Stat 428/528: Advanced Data Analysis 2

Chapters 17: Classification

Instructor: Yan Lu

Classification analysis

You might have several groups already defined and want to classify a new observation.

- If you have a PCA, then you could determine the linear combinations of variables corresponding to PC1 and PC2 that was determined from an original set of data.
  ——Then you could use those linear combinations on a new data point (even if it didnt contribute to the calculation of the PCs) and see where it fits on plot of PC2 versus PC1.
- If clusters have been defined from the original data, you might want to find which cluster the new observation is closest to.

Examples

- classifying a tumor as benign or malignant based on a medical image
- making a diagnosis (medical or psychiatric) on the basis of a set of symptoms (this is more open-ended than benign versus malignant)
- classifying a fossil bone as belonging to a male or female, adult or juvenille
- classifying student applicants as likely to complete college or drop out
- finding the best fit for an applicant for a college major, or for a job within the military
- determining disputed authorship (Hamilton versus Madison for the Federalist Papers or Shakespeares plays
- identifying speech patterns for automated voice recognition systems

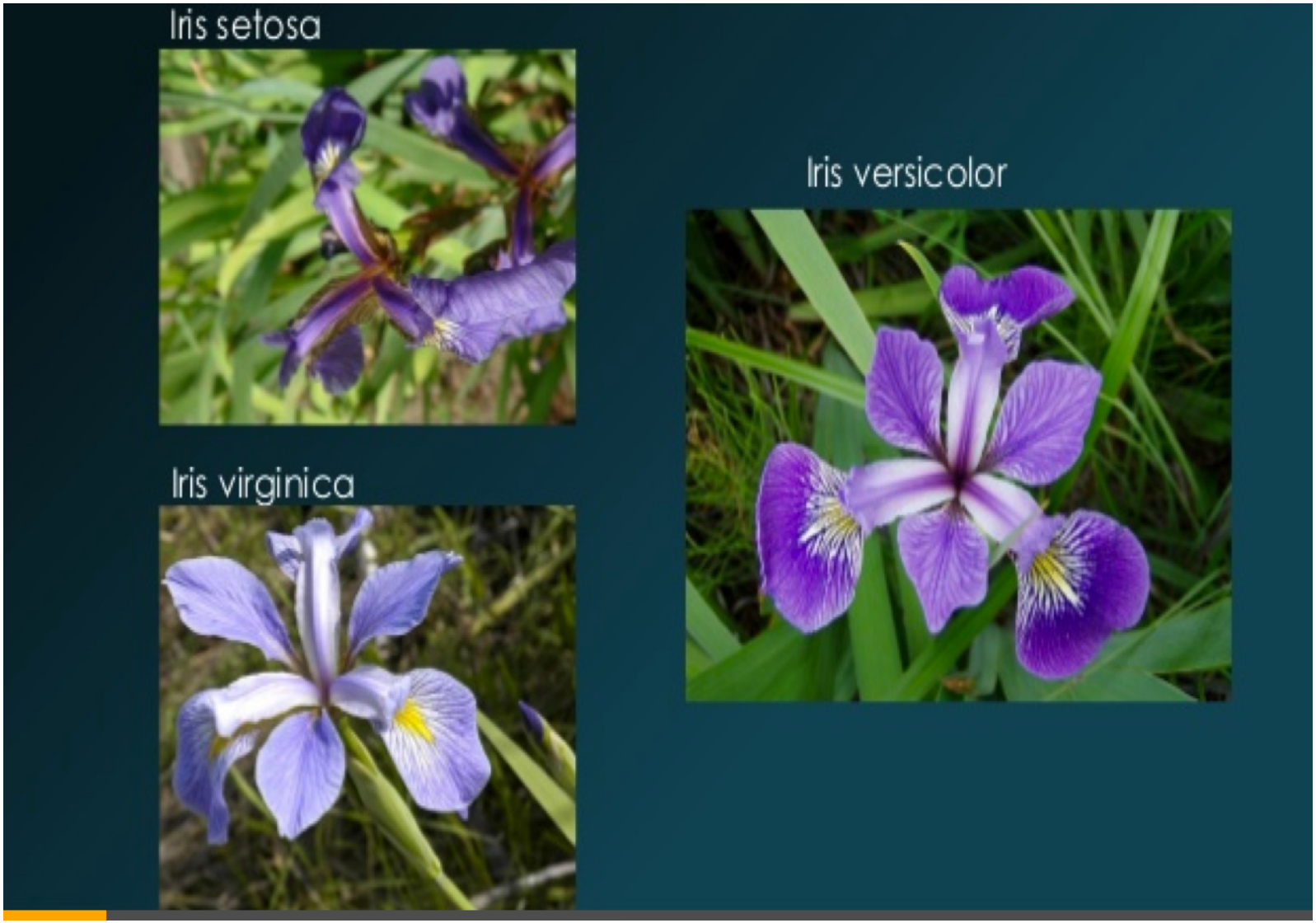
Example: Fisher's iris data

A famous data set used to illustrate classification is Fishers Iris data from 1936. There are three species of iris with 50 observations each. The species are:

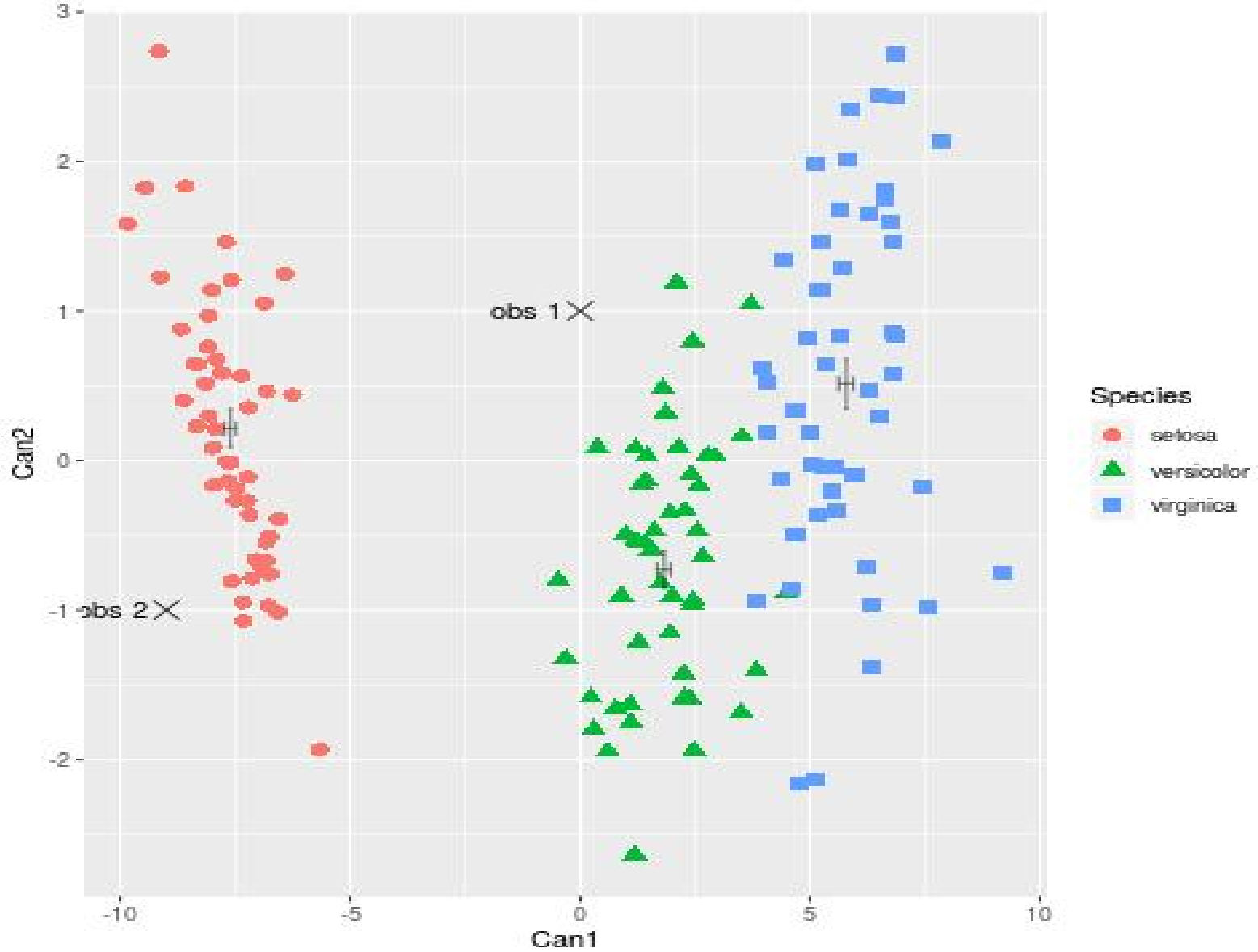
- Iris setosa
- Iris versicolour
- Iris virginica

The variables are (in cm) 1. sepal length
2. sepal width
3. petal length
4. petal width

Figure: Three types of Iris flowers



```
> 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
```



Obs 1 is classified as Versicolor and Obs 2 is classified as Setosa.

Classification using Mahalanobis distance

Suppose p features $X_1, X_2, \cdots, X_p$ are used to discriminate among the k groups.

- Group sizes are $n_1, n_2, \cdots, n_k$ with a total sample size of $n = n_1 + n_2 + \cdots + n_k$.
- Let $\bar{\mathbf{X}}_i = (\bar{X}_{i1}, \bar{X}_{i2}, \cdots, \bar{X}_{ip})'$ be the vector of mean responses for the ith group $i = 1, 2, \cdots, k$
- let $\mathbf{S}_i$ be the p-by-p variance-covariance matrix for the i th group.
- The pooled variance-covariance matrix is given by

$$\mathbf{S} = \frac{(n_1 - 1)\mathbf{S}_1 + (n_2 - 1)\mathbf{S}_2 + \cdots + (n_k - 1)\mathbf{S}_k}{n - k}$$

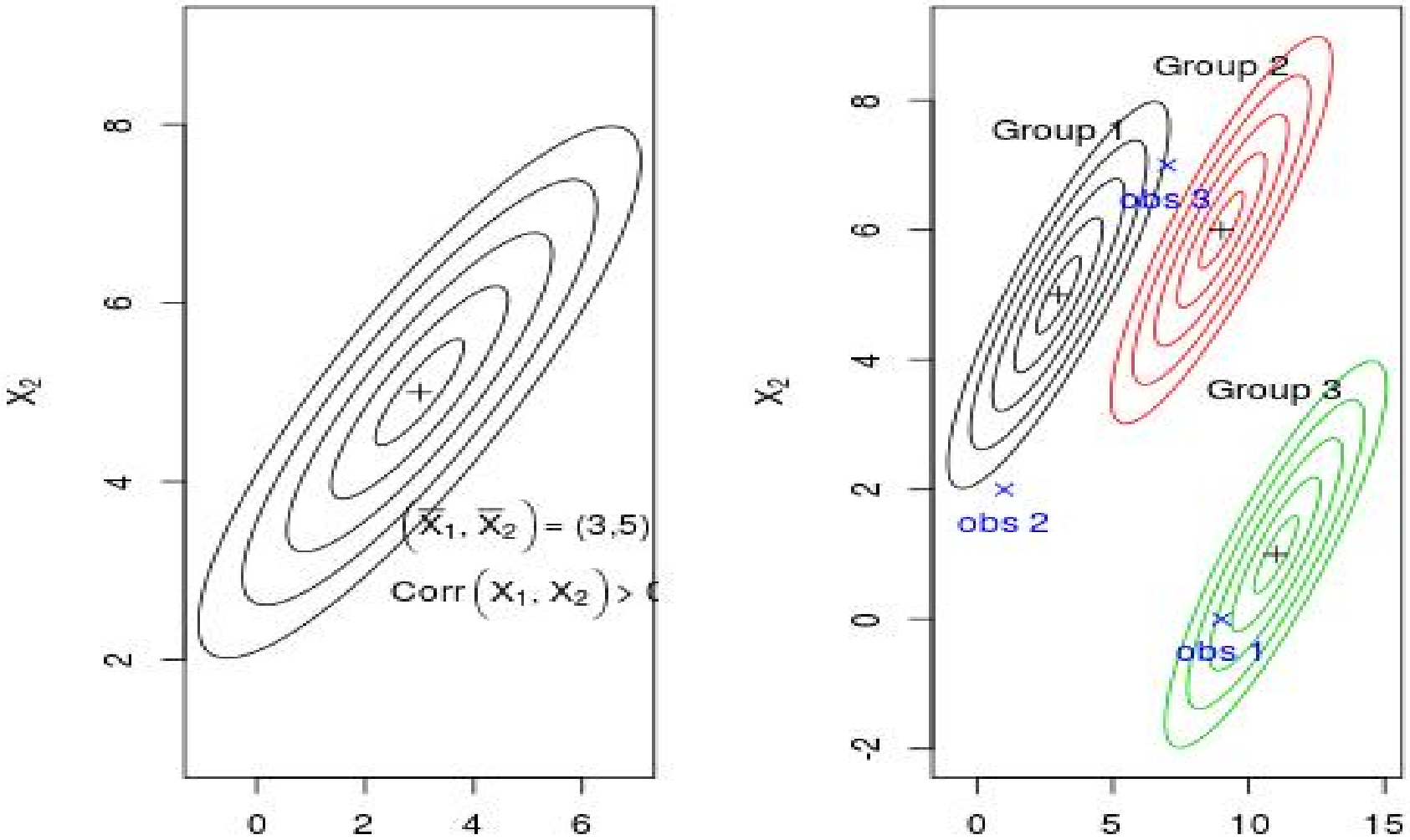
M-distance

The M -distance from an observation $\mathbf{X}$ to (the center of) the i th sample is

$$D_i^2(\mathbf{X}) = (\mathbf{X} - \bar{\mathbf{X}}_i)'\mathbf{S}^{-1}(\mathbf{X} - \bar{\mathbf{X}}_i),$$

- Note that if $\mathbf{S}$ is the identity matrix, then this is the Euclidean distance.
- Given the M -distance from $\mathbf{X}$ to each sample, classify $\mathbf{X}$ into the group which has the minimum M -distance.

Figure: How classification works, three groups and two features



- Observations 1 is closest in M -distance to the center of group 3. Thus, classify observation 1 into group 3.
- Observation 2 is closest to group 1. Thus, classify observation 2 into group 1.
- Observation 3 is closest to the center of group 2 in terms of the standard Euclidean (walking) distance.
  —— However, observation 3 is more similar to data in group 1 than it is to either of the other groups.
  ——The M -distance from observation 3 to group 1 is substantially smaller than the M -distances to either group 2 or 3.
  ———Thus, you would classify observation 3 into group 1.

The M -distance from the i th group to the j th group is the M -distance between the centers of the groups:

$$D^2(i,j) = D^2(j,i) = (\bar{\mathbf{X}}_i - \bar{\mathbf{X}}_j)'\mathbf{S}^{-1}(\bar{\mathbf{X}}_i - \bar{\mathbf{X}}_j).$$

- Larger values suggest relatively better potential for discrimination between groups.
- In the plot above, $D^2(1,2) < D^2(1,3)$ which implies that it should be easier to distinguish between groups 1 and 3 than groups 1 and 2.

Evaluating the Accuracy of a Classification Rule

The misclassification rate, or the expected proportion of misclassified observations, is a good measurement to evaluate a classification rule.

- Better rules have smaller misclassification rates, but there is no universal cutoff for what is considered good in a given problem.
- You should judge a classification rule relative to the current standards in your field for "good classification.

Table: Classification table or Confusion matrix

Actual group	Number of obns	Predicted group 1	Predicted group 2
1	n_1	n_{11}	n_{12}
2	n_2	n_{21}	n_{22}

Misclassification rate $= (n_{12} + n_{21})/(n_1 + n_2)$

Resubstitution

- evaluates the misclassification rate using the data from which the classification rule is constructed.
  ——The resubstitution estimate of the error rate is optimistic (too small).
  ——A greater percentage of misclassifications is expected when the rule is used on new data, or on data from which the rule is not constructed.

Cross-validation is a better way to estimate the misclassification rate.

- In many statistical packages, you can implement cross-validation by randomly splitting the data into a training or calibration set from which the classification rule is constructed.
- The remaining data, called the test data set, is used with the classification rule to estimate the error rate.
- This process is often repeated, say 10 times, and the error rate estimated to be the average of the error rates from the individual splits. With repeated random splitting
- it is common to use 10% of each split as the test data set (a 10-fold cross-validation).

Jackknife Another form of cross-validation uses a jackknife method where single cases are held out of the data (an n-fold), then classified after constructing the classification rule from the remaining data. The process is repeated for each case, giving an estimated misclassification rate as the proportion of cases misclassified.

- The jackknife method is necessary with small sized data sets so single observations dont greatly bias the classification.

Use real testing data You can also classify observations with unknown group membership, by treating the observations to be classified as a test data set.

Example: Fishers Iris Data cross-validation

- The 150 observations were randomly rearranged and separated into two batches of 75.
  —- assigned a label “test, whereas the rest are “train
- The 75 observations in the calibration set were used to develop a classification rule.
- This rule was applied to the remaining 75 flowers, which form the test data set.
- There is no general rule about the relative sizes of the test data and the training data. Many researchers use a 50-50 split.
- Combine the two data sets at the end of the cross-validation to create the actual rule for classifying future data.

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:
```

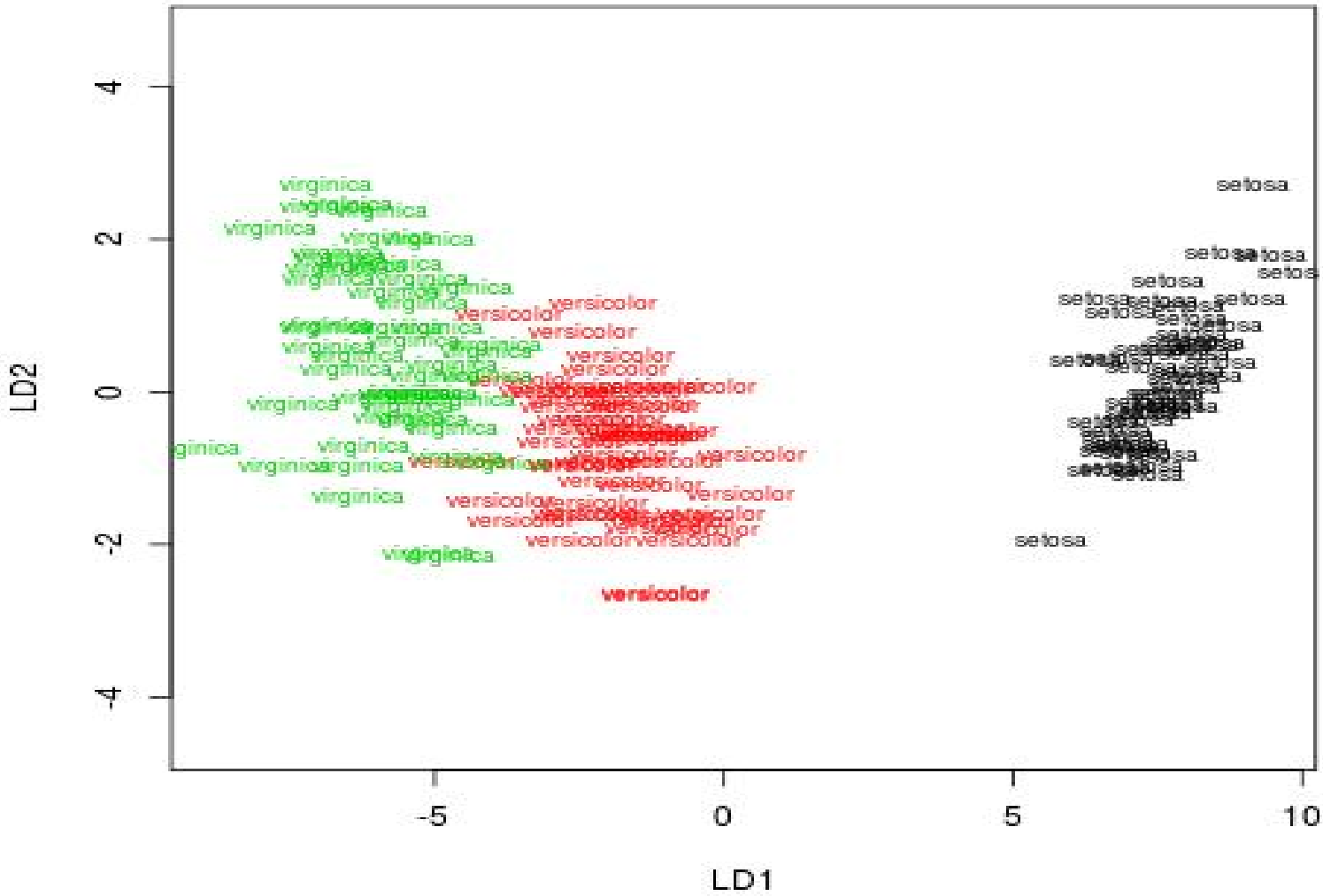
```
            Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa             5.006       3.428        1.462       0.246
versicolor         5.936       2.770        4.260       1.326
virginica          6.588       2.974        5.552       2.026

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
```

- Prior probabilities of groups: the proportion of observations in each group. For example, there are 33.3% of the observations in the setosa group etc.
- Group means: group center of gravity. Shows the mean of each variable in each group.
- Coefficients of linear discriminants: Shows the linear combination of predictor variables that are used to form the LDA decision rule.
  The linear discriminant functions that best classify the Species using Iris data are
  LD1 = 0.829 sepalL + 1.534 sepalW - 2.201 petalL -2.810 petalW
  LD2 = 0.024 sepalL + 2.165 sepalW -0.932 petalL + 2.839 petalW.
- Proportion of trace
  The first linear discriminant LD1 explains more than 99% of the between-group variance in the iris dataset.

The plots of the lda object shows the data on the LD scale.

```
plot(lda.iris0, dimen = 1, col = as.numeric(iris$Species))
plot(lda.iris0, dimen = 2, col = as.numeric(iris$Species))
```

Figure: The plots of the lda object shows the data on the LD scale., Iris data



Predict new data from Iris 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
```

- class: predicted classes of observations.
  ———The new flower with
  Sepal.Length=5.8,Sepal.Width=3.1,
  Petal.Length=3.8,Petal.Width=1.2
  is classified as Versicolor.
- posterior: is a matrix whose columns are the groups, rows are the individuals and values are the posterior probability that the corresponding observation belongs to the groups.
  ———-versicolor is with posterior probability 0.9999
- x: contains the linear discriminants

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

```
> summary(iris$test)
 test train
   75    75
> table(iris$Species, iris$test)

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

Figure: Scatterplot for train data, Iris

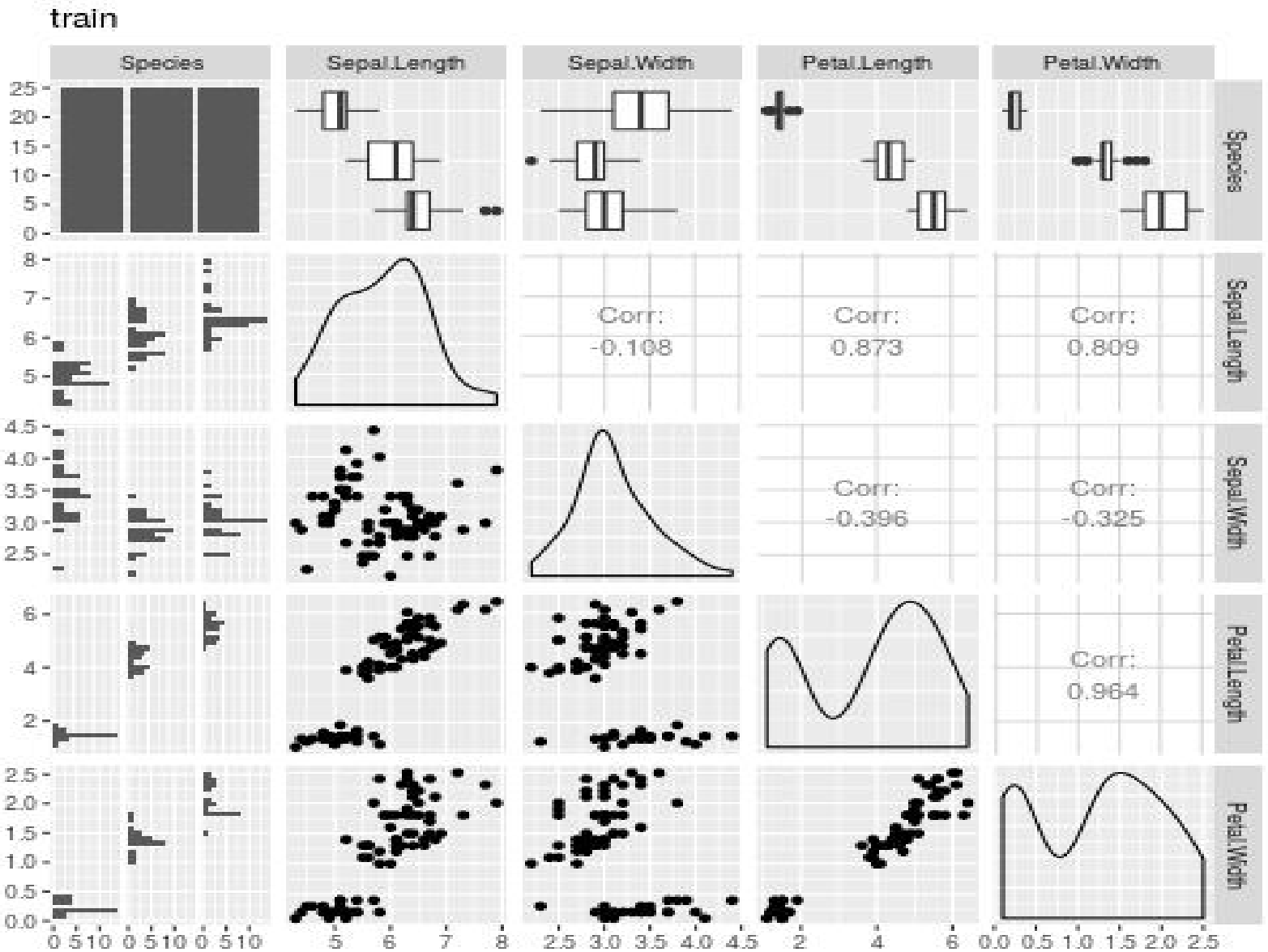
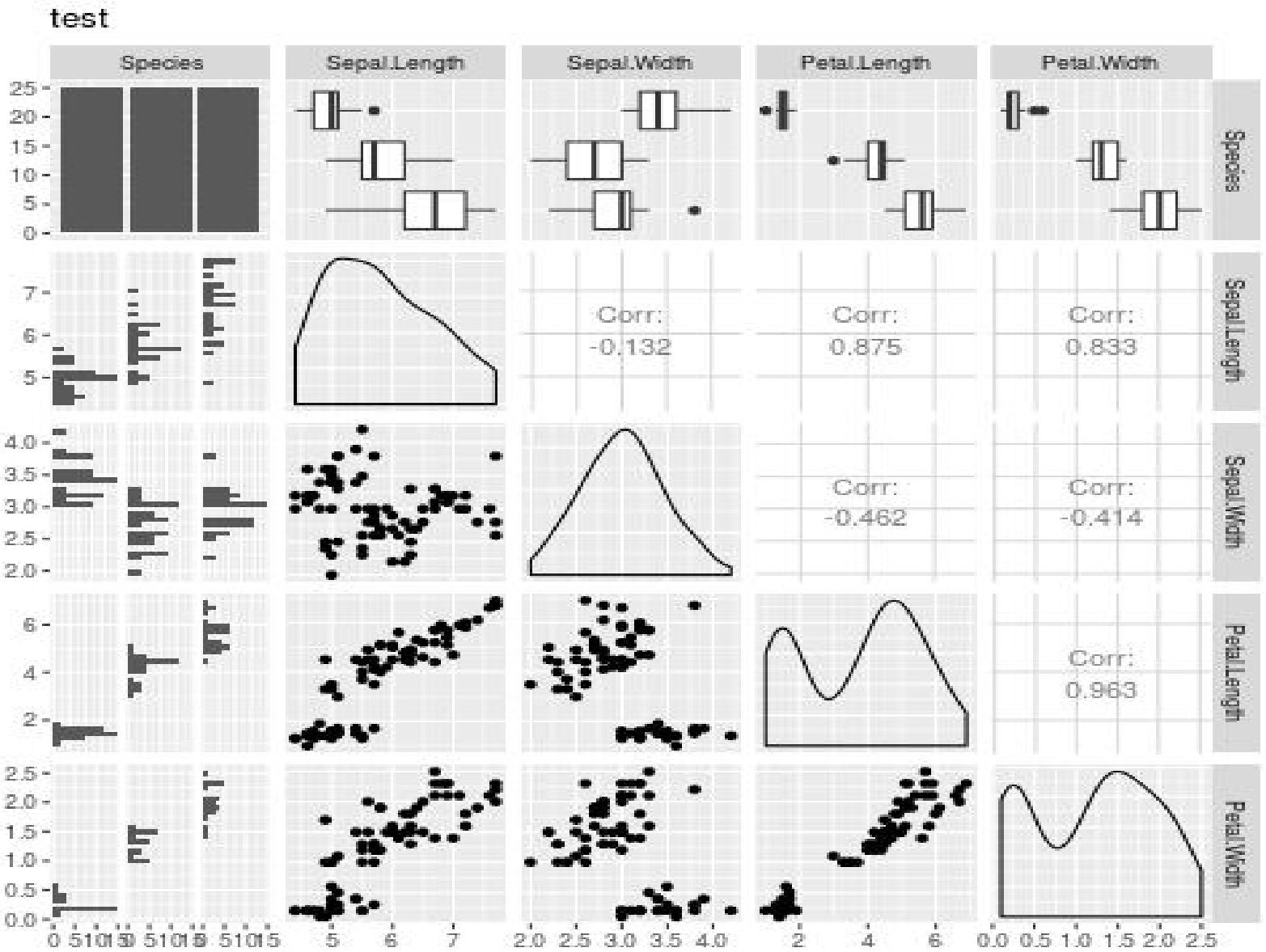
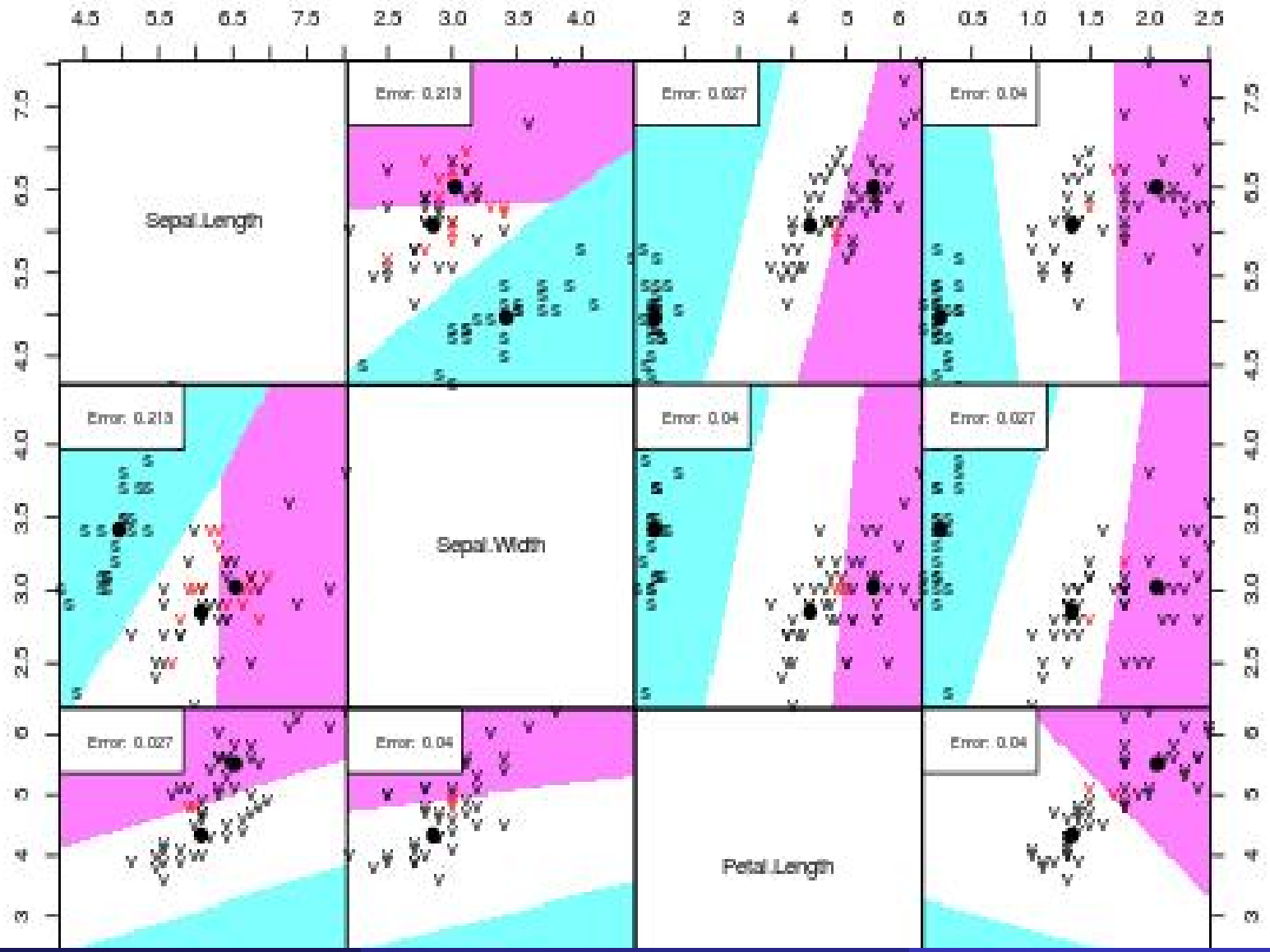


Figure: Scatterplot for test data, Iris



Sepal.Length is potentially not contribute much to the classification, less obvious pattern has been observed in scatterplot.
Also, from the partimat() plot below, more errors are introduced with Sepal.length than between other pairs.

Figure: partimat() plot for train data, Iris



```
> 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            4.980       3.444        1.484        0.26
```

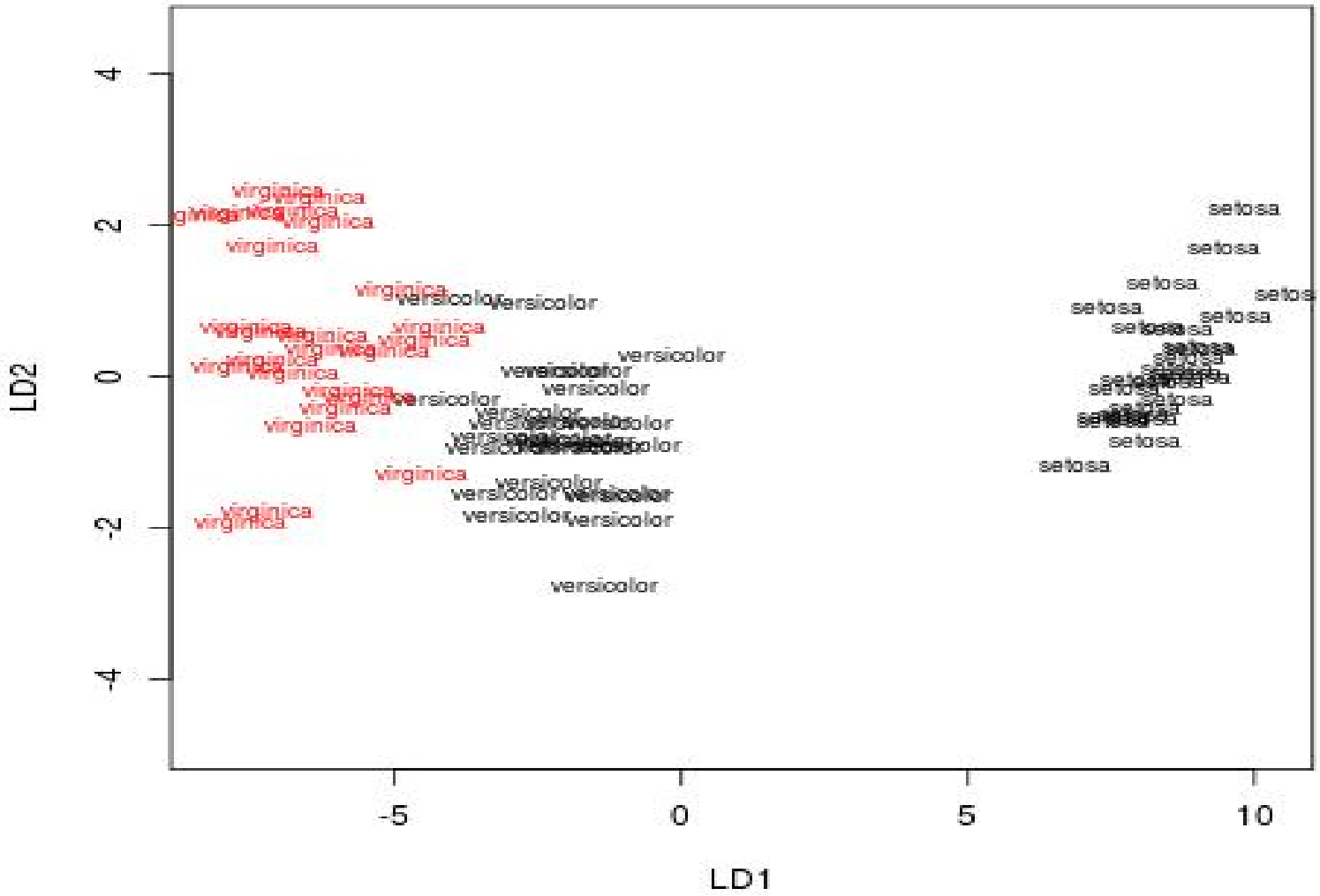
```
versicolor         5.992        2.788         4.332        1.38
virginica          6.588        2.980         5.536        2.02

Coefficients of linear discriminants:
                    LD1        LD2
Sepal.Length  0.5385574 -0.2442458
Sepal.Width   1.6681421  2.4278745
Petal.Length -2.2315596 -0.3772007
Petal.Width  -2.2243534  1.8948979

Proportion of trace:
  LD1   LD2
0.993 0.007
>
```

The linear discriminant functions that best classify the Species in the training set are
LD1 = 0.5385 sepalL + 1.668 sepalW - 2.232 petalL -2.224 petalW
LD2 = -0.244 sepalL + 2.428 sepalW -0.377 petalL + 1.895 petalW.

The plots of the lda object shows the data on the LD scale.

```
plot(lda.iris, dimen = 1, col = as.numeric(iris$Species))
plot(lda.iris, dimen = 2, col = as.numeric(iris$Species))
```

Figure: The plots of the lda object shows the data on the LD scale., Iris train data



Jackknife

```
# 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

# Create a table of classification and posterior probabiliti
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
> classify.iris
       Species      class error postsetosa postversicolor po
1       setosa     setosa                1          0.000
3       setosa     setosa                1          0.000
12      setosa     setosa                1          0.000
```

13	setosa	setosa	1	0.000
14	setosa	setosa	1	0.000
16	setosa	setosa	1	0.000
19	setosa	setosa	1	0.000
20	setosa	setosa	1	0.000
22	setosa	setosa	1	0.000
24	setosa	setosa	1	0.000
25	setosa	setosa	1	0.000
28	setosa	setosa	1	0.000
31	setosa	setosa	1	0.000
32	setosa	setosa	1	0.000
33	setosa	setosa	1	0.000
34	setosa	setosa	1	0.000
35	setosa	setosa	1	0.000
36	setosa	setosa	1	0.000
39	setosa	setosa	1	0.000
43	setosa	setosa	1	0.000

44	setosa	setosa		1	0.000
45	setosa	setosa		1	0.000
46	setosa	setosa		1	0.000
48	setosa	setosa		1	0.000
50	setosa	setosa		1	0.000
51	versicolor	versicolor		0	0.999
52	versicolor	versicolor		0	0.998
55	versicolor	versicolor		0	0.988
60	versicolor	versicolor		0	0.999
62	versicolor	versicolor		0	0.999
63	versicolor	versicolor		0	1.000
66	versicolor	versicolor		0	1.000
67	versicolor	versicolor		0	0.976
71	versicolor	virginica	-1	0	0.403
73	versicolor	versicolor		0	0.665
75	versicolor	versicolor		0	1.000
77	versicolor	versicolor		0	0.988

78	versicolor	versicolor		0	0.631
79	versicolor	versicolor		0	0.989
81	versicolor	versicolor		0	1.000
82	versicolor	versicolor		0	1.000
84	versicolor	virginica	-1	0	0.107
85	versicolor	versicolor		0	0.954
88	versicolor	versicolor		0	0.997
89	versicolor	versicolor		0	1.000
90	versicolor	versicolor		0	1.000
95	versicolor	versicolor		0	0.999
98	versicolor	versicolor		0	1.000
99	versicolor	versicolor		0	1.000
100	versicolor	versicolor		0	1.000
101	virginica	virginica		0	0.000
102	virginica	virginica		0	0.010
103	virginica	virginica		0	0.000
105	virginica	virginica		0	0.000

109	virginica	virginica		0	0.001
110	virginica	virginica		0	0.000
115	virginica	virginica		0	0.000
120	virginica	virginica		0	0.353
121	virginica	virginica		0	0.000
123	virginica	virginica		0	0.000
125	virginica	virginica		0	0.001
127	virginica	virginica		0	0.444
128	virginica	virginica		0	0.357
129	virginica	virginica		0	0.000
130	virginica	virginica		0	0.087
132	virginica	virginica		0	0.001
133	virginica	virginica		0	0.000
134	virginica	versicolor	1	0	0.727
135	virginica	virginica		0	0.139
136	virginica	virginica		0	0.000
139	virginica	versicolor	1	0	0.506

142	virginica	virginica		0	0.013
146	virginica	virginica		0	0.002
149	virginica	virginica		0	0.000
150	virginica	virginica		0	0.086
		1	0.000	0.000	
31	setosa	setosa		1	0.000
32	setosa	setosa		1	0.000
33	setosa	setosa		1	0.000
34	setosa	setosa		1	0.000
35	setosa	setosa		1	0.000
36	setosa	setosa		1	0.000
39	setosa	setosa		1	0.000
43	setosa	setosa		1	0.000
44	setosa	setosa		1	0.000
45	setosa	setosa		1	0.000
46	setosa	setosa		1	0.000
48	setosa	setosa		1	0.000

50	setosa	setosa		1	0.000
51	versicolor	versicolor		0	0.999
52	versicolor	versicolor		0	0.998
55	versicolor	versicolor		0	0.988
60	versicolor	versicolor		0	0.999
62	versicolor	versicolor		0	0.999
63	versicolor	versicolor		0	1.000
66	versicolor	versicolor		0	1.000
67	versicolor	versicolor		0	0.976
71	versicolor	virginica	-1	0	0.403
73	versicolor	versicolor		0	0.665
75	versicolor	versicolor		0	1.000
77	versicolor	versicolor		0	0.988
78	versicolor	versicolor		0	0.631
79	versicolor	versicolor		0	0.989
81	versicolor	versicolor		0	1.000
82	versicolor	versicolor		0	1.000

84	versicolor	virginica	-1	0	0.107
85	versicolor	versicolor		0	0.954
88	versicolor	versicolor		0	0.997
89	versicolor	versicolor		0	1.000
90	versicolor	versicolor		0	1.000
95	versicolor	versicolor		0	0.999
98	versicolor	versicolor		0	1.000
99	versicolor	versicolor		0	1.000
100	versicolor	versicolor		0	1.000
101	virginica	virginica		0	0.000
102	virginica	virginica		0	0.010
103	virginica	virginica		0	0.000
105	virginica	virginica		0	0.000
109	virginica	virginica		0	0.001
110	virginica	virginica		0	0.000
115	virginica	virginica		0	0.000
120	virginica	virginica		0	0.353

121	virginica	virginica		0	0.000
123	virginica	virginica		0	0.000
125	virginica	virginica		0	0.001
127	virginica	virginica		0	0.444
128	virginica	virginica		0	0.357
129	virginica	virginica		0	0.000
130	virginica	virginica		0	0.087
132	virginica	virginica		0	0.001
133	virginica	virginica		0	0.000
134	virginica	versicolor	1	0	0.727
135	virginica	virginica		0	0.139
136	virginica	virginica		0	0.000
139	virginica	versicolor	1	0	0.506
142	virginica	virginica		0	0.013
146	virginica	virginica		0	0.002
149	virginica	virginica		0	0.000
150	virginica	virginica		0	0.086

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

# proportion correct for each category
diag(prop.table(pred.freq, 1))
# total proportion correct
sum(diag(prop.table(pred.freq)))
# total error rate
1 - sum(diag(prop.table(pred.freq)))

> pred.freq
```

```
            setosa versicolor virginica
  setosa        25          0         0
  versicolor     0         23         2
  virginica      0          2        23
> 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       0.08      0.92
> # proportion correct for each category
> diag(prop.table(pred.freq, 1))
    setosa versicolor  virginica
      1.00       0.92       0.92
> # total proportion correct
> sum(diag(prop.table(pred.freq)))
[1] 0.9466667
```

```
> # total error rate
> 1 - sum(diag(prop.table(pred.freq)))
[1] 0.05333333
> # proportion correct for each category
> diag(prop.table(pred.freq, 1))
    setosa versicolor  virginica
      1.00       0.92       0.92
> # total proportion correct
> sum(diag(prop.table(pred.freq)))
[1] 0.9466667
> # total error rate
> 1 - sum(diag(prop.table(pred.freq)))
[1] 0.05333333
>
```

The misclassification error is low within the training set.

predict the test data from the training data LDFs

```
pred.iris <- predict(lda.iris, newdata = subset(iris,
test == "test"))
```

```
> pred.freq

            setosa versicolor virginica
  setosa        25          0         0
  versicolor     0         25         0
  virginica      0          0        25
> prop.table(pred.freq, 1) # proportions by row

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

```
  virginica          0             0             1
>
> # proportion correct for each category
> diag(prop.table(pred.freq, 1))
    setosa versicolor  virginica
         1          1           1
> # total proportion correct
> sum(diag(prop.table(pred.freq)))
[1] 1
> # total error rate
> 1 - sum(diag(prop.table(pred.freq)))
[1] 0
>
```

The classification rule based on the training set works well with the test data. Do not expect such nice results on all classification problems! Usually the error rate is slightly higher on the test data than on the training data.

Stepwise variable selection for classification

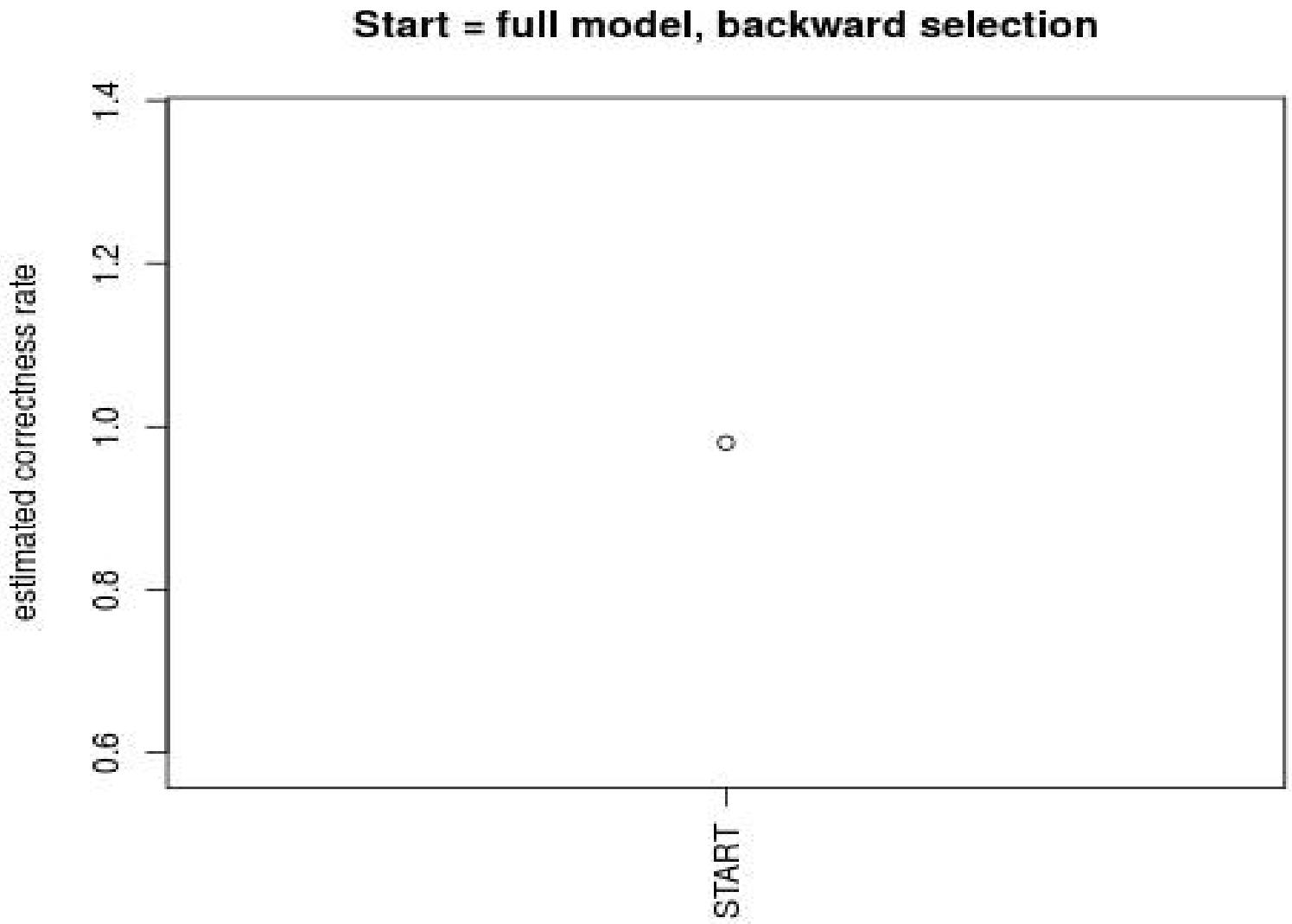
- performed using package klaR and function stepclass() ——apply to any specified classification function.
- classification performance is estimated by one of Uschis classification performance measures.
- the resulting model can be very sensitive to the starting model.
- running this repeatedly could result in slightly different models because the k-fold crossvalidation partitions the data at random. The formula object gives the selected model.

```
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")
plot(step.iris.b, main = "Start = full model,
 backward selection")
 > step.iris.b$formula
Species ~ Sepal.Length + Sepal.Width
+ Petal.Length + Petal.Width
```

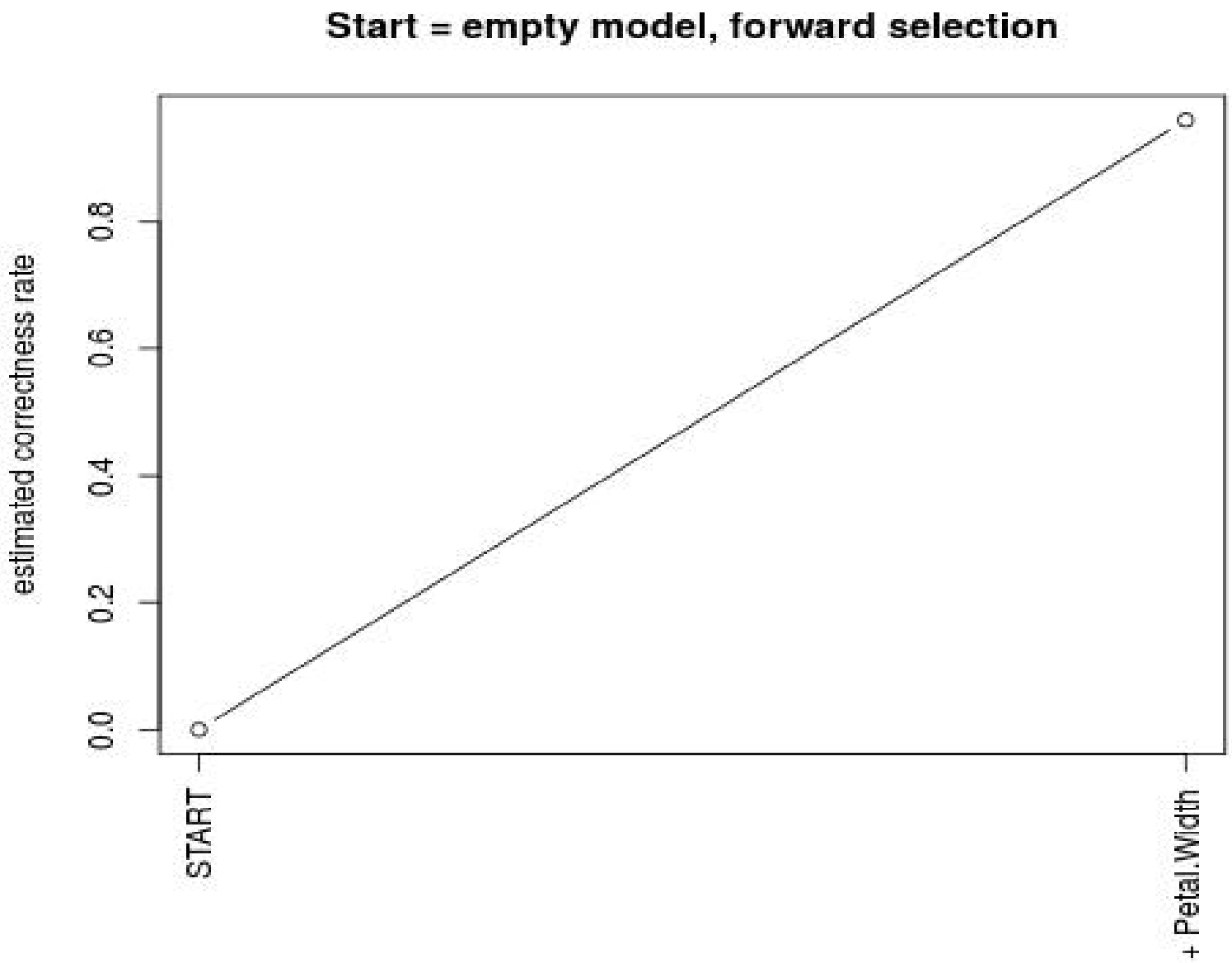
```
lda.iris.step <- lda(step.iris.b$formula
                     , data = iris)
> lda.iris.step
Call:
lda(step.iris.b$formula, data = iris)

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
```

Figure: Stepwise starts with full and ends with full.



```
# 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")
plot(step.iris.f, main = "Start = empty model,
 forward selection")
>step.iris.f$formula
Species ~ Petal.Width
```


Start = empty model, forward selection
estimated correctness rate
0.0
0.2
0.4
0.6
0.8
START
+ Petal.Width