

# Who Would Survive the Titanic Disaster

*Lanqin Zhao*

## Who Would Survive the Titanic Disaster?

This project is from <https://www.kaggle.com/c/titanic>

"The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

In this challenge, we ask you to complete the analysis of what sorts of people were likely to survive. In particular, we ask you to apply the tools of machine learning to predict which passengers survived the tragedy."

## Step 1 - Collecting data —

The data is from <https://www.kaggle.com/c/titanic/data>

The data has been split into two groups: training set (train.csv) test set (test.csv) Training set includes 891 examples, 12 variables: a label variable, survival indicating whether or not survival, and 11 features. Test set includes 418 examples, 11 features. The 12 variables are described as following:

Variable Definition Key survival Survival 0 = No, 1 = Yes pclass Ticket class 1 = 1st, 2 = 2nd, 3 = 3rd sex Sex Age Age in years

sibsp # of siblings / spouses aboard the Titanic

parch # of parents / children aboard the Titanic

ticket Ticket number

fare Passenger fare

cabin Cabin number

embarked Port of Embarkation C = Cherbourg, Q = Queenstown, S = Southampton Variable Notes

pclass: A proxy for socio-economic status (SES) 1st = Upper 2nd = Middle 3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way... Sibling = brother, sister, stepbrother, stepsister Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way... Parent = mother, father Child = daughter, son, stepdaughter, stepson Some children travelled only with a nanny, therefore parch=0 for them.

## Step 2: Exploring and preparing the data—

Explore the data to understand data, to clean data, to create new features, to obtain insights, to find predictive features, and prepare the data for modeling.

## Import data

Import the data into R

```
# load R packages
library(plyr) # data manipulation
library(dplyr) # data manipulation

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize

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

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

library(ggplot2) # data visualization
library(scales) # data visualization
library(gmodels) # crosstable
library(stringr) # String manipulation
library(caret) # tune parameters

## Loading required package: lattice

library(rpart) # Decision tree utils
library(randomForest) # Random Forest

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##   margin

## The following object is masked from 'package:dplyr':
##
##   combine

library(kernlab) # SVM

##
## Attaching package: 'kernlab'

## The following object is masked from 'package:scales':
##
##   alpha

## The following object is masked from 'package:ggplot2':
##
```

```

##      alpha
library(party) # Conditional inference trees

## Loading required package: grid
## Loading required package: mvtnorm
## Loading required package: modeltools
## Loading required package: stats4
##
## Attaching package: 'modeltools'
## The following object is masked from 'package:kernlab':
##
##      prior
## The following object is masked from 'package:plyr':
##
##      empty
## Loading required package: strucchange
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric
## Loading required package: sandwich
##
## Attaching package: 'strucchange'
## The following object is masked from 'package:stringr':
##
##      boundary
library(gbm) # gbm

## Loading required package: survival
##
## Attaching package: 'survival'
## The following object is masked from 'package:caret':
##
##      cluster
## Loading required package: splines
## Loading required package: parallel
## Loaded gbm 2.1.3
library(MASS) # glm

##
## Attaching package: 'MASS'

```

```
## The following object is masked from 'package:dplyr':
##
##      select

library(fastAdaboost) # AdaBoost
library(xgboost) # xgboost

##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##      slice

#Import the CSV file.
train <- read.csv("train.csv", header = TRUE, stringsAsFactors =FALSE)
test <- read.csv("test.csv", header = TRUE, stringsAsFactors =FALSE)
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       : chr   "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr   "male" "female" "female" "female" ...
##  $ 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   : chr   "S" "C" "S" "S" ...

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       : chr   "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis"
##  $ Sex        : chr   "male" "female" "male" "male" ...
##  $ 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     : chr   "330911" "363272" "240276" "315154" ...
##  $ Fare       : num   7.83 7  9.69 8.66 12.29 ...
##  $ Cabin      : chr   "" "" "" "" ...
##  $ Embarked   : chr   "Q" "S" "Q" "S" ...
```

## Combine data sets and convert data type

```
# Add a "Survived" variable to the test set to allow for combining data sets
test$Survived <- NA

# Combine data sets
data <- rbind(train, test)
```

```
# Convert data type to factor
data$Survived <- as.factor(data$Survived)
data$Pclass <- as.factor(data$Pclass)
data$Sex <- as.factor(data$Sex)
data$Embarked <- as.factor(data$Embarked)
```

## Data understanding

Age has a lot of missing values. Fare and Embarked have a few missing values. There seem no outliers for all features.

```
summary(data)
```

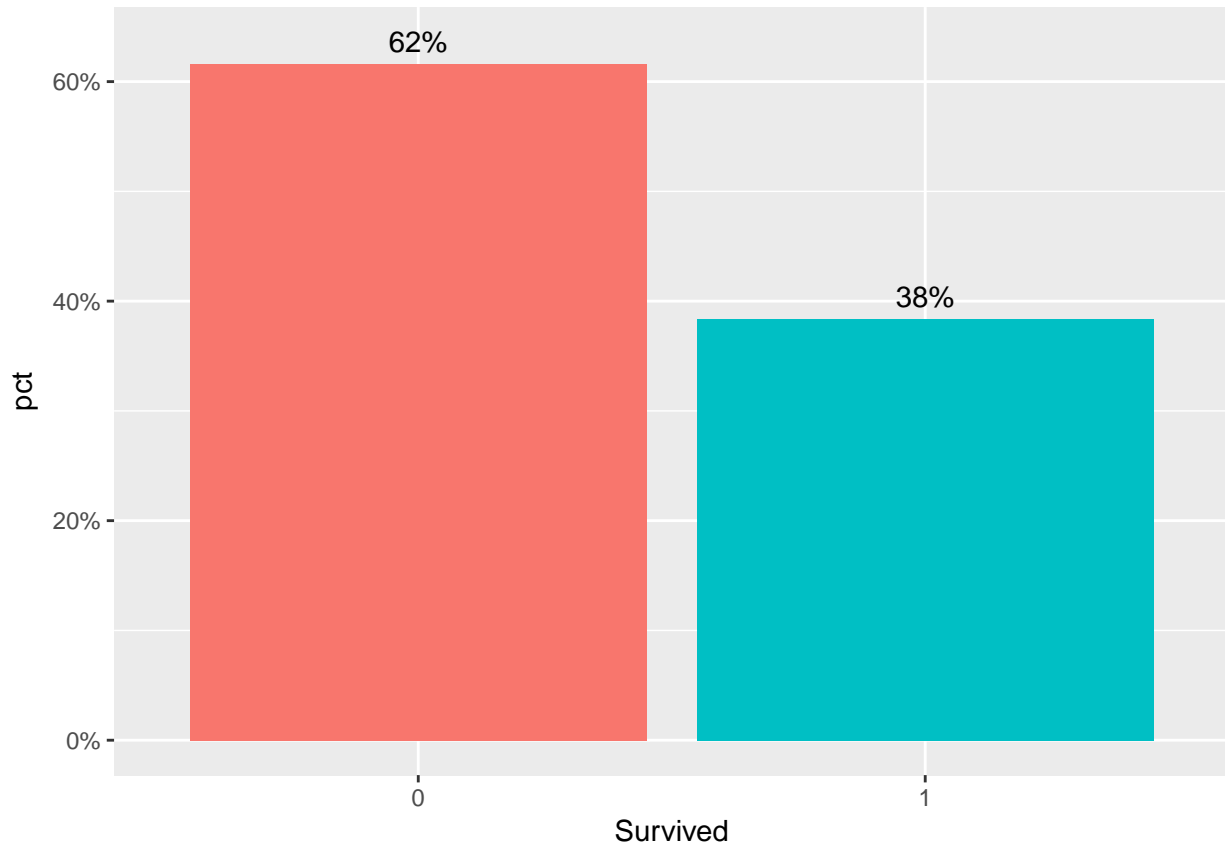
```
## PassengerId  Survived  Pclass      Name                Sex
## Min.   :    1      0   :549    1:323  Length:1309      female:466
## 1st Qu.:  328      1   :342    2:277  Class :character  male   :843
## Median :  655    NA's:418    3:709  Mode  :character
## Mean   :  655
## 3rd Qu.:  982
## Max.   :1309
##
##      Age              SibSp              Parch              Ticket
## Min.   : 0.17      Min.   :0.0000      Min.   :0.000      Length:1309
## 1st Qu.:21.00      1st Qu.:0.0000      1st Qu.:0.000      Class :character
## Median :28.00      Median :0.0000      Median :0.000      Mode  :character
## Mean   :29.88      Mean   :0.4989      Mean   :0.385
## 3rd Qu.:39.00      3rd Qu.:1.0000      3rd Qu.:0.000
## Max.   :80.00      Max.   :8.0000      Max.   :9.000
## NA's    :263
##      Fare              Cabin              Embarked
## Min.   : 0.000      Length:1309      : 2
## 1st Qu.: 7.896      Class :character  C:270
## Median :14.454      Mode  :character  Q:123
## Mean   :33.295                                S:914
## 3rd Qu.:31.275
## Max.   :512.329
## NA's    :1
```

## Data exploration, data cleaning, data manipulation, and feature engineering

Survived: the survival rate was 38.4%.

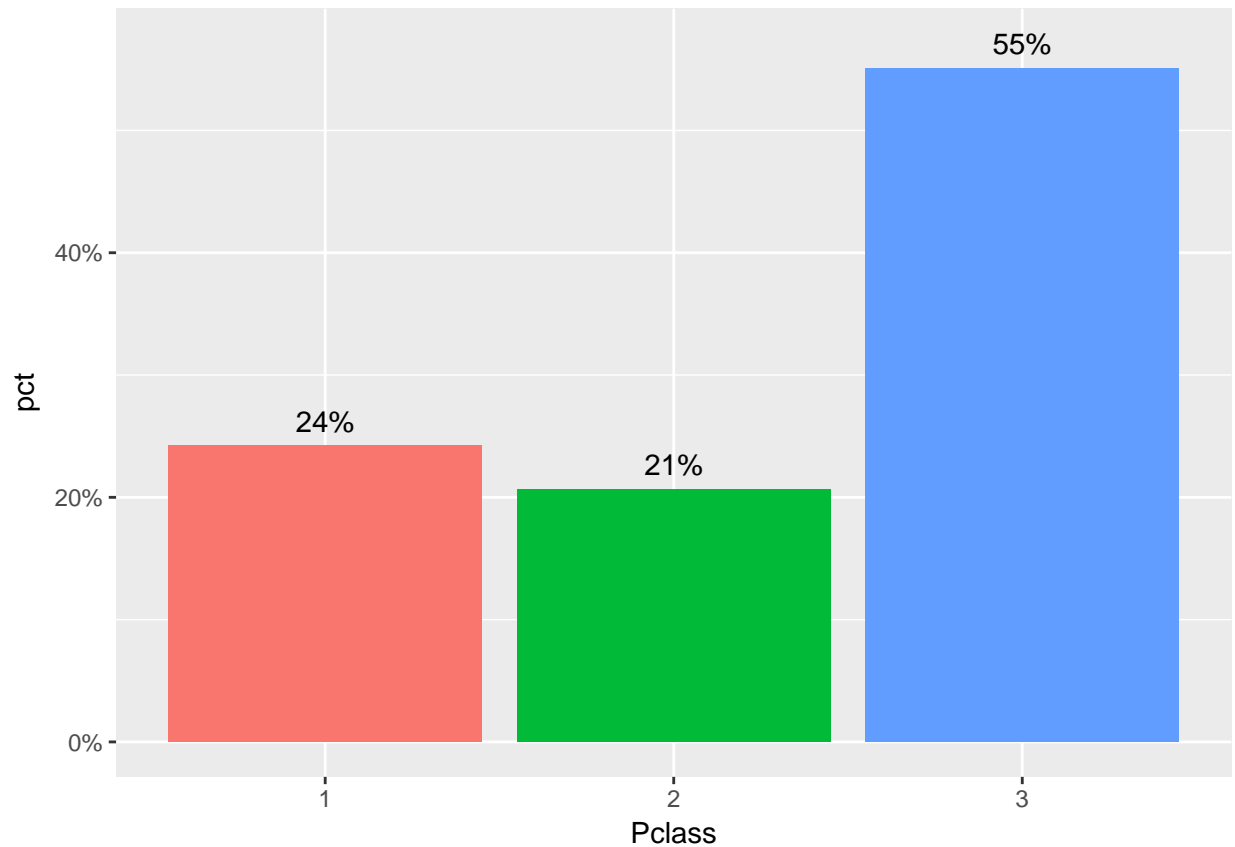
```
# Survival rate
data[1:891,] %>%
  group_by(Survived) %>%
  summarise(count=n()) %>%
  mutate(pct=count/sum(count)) %>%
  ggplot(aes(x=Survived, y=pct, fill=Survived)) +
  geom_bar(stat="identity") +
  scale_y_continuous(labels=percent) +
```

```
geom_text(aes(label=paste0(round(pct*100,0),"%"), y=pct+0.02), size=4, colour= "black")+
theme(legend.position = "NULL")
```

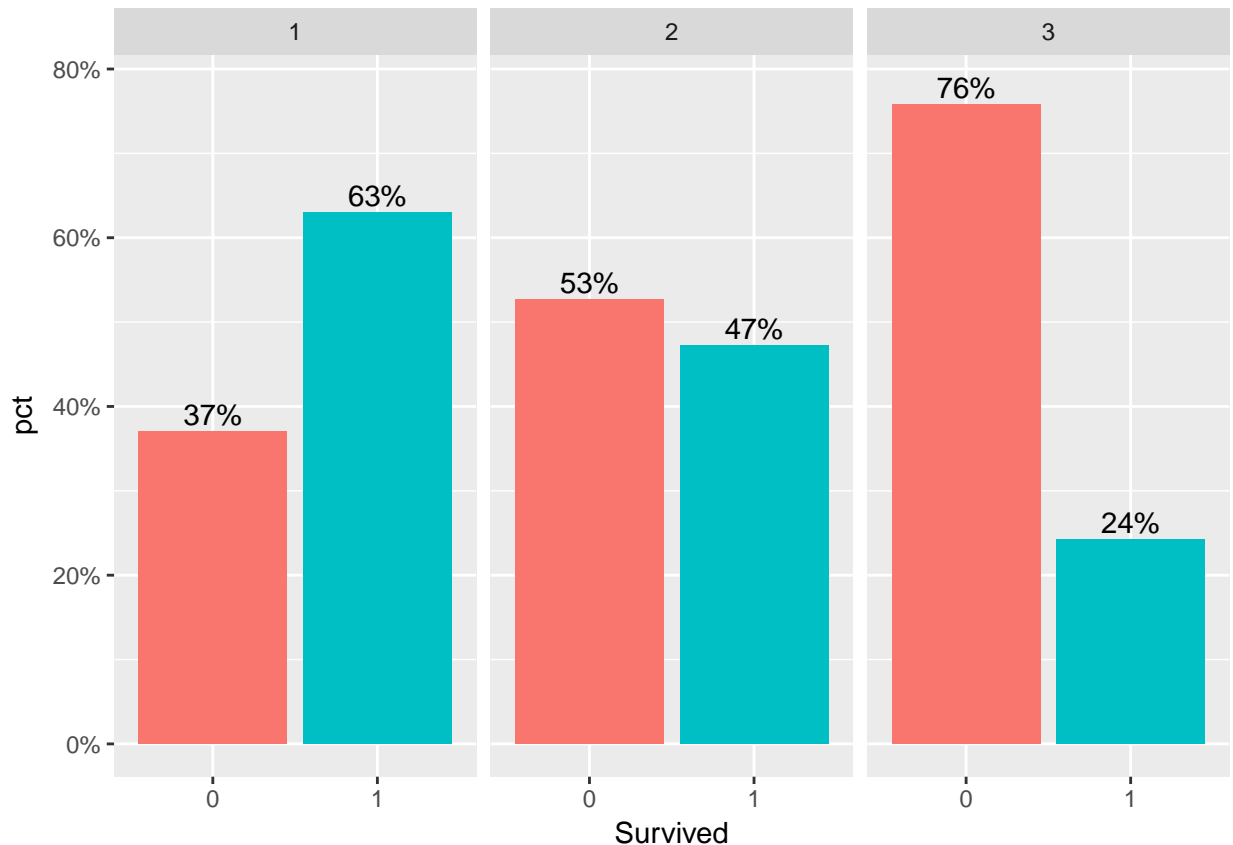


Pclass: There are much more passengers in first class. 24% of passengers were in first class, 21% in second class, 55% in third class. Rich people survived at a higher rate. The survival rate is 63%, 47%, and 24% for first, second, and third class respectively.

```
# Pclass
data[1:891,] %>%
  group_by(Pclass) %>%
  summarise(count=n()) %>%
  mutate(pct=count/sum(count)) %>%
  ggplot(aes(x=Pclass, y=pct, fill=Pclass)) +
  geom_bar(stat="identity") +
  scale_y_continuous(labels=percent) +
  geom_text(aes(label=paste0(round(pct*100,0),"%"), y=pct+0.02), size=4, colour= "black")+
  theme(legend.position = "none")
```



```
#Pclass VS Survival Rate
data[1:891,] %>% group_by(Pclass, Survived) %>%
  summarise(count=n()) %>%
  mutate(pct=count/sum(count)) %>%
ggplot(aes(x=Survived, y=pct, fill=Survived)) +
  geom_bar(stat="identity") +
  facet_grid(. ~ Pclass) +
  scale_y_continuous(labels=percent) +
  geom_text(aes(label=paste0(round(pct*100,0),"%"), y=pct+0.02), size=4, colour= "black")+
  theme(legend.position = "none")
```



Name: contains formal titles, which can be extracted as a potentially useful feature. Title, new variable derived by Name. Based on the plots, 60% of passengers were Mr; “Women and children first” is true in Titanic disaster. Women and children had more than 3 times that men had to survive; It’s obvious that Title and Pclass play important roles in predicting who would survive. Passengers having title of Master, Miss, and Mrs had more than 90% chance to survive in first and second class, but those in third class had less than 50% chance. Mr even had less than 40% chance to survive in first class and about 10% in second and third class.

```
#Name: new variable Title derived by Name is predictive
# Look at the first few names
data$Name[1:20]
```

```
## [1] "Braund, Mr. Owen Harris"
## [2] "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## [3] "Heikkinen, Miss. Laina"
## [4] "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
## [5] "Allen, Mr. William Henry"
## [6] "Moran, Mr. James"
## [7] "McCarthy, Mr. Timothy J"
## [8] "Palsson, Master. Gosta Leonard"
## [9] "Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)"
## [10] "Nasser, Mrs. Nicholas (Adele Achem)"
## [11] "Sandstrom, Miss. Marguerite Rut"
## [12] "Bonnell, Miss. Elizabeth"
## [13] "Saunderscock, Mr. William Henry"
## [14] "Andersson, Mr. Anders Johan"
## [15] "Vestrom, Miss. Hulda Amanda Adolfina"
## [16] "Hewlett, Mrs. (Mary D Kingcome) "
```



```
## [17] "Rice, Master. Eugene"
## [18] "Williams, Mr. Charles Eugene"
## [19] "Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)"
## [20] "Masselmani, Mrs. Fatima"
```

```
#extract title
data$Title = sapply(data$Name, FUN=function(x) { strsplit(x, split="[,.]")[[1]][2]})
data$Title = sub(' ', '', data$Title)
table(data$Title)
```

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

```
# combine special, rare titles
data$Title[data$Title %in% c('Capt', 'Col', 'Don', 'Major', 'Sir', 'Dr', 'Rev')] <- 'Mr'
data$Title[data$Title %in% c('Mme', 'Mlle', 'Ms', 'Dona', 'Lady', 'the Countess', 'Jonkheer')] <- 'Miss'
table(data$Title)
```

```
##
## Master  Miss   Mr   Mrs
##      61   269  782  197
```

```
# convert to factor
data$Title = factor(data$Title)
```

```
# explore Title
# Title=Master, boys with age of 0.33-14.5, median= 4.0
table(data$Sex[data$Title=="Master"]) # they are male
```

```
##
## female  male
##      0    61
```

```
summary(data$Age[data$Title=="Master"]) # 0.33-14.5, median= 4.0
```

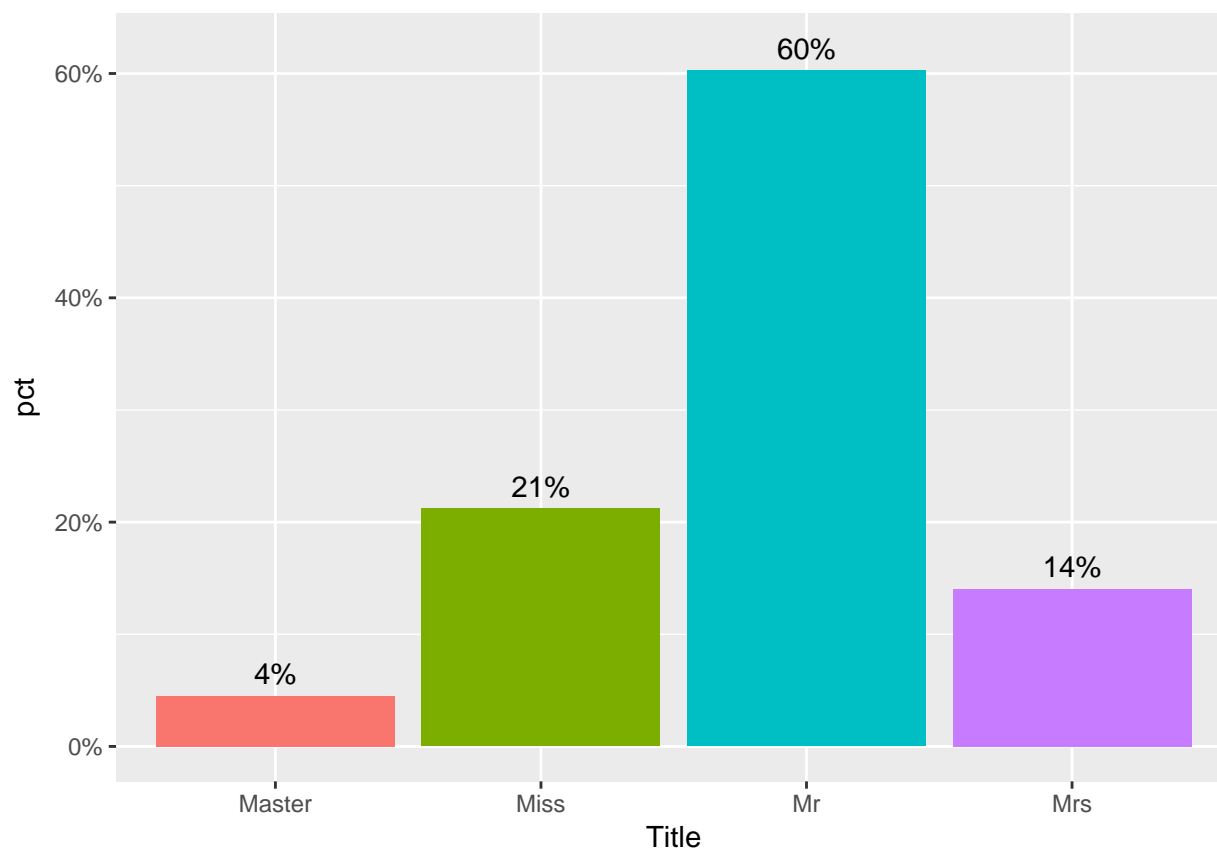
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      0.330  2.000  4.000   5.483  9.000  14.500      8
```

```
# Title=Miss, age of 0.17-63.0
summary(data$Age[data$Title=="Miss"]) # 0.17-63.0, median= 22.00, mean=22.16
```

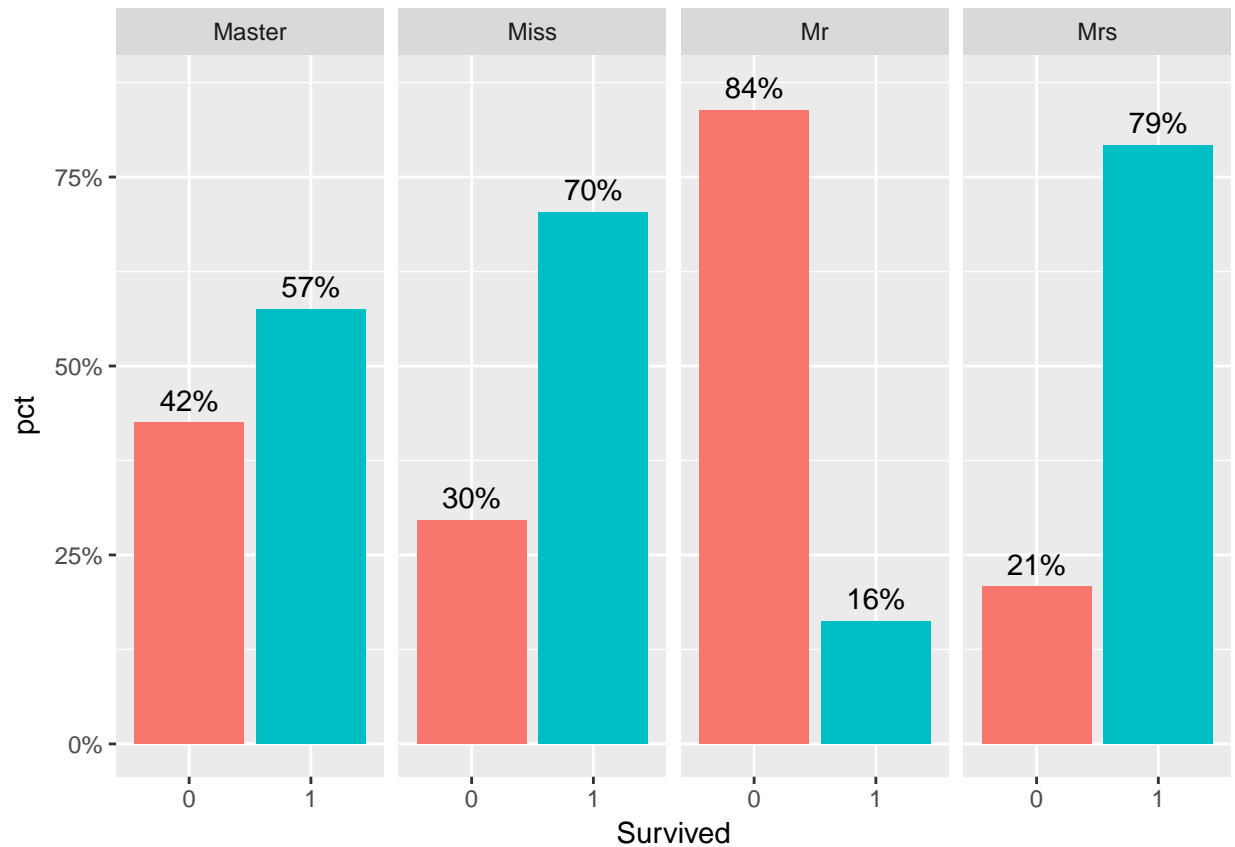
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      0.17  16.00  22.00  22.16  30.00  63.00     51
```

```
# Title
data[1:891,] %>% group_by(Title) %>%
  summarise(count=n()) %>%
  mutate(pct=count/sum(count)) %>%
ggplot(aes(x=Title, y=pct, fill=Title)) +
  geom_bar(stat="identity") +
  scale_y_continuous(labels=percent) +
  geom_text(aes(label=paste0(round(pct*100,0),"%"), y=pct+0.02), size=4, colour= "black")+
  
```

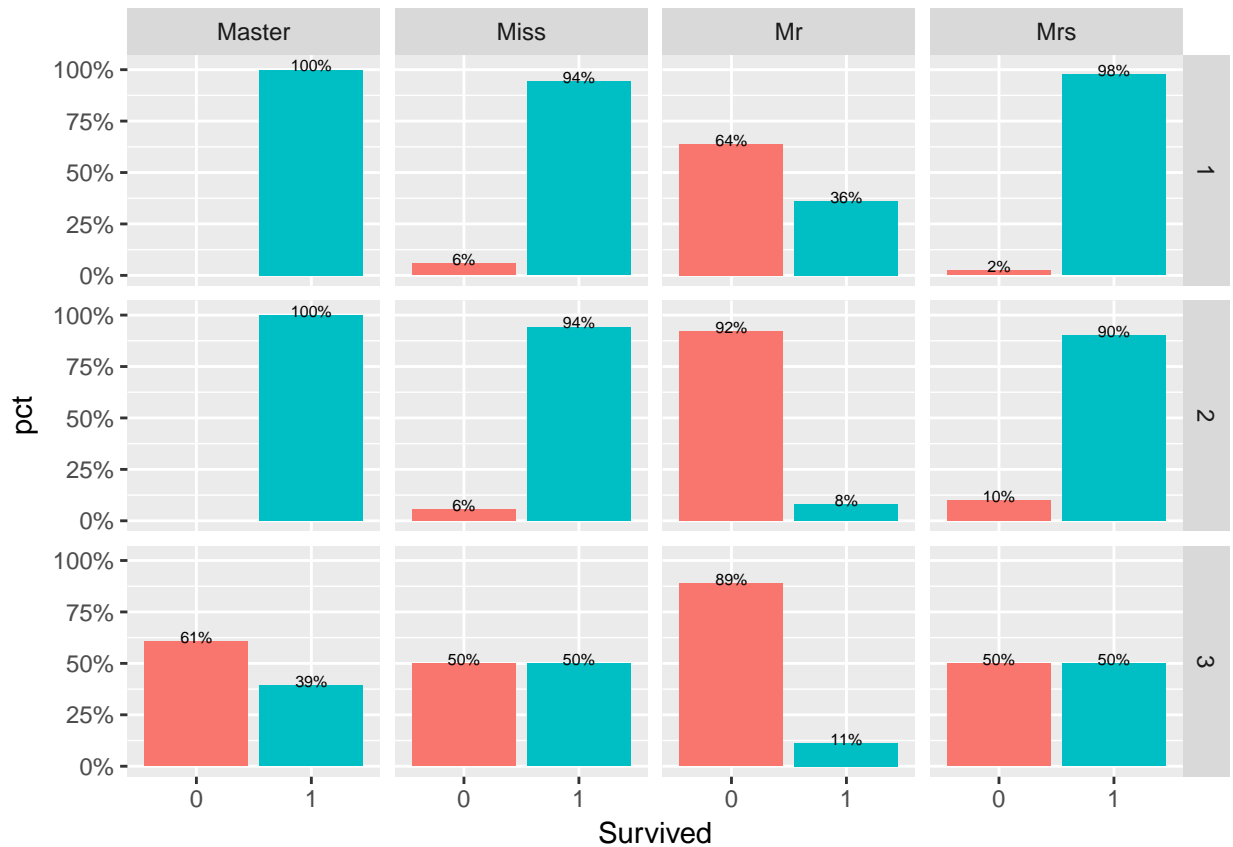
```
theme(legend.position = "none")
```



```
# Title vs Survival
data[1:891,] %>% group_by(Title, Survived) %>%
  summarise(count=n()) %>%
  mutate(pct=count/sum(count)) %>%
ggplot(aes(x=Survived, y=pct, fill=Survived)) +
  geom_bar(stat="identity") +
  facet_grid(. ~ Title) +
  scale_y_continuous(labels=percent) +
  geom_text(aes(label=paste0(round(pct*100,0),"%"), y=pct+0.03), size=4, colour= "black")+
  theme(legend.position = "none")
```

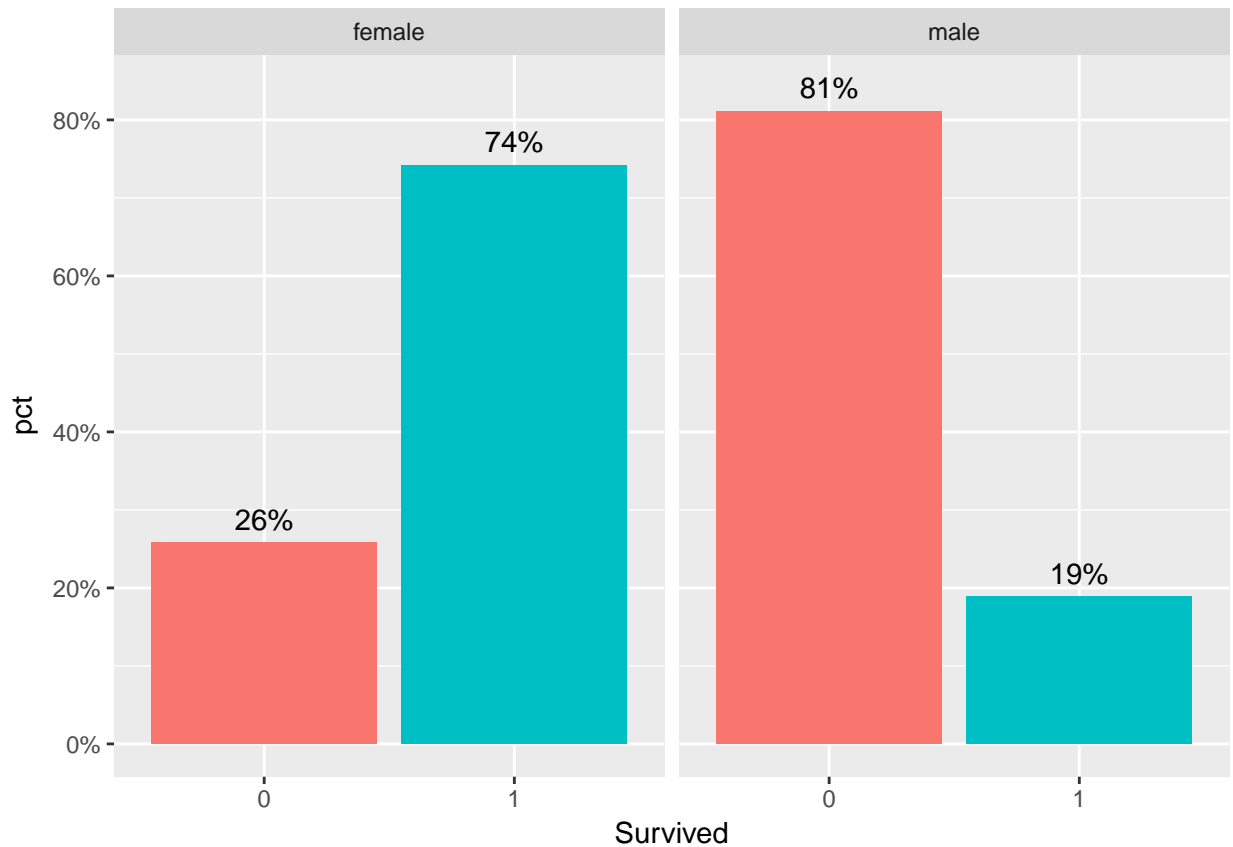


```
# Title Vs vs survial under class
data[1:891,] %>% group_by(Pclass, Title, Survived) %>%
  summarise(count=n()) %>%
  mutate(pct=count/sum(count)) %>%
ggplot(aes(x=Survived, y=pct, fill=Survived)) +
  geom_bar(stat="identity") +
  facet_grid(Pclass ~ Title) +
  scale_y_continuous(labels=percent) +
  geom_text(aes(label=paste0(round(pct*100,0),"%"), y=pct+0.02), size=2, colour= "black")+
  theme(legend.position = "none")
```



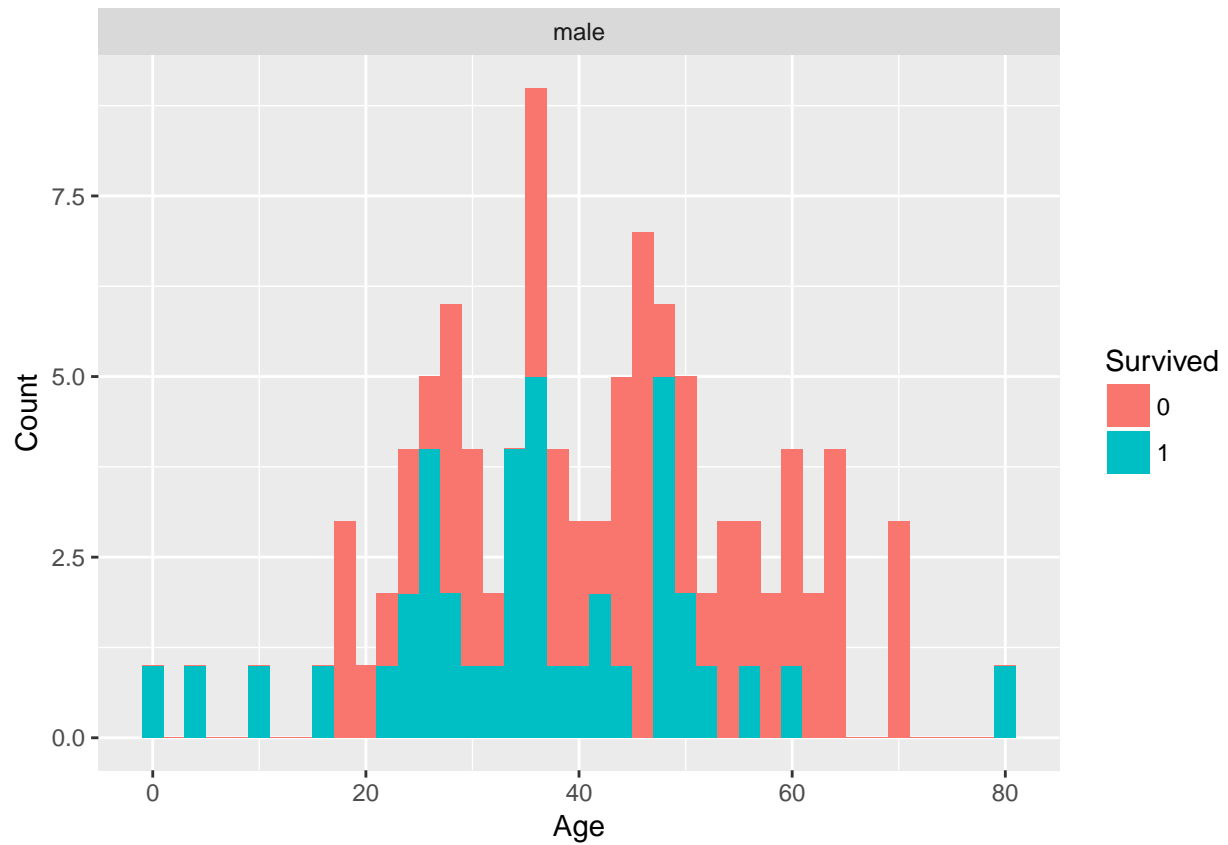
Sex: It is clearly obvious that female have much more chance, 74%, to survive than male, 19%.

```
#Sex
data[1:891,] %>% group_by(Sex, Survived) %>%
  summarise(count=n()) %>%
  mutate(pct=count/sum(count)) %>%
  ggplot(aes(x=Survived, y=pct, fill=Survived)) +
  geom_bar(stat="identity") +
  facet_grid(. ~ Sex) +
  scale_y_continuous(labels=percent) +
  geom_text(aes(label=paste0(round(pct*100,0),"%"), y=pct+0.03), size=4, colour= "black")+
  theme(legend.position = "none")
```

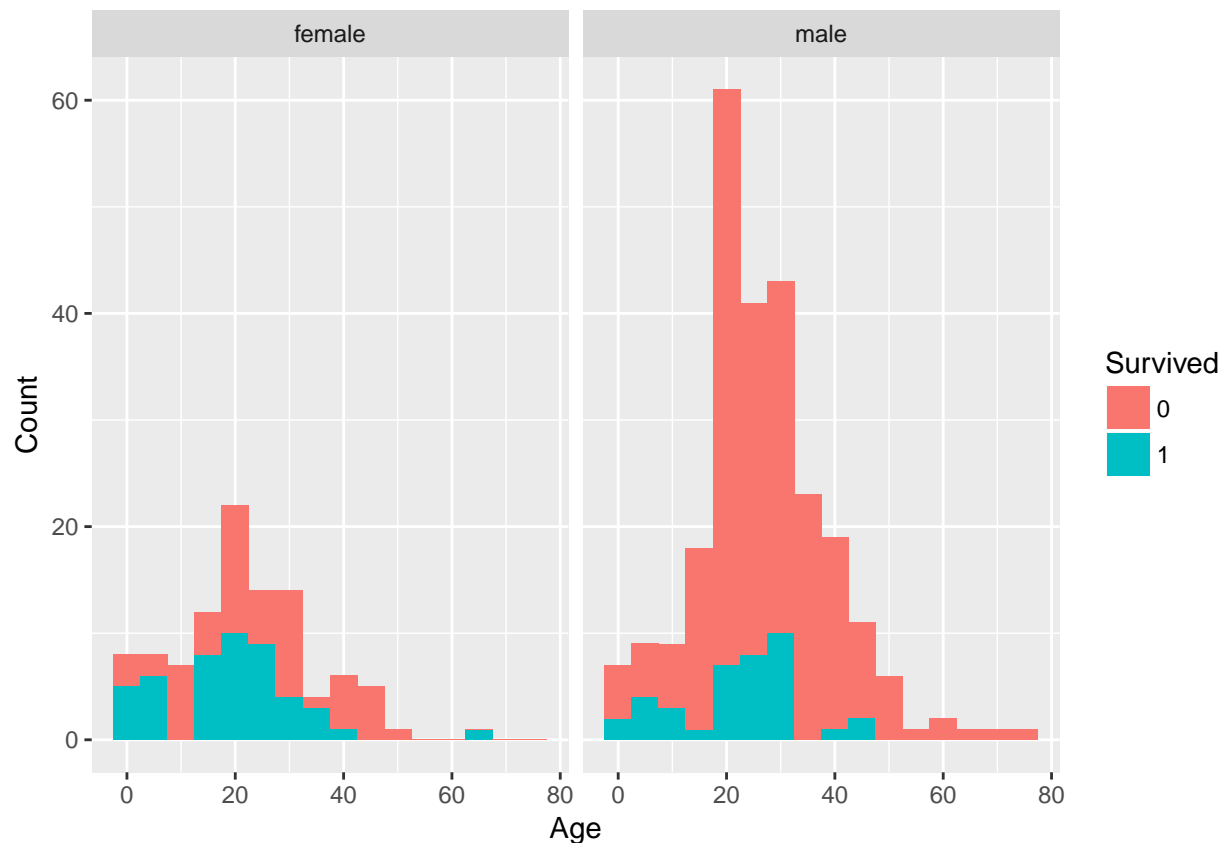


Age Based on the plot of Title above, Mr in first class and Master, Miss, and Mrs in third class are difficult to predict if they would survive, so I focus on these passengers. From the plot, Age is associated with survival rate. Then it is preferable to keep the age feature and to impute the missing values.

```
#Age vs Survival under Sex
ggplot( subset(data[1:891,],!is.na(Age) & Pclass=="1" & Sex=="male") , aes(x = Age, fill=Survived)) +
  geom_histogram( binwidth=2) +
  facet_wrap(~ Sex) +
  labs( x = "Age", y = "Count")
```



```
ggplot( subset(data[1:891,],!is.na(Age) & Pclass=="3") , aes(x = Age, fill=Survived)) +
  geom_histogram( binwidth=5) +
  facet_wrap(~ Sex) +
  labs( x = "Age", y = "Count")
```



how to impute missing values? method: Age1, impute Age by Title

```
#method1: Age1, impute Age by Title
#create Age1
data$Age1=data$Age
#Title=Master
summary(data$Age[data$Title=="Master"])# 0.33-14.5, boys with median= 4.0

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##    0.330   2.000   4.000   5.483   9.000  14.500     8

masterAge <- data$Title == "Master" & is.na(data$Age)
data[masterAge, "Age1"] <- 4.0

# Title=Miss
summary(data$Age[data$Title=="Miss"]) # 0.17-63.0, median= 22.00, mean=22.16

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##    0.17   16.00   22.00   22.16   30.00   63.00    51

missAlone <- data$Title == "Miss" & data$Parch==0 & data$SibSp==0
summary(data[missAlone, "Age"]) # mean=27

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##    5.00   21.00   26.00   27.38   33.00   58.00    34

missAloneAge <- missAlone & is.na(data$Age)
data[missAloneAge, "Age1"] <- 27
```

```
missNot <- data$Title == "Miss" & (data$Parch + data$SibSp >0 )
summary(data[missNot, "Age"]) # mean=15
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      0.17   4.25   15.00   15.27  22.00   63.00     17
```

```
missNotAge <- missNot & is.na(data$Age)
data[missNotAge, "Age1"] <- 15
```

```
#Title=Mrs
summary(data$Age[data$Title == "Mrs"]) # mean=37
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      14.00  27.00  35.50  36.99  46.50   76.00     27
```

```
mrsAge <- data$Title == "Mrs" & is.na(data$Age)
data[mrsAge, "Age1"] <- 37
```

```
#Title=Mr
summary(data$Age[data$Title == "Mr"]) # mean=33
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      11.0   23.0   30.0   32.8   40.0   80.0   177
```

```
mrAge <- data$Title == "Mr" & is.na(data$Age)
data[mrAge, "Age1"] <- 33
```

```
summary(data$Age1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.17  22.00  30.00  30.02  36.00   80.00
```

AgeGroup Based on the above Age plot, different Age group passengers had different chance to survive, so I group Age1. From the plot, the group of less than 7 years had the highest chance to survive, but the group of 28-35 had the least chance. Under pclass and Title, AgeGroup seems predictive.

```
# group Age1 to AgeGroup
```

```
AgeGroup= cut(data$Age1, breaks = c(0,7,14.5 ,21,28,35,50,80), labels = c("0-7", "7-14.5", "14.5-21", "21-28", "28-35", "35-50", "50-80"))
table(AgeGroup)
```

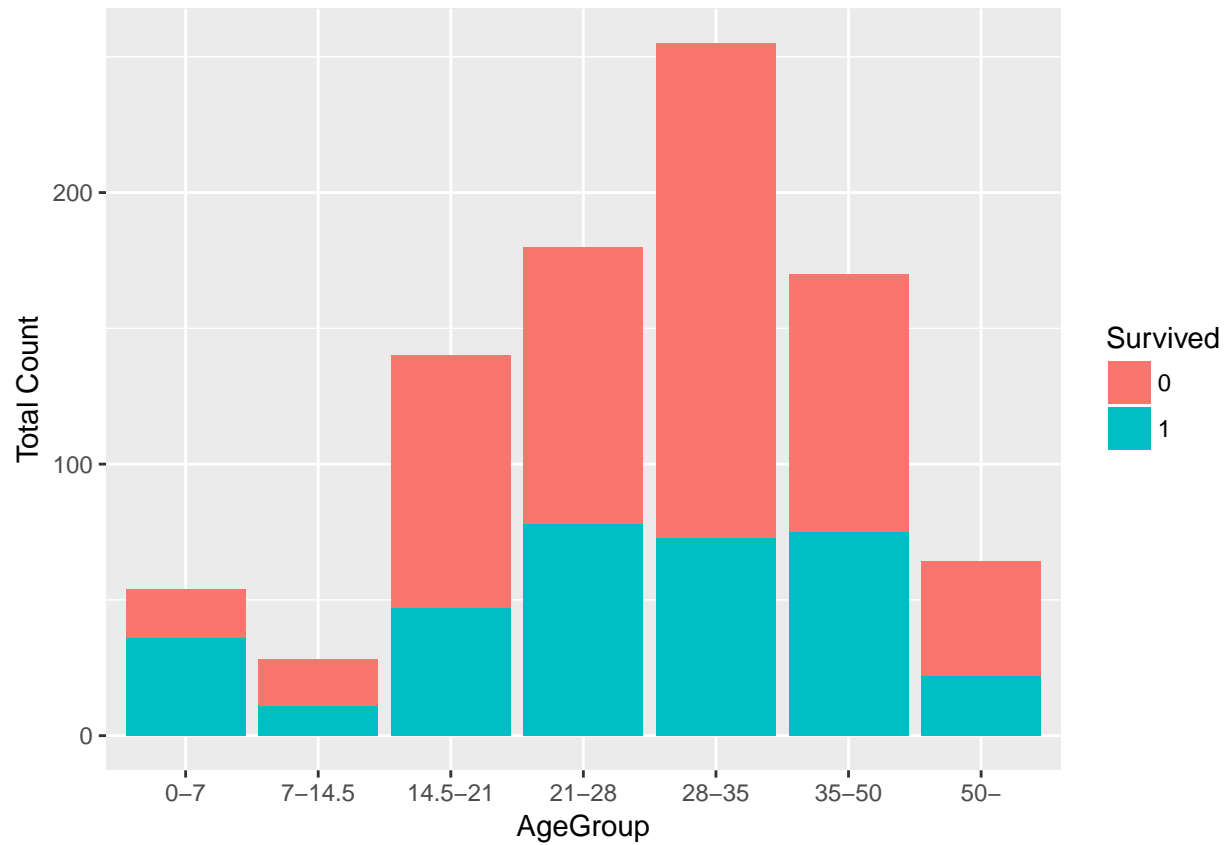
```
## AgeGroup
##      0-7  7-14.5 14.5-21  21-28  28-35  35-50  50-
##      74    43    198    280    365    254    95
```

```
data$AgeGroup=AgeGroup
```

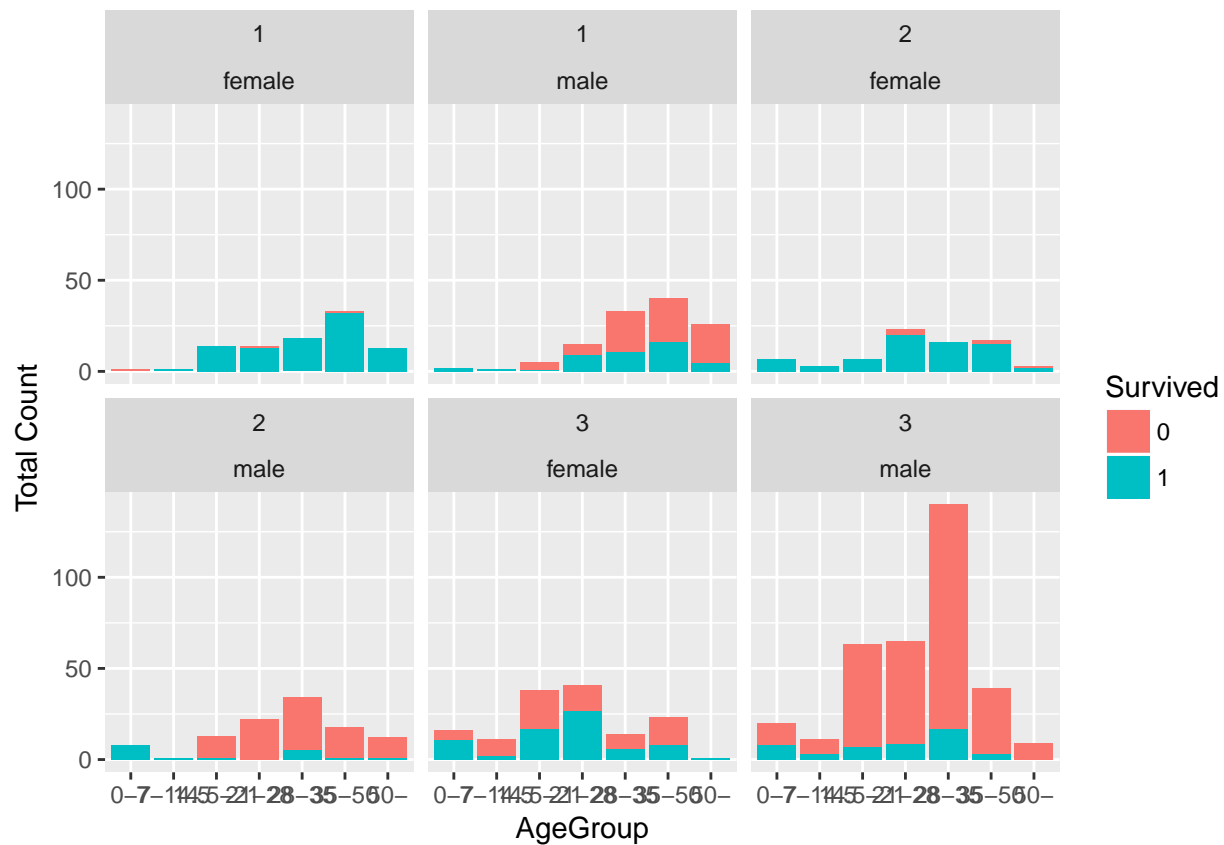
```
#AgeGroup Vs Survival
```

```
ggplot(data[1:891,], aes(x = AgeGroup, fill=Survived)) +
  geom_bar() +
  xlab("AgeGroup") +
  ylab("Total Count") +
  labs(fill = "Survived")
```





```
# AgeGroup Vs Title, under Pclass, Survival
ggplot(data[1:891,], aes(x = AgeGroup, fill=Survived)) +
  geom_bar() +
  facet_wrap(Pclass~Sex)+
  xlab("AgeGroup") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



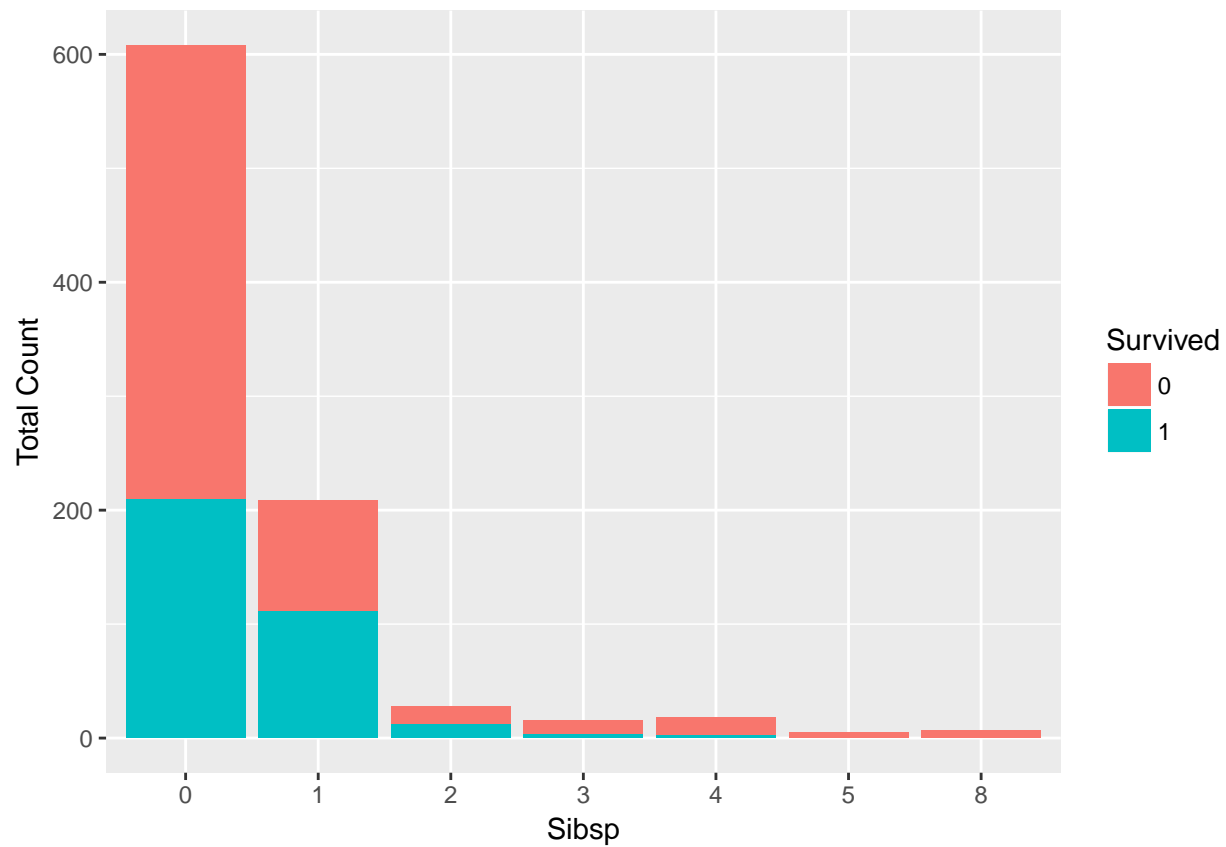
Sibsp: It seems that passengers having more than 2 siblings/spouses had very little chance to survive. Then I group it to SibGroup.

```
#Sibsp
```

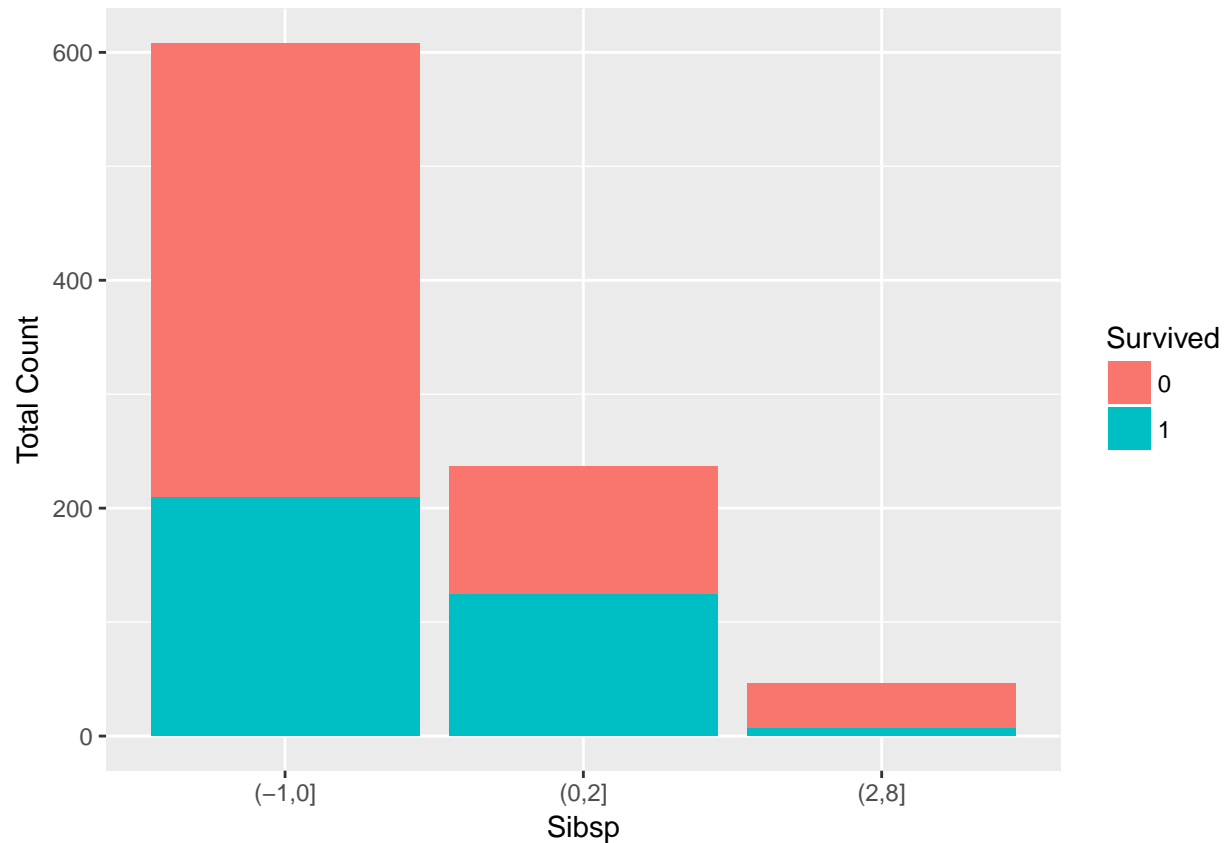
```
summary(data$SibSp)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000   0.0000  0.4989  1.0000   8.0000
```

```
ggplot(data[1:891,], aes(x = as.factor(SibSp), fill = Survived)) +
  geom_bar() +
  xlab("Sibsp") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



```
# SibGroup
data$SibGroup <- cut(data$SibSp, breaks=c(-1,0,2,8),levels=c("0","1-2","3-"))
# SibGroup vs Survival under pclass and title
ggplot(data[1:891,], aes(x = SibGroup, fill = Survived)) +
  geom_bar() +
  xlab("Sibsp") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



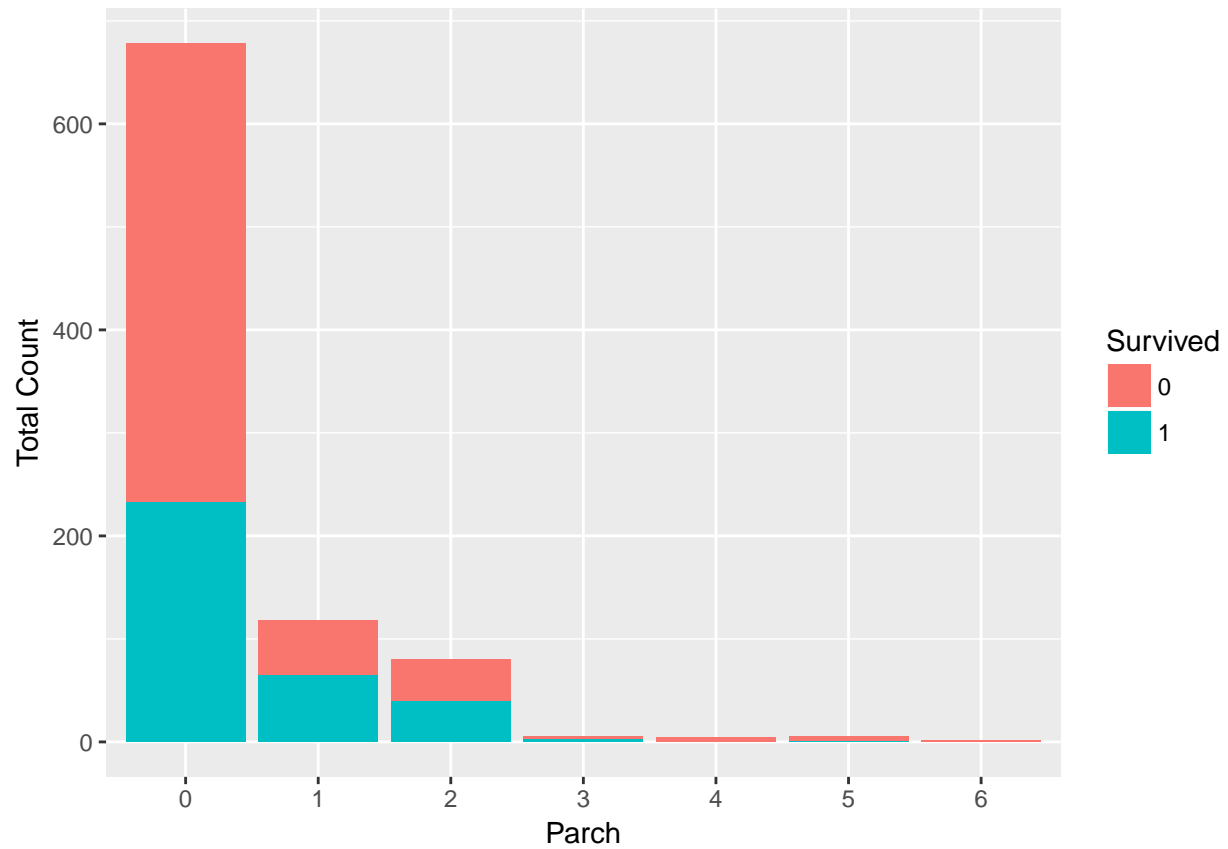
Parch: Similar with SibSp, passengers having 0 or more than 3 parents/children have less chance to survive. Then I group it to ParchGroup. Under Pclass and Title, ParchGroup is little predictive for third class.

*#Parch*

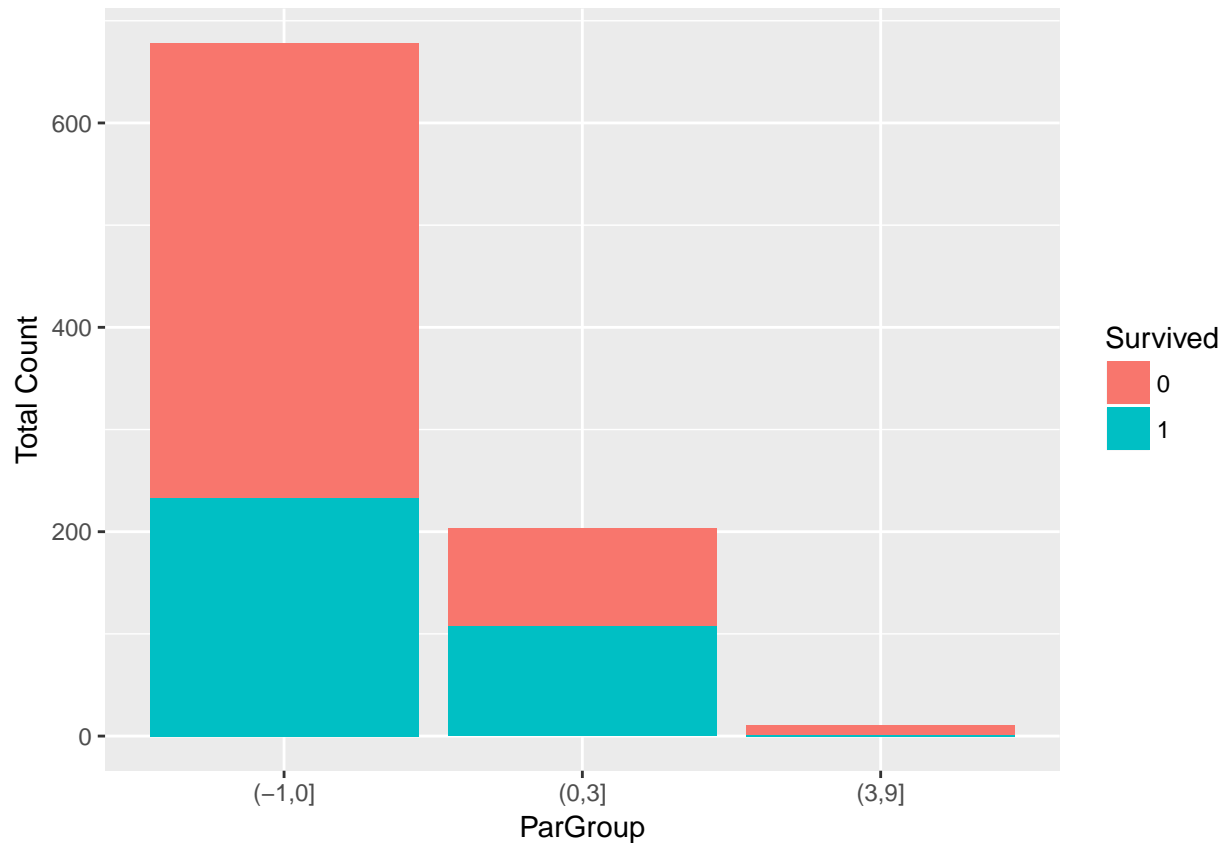
```
summary(data$Parch)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.000  0.000   0.000   0.385  0.000   9.000
```

```
ggplot(data[1:891,], aes(x = as.factor(Parch), fill = Survived)) +
  geom_bar() +
  xlab("Parch") +
  ylab("Total Count") +
  labs(fill = "Survived")
```

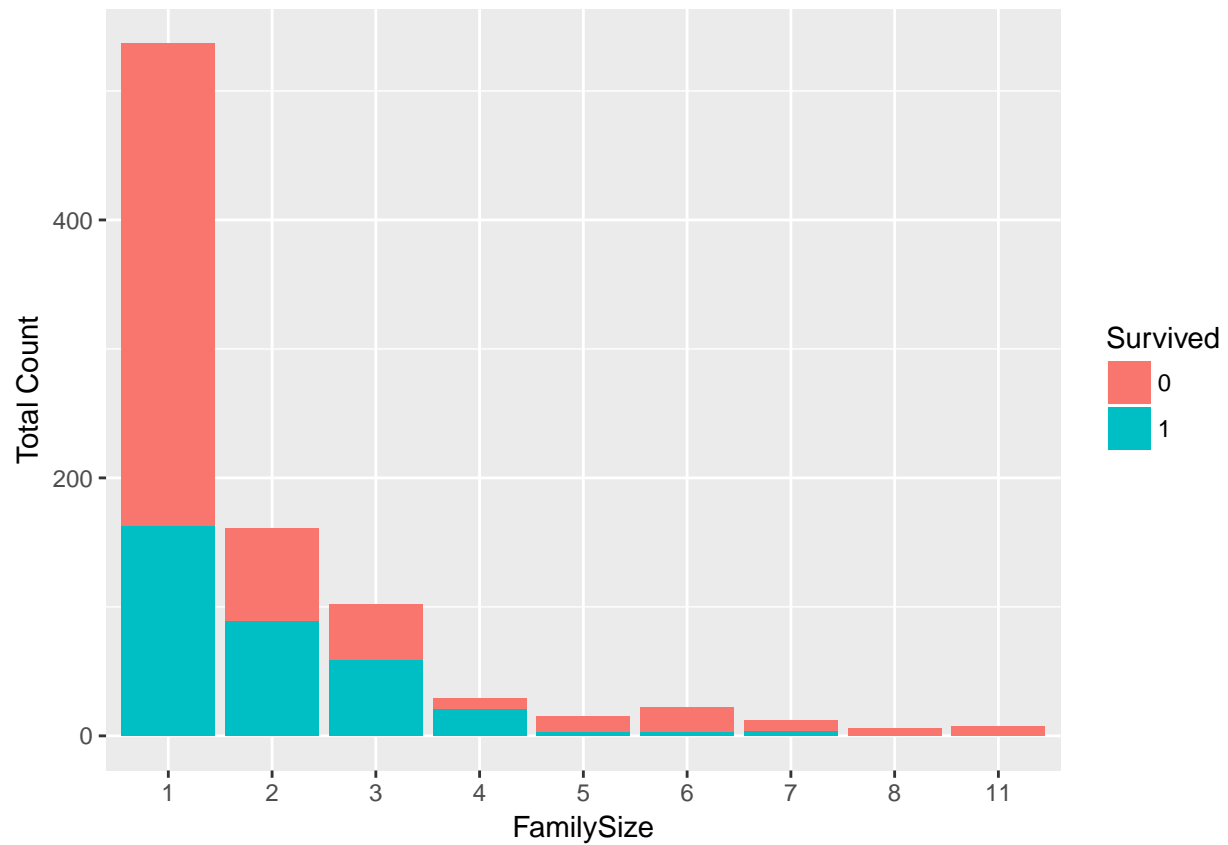


```
# ParGroup
data$ParGroup <- cut(data$Parch, breaks=c(-1,0,3,9), levels=c("0", "1-3", "4-"))
# ParGroup vs Survival under pclass and title
ggplot(data[1:891,], aes(x = ParGroup, fill = Survived)) +
  geom_bar() +
  xlab("ParGroup") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



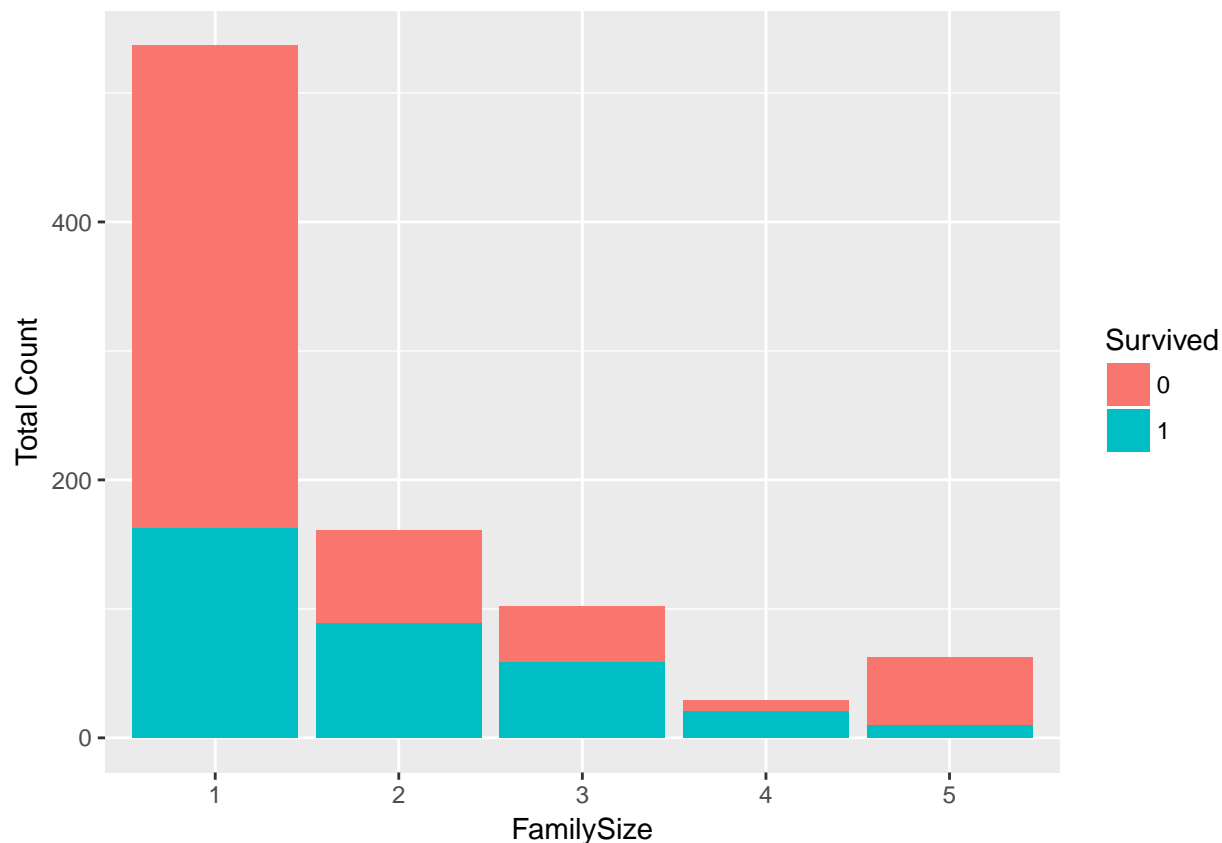
FamilySize FamilySize=SibSp+Parch+1. From the plots, about 60% of passengers were traveling alone, and passengers traveling alone and having a big family size had less chance to survive.

```
# Create FamilySize
data$FamilySize <- with(data,SibSp+Parch+1 )
# FamilySize and survived are associated? Yes
ggplot(data[1:891,], aes(x = as.factor(FamilySize), fill = Survived)) +
  geom_bar() +
  xlab("FamilySize") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



```
# recode FamilySize since there are few examples for FamilySize>4
data$FamilySize[data$FamilySize>4] <- 5

# FamilySize vs survival
ggplot(data[1:891,], aes(x = as.factor(FamilySize), fill = Survived)) +
  geom_bar() +
  xlab("FamilySize") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



Role \* Parents, who needed to take care of their children have more chance to survive? Especially, when there were more than 1 kid, fathers were needed to help have more chance to survive? Refine Title variable to look at. The result is surprising, parents have more chance to die. However, the differences between father and Mr, mother and Mrs are not much, so I will keep Title in modeling.

```
# derive Role variable
Role <- as.character(data$Title)
#Father role
#FamilySize>=4,
# father, 2 more kids
Father2 <- data$Title=="Mr" & data$SibSp==1 & data$FamilySize>3
Role[Father2 & (data$Age>20 | is.na(data$Age))]="father2"
# FamilySize==3,
# Father, one kid
Father1 <- data$Title=="Mr" & data$SibSp==1 & data$FamilySize==3
Role[Father1 & (data$Age>20 | is.na(data$Age))]="father1" # exclude cases of mother with two kids
#father, 2kids
Father2 <- data$Title=="Mr" & data$SibSp==0 & data$FamilySize==3
Role[Father2 & (data$Age>20 | is.na(data$Age))]="father2"
# FamilySize==2
#father, 1 kid
Father1 <- data$Title=="Mr" & data$SibSp==0 & data$FamilySize==2
Role[Father1 & (data$Age>25 | is.na(data$Age))]="father1" # exclude adult son

# Mother role
Mother <- data$Title=="Mrs" & data$Parch>0
Role[Mother]="mother"
```



```

# Role
table(Role)

## Role
## father1 father2 Master Miss mother Mr Mrs
##      33      19      61      269      87      730      110

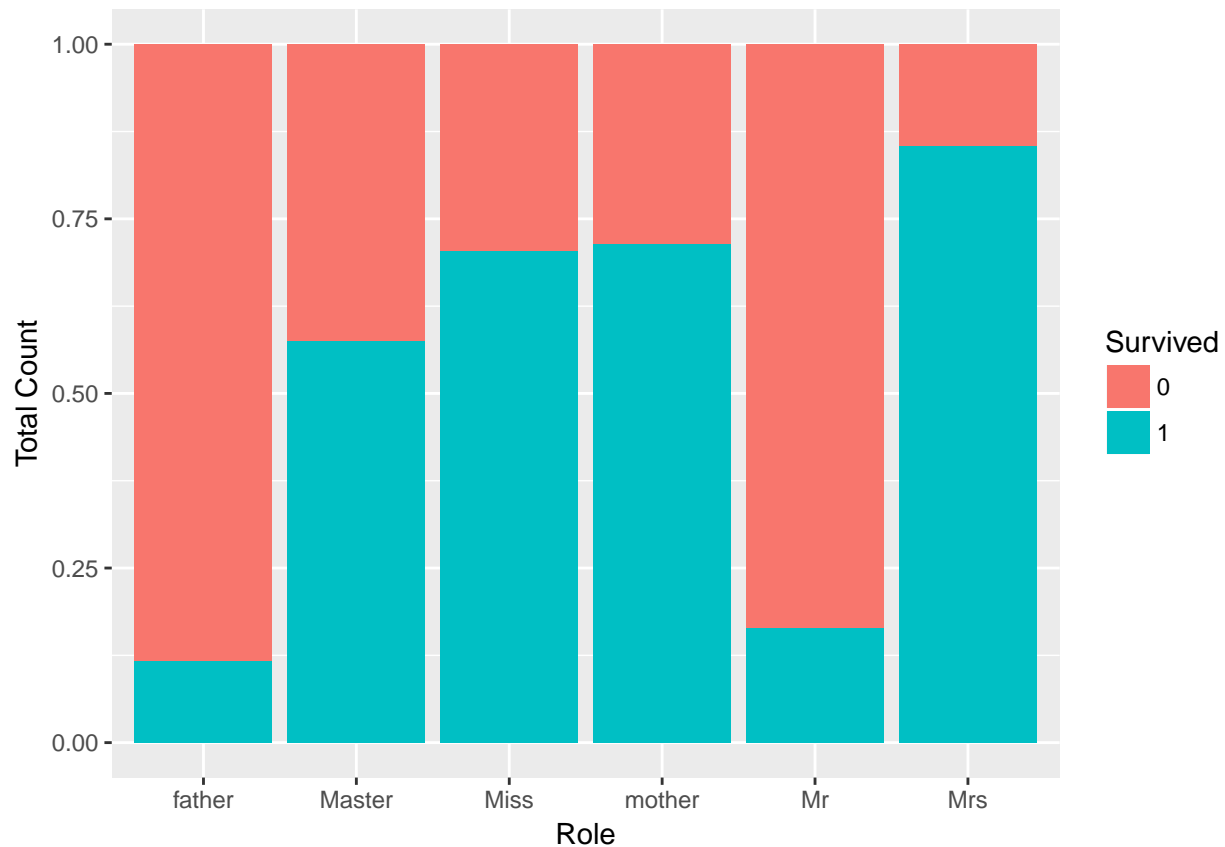
data$Role <- Role

# combine father1 and father2 into father since there are only a few examples.
data$Role[data$Role=="father1" | data$Role=="father2" ]<- "father"

# convert to factor
data$Role<- as.factor(data$Role)

# Role Vs Survival
ggplot(data[1:891,], aes(x = Role, fill = Survived)) +
  geom_bar(position = 'fill') +
  xlab("Role") +
  ylab("Total Count") +
  labs(fill = "Survived")

```



```

# convert type
data$FamilySize <- as.factor(data$FamilySize)

```

Ticket PartySize, the number of a group of people bought a joint ticket, so the fare for each person should be recalculated. PartySize, is like FamilySize, equals 1 or above 4 have high chance to die. Since the observations

are limited when PartySize>4, I will combine them into 5 after I calculate the Fare for each passenger.

```
# derive PartySize, the number of passengers sharing a ticket
arrange(filter(data,FamilySize=="5"),Ticket) # a group share a ticket
```

##	PassengerId	Survived	Pclass
## 1	28	0	1
## 2	89	1	1
## 3	342	1	1
## 4	439	0	1
## 5	945	<NA>	1
## 6	961	<NA>	1
## 7	775	1	2
## 8	438	1	2
## 9	69	1	3
## 10	51	0	3
## 11	165	0	3
## 12	267	0	3
## 13	639	0	3
## 14	687	0	3
## 15	825	0	3
## 16	1286	<NA>	3
## 17	26	1	3
## 18	183	0	3
## 19	234	1	3
## 20	262	1	3
## 21	1046	<NA>	3
## 22	1066	<NA>	3
## 23	1271	<NA>	3
## 24	14	0	3
## 25	120	0	3
## 26	542	0	3
## 27	543	0	3
## 28	611	0	3
## 29	814	0	3
## 30	851	0	3
## 31	64	0	3
## 32	168	0	3
## 33	361	0	3
## 34	635	0	3
## 35	643	0	3
## 36	820	0	3
## 37	1106	<NA>	3
## 38	8	0	3
## 39	25	0	3
## 40	375	0	3
## 41	568	0	3
## 42	1281	<NA>	3
## 43	17	0	3
## 44	172	0	3
## 45	279	0	3
## 46	788	0	3
## 47	886	0	3
## 48	947	<NA>	3
## 49	177	0	3

## 50	230	0	3
## 51	410	0	3
## 52	486	0	3
## 53	1024	<NA>	3
## 54	60	0	3
## 55	72	0	3
## 56	387	0	3
## 57	481	0	3
## 58	679	0	3
## 59	684	0	3
## 60	1031	<NA>	3
## 61	1032	<NA>	3
## 62	160	0	3
## 63	181	0	3
## 64	202	0	3
## 65	325	0	3
## 66	793	0	3
## 67	847	0	3
## 68	864	0	3
## 69	1080	<NA>	3
## 70	1234	<NA>	3
## 71	1252	<NA>	3
## 72	1257	<NA>	3
## 73	312	1	1
## 74	743	1	1
## 75	916	<NA>	1
## 76	956	<NA>	1
## 77	1034	<NA>	1
## 78	87	0	3
## 79	148	0	3
## 80	437	0	3
## 81	737	0	3
## 82	1059	<NA>	3

##	Name	Sex	Age
## 1	Fortune, Mr. Charles Alexander	male	19.0
## 2	Fortune, Miss. Mabel Helen	female	23.0
## 3	Fortune, Miss. Alice Elizabeth	female	24.0
## 4	Fortune, Mr. Mark	male	64.0
## 5	Fortune, Miss. Ethel Flora	female	28.0
## 6	Fortune, Mrs. Mark (Mary McDougald)	female	60.0
## 7	Hocking, Mrs. Elizabeth (Eliza Needs)	female	54.0
## 8	Richards, Mrs. Sidney (Emily Hocking)	female	24.0
## 9	Andersson, Miss. Erna Alexandra	female	17.0
## 10	Panula, Master. Juha Niilo	male	7.0
## 11	Panula, Master. Eino Viljami	male	1.0
## 12	Panula, Mr. Ernesti Arvid	male	16.0
## 13	Panula, Mrs. Juha (Maria Emilia Ojala)	female	41.0
## 14	Panula, Mr. Jaako Arnold	male	14.0
## 15	Panula, Master. Urho Abraham	male	2.0
## 16	Kink-Heilmann, Mr. Anton	male	29.0
## 17	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson)	female	38.0
## 18	Asplund, Master. Clarence Gustaf Hugo	male	9.0
## 19	Asplund, Miss. Lillian Gertrud	female	5.0
## 20	Asplund, Master. Edvin Rojj Felix	male	3.0

## 21	Asplund, Master. Filip Oscar	male	13.0
## 22	Asplund, Mr. Carl Oscar Vilhelm Gustafsson	male	40.0
## 23	Asplund, Master. Carl Edgar	male	5.0
## 24	Andersson, Mr. Anders Johan	male	39.0
## 25	Andersson, Miss. Ellis Anna Maria	female	2.0
## 26	Andersson, Miss. Ingeborg Constanzia	female	9.0
## 27	Andersson, Miss. Sigrid Elisabeth	female	11.0
## 28	Andersson, Mrs. Anders Johan (Alfrida Konstantia Brogren)	female	39.0
## 29	Andersson, Miss. Ebba Iris Alfrida	female	6.0
## 30	Andersson, Master. Sigvard Harald Elias	male	4.0
## 31	Skoog, Master. Harald	male	4.0
## 32	Skoog, Mrs. William (Anna Bernhardina Karlsson)	female	45.0
## 33	Skoog, Mr. Wilhelm	male	40.0
## 34	Skoog, Miss. Mabel	female	9.0
## 35	Skoog, Miss. Margit Elizabeth	female	2.0
## 36	Skoog, Master. Karl Thorsten	male	10.0
## 37	Andersson, Miss. Ida Augusta Margareta	female	38.0
## 38	Palsson, Master. Gosta Leonard	male	2.0
## 39	Palsson, Miss. Torborg Danira	female	8.0
## 40	Palsson, Miss. Stina Viola	female	3.0
## 41	Palsson, Mrs. Nils (Alma Cornelia Berglund)	female	29.0
## 42	Palsson, Master. Paul Folke	male	6.0
## 43	Rice, Master. Eugene	male	2.0
## 44	Rice, Master. Arthur	male	4.0
## 45	Rice, Master. Eric	male	7.0
## 46	Rice, Master. George Hugh	male	8.0
## 47	Rice, Mrs. William (Margaret Norton)	female	39.0
## 48	Rice, Master. Albert	male	10.0
## 49	Lefebre, Master. Henry Forbes	male	NA
## 50	Lefebre, Miss. Mathilde	female	NA
## 51	Lefebre, Miss. Ida	female	NA
## 52	Lefebre, Miss. Jeannie	female	NA
## 53	Lefebre, Mrs. Frank (Frances)	female	NA
## 54	Goodwin, Master. William Frederick	male	11.0
## 55	Goodwin, Miss. Lillian Amy	female	16.0
## 56	Goodwin, Master. Sidney Leonard	male	1.0
## 57	Goodwin, Master. Harold Victor	male	9.0
## 58	Goodwin, Mrs. Frederick (Augusta Tyler)	female	43.0
## 59	Goodwin, Mr. Charles Edward	male	14.0
## 60	Goodwin, Mr. Charles Frederick	male	40.0
## 61	Goodwin, Miss. Jessie Allis	female	10.0
## 62	Sage, Master. Thomas Henry	male	NA
## 63	Sage, Miss. Constance Gladys	female	NA
## 64	Sage, Mr. Frederick	male	NA
## 65	Sage, Mr. George John Jr	male	NA
## 66	Sage, Miss. Stella Anna	female	NA
## 67	Sage, Mr. Douglas Bullen	male	NA
## 68	Sage, Miss. Dorothy Edith "Dolly"	female	NA
## 69	Sage, Miss. Ada	female	NA
## 70	Sage, Mr. John George	male	NA
## 71	Sage, Master. William Henry	male	14.5
## 72	Sage, Mrs. John (Annie Bullen)	female	NA
## 73	Ryerson, Miss. Emily Borie	female	18.0
## 74	Ryerson, Miss. Susan Parker "Suzette"	female	21.0

## 75			Ryerson, Mrs. Arthur Larned (Emily Maria Borie)	female	48.0
## 76			Ryerson, Master. John Borie	male	13.0
## 77			Ryerson, Mr. Arthur Larned	male	61.0
## 78			Ford, Mr. William Neal	male	16.0
## 79			Ford, Miss. Robina Maggie "Ruby"	female	9.0
## 80			Ford, Miss. Doolina Margaret "Daisy"	female	21.0
## 81			Ford, Mrs. Edward (Margaret Ann Watson)	female	48.0
## 82			Ford, Mr. Edward Watson	male	18.0
##	SibSp	Parch	Ticket	Fare	Cabin Embarked Title Age1
## 1	3	2	19950	263.0000	C23 C25 C27 S Mr 19.0
## 2	3	2	19950	263.0000	C23 C25 C27 S Miss 23.0
## 3	3	2	19950	263.0000	C23 C25 C27 S Miss 24.0
## 4	1	4	19950	263.0000	C23 C25 C27 S Mr 64.0
## 5	3	2	19950	263.0000	C23 C25 C27 S Miss 28.0
## 6	1	4	19950	263.0000	C23 C25 C27 S Mrs 60.0
## 7	1	3	29105	23.0000	S Mrs 54.0
## 8	2	3	29106	18.7500	S Mrs 24.0
## 9	4	2	3101281	7.9250	S Miss 17.0
## 10	4	1	3101295	39.6875	S Master 7.0
## 11	4	1	3101295	39.6875	S Master 1.0
## 12	4	1	3101295	39.6875	S Mr 16.0
## 13	0	5	3101295	39.6875	S Mrs 41.0
## 14	4	1	3101295	39.6875	S Mr 14.0
## 15	4	1	3101295	39.6875	S Master 2.0
## 16	3	1	315153	22.0250	S Mr 29.0
## 17	1	5	347077	31.3875	S Mrs 38.0
## 18	4	2	347077	31.3875	S Master 9.0
## 19	4	2	347077	31.3875	S Miss 5.0
## 20	4	2	347077	31.3875	S Master 3.0
## 21	4	2	347077	31.3875	S Master 13.0
## 22	1	5	347077	31.3875	S Mr 40.0
## 23	4	2	347077	31.3875	S Master 5.0
## 24	1	5	347082	31.2750	S Mr 39.0
## 25	4	2	347082	31.2750	S Miss 2.0
## 26	4	2	347082	31.2750	S Miss 9.0
## 27	4	2	347082	31.2750	S Miss 11.0
## 28	1	5	347082	31.2750	S Mrs 39.0
## 29	4	2	347082	31.2750	S Miss 6.0
## 30	4	2	347082	31.2750	S Master 4.0
## 31	3	2	347088	27.9000	S Master 4.0
## 32	1	4	347088	27.9000	S Mrs 45.0
## 33	1	4	347088	27.9000	S Mr 40.0
## 34	3	2	347088	27.9000	S Miss 9.0
## 35	3	2	347088	27.9000	S Miss 2.0
## 36	3	2	347088	27.9000	S Master 10.0
## 37	4	2	347091	7.7750	S Miss 38.0
## 38	3	1	349909	21.0750	S Master 2.0
## 39	3	1	349909	21.0750	S Miss 8.0
## 40	3	1	349909	21.0750	S Miss 3.0
## 41	0	4	349909	21.0750	S Mrs 29.0
## 42	3	1	349909	21.0750	S Master 6.0
## 43	4	1	382652	29.1250	Q Master 2.0
## 44	4	1	382652	29.1250	Q Master 4.0
## 45	4	1	382652	29.1250	Q Master 7.0

## 46	4	1	382652	29.1250		Q Master	8.0
## 47	0	5	382652	29.1250		Q Mrs	39.0
## 48	4	1	382652	29.1250		Q Master	10.0
## 49	3	1	4133	25.4667		S Master	4.0
## 50	3	1	4133	25.4667		S Miss	15.0
## 51	3	1	4133	25.4667		S Miss	15.0
## 52	3	1	4133	25.4667		S Miss	15.0
## 53	0	4	4133	25.4667		S Mrs	37.0
## 54	5	2	CA 2144	46.9000		S Master	11.0
## 55	5	2	CA 2144	46.9000		S Miss	16.0
## 56	5	2	CA 2144	46.9000		S Master	1.0
## 57	5	2	CA 2144	46.9000		S Master	9.0
## 58	1	6	CA 2144	46.9000		S Mrs	43.0
## 59	5	2	CA 2144	46.9000		S Mr	14.0
## 60	1	6	CA 2144	46.9000		S Mr	40.0
## 61	5	2	CA 2144	46.9000		S Miss	10.0
## 62	8	2	CA. 2343	69.5500		S Master	4.0
## 63	8	2	CA. 2343	69.5500		S Miss	15.0
## 64	8	2	CA. 2343	69.5500		S Mr	33.0
## 65	8	2	CA. 2343	69.5500		S Mr	33.0
## 66	8	2	CA. 2343	69.5500		S Miss	15.0
## 67	8	2	CA. 2343	69.5500		S Mr	33.0
## 68	8	2	CA. 2343	69.5500		S Miss	15.0
## 69	8	2	CA. 2343	69.5500		S Miss	15.0
## 70	1	9	CA. 2343	69.5500		S Mr	33.0
## 71	8	2	CA. 2343	69.5500		S Master	14.5
## 72	1	9	CA. 2343	69.5500		S Mrs	37.0
## 73	2	2	PC 17608	262.3750	B57 B59 B63 B66	C Miss	18.0
## 74	2	2	PC 17608	262.3750	B57 B59 B63 B66	C Miss	21.0
## 75	1	3	PC 17608	262.3750	B57 B59 B63 B66	C Mrs	48.0
## 76	2	2	PC 17608	262.3750	B57 B59 B63 B66	C Master	13.0
## 77	1	3	PC 17608	262.3750	B57 B59 B63 B66	C Mr	61.0
## 78	1	3	W./C. 6608	34.3750		S Mr	16.0
## 79	2	2	W./C. 6608	34.3750		S Miss	9.0
## 80	2	2	W./C. 6608	34.3750		S Miss	21.0
## 81	1	3	W./C. 6608	34.3750		S Mrs	48.0
## 82	2	2	W./C. 6608	34.3750		S Mr	18.0
##	AgeGroup	SibGroup	ParGroup	FamilySize	Role		
## 1	14.5-21	(2,8]	(0,3]	5	Mr		
## 2	21-28	(2,8]	(0,3]	5	Miss		
## 3	21-28	(2,8]	(0,3]	5	Miss		
## 4	50-	(0,2]	(3,9]	5	father		
## 5	21-28	(2,8]	(0,3]	5	Miss		
## 6	50-	(0,2]	(3,9]	5	mother		
## 7	50-	(0,2]	(0,3]	5	mother		
## 8	21-28	(0,2]	(0,3]	5	mother		
## 9	14.5-21	(2,8]	(0,3]	5	Miss		
## 10	0-7	(2,8]	(0,3]	5	Master		
## 11	0-7	(2,8]	(0,3]	5	Master		
## 12	14.5-21	(2,8]	(0,3]	5	Mr		
## 13	35-50	(-1,0]	(3,9]	5	mother		
## 14	7-14.5	(2,8]	(0,3]	5	Mr		
## 15	0-7	(2,8]	(0,3]	5	Master		
## 16	28-35	(2,8]	(0,3]	5	Mr		

## 17	35-50	(0,2]	(3,9]	5 mother
## 18	7-14.5	(2,8]	(0,3]	5 Master
## 19	0-7	(2,8]	(0,3]	5 Miss
## 20	0-7	(2,8]	(0,3]	5 Master
## 21	7-14.5	(2,8]	(0,3]	5 Master
## 22	35-50	(0,2]	(3,9]	5 father
## 23	0-7	(2,8]	(0,3]	5 Master
## 24	35-50	(0,2]	(3,9]	5 father
## 25	0-7	(2,8]	(0,3]	5 Miss
## 26	7-14.5	(2,8]	(0,3]	5 Miss
## 27	7-14.5	(2,8]	(0,3]	5 Miss
## 28	35-50	(0,2]	(3,9]	5 mother
## 29	0-7	(2,8]	(0,3]	5 Miss
## 30	0-7	(2,8]	(0,3]	5 Master
## 31	0-7	(2,8]	(0,3]	5 Master
## 32	35-50	(0,2]	(3,9]	5 mother
## 33	35-50	(0,2]	(3,9]	5 father
## 34	7-14.5	(2,8]	(0,3]	5 Miss
## 35	0-7	(2,8]	(0,3]	5 Miss
## 36	7-14.5	(2,8]	(0,3]	5 Master
## 37	35-50	(2,8]	(0,3]	5 Miss
## 38	0-7	(2,8]	(0,3]	5 Master
## 39	7-14.5	(2,8]	(0,3]	5 Miss
## 40	0-7	(2,8]	(0,3]	5 Miss
## 41	28-35	(-1,0]	(3,9]	5 mother
## 42	0-7	(2,8]	(0,3]	5 Master
## 43	0-7	(2,8]	(0,3]	5 Master
## 44	0-7	(2,8]	(0,3]	5 Master
## 45	0-7	(2,8]	(0,3]	5 Master
## 46	7-14.5	(2,8]	(0,3]	5 Master
## 47	35-50	(-1,0]	(3,9]	5 mother
## 48	7-14.5	(2,8]	(0,3]	5 Master
## 49	0-7	(2,8]	(0,3]	5 Master
## 50	14.5-21	(2,8]	(0,3]	5 Miss
## 51	14.5-21	(2,8]	(0,3]	5 Miss
## 52	14.5-21	(2,8]	(0,3]	5 Miss
## 53	35-50	(-1,0]	(3,9]	5 mother
## 54	7-14.5	(2,8]	(0,3]	5 Master
## 55	14.5-21	(2,8]	(0,3]	5 Miss
## 56	0-7	(2,8]	(0,3]	5 Master
## 57	7-14.5	(2,8]	(0,3]	5 Master
## 58	35-50	(0,2]	(3,9]	5 mother
## 59	7-14.5	(2,8]	(0,3]	5 Mr
## 60	35-50	(0,2]	(3,9]	5 father
## 61	7-14.5	(2,8]	(0,3]	5 Miss
## 62	0-7	(2,8]	(0,3]	5 Master
## 63	14.5-21	(2,8]	(0,3]	5 Miss
## 64	28-35	(2,8]	(0,3]	5 Mr
## 65	28-35	(2,8]	(0,3]	5 Mr
## 66	14.5-21	(2,8]	(0,3]	5 Miss
## 67	28-35	(2,8]	(0,3]	5 Mr
## 68	14.5-21	(2,8]	(0,3]	5 Miss
## 69	14.5-21	(2,8]	(0,3]	5 Miss
## 70	28-35	(0,2]	(3,9]	5 father

```
## 71 7-14.5 (2,8] (0,3] 5 Master
## 72 35-50 (0,2] (3,9] 5 mother
## 73 14.5-21 (0,2] (0,3] 5 Miss
## 74 14.5-21 (0,2] (0,3] 5 Miss
## 75 35-50 (0,2] (0,3] 5 mother
## 76 7-14.5 (0,2] (0,3] 5 Master
## 77 50- (0,2] (0,3] 5 father
## 78 14.5-21 (0,2] (0,3] 5 Mr
## 79 7-14.5 (0,2] (0,3] 5 Miss
## 80 14.5-21 (0,2] (0,3] 5 Miss
## 81 35-50 (0,2] (0,3] 5 mother
## 82 14.5-21 (0,2] (0,3] 5 Mr
```

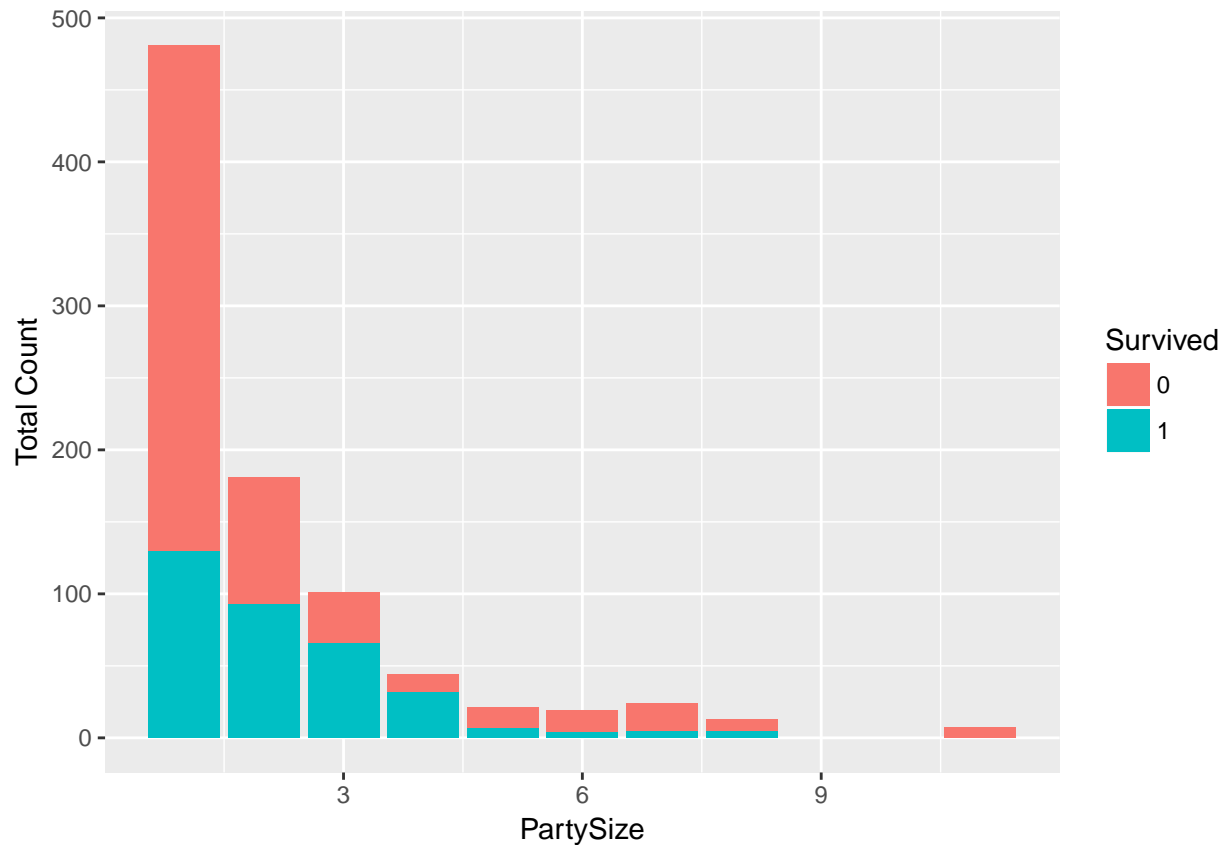
```
ticket.party <- data %>%
  group_by(Ticket) %>%
  summarise(PartySize=n())
# merge PartySize to data
data <- left_join(data,ticket.party,by="Ticket")

# look at PartySize
table(data$PartySize)
```

```
##
## 1 2 3 4 5 6 7 8 11
## 713 264 147 64 35 24 35 16 11
```

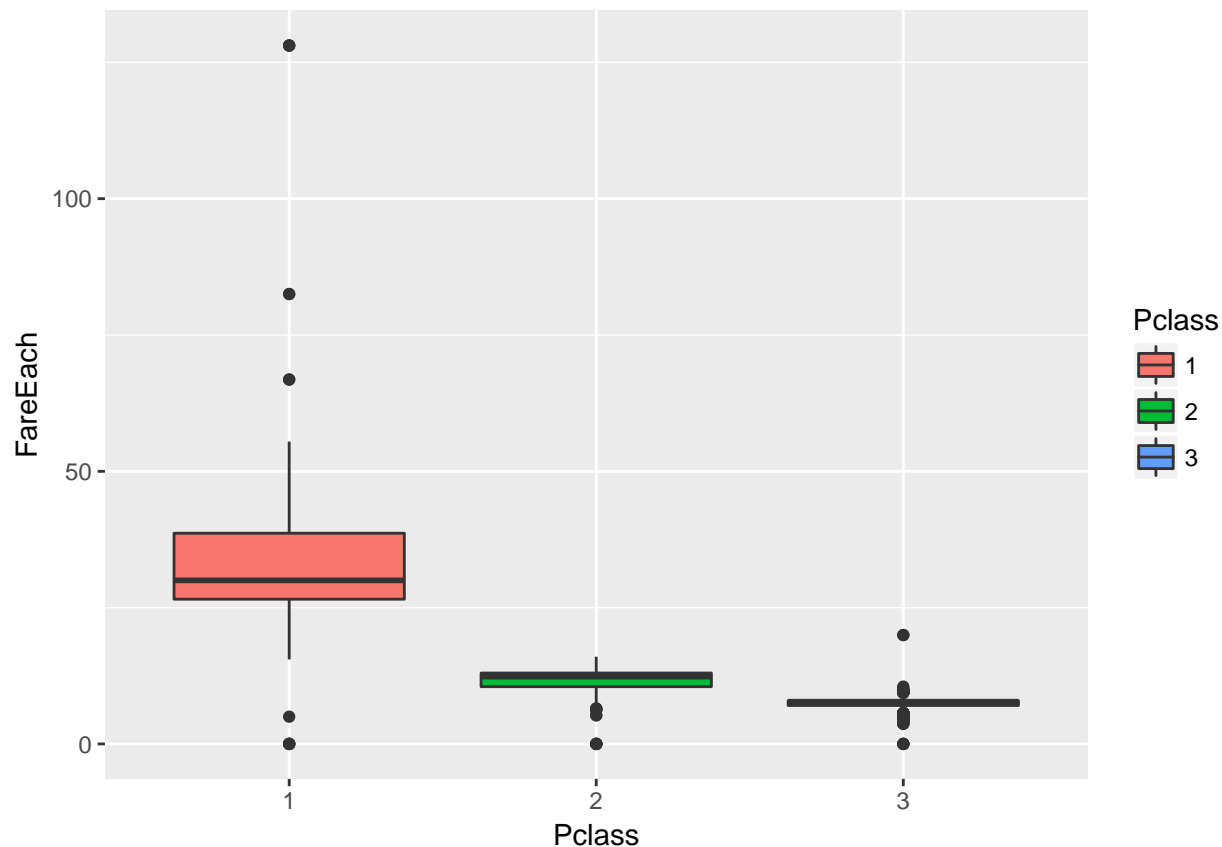
```
# Partysize vs survival
ggplot(data[1:891,], aes(x = PartySize, fill = Survived)) +
  geom_bar() +
  xlab("PartySize") +
  ylab("Total Count") +
  labs(fill = "Survived")
```





Fare FareEach, Fare for each passenger. FareGroup, group FareEach, the more a passenger paid, the more chance of survival they had.

```
# recalculate fare for each passenger
data$FareEach <- with(data, Fare/PartySize)
# impute missing values
# Intuitively, FareEach should be associated with Pclass, the boxplot proves this.
# distributions of FareEach for Pclass
ggplot(data[1:891,], aes(x = Pclass, y = FareEach, fill = Pclass)) +
  geom_boxplot() +
  xlab("Pclass") +
  ylab("FareEach")
```



```
summary(data$Fare)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##  0.000   7.896   14.450   33.300   31.280   512.300         1
```

```
filter(data, is.na(Fare))
```

```
##   PassengerId Survived Pclass      Name  Sex  Age SibSp Parch
## 1      1044      <NA>      3 Storey, Mr. Thomas male 60.5    0    0
##   Ticket Fare Cabin Embarked Title Age1 AgeGroup SibGroup ParGroup
## 1   3701   NA      S      Mr 60.5    50-  (-1,0]  (-1,0]
##   FamilySize Role PartySize FareEach
## 1          1  Mr          1      NA
```

```
summary(data[which(data$Pclass==3), "FareEach"]) # mean=7.329
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##  0.000   7.060   7.750   7.329   7.925   19.970         1
```

```
data$FareEach[which(is.na(data$Fare))] <- 7.329
```

```
# Fare is associated with Survival?
```

```
summary(data$FareEach)
```

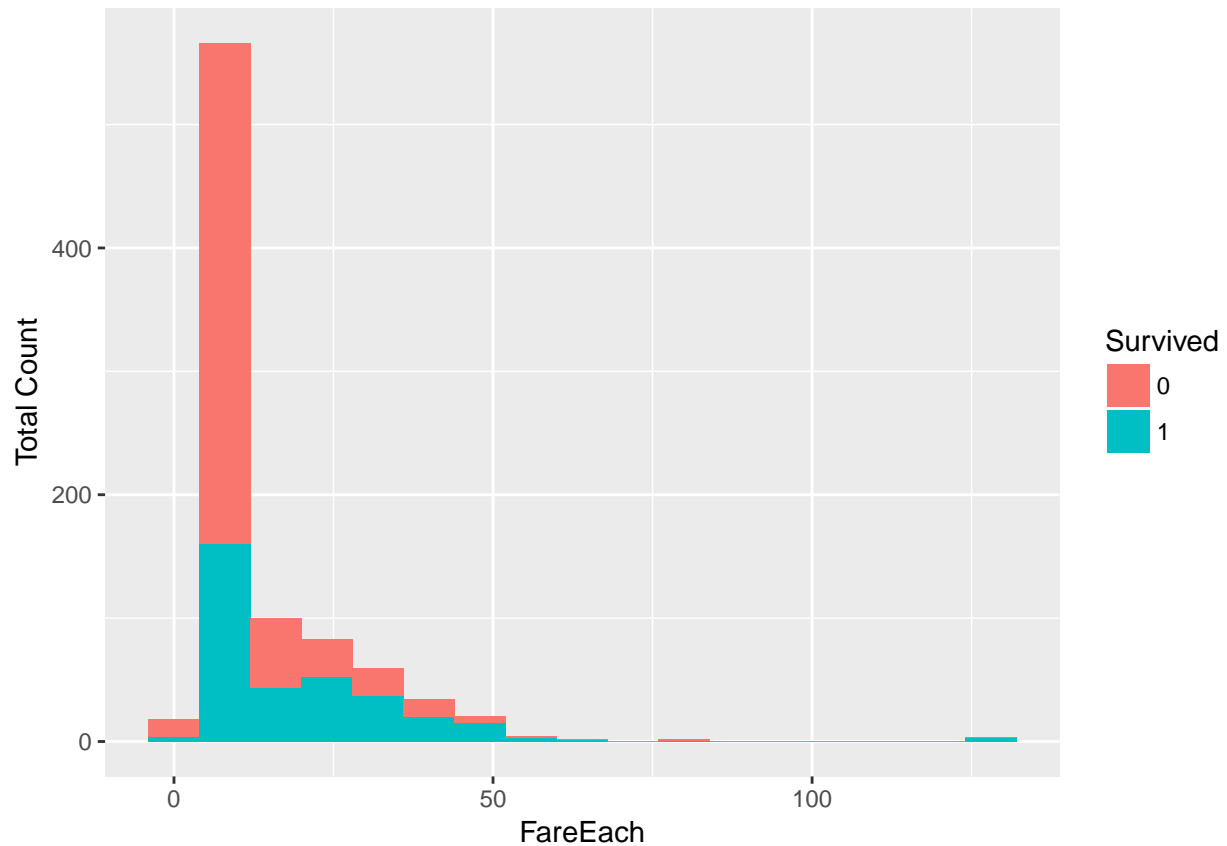
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.00   7.55   8.05   14.75   15.00   128.10
```

```
ggplot(data[1:891,], aes(x = FareEach, fill = Survived)) +
  geom_histogram(binwidth = 8) +
```

```

xlab("FareEach") +
ylab("Total Count") +
labs(fill = "Survived")

```



```

# group FareEach by quantile
summary(data$FareEach)

```

```

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.00   7.55   8.05   14.75   15.00   128.10

```

```

data$FareGroup <- cut(data$FareEach, breaks=c(-1,0,7.85,8.05,15.00,129))

```

```

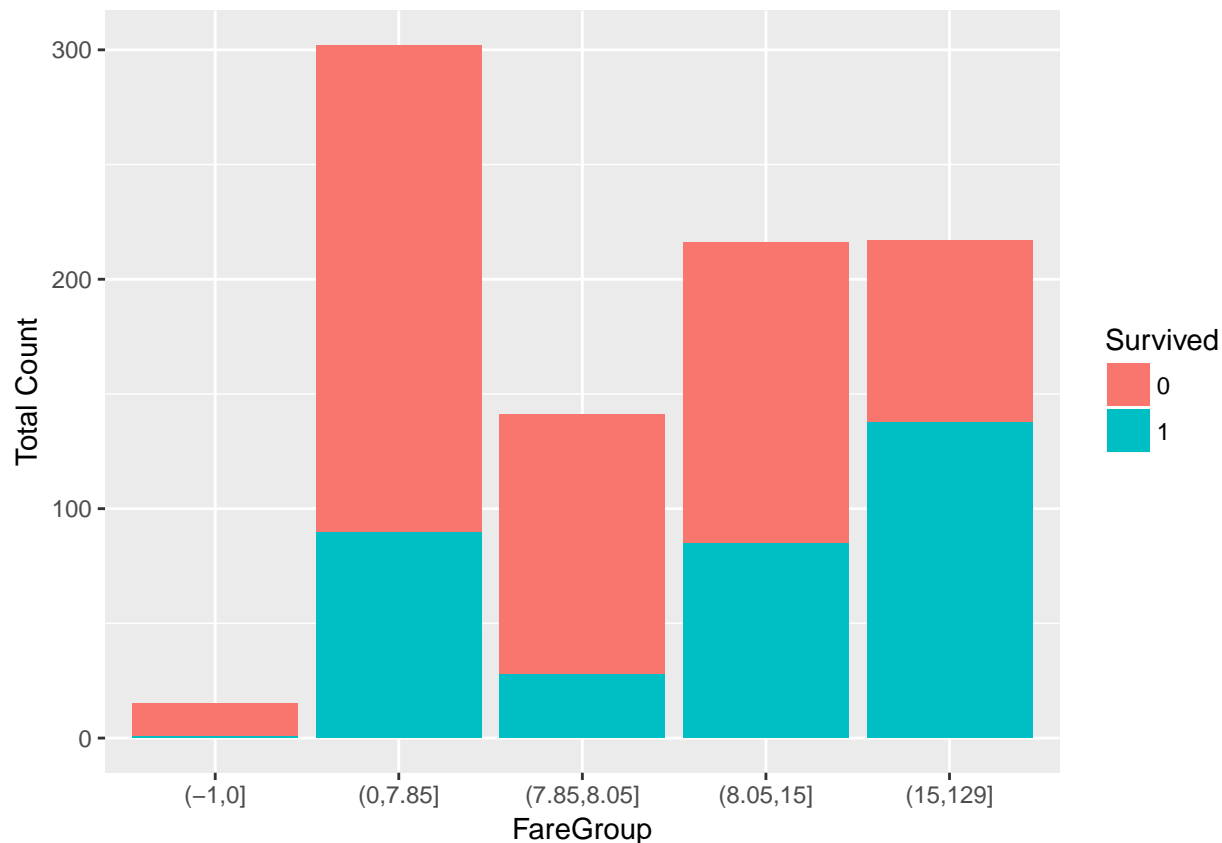
# FareGroup vs Survival

```

```

ggplot(data[1:891,], aes(x = FareGroup, fill = Survived)) +
geom_bar()+
xlab("FareGroup") +
ylab("Total Count") +
labs(fill = "Survived")

```



```
# recode PartySize since there are few examples for PartySize>4
data$PartySize[data$PartySize>5] <- 5
data$PartySize <- as.factor(data$PartySize)
```

Cabin CabinFirst, the first letter of Cabin, may represent different position of the ship, so it may associated with survival rate. The plot proves it. Passengers whose Cabins' first letters are B,C,D,E,F had more chance to survive. However, most passengers didn't have a cabin, and these had much less chance to survive than those having a cabin. I will create a feature, HaveCabin, indicate if a passenger had a cabin, and use it in modeling.

```
# Replace empty cabins with a "U"
length(unique(data$Cabin))
```

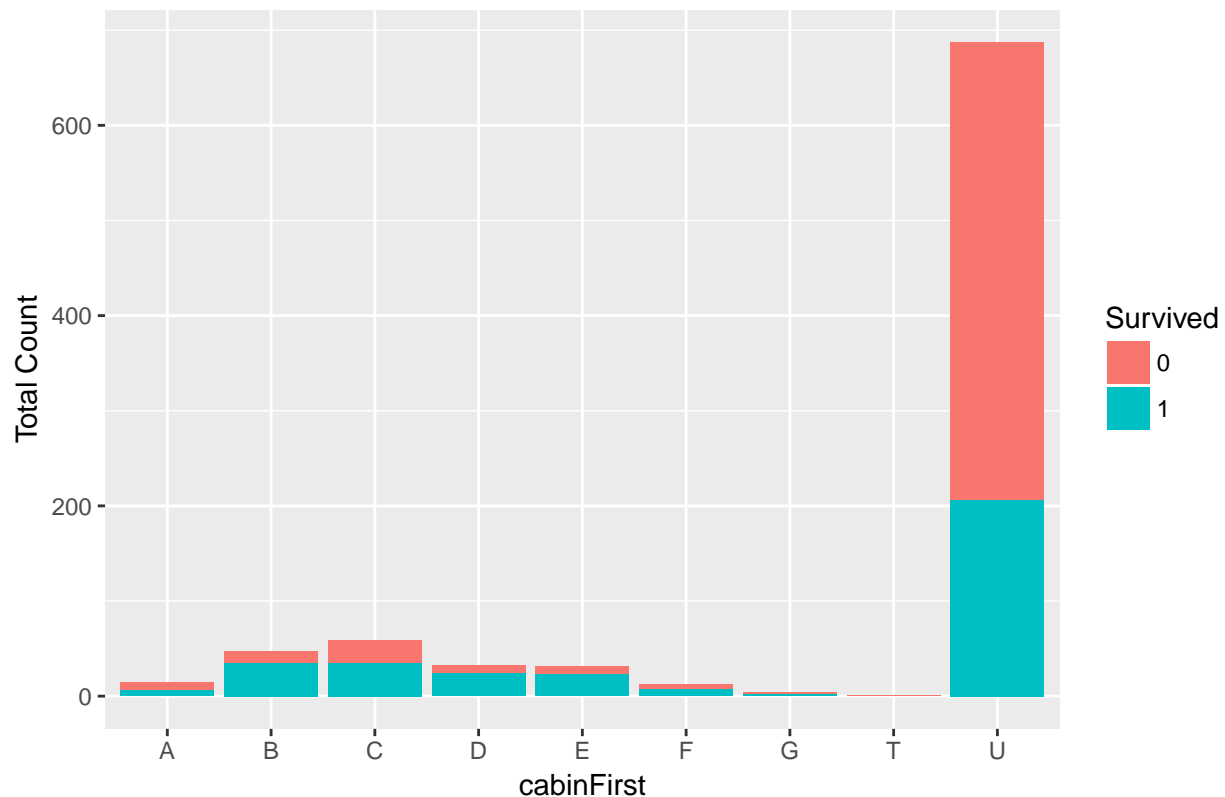
```
## [1] 187
```

```
data$Cabin[data$Cabin == ""] <- "U"
```

```
# Take a look at just the first letter as a factor
data$CabinFirst <- as.factor(substr(data$Cabin, 1, 1))
```

```
# Plot
# Cabin is associated with survival rate? Yes
ggplot(data[1:891,], aes(x = CabinFirst, fill = Survived)) +
  geom_bar() +
  ggtitle("Survivability by Pclass,CabinFirst") +
  xlab("cabinFirst") +
  ylab("Total Count") +
  labs(fill = "Survived")
```

## Survivability by Pclass,CabinFirst



```
data$HaveCabin <- as.factor(ifelse(data$Cabin=="U", "0", "1"))
```

Side: Since the ship was hit on the left side, maybe side is a good predictor. Maybe Cabin's last number, like house/room number, can have the information. The plot shows passengers having cabins on the left side of the ship had slightly less chance to survive than those on the right side.

```
CabinLast <- str_sub(data$Cabin, -1, -1)
table(CabinLast)
```

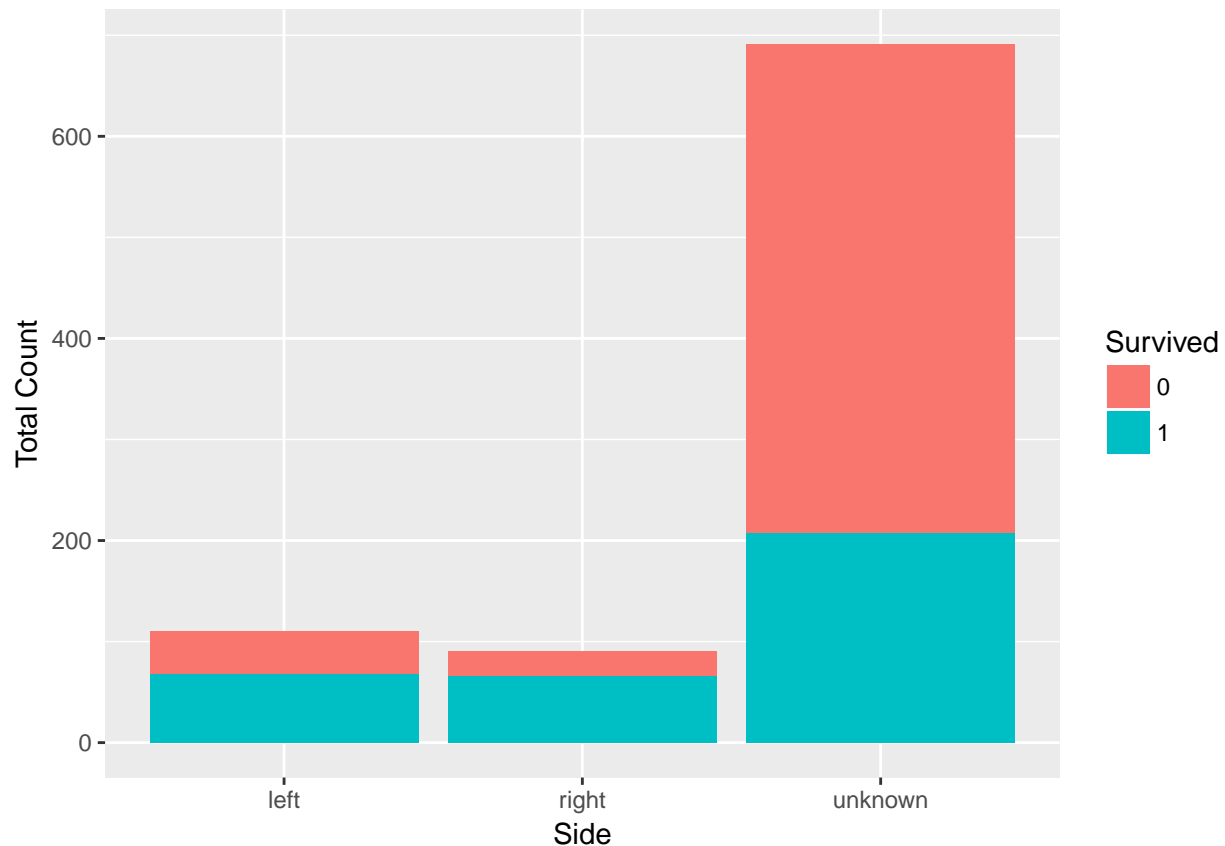
```
## CabinLast
##    0    1    2    3    4    5    6    7    8    9    D    F    T    U
##   30   29   24   25   28   27   45   28   32   21    4    1    1 1014
```

```
Side <- rep("unknown", length(CabinLast))
Side[CabinLast %in% c("0", "2", "4", "6", "8")] <- "left"
Side[CabinLast %in% c("1", "3", "5", "7", "9")] <- "right"
# convert into factor
table(Side)
```

```
## Side
##    left    right unknown
##   159     130     1020
```

```
data$Side <- factor(Side)
# Side Vs Survival
ggplot(data[1:891,], aes(x = Side, fill = Survived)) +
  geom_bar() +
  xlab("Side") +
  ylab("Total Count") +
```

```
labs(fill = "Survived")
```



Embarked: It seems that passenger coming from Cherbourg (C) had more chance to survive. Maybe the proportion of first class passengers was higher for those from Cherbourg than those from Queenstown (Q), Southampton (S). The plot proves it. The passengers from Queenstown (Q) are almost third class, but the survival rate is much higher than that of the third class. From the table, there are more children and women, 53%, in those from Queenstown (Q).

```
# understand Embarked
```

```
table(data$Embarked)
```

```
##
```

```
##      C    Q    S
```

```
##  2 270 123 914
```

```
# replace missing values with mode
```

```
data[which(data$Embarked==""), "Embarked"] <- "S"
```

```
# drop missing values level
```

```
data$Embarked <- factor(data$Embarked, levels = c("C", "Q", "S"))
```

```
# the survival rate is associated with where the passengers are from?
```

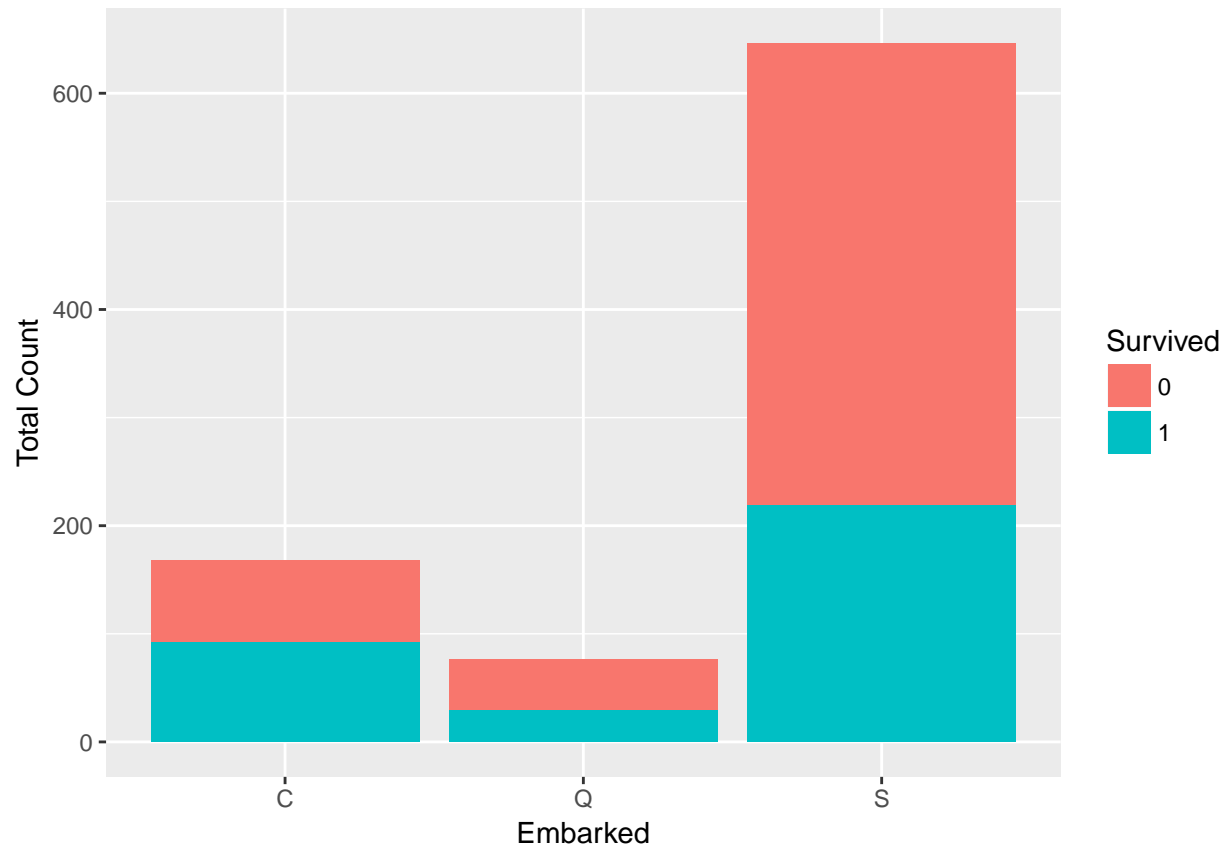
```
ggplot(data[1:891,], aes(x = Embarked, fill = Survived)) +
```

```
  geom_bar() +
```

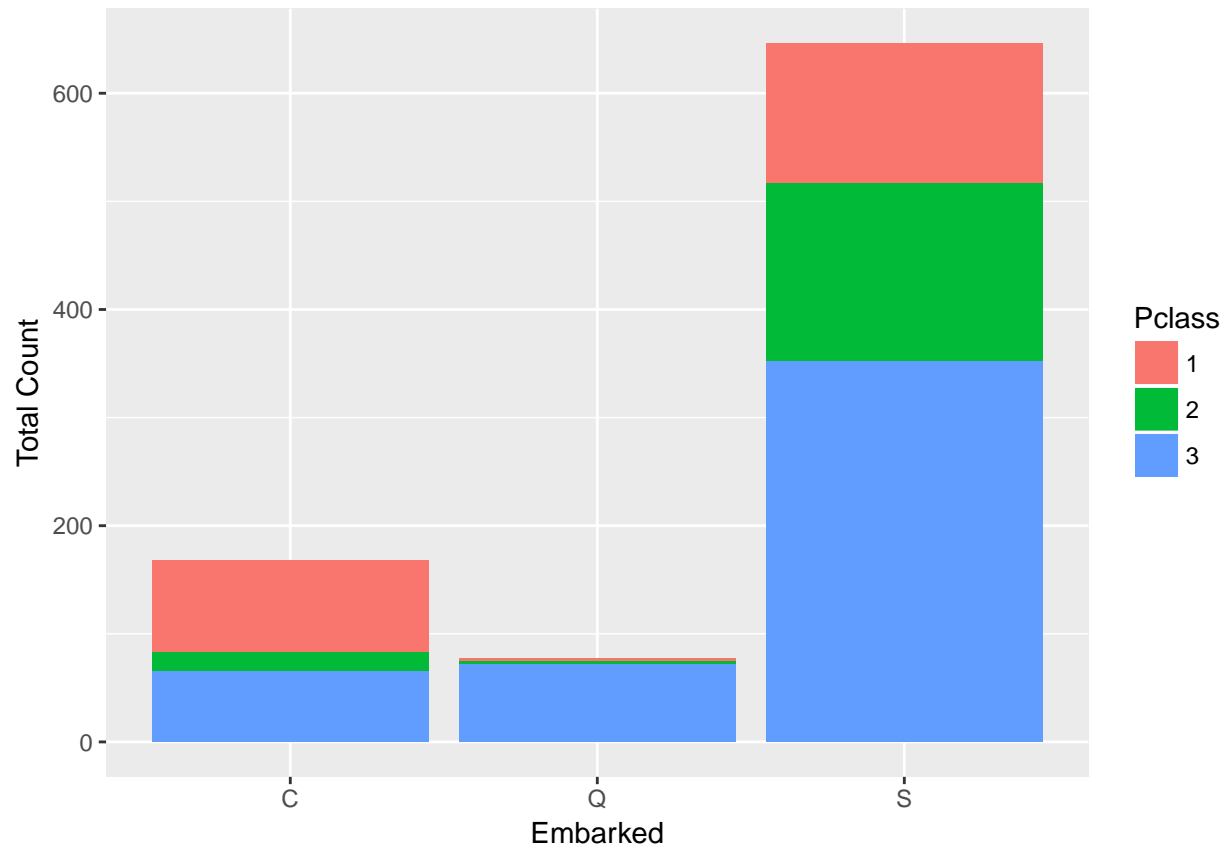
```
  xlab("Embarked") +
```

```
  ylab("Total Count") +
```

```
  labs(fill = "Survived")
```

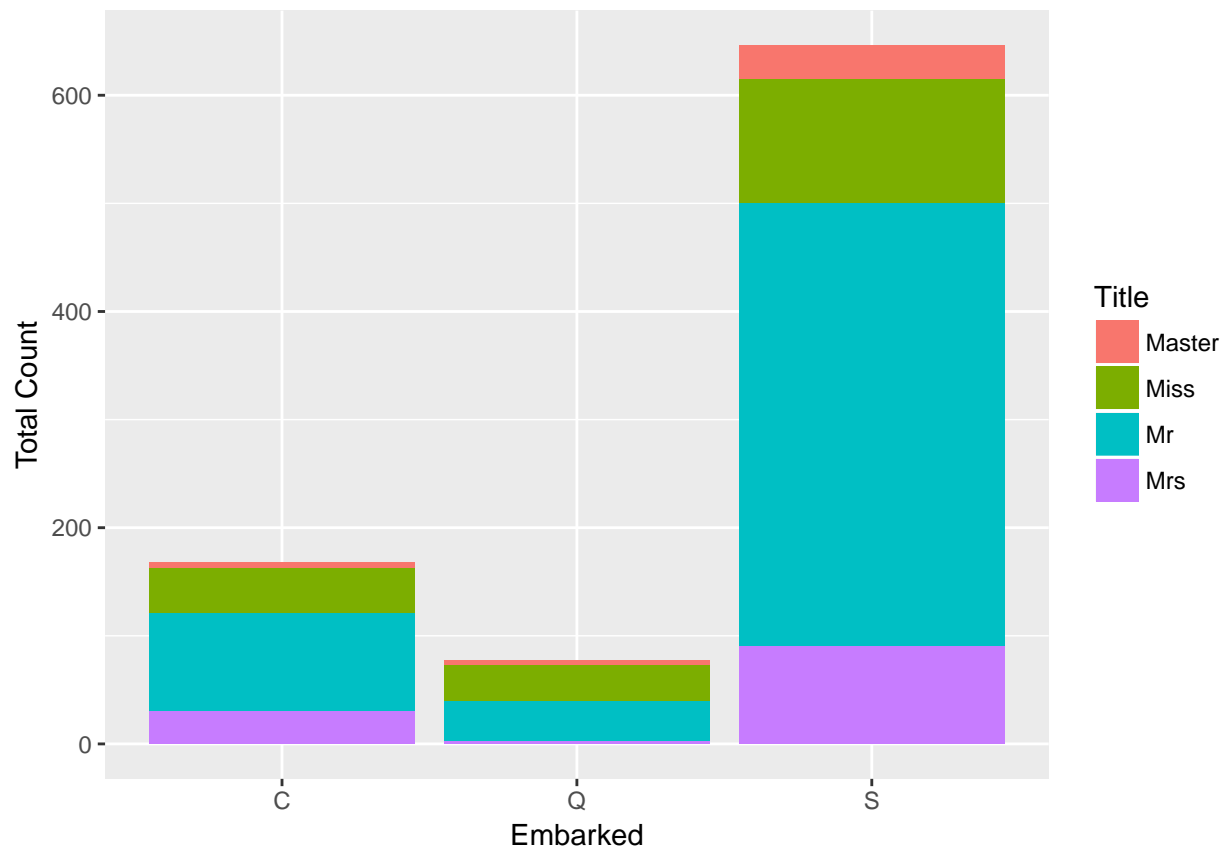


```
# the proportion of first class passengers is higher for those from Cherbourg? Yes  
ggplot(data[1:891,], aes(x = Embarked, fill = Pclass)) +  
  geom_bar() +  
  xlab("Embarked") +  
  ylab("Total Count") +  
  labs(fill = "Pclass")
```



```
# there are more children and women in those from Queenstown (Q)? Yes  
ggplot(data[1:891,], aes(x = Embarked, fill = Title)) +  
  geom_bar() +  
  xlab("Embarked") +  
  ylab("Total Count") +  
  labs(fill = "Title")
```





```
prop.table(table(data$Embarked, data$Title),1)
```

```
##
##      Master      Miss      Mr      Mrs
##  C 0.04074074 0.20000000 0.54074074 0.21851852
##  Q 0.04065041 0.45528455 0.47154472 0.03252033
##  S 0.04912664 0.17358079 0.63100437 0.14628821
```

## Data preparation - creating training and test datasets

Create training dataset to build models and test dataset to make predictions.

```
# split the data frames to training and test datasets
train.df <- data[1:891, c( "Survived", "Pclass", "Sex", "Embarked", "Title", "AgeGroup", "SibGroup", "P
train.df$Survived <- factor(train.df$Survived, levels = c("0","1"))

test.df <- data[892:1309, c( "Pclass", "Sex", "Embarked", "Title", "AgeGroup", "SibGroup", "ParGroup",
```

## check type of features

```
str(train.df)
```

```
## 'data.frame':   891 obs. of  12 variables:
```

```
## $ 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 ...
## $ Sex : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
## $ Embarked : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
## $ Title : Factor w/ 4 levels "Master","Miss",...: 3 4 2 4 3 3 3 1 4 4 ...
## $ AgeGroup : Factor w/ 7 levels "0-7","7-14.5",...: 4 6 4 5 5 5 7 1 4 2 ...
## $ SibGroup : Factor w/ 3 levels "(-1,0]","(0,2]",...: 2 2 1 2 1 1 1 3 1 2 ...
## $ ParGroup : Factor w/ 3 levels "(-1,0]","(0,3]",...: 1 1 1 1 1 1 1 2 2 1 ...
## $ FamilySize: Factor w/ 5 levels "1","2","3","4",...: 2 2 1 2 1 1 1 5 3 2 ...
## $ PartySize : Factor w/ 5 levels "1","2","3","4",...: 1 2 1 2 1 1 2 5 3 2 ...
## $ FareGroup : Factor w/ 5 levels "(-1,0]","(0,7.85]",...: 2 5 3 5 3 4 5 2 2 5 ...
## $ HaveCabin : Factor w/ 2 levels "0","1": 1 2 1 2 1 1 2 1 1 1 ...
```

### Step 3: Training and evaluating models on the training dataset —

Since the training dataset is small, I will use cross validation to tune parameters for every algorithm and evaluate models. Based on the values of Cross validation accuracy of the models, the gbm model, whose CV accuracy is 0.8417, is the best. Therefore, I will use the gbm model as the final model to make predictions on the test dataset.

store fitted accuracy and cross validation accuracy.

```
train.acc <- numeric(5)
cv.acc <- numeric(5)
```

### build a gbm model

```
# tune parameters
ctrl <- trainControl(method = "cv",
                     number = 10 )
grid_gbm <- expand.grid(interaction.depth = c(5, 7, 9),
                       n.trees = (1:4)*50,
                       shrinkage = 0.01,
                       n.minobsinnode = c(12, 14, 16, 18))

set.seed(1234)
m_gbm1 <- train(Survived ~ ., data = train.df,
               method = "gbm",
               trControl = ctrl,
               verbose = FALSE,
               tuneGrid = grid_gbm,
               metric = "Accuracy")
m_gbm1$bestTune

##      n.trees interaction.depth shrinkage n.minobsinnode
## 22      100                7      0.01              14

# Evaluate model performance by cross validation
set.seed(1234)
m_gbm <- train(Survived ~ ., data = train.df,
```

```

        method = "gbm",
        trControl = ctrl,
        verbose = FALSE,
        tuneGrid = m_gbm1$bestTune,
        metric = "Accuracy")

#accuracy
train.acc[1] <- mean(predict(m_gbm,train.df)== train.df$Survived)
train.acc[1]

## [1] 0.8439955

cv.acc[1] <- m_gbm$results$Accuracy
cv.acc[1]

## [1] 0.839508
# look at feature importance
imp.gbm <- varImp(m_gbm, scale = FALSE)
imp.gbm

## gbm variable importance
##
##    only 20 most important variables shown (out of 31)
##
##               Overall
## TitleMr          1272.918
## Pclass3           370.200
## Sexmale           191.694
## PartySize5        143.111
## FamilySize5         86.155
## FareGroup(15,129]   81.265
## HaveCabin1          76.702
## EmbarkedS           41.661
## FareGroup(0,7.85]   37.984
## SibGroup(2,8]       24.552
## AgeGroup21-28       11.836
## PartySize2          11.090
## SibGroup(0,2]       10.631
## AgeGroup35-50       10.037
## ParGroup(0,3]        9.452
## AgeGroup50-         9.179
## TitleMiss           8.718
## EmbarkedQ           7.754
## TitleMrs            6.433
## FamilySize2         6.410

```

## build a random forest model

```

# tune parameters
grid_rf <- expand.grid( .mtry = c(2, 3, 4, 5))
set.seed(1234)
m_rf1 <- train(Survived ~ ., data=train.df,

```

```

        method = "rf",
        metric = "Accuracy",
        trControl = ctrl,
        tuneGrid = grid_rf)
m_rf1$bestTune

##      mtry
## 2      3

# Evaluate model performance by cross validation
set.seed(1234)
m_rf <- train(Survived ~ ., data = train.df,
              method = "rf",
              metric = "Accuracy",
              trControl = ctrl,
              tuneGrid = m_rf1$bestTune
            )

#accuracy
train.acc[2] <- mean(predict(m_rf,train.df)== train.df$Survived)
train.acc[2]

## [1] 0.8664422

cv.acc[2] <- m_rf$results$Accuracy
cv.acc[2]

## [1] 0.8294337

# look at feature importance
imp.rf <- varImp(m_rf, scale = FALSE)
imp.rf

## rf variable importance
##
##      only 20 most important variables shown (out of 31)
##
##              Overall
## TitleMr          46.703
## Sexmale          35.384
## TitleMiss        17.111
## TitleMrs         16.272
## Pclass3          13.988
## HaveCabin1       11.808
## FareGroup(15,129] 10.567
## PartySize5        6.659
## PartySize3        5.121
## FareGroup(0,7.85] 5.054
## FamilySize5       4.941
## EmbarkedS        4.845
## ParGroup(0,3]     4.765
## Pclass2           4.714
## SibGroup(0,2]     4.444
## AgeGroup21-28     3.756
## FareGroup(8.05,15] 3.656
## FamilySize3       3.544

```

```
## PartySize2          3.525
## FamilySize2         3.457
```

## build a xgboost model

```
# tune parameters
grid_xg <- expand.grid( nrounds= (1:4)*50,
                      max_depth= c(7, 9, 11),
                      eta= 0.3,
                      gamma=0,
                      min_child_weight=c(6,8,10),
                      colsample_bytree= c(0.6, 0.8, 1),
                      subsample=1)

set.seed(1234)
m_xg1 <- train(Survived ~ ., data=train.df,
               method = "xgbTree",
               metric = "Accuracy",
               trControl = ctrl,
               tuneGrid = grid_xg)
m_xg1$bestTune

##      nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 62      100      9 0.3    0                1                6            1

# Evaluate model performance by cross validation
set.seed(1234)
m_xg <- train(Survived ~ ., data = train.df,
              method = "xgbTree",
              metric = "Accuracy",
              trControl = ctrl,
              tuneGrid = m_xg1$bestTune
              )

#accuracy
train.acc[3] <- mean(predict(m_xg,train.df)== train.df$Survived)
train.acc[3]

## [1] 0.8799102
cv.acc[3] <- m_xg$results$Accuracy
cv.acc[3]

## [1] 0.8338268

# look at feature importance
imp.xg <- varImp(m_xg, scale = FALSE)
imp.xg

## xgbTree variable importance
##
## only 20 most important variables shown (out of 31)
##
## Overall
## TitleMr      0.471658
## Pclass3      0.153248
```

```
## PartySize5          0.075372
## FareGroup(15,129]   0.039956
## HaveCabin1          0.037084
## FareGroup(0,7.85]   0.030441
## EmbarkedS           0.022529
## FareGroup(7.85,8.05] 0.019584
## AgeGroup21-28        0.017209
## AgeGroup35-50        0.014784
## FareGroup(8.05,15]   0.012722
## SibGroup(0,2]        0.011996
## AgeGroup28-35        0.011380
## TitleMiss           0.011131
## PartySize2           0.011126
## ParGroup(0,3]        0.009285
## AgeGroup14.5-21      0.008611
## Sexmale              0.007275
## PartySize3           0.006151
## EmbarkedQ           0.006010
```

## build a logistic regression model

The fit accuracy is 0.8507, Cv accuracy is 0.8385.

```
# select features
# fit a logistic regression model with all features
model.glm1=glm(Survived ~ ., data=train.df,
               family= "binomial" )
# significnat test
anova(model.glm1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
## NULL			890	1186.66	
## Pclass	2	103.547	888	1083.11	< 2.2e-16 ***
## Sex	1	256.220	887	826.89	< 2.2e-16 ***
## Embarked	2	7.863	885	819.03	0.01962 *
## Title	3	49.215	882	769.81	1.174e-10 ***
## AgeGroup	6	13.217	876	756.59	0.03972 *
## SibGroup	2	35.077	874	721.52	2.416e-08 ***
## ParGroup	2	8.711	872	712.81	0.01284 *
## FamilySize	4	4.704	868	708.10	0.31909
## PartySize	4	4.220	864	703.88	0.37706
## FareGroup	4	5.844	860	698.04	0.21108
## HaveCabin	1	4.268	859	693.77	0.03883 *

```
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# drop insignificant features and fit a model
model.glm2=glm(Survived ~ Pclass + Sex + Embarked + Title + AgeGroup + SibGroup + ParGroup + HaveCabin,
               data=train.df,
               family = "binomial" )

# goodness of fit test
library(ResourceSelection)

## ResourceSelection 0.3-2    2017-02-28

hl <- hoslem.test(model.glm2$y, fitted(model.glm2), g=10)
hl # p-value=0.08, poor fit

##
## Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  model.glm2$y, fitted(model.glm2)
## X-squared = 14.139, df = 8, p-value = 0.07821

# add interaction effects and use sepwise to select features
step.glm <- step(model.glm2,
                 scope = list(upper = as.formula(Survived ~ .^2),
                              lower = as.formula(Survived ~ .)),
                 direction = "both")

## Start:  AIC=748
## Survived ~ Pclass + Sex + Embarked + Title + AgeGroup + SibGroup +
##      ParGroup + HaveCabin
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##           Df Deviance   AIC
## + Pclass:Sex      2   685.99 729.99
## + Pclass:Title     6   681.38 733.38
## + Sex:SibGroup     2   691.57 735.57
## + Title:SibGroup   6   687.92 739.92
## + Pclass:ParGroup  3   698.77 744.77
## + Embarked:ParGroup 3   699.35 745.35
## + SibGroup:ParGroup 3   699.50 745.50
## + Pclass:SibGroup  4   698.34 746.34
## + AgeGroup:SibGroup 10  686.85 746.85
## + SibGroup:HaveCabin 2   703.64 747.64
## <none>              708.00 748.00
## + Embarked:SibGroup 3   702.72 748.72
## + Sex:HaveCabin     1   707.71 749.71
## + Sex:Embarked      2   705.82 749.82
## + Sex:Title         1   708.00 750.00
## + Embarked:HaveCabin 2   706.75 750.75
## + Sex:ParGroup      2   707.48 751.48
## + Pclass:HaveCabin  2   707.62 751.62
## + ParGroup:HaveCabin 2   707.94 751.94
## + Title:HaveCabin   3   706.34 752.34
## + Embarked:Title    6   700.83 752.83
## + AgeGroup:HaveCabin 6   701.45 753.45

```

```

## + Title:ParGroup      4   706.13 754.13
## + Pclass:AgeGroup     12   690.13 754.13
## + Pclass:Embarked      4   707.14 755.14
## + Sex:AgeGroup         6   703.66 755.66
## + Embarked:AgeGroup    12   695.94 759.94
## + AgeGroup:ParGroup    8   706.56 762.56
## + Title:AgeGroup       12   701.42 765.42
##
## Step:  AIC=729.99
## Survived ~ Pclass + Sex + Embarked + Title + AgeGroup + SibGroup +
##           ParGroup + HaveCabin + Pclass:Sex

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##           Df Deviance   AIC
## + Sex:SibGroup      2   665.81 713.81
## + Title:SibGroup     6   661.49 717.49
## + SibGroup:ParGroup   3   676.62 726.62
## + Sex:Embarked        2   679.23 727.23
## + Embarked:ParGroup   3   677.46 727.46
## + AgeGroup:SibGroup  10   664.40 728.40
## + Sex:HaveCabin       1   683.80 729.80
## <none>                685.99 729.99
## + Embarked:SibGroup   3   680.53 730.53
## + Embarked:Title      6   674.68 730.68
## + SibGroup:HaveCabin  2   683.09 731.09
## + Sex:Title           1   685.99 731.99
## + Embarked:HaveCabin  2   684.25 732.25
## + Pclass:SibGroup     4   680.34 732.34
## + Pclass:ParGroup     3   682.92 732.92
## + Title:HaveCabin     3   682.95 732.95
## + Pclass:Title        4   681.38 733.38
## + Pclass:AgeGroup     12   665.45 733.45
## + Pclass:HaveCabin    2   685.78 733.78
## + Sex:ParGroup        2   685.82 733.82
## + ParGroup:HaveCabin  2   685.92 733.92
## + AgeGroup:HaveCabin  6   680.65 736.65
## + Pclass:Embarked     4   684.94 736.94
## + Title:ParGroup      4   685.06 737.06
## + Sex:AgeGroup        6   681.36 737.36
## + Embarked:AgeGroup   12   669.90 737.90
## + Title:AgeGroup      12   676.46 744.46
## + AgeGroup:ParGroup   8   684.59 744.59
## - Pclass:Sex          2   708.00 748.00
##
## Step:  AIC=713.81
## Survived ~ Pclass + Sex + Embarked + Title + AgeGroup + SibGroup +
##           ParGroup + HaveCabin + Pclass:Sex + Sex:SibGroup

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

```



```

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
##          Df Deviance      AIC
## + Sex:Embarked      2    658.8   710.8
## <none>                665.8   713.8
## + Embarked:ParGroup  3    660.0   714.0
## + Sex:HaveCabin      1    664.0   714.0
## + SibGroup:ParGroup  3    660.6   714.6
## + Embarked:HaveCabin 2    663.7   715.7
## + Sex:Title          1    665.8   715.8
## + Embarked:SibGroup  3    662.3   716.3
## + SibGroup:HaveCabin 2    664.5   716.5
## + Pclass:AgeGroup    12    645.2   717.2
## + Embarked:Title     6    657.2   717.2
## + Pclass:HaveCabin   2    665.6   717.6
## + Pclass:ParGroup    3    663.7   717.7
## + Sex:ParGroup       2    665.7   717.7
## + ParGroup:HaveCabin 2    665.7   717.7
## + AgeGroup:HaveCabin 6    657.8   717.8
## + Title:HaveCabin    3    663.8   717.8
## + Pclass:SibGroup    4    663.4   719.4
## + Title:SibGroup     5    661.5   719.5
## + Pclass:Embarked    4    664.6   720.6
## + Pclass:Title       4    665.2   721.2
## + Title:ParGroup     4    665.6   721.6
## + AgeGroup:SibGroup  10    654.5   722.5
## + Embarked:AgeGroup  12    651.2   723.2
## + Sex:AgeGroup       6    663.6   723.6
## - Sex:SibGroup       2    686.0   730.0
## + AgeGroup:ParGroup  9    666.0   732.0
## - Pclass:Sex         2    691.6   735.6
## + Title:AgeGroup     11  11173.5 11243.5
##
## Step:  AIC=710.82
## Survived ~ Pclass + Sex + Embarked + Title + AgeGroup + SibGroup +
##           ParGroup + HaveCabin + Pclass:Sex + Sex:SibGroup + Sex:Embarked

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

```

```

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
##          Df Deviance      AIC
## + Embarked:ParGroup    3    651.6   709.6
## <none>                  658.8   710.8
## + Sex:HaveCabin        1    657.2   711.2
## + SibGroup:ParGroup    3    653.5   711.5
## + Embarked:HaveCabin   2    656.4   712.4
## + Sex:Title            1    658.8   712.8
## + SibGroup:HaveCabin   2    657.6   713.6
## - Sex:Embarked         2    665.8   713.8
## + Embarked:SibGroup    3    656.0   714.0
## + Pclass:AgeGroup     12    638.5   714.5
## + Pclass:ParGroup      3    656.6   714.6
## + Sex:ParGroup         2    658.7   714.7
## + ParGroup:HaveCabin   2    658.7   714.7
## + Pclass:HaveCabin     2    658.7   714.7
## + AgeGroup:HaveCabin   6    650.7   714.7
## + Title:HaveCabin      3    657.1   715.1
## + Pclass:SibGroup      4    656.4   716.4
## + Title:SibGroup       5    654.9   716.9
## + Embarked:Title       4    657.2   717.2
## + Pclass:Title         4    657.9   717.9
## + Pclass:Embarked      4    658.4   718.4
## + Title:ParGroup       4    658.7   718.7
## + AgeGroup:SibGroup    10    647.8   719.8
## + Sex:AgeGroup         6    656.7   720.7
## + AgeGroup:ParGroup    8    657.4   725.4
## + Embarked:AgeGroup    12    650.4   726.4
## - Sex:SibGroup         2    679.2   727.2
## - Pclass:Sex           2    689.5   737.5
## + Title:AgeGroup       11  13047.8 13121.8
##
## Step:  AIC=709.56
## Survived ~ Pclass + Sex + Embarked + Title + AgeGroup + SibGroup +
##           ParGroup + HaveCabin + Pclass:Sex + Sex:SibGroup + Sex:Embarked +
##           Embarked:ParGroup

```

```

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##           Df Deviance      AIC
## <none>           651.6   709.6
## + Sex:HaveCabin      1    650.0   710.0
## - Embarked:ParGroup  3    658.8   710.8
## + Embarked:HaveCabin 2    649.4   711.4
## + SibGroup:ParGroup  3    647.5   711.5
## + Sex:Title          1    651.6   711.6
## + Embarked:SibGroup  3    647.7   711.7
## + SibGroup:HaveCabin 2    650.3   712.3
## + Sex:ParGroup       2    651.2   713.2
## + ParGroup:HaveCabin 2    651.3   713.3
## + Pclass:HaveCabin   2    651.5   713.5
## + Pclass:AgeGroup    12   631.6   713.6
## + AgeGroup:HaveCabin 6    643.7   713.7
## + Title:HaveCabin    3    649.8   713.8
## + Pclass:ParGroup    3    649.9   713.9
## - Sex:Embarked       2    660.0   714.0
## + Pclass:SibGroup    4    649.0   715.0
## + Pclass:Title       4    650.3   716.3
## + Pclass:Embarked    4    651.1   717.1
## + Embarked:Title     4    651.1   717.1
## + Title:ParGroup     4    651.2   717.2
## + Sex:AgeGroup       6    648.9   718.9
## + AgeGroup:SibGroup  10   641.7   719.7
## + AgeGroup:ParGroup  8    649.1   723.1
## - Sex:SibGroup       2    670.5   724.5
## + Embarked:AgeGroup  12   645.6   727.6
## - Pclass:Sex         2    682.4   736.4
## + Title:SibGroup     4  12182.8 12248.8
## + Title:AgeGroup     12  22563.3 22645.3

# train a model with the features stepwise selected
model.glm <- glm(Survived ~ Pclass + Sex + Embarked + Title + AgeGroup + SibGroup +

```

```

    ParGroup + HaveCabin + Pclass:Sex + Sex:SibGroup + Sex:Embarked + Embarked:ParGroup,
    data=train.df,
    family ="binomial")
summary(model.glm)

```

```

##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Embarked + Title + AgeGroup +
##     SibGroup + ParGroup + HaveCabin + Pclass:Sex + Sex:SibGroup +
##     Sex:Embarked + Embarked:ParGroup, family = "binomial", data = train.df)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8768  -0.5156  -0.4121   0.3137   2.5778
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      39.2872   2730.4427   0.014  0.98852
## Pclass2          -0.4702     0.8212  -0.573  0.56692
## Pclass3          -3.1542     0.7219  -4.369 1.25e-05 ***
## Sexmale          -19.8254  2664.3652  -0.007  0.99406
## EmbarkedQ         1.0162     0.7317   1.389  0.16489
## EmbarkedS        -0.7530     0.5774  -1.304  0.19220
## TitleMiss        -34.0362  2730.4426  -0.012  0.99005
## TitleMr          -18.4822   597.0554  -0.031  0.97530
## TitleMrs         -33.0352  2730.4426  -0.012  0.99035
## AgeGroup7-14.5    -2.1368     0.8796  -2.429  0.01512 *
## AgeGroup14.5-21   -1.6789     0.6998  -2.399  0.01644 *
## AgeGroup21-28     -1.6126     0.7246  -2.225  0.02605 *
## AgeGroup28-35     -1.6244     0.7435  -2.185  0.02891 *
## AgeGroup35-50     -2.0853     0.7639  -2.730  0.00634 **
## AgeGroup50-       -2.5923     0.8418  -3.080  0.00207 **
## SibGroup(0,2]     -0.5442     0.3995  -1.362  0.17319
## SibGroup(2,8]     -1.2339     0.7350  -1.679  0.09322 .
## ParGroup(0,3]     -0.1330     0.5264  -0.253  0.80053
## ParGroup(3,9]     -1.7556     1.1481  -1.529  0.12621
## HaveCabin1         0.7601     0.3916   1.941  0.05226 .
## Pclass2:Sexmale   -0.9340     0.8691  -1.075  0.28251
## Pclass3:Sexmale    2.1054     0.7069   2.978  0.00290 **
## Sexmale:SibGroup(0,2]  0.6762     0.5109   1.323  0.18568
## Sexmale:SibGroup(2,8] -18.5710   597.0567  -0.031  0.97519
## Sexmale:EmbarkedQ  -1.7711     0.9883  -1.792  0.07313 .
## Sexmale:EmbarkedS   0.4840     0.6198   0.781  0.43488
## EmbarkedQ:ParGroup(0,3] -17.4531  1172.2527  -0.015  0.98812
## EmbarkedS:ParGroup(0,3] -0.4591     0.6517  -0.704  0.48113
## EmbarkedQ:ParGroup(3,9] -17.8393  3956.1805  -0.005  0.99640
## EmbarkedS:ParGroup(3,9]      NA         NA      NA      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  651.56  on 862  degrees of freedom

```

```

## AIC: 709.56
##
## Number of Fisher Scoring iterations: 16
# goodness of fit test
hl <- hoslem.test(model.glm$y, fitted(model.glm), g=10)
hl # p-value=0.94, good fit

##
## Hosmer and Lemeshow goodness of fit (GOF) test
##
## data: model.glm$y, fitted(model.glm)
## X-squared = 2.8692, df = 8, p-value = 0.9423
# evaluate model by cross validation
m_glm <- train(Survived ~ Pclass + Sex + Embarked + Title + AgeGroup + SibGroup +
  ParGroup + HaveCabin + Pclass:Sex + Sex:SibGroup + Sex:Embarked + Embarked:ParGroup, data=train.df,
  method = "glm",
  family = "binomial",
  metric = "Accuracy",
  trControl = ctrl,
  tuneLength = 5)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

```

```
m_glm
```

```
## Generalized Linear Model
##
## 891 samples
## 8 predictor
## 2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 801, 803, 802, 802, 802, 802, ...
## Resampling results:
##
## Accuracy Kappa
## 0.8385362 0.6441233
```

```
# look at feature importance
imp.glm <- varImp(m_glm, scale = FALSE)
imp.glm
```

```
## glm variable importance
##
## only 20 most important variables shown (out of 28)
##
## Overall
## Pclass3 4.3692
## `AgeGroup50-` 3.0796
## `Pclass3:Sexmale` 2.9785
## `AgeGroup35-50` 2.7296
## `AgeGroup7-14.5` 2.4294
## `AgeGroup14.5-21` 2.3991
## `AgeGroup21-28` 2.2255
## `AgeGroup28-35` 2.1848
## HaveCabin1 1.9410
## `Sexmale:EmbarkedQ` 1.7920
## `SibGroup(2,8)` 1.6787
## `ParGroup(3,9)` 1.5292
## EmbarkedQ 1.3888
## `SibGroup(0,2)` 1.3620
## `Sexmale:SibGroup(0,2)` 1.3235
## EmbarkedS 1.3041
## `Pclass2:Sexmale` 1.0747
## `Sexmale:EmbarkedS` 0.7809
## `EmbarkedS:ParGroup(0,3)` 0.7045
## Pclass2 0.5726
```

```
# accuracy
train.acc[4] <- mean(predict(m_glm,train.df)==train.df$Survived)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
train.acc[4]
```

```
## [1] 0.8507295
```

```
cv.acc[4] <- m_glm$results$Accuracy
cv.acc[4]
```

```
## [1] 0.8385362
```

## build a svm model

```
# tune parameters
grid_svm <- expand.grid(sigma = c(.01, .015, 0.2),
                        C= c(0.7, 0.8, 0.9, 1, 1.1))
set.seed(1234)
m_svm1 <- train(Survived ~ ., data=train.df,
               method = "svmRadial",
               metric = "Accuracy",
               trControl = ctrl,
               tuneGrid = grid_svm)
m_svm1$bestTune
```

```
##      sigma      C
## 8 0.015 0.9
```

```
# Evaluate model performance by cross validation
set.seed(1234)
m_svm <- train(Survived ~ ., data=train.df,
               method = "svmRadial",
               metric = "Accuracy",
               trControl = ctrl,
               tuneGrid = m_svm1$bestTune)
```

```
# look at feature importance
imp.svm <- varImp(m_svm, scale = FALSE)
imp.svm
```

```
## ROC curve variable importance
```

```
##
##              Importance
## Sex           0.7669
## Pclass        0.6814
## FareGroup      0.6573
## HaveCabin      0.6369
## PartySize      0.6173
## FamilySize     0.5866
## Embarked       0.5744
## ParGroup       0.5623
## Title          0.5464
## SibGroup       0.5446
## AgeGroup       0.5317
```

```
# accuracy
train.acc[5] <- mean(predict(m_svm,train.df)==train.df$Survived)
train.acc[5]
```

```
## [1] 0.8338945
```

```
cv.acc[5] <- m_svm$results$Accuracy
cv.acc[5]
```

```
## [1] 0.8316553
```

## Summarize the performance of the 5 models

```
model.name<- c("gbm", "randomForest", "xgboost","GLM","SVM")
result <- data.frame(model.name, train.acc, cv.acc)
result
```

```
##      model.name train.acc   cv.acc
## 1          gbm 0.8439955 0.8395080
## 2 randomForest 0.8664422 0.8294337
## 3          xgboost 0.8799102 0.8338268
## 4           GLM 0.8507295 0.8385362
## 5           SVM 0.8338945 0.8316553
```

## Step 4: Making prediction —

The accuracy on test dataset is 0.80861, which gets me in the top 10% teams in the Titanic competition using only one model.

## Use the gbm model to make predictions

```
# use the gbm model to make predictions
prediction.gbm <- predict(m_gbm, test.df)
table(prediction.gbm)
```

```
## prediction.gbm
##      0      1
## 271 147
```

```
# Write out a CSV file for submission to Kaggle
submit.gbm <- data.frame(PassengerId = 892:1309, Survived = prediction.gbm)
write.csv(submit.gbm, file = "titanic_zlqgbm.csv", row.names = FALSE) #0.80861
```