Titanic Capstone Report

Contents

1 Introduction **2**

2 Method and Analysis **2**

2.1 Data Generation 2

2.2 Data Cleaning 3

2.2.1 Examine NA data and data dimension 3

2.2.2 Select feature predictors 3

2.2.3 Process age and family size data 4

2.2.4 Fill empty factor data in Embarked, Lifeboat and Cabin. 4

2.2.5 Check cabin and lifeboat categories, Examine full_clean data 5

2.3 Data Preparation 5

2.4 Data Exploration and Visualization 6

2.4.1 Overall Survival rate 6

2.4.2 Survival rate by Sex 6

2.4.3 Survival rate by Pclass 7

2.4.4 Survival rate by Family size 9

2.4.5 Survival rate by Fare 11

2.4.6 Survival rate by Age 14

2.4.7 Survival rate by Embarked 17

2.4.8 Survival rate by Lifeboat 19

2.4.9 Survival rate by Cabin 21

2.5 Modeling Approaches 23

3 Results **24**

3.1 Linear discriminant analysis (LDA) model 24

3.2 Quadratic discriminant analysis (QDA) model 24

3.3 Logistic regression of Generalized linear model 24

3.4 kNN model excluding Lifeboat data 25

3.5 Cross-validated kNN model excluding Lifeboat data 27

3.6 Classification tree model excluding Lifeboat data 28

3.7 Random forest model excluding Lifeboat data 30
3.8 Ensemble of different models 32
3.9 Classification tree model including Lifeboat data 33
3.10 Random forest model including Lifeboat data 35

4 Conclusion **37**

1 Introduction

This project analyzes the Titanic data, uses different models and predictors to train and predict the survival of the Titanic passengers. We use the full Titanic extended dataset (Kaggle + Wikipedia) at https://www.kaggle.com/pavlofesenko/titanic-extended?select=full.csv which has more completed data than the one in the Titanic package.

The Titanic extended dataset include extra age data in Age_wiki. We will remove the Age field and use Age_wiki as Age. Other extra data will be used for our training are Embarked, Cabin and Lifeboat. We will remove other unused extra fields such as Name, Hometown, WikiId and Destination. The full data set will be split into training and test data sets.

Using the skills and knowledge learning from the EDX PH125.8x: Data Science: Machine Learning course and other parts from the Data science course we will experiment different data modeling methods on the Titanic data training and prediction for the survival of the Titanic passenger.

The processes including data preparation, cleaning, exploration to modeling, training and conclusion are listed below, which are also aligned with the document structure:

Data Generation Import the required libraries and the Titanic full.csv data downloaded from Kaggle.

Data Cleaning Clean unused data columns to free memory. Prepare data for Age, Embarked, Lifeboat, FamilySize and Cabin. Fill empty factor data in Embarked, Lifeboat and Cabin. Remove temporary data.

Data Preparation Split full_clean data into training and test data set by 80/20,

Data Exploration and Visualization Inspect data dimension, content, show statics and visualization of Survival rates by Age, Sex, Fare, FamilySize, Embarked, Cabin, Lifeboat etc distributions to gain more insight and understanding of the data and the modeling methods can be used.

Modeling Approaches Go through the selected modeling methods, from Linear discriminant analysis (LDA)Quadratic discriminant analysis (QDA), Logistic regression of Generalized linear model, to k-nearest neighbors (kNN), Classification tree and Random forest modelings, and the special factor Lifeboat.

Results User R code to implement the modeling approaches. Training set is used for model training and test set for Survival prediction and accuracy rate comparison. Show the target, modeling results and comparisons.

Conclusion Conclude the modeling results, limitation and future work.

2 Method and Analysis

2.1 Data Generation

Import the Titanic full.csv data from Kaggle.

```r
#install required libraries if they have not been installed
if(!require(caret))
  install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(tidyverse))
  install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(rpart))
  install.packages("randomForest", repos = "http://cran.us.r-project.org")
#load required libraries
library(caret, quietly = TRUE)
library(tidyverse, quietly = TRUE)
library(rpart, quietly = TRUE)

#import data from CSV
full <- read.csv("full.csv", stringsAsFactors = TRUE)
```

2.2 Data Cleaning

Clean unused data columns to free memory. Prepare data for Age, Embarked, Lifeboat, FamilySize and Cabin. Fill empty factor data in Embarked, Lifeboat and Cabin. Remove temporary data.

2.2.1 Examine NA data and data dimension

```r
# extract only data contained survived for training and test
full_clean <- full[!is.na(full$Survived),]
#Examine NA data and data dimension
colSums(is.na(full_clean))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
##      WikiId   Name_wiki    Age_wiki    Hometown     Boarded Destination
##           2           0           4           0           0           0
##    Lifeboat        Body       Class
##           0           0           2
```

```r
dim(full_clean)
```

```
## [1] 891  21
```

2.2.2 Select feature predictors

```r
#  select feature predictors
full_clean <- full_clean %>%
  select(Survived,  Sex, Pclass, Age_wiki, Fare, SibSp, Parch, Embarked, Lifeboat, Cabin)
```

2.2.3 Process age and family size data

```
# process age and family size
full_clean <- full_clean %>%
  mutate(Survived = factor(Survived),
         # NA age to median age
         Age = ifelse(is.na(Age_wiki), median(Age_wiki, na.rm = TRUE), Age_wiki),
         # count family members
         FamilySize = SibSp + Parch + 1) %>%
  select(Survived, Sex, Pclass, Age, Fare, SibSp, Parch,
         FamilySize, Embarked, Lifeboat, Cabin)
```

2.2.4 Fill empty factor data in Embarked, Lifeboat and Cabin.

```
# fill empty Embarked with X
levels(full_clean$Embarked)[match("",levels(full_clean$Embarked))] <- "X"

#Examine NA data again
colSums(is.na(full_clean))
```

```
##   Survived        Sex     Pclass        Age       Fare      SibSp      Parch
##          0          0          0          0          0          0          0
## FamilySize   Embarked   Lifeboat      Cabin
##          0          0          0          0
```

```
#Examine full_clean data
str(full_clean)
```

```
## 'data.frame':    891 obs. of  11 variables:
##  $ Survived  : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Sex       : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Pclass    : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Age       : num  22 35 26 35 35 22 54 2 26 14 ...
##  $ Fare      : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ SibSp     : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch     : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ FamilySize: num  2 2 1 2 1 1 1 5 3 2 ...
##  $ Embarked  : Factor w/ 4 levels "X","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ Lifeboat  : Factor w/ 25 levels "","?","1","10",..: 1 15 9 25 1 1 1 1 10 2 ...
##  $ Cabin     : Factor w/ 187 levels "","A10","A11",..: 1 108 1 72 1 1 165 1 1 1 ...
```

```
# Keep only cabin character
full_clean$CabinN <- vector("character", nrow(full_clean))
for (i in 1:nrow(full_clean)) {
    pattern <- "[0-9]*|\\s"
    full_clean$CabinN[i] <- substr(gsub(pattern, "", full_clean$Cabin[i]),1,1)
}
full_clean$Cabin <- factor(full_clean$CabinN)
full_clean$CabinN <- NULL
```

```
# fill empty Cabin with X
levels(full_clean$Cabin)[match("",levels(full_clean$Cabin))] <- "X"

# fill empty Lifeboat with X, ? with Q
levels(full_clean$Lifeboat)[match("",levels(full_clean$Lifeboat))] <- "X"
levels(full_clean$Lifeboat)[match("?",levels(full_clean$Lifeboat))] <- "Q"
```

2.2.5 Check cabin and lifeboat categories, Examine full_clean data

```
#check cabin categories
levels(full_clean$Cabin)
```

```
## [1] "X" "A" "B" "C" "D" "E" "F" "G" "T"
```

```
#check lifeboat categories
levels(full_clean$Lifeboat)
```

```
##  [1] "X"     "Q"     "1"     "10"    "11"    "12"    "13"    "14"    "14?"
## [10] "15"    "15?"   "16"    "2"     "3"     "4"     "5"     "6"     "7"
## [19] "8"     "9"     "A"     "A[64]" "B"     "C"     "D"
```

```
#Examine full_clean data again
str(full_clean)
```

```
## 'data.frame':    891 obs. of  11 variables:
##  $ Survived  : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Sex       : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Pclass    : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Age       : num  22 35 26 35 35 22 54 2 26 14 ...
##  $ Fare      : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ SibSp     : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch     : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ FamilySize: num  2 2 1 2 1 1 1 5 3 2 ...
##  $ Embarked  : Factor w/ 4 levels "X","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ Lifeboat  : Factor w/ 25 levels "X","Q","1","10",..: 1 15 9 25 1 1 1 1 10 2 ...
##  $ Cabin     : Factor w/ 9 levels "X","A","B","C",..: 1 4 1 4 1 1 6 1 1 1 ...
```

```
dim(full_clean)
```

```
## [1] 891  11
```

```
#Remove temporary data
rm(full)
```

2.3 Data Preparation

Generate training and test set by 80% and 20% split to make it consistent with the principle that most of the data are in the training set, also align with the 80/20 Pareto principle.

```r
set.seed(8,sample.kind = "Rounding")

#Generate training and test set by 80% and 20% split.
test_index <- createDataPartition(full_clean$Survived, times = 1, p = 0.2, list = FALSE)
train_set <- full_clean[-test_index,]
test_set <- full_clean[test_index,]
```

2.4 Data Exploration and Visualization

2.4.1 Overall Survival rate

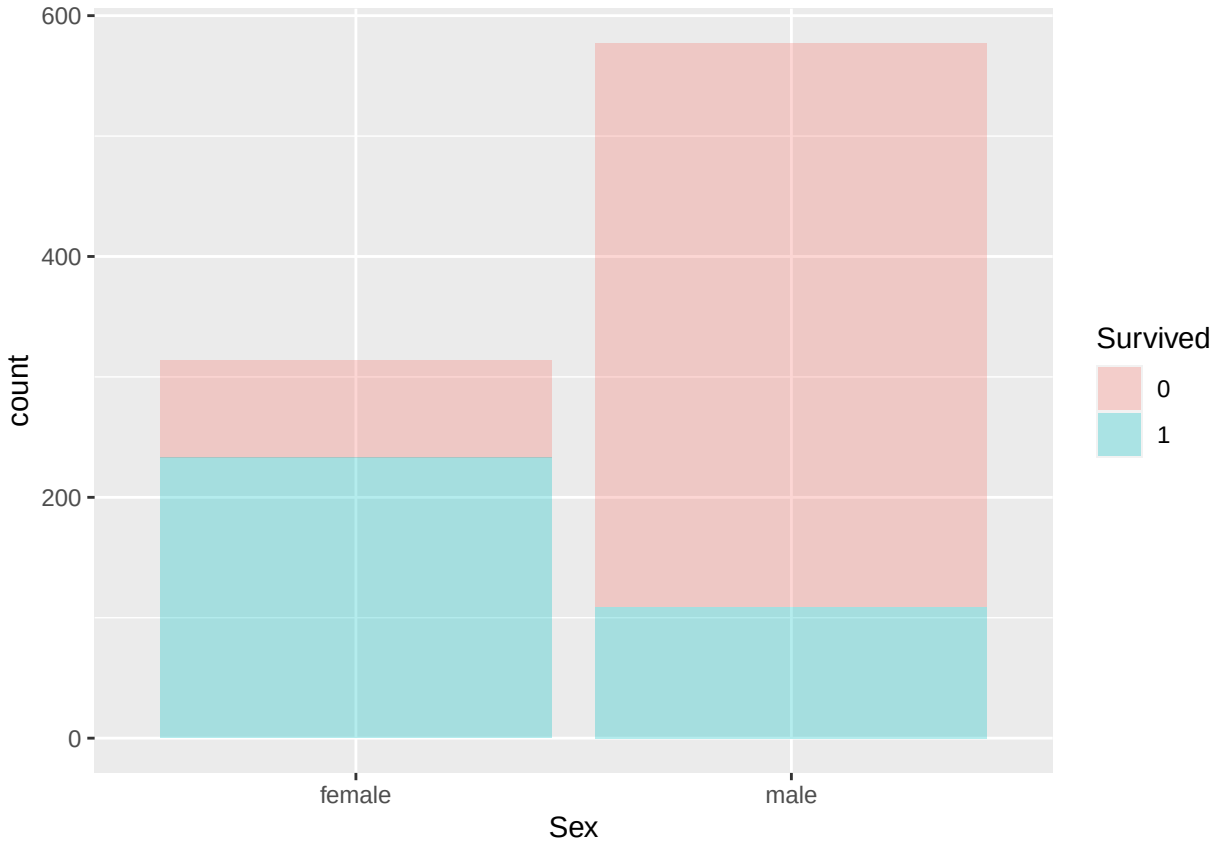
```r
mean(full_clean$Survived == 1)
```

```
## [1] 0.3838384
```

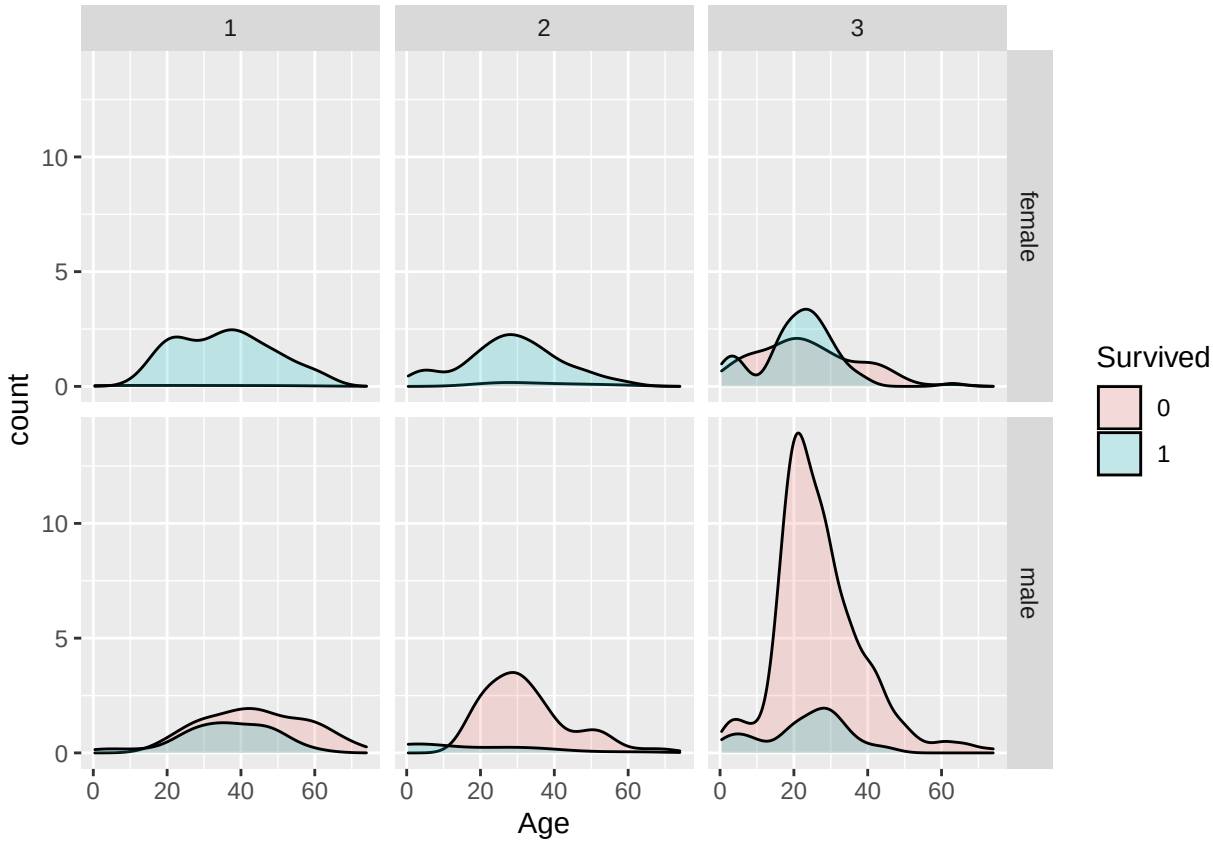
2.4.2 Survival rate by Sex

We can see female survival rate is higher, and among the first two Pclass this is particularly obvious.

```r
full_clean %>%
  ggplot(aes(Sex,y=..count.., fill = Survived)) +
  geom_bar(alpha = 0.3)
```



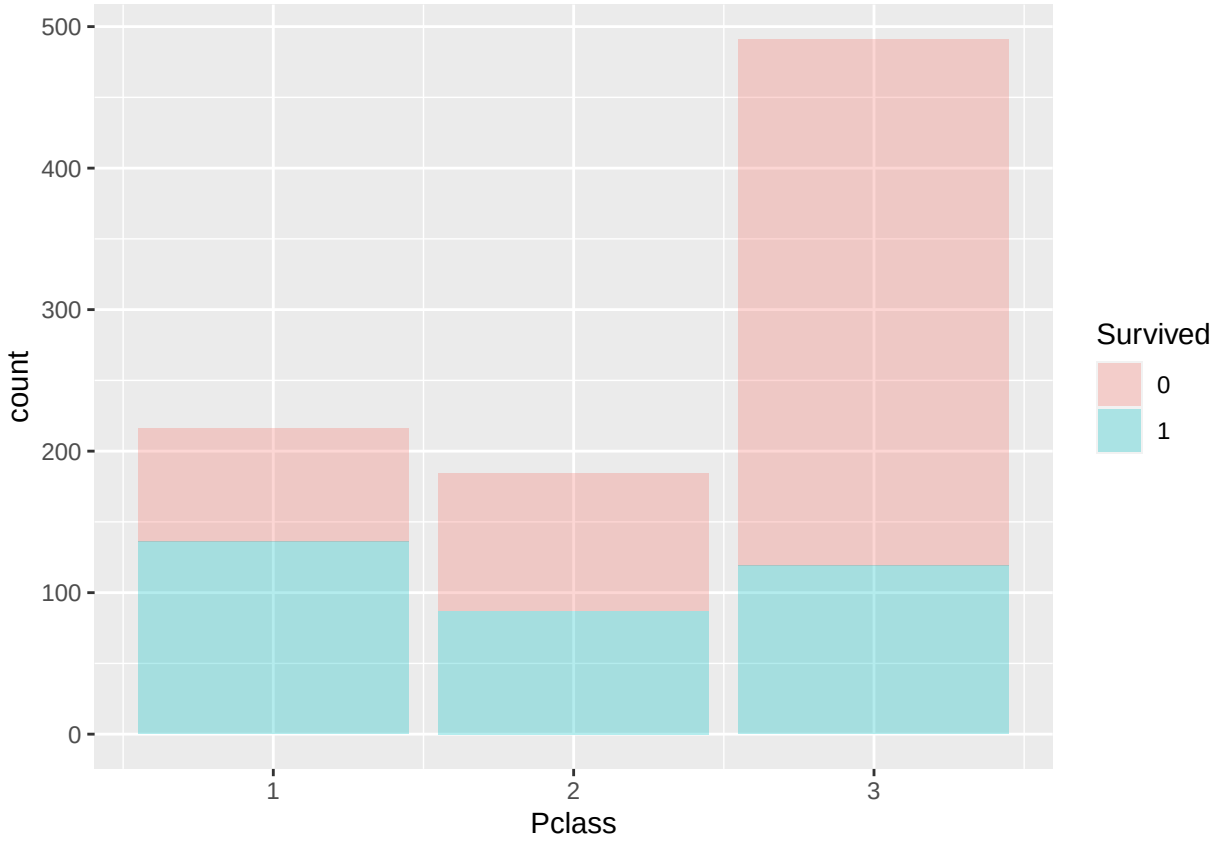
```
full_clean %>%
  ggplot(aes(Age, y = ..count.., fill = Survived)) +
  geom_density(alpha = 0.2, position = "identity") +
  facet_grid(Sex ~ Pclass)
```



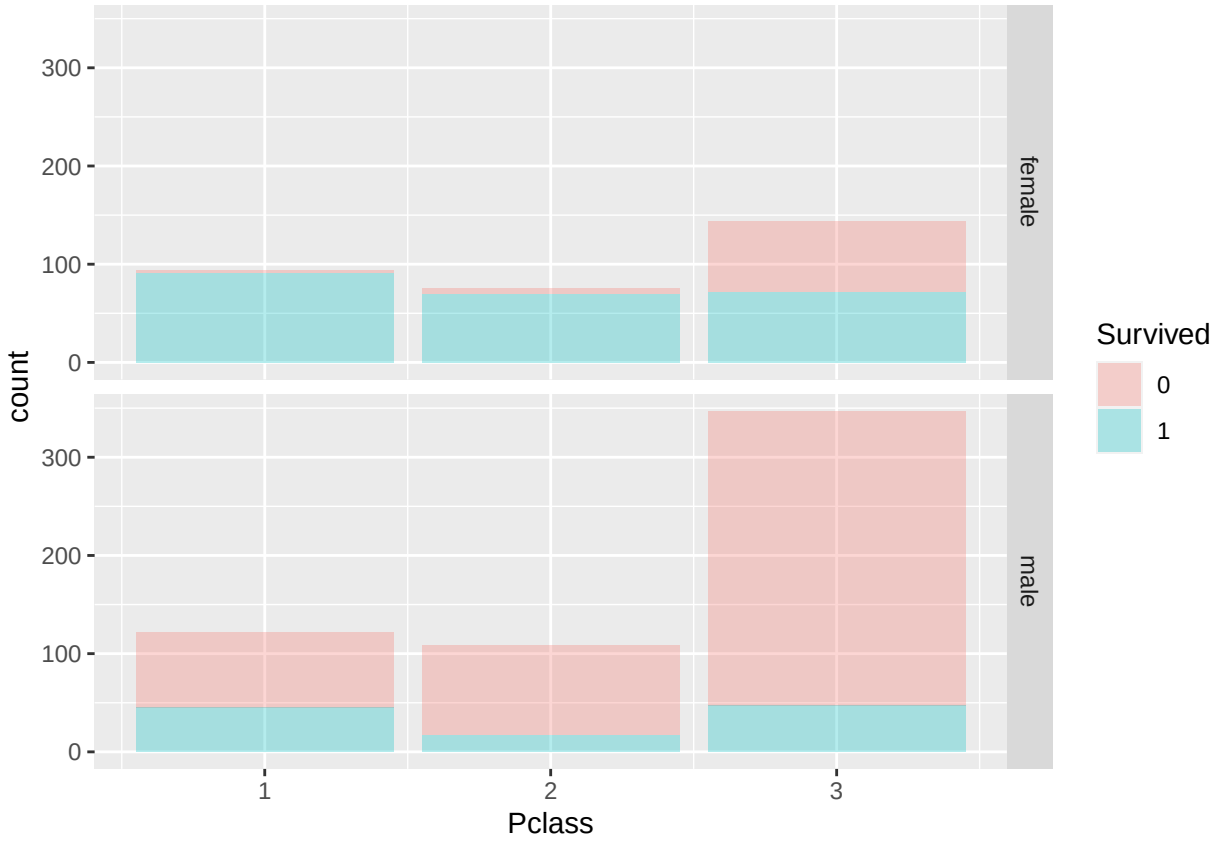
2.4.3 Survival rate by Pclass

Pclass 1 and 2 have higher survival rate, but for male passengers the survival rate is relatively low across all Pclass.

```
full_clean %>%
  ggplot(aes(Pclass,y=..count.., fill = Survived)) +
  geom_bar(alpha = 0.3, position = "stack")
```



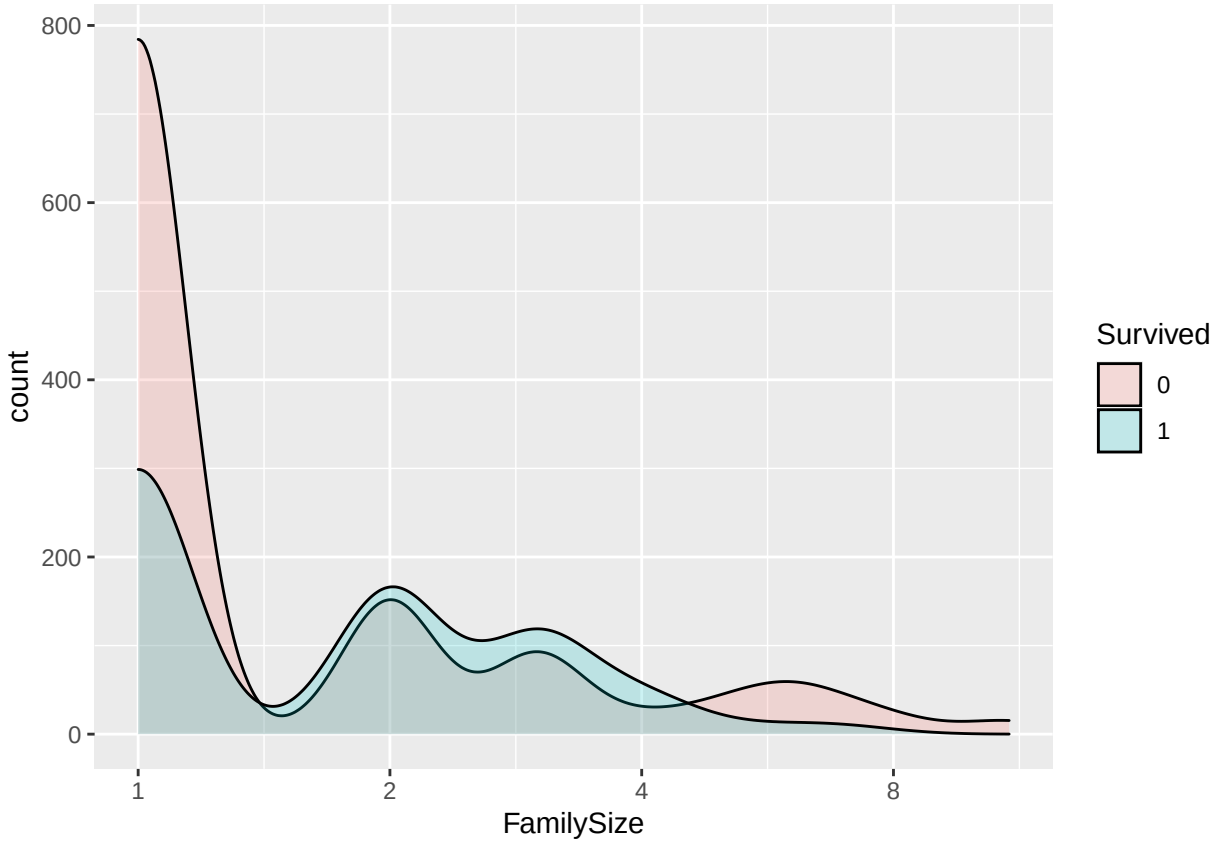

```
full_clean %>%
  ggplot(aes(Pclass, y = ..count.., fill = Survived)) +
  geom_bar(alpha = 0.3, position = "stack") +
  facet_grid(Sex ~ .)
```



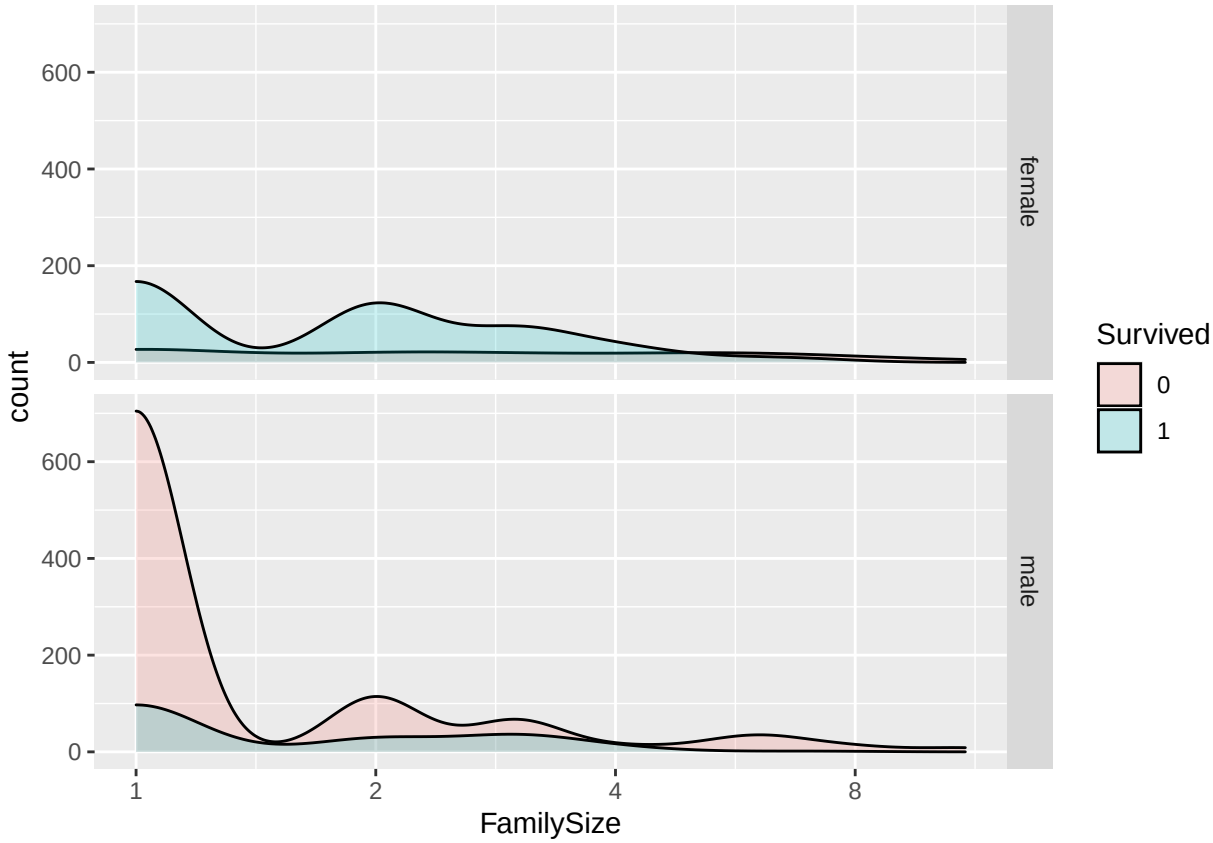
2.4.4 Survival rate by Family size

Male passengers with low family number have lower survival rate; female passengers seem to be the opposite.

```
full_clean %>%
  ggplot(aes(FamilySize, y = ..count.., fill = Survived)) +
  #geom_density(alpha = 0.2, position = "stack") +
  geom_density(alpha = 0.2) +
  scale_x_continuous(trans = "log2")
```



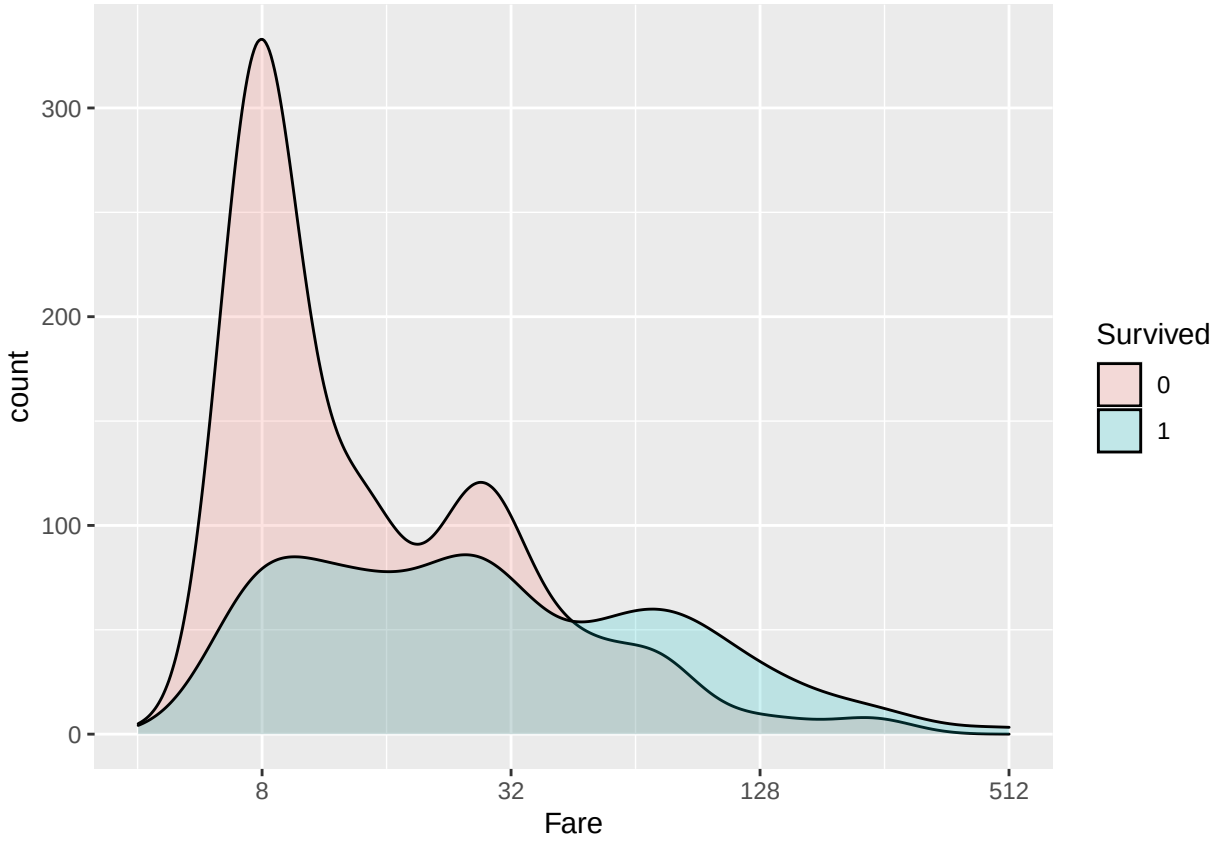

```
full_clean %>%
  ggplot(aes(FamilySize, y = ..count.., fill = Survived)) +
  #geom_density(alpha = 0.2, position = "stack") +
  geom_density(alpha = 0.2) +
  scale_x_continuous(trans = "log2")+
  facet_grid(Sex ~ .)
```



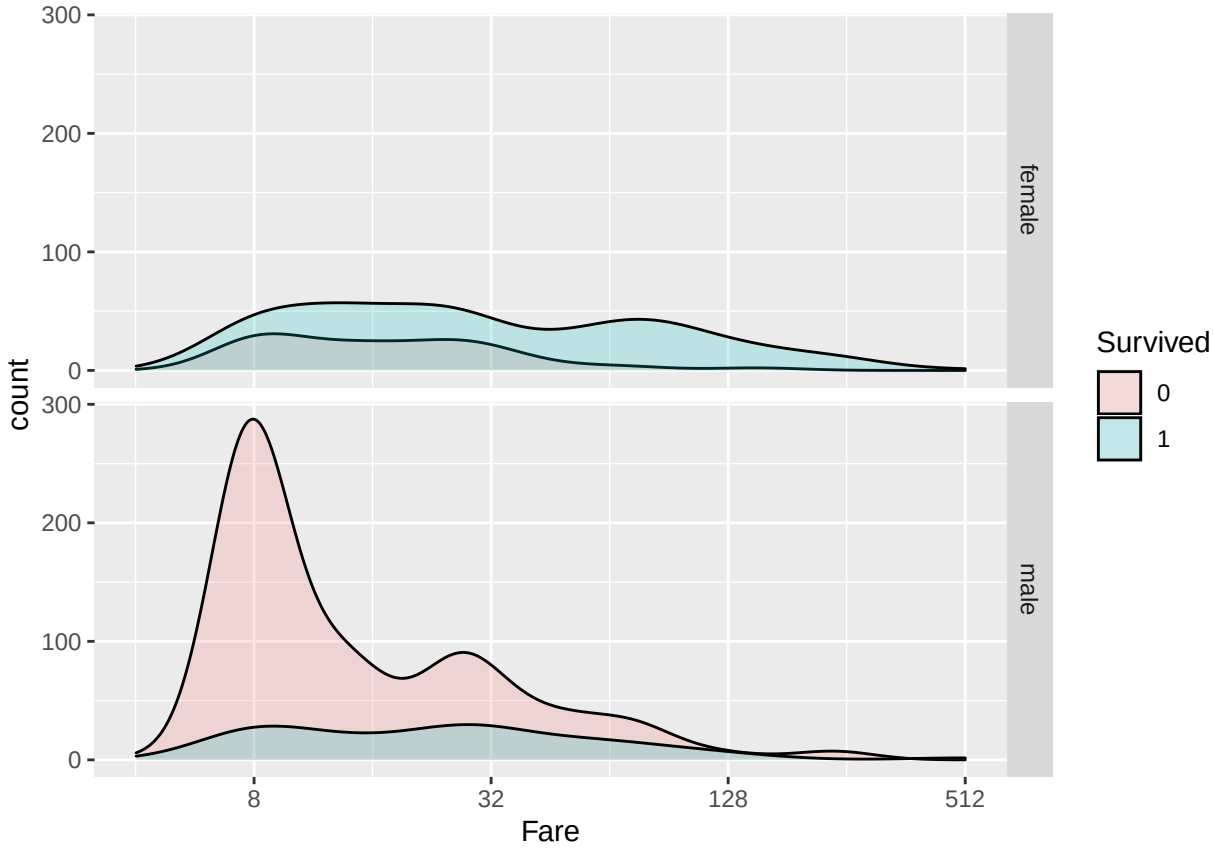
2.4.5 Survival rate by Fare

It seems lower fare indicates lower survival rates, this might be related to Pclass and Cabin, and how easy it is for them to access a Lifeboat.

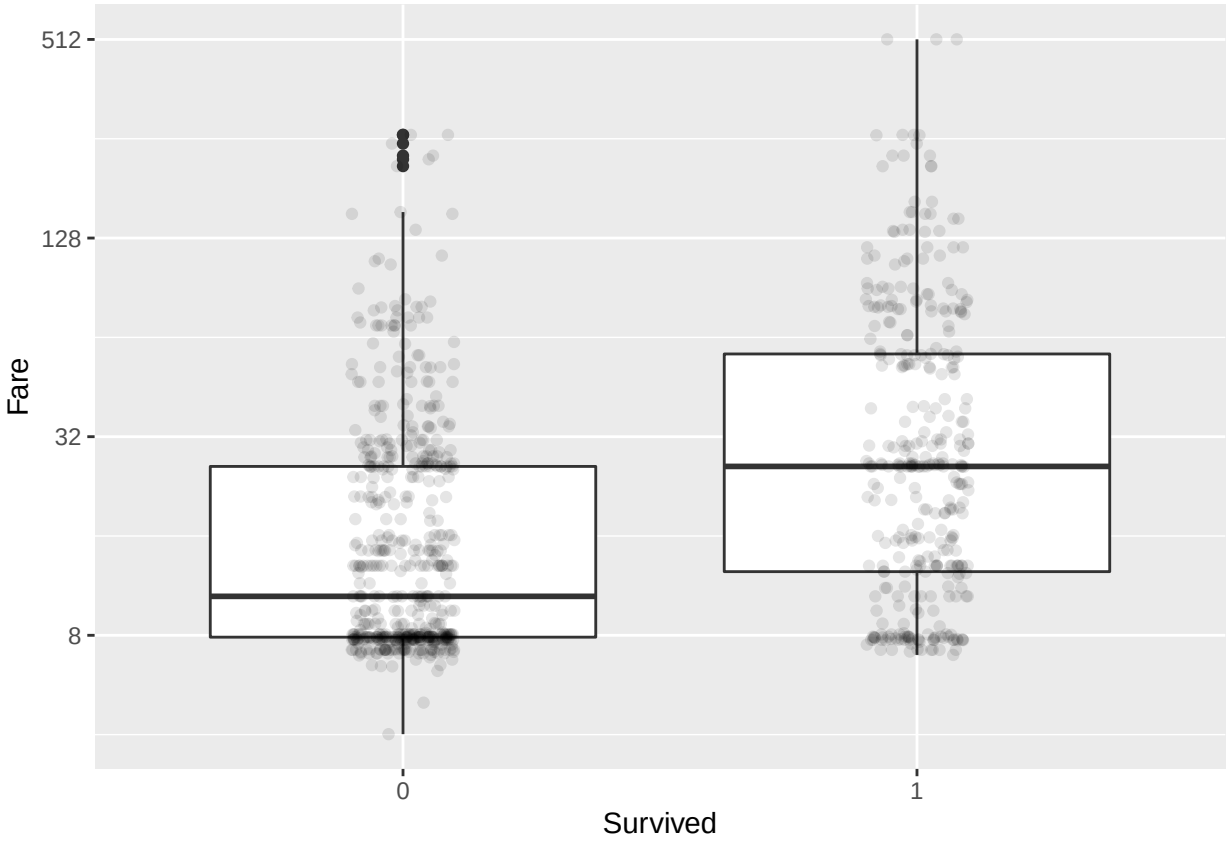
```
full_clean %>%
  ggplot(aes(Fare, y = ..count.., fill = Survived)) +
  geom_density(alpha = 0.2) +
  scale_x_continuous(trans = "log2")
```




```
full_clean %>%
  ggplot(aes(Fare, y = ..count.., fill = Survived)) +
  geom_density(alpha = 0.2) +
  scale_x_continuous(trans = "log2")+
  facet_grid(Sex ~ .)
```



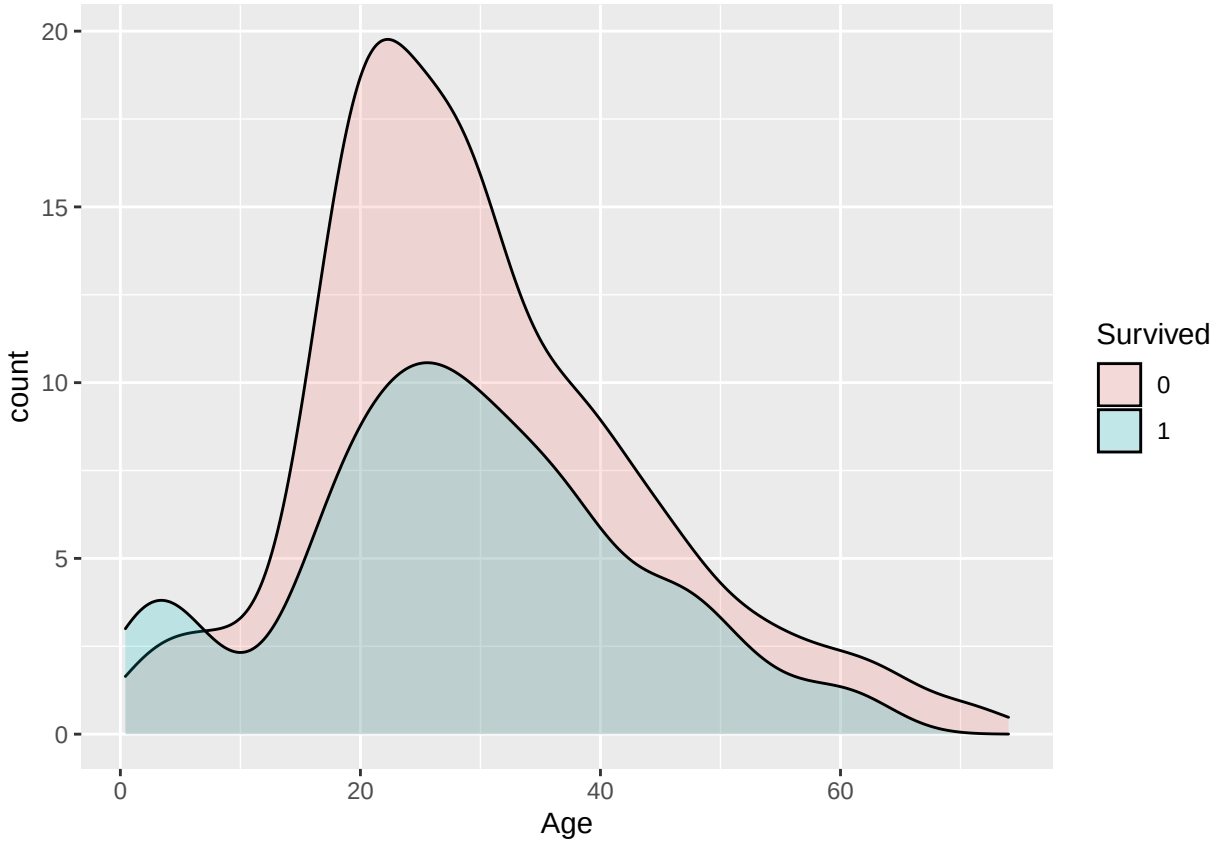

```
full_clean %>%
  filter(Fare > 0) %>%
  ggplot(aes(Survived,Fare)) + geom_boxplot()+
  geom_point(position = position_jitter(width = 0.1), alpha = 0.1) +
  scale_y_continuous(trans = "log2")
```



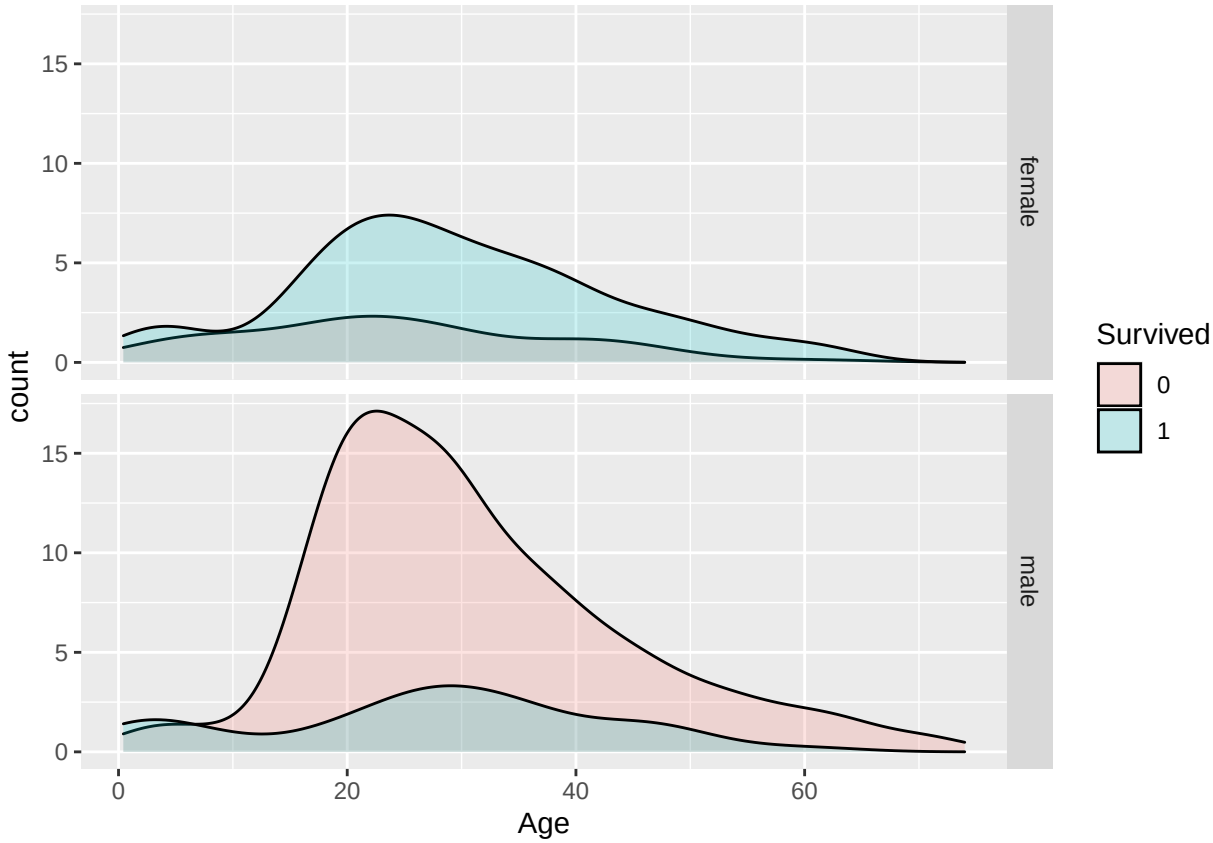
2.4.6 Survival rate by Age

Male passengers aged 20-40 and older passengers have less chance to survive, latter is particularly the case for people aged over 60.

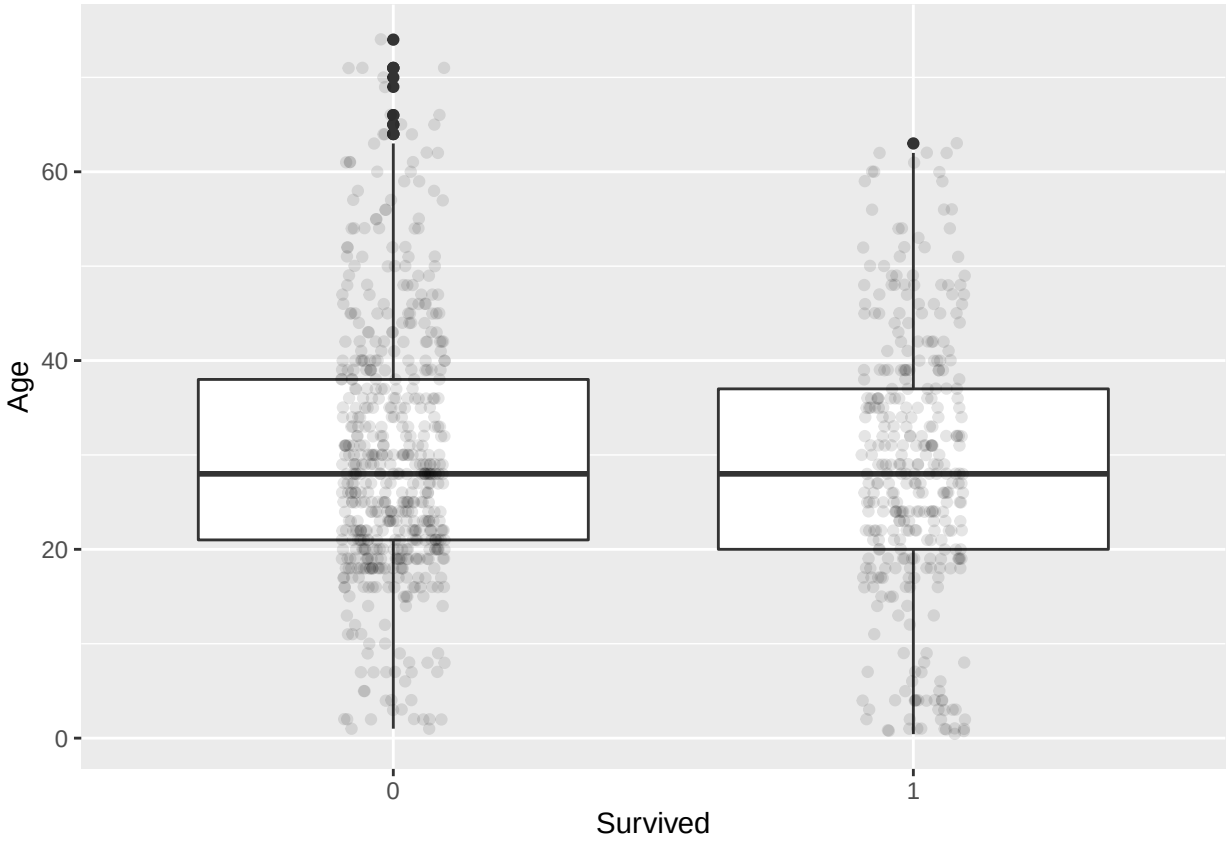
```
full_clean %>%
  ggplot(aes(Age, y = ..count.., fill = Survived)) +
  geom_density(alpha = 0.2)
```




```
full_clean %>%
  ggplot(aes(Age, y = ..count.., fill = Survived)) +
  geom_density(alpha = 0.2) +
  facet_grid(Sex ~ .)
```



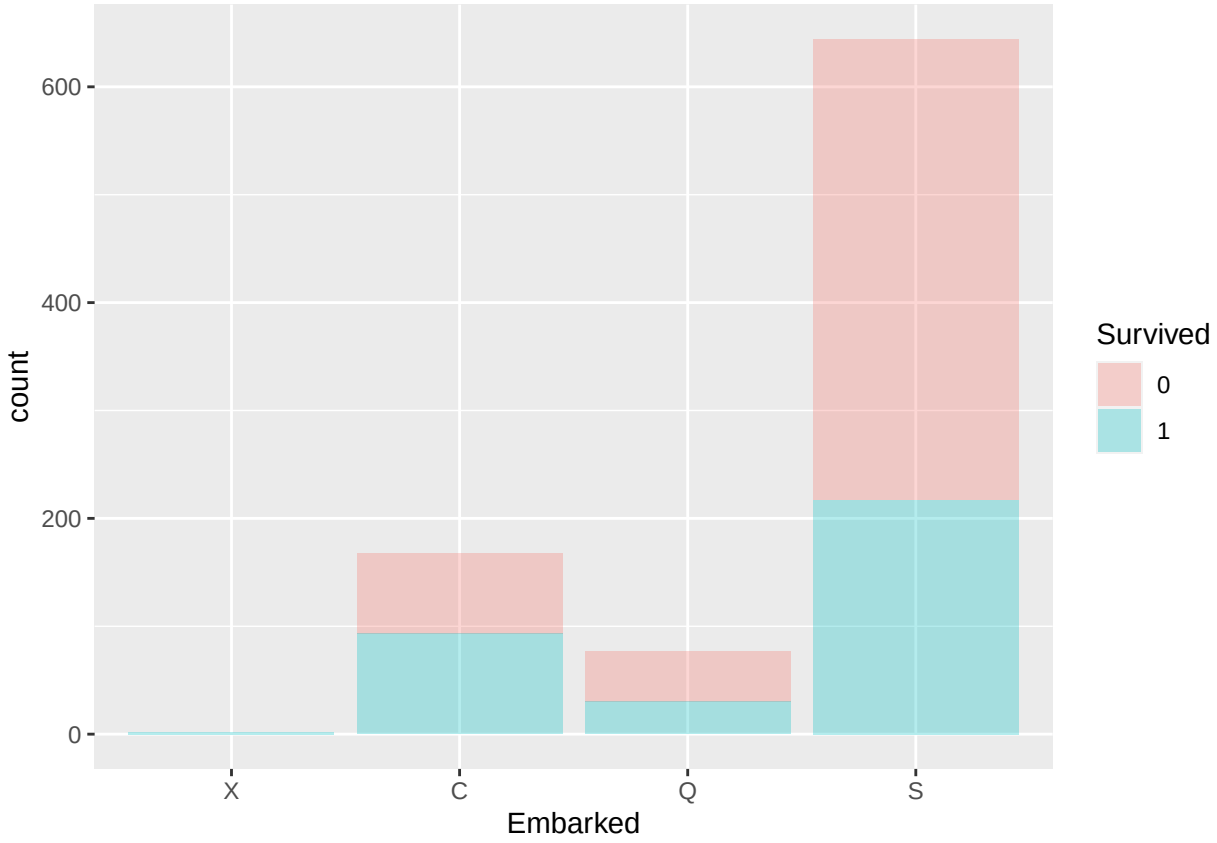

```
full_clean %>%
  ggplot(aes(Survived,Age)) + geom_boxplot()+
  geom_point(position = position_jitter(width = 0.1), alpha = 0.1)
```



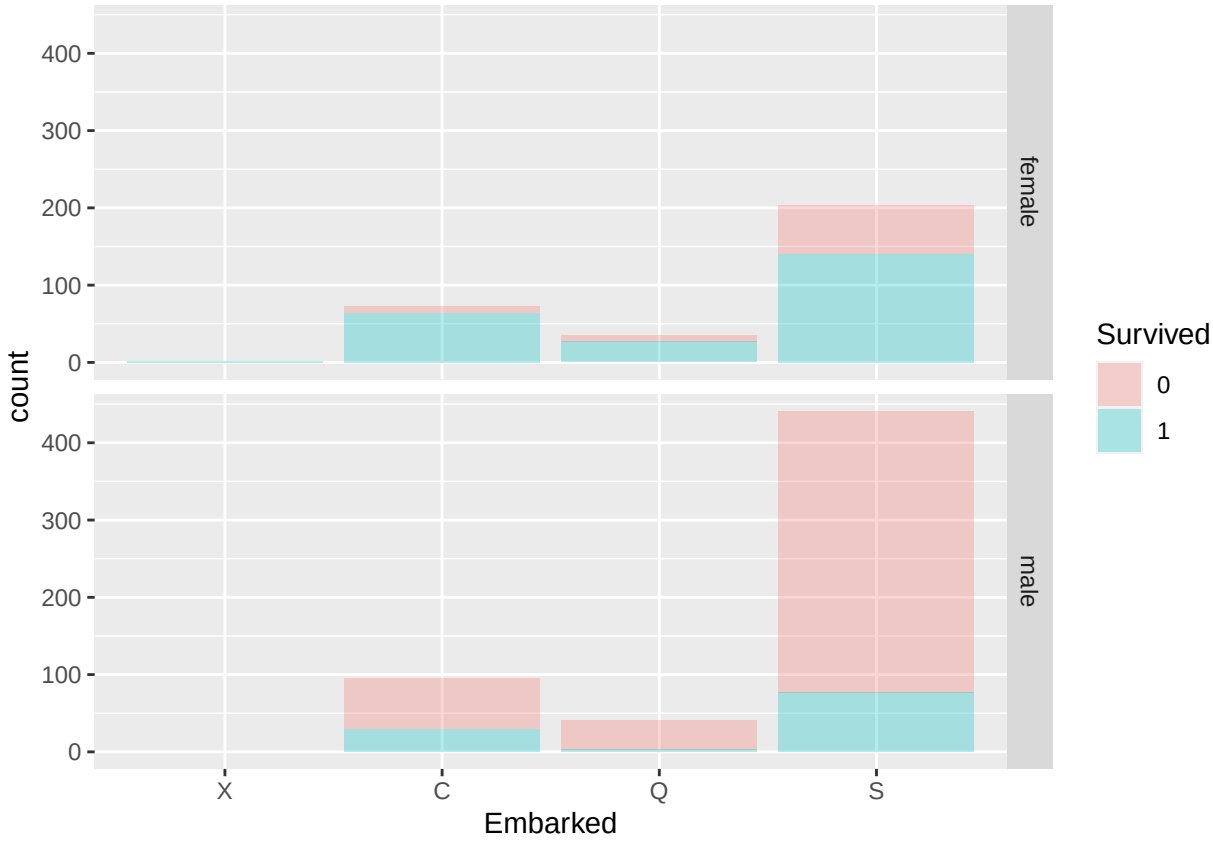
2.4.7 Survival rate by Embarked

Passenger embarked data marked with S seems to has less chance to survive, however this may be to do with the fact that more people are in this category.

```
full_clean %>%
  ggplot(aes(Embarked, y = ..count.., fill = Survived)) +
  geom_bar(alpha = 0.3)
```



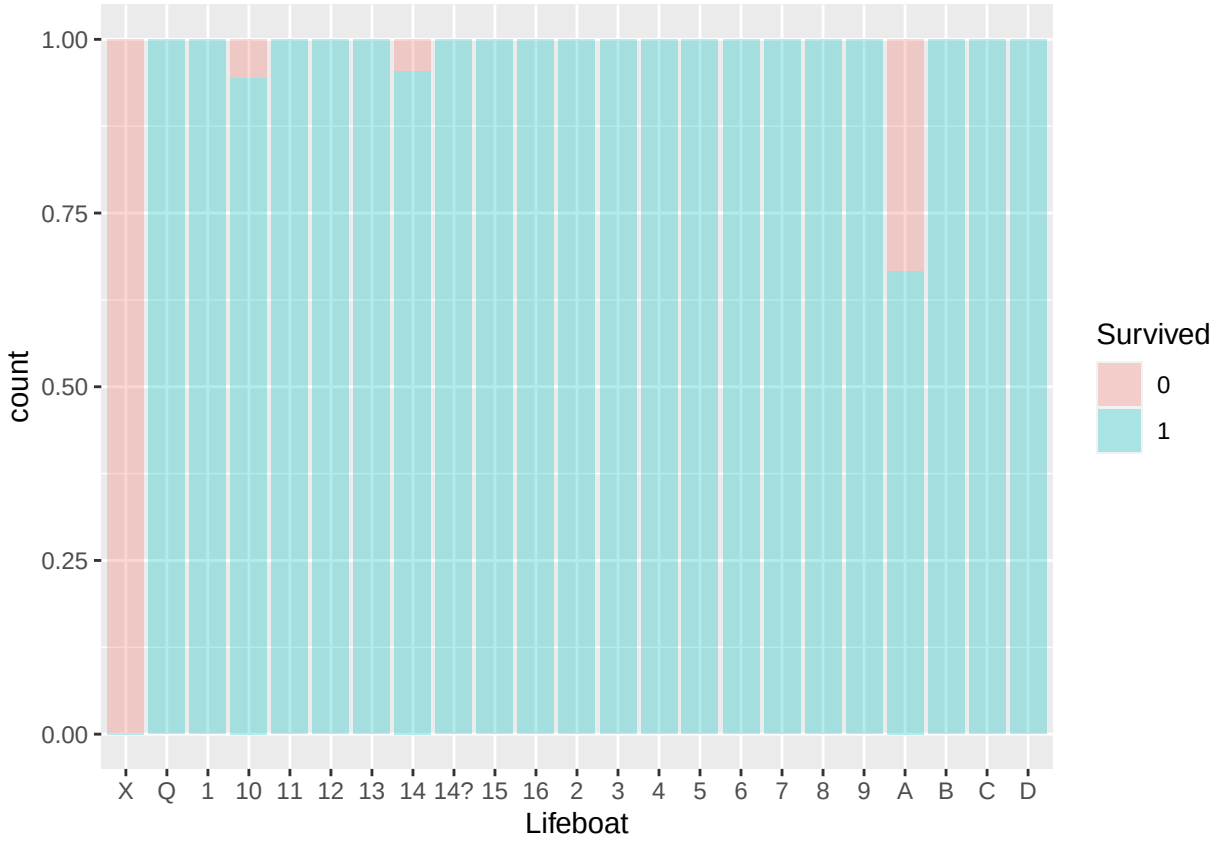

```
full_clean %>%
  ggplot(aes(Embarked, y = ..count.., fill = Survived)) +
  geom_bar(alpha = 0.3) +
  facet_grid(Sex ~ .)
```



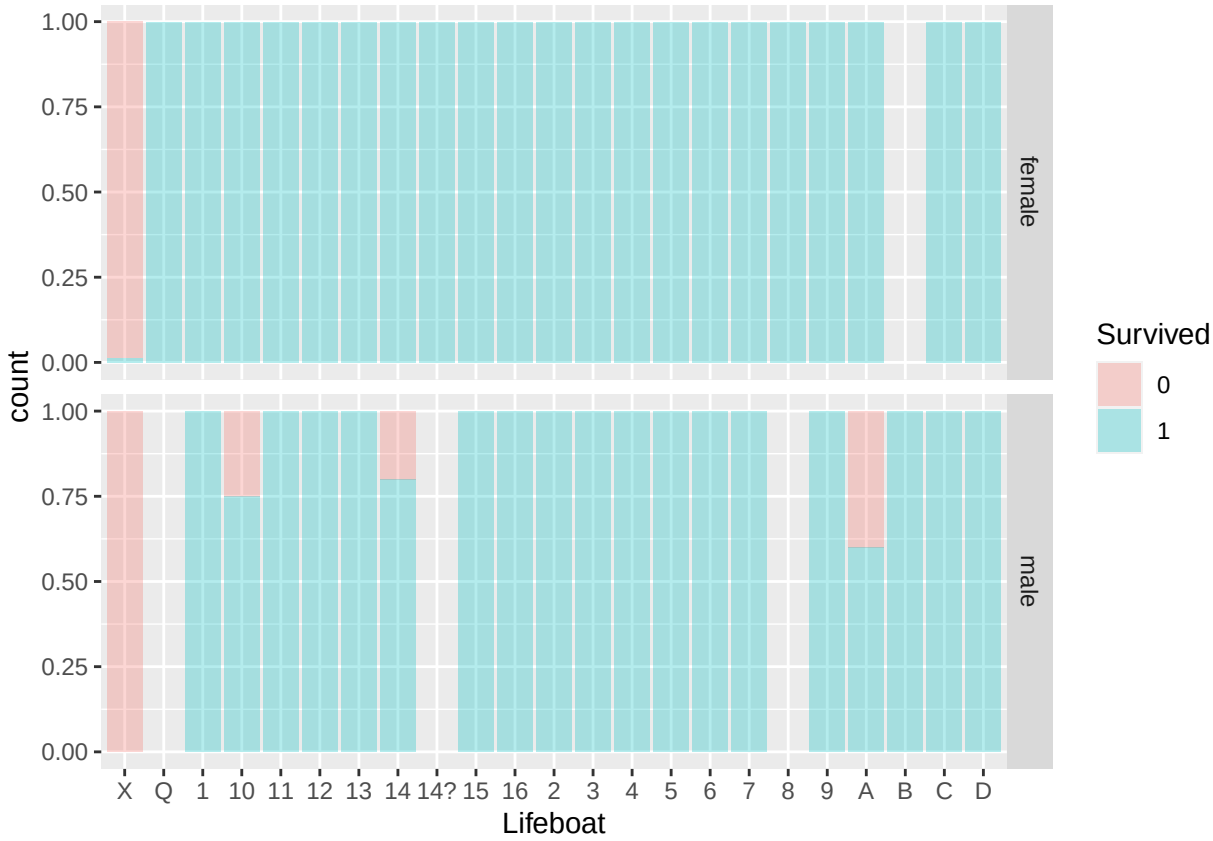
2.4.8 Survival rate by Lifeboat

We can see that passengers that assigned with a lifeboat have a very high chance to survive.

```
full_clean %>%
  ggplot(aes(Lifeboat, y = ..count.., fill = Survived)) +
  geom_bar(alpha = 0.3,position="fill")
```




```
full_clean %>%
  ggplot(aes(Lifeboat, y = ..count.., fill = Survived)) +
  geom_bar(alpha = 0.3,position="fill") +
  facet_grid(Sex ~ .)
```



survival rate when Lifeboat data is empty

```
mean(full_clean[full_clean$Lifeboat=="X",]$Survived == 1)
```

```
## [1] 0.001831502
```

survival rate when Lifeboat data is not empty

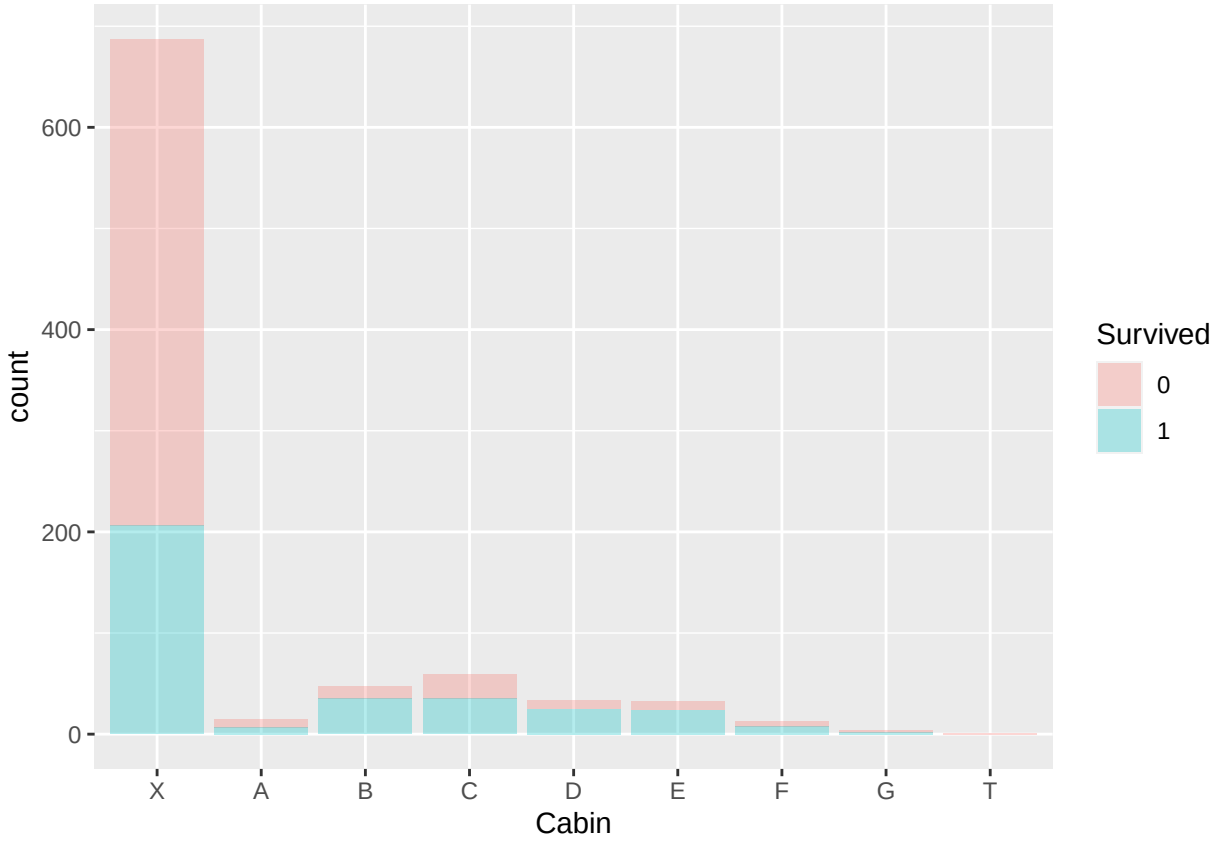
```
mean(full_clean[full_clean$Lifeboat!="X",]$Survived == 1)
```

```
## [1] 0.9884058
```

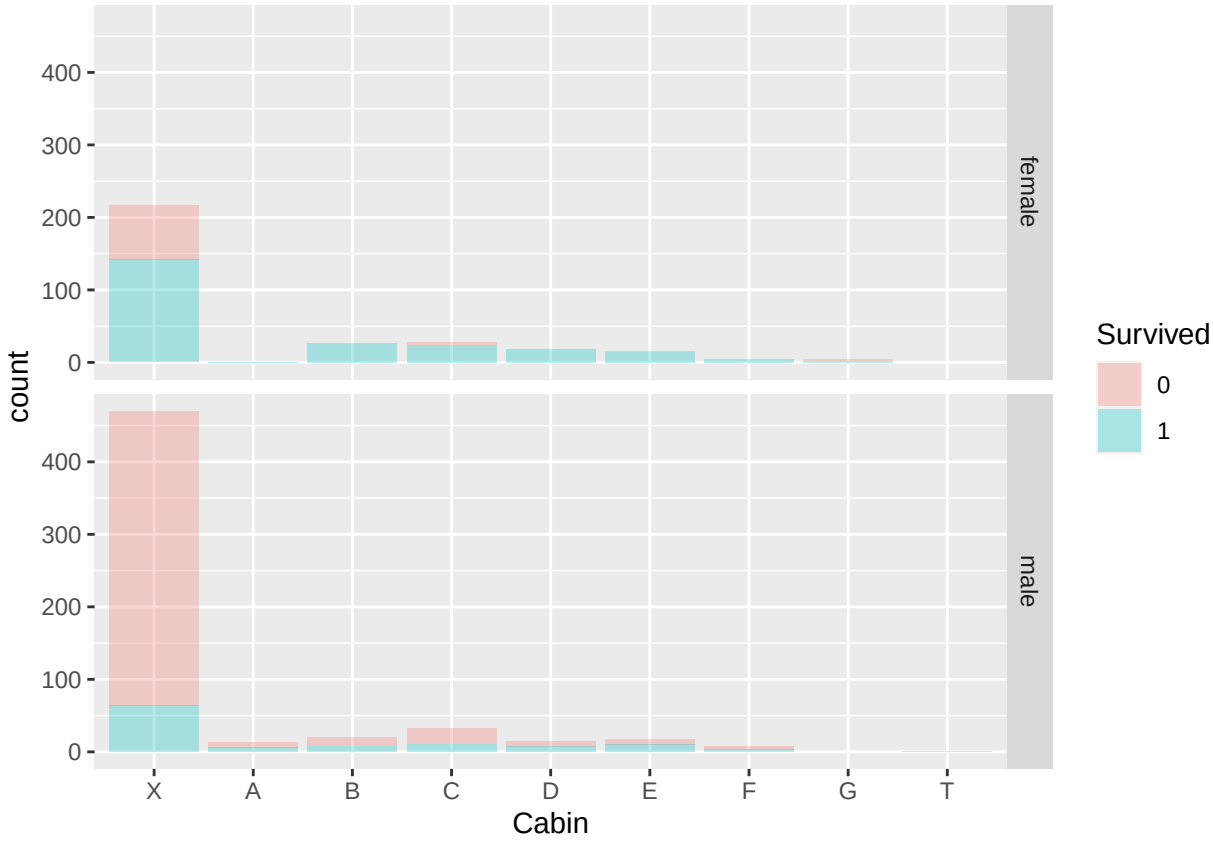
2.4.9 Survival rate by Cabin

Passengers without a cabin have much lower chance to survive.

```
full_clean %>%
  ggplot(aes(Cabin, y = ..count.., fill = Survived)) +
  geom_bar(alpha = 0.3)
```




```
full_clean %>%
  ggplot(aes(Cabin, y = ..count.., fill = Survived)) +
  geom_bar(alpha = 0.3) +
  facet_grid(Sex ~ .)
```



2.5 Modeling Approaches

The data has been split into training and test sets by 80% and 20%. We will use the training data set to experiment different modeling methods, and then use the test data to predict and calculate the prediction accuracy. The project target is set to be 86% accuracy but the final rate is expected to be much higher than this. As the insight we get from previous data analysis and visualization section, the passengers whose Lifeboat data is not empty has a survival rate over 98%, survival rate of those with empty Lifeboat data is merely 0.18%. We can see whether passenger can get on a lifeboat is the deciding factor for their survival. We will remove Lifeboat data first for the training and add the Lifeboat data at the end to see if that confirm our expectation that Lifeboat data is the major deciding factor and will significantly increase the prediction accuracy. It also makes sense to remove the Lifeboat data first since Lifeboat data is unknown before the tragedy happened.

Linear discriminant analysis (LDA) and Quadratic discriminant analysis (QDA) will be first used for the training by assuming that the correlation is the same for all factors, which can hep reduce the number of parameters need to be estimated. Different combinations of factors will be trialed and the ones with the highest prediction accuracy will be chosen. Logistic regression of Generalized linear model with all factors excluding Lifeboat will also be used, so factors used can be modeled as a linear combination of predictors. They will be compared with k-nearest neighbors (kNN), Classification tree and Random forest modelings. Compared to regression models kNN is easier to adapt to multiple dimensions. Classification tree and Random forest can help further reduce the impact of dimensionality. We will then choose the best prediction accuracy among the modelings for our Ensemble model. Ensemble can usually help improve the final results by combining the results of different algorithms. Finally we will add the Lifeboat data back to some models to see if it will significantly increase the prediction accuracy.

3 Results

3.1 Linear discriminant analysis (LDA) model

LDA model with Sex, Pclass, Fare and Age: After test adding other factors, LDA with Sex, Pclass, Fare and Age produces the highest accuracy among different factors using LDA.

```
model_lda <- train(Survived ~ Sex + Pclass + Fare + Age, data = train_set, method = 'lda')
predict_lda <- predict(model_lda, test_set)
#Add project target to result table
result <- tibble(Method = "Project Target", Accuracy = 0.86)

#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "LDA model with Sex,Pclass,Fare,Age",
  Accuracy = mean(test_set$Survived == predict_lda)))
result
```

```
## # A tibble: 2 x 2
##   Method                               Accuracy
##   <chr>                                   <dbl>
## 1 Project Target                         0.86
## 2 LDA model with Sex,Pclass,Fare,Age 0.8603352
```

3.2 Quadratic discriminant analysis (QDA) model

QDA model with Sex, Pclass, Fare, Age and FamilySize: After test adding other factors, QDA with with Sex, Pclass, Fare, Age and FamilySize produces the highest accuracy among different factors using QDA, however it is lower than LDA model result.

```
model_qda <- train(Survived ~ Sex + Pclass + Fare + Age + FamilySize, data = train_set, method = 'qda')
predict_qda <- predict(model_qda, test_set)
#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "QDA model with Sex,Pclass,Fare,Age,FamilySize",
  Accuracy = mean(test_set$Survived == predict_qda)))
result
```

```
## # A tibble: 3 x 2
##   Method                                        Accuracy
##   <chr>                                            <dbl>
## 1 Project Target                                  0.86
## 2 LDA model with Sex,Pclass,Fare,Age              0.8603352
## 3 QDA model with Sex,Pclass,Fare,Age,FamilySize 0.8379888
```

3.3 Logistic regression of Generalized linear model

Logistic regression of Generalized linear model with all data excluding Lifeboat data: Logistic regression model using glm is also lower than LDA results

```
model_log <- glm(Survived ~ . -Lifeboat, data=train_set, family="binomial")
model_log$xlevels[["Embarked"]] <- union(model_log$xlevels[["Embarked"]],
                                         levels(test_set$Embarked))
model_log$xlevels[["Lifeboat"]] <- union(model_log$xlevels[["Lifeboat"]],
                                         levels(test_set$Lifeboat))
#predict testset with trained model
predict_log <- ifelse(predict(model_log, test_set) >= 0, 1, 0)
#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "Logistic regression of glm",
  Accuracy = mean(predict_log == test_set$Survived)))
result
```

```
## # A tibble: 4 x 2
##   Method                                          Accuracy
##   <chr>                                              <dbl>
## 1 Project Target                                  0.86
## 2 LDA model with Sex,Pclass,Fare,Age              0.8603352
## 3 QDA model with Sex,Pclass,Fare,Age,FamilySize 0.8379888
## 4 Logistic regression of glm                      0.8212291
```

estimate variable importance

```
#estimate variable importance
varImp(model_log)
```

```
##               Overall
## Sexmale   11.68185718
## Pclass     4.57267085
## Age        5.32502280
## Fare       0.41831461
## SibSp      3.12838702
## Parch      1.00009816
## EmbarkedQ  0.57497842
## EmbarkedS  1.23109334
## CabinA     1.12197065
## CabinB     1.59876046
## CabinC     0.98910198
## CabinD     2.04659300
## CabinE     3.10288533
## CabinF     1.34840959
## CabinG     0.71090911
## CabinT     0.02315112
```

3.4 kNN model excluding Lifeboat data

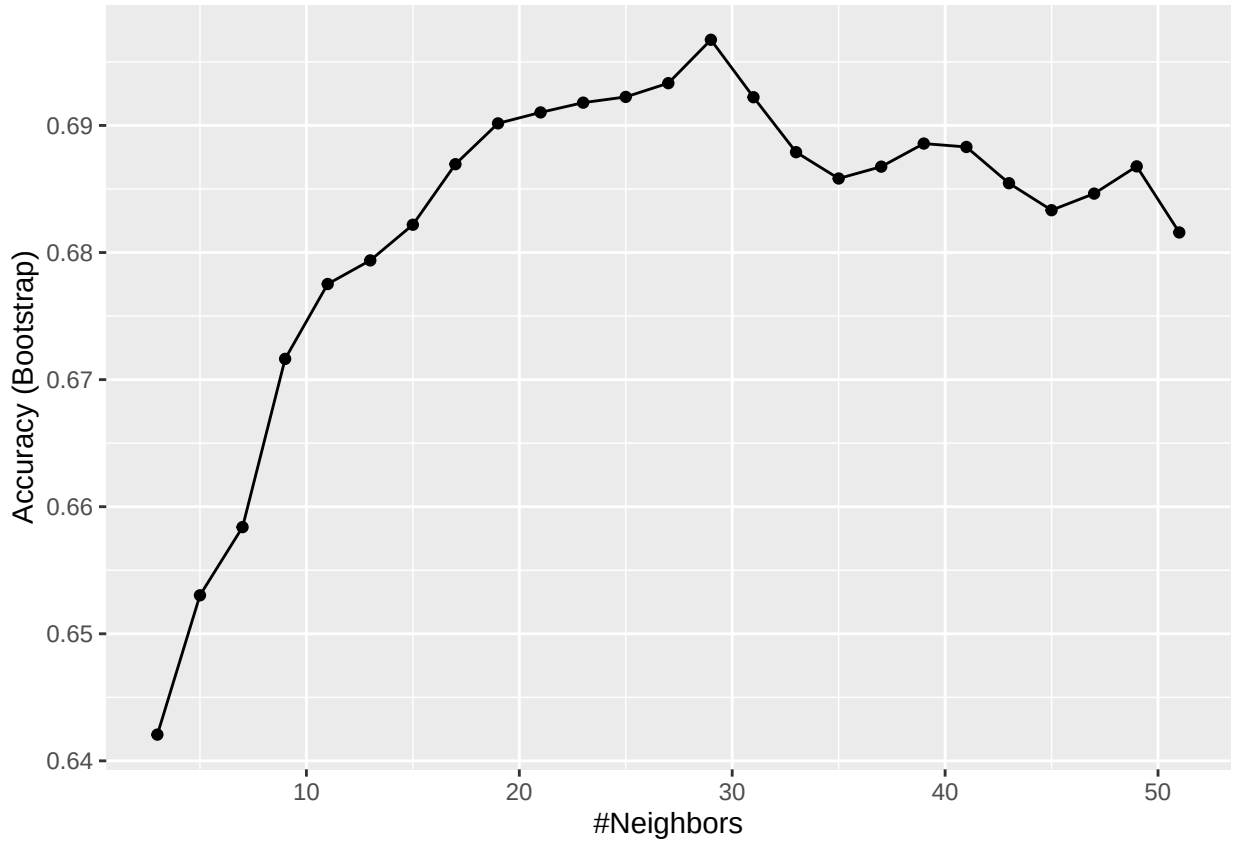
Both kNN and cross-validated kNN models have lower accuracy than previous models. Unlike the training set, The prediction on test set is lower on the cross-validated kNN.

```
k <- seq(3,51,2)
model_knn <- train(Survived ~ . -Lifeboat, data = train_set, method = "knn",
```

```
                    tuneGrid = data.frame(k))
#find the best k
model_knn$bestTune
```

```
##     k
## 14 29
```

```
#show knn plot
ggplot(model_knn)
```



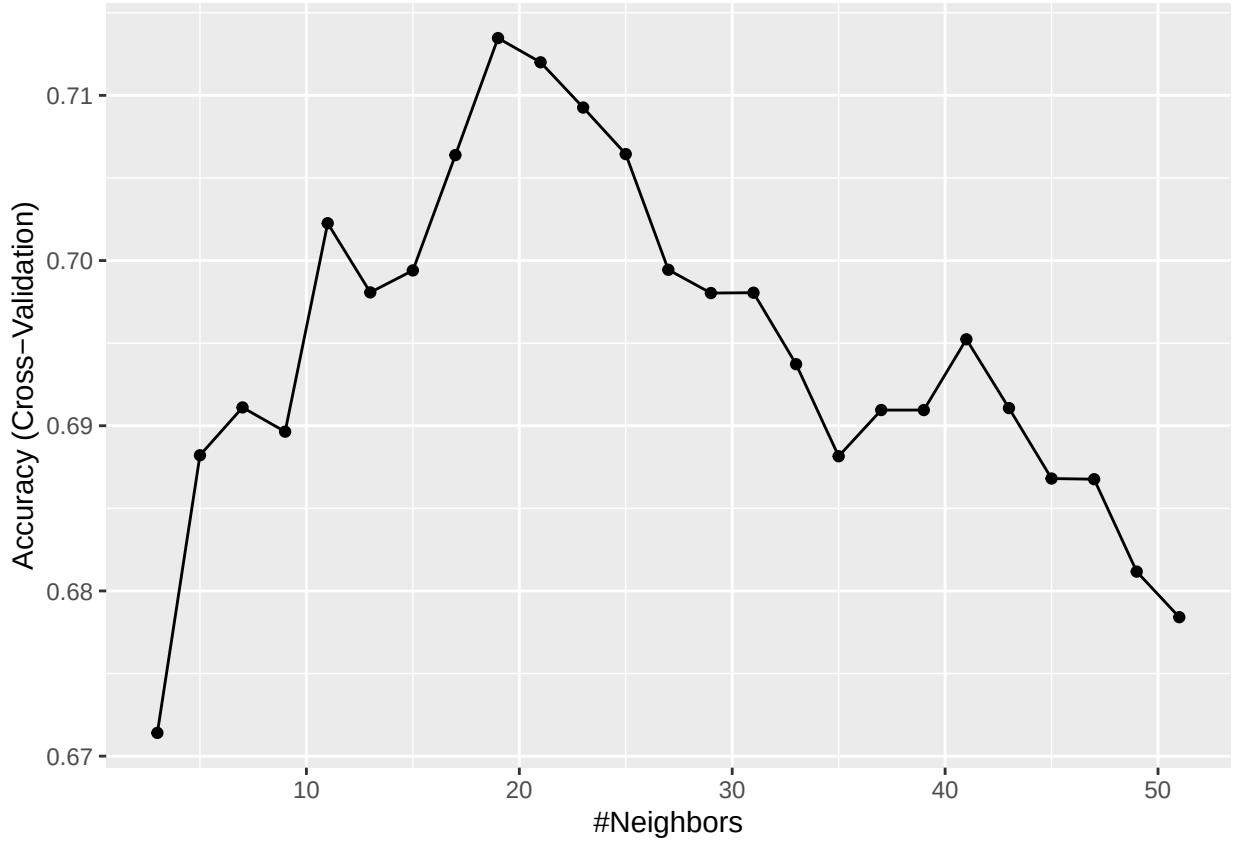

```
#predict testset with trained model
predict_knn <- predict(model_knn, test_set) %>% factor(levels = levels(test_set$Survived))
cm_test <- confusionMatrix(data = predict_knn, reference = test_set$Survived)

#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "kNN model",
  Accuracy = cm_test$overall["Accuracy"]))
result
```

```
## # A tibble: 5 x 2
##   Method                                    Accuracy
##   <chr>                                        <dbl>
## 1 Project Target                             0.86
```

```
## 2 LDA model with Sex,Pclass,Fare,Age            0.8603352
## 3 QDA model with Sex,Pclass,Fare,Age,FamilySize 0.8379888
## 4 Logistic regression of glm                    0.8212291
## 5 kNN model                                     0.7094972
```

3.5 Cross-validated kNN model excluding Lifeboat data

```
model_knn_cv <- train(Survived ~ . -Lifeboat,
                      data=train_set,
                      method = "knn",
                      tuneGrid = data.frame(k = seq(3, 51, 2)),
                      trControl = trainControl(method = "cv", number=10, p=0.9))
#show model plot
ggplot(model_knn_cv)
```



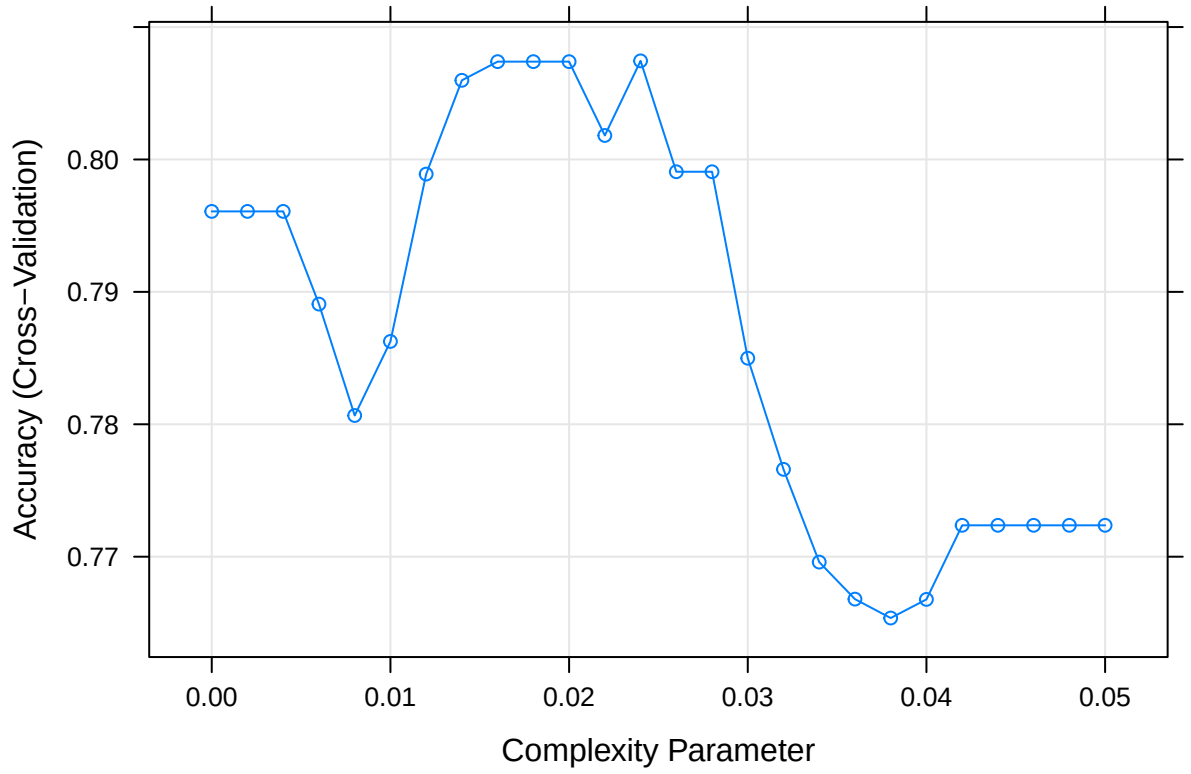

```
#predict testset with trained model
predict_knn_cv <- predict(model_knn_cv, test_set)
cm_test <- confusionMatrix(data = predict_knn_cv, reference = test_set$Survived)

#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "cross-validated kNN model",
  Accuracy = cm_test$overall["Accuracy"]))
result
```

```
## # A tibble: 6 x 2
##   Method                                          Accuracy
##   <chr>                                              <dbl>
## 1 Project Target                                  0.86
## 2 LDA model with Sex,Pclass,Fare,Age              0.8603352
## 3 QDA model with Sex,Pclass,Fare,Age,FamilySize 0.8379888
## 4 Logistic regression of glm                      0.8212291
## 5 kNN model                                       0.7094972
## 6 cross-validated kNN model                       0.6871508
```

3.6 Classification tree model excluding Lifeboat data

The classification tree model has significantly higher performance than other models

```
model_rpart <- train(Survived ~ . -Lifeboat,
                     data=train_set,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0, 0.05, 0.002)),
                    trControl = trainControl(method = "cv", number=10, p=0.9))
#show model plot
plot(model_rpart)
```




```
#predict testset with trained model
predict_rpart <- predict(model_rpart, test_set)
```

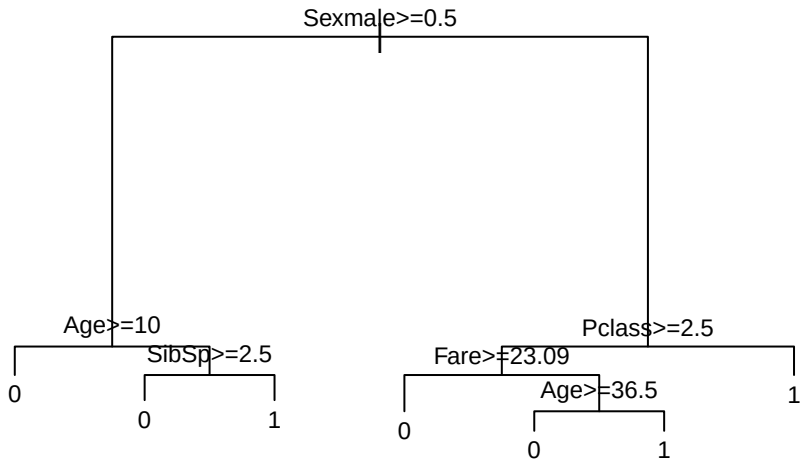
```r
cm_test <- confusionMatrix(data = predict_rpart, reference = test_set$Survived)
#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "Classification tree model",
  Accuracy = cm_test$overall["Accuracy"]))
result
```

```
## # A tibble: 7 x 2
##   Method                                          Accuracy
##   <chr>                                              <dbl>
## 1 Project Target                                  0.86
## 2 LDA model with Sex,Pclass,Fare,Age              0.8603352
## 3 QDA model with Sex,Pclass,Fare,Age,FamilySize 0.8379888
## 4 Logistic regression of glm                      0.8212291
## 5 kNN model                                       0.7094972
## 6 cross-validated kNN model                       0.6871508
## 7 Classification tree model                       0.8715084
```

```r
#print and visualize final model
model_rpart$finalModel
```

```
## n= 712
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 712 273 0 (0.61657303 0.38342697)
##    2) Sexmale>=0.5 463  93 0 (0.79913607 0.20086393)
##      4) Age>=10 435  75 0 (0.82758621 0.17241379) *
##      5) Age< 10 28  10 1 (0.35714286 0.64285714)
##       10) SibSp>=2.5 11   1 0 (0.90909091 0.09090909) *
##       11) SibSp< 2.5 17   0 1 (0.00000000 1.00000000) *
##    3) Sexmale< 0.5 249  69 1 (0.27710843 0.72289157)
##      6) Pclass>=2.5 117  56 0 (0.52136752 0.47863248)
##       12) Fare>=23.0875 20   1 0 (0.95000000 0.05000000) *
##       13) Fare< 23.0875 97  42 1 (0.43298969 0.56701031)
##         26) Age>=36.5 9   1 0 (0.88888889 0.11111111) *
##         27) Age< 36.5 88  34 1 (0.38636364 0.61363636) *
##      7) Pclass< 2.5 132   8 1 (0.06060606 0.93939394) *
```

```r
plot(model_rpart$finalModel, margin=0.1)
text(model_rpart$finalModel, cex = 0.75)
```



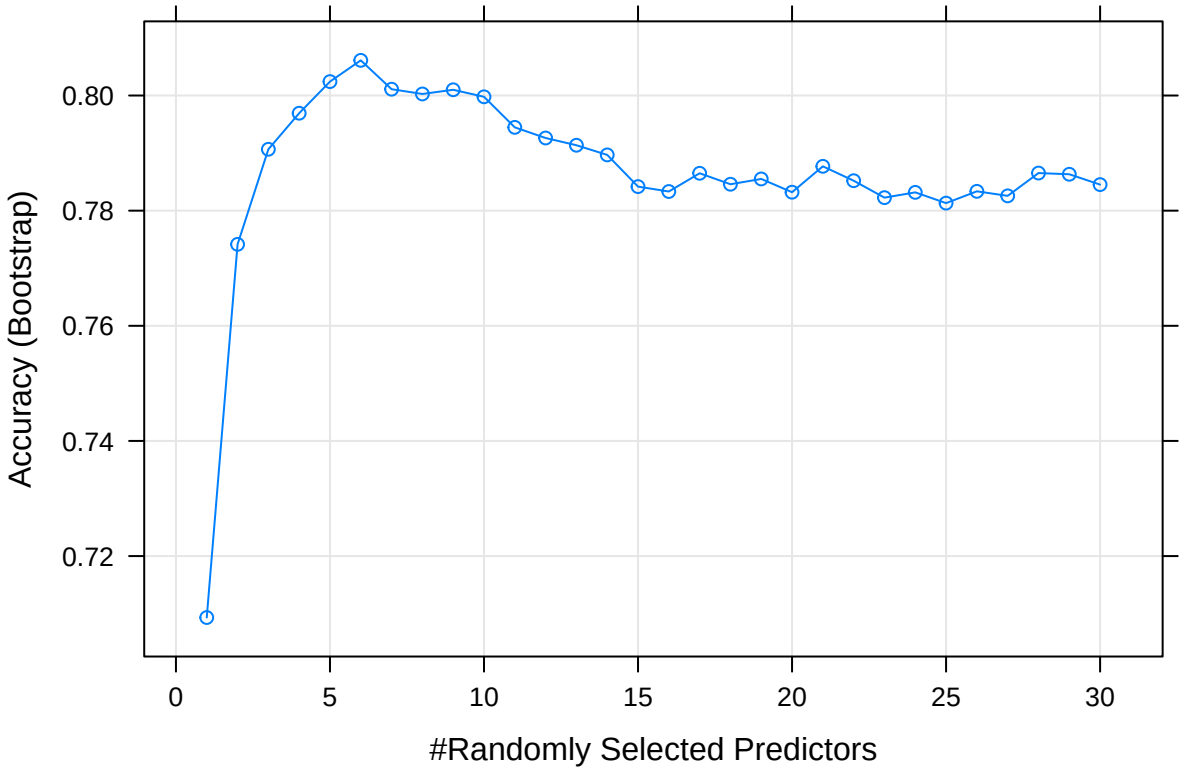
3.7 Random forest model excluding Lifeboat data

Random forest model accuracy on test set prediction is a bit lower than the classification tree model but it is higher than other models.

```
model_rf <- train(Survived ~. -Lifeboat,
                  data = train_set,
                  method = "rf",
                  tuneGrid = data.frame(mtry = seq(1, 30)),
                  ntree = 100)
#best tuning value
model_rf$bestTune
```

```
##   mtry
## 6    6
```

```
#show model plot
plot(model_rf)
```




```
#predict testset with trained model
predict_rf <- predict(model_rf, test_set)

#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "Random forest model",
  Accuracy = mean(predict_rf == test_set$Survived)))
result
```

```
## # A tibble: 8 x 2
##   Method                                        Accuracy
##   <chr>                                            <dbl>
## 1 Project Target                                0.86
## 2 LDA model with Sex,Pclass,Fare,Age            0.8603352
## 3 QDA model with Sex,Pclass,Fare,Age,FamilySize 0.8379888
## 4 Logistic regression of glm                    0.8212291
## 5 kNN model                                     0.7094972
## 6 cross-validated kNN model                     0.6871508
## 7 Classification tree model                     0.8715084
## 8 Random forest model                           0.8659218
```

estimate variable importance

```
#estimate variable importance
varImp(model_rf)
```

```
## rf variable importance
##
##             Overall
## Sexmale    100.0000
## Age         75.1845
## Fare        74.2501
## Pclass      32.7349
## FamilySize  19.1789
## SibSp       11.6722
## Parch       10.5443
## CabinE       6.4578
## EmbarkedS    5.8413
## CabinB       3.8356
## EmbarkedC    3.3522
## CabinC       2.8872
## CabinD       2.5560
## EmbarkedQ    2.3907
## CabinA       1.2409
## CabinF       0.6807
## CabinG       0.4879
## CabinT       0.0000
```

3.8 Ensemble of different models

We will choose the models with accuracy higher than 80% for our ensemble: LDA, QDA, Logistic regression, Classification tree model and Random forest. We get the same result as our Random forest.

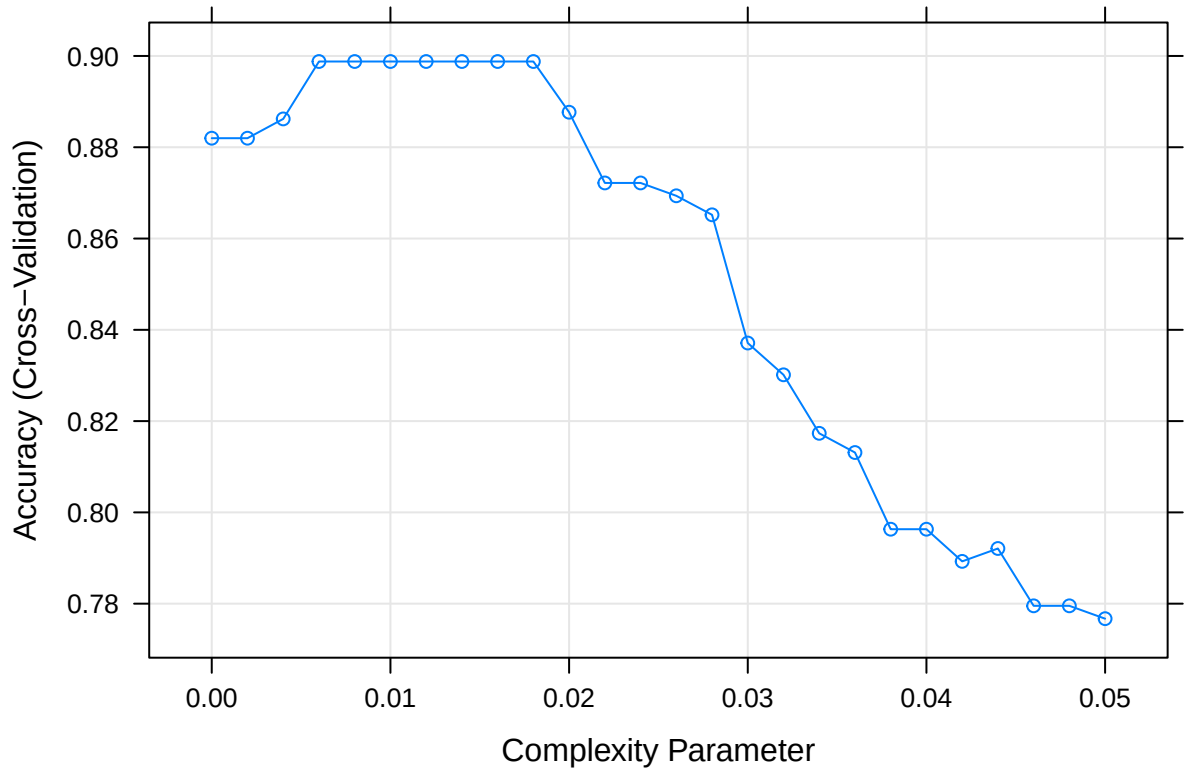
```
ensemble <- cbind(lda=ifelse(predict_lda == "0", 0, 1),
                  qda=ifelse(predict_qda == "0", 0, 1),
                  log=ifelse(predict_log == "0", 0, 1),
                  rpart=ifelse(predict_rpart == "0", 0, 1),
                  rf=ifelse(predict_rf== "0", 0, 1))
#ensemble prediction
ensemble_predict <- ifelse(rowMeans(ensemble) < 0.5, 0, 1)

#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "Ensemble of different models",
  Accuracy = mean(ensemble_predict == test_set$Survived)))
result
```

```
## # A tibble: 9 x 2
##   Method                                          Accuracy
##   <chr>                                              <dbl>
## 1 Project Target                                  0.86
## 2 LDA model with Sex,Pclass,Fare,Age              0.8603352
## 3 QDA model with Sex,Pclass,Fare,Age,FamilySize 0.8379888
## 4 Logistic regression of glm                      0.8212291
## 5 kNN model                                       0.7094972
## 6 cross-validated kNN model                       0.6871508
## 7 Classification tree model                       0.8715084
## 8 Random forest model                             0.8659218
## 9 Ensemble of different models                    0.8659218
```

3.9 Classification tree model including Lifeboat data

Finally we add Lifeboat data back to the training. We get over 92% from the classification tree model.

```
model_rpart_lb <- train(Survived ~ .,
                        data=train_set,
                        method = "rpart",
                        tuneGrid = data.frame(cp = seq(0, 0.05, 0.002)),
                      trControl = trainControl(method = "cv", number=10, p=0.9))
#plot model
plot(model_rpart_lb)
```




```
#predict testset with trained model
predict_rpart_lb <- predict(model_rpart_lb, test_set)
cm_test <- confusionMatrix(data = predict_rpart_lb, reference = test_set$Survived)
#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "Classification tree model including Lifeboat",
  Accuracy = cm_test$overall["Accuracy"]))
result
```

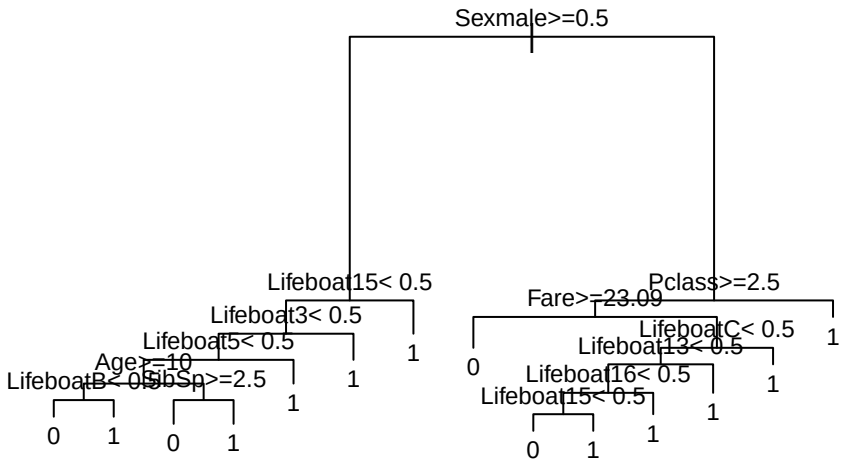
```
## # A tibble: 10 x 2
##    Method                                          Accuracy
##    <chr>                                              <dbl>
##  1 Project Target                                    0.86
```

```
##  2 LDA model with Sex,Pclass,Fare,Age            0.8603352
##  3 QDA model with Sex,Pclass,Fare,Age,FamilySize 0.8379888
##  4 Logistic regression of glm                    0.8212291
##  5 kNN model                                     0.7094972
##  6 cross-validated kNN model                     0.6871508
##  7 Classification tree model                     0.8715084
##  8 Random forest model                           0.8659218
##  9 Ensemble of different models                  0.8659218
## 10 Classification tree model including Lifeboat  0.9217877
```

```
#print and visualize final model
model_rpart_lb$finalModel
```

```
## n= 712
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 712 273 0 (0.61657303 0.38342697)
##     2) Sexmale>=0.5 463  93 0 (0.79913607 0.20086393)
##       4) Lifeboat15< 0.5 449  79 0 (0.82405345 0.17594655)
##         8) Lifeboat3< 0.5 438  68 0 (0.84474886 0.15525114)
##          16) Lifeboat5< 0.5 428  58 0 (0.86448598 0.13551402)
##            32) Age>=10 402  42 0 (0.89552239 0.10447761)
##              64) LifeboatB< 0.5 395  35 0 (0.91139241 0.08860759) *
##              65) LifeboatB>=0.5 7   0 1 (0.00000000 1.00000000) *
##            33) Age< 10 26  10 1 (0.38461538 0.61538462)
##              66) SibSp>=2.5 10   0 0 (1.00000000 0.00000000) *
##              67) SibSp< 2.5 16   0 1 (0.00000000 1.00000000) *
##          17) Lifeboat5>=0.5 10   0 1 (0.00000000 1.00000000) *
##         9) Lifeboat3>=0.5 11   0 1 (0.00000000 1.00000000) *
##       5) Lifeboat15>=0.5 14   0 1 (0.00000000 1.00000000) *
##     3) Sexmale< 0.5 249  69 1 (0.27710843 0.72289157)
##       6) Pclass>=2.5 117  56 0 (0.52136752 0.47863248)
##        12) Fare>=23.0875 20   1 0 (0.95000000 0.05000000) *
##        13) Fare< 23.0875 97  42 1 (0.43298969 0.56701031)
##          26) LifeboatC< 0.5 83  41 0 (0.50602410 0.49397590)
##            52) Lifeboat13< 0.5 71  29 0 (0.59154930 0.40845070)
##             104) Lifeboat16< 0.5 62  20 0 (0.67741935 0.32258065)
##               208) Lifeboat15< 0.5 55  13 0 (0.76363636 0.23636364) *
##               209) Lifeboat15>=0.5 7   0 1 (0.00000000 1.00000000) *
##             105) Lifeboat16>=0.5 9   0 1 (0.00000000 1.00000000) *
##            53) Lifeboat13>=0.5 12   0 1 (0.00000000 1.00000000) *
##          27) LifeboatC>=0.5 14   0 1 (0.00000000 1.00000000) *
##       7) Pclass< 2.5 132   8 1 (0.06060606 0.93939394) *
```

```
plot(model_rpart_lb$finalModel, margin=0.1)
text(model_rpart_lb$finalModel, cex = 0.75)
```



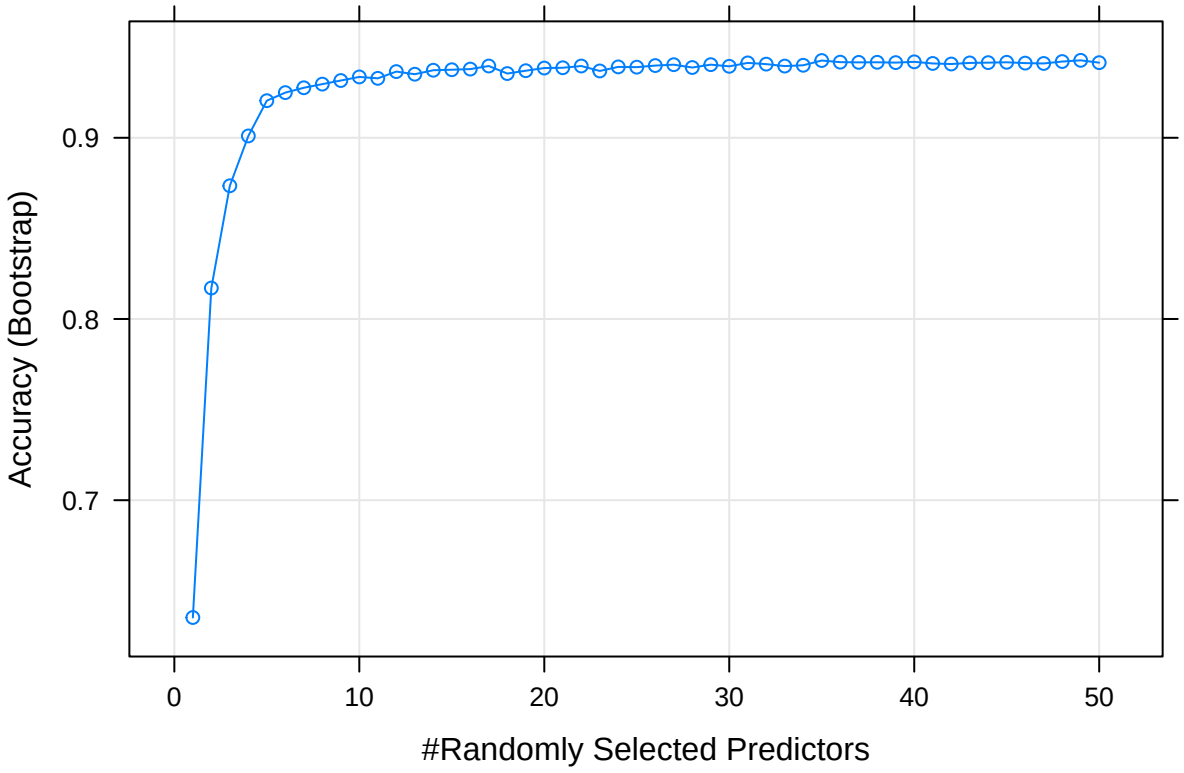
3.10 Random forest model including Lifeboat data

The random forest model get over 97% accuracy with the extra Lifeboat data, which is close to the survival rate of passengers managed to get on a lifeboat or with lifeboat data in their record.

```
model_rf_lb <- train(Survived ~ .,
                     data = train_set,
                     method = "rf",
                     tuneGrid = data.frame(mtry = seq(1, 50)),
                     ntree = 100)
#best tuning value
model_rf_lb$bestTune
```

```
##    mtry
## 49   49
```

```
#plot model
plot(model_rf_lb)
```




```
#predict testset with trained model
predict_rf_lb <- predict(model_rf_lb, test_set)

#add prediction accuracy to the result, and print
result <- bind_rows(result,tibble(
  Method = "Random forest model including Lifeboat",
  Accuracy = mean(predict_rf_lb == test_set$Survived)))
result
```

```
## # A tibble: 11 x 2
##    Method                                        Accuracy
##    <chr>                                            <dbl>
##  1 Project Target                                0.86
##  2 LDA model with Sex,Pclass,Fare,Age            0.8603352
##  3 QDA model with Sex,Pclass,Fare,Age,FamilySize 0.8379888
##  4 Logistic regression of glm                    0.8212291
##  5 kNN model                                     0.7094972
##  6 cross-validated kNN model                     0.6871508
##  7 Classification tree model                     0.8715084
##  8 Random forest model                           0.8659218
##  9 Ensemble of different models                  0.8659218
## 10 Classification tree model including Lifeboat  0.9217877
## 11 Random forest model including Lifeboat        0.9776536
```

estimate variable importance

```r
#estimate variable importance
varImp(model_rf_lb)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 42)
##
##            Overall
## Sexmale    100.000
## Pclass      29.608
## Lifeboat15  29.567
## Age         28.747
## LifeboatC   17.457
## Lifeboat3   16.467
## Fare        15.433
## Lifeboat5   15.329
## Lifeboat13  13.955
## LifeboatB   12.048
## Lifeboat9   10.776
## LifeboatD   10.489
## Lifeboat16   9.187
## Lifeboat7    8.306
## FamilySize   5.899
## SibSp        5.194
## LifeboatA    4.906
## LifeboatQ    4.761
## Lifeboat14   4.728
## Lifeboat11   3.977
```

4 Conclusion

In this project we test different models to predict the survival of the Titanic passengers using the full Titanic extended dataset. The Classification tree, Random forest and Ensemble all achieve over 86% accuracy. Adding the Lifeboat data with Classification tree and Random forest models reaches over 92% and 97% accuracy respectively, which confirms our analysis that the Lifeboat data is the major deciding factor and can significantly increase the prediction accuracy.

Although the project target is reached, there is limitation, the result still has much space to improve when the Lifeboat data is not included. We haven't experimented hyperparameter tuning for different models or considered other measurements such as Sensitivity, Specificity, Precision, Recall, or F-Score.

Future work would involve further develop and extend the training and prediction with other modeling methods such as Neural Network with LDA, QDA models, hyperparameter tuning, and other measurement methods.