Machine Learning

Ravi Kumar Tiwari

14 June 2016

```
library(caret)
library(rpart.plot)
library(rattle)
library(calibrate)
library(randomForest)
library(e1071)
library(class)
library(knitr)
```

knn

Decision Tree Example

Problem Description

Given a data set that contains some observation and corresponding class label, can a machine learning algorithm be trained to determine the class label of any data set (not necessarily the data that was used for training) from its observation

Solution using decision tree

```
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
```

Create data partition

```
set.seed(100)
inTrain <- createDataPartition(iris$Species, p = 0.6, list = FALSE)
trainData <- iris[inTrain,]
testData <- iris[-inTrain,]
```

Build a decision tree model and use it for prediction on test data set

```
treeModel <- train(Species ~ ., data = trainData, method = "rpart")
preClass <- predict(treeModel, newdata = testData)
cMatrix <- confusionMatrix(preClass, testData$Species)
cMatrix$table
```

```
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         20          0         0
##   versicolor      0         19         2
##   virginica       0          1        18
```

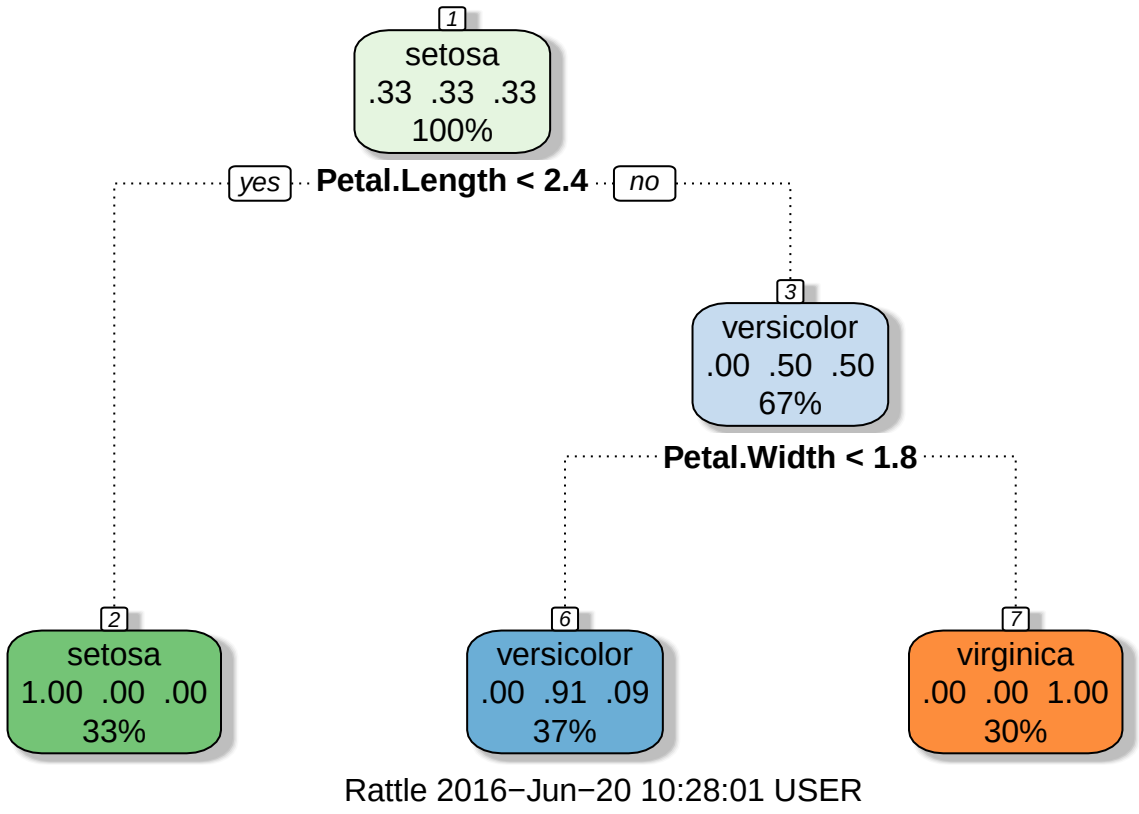
Look at what are the important variables

```
varImp(treeModel)
```

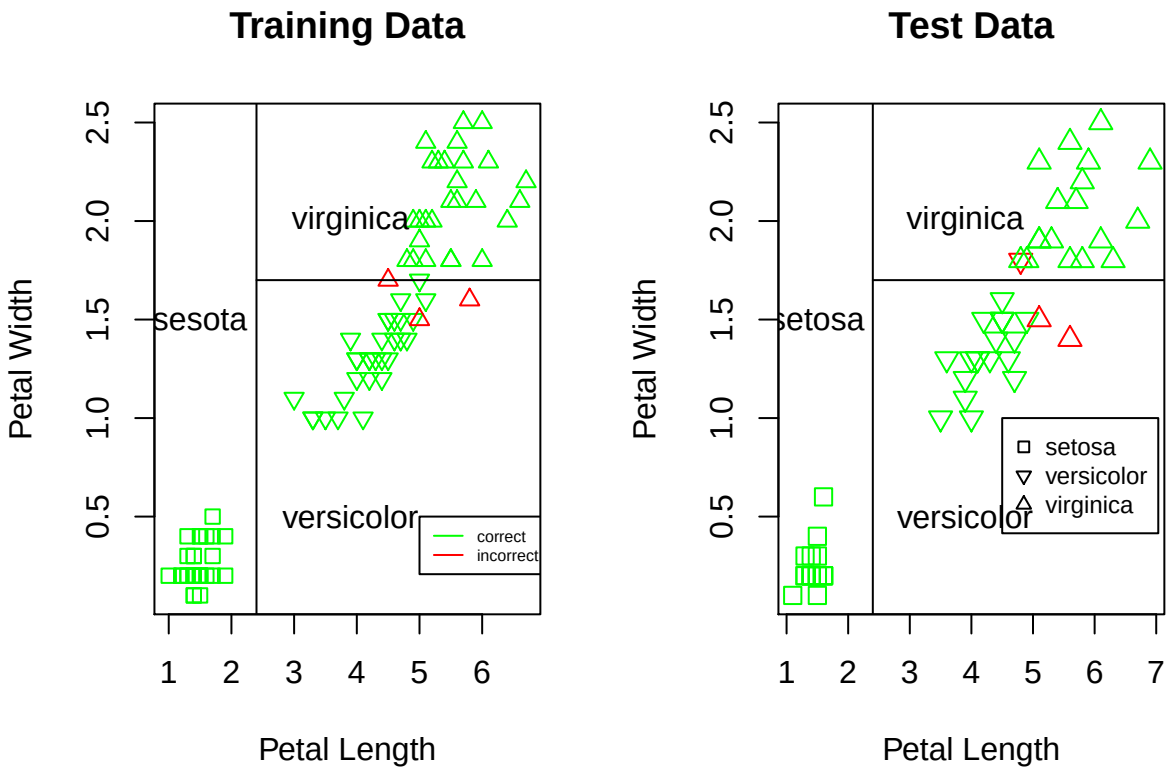
```
## rpart variable importance
##
##              Overall
## Petal.Width   100.00
## Petal.Length   89.53
## Sepal.Length   18.24
## Sepal.Width     0.00
```

Visualization of the decision tree

```
fancyRpartPlot(treeModel$finalModel)
```

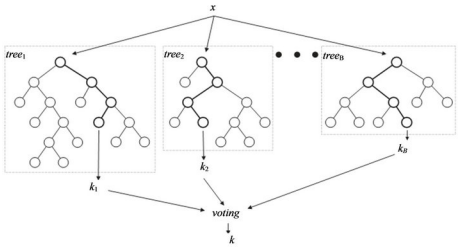


Visualization of the decision tree



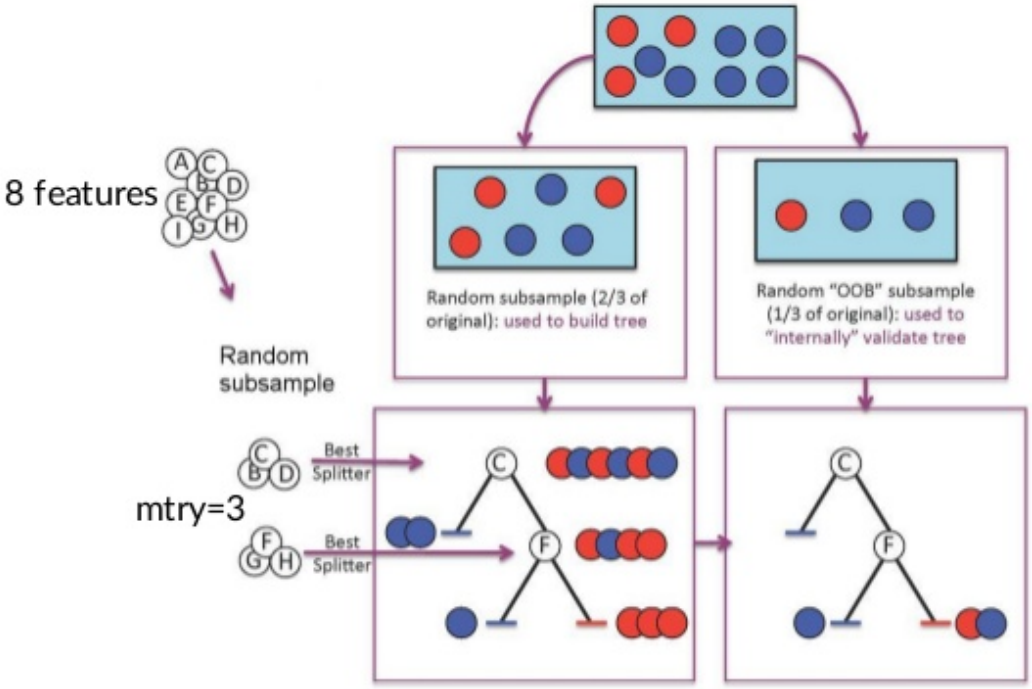
random Forest

ntree

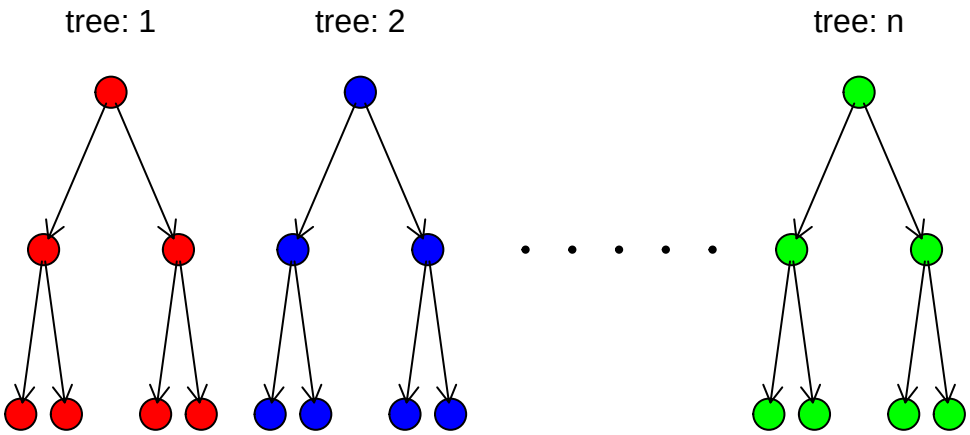


Random Forest classifier

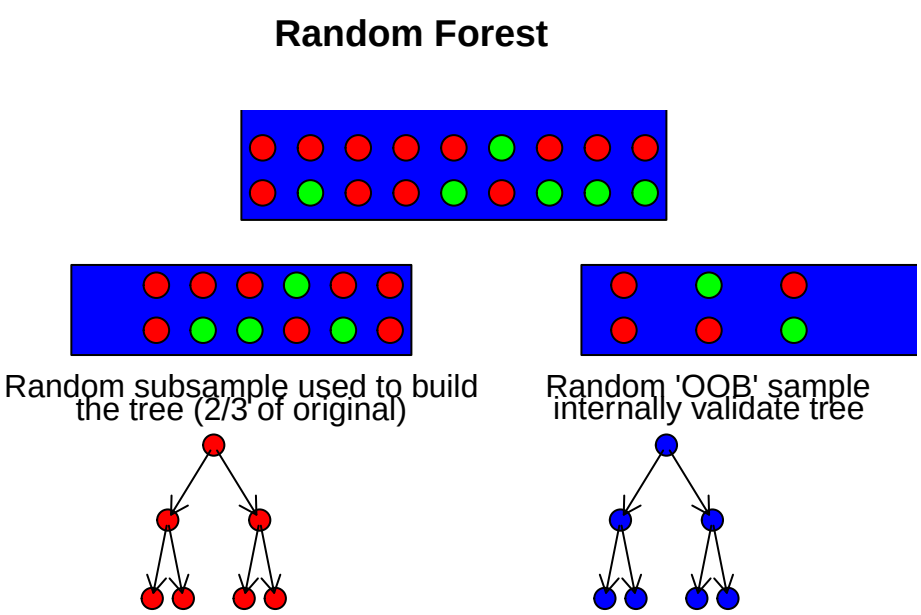
Individual tree



Random Forest



mtry


Random Forest
Random subsample used to build the tree (2/3 of original)
Random 'OOB' sample internally validate tree

knn2

```
myIris <- iris[,3:5]
head(myIris)
```

```
##   Petal.Length Petal.Width Species
## 1          1.4         0.2  setosa
## 2          1.4         0.2  setosa
## 3          1.3         0.2  setosa
## 4          1.5         0.2  setosa
## 5          1.4         0.2  setosa
## 6          1.7         0.4  setosa
```

```
nI <- nrow(myIris)
ind <- sample(1:nI, 0.8*nI)
trainData <- myIris[ind, 1:2]
trainClass <- myIris[ind, 3]
testData <- myIris[-ind, 1:2]
testClass <- myIris[-ind, 3]
preClass <- knn(trainData, testData, cl = trainClass, k = 2)
table(preClass, testClass)
```

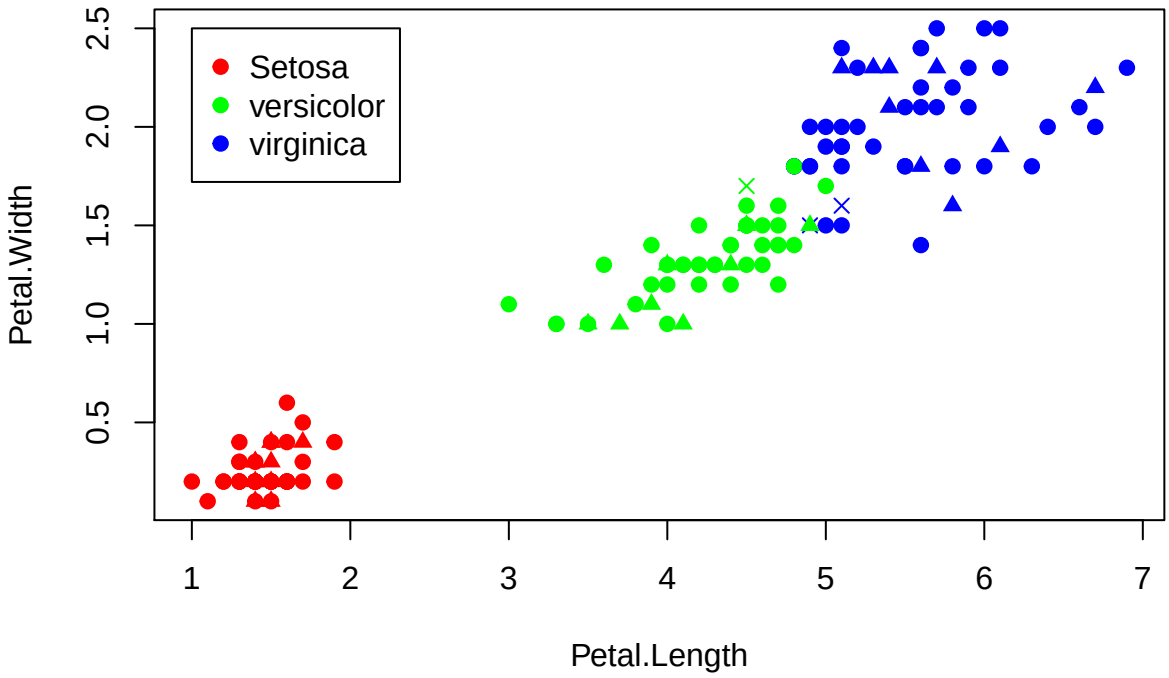
```
##             testClass
## preClass     setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0          8         1
##   virginica       0          2         9
```

```
color <- ifelse(trainClass=="setosa", "red", ifelse(trainClass=="versicolor", "green",
                "blue"))

plot(trainData$Petal.Length, trainData$Petal.Width, pch = 19, col = color,
     xlab = "Petal.Length", ylab = "Petal.Width")
legend(x = 1, y = 2.5, legend = c("Setosa", "versicolor", "virginica"),
       col = c("red", "green", "blue"), pch = 19)

color <- ifelse(preClass=="setosa", "red", ifelse(preClass=="versicolor", "green",
                "blue"))

pType = ifelse(preClass == testClass, 17, 4)
points(testData$Petal.Length, testData$Petal.Width, pch = pType, col = color)
```


Setosa
versicolor
virginica
Petal.Width
Petal.Length
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

clustering example

kmeans clustering

```
myIris <- iris[3:4]
group <- iris$Species
predGroup <- kmeans(myIris, centers = 3)
predGroupC <- ifelse(predGroup$cluster==2, "setosa", ifelse(predGroup$cluster==3,
                                                            "versicolor", "virginnica"))
predGroupC <- factor(predGroupC)
predGroupC
```

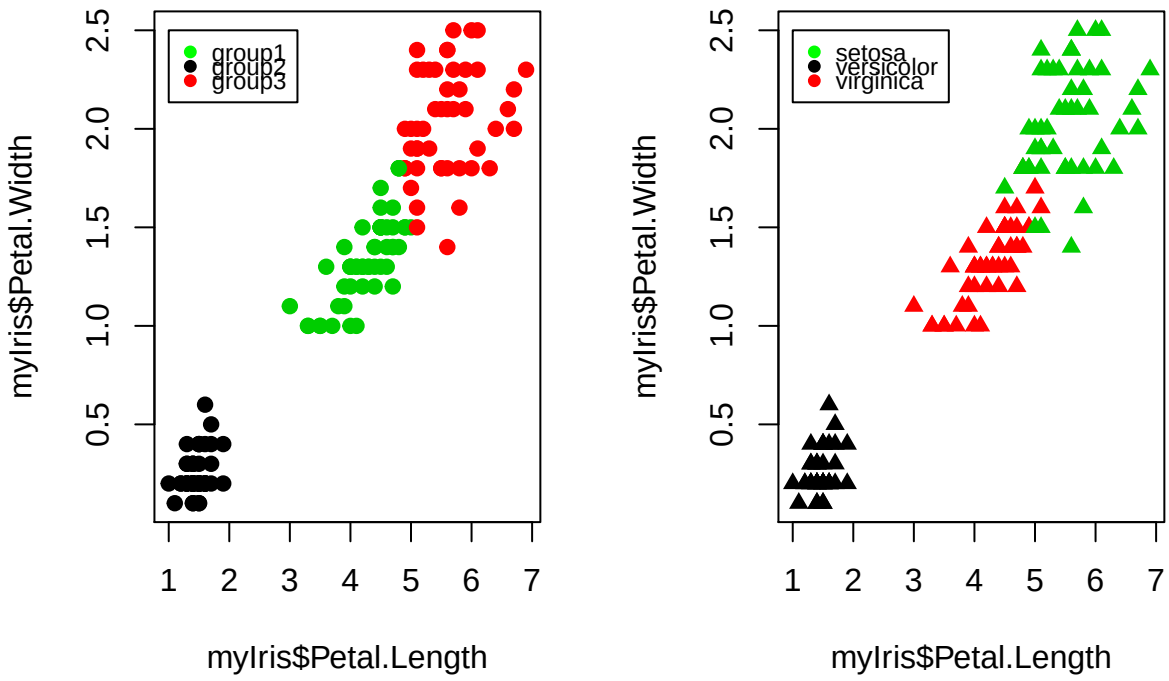
```
##   [1] setosa     setosa     setosa     setosa     setosa     setosa
##   [7] setosa     setosa     setosa     setosa     setosa     setosa
##  [13] setosa     setosa     setosa     setosa     setosa     setosa
##  [19] setosa     setosa     setosa     setosa     setosa     setosa
##  [25] setosa     setosa     setosa     setosa     setosa     setosa
##  [31] setosa     setosa     setosa     setosa     setosa     setosa
##  [37] setosa     setosa     setosa     setosa     setosa     setosa
##  [43] setosa     setosa     setosa     setosa     setosa     setosa
##  [49] setosa     setosa     virginnica virginnica virginnica virginnica
##  [55] virginnica virginnica virginnica virginnica virginnica virginnica
##  [61] virginnica virginnica virginnica virginnica virginnica virginnica
##  [67] virginnica virginnica virginnica virginnica virginnica virginnica
##  [73] virginnica virginnica virginnica virginnica virginnica versicolor
##  [79] virginnica virginnica virginnica virginnica virginnica versicolor
##  [85] virginnica virginnica virginnica virginnica virginnica virginnica
##  [91] virginnica virginnica virginnica virginnica virginnica virginnica
##  [97] virginnica virginnica virginnica virginnica versicolor versicolor
## [103] versicolor versicolor versicolor versicolor virginnica versicolor
## [109] versicolor versicolor versicolor versicolor versicolor versicolor
## [115] versicolor versicolor versicolor versicolor versicolor virginnica
## [121] versicolor versicolor versicolor versicolor versicolor versicolor
## [127] virginnica versicolor versicolor versicolor versicolor versicolor
## [133] versicolor versicolor versicolor versicolor versicolor versicolor
## [139] virginnica versicolor versicolor versicolor versicolor versicolor
## [145] versicolor versicolor versicolor versicolor versicolor versicolor
## Levels: setosa versicolor virginnica
```

```
table(predGroupC, group)
```

```
##             group
## predGroupC   setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0          2        46
##   virginnica      0         48         4
```

```
par(mfrow = c(1,2))
plot(myIris$Petal.Length, myIris$Petal.Width, pch = 19, col = predGroupC)
legend(x=1,y=2.5, legend = c("group1", "group2", "group3"),
       col = c("green", "black", "red"), pch = 19, y.intersp=0.5, cex = 0.75)
```

```
plot(myIris$Petal.Length, myIris$Petal.Width, pch = 17, col = group)
legend(x=1,y=2.5, legend = c("setosa", "versicolor", "virginica"),
       col = c("green", "black", "red"), pch = 19, y.intersp=0.5, cex = 0.75)
```




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

hierarchichal clustering

```
group <- iris$Species
disM <- dist(myIris)
irisClust <- hclust(disM)
clusters <- cutree(irisClust, k = 3)

clusters <- ifelse(clusters==1, "setosa", ifelse(clusters==2,
                                                 "virginnica", "versicolor"))
clusters <- factor(clusters)
clusters
```

```
##   [1] setosa     setosa     setosa     setosa     setosa     setosa
##   [7] setosa     setosa     setosa     setosa     setosa     setosa
##  [13] setosa     setosa     setosa     setosa     setosa     setosa
##  [19] setosa     setosa     setosa     setosa     setosa     setosa
```

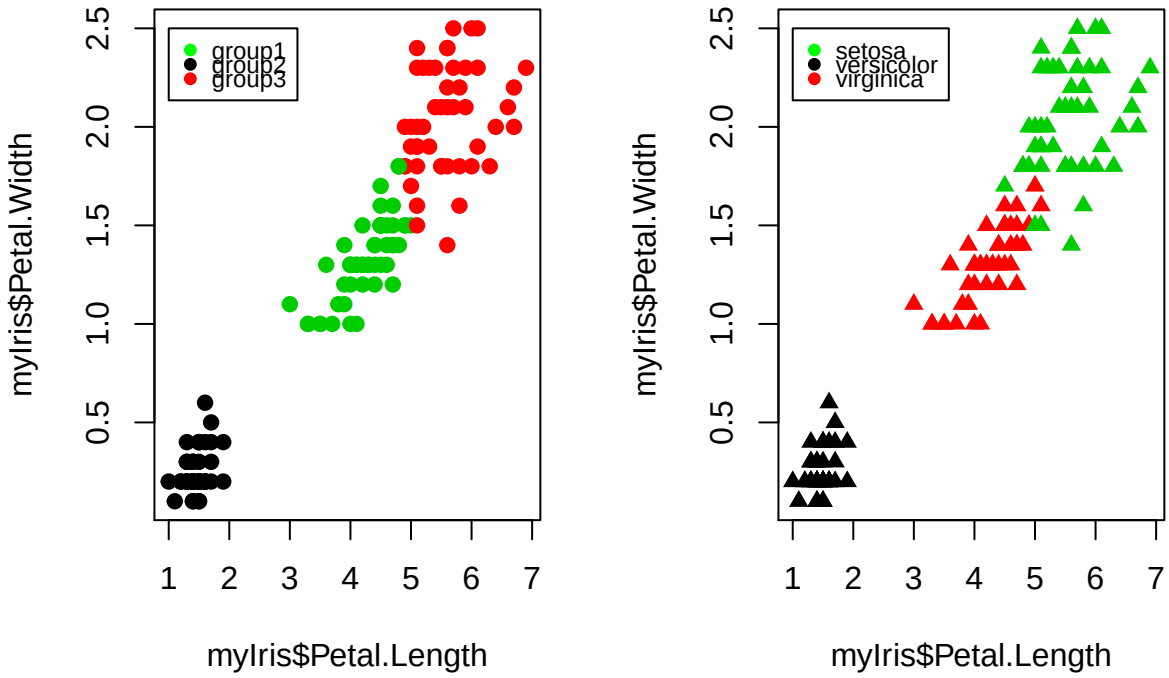
```
##  [25] setosa     setosa     setosa     setosa     setosa     setosa
##  [31] setosa     setosa     setosa     setosa     setosa     setosa
##  [37] setosa     setosa     setosa     setosa     setosa     setosa
##  [43] setosa     setosa     setosa     setosa     setosa     setosa
##  [49] setosa     setosa     virginnica virginnica virginnica versicolor
##  [55] virginnica virginnica virginnica versicolor virginnica versicolor
##  [61] versicolor versicolor versicolor virginnica versicolor versicolor
##  [67] virginnica versicolor virginnica versicolor virginnica versicolor
##  [73] virginnica virginnica versicolor versicolor virginnica virginnica
##  [79] virginnica versicolor versicolor versicolor versicolor virginnica
##  [85] virginnica virginnica virginnica versicolor versicolor versicolor
##  [91] versicolor virginnica versicolor versicolor versicolor versicolor
##  [97] versicolor versicolor versicolor versicolor virginnica virginnica
## [103] virginnica virginnica virginnica virginnica virginnica virginnica
## [109] virginnica virginnica virginnica virginnica virginnica virginnica
## [115] virginnica virginnica virginnica virginnica virginnica virginnica
## [121] virginnica virginnica virginnica virginnica virginnica virginnica
## [127] virginnica virginnica virginnica virginnica virginnica virginnica
## [133] virginnica virginnica virginnica virginnica virginnica virginnica
## [139] virginnica virginnica virginnica virginnica virginnica virginnica
## [145] virginnica virginnica virginnica virginnica virginnica virginnica
## Levels: setosa versicolor virginnica
```

```
table(clusters, group)
```

```
##             group
## clusters     setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         29         0
##   virginnica      0         21        50
```

```
par(mfrow = c(1,2))
plot(myIris$Petal.Length, myIris$Petal.Width, pch = 19, col = predGroupC)
legend(x=1,y=2.5, legend = c("group1", "group2", "group3"),
       col = c("green", "black", "red"), pch = 19, y.intersp=0.5, cex = 0.75)

plot(myIris$Petal.Length, myIris$Petal.Width, pch = 17, col = group)
legend(x=1,y=2.5, legend = c("setosa", "versicolor", "virginica"),
       col = c("green", "black", "red"), pch = 19, y.intersp=0.5, cex = 0.75)
```



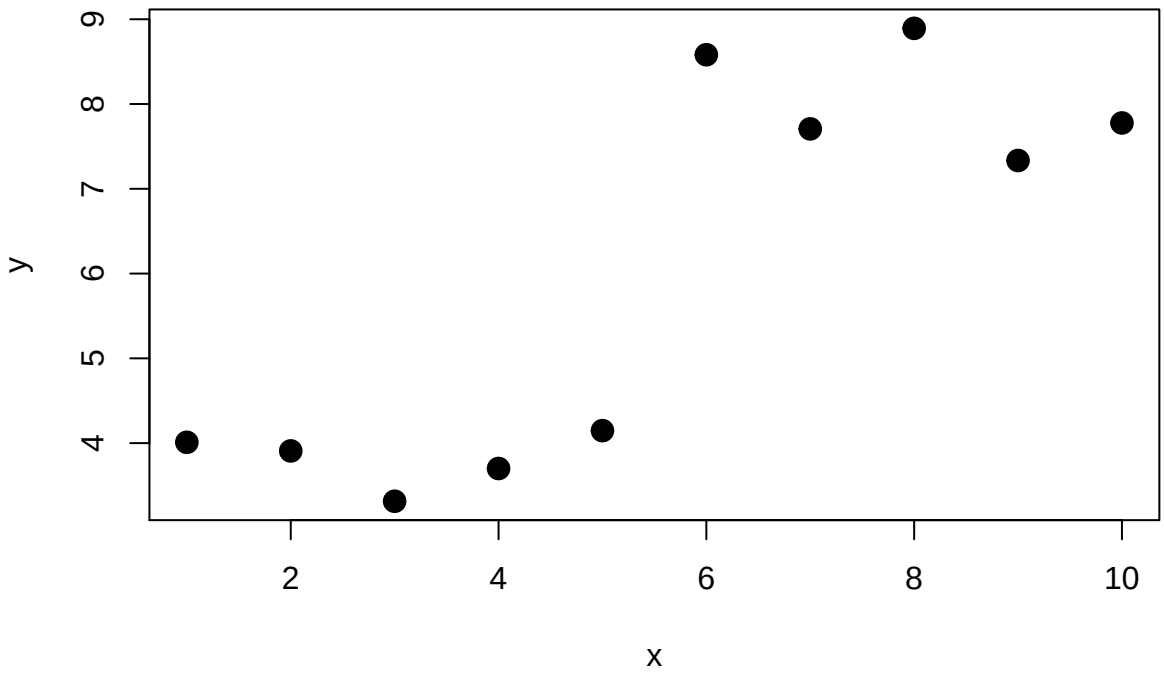

```
par(mfrow = c(1,1))
```

Clustering Example

K-means clustering

```
head(obs)
```

```
##   x        y
## 1 1 4.009373
## 2 2 3.907874
## 3 3 3.314335
## 4 4 3.700416
## 5 5 4.147273
## 6 6 8.581343
```



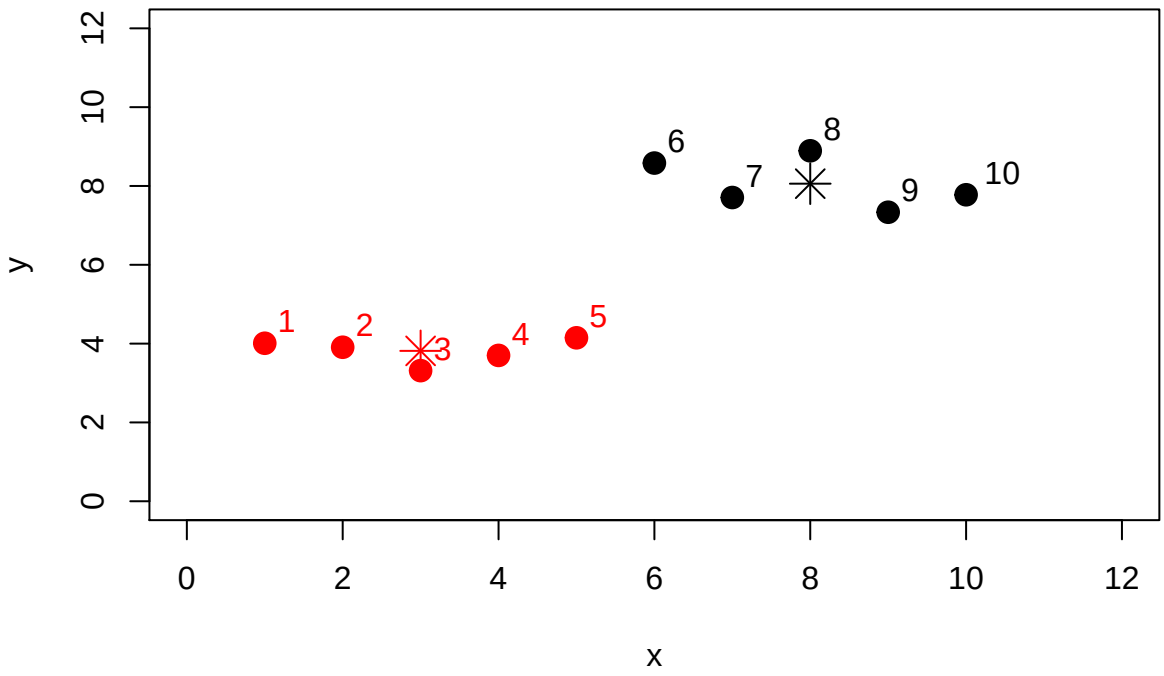
Create 2 clusters

```
kmeansObj <-  kmeans(obs, centers = 2)
data.frame(obs, cluster = kmeansObj$cluster)
```

```
##     x        y cluster
## 1   1 4.009373       2
## 2   2 3.907874       2
## 3   3 3.314335       2
## 4   4 3.700416       2
## 5   5 4.147273       2
## 6   6 8.581343       1
## 7   7 7.707038       1
## 8   8 8.892733       1
## 9   9 7.333703       1
## 10 10 7.776717       1
```

```
kmeansObj$centers
```

```
##   x        y
## 1 8 8.058307
## 2 3 3.815854
```



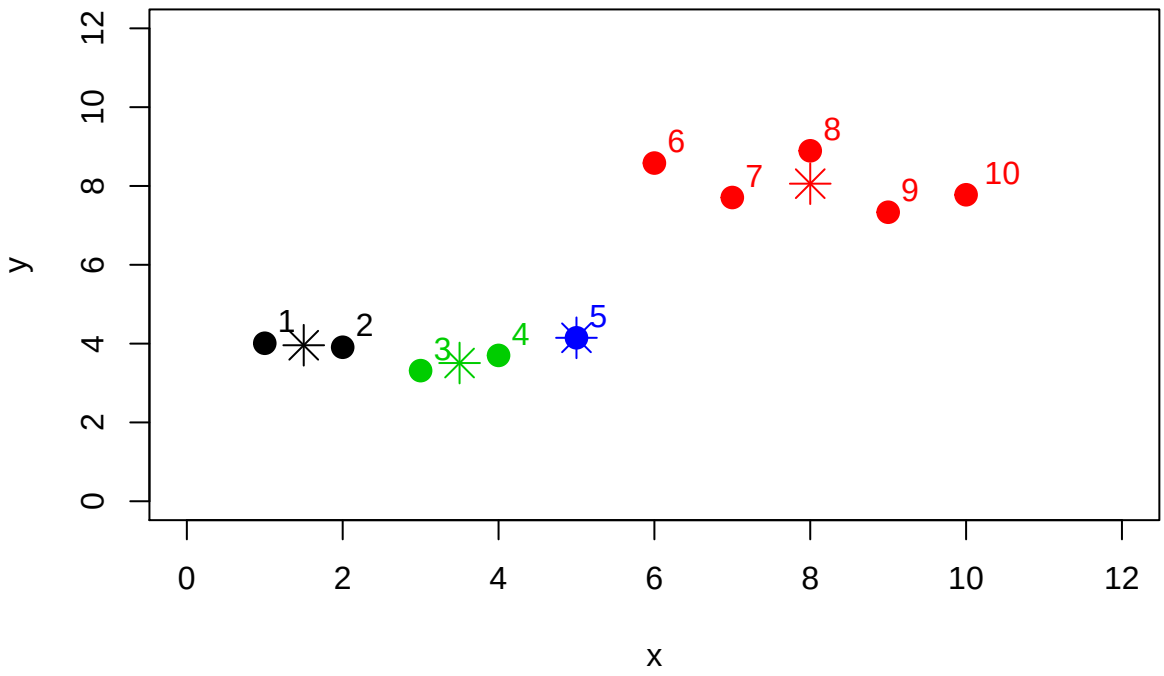
Create 4 clusters

```
kmeansObj <-  kmeans(obs, centers = 4)
data.frame(obs, cluster = kmeansObj$cluster)
```

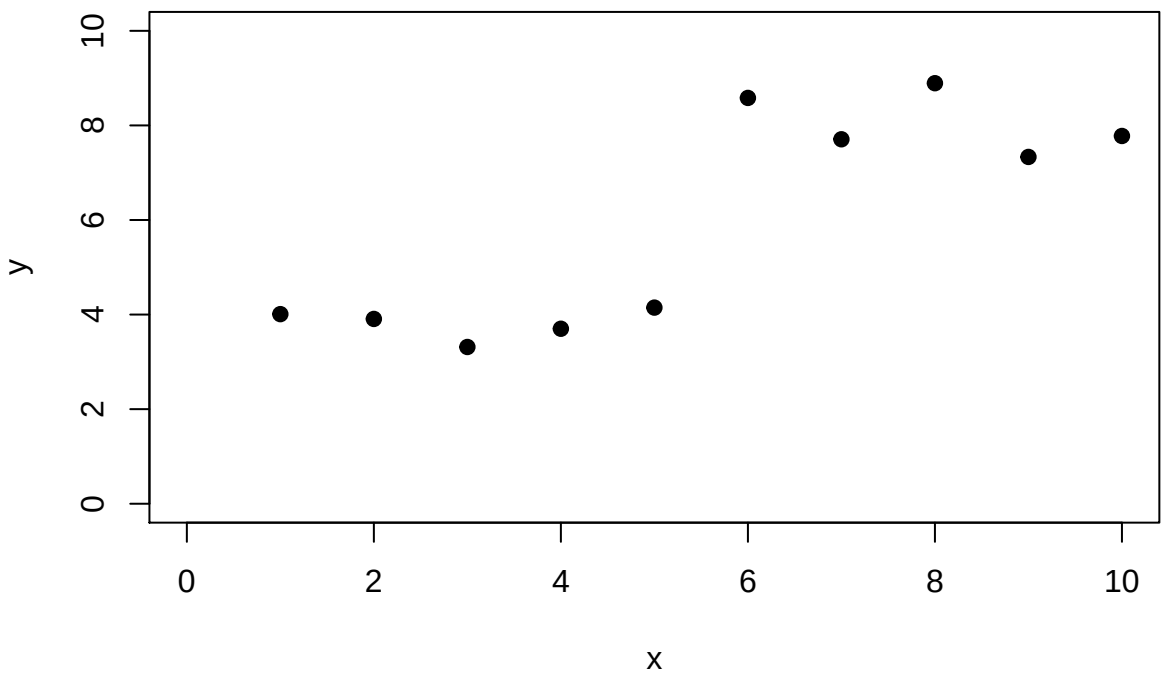
```
##     x        y cluster
## 1   1 4.009373       1
## 2   2 3.907874       1
## 3   3 3.314335       3
## 4   4 3.700416       3
## 5   5 4.147273       4
## 6   6 8.581343       2
## 7   7 7.707038       2
## 8   8 8.892733       2
## 9   9 7.333703       2
## 10 10 7.776717       2
```

```
kmeansObj$centers
```

```
##     x        y
## 1 1.5 3.958623
## 2 8.0 8.058307
## 3 3.5 3.507375
## 4 5.0 4.147273
```


y
x
0
2
4
6
8
10
12
1
2
3
4
5
6
7
8
9
10

Hierarchical Clustering

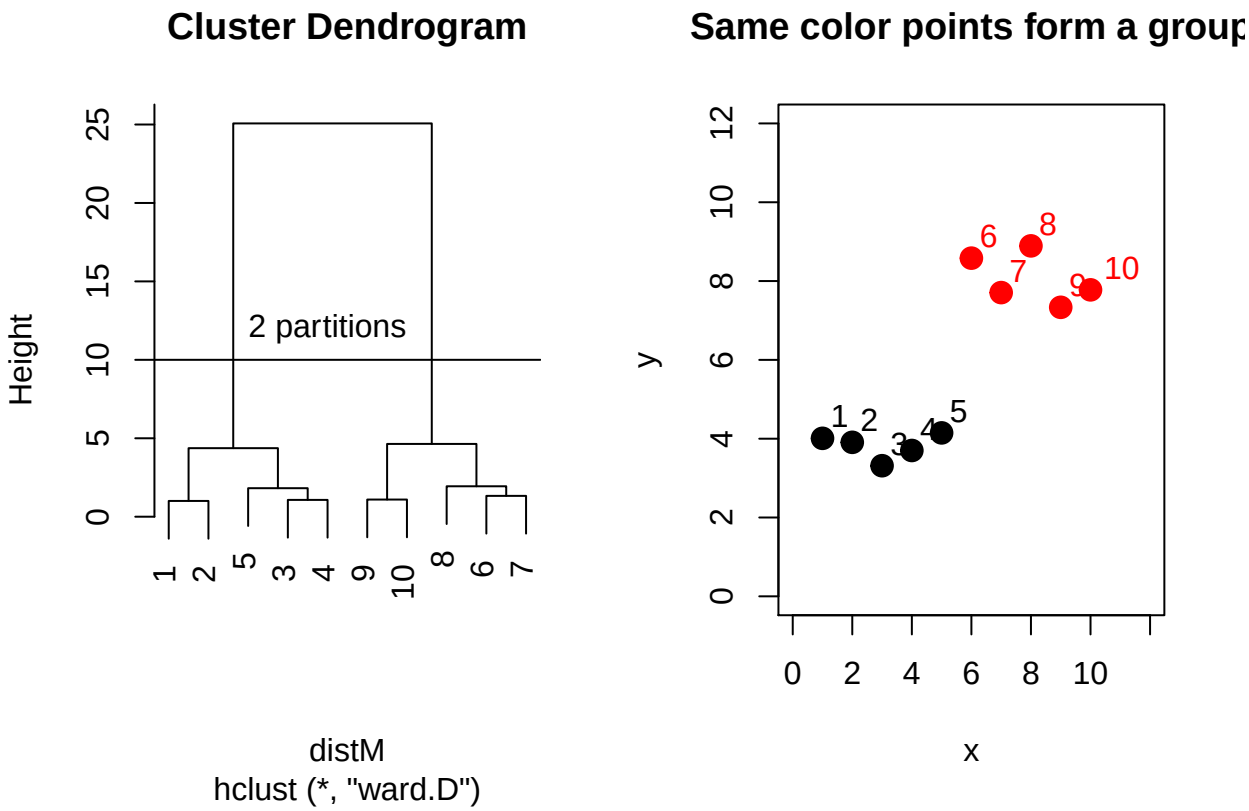



```
distM <- dist(obs)
clusters <- hclust(distM, method = "ward.D")
```

Create 2 partitions

```
mem <- cutree(clusters, k =2)
data.frame(obs, cluster = mem)
```

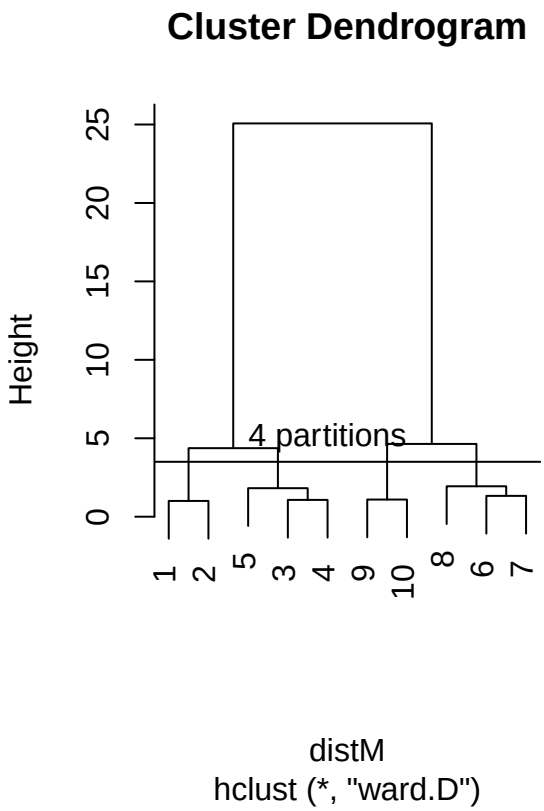
```
##     x        y cluster
## 1   1 4.009373       1
## 2   2 3.907874       1
## 3   3 3.314335       1
## 4   4 3.700416       1
## 5   5 4.147273       1
## 6   6 8.581343       2
## 7   7 7.707038       2
## 8   8 8.892733       2
## 9   9 7.333703       2
## 10 10 7.776717       2
```

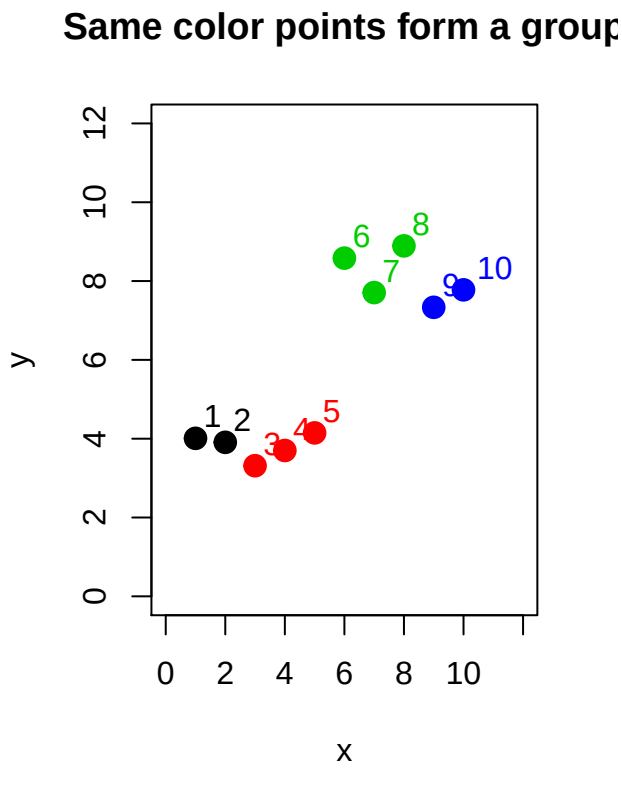


Create 4 partitions

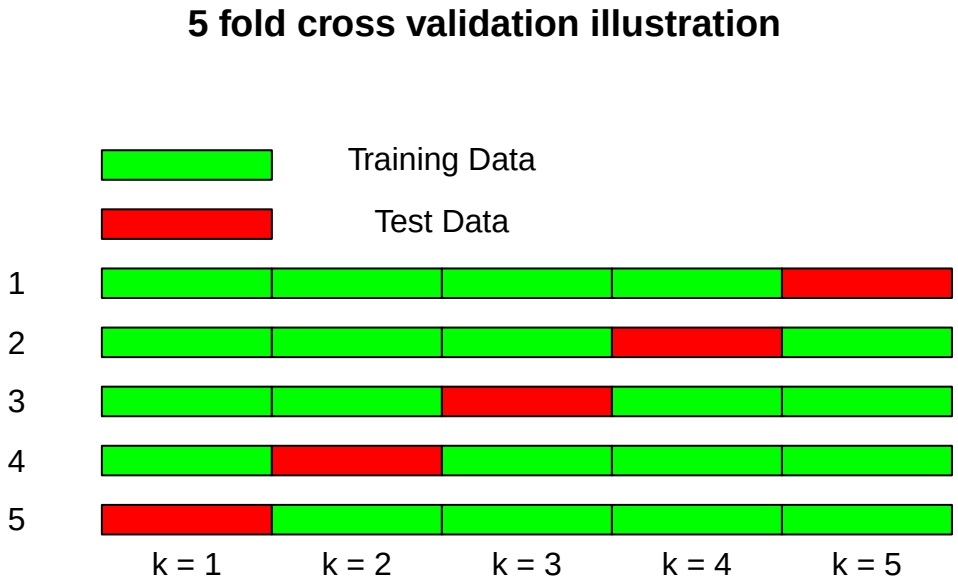
```
mem <- cutree(clusters, k =4)
data.frame(obs, cluster = mem)
```

```
##     x        y cluster
## 1   1 4.009373       1
## 2   2 3.907874       1
## 3   3 3.314335       2
## 4   4 3.700416       2
## 5   5 4.147273       2
## 6   6 8.581343       3
## 7   7 7.707038       3
## 8   8 8.892733       3
## 9   9 7.333703       4
## 10 10 7.776717       4
```





Cross-validation



```
#rfModel <- randomForest(Species ~ . , data = trainData, ntree = 3)
```

knn

```
head(trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

```
set.seed(100)
index <- sample(nrow(iris), 0.6*nrow(iris))
p <- knn(iris[index, 1:4], iris[-index, 1:4], iris[index, 5], 1)
data.frame(iris[-index, 5], p)
```

```
##    iris..index..5.          p
## 1           setosa     setosa
## 2           setosa     setosa
## 3           setosa     setosa
## 4           setosa     setosa
## 5           setosa     setosa
## 6           setosa     setosa
## 7           setosa     setosa
## 8           setosa     setosa
## 9           setosa     setosa
## 10          setosa     setosa
## 11          setosa     setosa
## 12          setosa     setosa
## 13          setosa     setosa
## 14          setosa     setosa
## 15          setosa     setosa
## 16          setosa     setosa
## 17          setosa     setosa
## 18          setosa     setosa
## 19          setosa     setosa
## 20          setosa     setosa
## 21          setosa     setosa
## 22          setosa     setosa
## 23          setosa     setosa
## 24          setosa     setosa
## 25      versicolor versicolor
## 26      versicolor versicolor
## 27      versicolor versicolor
## 28      versicolor versicolor
## 29      versicolor versicolor
## 30      versicolor versicolor
```

```
## 31      versicolor  virginica
## 32      versicolor versicolor
## 33      versicolor versicolor
## 34      versicolor versicolor
## 35      versicolor versicolor
## 36      versicolor versicolor
## 37      versicolor versicolor
## 38      versicolor versicolor
## 39      versicolor versicolor
## 40      versicolor versicolor
## 41      versicolor versicolor
## 42       virginica  virginica
## 43       virginica  virginica
## 44       virginica versicolor
## 45       virginica  virginica
## 46       virginica  virginica
## 47       virginica  virginica
## 48       virginica  virginica
## 49       virginica  virginica
## 50       virginica  virginica
## 51       virginica  virginica
## 52       virginica  virginica
## 53       virginica  virginica
## 54       virginica versicolor
## 55       virginica  virginica
## 56       virginica versicolor
## 57       virginica  virginica
## 58       virginica  virginica
## 59       virginica  virginica
## 60       virginica  virginica
```

```
table(iris[-index, 5], p)
```

```
##             p
##              setosa versicolor virginica
##   setosa         24          0         0
##   versicolor      0         16         1
##   virginica       0          3        16
```

```
## show cross validation
## show parameter selection
## show visualization
```

knn2

```
head(trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
```

```
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

```
set.seed(100)
index <- sample(nrow(trees), 0.6*nrow(trees))
p <- knn(trees[index, 1:2], trees[-index, 1:2], iris[index, 3], 4)
data.frame(trees[-index, 3], p)
```

```
##    trees..index..3.   p
## 1              10.3 1.1
## 2              10.2 1.7
## 3              18.8 1.5
## 4              15.6 1.4
## 5              21.0 1.5
## 6              19.1 1.5
## 7              22.2 1.5
## 8              24.9 1.5
## 9              38.3   1
## 10             42.6 1.5
## 11             58.3 1.6
## 12             51.5 1.6
## 13             77.0 1.6
```

baye's theorem

```
head(Titanic)
```

```
## [1]  0  0 35  0  0  0
```

svm

```
#svmModel <- svm()
```