Mining Titanic Dataset

Colin Manuel Fernandes - 914777

Contents

1 1.Introduction

The data is taken from one of **Kaggle prediction competitions**. The Competition Description can be found **here**.

Main features of my EDA
- Null value imputation using MICE
- Feature engineering
- Predition using Random Forest ML algorithm

2 2.Exploratory Analysis of Data

2.1 Data Overview

The data has been split into two groups:

- Training set (train.csv)
- Test set (test.csv)
- Results for the test('gender_submission.csv')

Dataset has 12 variables

2.2 Data Load

```
library('ggplot2') # visualization
library('ggthemes') # visualization
library('scales') # visualization
library('dplyr') # data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library('stringr') #string operations

setwd("D:/Project/Information Management/Titanic")

#load all the datafiles in different variables
train <- read.csv("train.csv", header = TRUE)
test <- read.csv("test.csv", header = TRUE)

#result data
gender_submission <- read.csv('gender_submission.csv', header = TRUE)


#join the submission file to get a bigger dataframe
titanic <- rbind(train,merge(test,gender_submission,by='PassengerId'))

#go through all the variables of the file
str(titanic)
```

```
## 'data.frame':    1309 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/ 1307 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
##  $ 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/ 929 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/ 187 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 ...
```

2.3 Data reformating

We see most of the labels are miss interpreted by 'R' eg. 'Name' which is showing factor of 1307 levels, which otherwise should be a character. Also in the 'Name' variable we can see that there are only 1307 unique entries instead of 1309.

```r
#reformat the datatypes using basic domain knowledge
titanic$Survived <- as.factor(titanic$Survived)
titanic$Pclass <- as.factor(titanic$Pclass)
titanic$Name <- as.character(titanic$Name)
titanic$Cabin <- as.character(titanic$Cabin)
titanic$Ticket <- as.character(titanic$Ticket)

#go through all the variables of the file
str(titanic)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ 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     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

2.4 Investigation

```r
#no of unique passengers
length(unique(titanic$Name))
```

```
## [1] 1307
```

```r
# Two duplicate names, take a closer look
# First, get the duplicate names and store them as a vector

dup.names <- titanic[which(duplicated(titanic$Name)), "Name"]


# Next, take a look at the records in the combined data set
titanic[which(titanic$Name %in% dup.names),]
```

```
##     PassengerId Survived Pclass                 Name    Sex  Age SibSp
## 290         290        1      3 Connolly, Miss. Kate female 22.0     0
## 697         697        0      3     Kelly, Mr. James   male 44.0     0
## 892         892        0      3     Kelly, Mr. James   male 34.5     0
## 898         898        1      3 Connolly, Miss. Kate female 30.0     0
##     Parch Ticket   Fare Cabin Embarked
## 290     0 370373 7.7500              Q
## 697     0 363592 8.0500              S
## 892     0 330911 7.8292              Q
## 898     0 330972 7.6292              Q
```

2.5 Feature Engineering

```
# Extract title from passenger names
titanic$Title <- gsub('(.*, )|(\\..*)', '', titanic$Name)

# Show title counts by sex
table(titanic$Sex, titanic$Title)
```

```
##
##           Capt Col Don Dona  Dr Jonkheer Lady Major Master Miss Mlle Mme
##   female     0   0   0    1   1        0    1     0      0  260    2   1
##   male       1   4   1    0   7        1    0     2     61    0    0   0
##
##            Mr Mrs  Ms Rev Sir the Countess
##   female    0 197   2   0   0            1
##   male    757   0   0   8   1            0
```

```
# Titles with very low cell counts to be combined to "Others" level
Others <- c('Dona', 'Lady', 'the Countess','Capt', 'Col', 'Don',
                'Dr', 'Major', 'Rev', 'Sir', 'Jonkheer')

# Also re-assign mlle, ms, and mme accordingly
titanic$Title[titanic$Title == 'Mlle']        <- 'Miss'
titanic$Title[titanic$Title == 'Ms']          <- 'Miss'
titanic$Title[titanic$Title == 'Mme']         <- 'Mrs'
titanic$Title[titanic$Title %in% Others]  <- 'Others'


# Family Size feature
titanic$FamilySize <- as.factor(titanic$SibSp + titanic$Parch + 1)
```

3 3.Data Preprocessing

3.1 Missing value Imputation

```
library('mice') # imputation
```

```
## Loading required package: lattice
```

```
# We Check the levels and top values when 'str(titanic)' is run and see that cabin and embarked has mis

#Also we check columns having na values
names(which(sapply(titanic, anyNA)))
```

```
## [1] "Age"  "Fare"
```

```
#to add Embarkment class we check the class
titanic[titanic$Embarked=='','PassengerId']
```

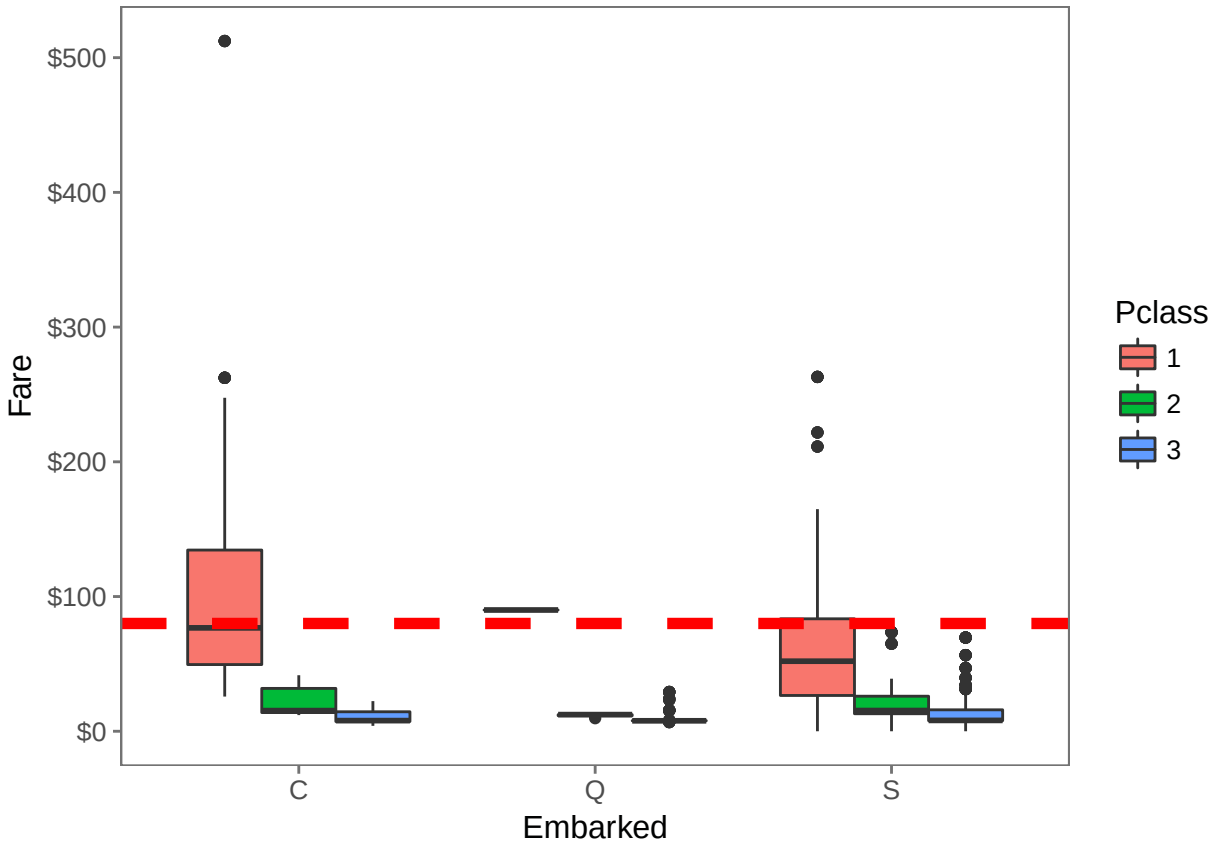
```
## [1]  62 830
```

```
#to impute missing values we check corresponding likely values in class and fare
titanic[c(62, 830), c('Embarked','Fare','Pclass')]
```

```
##     Embarked Fare Pclass
## 62         80      1
## 830        80      1
```

```
#check embarkment againdt fare
embark_fare <- subset(titanic, PassengerId!=62 & PassengerId!=830)

# Use ggplot2 to visualize embarkment, passenger class, & median fare
ggplot(embark_fare, aes(x = Embarked, y = Fare, fill = Pclass)) +
  geom_boxplot() +
  geom_hline(aes(yintercept=80),
    colour='red', linetype='dashed', lwd=2) +
  scale_y_continuous(labels=dollar_format()) +
  theme_few()
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```

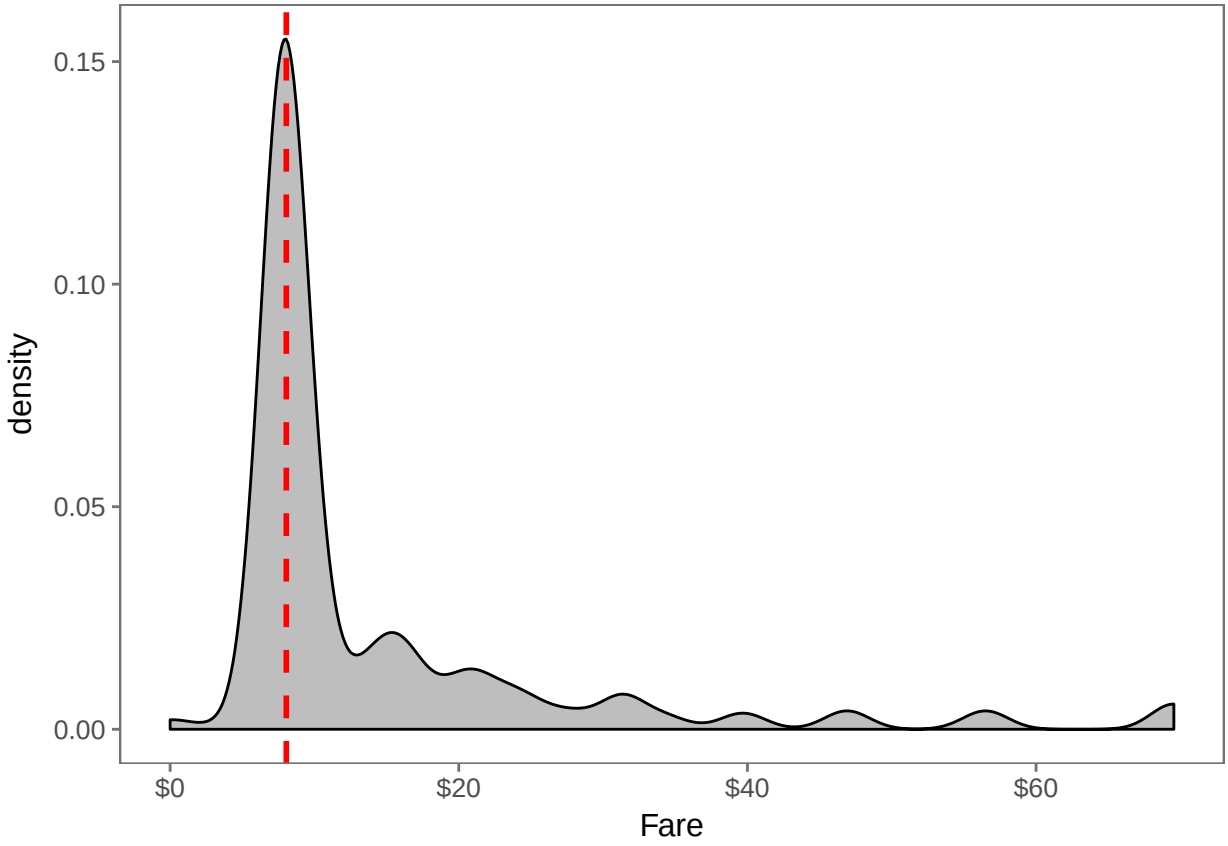



```
# Since their fare was $80 for 1st class, they most likely embarked from 'C'
titanic$Embarked[c(62, 830)] <- 'C'


#Find NA values in Age and Fare
subset(titanic, is.na(Fare))
```

```
##      PassengerId Survived Pclass               Name  Sex  Age SibSp Parch
## 1044        1044        0      3 Storey, Mr. Thomas male 60.5     0     0
##      Ticket Fare Cabin Embarked Title FamilySize
## 1044   3701   NA              S    Mr          1
```

```
#This is a third class passenger who has departed value ('S').Visualize Fares among all others sharing

ggplot(titanic[titanic$Pclass == '3' & titanic$Embarked == 'S', ],
  aes(x = Fare)) +
  geom_density(fill = 'grey') +
  geom_vline(aes(xintercept=median(Fare, na.rm=TRUE)),
    colour='red', linetype='dashed', lwd=1) +
  scale_x_continuous(labels=dollar_format()) +
  theme_few()
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```




```
# Replace missing fare value with median fare for class/embarkment
titanic$Fare[1044] <- median(titanic[titanic$Pclass == '3' & titanic$Embarked == 'S', ]$Fare, na.rm = T

# Show number of missing Age values
sum(is.na(titanic$Age))
```

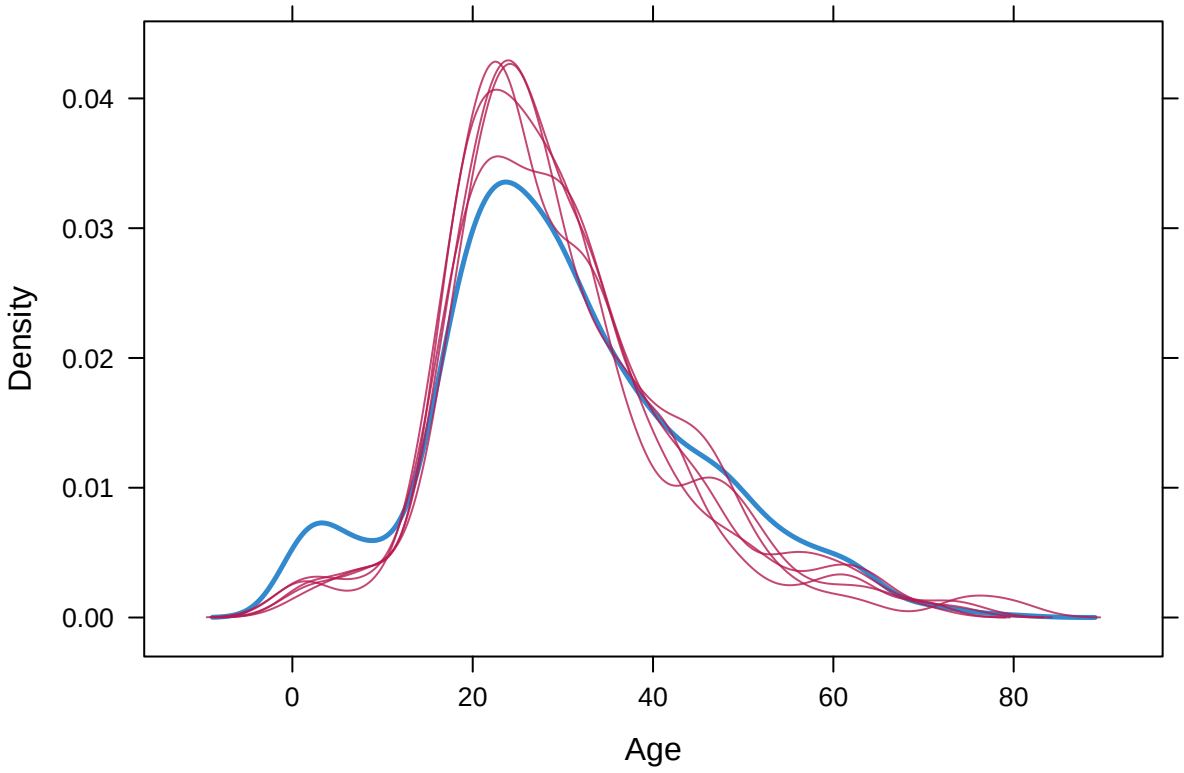
```
## [1] 263
```

```
# Set a random seed
set.seed(129)

# Perform mice imputation, excluding certain less-than-useful variables:
mice_mod <- mice(titanic[, !names(titanic) %in% c('PassengerId','Name','Ticket','Cabin','Survived','Fare
```

```
##
```

```
##  iter imp variable
##    1   1  Age
##    1   2  Age
##    1   3  Age
##    1   4  Age
##    1   5  Age
##    2   1  Age
##    2   2  Age
##    2   3  Age
##    2   4  Age
##    2   5  Age
##    3   1  Age
##    3   2  Age
##    3   3  Age
##    3   4  Age
##    3   5  Age
##    4   1  Age
##    4   2  Age
##    4   3  Age
##    4   4  Age
##    4   5  Age
##    5   1  Age
##    5   2  Age
##    5   3  Age
##    5   4  Age
##    5   5  Age
```

```r
# Save the complete output
mice_output <- complete(mice_mod)

# Plot age distributions
densityplot(mice_mod)
```




```
# Replace Age variable from the mice model

titanic$Age <- mice_output$Age

# Show new number of missing Age values
sum(is.na(titanic $Age))
```

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

4 4.Plots based on different factors

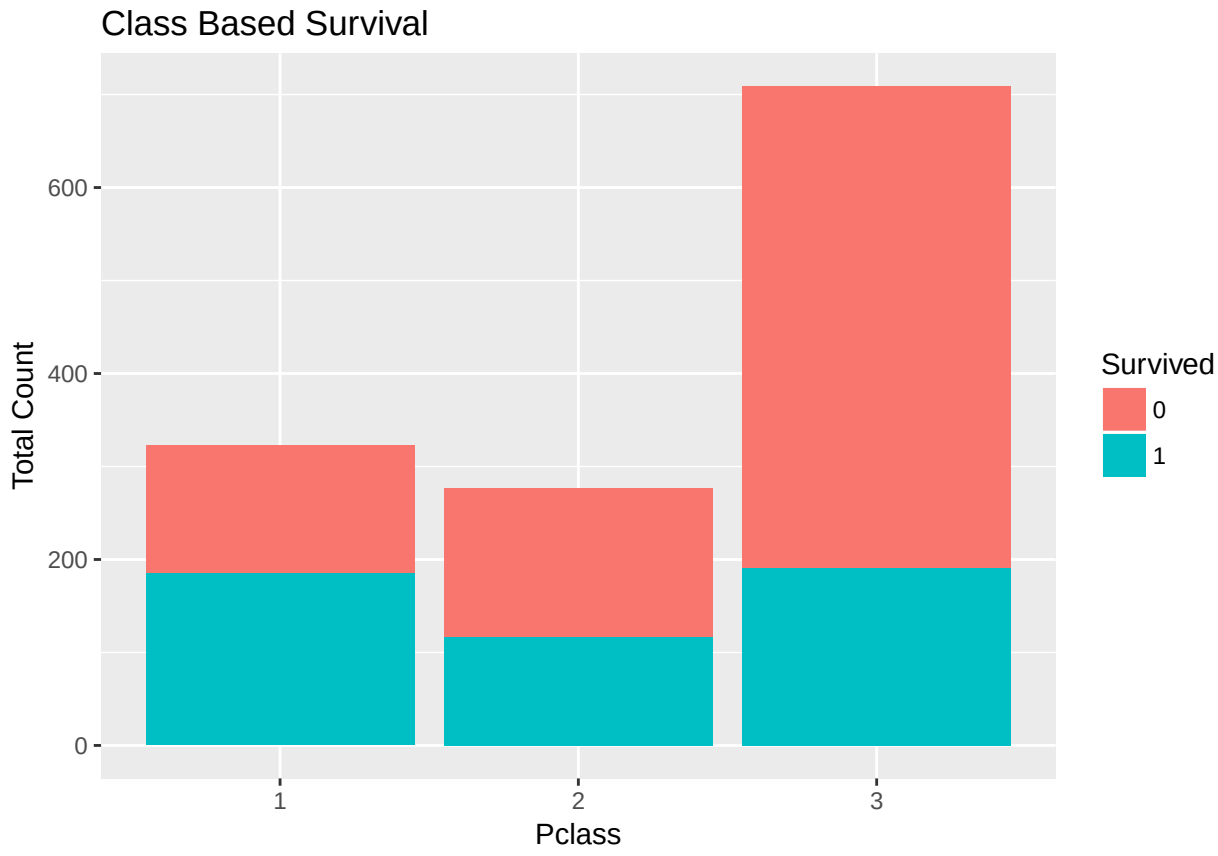
4.1 Survival Based on Class

```
# Distribution of survival in correlation with classes
table(titanic$Pclass,titanic$Survived)
```

```
##
##       0   1
##   1 137 186
##   2 160 117
##   3 518 191
```

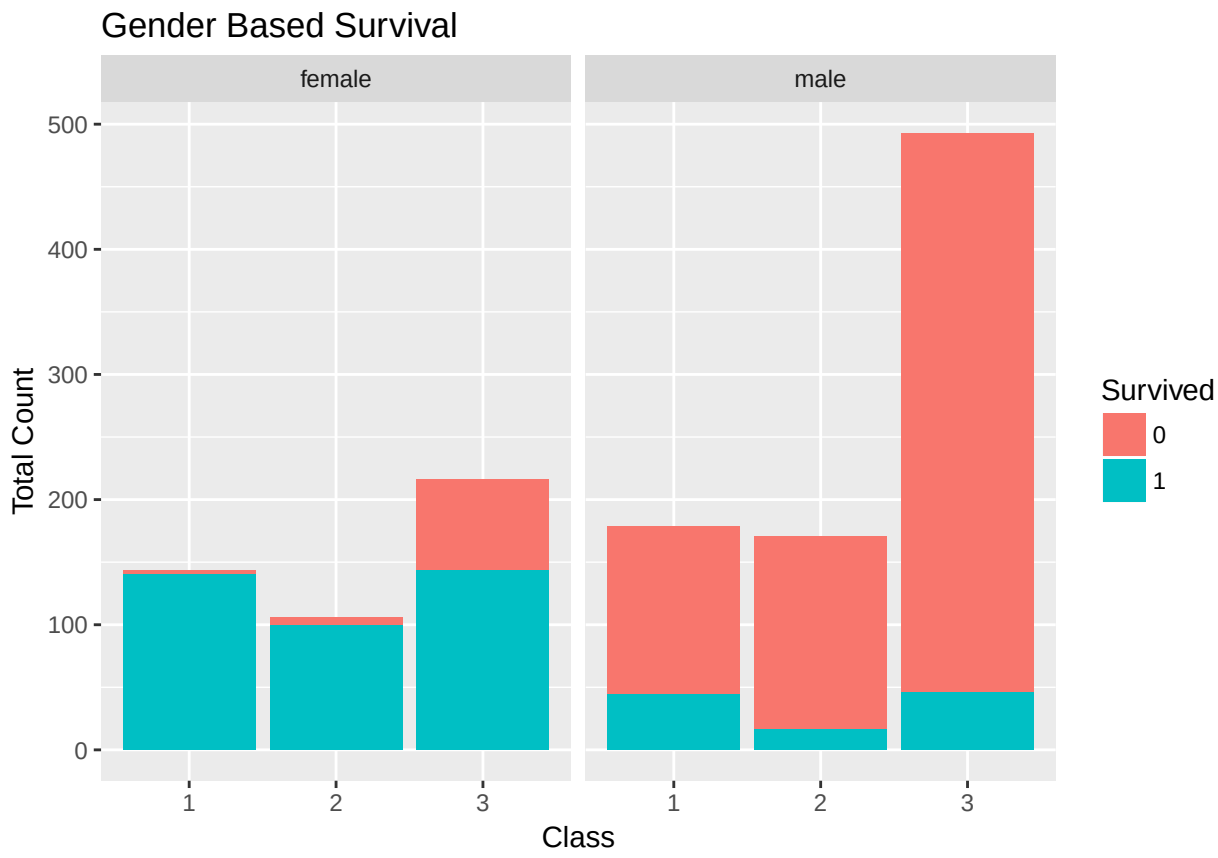
```
# Checking Survival based on class
ggplot(titanic, aes(x = Pclass, fill = Survived)) +
  geom_bar() +
```

```
  ggtitle("Class Based Survival") +
  xlab("Pclass") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



4.2 Survival of people based on Age, Gender, Title, Family size

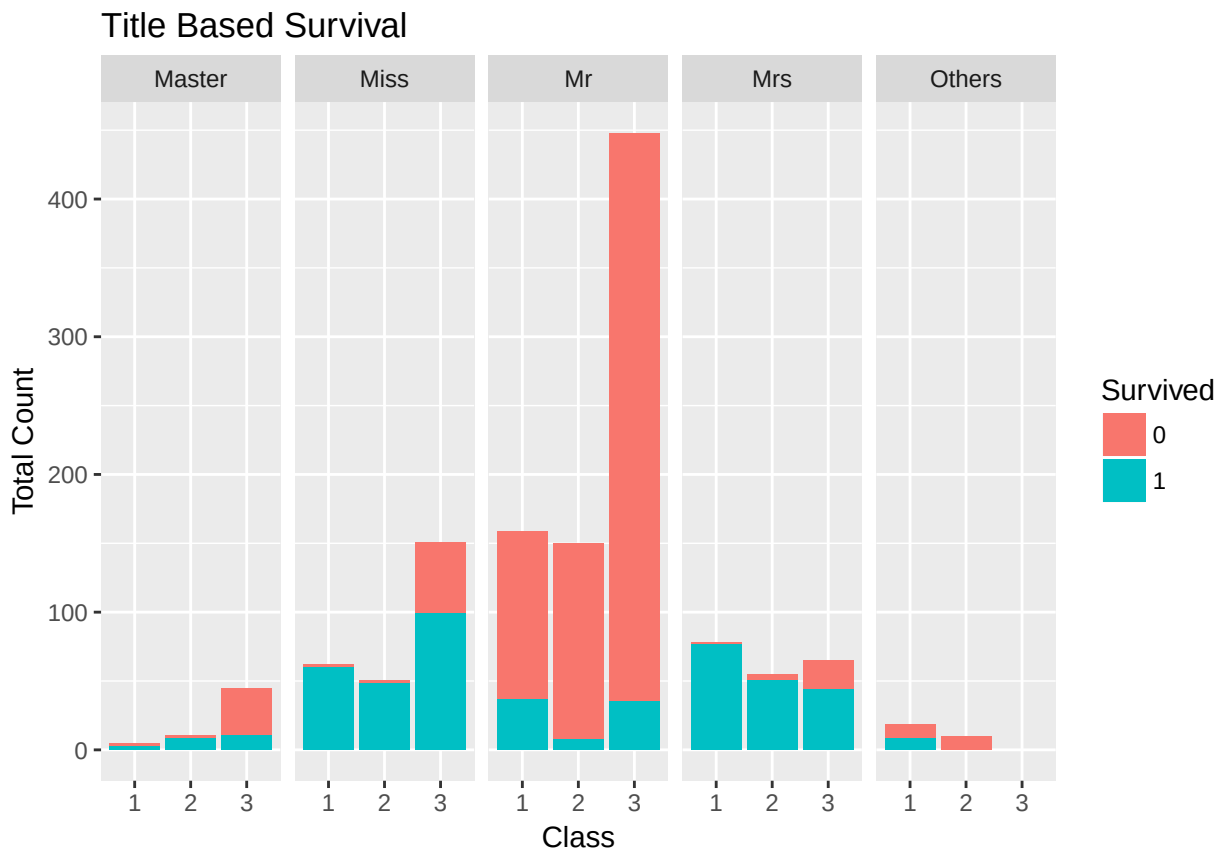
```
# Plot by Gender
ggplot(titanic, aes(x = Pclass, fill = Survived)) +
  geom_bar() +
  facet_wrap(~Sex) +
  ggtitle("Gender Based Survival") +
  xlab("Class") +
  ylab("Total Count") +
  labs(fill = "Survived")
```




```
# Show title counts by sex again
table(titanic$Sex, titanic$Title)
```

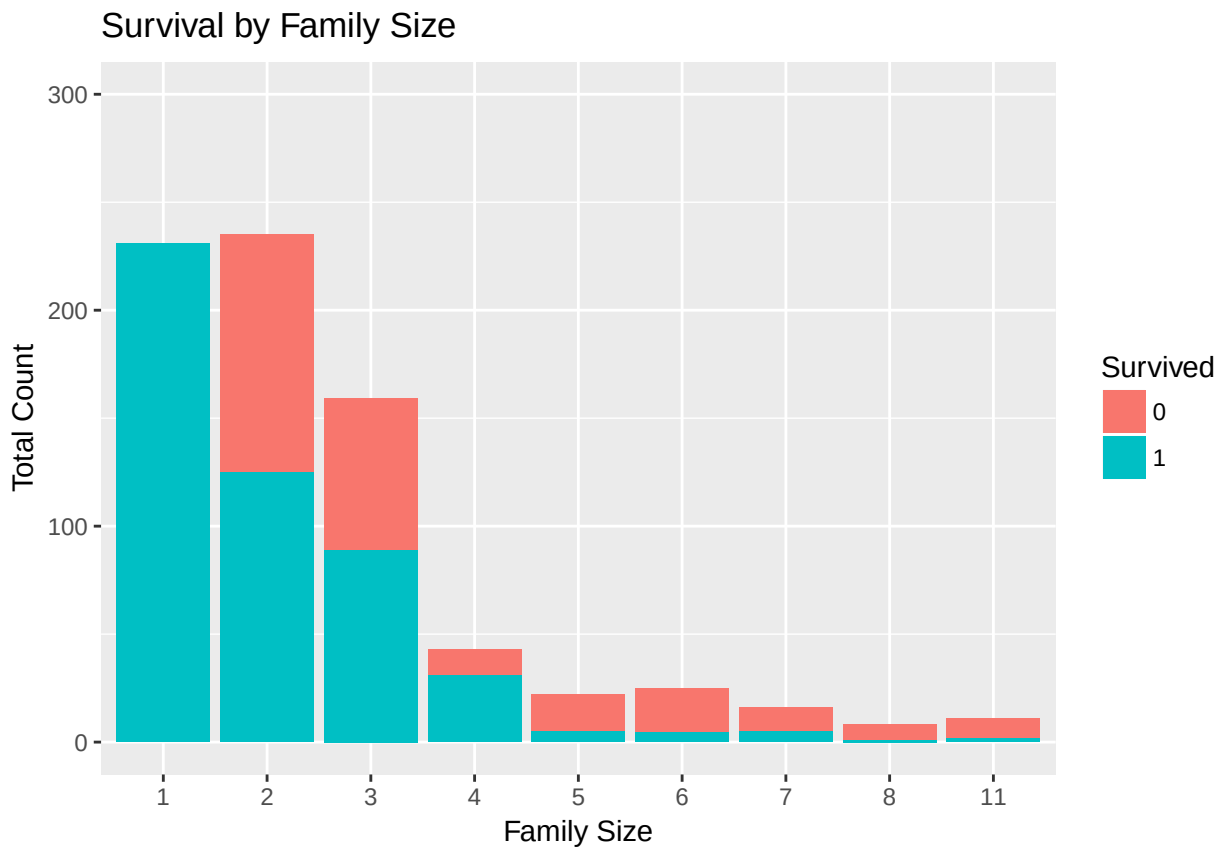
```
##
##          Master Miss  Mr Mrs Others
##   female      0  264   0 198      4
##   male       61    0 757   0     25
```

```
# Plot by Title
ggplot(titanic, aes(x = Pclass, fill = Survived)) +
  geom_bar() +
  facet_wrap(~Title, ncol=5 ) +
  ggtitle("Title Based Survival") +
  xlab("Class") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



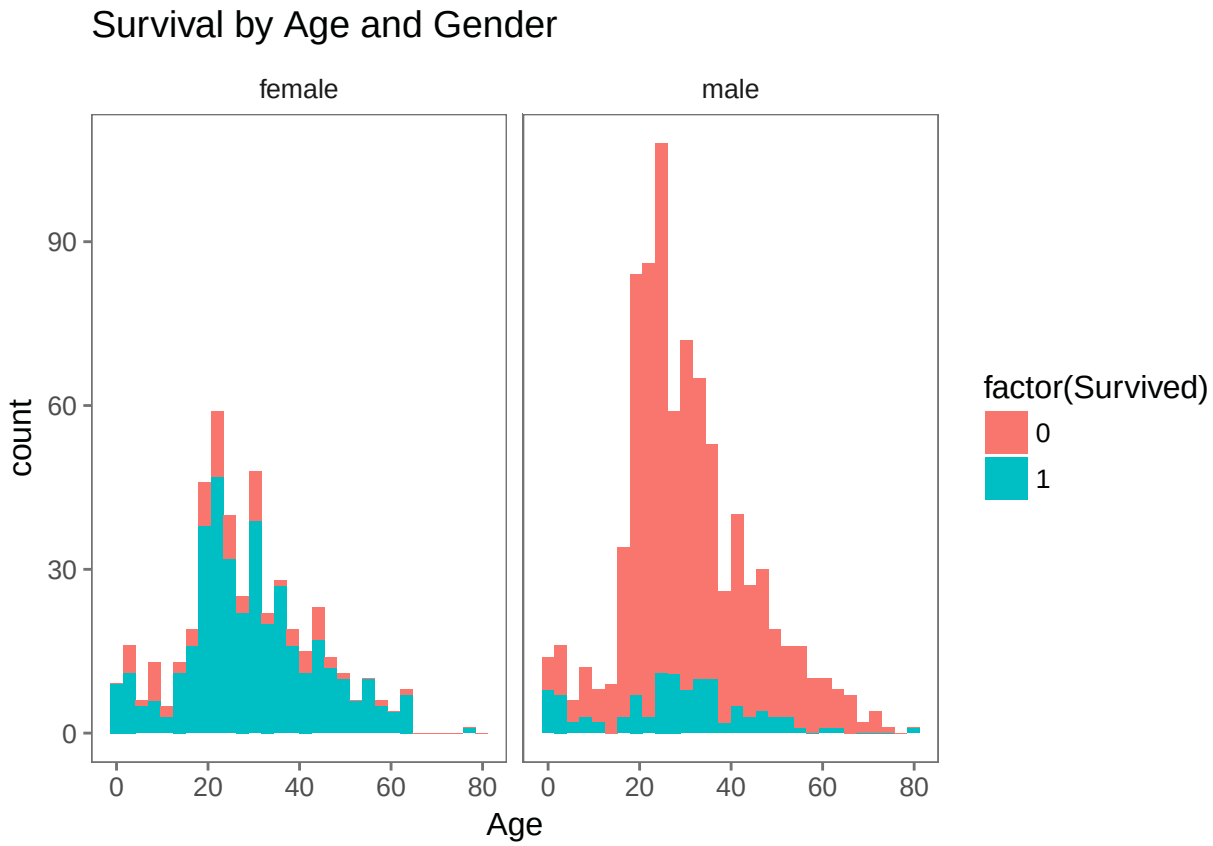

```
# Plot by family size
ggplot(titanic, aes(x = FamilySize, fill = Survived)) +
  geom_bar() +
  ggtitle("Survival by Family Size") +
  xlab("Family Size") +
  ylab("Total Count") +
  ylim(0,300) +
  labs(fill = "Survived")
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```



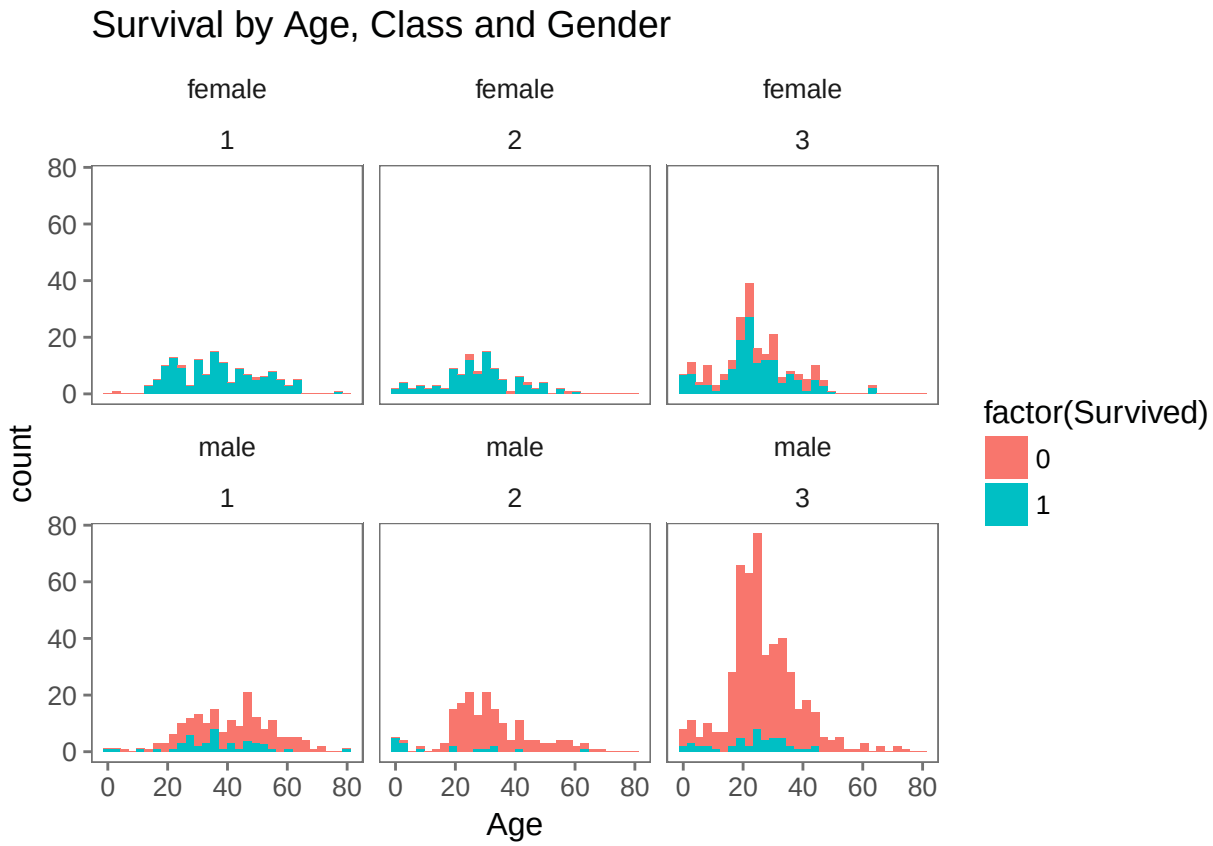

```
#Plot survival with age and sex
ggplot(titanic, aes(Age, fill = factor(Survived))) +
  geom_histogram() +
  ggtitle("Survival by Age and Gender") +
  facet_grid(.~Sex) +
  theme_few()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#Plot survival with age, sex and class
ggplot(titanic, aes(Age, fill = factor(Survived))) +
  geom_histogram()  +
  ggtitle("Survival by Age, Class and Gender") +
  facet_wrap(~Sex + Pclass)  +
  theme_few()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



5 5.Predicting Survival using Random Forest classifier

```
library(randomForest) # classification algorithm
```

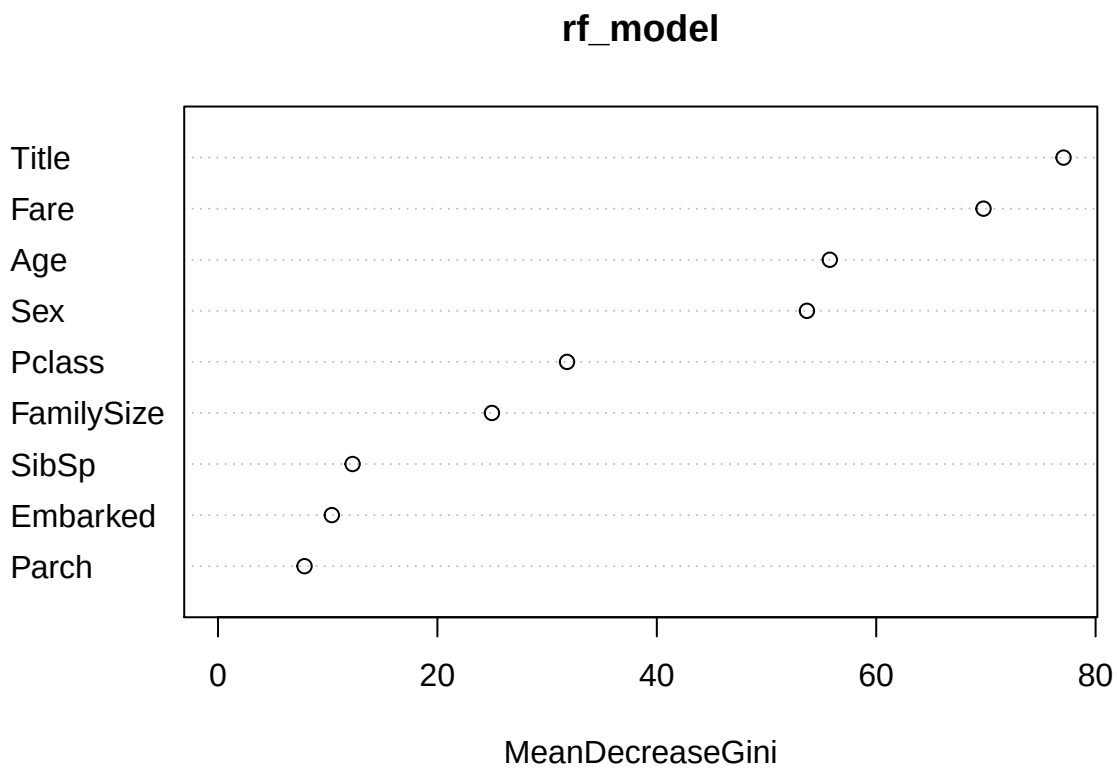
```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:dplyr':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
#Convert all characters to factors
titanic=titanic %>% mutate_if(is.character, as.factor)

#Check is any missing values
md.pattern(titanic)
```

```
##      PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare
## [1,]           1        1      1    1   1   1     1     1      1    1
```

```
## [2,]           0        0      0    0   0   0     0     0      0    0
##       Cabin Embarked Title FamilySize
## [1,]     1        1     1          1 0
## [2,]     0        0     0          0 0
```

```
titanic=titanic %>% mutate_if(is.character, as.factor)

# Split the data back into a train set and a test set
train <- titanic[1:891,]
test <- titanic[892:1309,]


# Set a random seed
set.seed(754)

# Build the model (note: not all possible variables are used)
rf_model <- randomForest(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch +
                                            Fare + Embarked + Title + FamilySize, data = train)

# Show Variable importance
varImpPlot(rf_model)
```




```
# Predict using the test set
prediction <- predict(rf_model, test)
table(prediction)
```

```
## prediction
```

```
##   0   1
## 265 153
```

```
# Save the solution to a dataframe with two columns: PassengerId and Survived (prediction)
solution <- data.frame(PassengerID = test$PassengerId, Survived = prediction)


# Write the solution to file
write.csv(solution, file = 'rf_mod_Solution.csv', row.names = F)
```

6 6.Validation

```
library(caret) #classification and regression training
library(doSNOW) #exploiting parllel execution of the resource
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: snow
```

```
#OOB and Confusion matrix
rf_model
```

```
##
## Call:
##  randomForest(formula = factor(Survived) ~ Pclass + Sex + Age +      SibSp + Parch + Fare + Embarked
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 17.17%
## Confusion matrix:
##     0   1 class.error
## 0 489  60   0.1092896
## 1  93 249   0.2719298
```

```
#10 Fold cross validation
set.seed(2348)
Validation <- createMultiFolds(train$Survived, k = 10, times = 10)



# Set up caret's trainControl object per above.
ctrl.1 <- trainControl(method = "repeatedcv", number = 10, repeats = 10,
                       index = Validation)


# Set up doSNOW package for multi-core training. This is helpful as we're going
cl <- makeCluster(6, type = "SOCK")
registerDoSNOW(cl)

rf.train <- train[, c("Pclass","Sex","Age","SibSp","Parch","Fare","Embarked","Title","FamilySize")]

# Set seed for reproducibility and train
```

```r
set.seed(34324)
rf_val <- train(x = rf.train, y = train$Survived, method = "rf", tuneLength = 3,ntree = 1000, trControl

#Shutdown cluster
stopCluster(cl)

# Check out results
rf_val
```

```
## Random Forest
##
## 891 samples
##   9 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 801, 802, 802, 803, 802, 801, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.8240230  0.6172180
##   5     0.8205549  0.6145851
##   9     0.8047331  0.5835486
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

This page is created in an R Markdown document or RMD. RMD is an cleaner way of sharing code and documentation.
Reference for RMD - **udacity**