# Titanic - Decision Trees

Code

Hide

```
library(rpart)
library(rattle)
library(dplyr)
library(RCurl)
library(ggplot2)
```

## 1. Reading data

Hide

```
url <- getURL('https://raw.githubusercontent.com/frankwwu/R-Knots/master/Titanic/train.csv')
train <- read.csv(text = url)
url <- getURL('https://raw.githubusercontent.com/frankwwu/R-Knots/master/Titanic/test.csv')
test <- read.csv(text = url)
```

## 2. Displaying data

Hide

```
str(train)
```

```
'data.frame':   891 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 581 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
 $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

Hide

```
str(test)
```

```
'data.frame':   418 obs. of  11 variables:
 $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
 $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
 $ Name       : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 409 273 414 182 370 85 58 5 104 ..
.
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
 $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
 $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
 $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
 $ Ticket     : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 262 159 85 101 270 ...
 $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
 $ Cabin      : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

## 3. Removing NAs

Hide

```
train <- train %>% na.omit()
test <- test %>% na.omit()
```

## 4. Converting categorical variables to factors

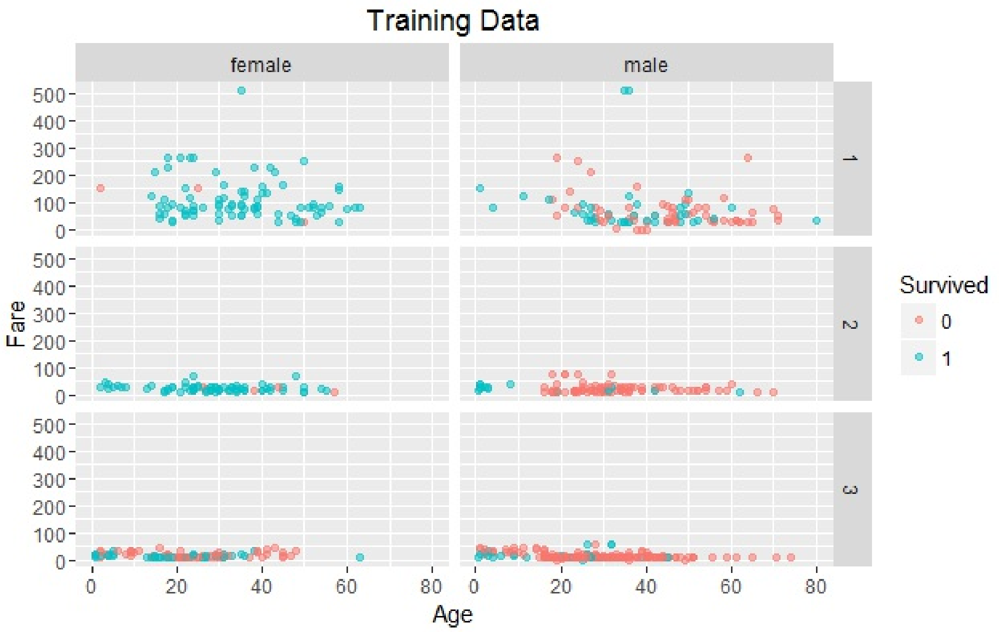
Hide

```
train$Survived <- factor(train$Survived)
train$Pclass <- factor(train$Pclass)
test$Pclass <- factor(test$Pclass)
```

## 5. Visualizing the training data

Hide

```
ggplot(train, aes(Age, Fare, color=Survived)) +
  geom_point(alpha = 0.5) +
  facet_grid(Pclass~Sex) +
  ggtitle("Training Data")
```



## 6.Selecting features

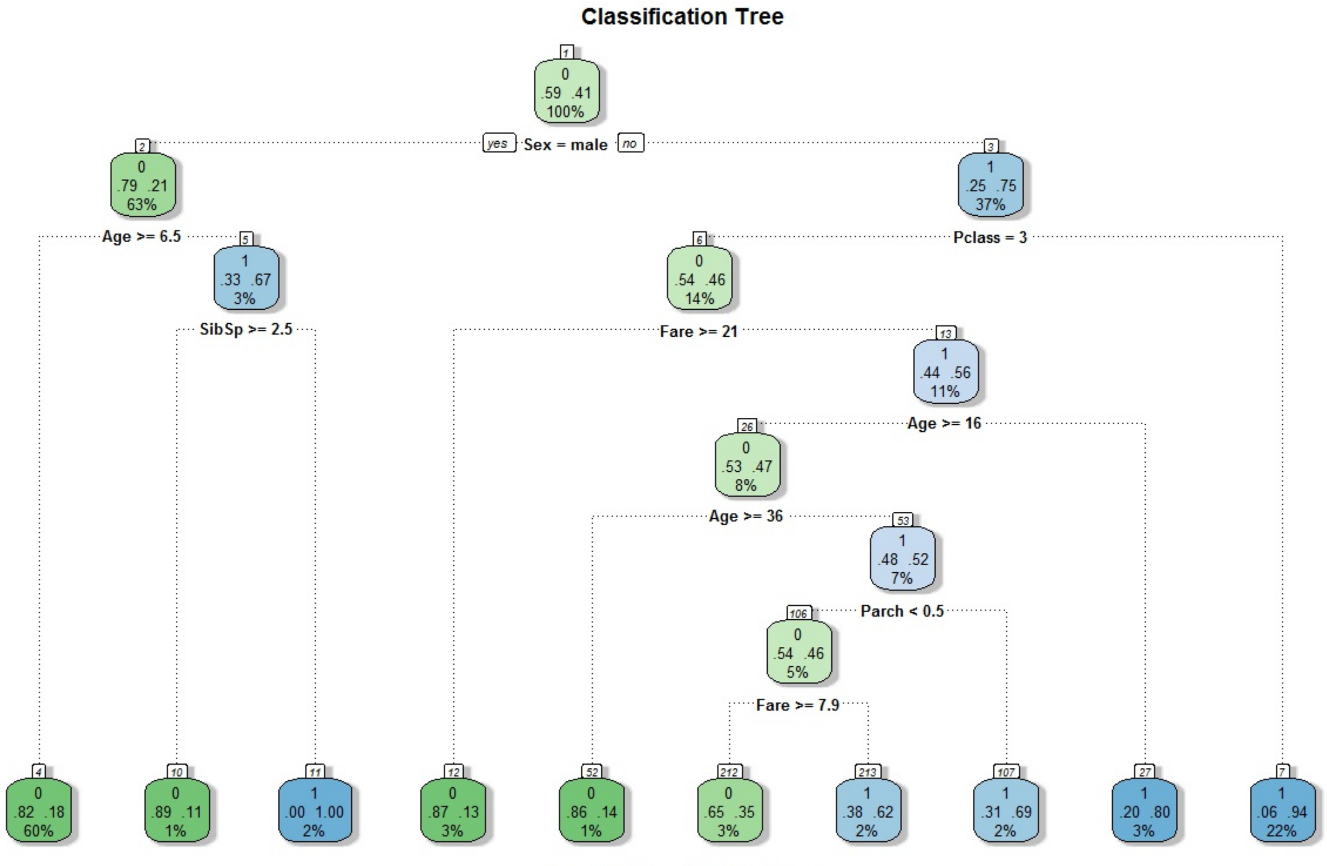
Hide

```
formula = Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked
```

## 7. Creating the Classification Tree

Hide

```
set.seed(9)
tree <- rpart(formula, data=train, method="class")
fancyRpartPlot(tree, uniform=TRUE, main="Classification Tree")
```



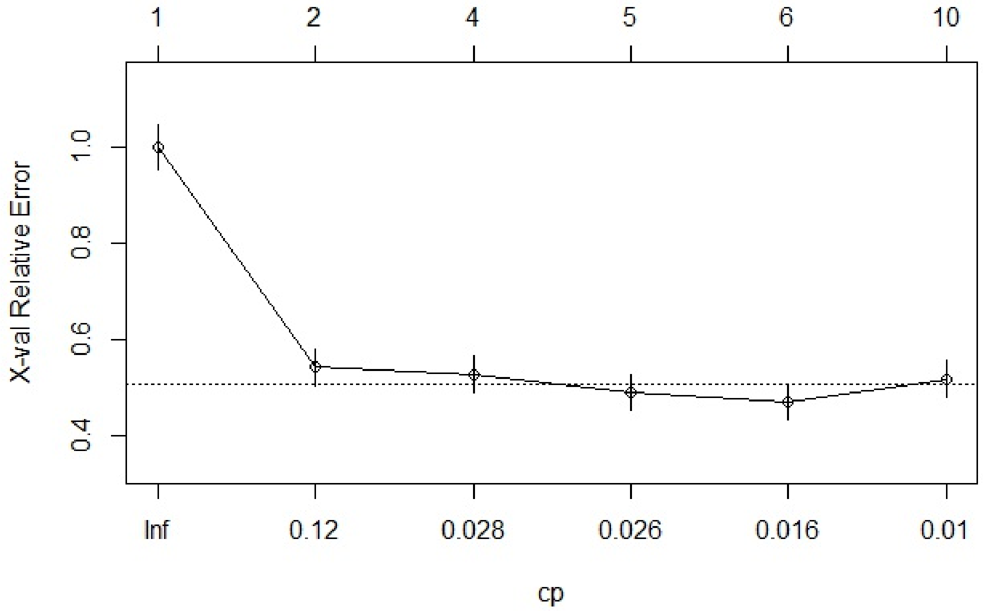
Rattle 2016-Nov-16 20:34:49 frank

## 8. Cross-Validation

To examine whether the tree model is over fitting, find the size of tree with the minimum error.

Hide

```
plotcp(tree)
```



Hide

```
tree$cptable[which.min(tree$cptable[,"xerror"]),"CP"]
```

```
[1] 0.01034483
```

Hide

```
printcp(tree)
```

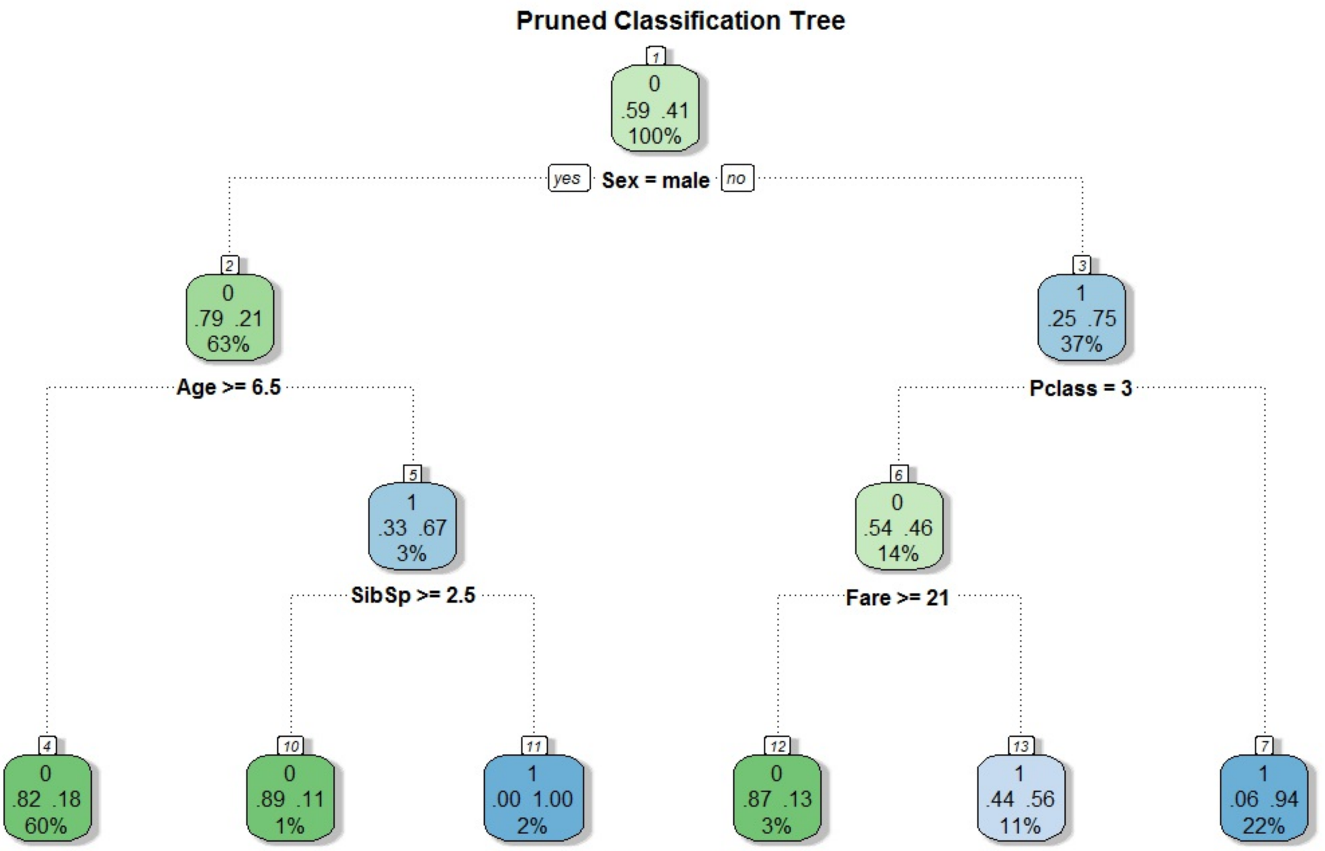
```


Classification tree:
rpart(formula = formula, data = train, method = "class")

Variables actually used in tree construction:
[1] Age    Fare   Parch  Pclass Sex    SibSp 

Root node error: 290/714 = 0.40616

n= 714 

        CP nsplit rel error  xerror     xstd
1 0.458621      0   1.00000 1.00000 0.045252
2 0.029310      1   0.54138 0.54138 0.038162
3 0.027586      3   0.48276 0.52759 0.037808
4 0.024138      4   0.45517 0.48966 0.036779
5 0.010345      5   0.43103 0.46897 0.036181
6 0.010000      9   0.38966 0.51724 0.037535
```

# 9. Pruning the Tree

Prune the over fitting notes.

Hide

```
trim <- tree$cptable[which.min(tree$cptable[,"xerror"]),"CP"]
ptree<- prune(tree, cp=trim)
fancyRpartPlot(ptree, uniform=TRUE, main="Pruned Classification Tree")
```



Rattle 2016-Nov-16 20:34:50 frank

## 10. Predicting with the test data

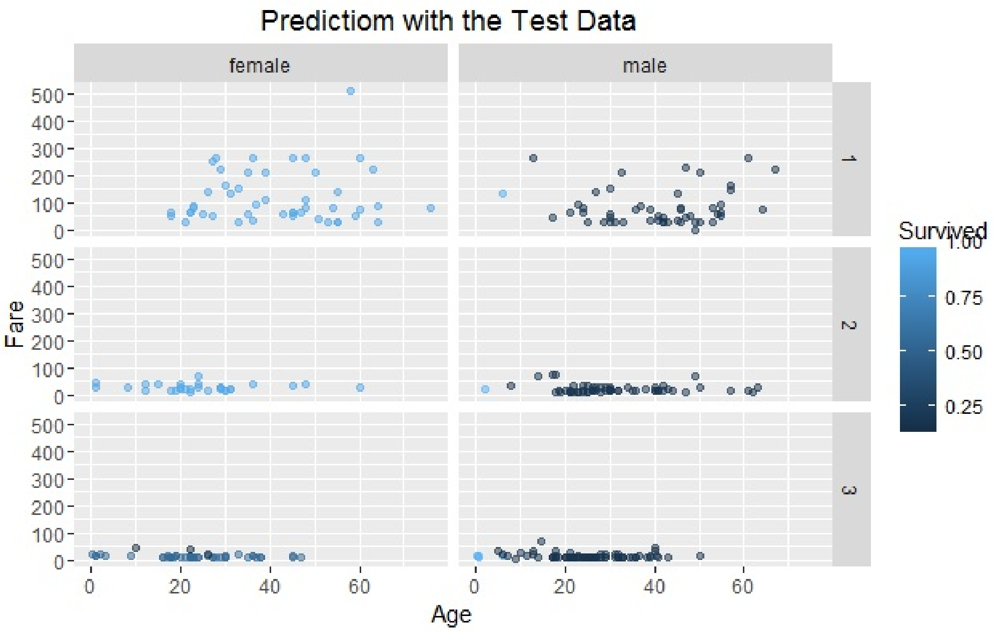
Hide

```
predict <- predict(ptree, test, type = "prob")
```

## 11. Visualizing the result

Hide

```
test$Survived <- predict[,2]
ggplot(test, aes(Age, Fare, color=Survived)) +
  geom_point(alpha = 0.5) +
  facet_grid(Pclass~Sex) +
  ggtitle("Predictiom with the Test Data ")
```


Predictiom with the Test Data
female
male
Fare
Age
Survived