

DSA 8020 R Session 12: Classification and Cluster Analysis

Whitney

April 01, 2021

Contents

Classification	1
Iris data	1
Binary classification	2
PCA	3
LDA	4
LDA vs. QDA	5
Logistic Regression	6
Clustering	8
K-Means Clustering	8
Geyser Example	9
Model-based	10

Classification

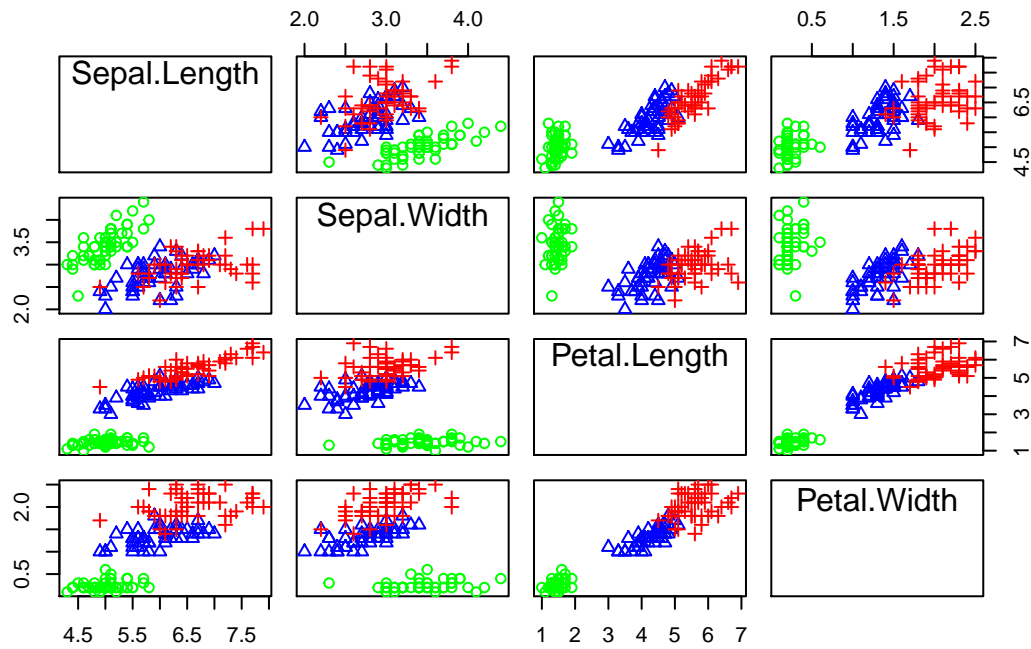
Iris data

```
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)

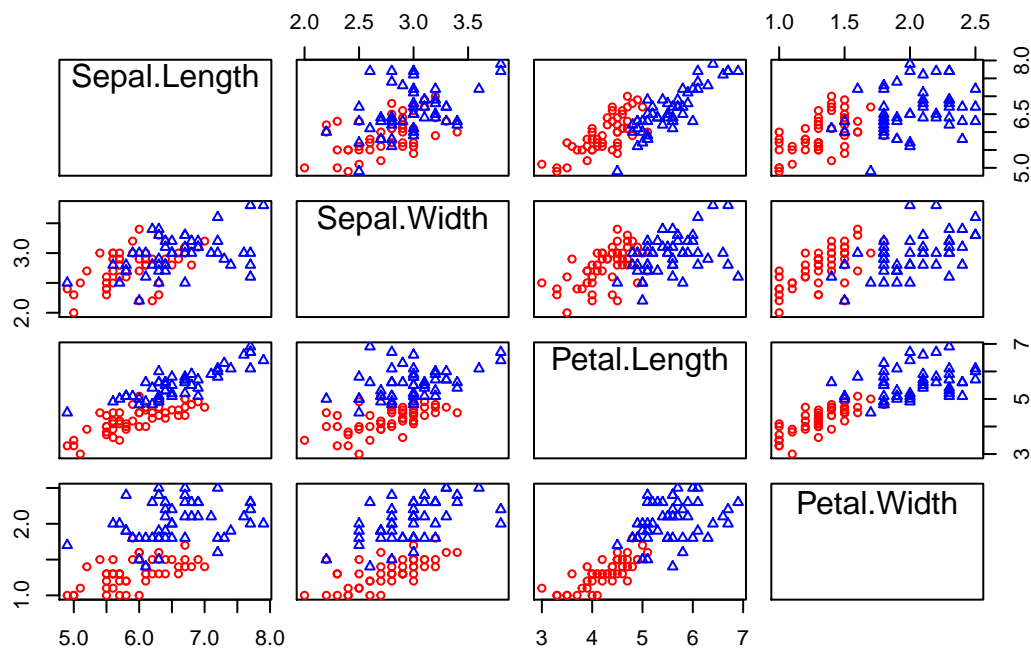
library(car)
scatterplotMatrix(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width | Species,
                  col = c("green", "blue", "red"), diagonal = F,
                  smooth = F, regLine = F, legend = F)
```



Binary classification

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

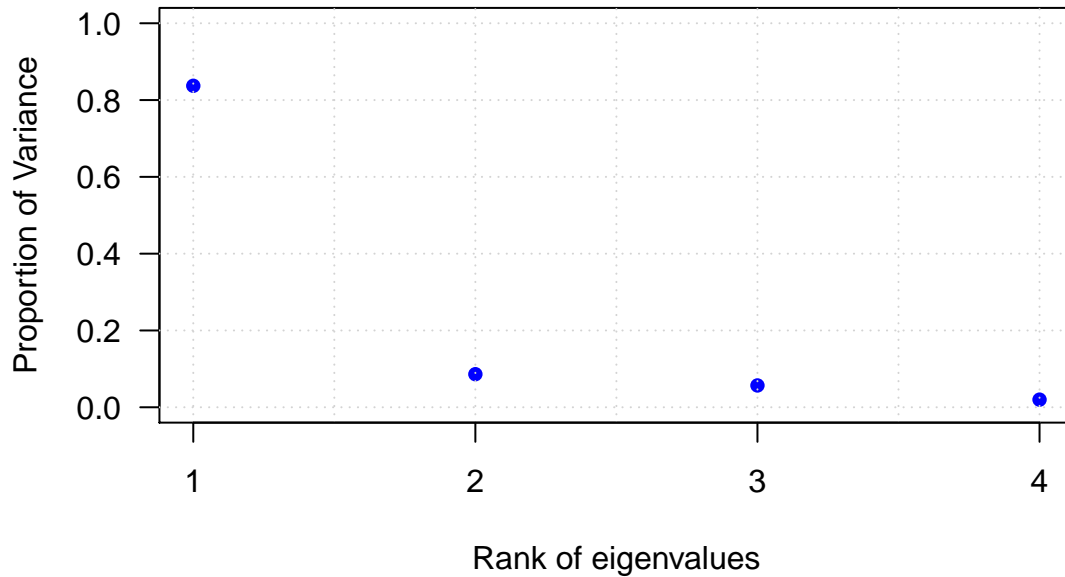
scatterplotMatrix(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width | Species,
  col = c("red", "blue"), diagonal = F,
  smooth = F, regLine = F, legend = F, cex = 0.75)
```



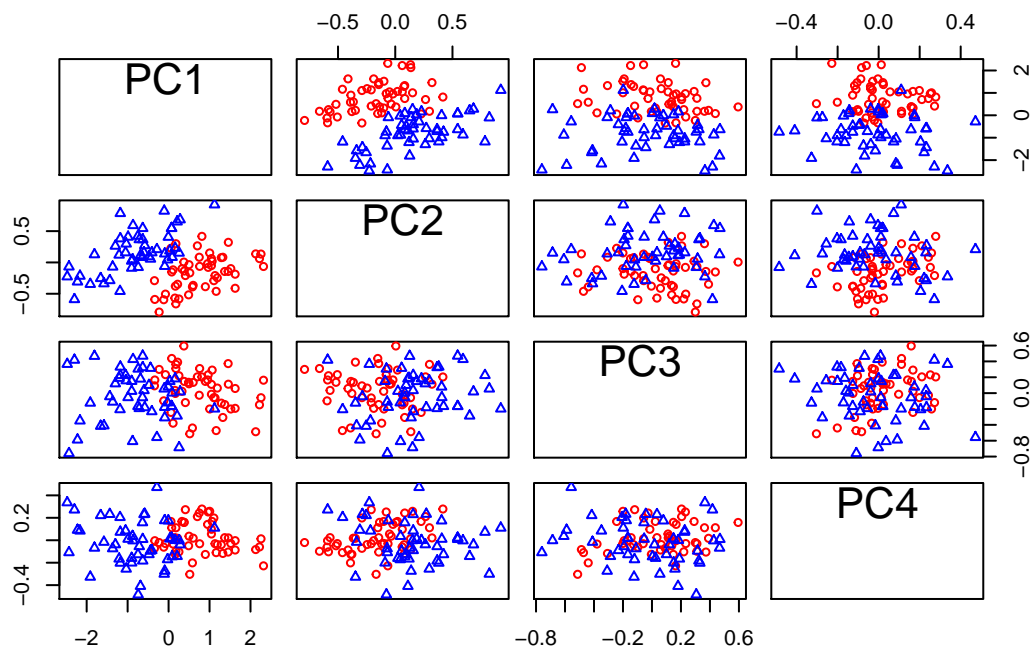
PCA

```
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)
```

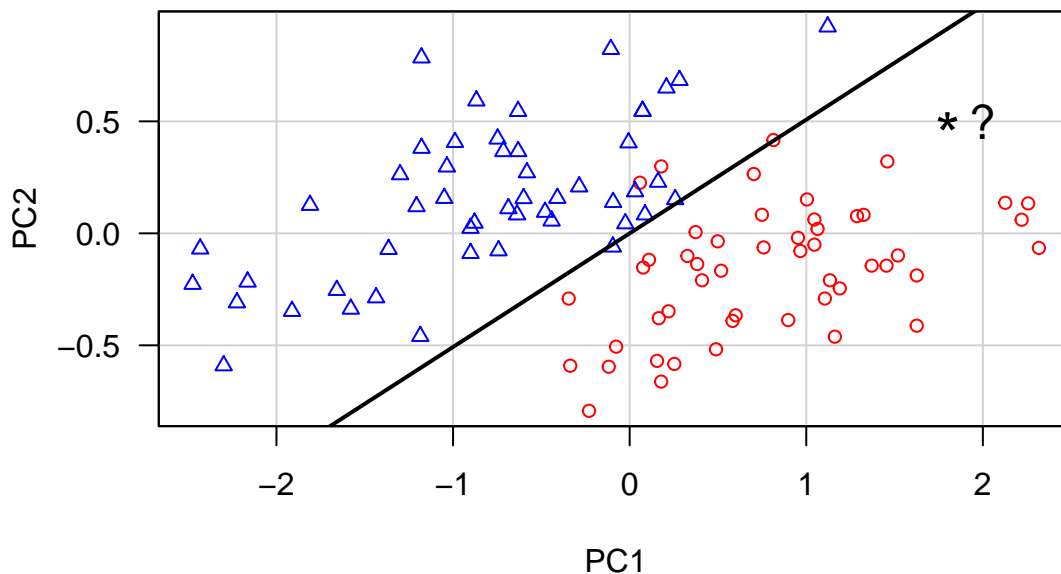


LDA

```
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 vs. 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
```

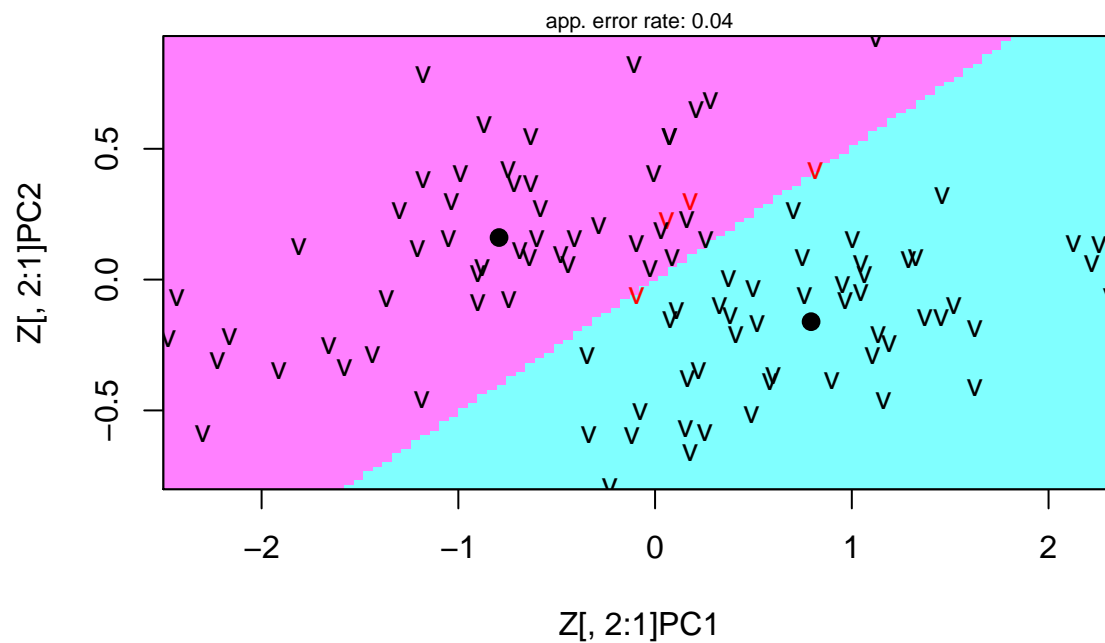
```
## virginica       2          48
```

```
# show results
```

```
library(klaR)
```

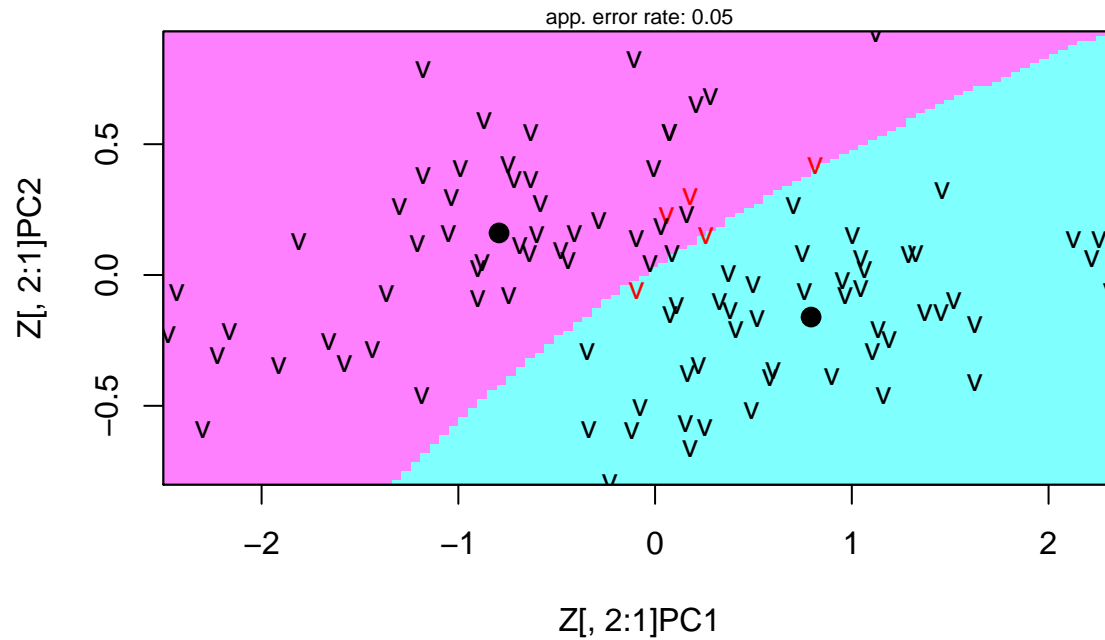
```
partimat(Species ~ Z[, 2:1], method = "lda", prec = 100, pch = 16, xaxt = "")
```

Partition Plot



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

Partition Plot

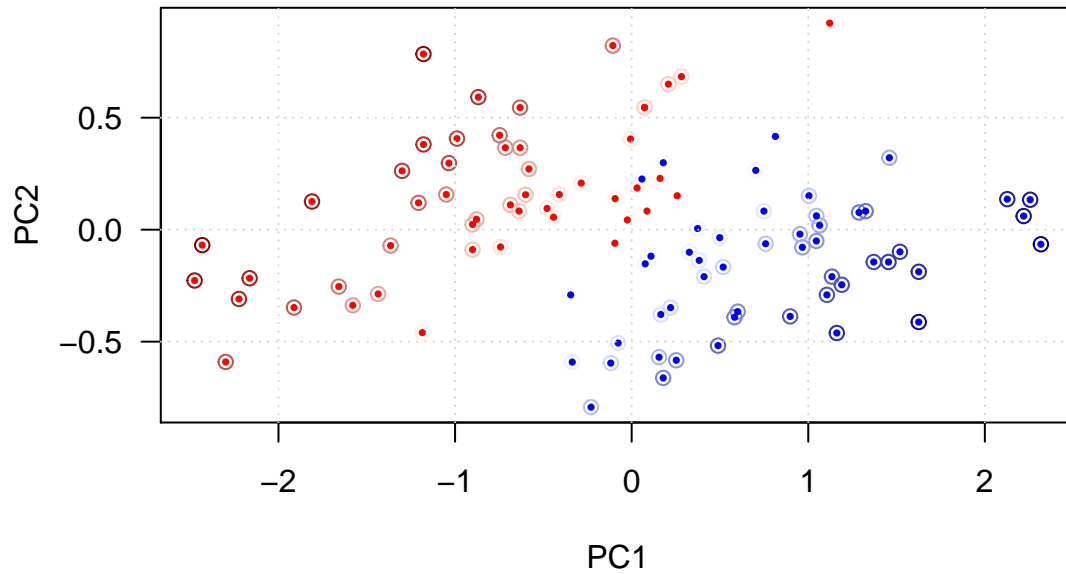


Logistic Regression

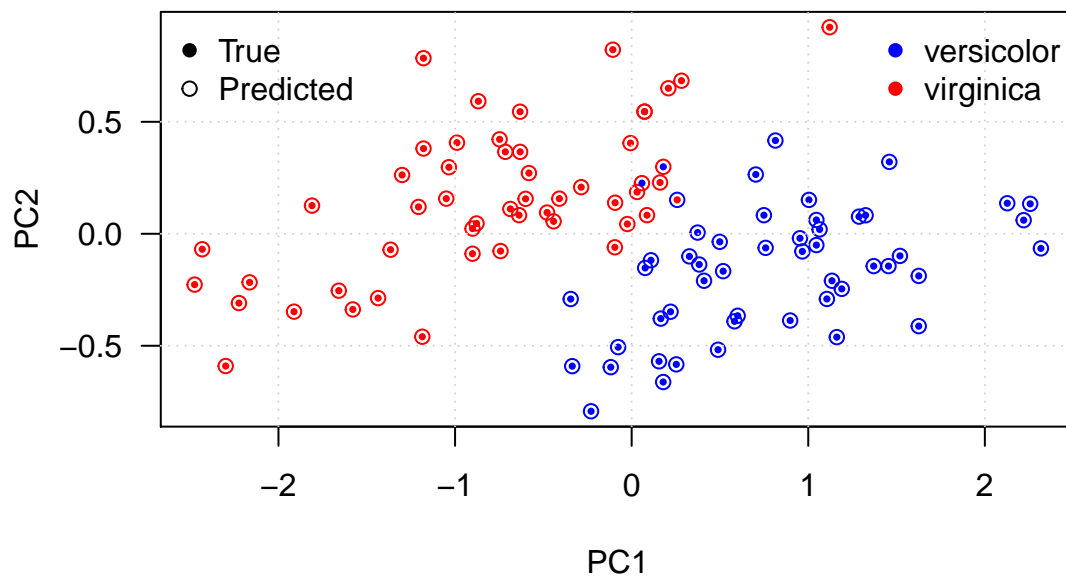
```
logfit <- glm(irisv$Species ~ z[, 1:2], family = binomial)
logpred <- predict(logfit, type = "response")
library(fields)
cols <- two.colors(n = 100, "darkblue", "darkred")
order <- order(logpred)

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")
```



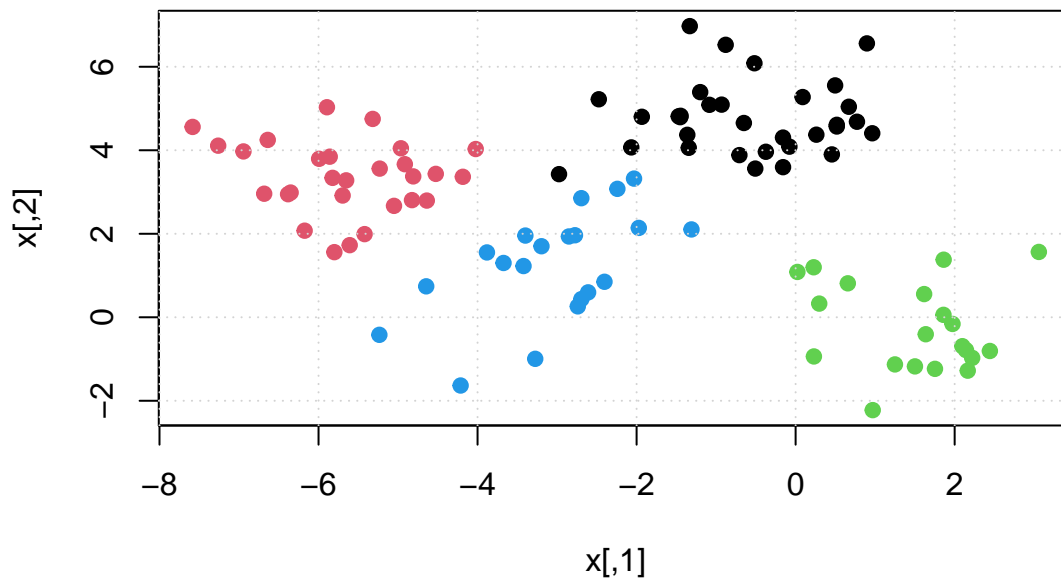
```
logisticPred <- ifelse(logpred <= 0.5, "versicolor", "virginica")
table(irisv$Species, logisticPred)
```

```
##          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()
```



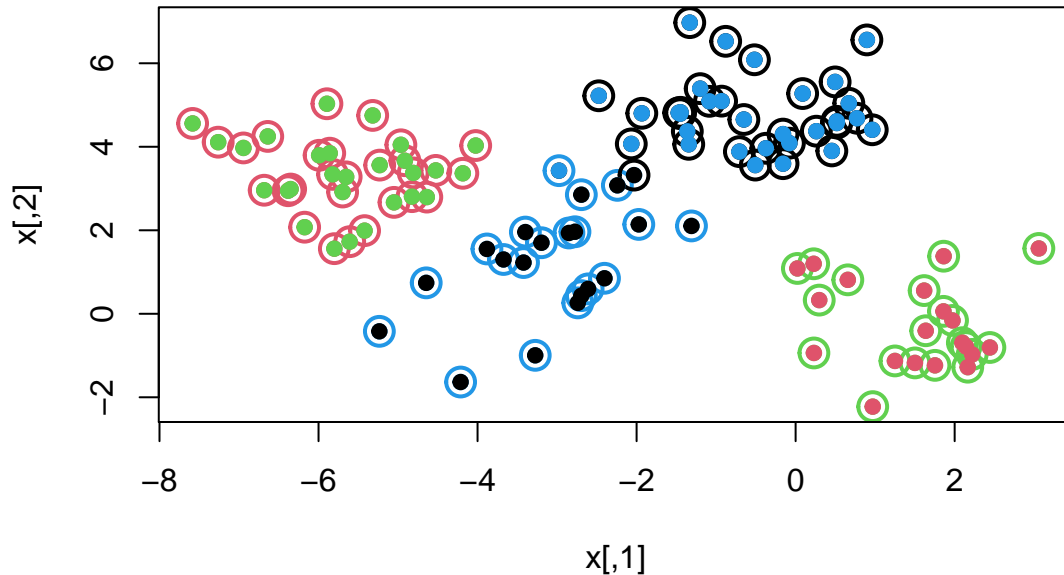
```
km.out <- kmeans(x, 4, nstart = 15)
km.out
```

```
## 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"       "
```

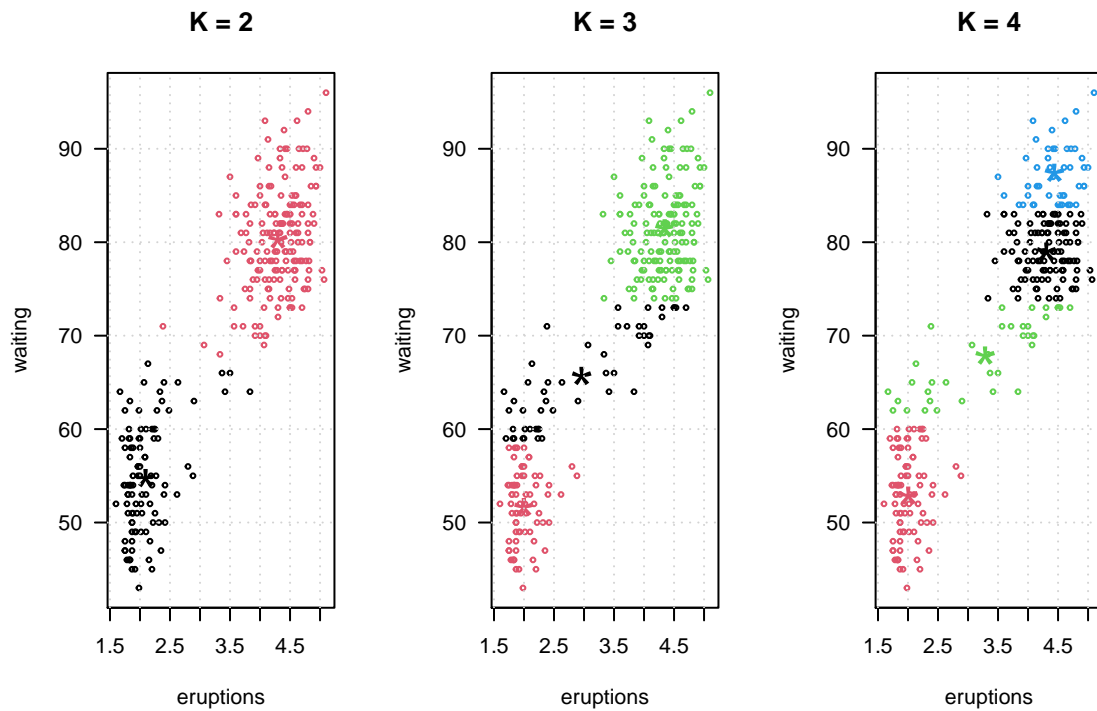
```
plot(x, col = km.out$cluster, cex = 2, pch = 1, lwd = 2)
points(x, col = which, pch = 19)
points(x, col = c(4, 3, 2, 1)[which], pch = 19)
```



Geyser Example

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

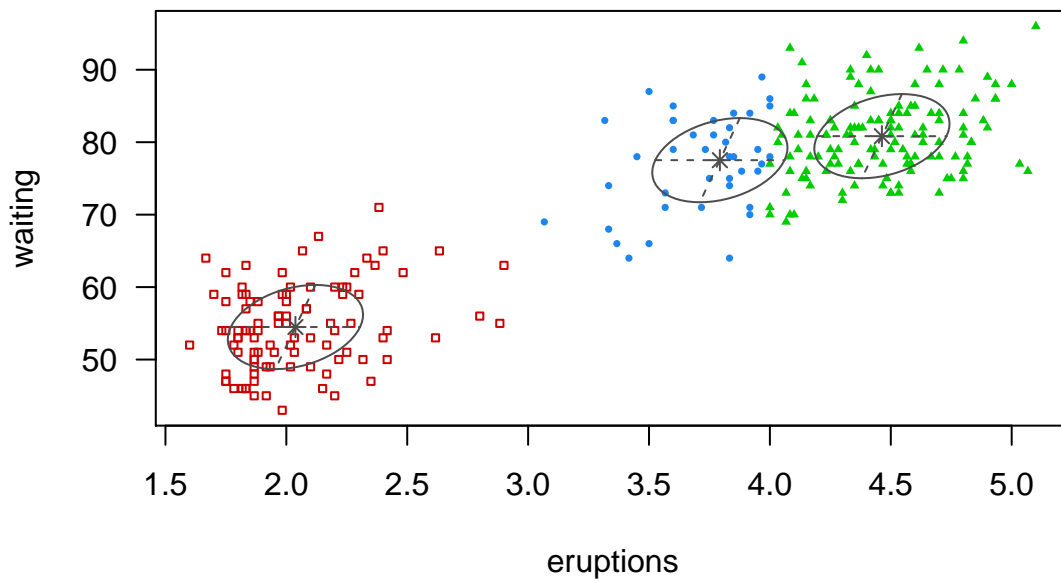
par(las = 1, mfrow = c(1, 3))
plot(fairful, col = km2.fairful$cluster, cex = 0.5, main = "K = 2")
points(km2.fairful$centers, cex = 3, pch = "*", col = 1:2)
grid()
plot(fairful, col = km3.fairful$cluster, cex = 0.5, main = "K = 3")
points(km3.fairful$centers, cex = 3, pch = "*", col = 1:3)
grid()
plot(fairful,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



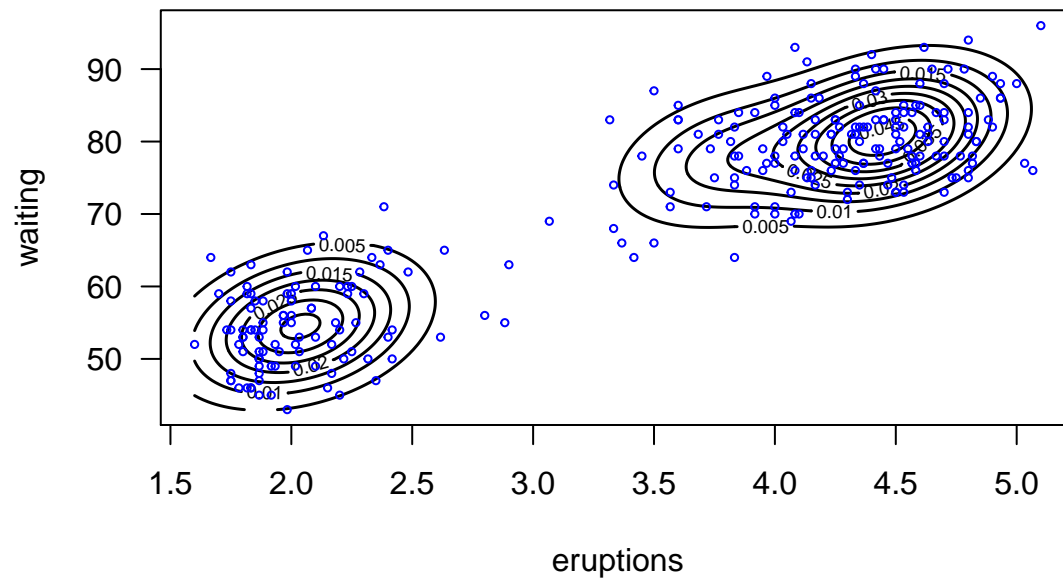
Model-based

```
library(mclust)
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          0.549
```