Chapter17 output

```
#### Example: Fisher's iris data
# first fit lm() with formula = continuous variables ~ factor variables
lm.iris <- lm(cbind(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) ~ Species, data = iris)
# fit canonical discrimant function
library(candisc)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

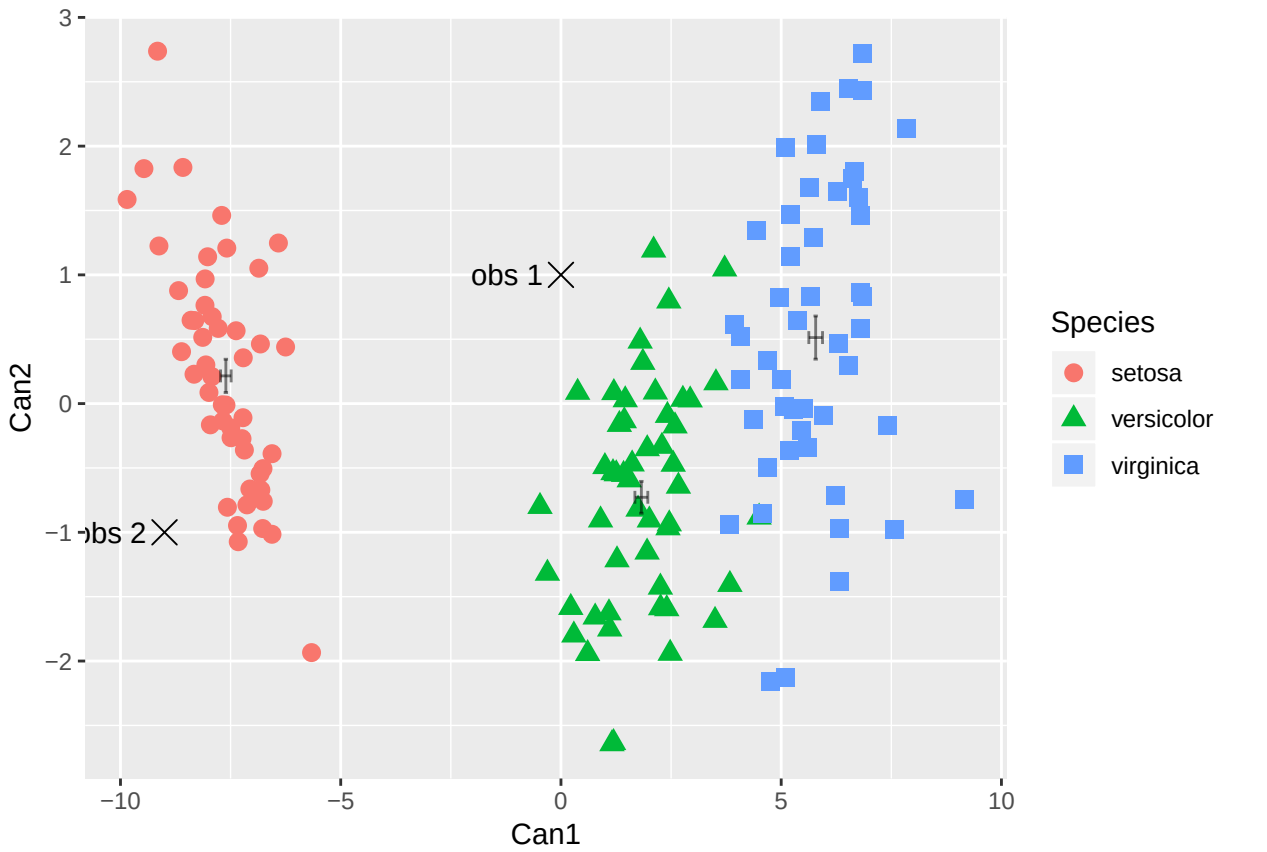
```
## Loading required package: heplots
```

```
##
## Attaching package: 'candisc'
```

```
## The following object is masked from 'package:stats':
##
##     cancor
```

```
can.iris <- candisc(lm.iris)

# Error bars for means of each population
library(plyr)
Can.means <- ddply(can.iris$scores, .(Species), function(x) {
                      Can1 = mean(x$Can1)
                      Can2 = mean(x$Can2)
                      Can1SE = sd(x$Can1)/sqrt(length(x$Can1))
                      Can2SE = sd(x$Can2)/sqrt(length(x$Can2))
                      data.frame(Can1, Can2, Can1SE, Can2SE)
                  })

# two new observations for classification
df.obs <- data.frame(Can1 = c(0, -9), Can2 = c(1, -1), name = c("obs 1", "obs 2"))

library(ggplot2)
p <- ggplot(can.iris$scores, aes(x = Can1, y = Can2))
p <- p + geom_point(aes(shape = Species, colour = Species), size = 3)

# mean of each species with error bars
#p <- p + geom_point(data = Can.means, aes(colour = Species), size = 1)
p <- p + geom_errorbarh(data = Can.means
                      , aes(xmin = Can1 - Can1SE
                          , xmax = Can1 + Can1SE
                          , y = Can2, colour = Species, height = 0.1)
                      , color = "black", alpha = 0.5)
p <- p + geom_errorbar(data = Can.means
                      , aes(ymin = Can2 - Can2SE
                          , ymax = Can2 + Can2SE
                          , x = Can1, colour = Species, width = 0.1)
                      , color = "black", alpha = 0.5)

# a couple example new observations
p <- p + geom_point(data=df.obs, aes(x = Can1, y = Can2), shape = 4, size = 4, colour = "black")
```

```
p <- p + geom_text(data=df.obs, aes(x = Can1, y = Can2, label = name)
                   , colour = "black", vjust = 0.5, hjust = 1.25)
print(p)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch17plot1.jpg")
```

```
## jpeg
##   3
```

```
dev.off()
```

```
## pdf
##   2
```

```
################################################################################
# M-distance
# R code drawing ellipses
library(ellipse)
```

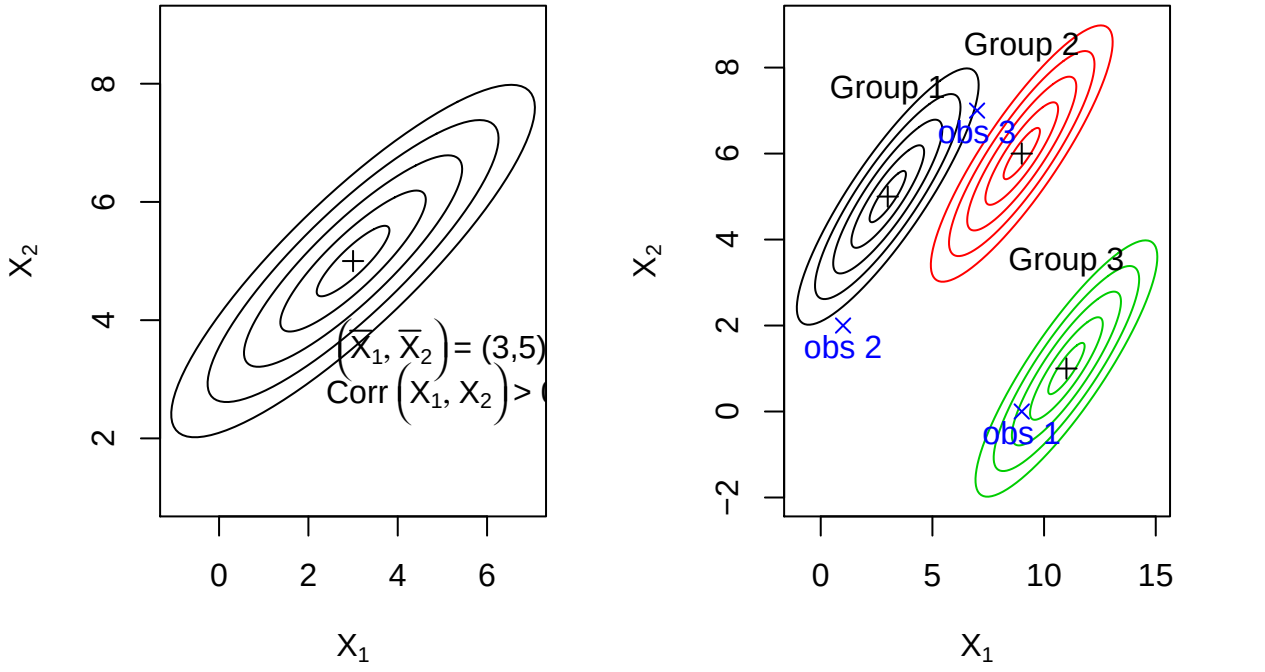
```
##
## Attaching package: 'ellipse'
```

```
## The following object is masked from 'package:car':
##
##     ellipse
```

```
## The following object is masked from 'package:graphics':
##
##     pairs
```

```r
par(mfrow=c(1,2))
xy.loc <- c(3,5);

xy.range <- c(-4, 4);
plot(0, xlim=xy.range+xy.loc[1], ylim=xy.range+xy.loc[2], type='n'
  , xlab=expression(X[1]), ylab=expression(X[2]))

for (i in 1:5) {
  diam <- 0.1*i;
  # start with horizontal ellipse
  ee <- ellipse(matrix(c(1, 0, 0, 1),ncol=2), scale=diam*c(4,1), centre=c(0,0), level=0.95)
  # rotate counterclockwise 35 degrees
  theta <- 35*pi/180; # rotate 35 degrees
  rot.35deg <- matrix(c(cos(theta),sin(theta),-sin(theta),cos(theta)),ncol=2);
  ee.rot <- t( rot.35deg %*% t(ee) + xy.loc);
  points(ee.rot    , type='l') # ellipse
}

points(x=xy.loc[1], y=xy.loc[2], pch=3)
text(x=xy.loc[1]+2, y=xy.loc[2]-1.5, expression(paste(bgroup("(", list( bar(X)[1], bar(X)[2] ), ")"), "
text(x=xy.loc[1]+2, y=xy.loc[2]-2.25, expression(paste("Corr ", bgroup("(", list( X[1], X[2] ), ")"), "


################################################################################
# Classification
# R code drawing ellipses
library(ellipse)

plot(0, xlim=c(-1,15), ylim=c(-2,9), type='n'
  , xlab=expression(X[1]), ylab=expression(X[2]))


for (j in 1:3) {
  if (j==1) xy.loc <- c(3,5);
  if (j==2) xy.loc <- c(9,6);
  if (j==3) xy.loc <- c(11,1);

  for (i in 1:5) {
    diam <- 0.1*i;
    # start with horizontal ellipse
    ee <- ellipse(matrix(c(1, 0, 0, 1),ncol=2), scale=diam*c(4,1), centre=c(0,0), level=0.95)
    # rotate counterclockwise 35 degrees
    theta <- 35*pi/180; # rotate 35 degrees
    rot.35deg <- matrix(c(cos(theta),sin(theta),-sin(theta),cos(theta)),ncol=2);
    ee.rot <- t( rot.35deg %*% t(ee) + xy.loc);
    points(ee.rot    , type='l', col=j) # ellipse
  }

  points(x=xy.loc[1], y=xy.loc[2], pch=3)
  text(x=xy.loc[1], y=xy.loc[2]+2.5, paste("Group ", j, sep="" ) )

}
```

```
obs1 <- c(9,0)
obs2 <- c(1,2)
obs3 <- c(7,7)
points(x=obs1[1], y=obs1[2], pch=4, col=4); text(x=obs1[1], y=obs1[2]-.5, "obs 1", col=4);
points(x=obs2[1], y=obs2[2], pch=4, col=4); text(x=obs2[1], y=obs2[2]-.5, "obs 2", col=4);
points(x=obs3[1], y=obs3[2], pch=4, col=4); text(x=obs3[1], y=obs3[2]-.5, "obs 3", col=4);
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch17plot2.jpg")
```

```
## jpeg
##   3
```

```
dev.off()
```

```
## pdf
##   2
```

```
par(mfrow=c(1,1))

#### Example: Fisher's iris data
# The "iris" dataset is included with R in the library(datasets)
data(iris)
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
head(iris)
```

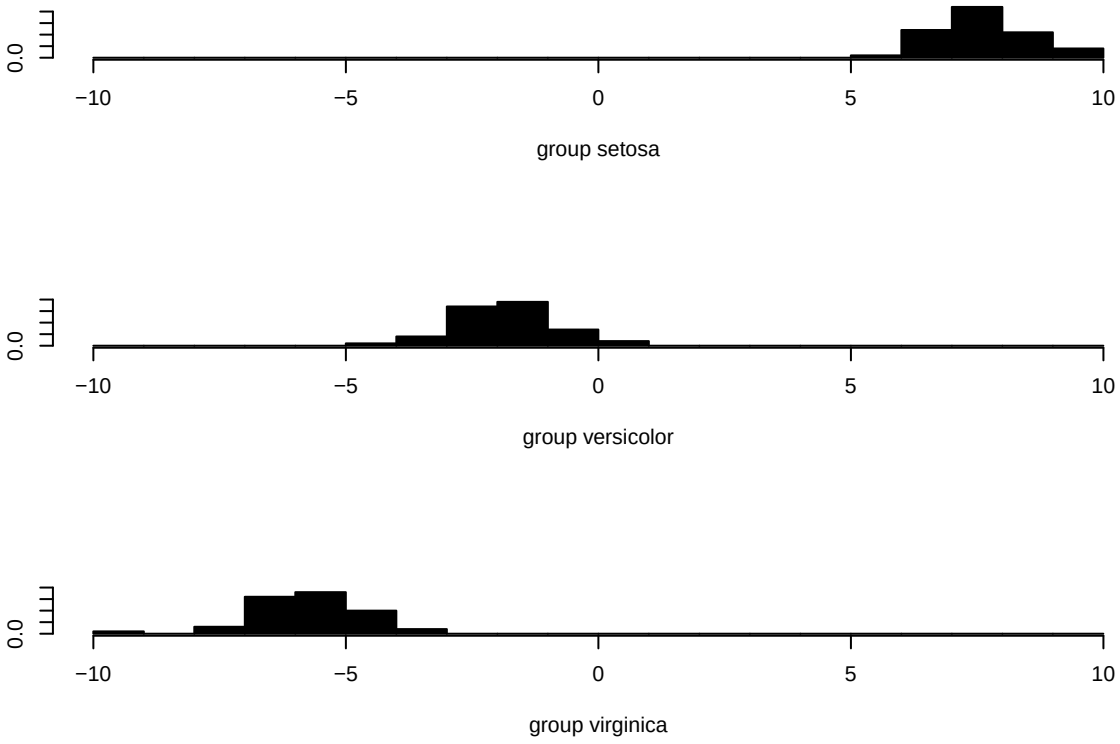
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
```

```
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

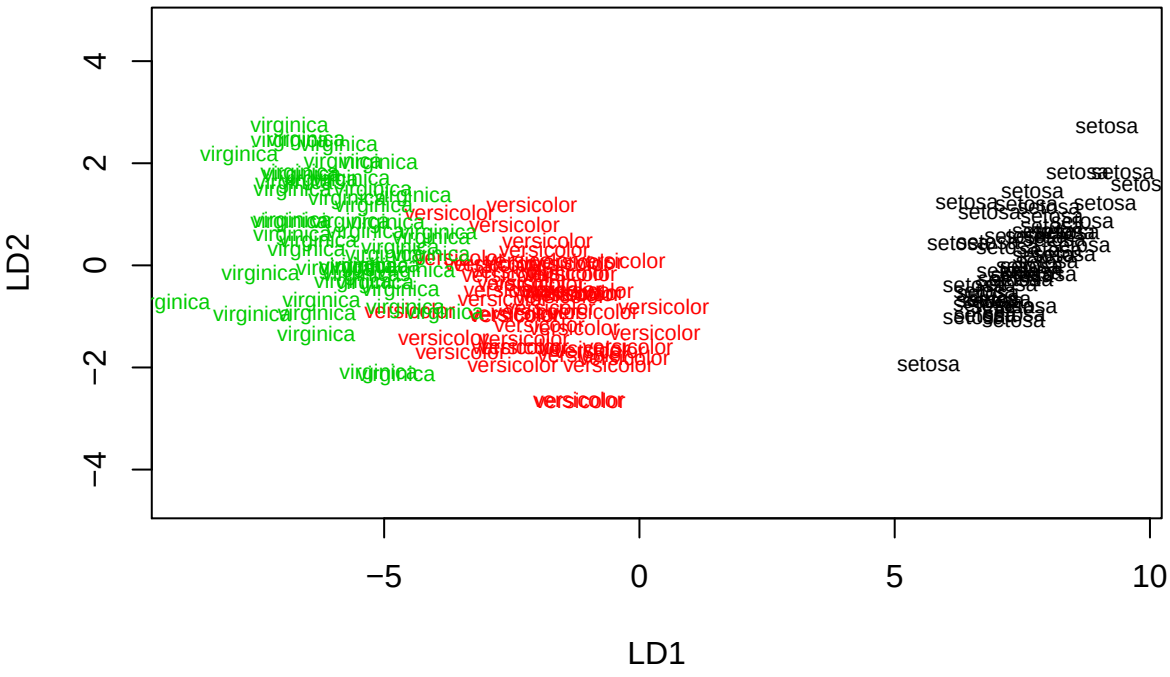
```
####construct classification rules

library(MASS)
lda.iris0 <- lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
                 , data = iris)
lda.iris0
```

```
## Call:
## lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
##     data = iris)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.006       3.428        1.462       0.246
## versicolor        5.936       2.770        4.260       1.326
## virginica         6.588       2.974        5.552       2.026
##
## Coefficients of linear discriminants:
##                     LD1         LD2
## Sepal.Length  0.8293776  0.02410215
## Sepal.Width   1.5344731  2.16452123
## Petal.Length -2.2012117 -0.93192121
## Petal.Width  -2.8104603  2.83918785
##
## Proportion of trace:
##    LD1    LD2
## 0.9912 0.0088
```

```
plot(lda.iris0, dimen = 1, col = as.numeric(iris$Species))
```




```
plot(lda.iris0, dimen = 2, col = as.numeric(iris$Species))
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch17plot9.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
```

```
##   2
```

```
# predict the new data from the original data LDFs
newdata <- data.frame(Sepal.Length=5.8,Sepal.Width=3.1,Petal.Length=3.8,Petal.Width=1.2)
predict(lda.iris0, newdata = newdata)
```

```
## $class
## [1] versicolor
## Levels: setosa versicolor virginica
##
## $posterior
##        setosa versicolor    virginica
## 1 1.04104e-12  0.9999995 4.579081e-07
##
## $x
##           LD1        LD2
## 1 -0.06479338 0.05406058
```

```
# Randomly assign equal train/test by Species strata
library(plyr)
iris <- ddply(iris, .(Species), function(X) {
      ind <- sample.int(nrow(X), size = round(nrow(X)/2))
      sort(ind)
      X$test      <- "train"
      X$test[ind] <- "test"
      X$test <- factor(X$test)
      X$test
      return(X)
      })
summary(iris$test)
```

```
##  test train
##    75    75
```

```
table(iris$Species, iris$test)
```

```
##
##              test train
##   setosa       25    25
##   versicolor   25    25
##   virginica    25    25
```

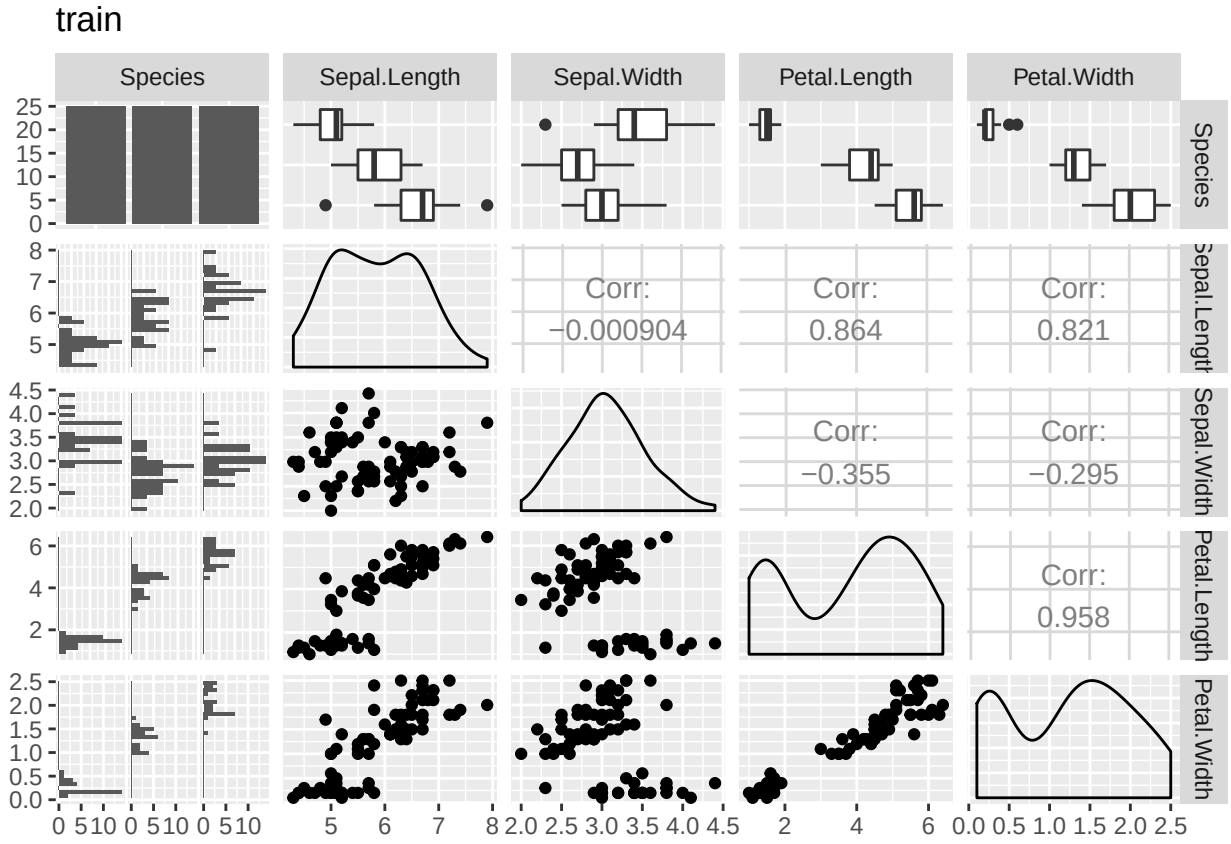
```
## Scatterplot matrix
library(ggplot2)
suppressMessages(suppressWarnings(library(GGally)))
p <- ggpairs(subset(iris, test == "train")[,c(5,1,2,3,4)], colour = "Species", title = "train")
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'colour' are
## being ignored. If these are meant to be aesthetics, submit them using the
## 'mapping' variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```

```
print(p)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch17plot3.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

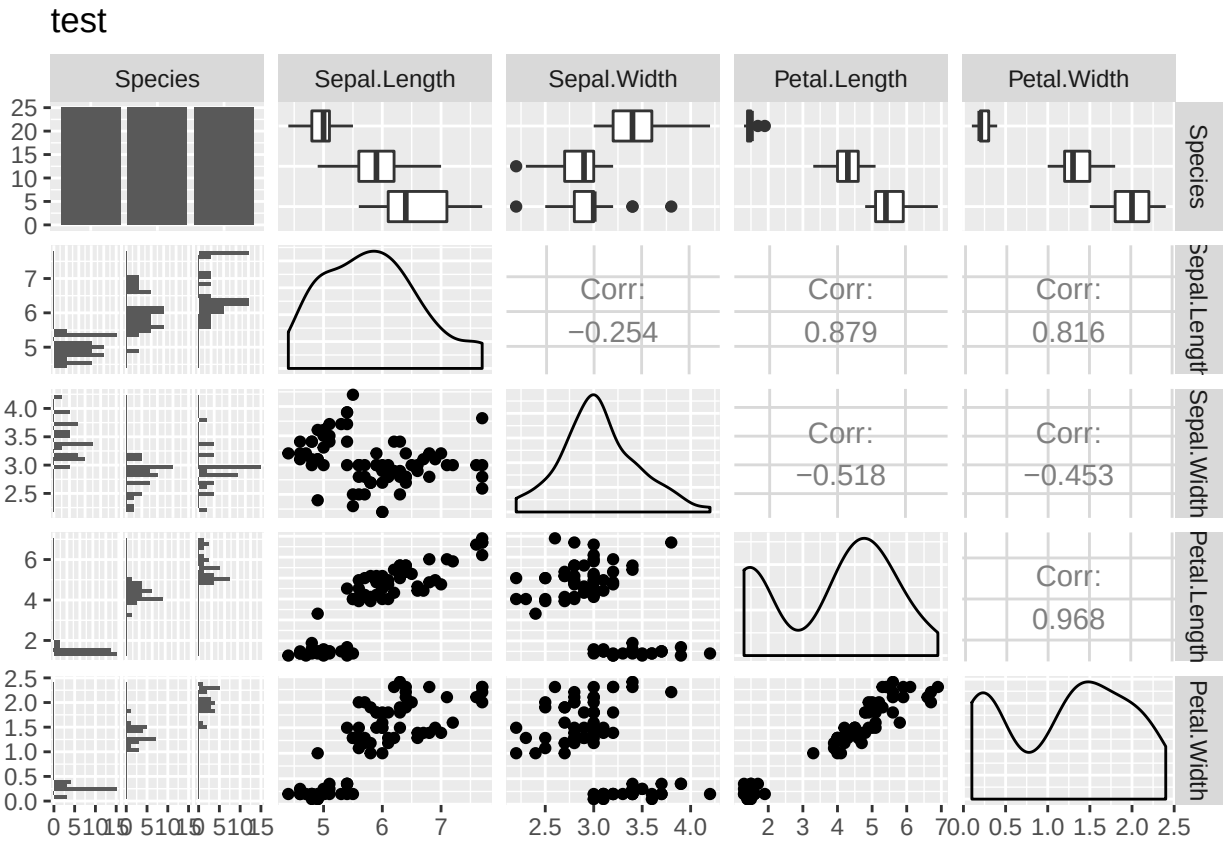
```
## pdf
##   2
```

```
p <- ggpairs(subset(iris, test == "test")[,c(5,1,2,3,4)], colour = "Species", title = "test")
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'colour' are
## being ignored. If these are meant to be aesthetics, submit them using the
## 'mapping' variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```

```
print(p)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



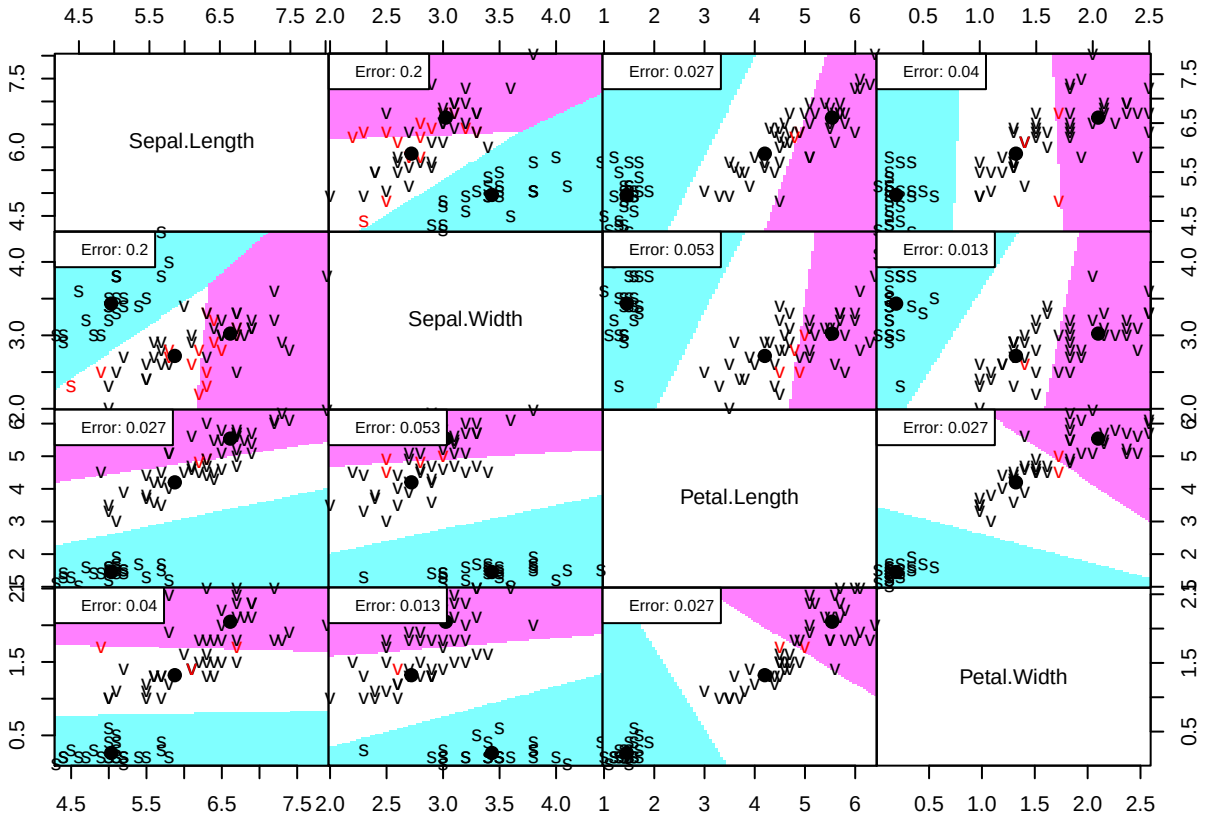

```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch17plot4.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
  # detach package after use so reshape2 works (old reshape (v.1) conflicts)
  #detach("package:GGally", unload=TRUE)
  #detach("package:reshape", unload=TRUE)



# classification of observations based on classification methods
#   (e.g. lda, qda) for every combination of two variables.
library(klaR)
partimat(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
        , data = subset(iris, test == "train")
        , plot.matrix = TRUE)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch17plot5.jpg")
```

```
## jpeg
##   3
```

```
dev.off()
```

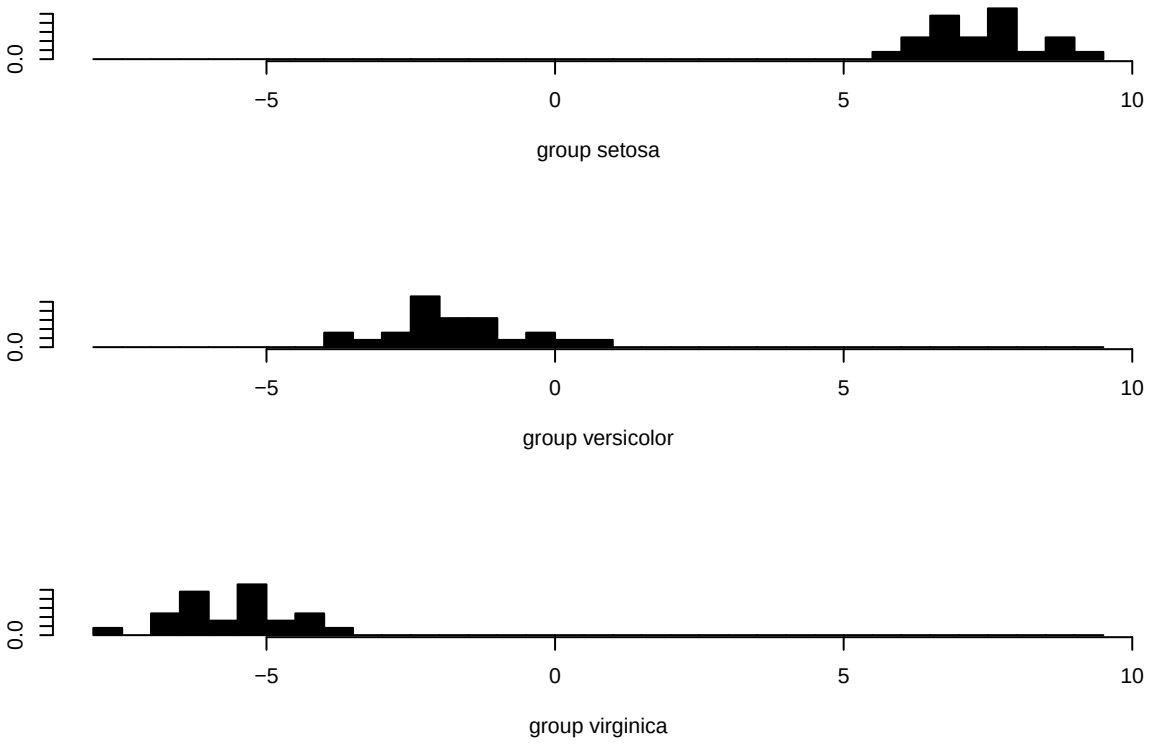
```
## pdf
##   2
```

```
library(MASS)
lda.iris <- lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
            , data = subset(iris, test == "train"))
lda.iris
```

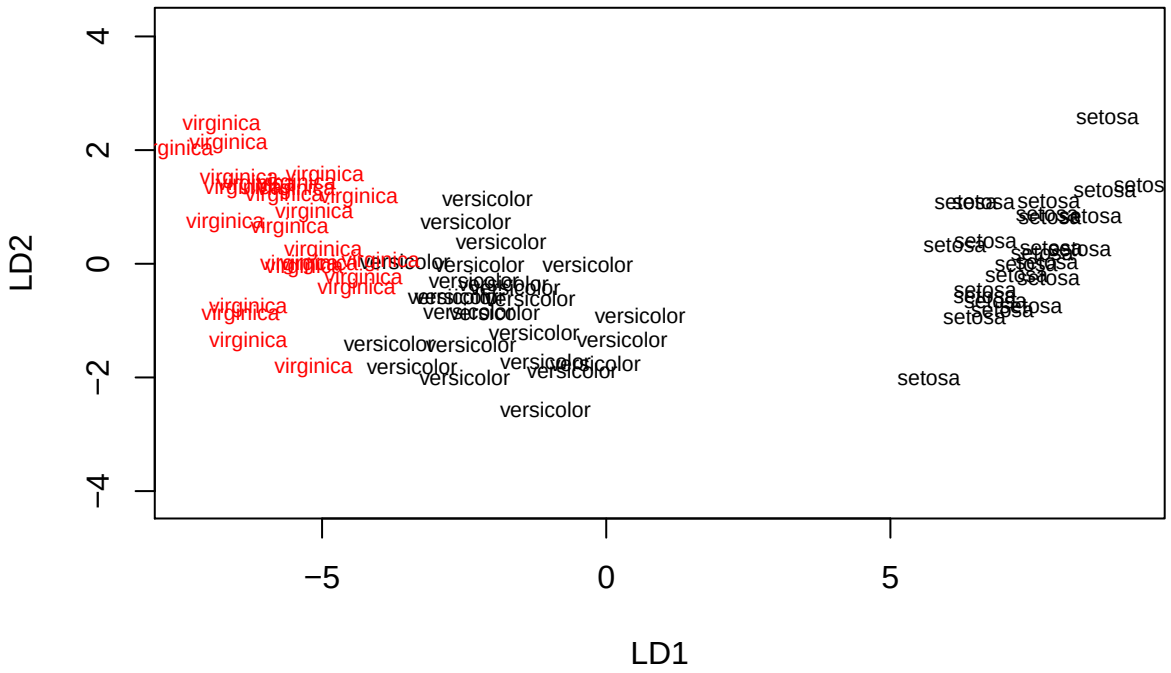
```
## Call:
## lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
##     data = subset(iris, test == "train"))
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.032       3.432        1.452       0.260
## versicolor        5.876       2.720        4.200       1.320
## virginica         6.612       3.024        5.540       2.048
##
## Coefficients of linear discriminants:
```

```
##                     LD1        LD2
## Sepal.Length  0.7027361 -0.1847942
## Sepal.Width   1.4256656  2.2405813
## Petal.Length -2.2536621 -0.6396103
## Petal.Width  -2.4444573  2.3225332
##
## Proportion of trace:
##    LD1    LD2
## 0.9886 0.0114
```

```r
plot(lda.iris, dimen = 1, col = as.numeric(iris$Species))
```




```r
plot(lda.iris, dimen = 2, col = as.numeric(iris$Species))
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch17plot6.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
#pairs(lda.iris, col = as.numeric(iris$Species))



# CV = TRUE does jackknife (leave-one-out) crossvalidation
lda.iris.cv <- lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
                 , data = subset(iris, test == "train"), CV = TRUE)

# Create a table of classification and posterior probabilities for each observation
classify.iris <- data.frame(Species = subset(iris, test == "train")$Species
                   , class = lda.iris.cv$class
                   , error = ""
                   , round(lda.iris.cv$posterior,3))
colnames(classify.iris) <- c("Species", "class", "error"
                         , paste("post", colnames(lda.iris.cv$posterior), sep=""))

# error column
classify.iris$error <- as.character(classify.iris$error)
classify.agree <- as.character(as.numeric(subset(iris, test == "train")$Species)
                               - as.numeric(lda.iris.cv$class))
classify.iris$error[!(classify.agree == 0)] <- classify.agree[!(classify.agree == 0)]
```

```
# print table
classify.iris
```

```
##        Species      class error postsetosa postversicolor postvirginica
## 1       setosa     setosa                1          0.000         0.000
## 2       setosa     setosa                1          0.000         0.000
## 8       setosa     setosa                1          0.000         0.000
## 9       setosa     setosa                1          0.000         0.000
## 14      setosa     setosa                1          0.000         0.000
## 15      setosa     setosa                1          0.000         0.000
## 16      setosa     setosa                1          0.000         0.000
## 19      setosa     setosa                1          0.000         0.000
## 20      setosa     setosa                1          0.000         0.000
## 21      setosa     setosa                1          0.000         0.000
## 23      setosa     setosa                1          0.000         0.000
## 24      setosa     setosa                1          0.000         0.000
## 27      setosa     setosa                1          0.000         0.000
## 28      setosa     setosa                1          0.000         0.000
## 29      setosa     setosa                1          0.000         0.000
## 30      setosa     setosa                1          0.000         0.000
## 33      setosa     setosa                1          0.000         0.000
## 36      setosa     setosa                1          0.000         0.000
## 37      setosa     setosa                1          0.000         0.000
## 39      setosa     setosa                1          0.000         0.000
## 42      setosa     setosa                1          0.000         0.000
## 44      setosa     setosa                1          0.000         0.000
## 45      setosa     setosa                1          0.000         0.000
## 46      setosa     setosa                1          0.000         0.000
## 47      setosa     setosa                1          0.000         0.000
## 52  versicolor versicolor                0          0.998         0.002
## 55  versicolor versicolor                0          0.992         0.008
## 56  versicolor versicolor                0          0.997         0.003
## 57  versicolor versicolor                0          0.965         0.035
## 60  versicolor versicolor                0          0.999         0.001
## 61  versicolor versicolor                0          1.000         0.000
## 64  versicolor versicolor                0          0.988         0.012
## 65  versicolor versicolor                0          1.000         0.000
## 69  versicolor versicolor                0          0.937         0.063
## 73  versicolor versicolor                0          0.722         0.278
## 75  versicolor versicolor                0          1.000         0.000
## 78  versicolor versicolor                0          0.526         0.474
## 80  versicolor versicolor                0          1.000         0.000
## 81  versicolor versicolor                0          1.000         0.000
## 82  versicolor versicolor                0          1.000         0.000
## 86  versicolor versicolor                0          0.981         0.019
## 87  versicolor versicolor                0          0.995         0.005
## 88  versicolor versicolor                0          0.999         0.001
## 91  versicolor versicolor                0          0.999         0.001
## 92  versicolor versicolor                0          0.996         0.004
## 93  versicolor versicolor                0          1.000         0.000
## 94  versicolor versicolor                0          1.000         0.000
## 95  versicolor versicolor                0          0.999         0.001
## 97  versicolor versicolor                0          1.000         0.000
## 99  versicolor versicolor                0          1.000         0.000
```

```
## 101  virginica  virginica              0          0.000         1.000
## 105  virginica  virginica              0          0.000         1.000
## 107  virginica  virginica              0          0.364         0.636
## 108  virginica  virginica              0          0.000         1.000
## 109  virginica  virginica              0          0.000         1.000
## 110  virginica  virginica              0          0.000         1.000
## 111  virginica  virginica              0          0.017         0.983
## 113  virginica  virginica              0          0.000         1.000
## 115  virginica  virginica              0          0.000         1.000
## 117  virginica  virginica              0          0.006         0.994
## 121  virginica  virginica              0          0.000         1.000
## 124  virginica  virginica              0          0.147         0.853
## 125  virginica  virginica              0          0.000         1.000
## 126  virginica  virginica              0          0.002         0.998
## 127  virginica  virginica              0          0.261         0.739
## 131  virginica  virginica              0          0.000         1.000
## 132  virginica  virginica              0          0.000         1.000
## 135  virginica  virginica              0          0.192         0.808
## 138  virginica  virginica              0          0.006         0.994
## 140  virginica  virginica              0          0.001         0.999
## 141  virginica  virginica              0          0.000         1.000
## 142  virginica  virginica              0          0.002         0.998
## 143  virginica  virginica              0          0.002         0.998
## 145  virginica  virginica              0          0.000         1.000
## 146  virginica  virginica              0          0.000         1.000
```

```r
# Assess the accuracy of the prediction
#     row = true Species, col = classified Species
pred.freq <- table(subset(iris, test == "train")$Species, lda.iris.cv$class)
pred.freq
```

```
##
##              setosa versicolor virginica
##   setosa         25          0         0
##   versicolor      0         25         0
##   virginica       0          0        25
```

```r
prop.table(pred.freq, 1) # proportions by row
```

```
##
##              setosa versicolor virginica
##   setosa          1          0         0
##   versicolor      0          1         0
##   virginica       0          0         1
```

```r
# proportion correct for each category
diag(prop.table(pred.freq, 1))
```

```
##     setosa versicolor  virginica
##          1          1          1
```

```r
# total proportion correct
sum(diag(prop.table(pred.freq)))
```

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

```r
# total error rate
1 - sum(diag(prop.table(pred.freq)))
```

```
## [1] 0
```

```r
# predict the test data from the training data LDFs
pred.iris <- predict(lda.iris, newdata = subset(iris, test == "test"))

# Create a table of classification and posterior probabilities for each observation
classify.iris <- data.frame(Species = subset(iris, test == "test")$Species
                  , class = pred.iris$class
                  , error = ""
                  , round(pred.iris$posterior,3))
colnames(classify.iris) <- c("Species", "class", "error"
                        , paste("P", colnames(lda.iris.cv$posterior), sep=""))

# error column
classify.iris$error <- as.character(classify.iris$error)
classify.agree <- as.character(as.numeric(subset(iris, test == "test")$Species)
                            - as.numeric(pred.iris$class))
classify.iris$error[!(classify.agree == 0)] <- classify.agree[!(classify.agree == 0)]

# print table
classify.iris
```

```
##        Species      class error Psetosa Pversicolor Pvirginica
## 3       setosa     setosa             1       0.000      0.000
## 4       setosa     setosa             1       0.000      0.000
## 5       setosa     setosa             1       0.000      0.000
## 6       setosa     setosa             1       0.000      0.000
## 7       setosa     setosa             1       0.000      0.000
## 10      setosa     setosa             1       0.000      0.000
## 11      setosa     setosa             1       0.000      0.000
## 12      setosa     setosa             1       0.000      0.000
## 13      setosa     setosa             1       0.000      0.000
## 17      setosa     setosa             1       0.000      0.000
## 18      setosa     setosa             1       0.000      0.000
## 22      setosa     setosa             1       0.000      0.000
## 25      setosa     setosa             1       0.000      0.000
## 26      setosa     setosa             1       0.000      0.000
## 31      setosa     setosa             1       0.000      0.000
## 32      setosa     setosa             1       0.000      0.000
## 34      setosa     setosa             1       0.000      0.000
## 35      setosa     setosa             1       0.000      0.000
## 38      setosa     setosa             1       0.000      0.000
## 40      setosa     setosa             1       0.000      0.000
## 41      setosa     setosa             1       0.000      0.000
## 43      setosa     setosa             1       0.000      0.000
## 48      setosa     setosa             1       0.000      0.000
## 49      setosa     setosa             1       0.000      0.000
## 50      setosa     setosa             1       0.000      0.000
## 51  versicolor versicolor             0       1.000      0.000
## 53  versicolor versicolor             0       0.990      0.010
## 54  versicolor versicolor             0       1.000      0.000
## 58  versicolor versicolor             0       1.000      0.000
```

```
## 59  versicolor versicolor            0       1.000      0.000
## 62  versicolor versicolor            0       0.999      0.001
## 63  versicolor versicolor            0       1.000      0.000
## 66  versicolor versicolor            0       1.000      0.000
## 67  versicolor versicolor            0       0.974      0.026
## 68  versicolor versicolor            0       1.000      0.000
## 70  versicolor versicolor            0       1.000      0.000
## 71  versicolor  virginica    -1      0       0.241      0.759
## 72  versicolor versicolor            0       1.000      0.000
## 74  versicolor versicolor            0       0.999      0.001
## 76  versicolor versicolor            0       1.000      0.000
## 77  versicolor versicolor            0       0.996      0.004
## 79  versicolor versicolor            0       0.990      0.010
## 83  versicolor versicolor            0       1.000      0.000
## 84  versicolor  virginica    -1      0       0.134      0.866
## 85  versicolor versicolor            0       0.953      0.047
## 89  versicolor versicolor            0       1.000      0.000
## 90  versicolor versicolor            0       1.000      0.000
## 96  versicolor versicolor            0       1.000      0.000
## 98  versicolor versicolor            0       1.000      0.000
## 100 versicolor versicolor            0       1.000      0.000
## 102  virginica  virginica            0       0.002      0.998
## 103  virginica  virginica            0       0.000      1.000
## 104  virginica  virginica            0       0.001      0.999
## 106  virginica  virginica            0       0.000      1.000
## 112  virginica  virginica            0       0.002      0.998
## 114  virginica  virginica            0       0.001      0.999
## 116  virginica  virginica            0       0.000      1.000
## 118  virginica  virginica            0       0.000      1.000
## 119  virginica  virginica            0       0.000      1.000
## 120  virginica  virginica            0       0.264      0.736
## 122  virginica  virginica            0       0.002      0.998
## 123  virginica  virginica            0       0.000      1.000
## 128  virginica  virginica            0       0.140      0.860
## 129  virginica  virginica            0       0.000      1.000
## 130  virginica  virginica            0       0.045      0.955
## 133  virginica  virginica            0       0.000      1.000
## 134  virginica versicolor     1      0       0.627      0.373
## 136  virginica  virginica            0       0.000      1.000
## 137  virginica  virginica            0       0.000      1.000
## 139  virginica  virginica            0       0.209      0.791
## 144  virginica  virginica            0       0.000      1.000
## 147  virginica  virginica            0       0.011      0.989
## 148  virginica  virginica            0       0.004      0.996
## 149  virginica  virginica            0       0.000      1.000
## 150  virginica  virginica            0       0.019      0.981
```

```
# Assess the accuracy of the prediction
#     row = true Species, col = classified Species
pred.freq <- table(subset(iris, test == "test")$Species, pred.iris$class)
pred.freq
```

```
##
##              setosa versicolor virginica
##   setosa         25          0         0
```

```
##   versicolor      0         23         2
##   virginica       0          1        24
```

```
prop.table(pred.freq, 1) # proportions by row
```

```
##
##              setosa versicolor virginica
##   setosa       1.00       0.00      0.00
##   versicolor   0.00       0.92      0.08
##   virginica    0.00       0.04      0.96
```

```
# proportion correct for each category
diag(prop.table(pred.freq, 1))
```

```
##     setosa versicolor  virginica
##       1.00       0.92       0.96
```

```
# total proportion correct
sum(diag(prop.table(pred.freq)))
```

```
## [1] 0.96
```

```
# total error rate
1 - sum(diag(prop.table(pred.freq)))
```

```
## [1] 0.04
```

```
library(klaR)
# start with full model and do stepwise (direction = "backward")
step.iris.b <- stepclass(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
                     , data = iris
                     , method = "lda"
                     , improvement = 0.01 # stop criterion: improvement less than 1%
                                          # default of 5% is too coarse
                     , direction = "backward")
```

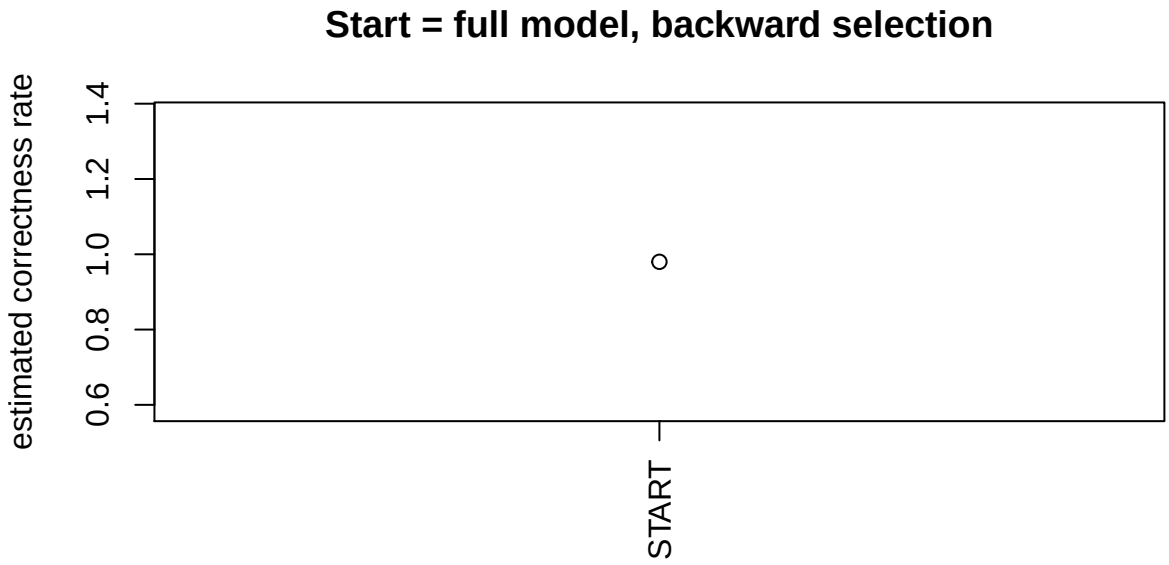
```
##  `stepwise classification', using 10-fold cross-validated correctness rate of method lda'.
```

```
## 150 observations of 4 variables in 3 classes; direction: backward
```

```
## stop criterion: improvement less than 1%.
```

```
## correctness rate: 0.98;  starting variables (4): Sepal.Length, Sepal.Width, Petal.Length, Petal.Widt
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000       0.226
```

```
plot(step.iris.b, main = "Start = full model, backward selection")
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch17plot7.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
step.iris.b$formula
```

```
## Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
## <environment: 0xb13e3a0>
```

```
lda.iris.step <- lda(step.iris.b$formula
                     , data = iris)
lda.iris.step
```

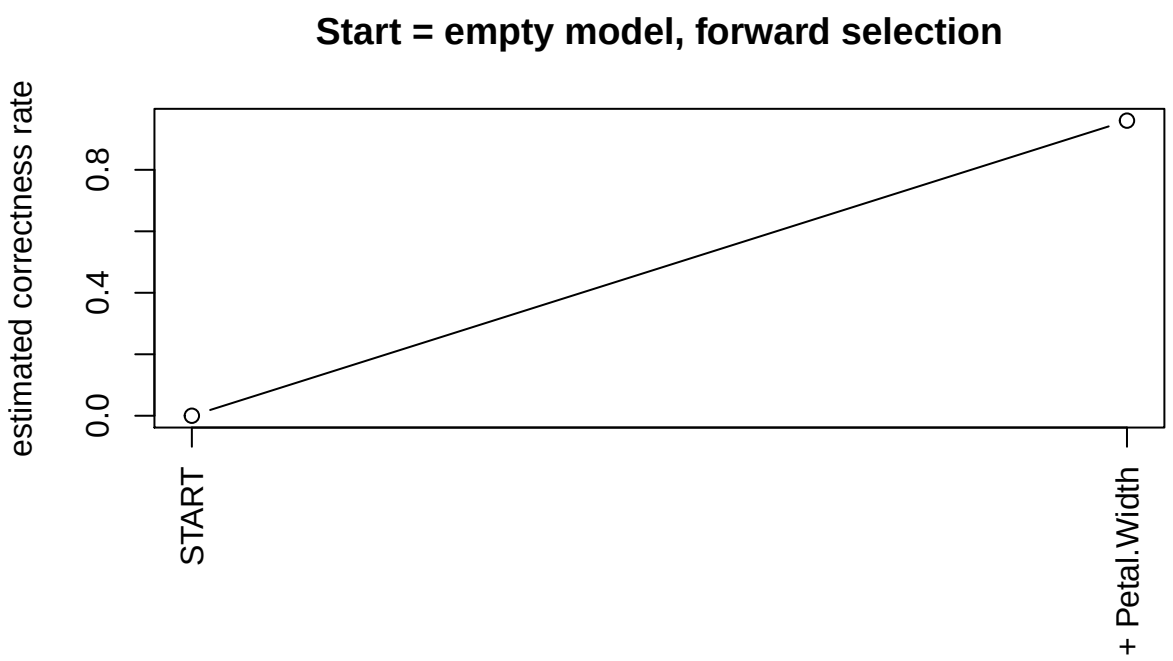
```
## Call:
## lda(step.iris.b$formula, data = iris)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.006       3.428        1.462       0.246
## versicolor        5.936       2.770        4.260       1.326
## virginica         6.588       2.974        5.552       2.026
##
## Coefficients of linear discriminants:
##                     LD1         LD2
## Sepal.Length  0.8293776  0.02410215
## Sepal.Width   1.5344731  2.16452123
## Petal.Length -2.2012117 -0.93192121
## Petal.Width  -2.8104603  2.83918785
##
```

```
## Proportion of trace:
##    LD1    LD2
## 0.9912 0.0088
```

```
# start with empty model and do stepwise (direction = "both")
step.iris.f <- stepclass(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
                     , data = iris
                     , method = "lda"
                     , improvement = 0.01 # stop criterion: improvement less than 1%
                                          # default of 5% is too coarse
                     , direction = "forward")
```

```
##  `stepwise classification', using 10-fold cross-validated correctness rate of method lda'.
## 150 observations of 4 variables in 3 classes; direction: forward
## stop criterion: improvement less than 1%.
## correctness rate: 0.96;  in: "Petal.Width";  variables (1): Petal.Width
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000       0.183
```

```
plot(step.iris.f, main = "Start = empty model, forward selection")
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch17plot8.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
step.iris.f$formula
```

```
## Species ~ Petal.Width
```

```
## <environment: 0x5ebe920>
```