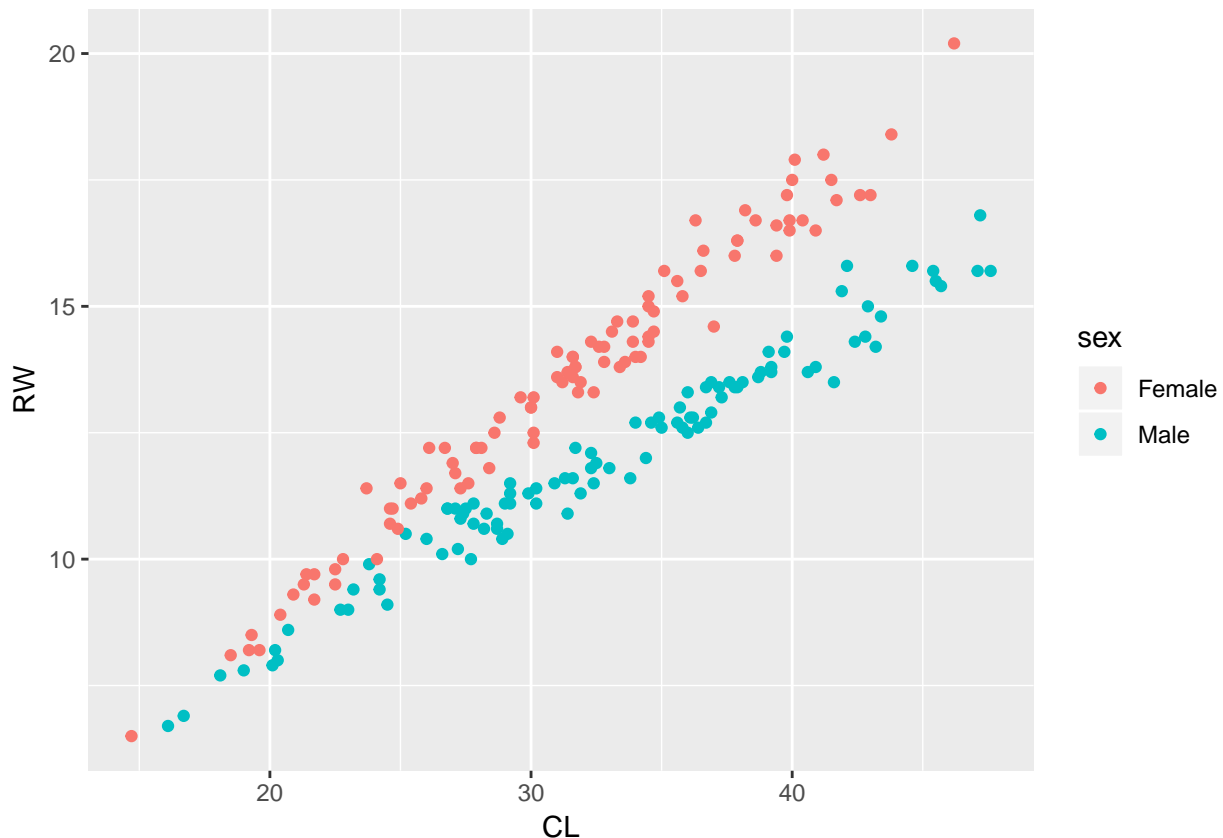
R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Assignment 1.1

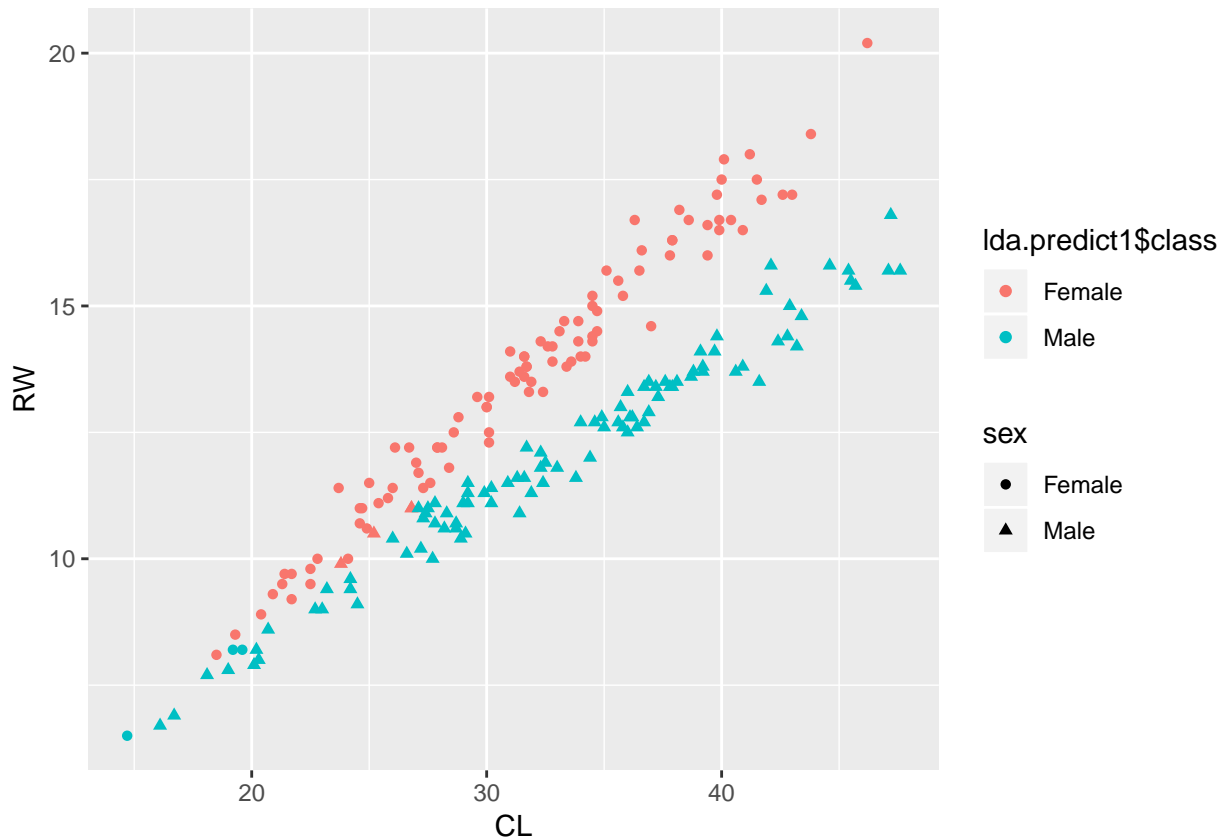
```
library("ggplot2")
library("MASS")
data <- data.frame(read.csv("australian-crabs.csv"))
ggplot(data = data, mapping = aes(CL, RW, color = sex)) + geom_point()
```



plot of carapace length versus rear width where the observations are colored by sex. Looking at the graph the data seems reasonably easy to classify by linear discriminant analysis. Because there seems to be a line between the two sexes.

Assignment 1.2

```
lda.model1 <- lda(sex ~ CL + RW, data = data)
lda.predict1 <- predict(lda.model1, data)
ggplot.0.5 <- ggplot(data = data, mapping = aes(CL, RW, color = lda.predict1$class, shape = sex )) + geom_point()
ggplot.0.5
```

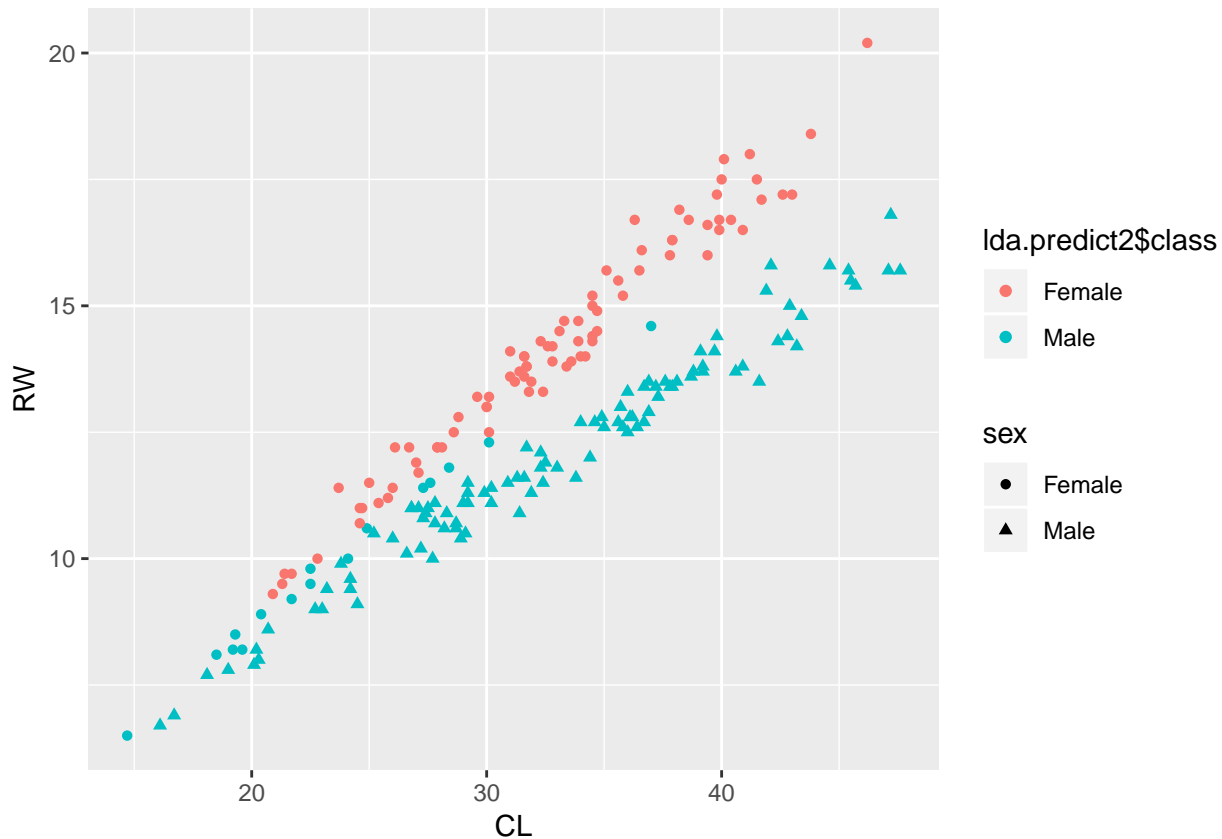


```
mcr.0.5 <- mean(lda.predict1$class != data$sex)
```

The missclassification rate for the linear discriminant analysis is **0.035**. This is pretty reasonable considering we saw on the original graph that there was one area with a bit of an overlap. If this is good enough for actual use is hard to say, it mostly depends on how much we would lose on an incorrect classification.

Assignment 1.3

```
lda.model2 <- lda(sex ~ CL + RW, data = data, prior = c(Female = 0.1, Male = 0.9))
lda.predict2 <- predict(lda.model2, data)
ggplot(data = data, mapping = aes(CL, RW, color = lda.predict2$class, shape = sex )) + geom_point()
```



The number of males increased since we are assuming a weighted distribution. Especially the areas containing both types of observations are now classified as only males instead of both.

```
mcr.0.9 <- mean(lda.predict2$class != data$sex)
```

The missclassification rate for the weighted linear discriminant analysis is **0.08**. ## Assignment 1.4

```
glm.model <- glm(as.factor(sex) ~ CL + RW, family = binomial, data = data)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
glm.predict <- predict(glm.model, data, type = 'response')
```

```
mcr.glm <- mean(glm.predict != data$sex)
```

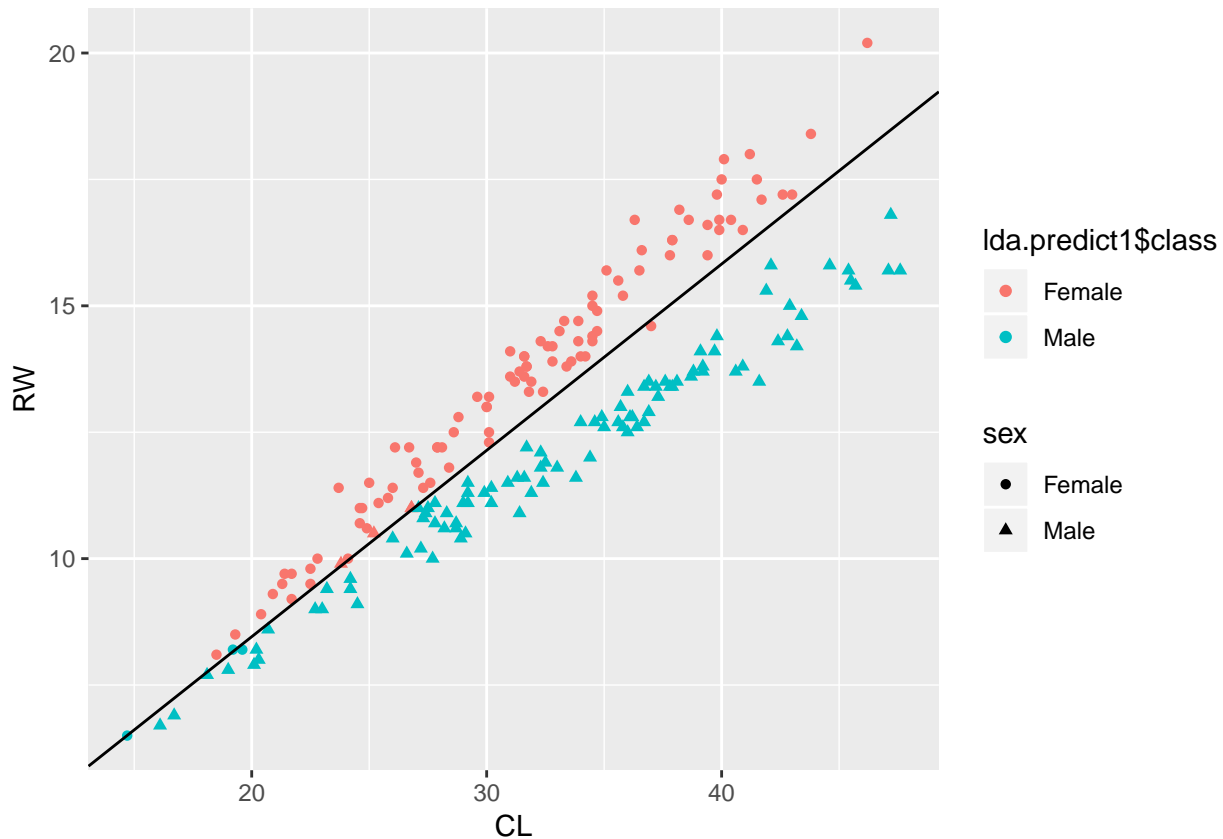
The missclassification rate is **0.035** which is the same as the original linear discriminant analysis.

```
glm.predict.0.5 <- ifelse(glm.predict > 0.5, "Male", "Female")
```

```
glm.slope <- coef(glm.model)[2]/(-coef(glm.model)[3])
```

```
glm.intercept <- coef(glm.model)[1]/(-coef(glm.model)[3])
```

```
ggplot.0.5 + geom_abline(slope = glm.slope, intercept = glm.intercept)
```



The decision line is drawn in the graph.

Assignment 2

Splitting the data into partitions

```
library("e1071")
library("MASS")
library("tree")
library("ggplot2")
setwd("~/TDDE01/lab2")
data.credit <- data.frame(read.csv("creditscoring.csv"))
n=dim(data.credit)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.5))
train=data.credit[id,]
id1=setdiff(1:n, id)
set.seed(12345)
id2=sample(id1, floor(n*0.25))
valid=data.credit[id2,]
id3=setdiff(id1,id2)
test=data.credit[id3,]

tree.fit.dev <- tree(good_bad ~., data = train, split = "deviance")
tree.fit.gini <- tree(good_bad ~., data = train, split = "gini")
```

The models used in calculating the following confusion matrices.

```
pred.dev.train <- predict(tree.fit.dev, newdata = train, type="class")
table(train$good_bad, pred.dev.train)
```

```
##      pred.dev.train
##      bad good
## bad    61   86
## good   20  333
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.