STAT 8020 R Lab 21

Whitney

November 23, 2020

Contents

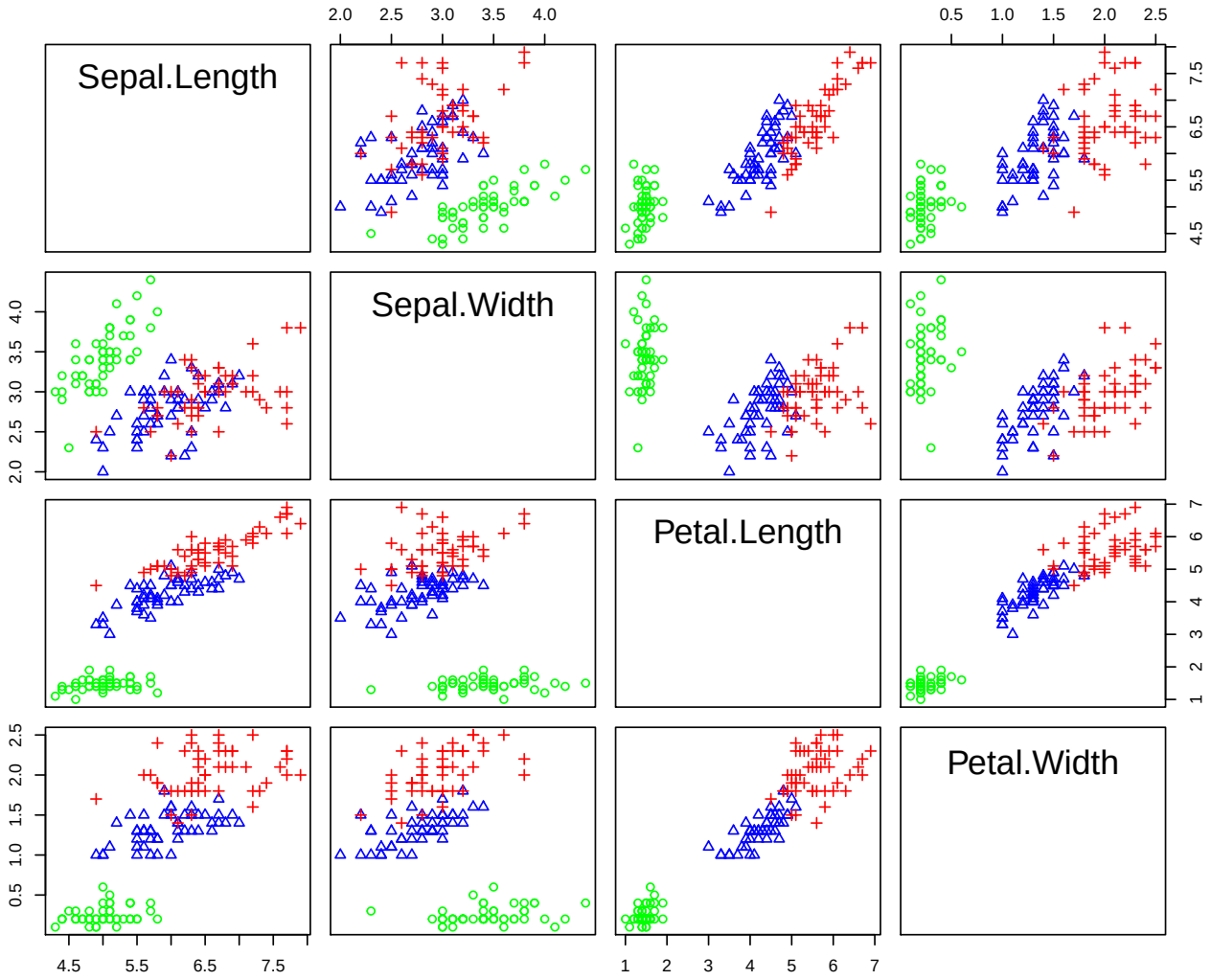
Classification

Iris data

```
data(iris)
head(iris)
```

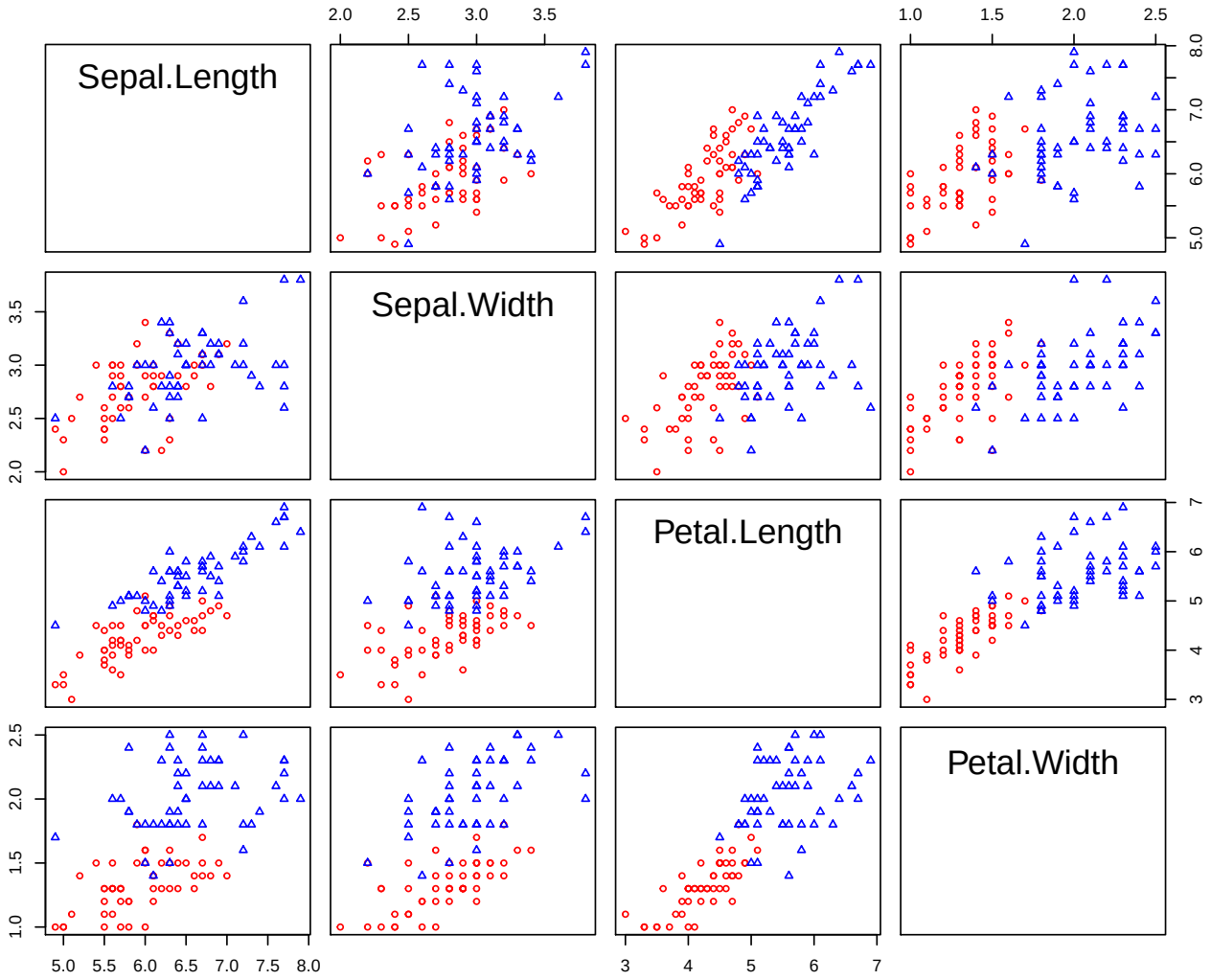
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
attach(iris)

library(car)
scatterplotMatrix(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width | Species,
                  col = c("green", "blue", "red"), diagonal = F,smooth = F, regLine = F,
                  legend = F)
```



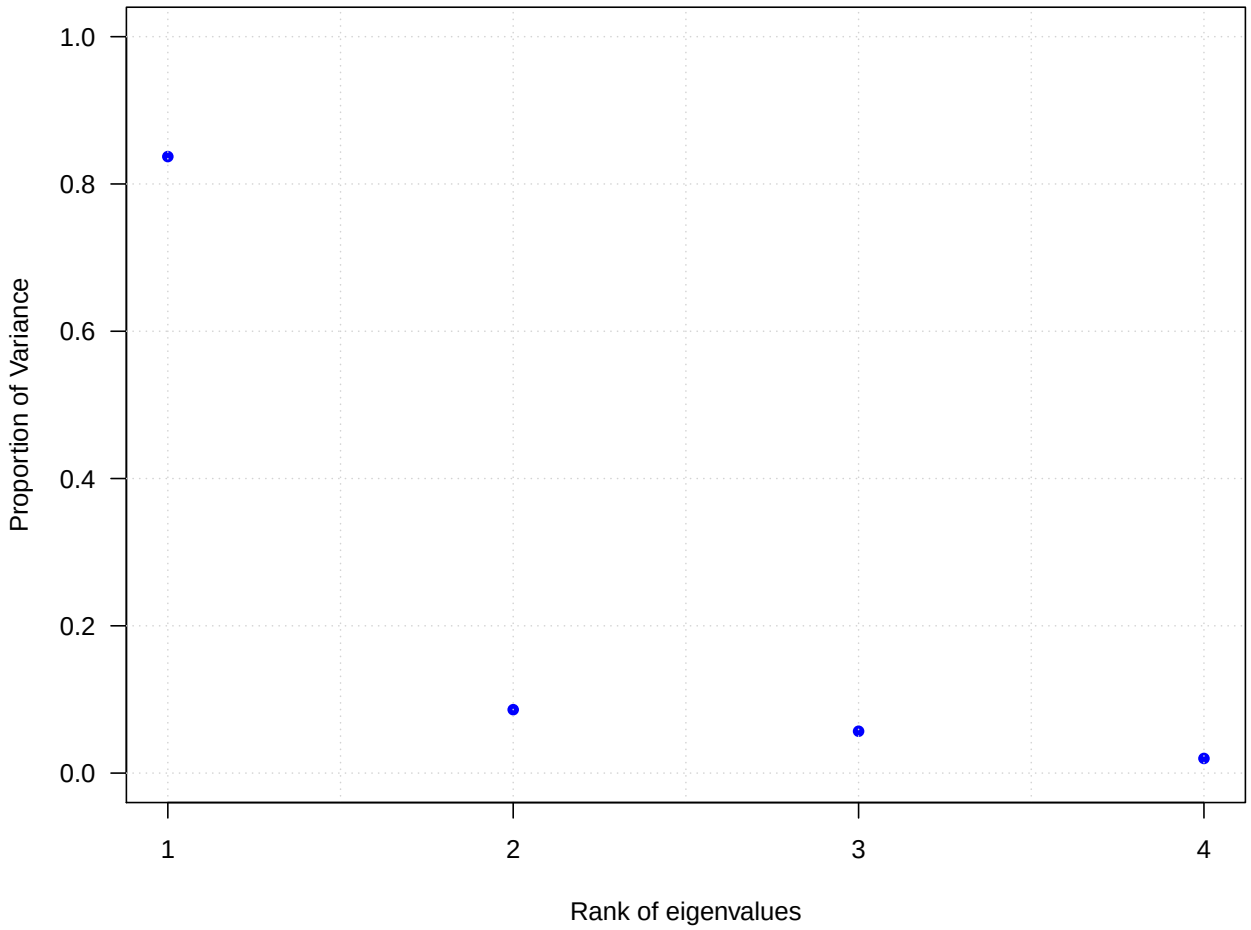
Binary classification

```
irisv = iris[51:150,]
irisv$Species <- factor(irisv$Species)
attach(irisv)

scatterplotMatrix(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width | Species,
                  col = c("red", "blue"), diagonal = F,smooth = F, regLine = F,
                  legend = F, cex = 0.75)
```

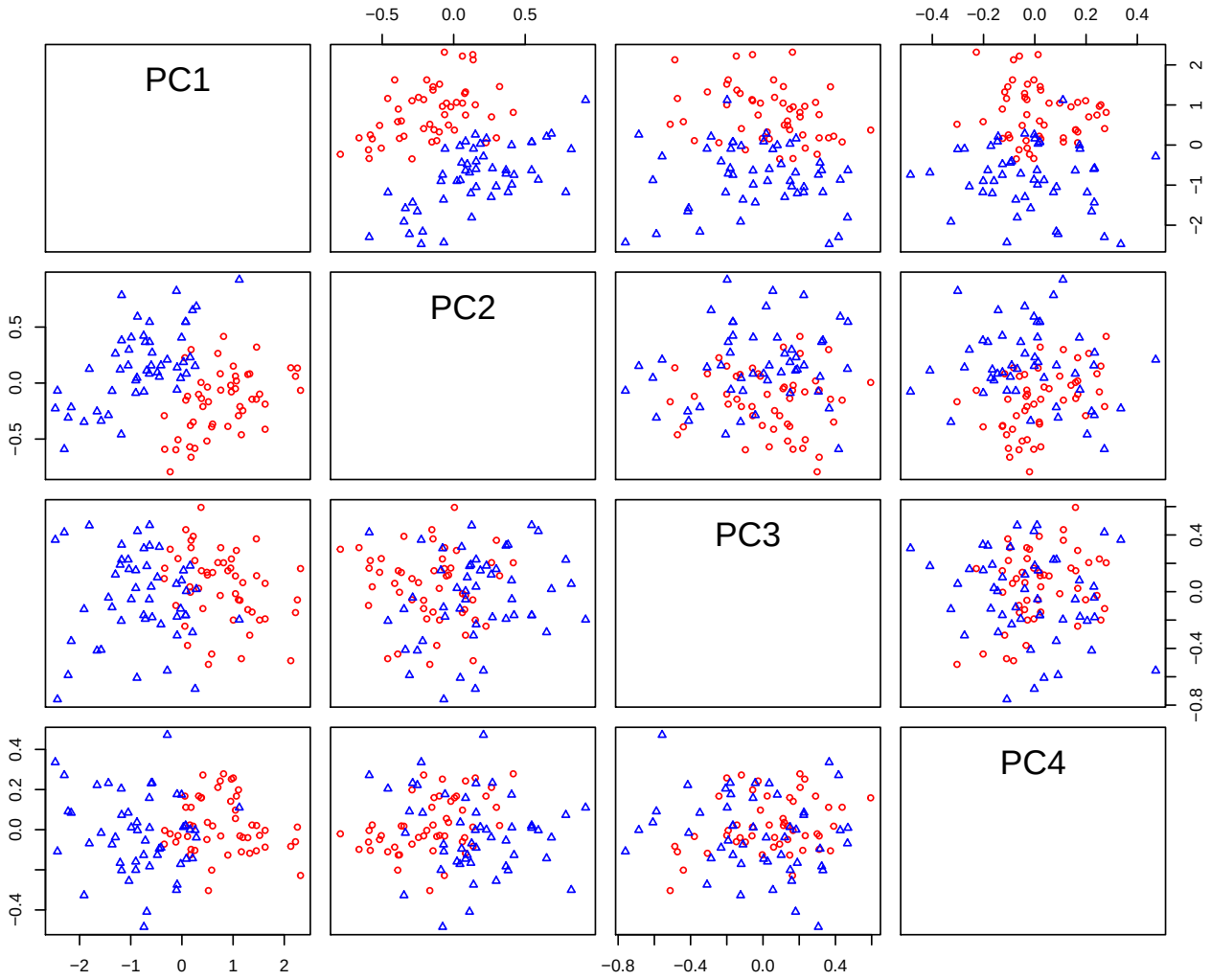


PCA

```
pca <- prcomp(irisv[, 1:4])
Z <- pca$x
lambda <- pca$sdev^2

plot(1:4, lambda / sum(lambda), xaxt = "n", las = 1, xlab = "Rank of eigenvalues",
     ylab = "Proportion of Variance", pch = 16, col = "blue", cex = 1, ylim = c(0, 1))
grid(); axis(1, at = 1:4)
```




```
scatterplotMatrix(~ Z | Species, col = c("red", "blue"), diagonal = F,smooth = F,
                  regLine = F, legend = F, cex = 0.75)
```



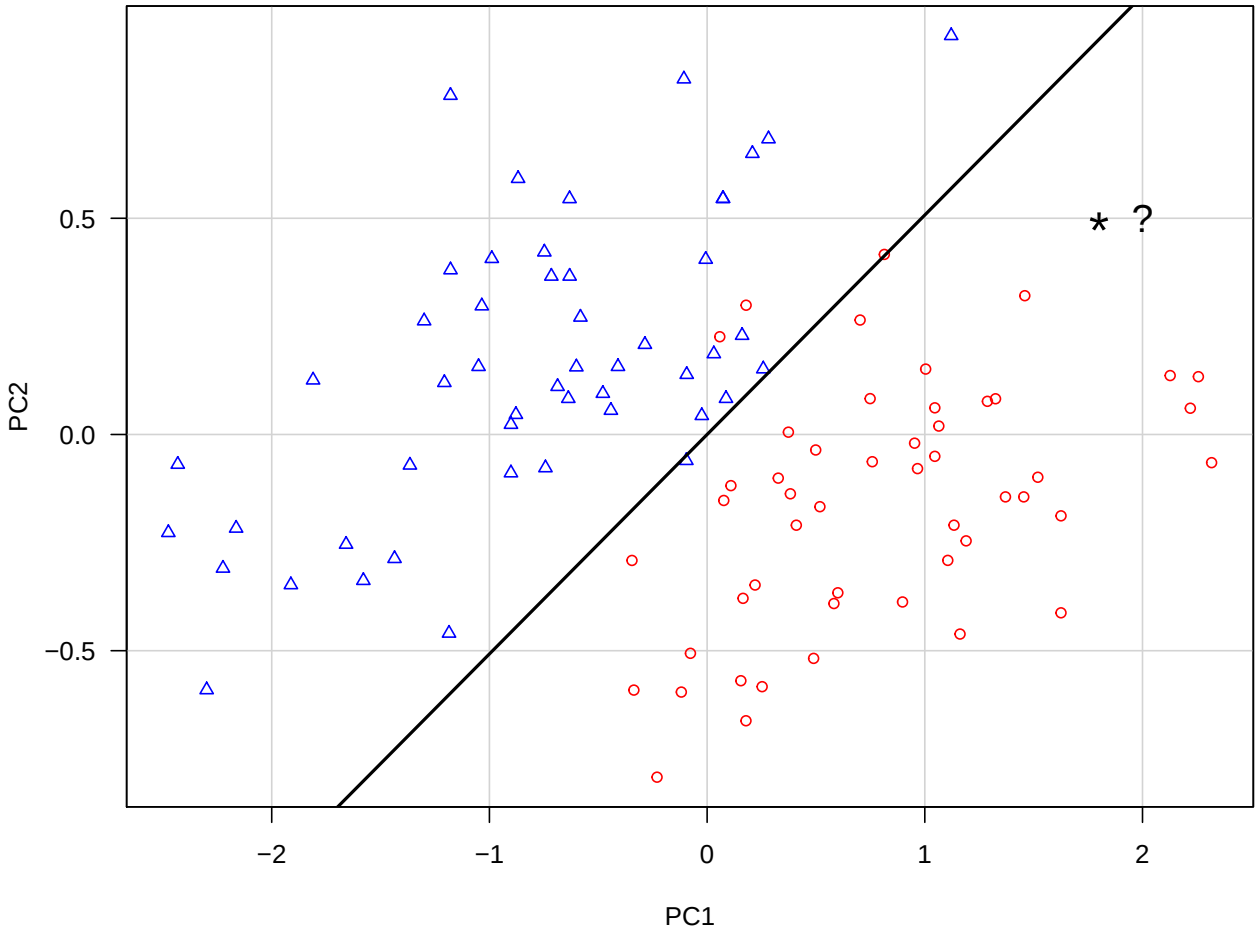
LDA

```
library(MASS)
par(las = 1)
scatterplot(PC2 ~ PC1 | Species , Z, smooth = F, regLine = F, legend = F, cex = 0.85,
            col = c("red", "blue"))
fit <- lda(Species ~ Z[, 1:2])
fit # show results
```

```
## Call:
## lda(Species ~ Z[, 1:2])
##
## Prior probabilities of groups:
## versicolor  virginica
##        0.5        0.5
##
## Group means:
##            Z[, 1:2]PC1 Z[, 1:2]PC2
## versicolor   0.7930189  -0.1607571
## virginica   -0.7930189   0.1607571
##
```

```
## Coefficients of linear discriminants:
##                     LD1
## Z[, 1:2]PC1 -1.553249
## Z[, 1:2]PC2  3.060560
```

```
abline(0, -fit$scaling[1] / fit$scaling[2], pch = 5, lwd = 2)
points(2, 0.5, pch = "?", cex = 1.5)
points(1.8, 0.5, pch = "*", cex = 2)
```



LDA and QDA

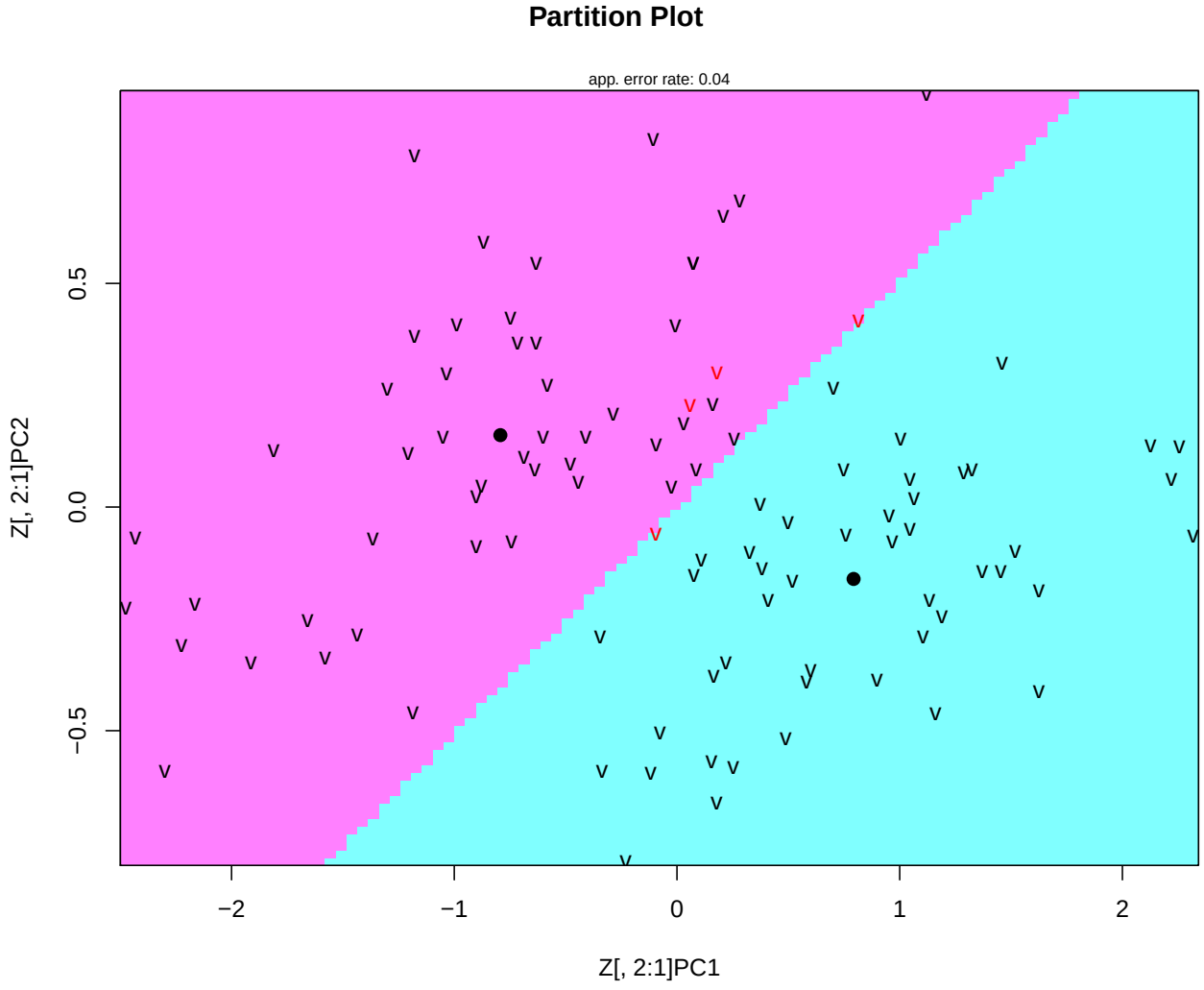
```
#treat data as matrix
z = as.matrix(Z)

lda <- lda(irisv$Species ~ Z[, 1:2])
qda <- qda(irisv$Species ~ Z[, 1:2])

fit.LDA = predict(lda)$class
table(irisv$Species, fit.LDA)
```

```
##             fit.LDA
##              versicolor virginica
##   versicolor         47         3
##   virginica           1        49
```

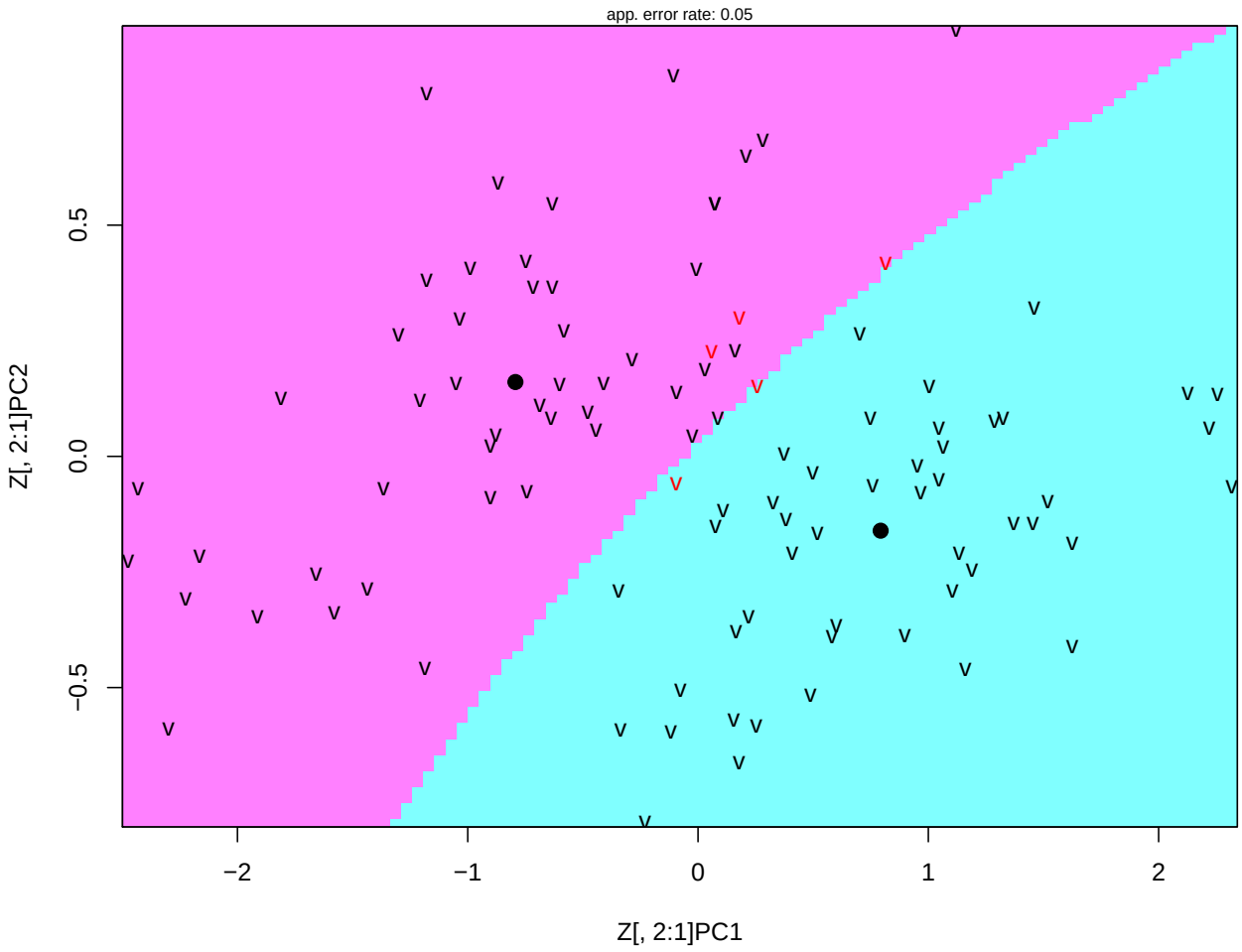
```
fit.QDA = predict(qda)$class
table(irisv$Species, fit.QDA)
```

```
##             fit.QDA
##              versicolor virginica
##   versicolor         47         3
##   virginica           2        48
```

```
# show results
library(klaR)
partimat(Species ~ Z[, 2:1], method = "lda", prec = 100, pch = 16, xaxt = "")
```




```
partimat(Species ~ Z[, 2:1], method = "qda", prec = 100)
```



Logistic Regression

```
logfit <- glm(irisv$Species ~ z[, 1:2], family = binomial)
logpred <- predict(logfit, type = "response")
library(fields)
```

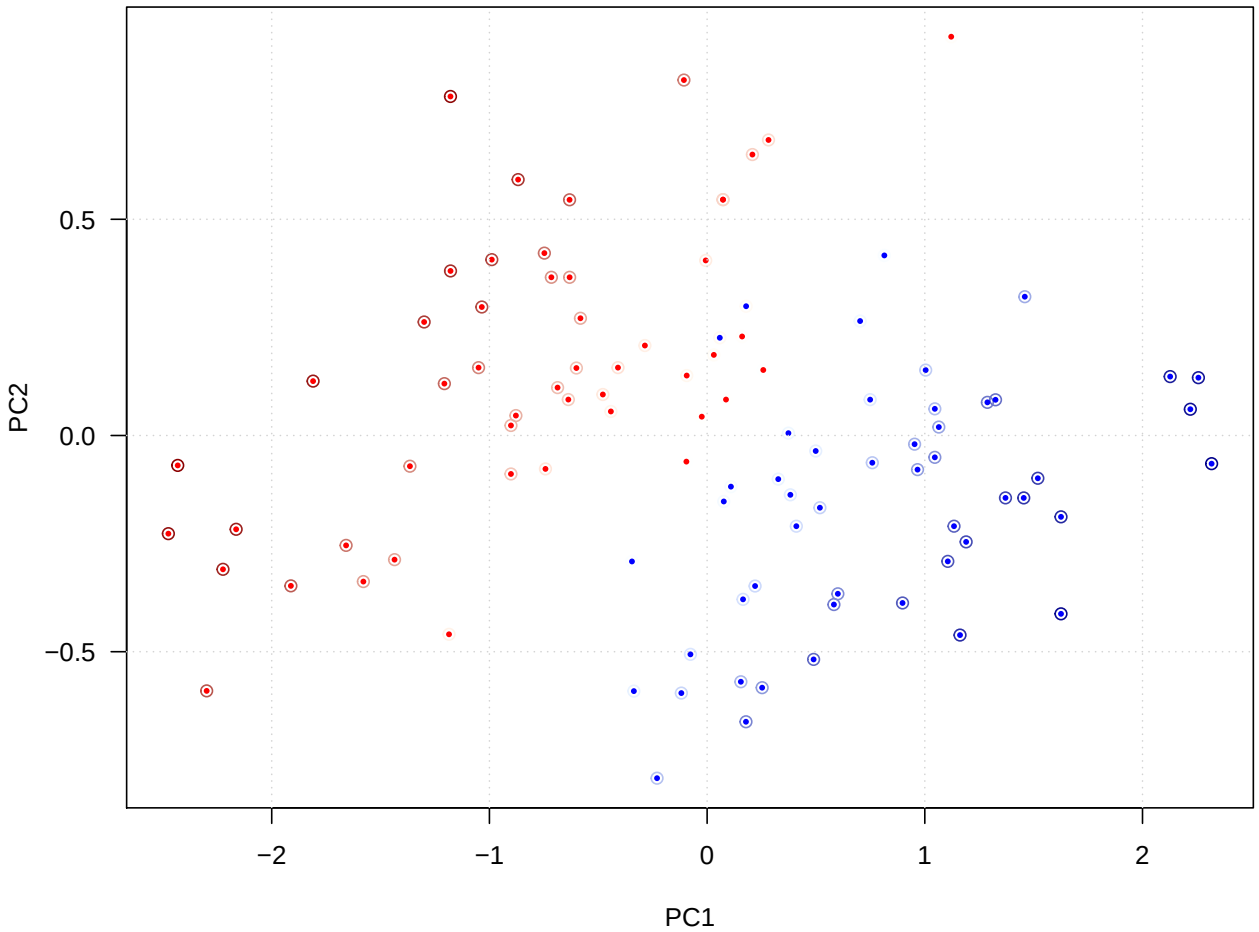
```
## Loading required package: spam
## Loading required package: dotCall64
## Loading required package: grid
## Spam version 2.4-0 (2019-11-01) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
## The following objects are masked from 'package:base':
##
```

```
##     backsolve, forwardsolve
```

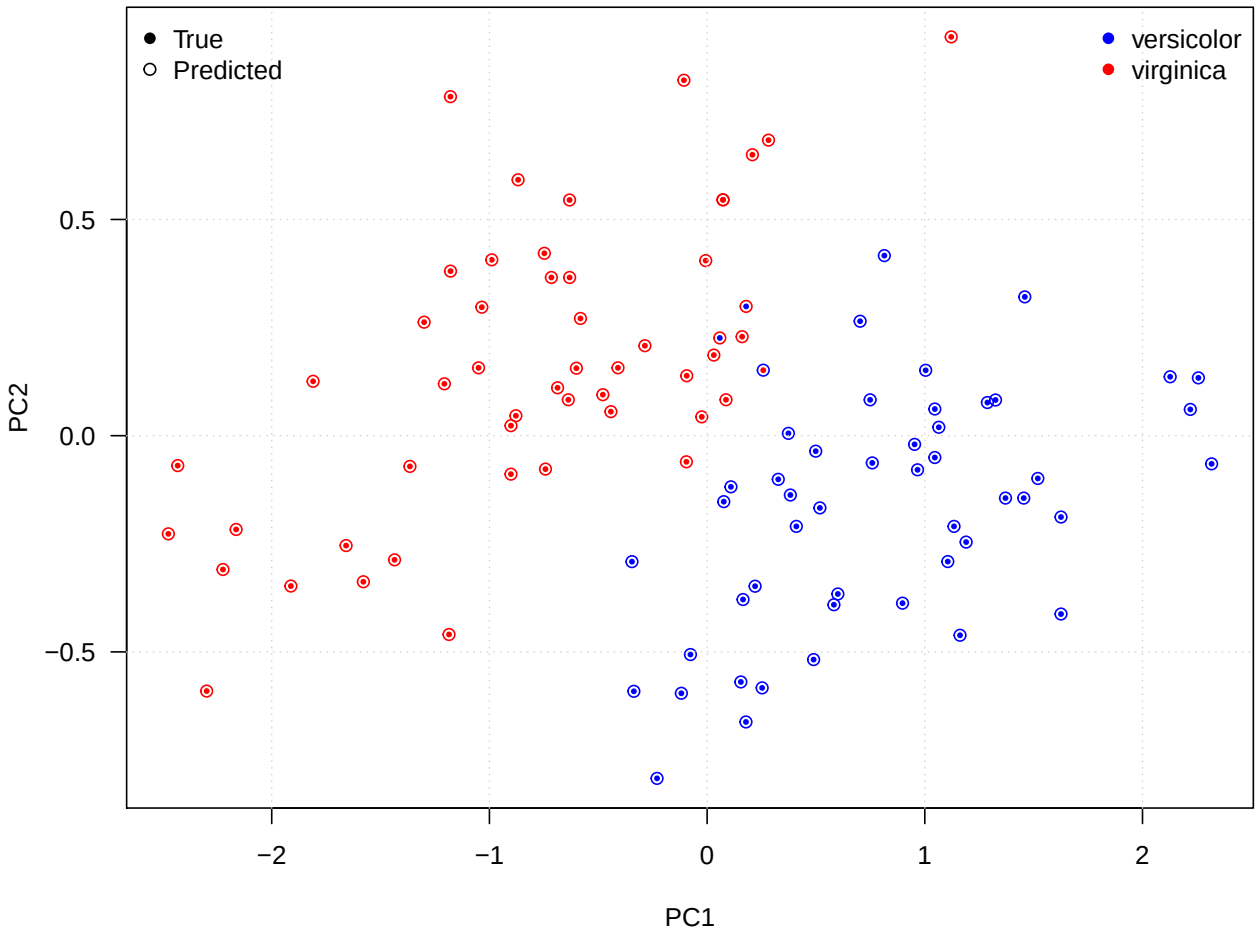
```
## Loading required package: maps
```

```
## See https://github.com/NCAR/Fields for
##  an extensive vignette, other supplements and source code
```

```
cols <- two.colors(n = 100, "darkblue", "darkred")
order <- order(logpred)

predCol <- ifelse(logpred <= 0.5, "blue", "red")
Col <- rep(c("blue", "red"), each = 50)

plot(z[order, 1:2], col = cols, pch = 1, las = 1)
points(z[order, 1:2], col = Col[order], pch = 16, cex = 0.5)
grid()
```




```
plot(z[, 1:2], col = predCol, pch = 1, las = 1)
points(z[, 1:2], col = Col, pch = 16, cex = 0.5)
grid()
legend("topleft", legend = c("True", "Predicted"), pch = c(16, 1), bty = "n")
legend("topright", legend = c("versicolor", "virginica"),
       col = c("blue", "red"), pch = 16, bty = "n")
```



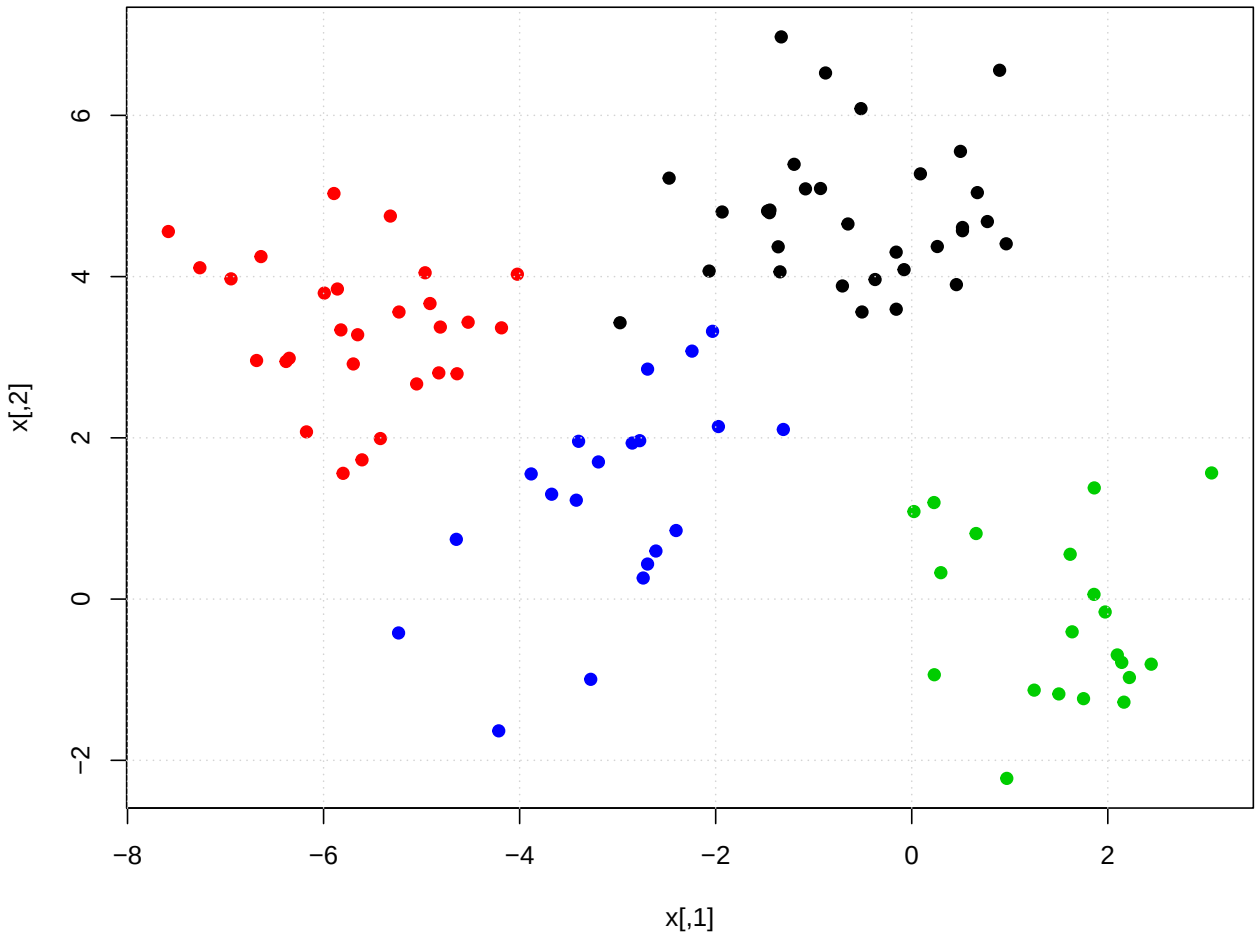

```
logisticPred <- ifelse(logpred <= 0.5, "versicolor", "virginica")
table(irisv$Species, logisticPred)
```

```
##             logisticPred
##              versicolor virginica
##   versicolor         48         2
##   virginica           1        49
```

Clustering

K-Means Clustering

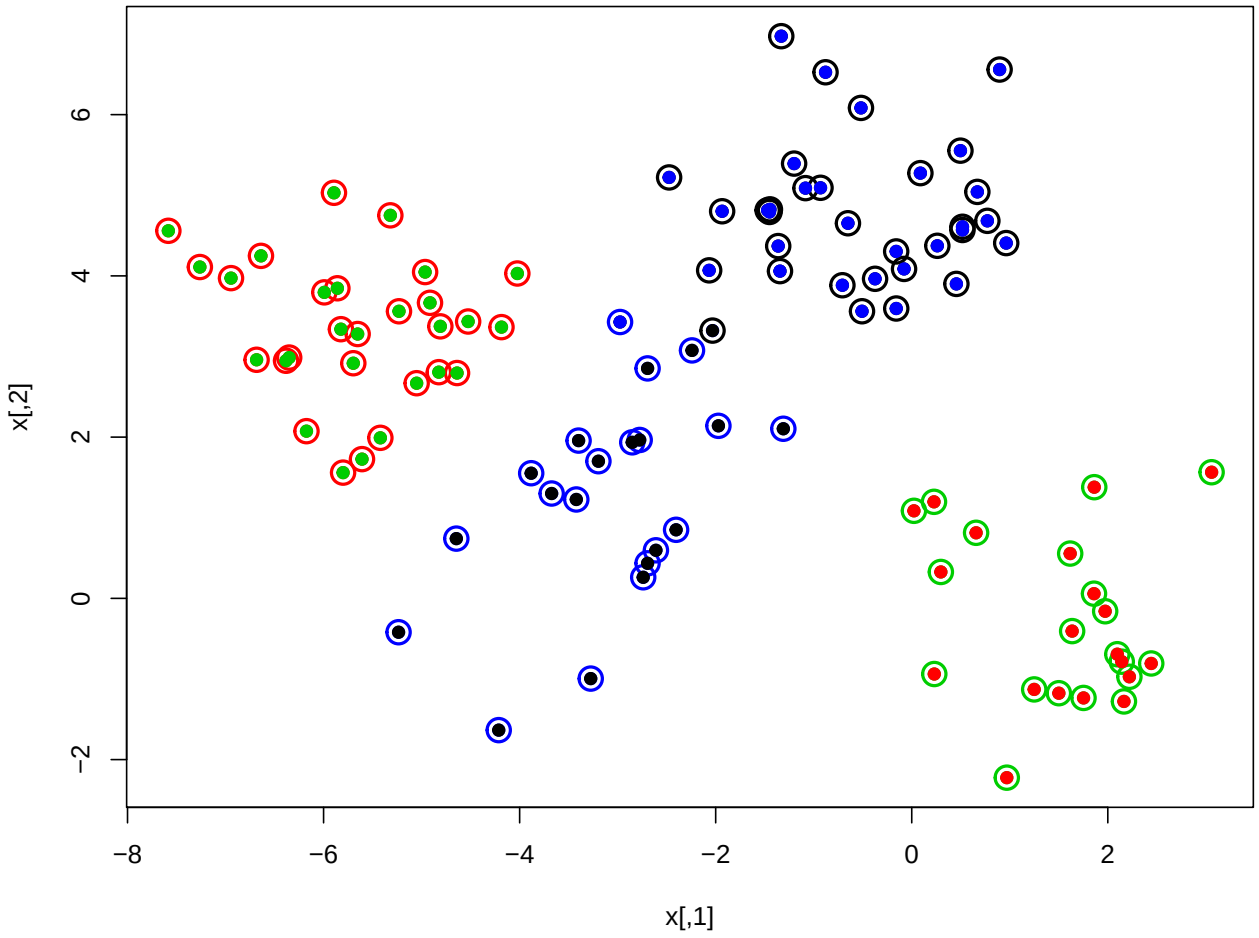
```
set.seed(101)
library(scales)
x <- matrix(rnorm(100 * 2), 100, 2)
xmean <- matrix(rnorm(8, sd = 4), 4, 2)
which <- sample(1:4, 100, replace = TRUE)
x = x + xmean[which,]
plot(x, col = which, pch = 19)
grid()
```




```
km.out <- kmeans(x, 4, nstart = 15)
km.out
```

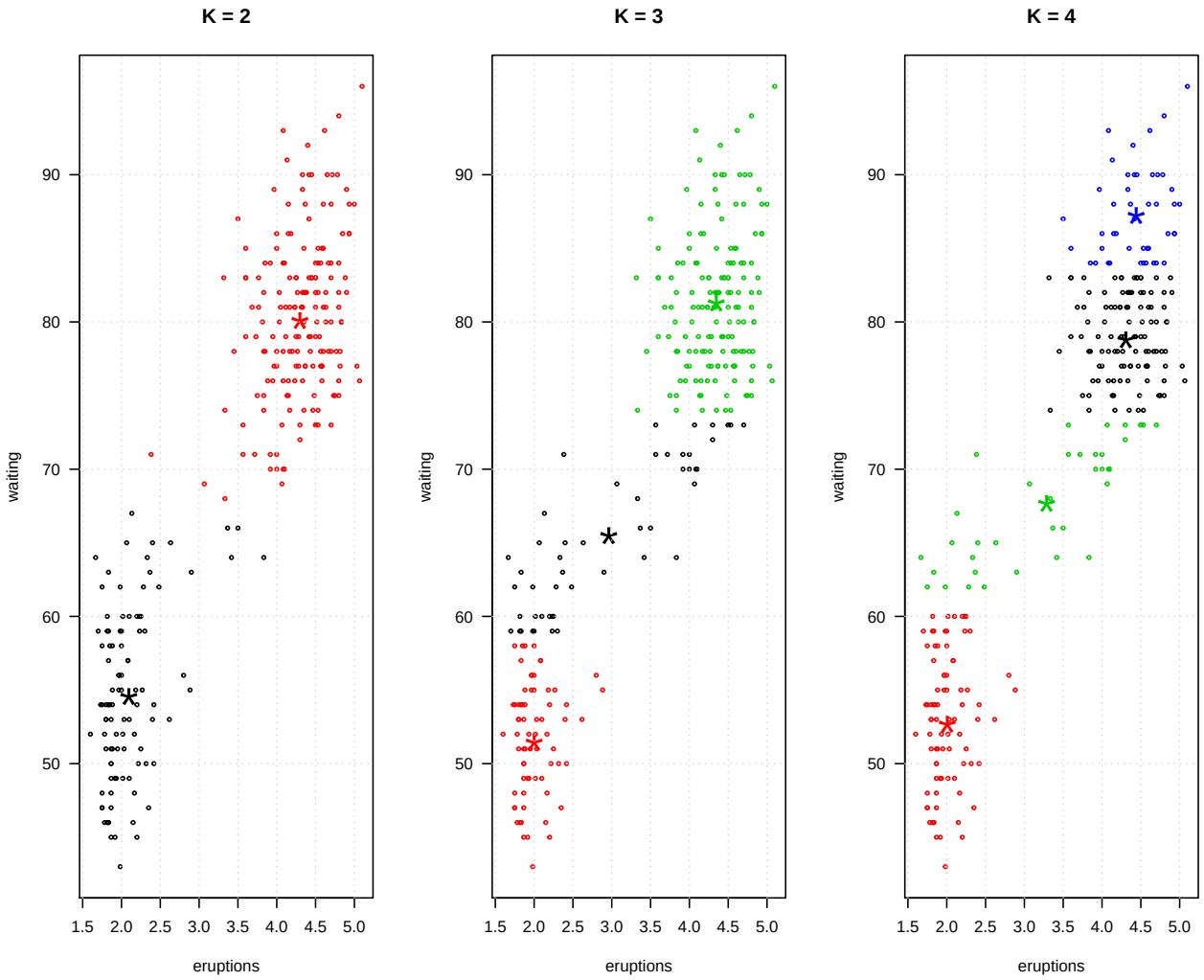
```
## K-means clustering with 4 clusters of sizes 32, 28, 20, 20
##
## Cluster means:
##         [,1]       [,2]
## 1 -0.5787702  4.7639233
## 2 -5.6518323  3.3513316
## 3  1.4989983 -0.2412154
## 4 -3.1104142  1.2535711
##
## Clustering vector:
##   [1] 2 4 1 2 4 1 2 4 1 1 3 1 1 3 4 3 2 3 2 2 2 2 2 3 1 1 4 2 4 1 2 3 2 4 4 3 3
##  [38] 4 3 3 2 4 4 2 2 3 2 1 2 4 2 1 1 3 3 4 3 1 1 1 4 2 2 2 4 4 1 1 3 2 2 1 1 3
##  [75] 1 3 2 1 1 1 4 1 4 1 2 3 1 2 2 1 1 4 2 4 1 1 3 3 1 1
##
## Within cluster sum of squares by cluster:
## [1] 53.04203 42.40322 34.95921 48.52107
##  (between_SS / total_SS =  85.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
plot(x, col=km.out$cluster, cex = 2, pch = 1, lwd = 2)
points(x, col = which, pch = 19)
points(x, col = c(4, 3, 2, 1)[which], pch = 19)
```

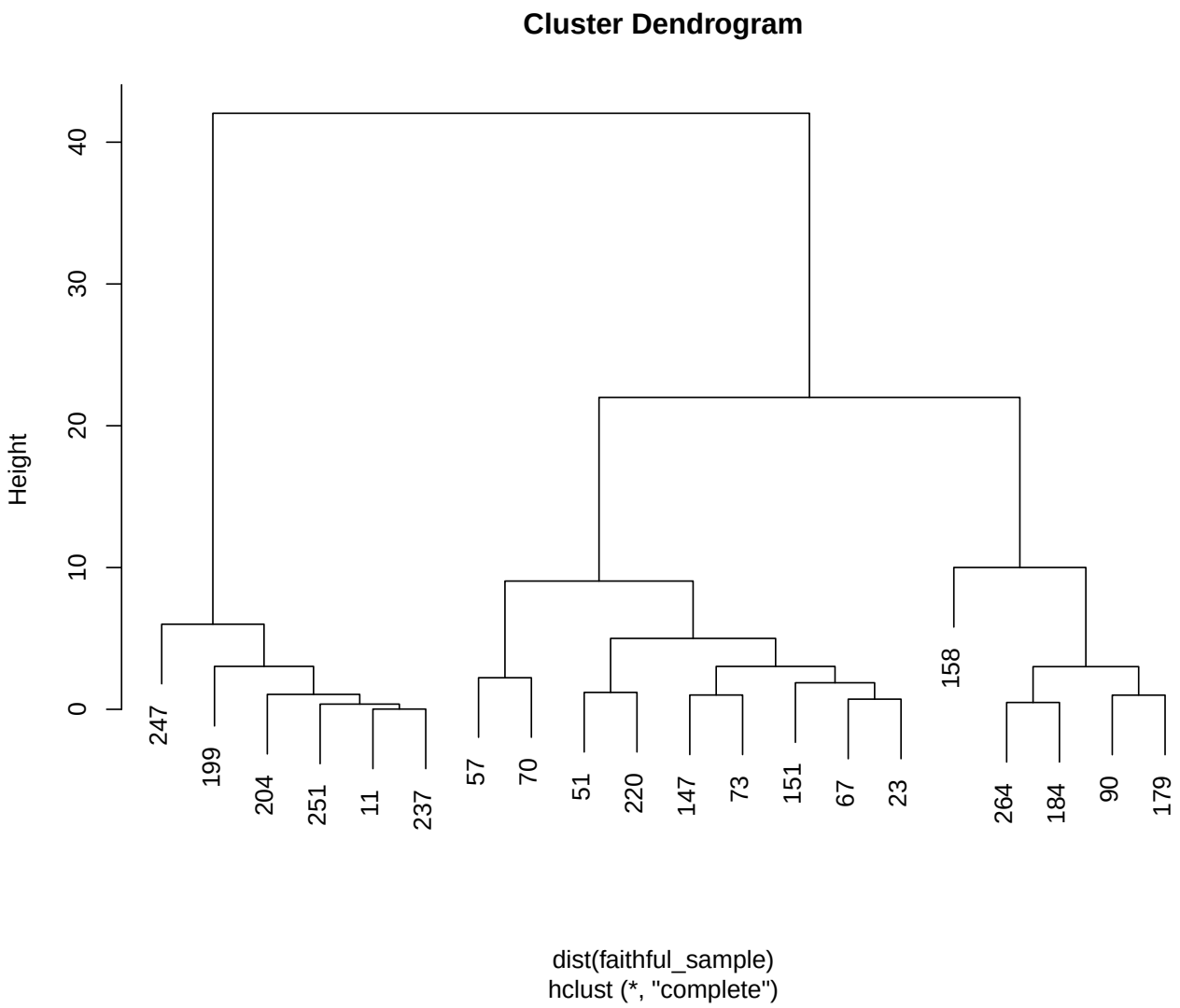


Geyser Example

```r
km3.faithful <- kmeans(faithful, 3)
km2.faithful <- kmeans(faithful, 2)
km4.faithful <- kmeans(faithful, 4)

par(las = 1, mfrow = c(1, 3))
plot(faithful, col = km2.faithful$cluster, cex = 0.5, main = "K = 2")
points(km2.faithful$centers, cex = 3, pch = "*", col = 1:2)
grid()
plot(faithful, col = km3.faithful$cluster, cex = 0.5, main = "K = 3")
points(km3.faithful$centers, cex = 3, pch = "*", col = 1:3)
grid()
plot(faithful, col = km4.faithful$cluster, cex = 0.5, main = "K = 4")
grid()
points(km4.faithful$centers, cex = 3, pch = "*", col = 1:4)
```

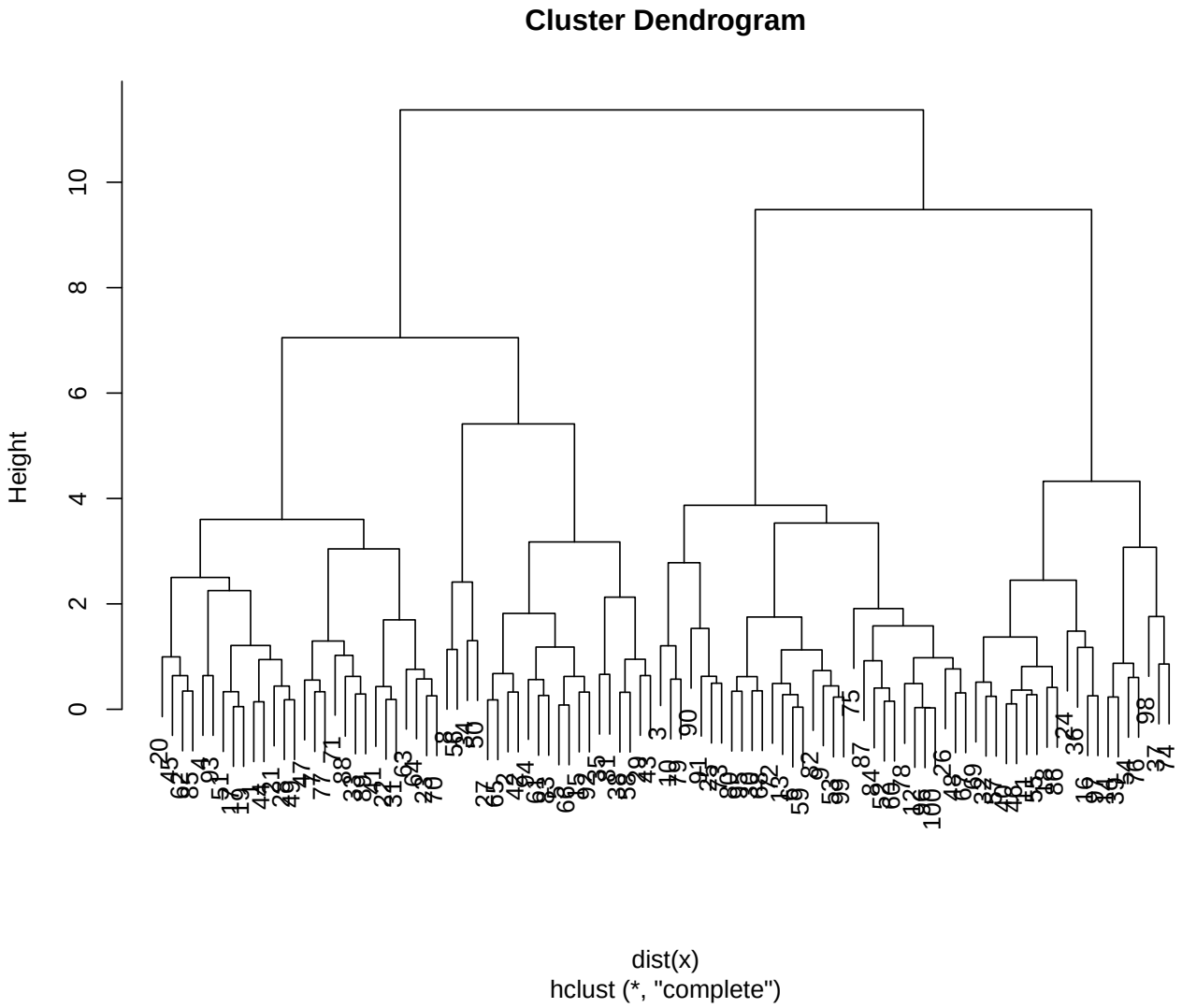



```
kmean3.faithful <- kmeans(x = faithful, centers = 3)

id <- sample(1:272, 20)
faithful_sample <- faithful[id, ]
hc.faithful <- hclust(dist(faithful_sample))
plot(hc.faithful)
```

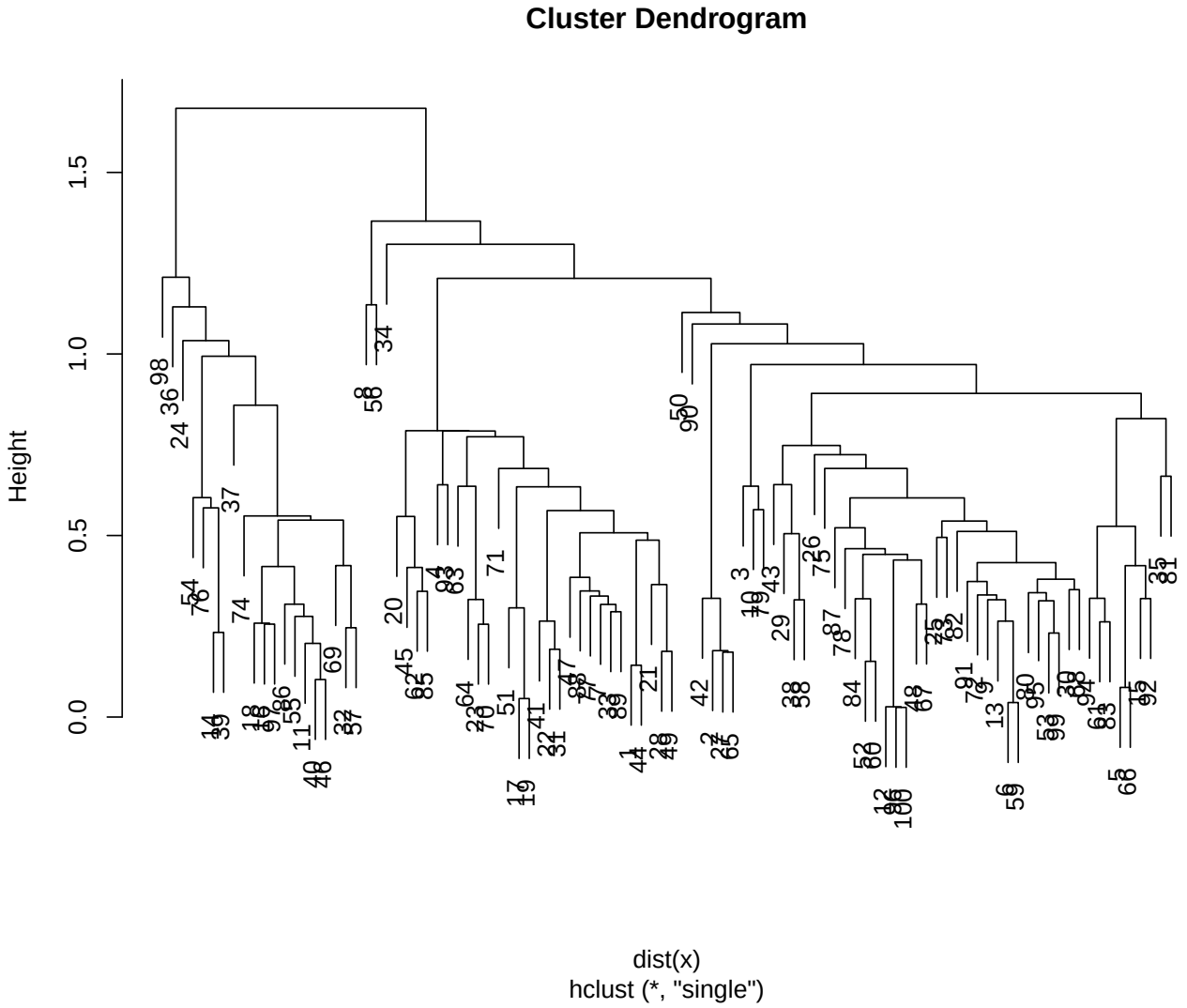


Hierarchical Clustering

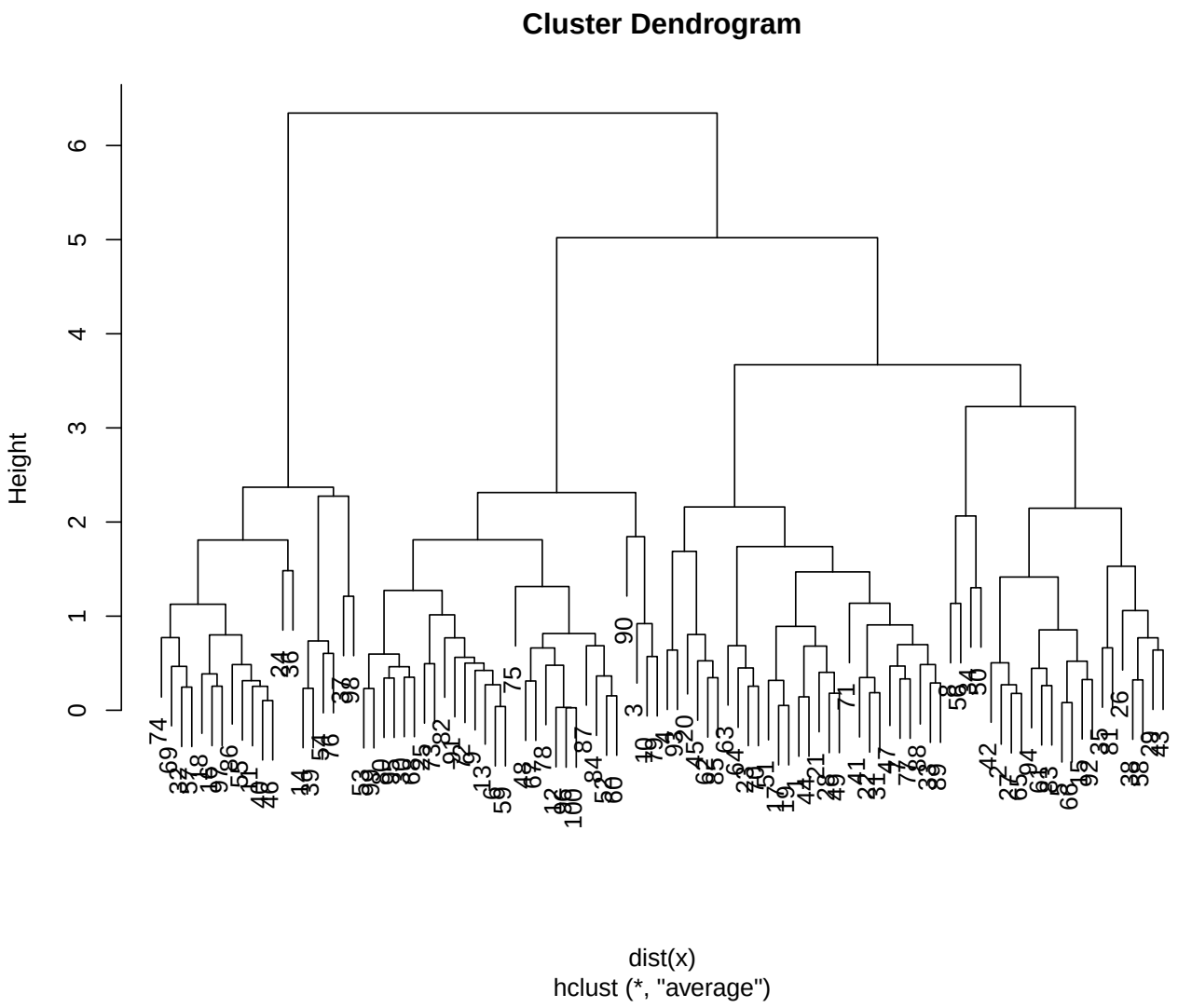
```
hc.complete <- hclust(dist(x), method = "complete")
plot(hc.complete)
```



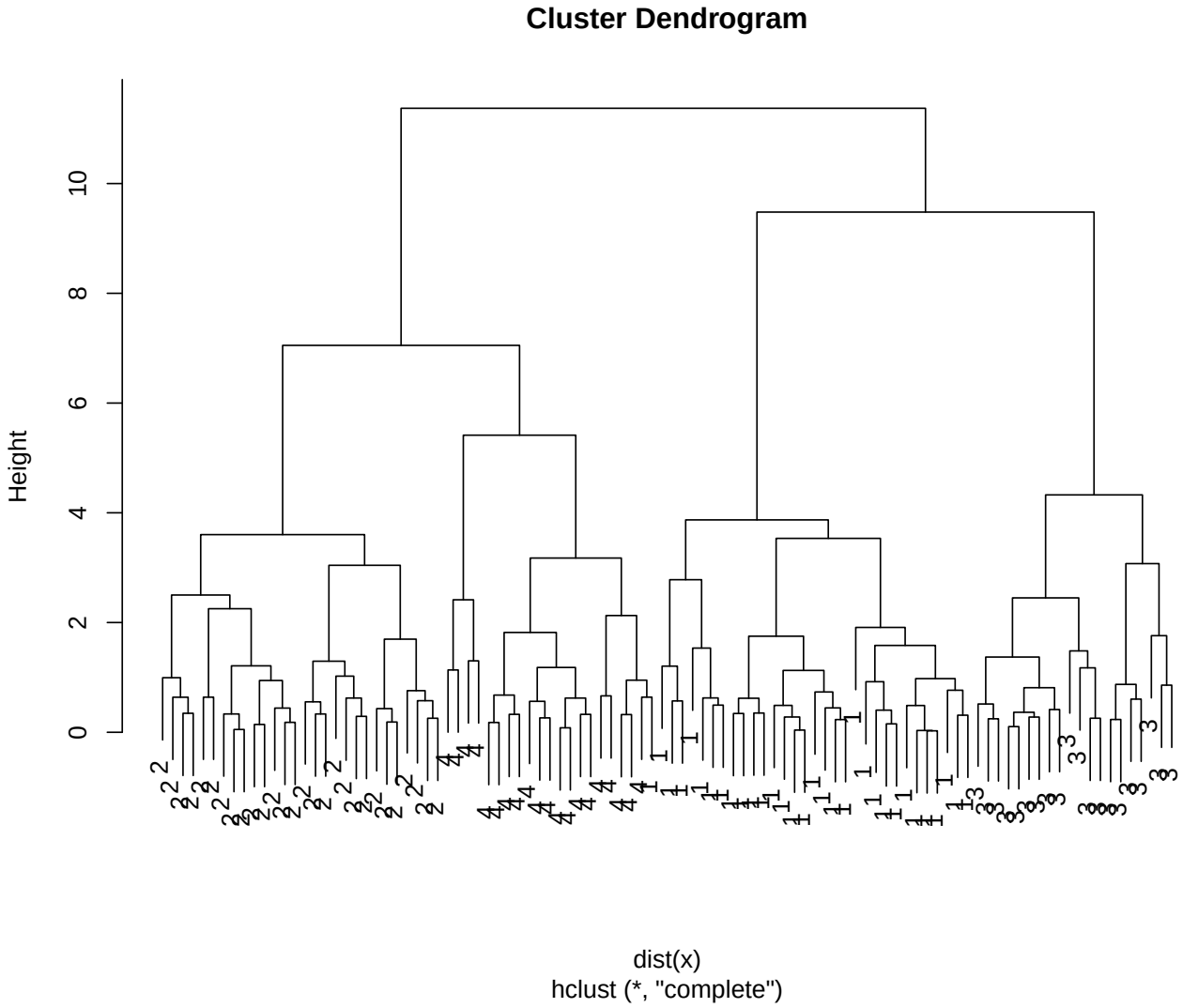

```
hc.single <- hclust(dist(x), method = "single")
plot(hc.single)
```




```
hc.average <- hclust(dist(x), method = "average")
plot(hc.average)
```




```
plot(hc.complete, labels = which)
```



Model-based

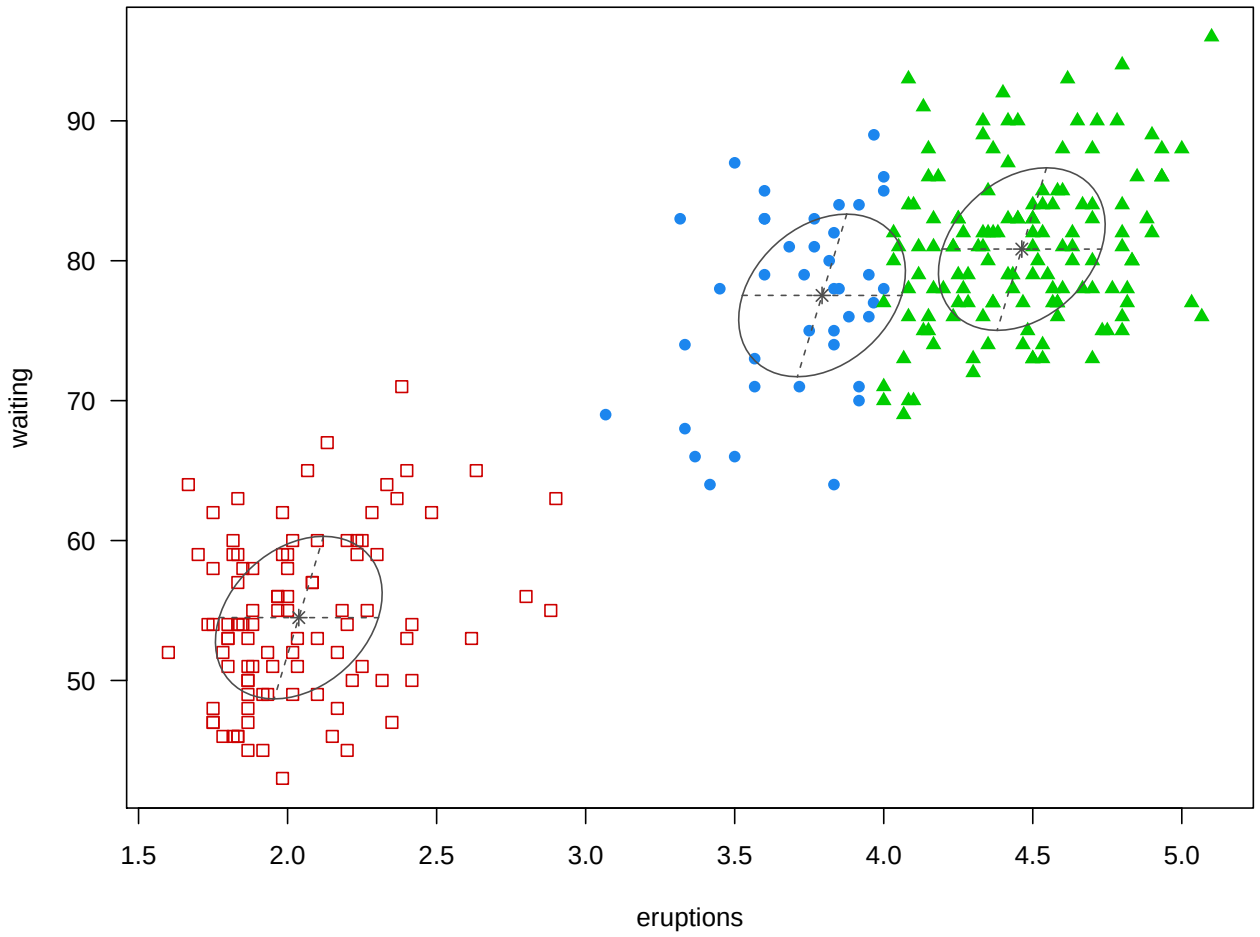
```
library(mclust)
```

```
## Package 'mclust' version 5.4.5
## Type 'citation("mclust")' for citing this R package in publications.
```

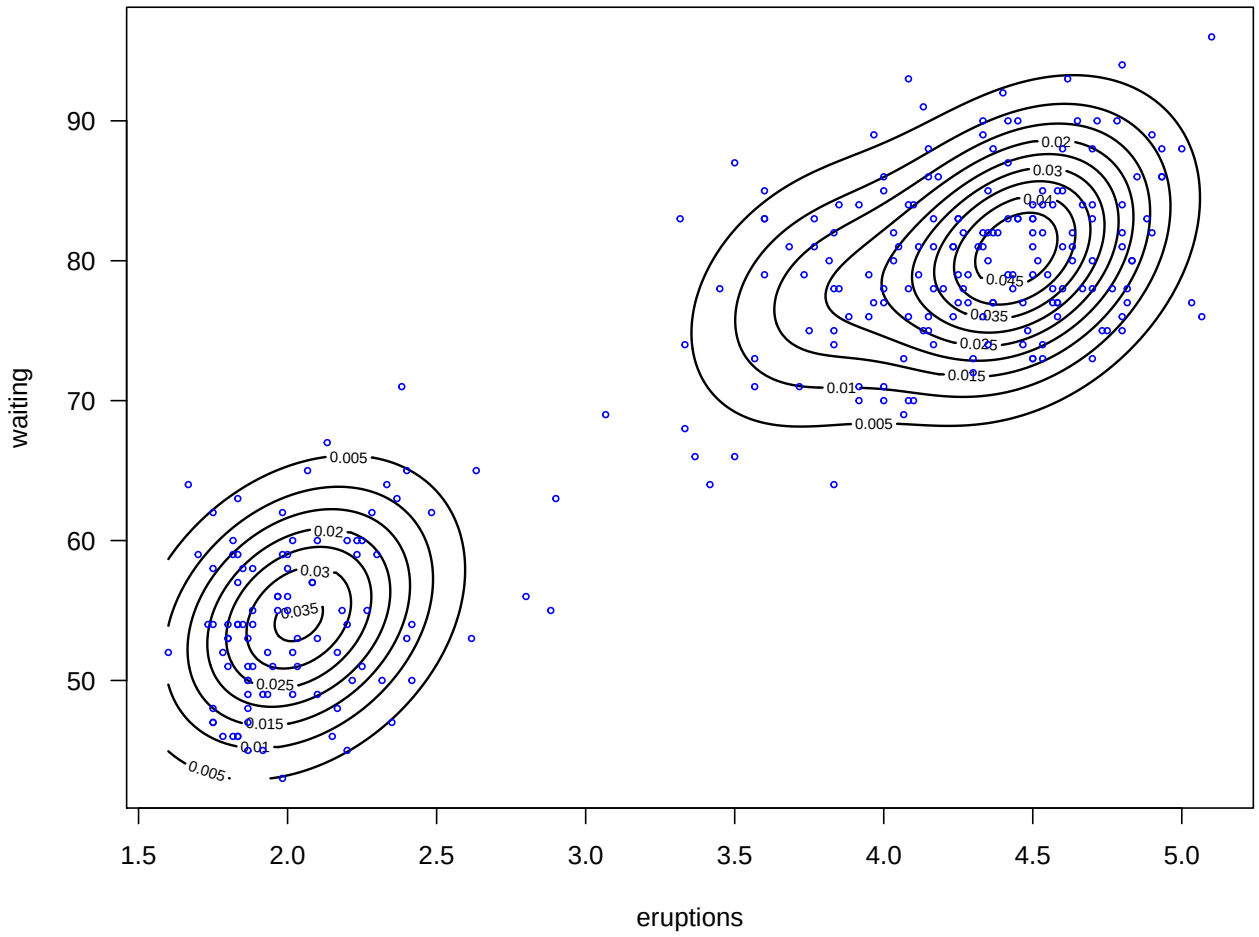
```
##
## Attaching package: 'mclust'
```

```
## The following object is masked from 'package:maps':
##
##     map
```

```
BIC <- mclustBIC(faithful)
model1 <- Mclust(faithful, x = BIC)


plot(model1, what = "classification", cex = 0.5, las = 1)
```




```
plot(model1, what = "density", col = "black", lwd = 1.5, las = 1)
points(faithful, col = "blue", cex = 0.5)
```




```
(LRT <- mclustBootstrapLRT(faithful, modelName = "VVV"))
```

```
## ---------------------------------------------------------
## Bootstrap sequential LRT for the number of mixture components
## ---------------------------------------------------------
## Model        = VVV
## Replications = 999
##                LRTS bootstrap p-value
## 1 vs 2   319.065354             0.001
## 2 vs 3     6.130516             0.564
```

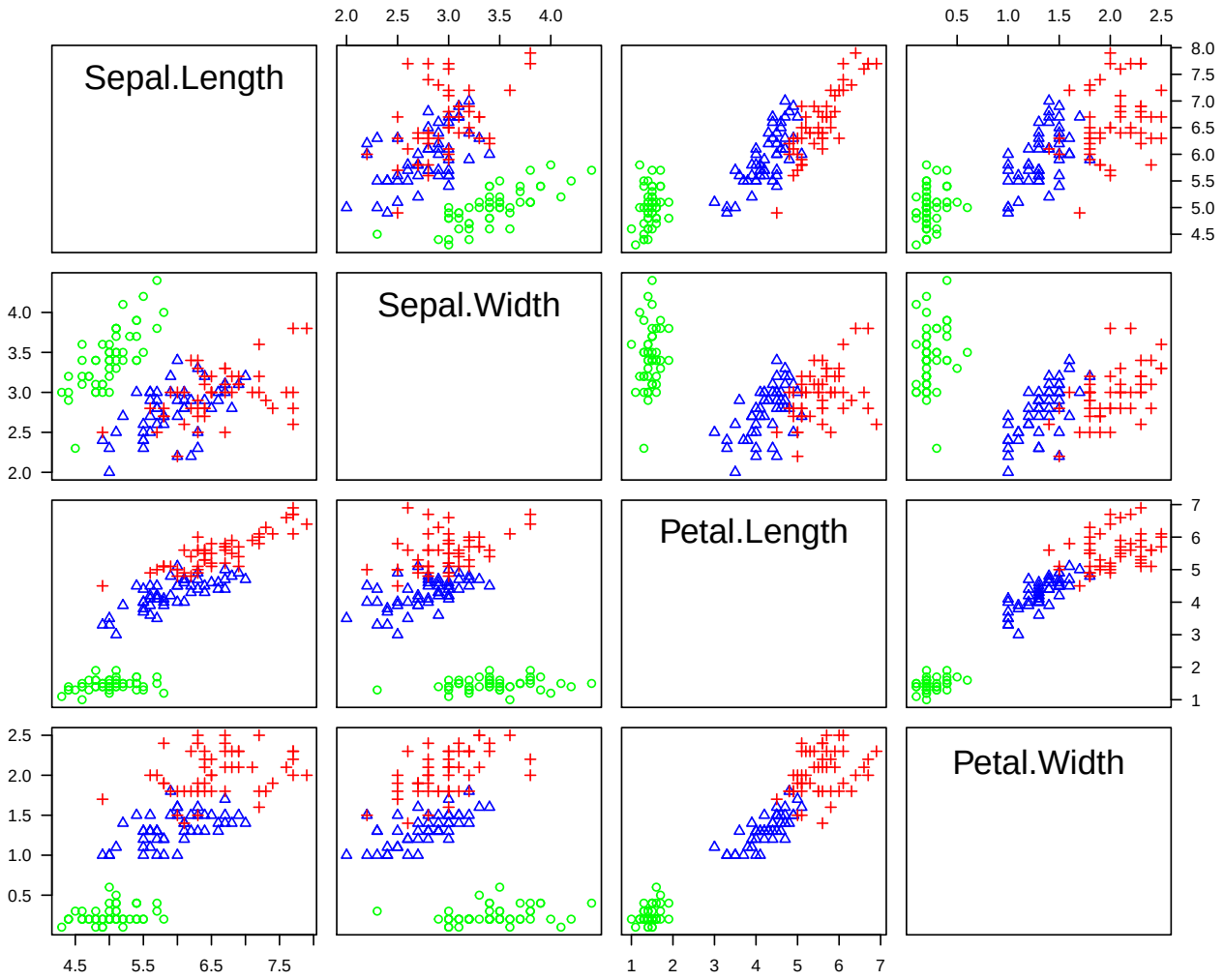
```
data(iris)
attach(iris)
```

```
## The following objects are masked from irisv:
##
##     Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species
```

```
## The following objects are masked from iris (pos = 15):
##
##     Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species
```

```
iris$Species <- factor(iris$Species)
dat <- iris[, 1:4]
BIC <- mclustBIC(dat)
model2 <- Mclust(dat, x = BIC)
```

```
par(las = 1)
scatterplotMatrix(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width | Species, col = c("green",
```




```
dev.off()
```

```
## null device
##           1
```

```
pdf("Exam4_cluster2.pdf", 6, 6)
par(las = 1)
plot(model2, what = "classification", cex = 0.5, col = c("green", "blue"))
```