

Titanic Forever (Prediction Model for Survival with 84% accuracy)

Kshitiz Sirohi

January 7, 2019

Introduction

Titanic dataset is famous for building a prediction model; whosoever have started his/her journey into the field of Machine Learning would know that. The dataset has 1309 observations and 12 variables. The prediction that has to be made is that “given certain set of variables, predict that a passenger would have survived or died”. Since we can only have two values as output, this is a classification problem. Let's get started.

Loading packages

```
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

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

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

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.5.2
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

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

Loading Data

```
setwd("C:/Users/ksiro/Documents/")
t<- read.csv("train.csv",sep=";",header=TRUE)

te <- read.csv("test.csv",sep=";",header=TRUE)
te$Survived <- NA
df<-rbind(t,te)
df <- as.data.frame(df)
```

Understanding Data

This is how it looks after we load the data and combine them by rows.

```
summary(df)
```

```
## PassengerId      Survived  Pclass
## Min.   :  1   Min.   :0.0000   Min.   :1.000
## 1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000
## Median : 655   Median :0.0000   Median :3.000
## Mean   : 655   Mean   :0.3838   Mean   :2.295
## 3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
## Max.   :1309   Max.   :1.0000   Max.   :3.000
##                NA's   :418
##
##                Name      Sex      Age
## Connolly, Miss. Kate    :  2   female:466   Min.   : 0.17
## Kelly, Mr. James       :  2   male  :843   1st Qu.:21.00
## Abbing, Mr. Anthony    :  1                      Median :28.00
## Abbott, Mr. Rossmore Edward :  1                      Mean   :29.88
## Abbott, Mrs. Stanton (Rosa Hunt):  1                      3rd Qu.:39.00
## Abelson, Mr. Samuel    :  1                      Max.   :80.00
## (Other)                :1301                     NA's   :263
## SibSp      Parch      Ticket      Fare
## Min.   :0.0000   Min.   :0.000   CA. 2343: 11   Min.   : 0.000
## 1st Qu.:0.0000   1st Qu.:0.000   1601      :  8   1st Qu.: 7.896
## Median :0.0000   Median :0.000   CA 2144   :  8   Median :14.454
## Mean   :0.4989   Mean   :0.385   3101295   :  7   Mean   :33.295
## 3rd Qu.:1.0000   3rd Qu.:0.000   347077    :  7   3rd Qu.:31.275
## Max.   :8.0000   Max.   :9.000   347082    :  7   Max.   :512.329
##                (Other) :1261   NA's   :1
## Cabin      Embarked
##           :1014    :  2
## C23 C25 C27 :  6   C:270
## B57 B59 B63 B66:  5   Q:123
## G6          :  5   S:914
## B96 B98     :  4
## C22 C26     :  4
## (Other)     : 271
```

This is how top 6 values looks like in "df" data frame

```
head(df)
```

```
## PassengerId Survived Pclass
## 1          1         0      3
## 2          2         1      1
## 3          3         1      3
## 4          4         1      1
## 5          5         0      3
## 6          6         0      3
##
##                               Name      Sex Age SibSp
## 1                               Braund, Mr. Owen Harris    male  22     1
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
## 3                               Heikkinen, Miss. Laina female  26     0
## 4 Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1
## 5                               Allen, Mr. William Henry    male  35     0
## 6                               Moran, Mr. James          male  NA     0
## Parch      Ticket      Fare Cabin Embarked
## 1     0      A/5 21171  7.2500      S
## 2     0      PC 17599 71.2833    C85      C
## 3     0 STON/O2. 3101282  7.9250      S
## 4     0      113803 53.1000   C123      S
## 5     0      373450  8.0500      S
## 6     0      330877  8.4583      Q
```

This is how last 6 values looks like in “df” data frame

```
tail(df)
```

```
## PassengerId Survived Pclass                               Name      Sex
## 1304        1304         NA      3 Henriksson, Miss. Jenny Lovisa female
## 1305        1305         NA      3 Spector, Mr. Woolf      male
## 1306        1306         NA      1 Oliva y Ocana, Dona. Fermina female
## 1307        1307         NA      3 Saether, Mr. Simon Sivertsen    male
## 1308        1308         NA      3 Ware, Mr. Frederick      male
## 1309        1309         NA      3 Peter, Master. Michael J    male
## Age SibSp Parch      Ticket      Fare Cabin Embarked
## 1304 28.0     0     0      347086  7.7750      S
## 1305  NA     0     0      A.5. 3236  8.0500      S
## 1306 39.0     0     0      PC 17758 108.9000   C105      C
## 1307 38.5     0     0 SOTON/O.Q. 3101262  7.2500      S
## 1308  NA     0     0      359309  8.0500      S
## 1309  NA     1     1      2668  22.3583      C
```

If we check the 11th column , we can see that there are a lot of missing values. So it makes sense to remove the whole column. So, after removing the 11th coulmn we can see down below how many missing values each coulumn in the table have.

```
df <- df[, -11] ##removing the 11th column
```

```
m <- as.data.frame(matrix(ncol=1,nrow=1)) ##creating an empty data frame
for (i in c(1:11)) { m[i]<- sum(is.na(df[,i])) } ## storing the values of
```

```
number of missing values in data frame  
m
```

```
##   V1  V2 V3 V4 V5  V6 V7 V8 V9 V10 V11  
## 1  0 418  0  0  0 263  0  0  0  1  0
```

Since we know how many NAs we have in each column, we now need to impute those which actual values. In the summary below, we can check the mean, median, min ,max of Age variable.

```
summary(df$Age)
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##   0.17  21.00   28.00   29.88  39.00   80.00   263
```

After imputing we can see the change below.

```
df[which(is.na(df$Age)),6] <- mean(df$Age,na.rm=TRUE)  
summary(df$Age)
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##   0.17  22.00   29.88   29.88  35.00   80.00
```

Observation

- column 6 has 263 NAs, so we need to impute some value. It can be mode, median or mean depending upon the situation. In our case, I have imputed mean because the mode and median are almost similar so it did not matter which one I use.
- Column 2 also have 418 NAs but that is what we have to predict, so we do not bother filling it in.
- I removed the cabin column since it had so many missing values.

Let's also impute values in Fare variable. We can see the summary of fare down below. It has one NA, so after finding the index of that NA I have imputed the value.

```
summary(df$Fare)
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##   0.000   7.896  14.454  33.295  31.275  512.329     1
```

```
which(is.na(df$Fare)) ## indexes are
```

```
## [1] 1044
```

```
df$Fare[1044] <- median(df$Fare,na.rm=TRUE)
```

The reason to impute median is that the variable fare has some outliers which is skweing the mean value higher than median, which mean mean does not represent the actual picture.

Imputing values in Embarked variable. We can see down below that one element in embarked variables is empty space. So, I have found out the index of it, that is 62 and 830 and imputed the values.

```
table(df$Embarked)

##
##      C      Q      S
##    270  123  914

which(df$Embarked=="")

## [1]  62 830

df$Embarked[c(62,830)] <- "S"
df$Embarked <- droplevels(df$Embarked)
table(df$Embarked)

##
##      C      Q      S
##    270  123  916
```

- Above we can see that most of the time the embarked had value "S", that is why it makes sense to impute "S" in empty spaces whose index number we got from "which" function and used it to impute value.
- We also had to drop the empty level, since it was a factor we only need those factor that hold some value.

We do not need passenger ID for the prediction of survival because unique ID cannot help in prediction.

```
##drop pasenger id
df <- df[, -1]
```

In name column we are given full names of the passenger, I have taken out only the title of their name so that we can predict does having a higher title or lower title had any effect on the survival. Maybe higher title are given higher preference than people that have lower title. Take a look down below how many different titles we have.

```
##separating title out of names in training
df$Name <- as.character(df$Name)
lastname <- strsplit(df$Name, ",")
a <- data.frame(matrix(nrow=1,ncol=1))
a<-as.list(a)
for (i in c(1:nrow(df))) { a[i] <- lastname[[i]][2]}
a<-as.character(a)
```

```

b <- strsplit(a[1:nrow(df)], ". ")
title <- data.frame(matrix(ncol=1,nrow=1))
for (i in c(1:nrow(df))) { title [i] <- b[[i]][1]}
title <- trimws(title)
title <- as.data.frame(title)
df <- cbind(df,title)
df$title <- as.character(df$title)
table(df$title)

```

```

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

```

Data Splitting

Now we are in the stage where have cleared the data. Now we can split them back into training and testing. I split the data in two parts, one part is “df” again, which holds training values, and another is “test” which holds test values.

```

test <- df[892: 1309,]
df <- df[1:891,] ## I am strong training as df again

```

Exploration

Pclass vs survival

Let’s check out how does class variable affects the survival of the passenger.

```

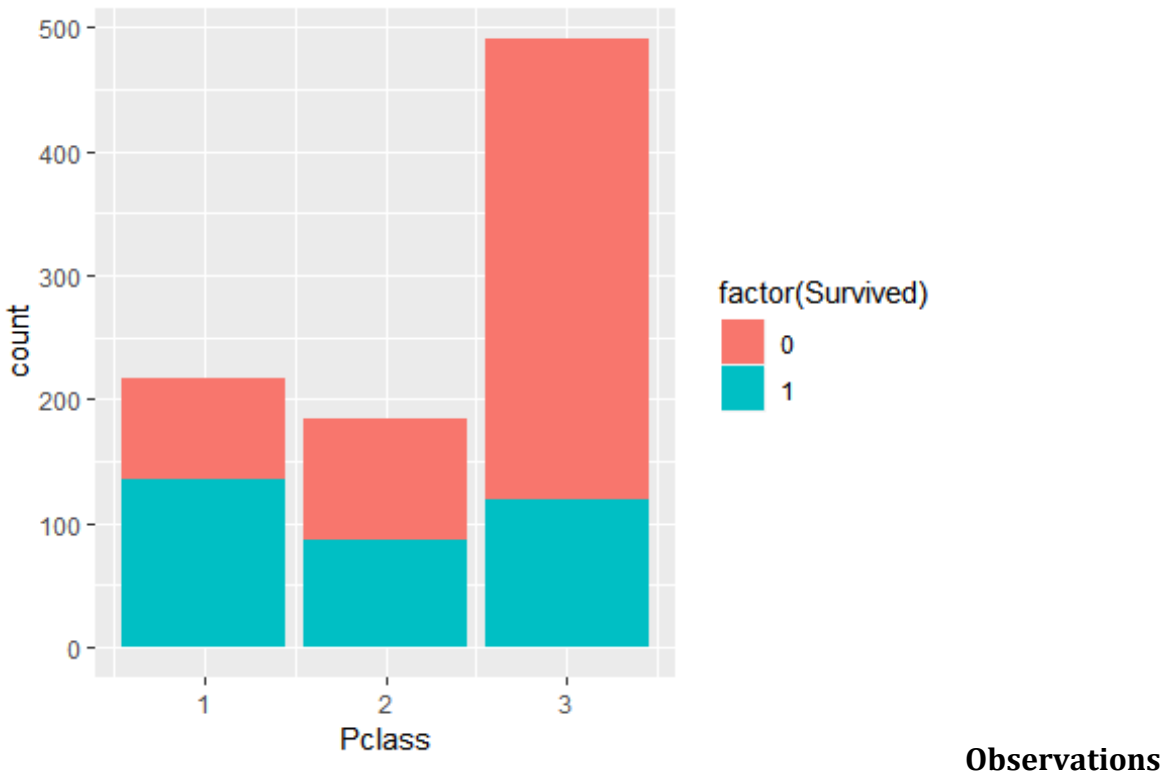
##most number of 3rd class passengers, almost half
aggregate(df$Survived,by=list(df$Pclass),FUN=mean) ##varying survival rate
depending upon the class

##  Group.1      x
## 1      1 0.6296296
## 2      2 0.4728261
## 3      3 0.2423625

ggplot(df,aes(x=Pclass,fill=factor(Survived))) + geom_histogram(stat="count")

## Warning: Ignoring unknown parameters: binwidth, bins, pad

```



- Above we see that most of the passenger are from 3rd class category on ship.
- In the plot we can see the survival rate of each one of those classes.

Also check out the percentage of people survived in each category below.

```
#we see that people with high class had better chances of survival.
percent_survive_by_class <- df %>% group_by(Pclass) %>%
  summarise(survival_rate=sum(Survived==1)*100/ (survival_rate= sum(Survived
==1)+sum(Survived==0)) )
percent_survive_by_class

## # A tibble: 3 x 2
##   Pclass survival_rate
##   <int>         <dbl>
## 1     1          63.0
## 2     2          47.3
## 3     3          24.2
```

Observations

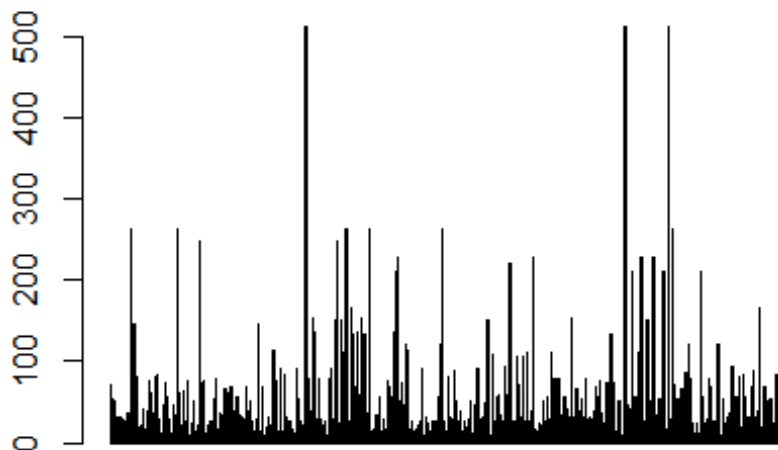
- We see that people from 1st class have higher rate of survival than people from 3rd class.

Fare vs Survival

Let's check how the fare variable affects the survival of passenger.

Here is the simple barplot of fare.

```
#survival vs fare  
barplot(df$Fare)
```



I see a few outliers

up there. Let's see the top 6 of those outliers below.

```
fare <- sort(df$Fare,decreasing = T)  
head(fare)  
  
## [1] 512.3292 512.3292 512.3292 263.0000 263.0000 263.0000  
  
top_fare <- tail(order(df$Fare),3)  
df[top_fare,]  
  
##      Survived Pclass                Name    Sex Age SibSp  
## 259         1      1      Ward, Miss. Anna female  35     0  
## 680         1      1 Cardeza, Mr. Thomas Drake Martinez  male  36     0  
## 738         1      1    Lesurer, Mr. Gustave J    male  35     0  
##      Parch  Ticket      Fare Embarked title  
## 259      0 PC 17755 512.3292         C  Miss  
## 680      1 PC 17755 512.3292         C   Mr  
## 738      0 PC 17755 512.3292         C   Mr  
  
#all 3 with highest fare survived.
```

Observations

- I saw that there are few outliers in fare column. In the barplot above it can be seen that three are above 500 and there are more that are higher than 200 but their quantity is very less in comparison to the whole dataset.
- Above we can also see that all 3 passengers who paid 500 bucks have survived.

Let's analyse more on fare.

```
fare <- fare[-c(1,2,3)]
head(fare)
```

```
## [1] 263.000 263.000 263.000 263.000 262.375 262.375
```

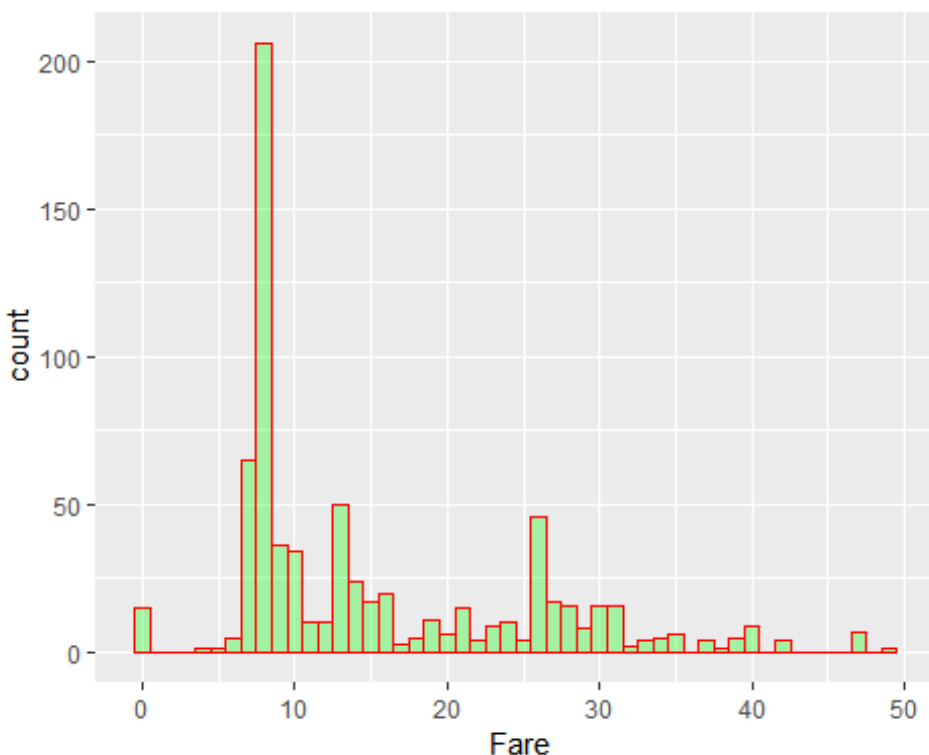
```
summary(fare)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.000   7.896  14.454   30.582  30.772 263.000
```

```
ggplot(df, aes(Fare, color=I("red"), fill=I("green"), alpha=I(0.3))) +
  geom_histogram(binwidth = 1) + xlim(NA, 50)
```

```
## Warning: Removed 160 rows containing non-finite values (stat_bin).
```

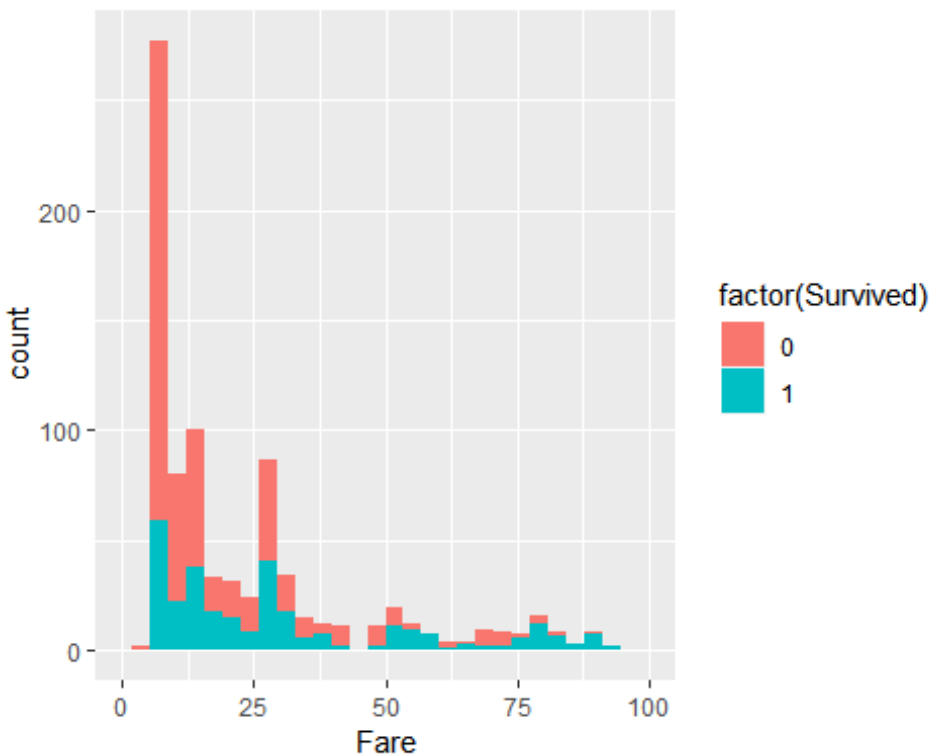
```
## Warning: Removed 1 rows containing missing values (geom_bar).
```



```
ggplot(df, aes(x=Fare, fill=factor(Survived))) + geom_histogram() + xlim(0, 100)
```

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

```
## Warning: Removed 53 rows containing non-finite values (stat_bin).  
## Warning: Removed 4 rows containing missing values (geom_bar).
```

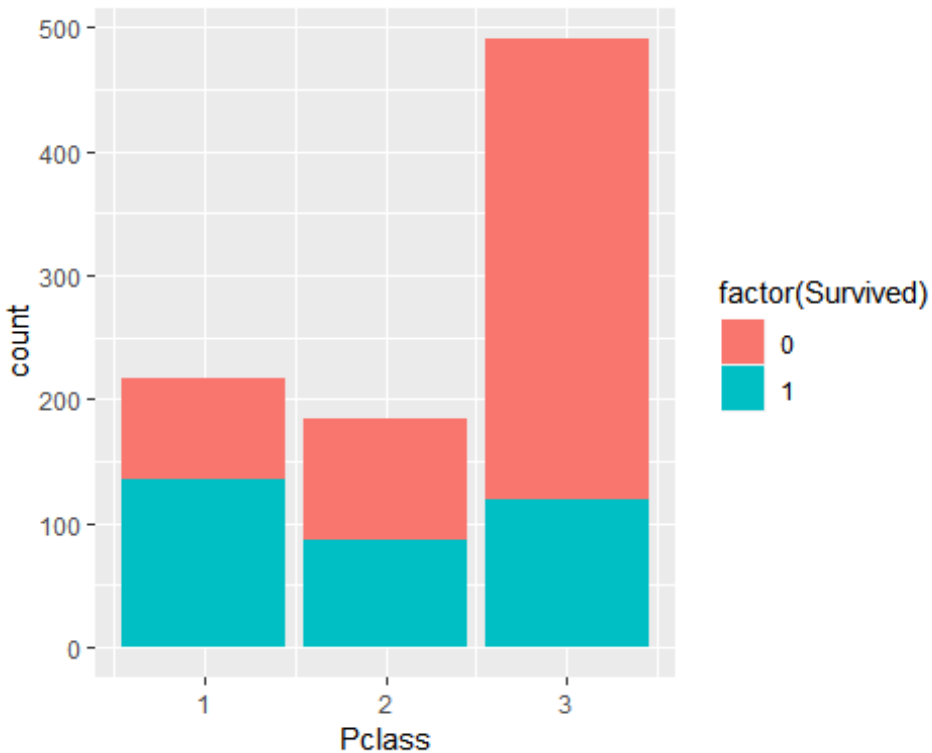


Observations

- After removing the top three outliers, we can see that the mean value has changed. Not a very significant improvement but in some cases it can be significant.
- In the one of the histogram above we can see that the most of the people paid between 5 to 30.
- In the second chart above, we can see that higher the fare amount higher is the survival. That means people who paid more were given more preference during emergency.

Survival vs class

```
r ggplot(df,aes(x=Pclass,fill=factor(Survived))) + geom_bar()
```



```
r percent_survive_by_class <- df %>% group_by(Pclass) %>%
  summarise(survival_rate=sum(Survived==1)*100/ (survival_rate= sum(Survived
==1)+sum(Survived==0)) ) percent_survive_by_class
```

##	##	Pclass	survival_rate	##	<int>	<dbl>	
## 1	1	63.0	## 2	2	47.3	## 3	3

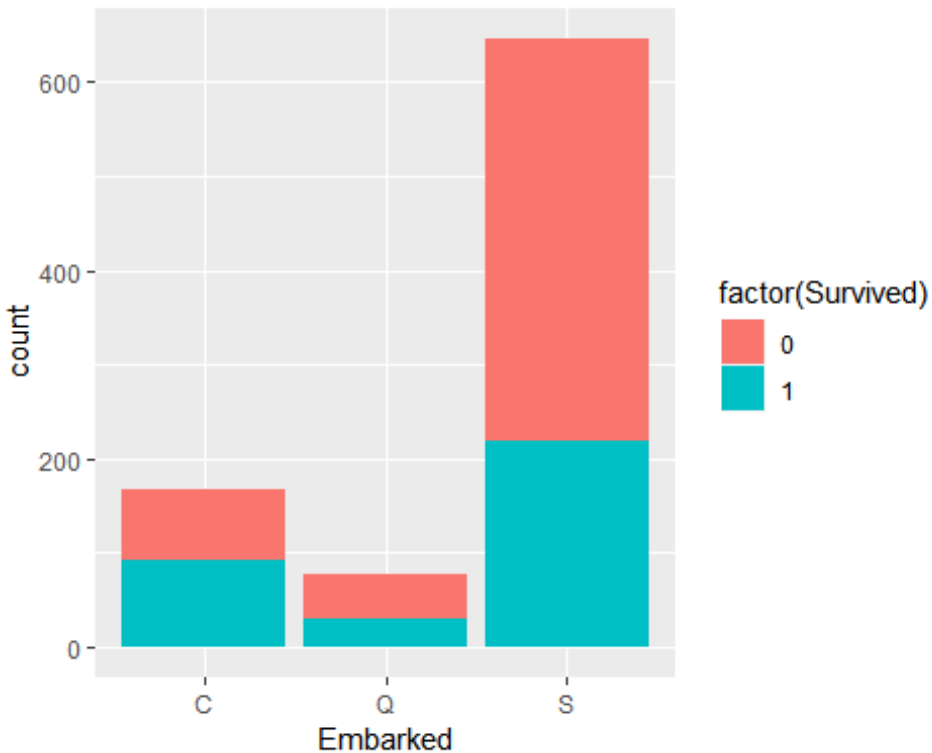
24.2 Observations

- We see that people from 1st class have higher rate of survival than people from 3rd class.

Survival vs Embarked

```
r ggplot(df,aes(x=Embarked,fill=factor(Survived))) +
  geom_histogram(stat="count")
```

Warning: Ignoring unknown parameters: binwidth, bins, pad



```
r aggregate(df$Survived,by=list(df$Embarked),FUN=mean)
```

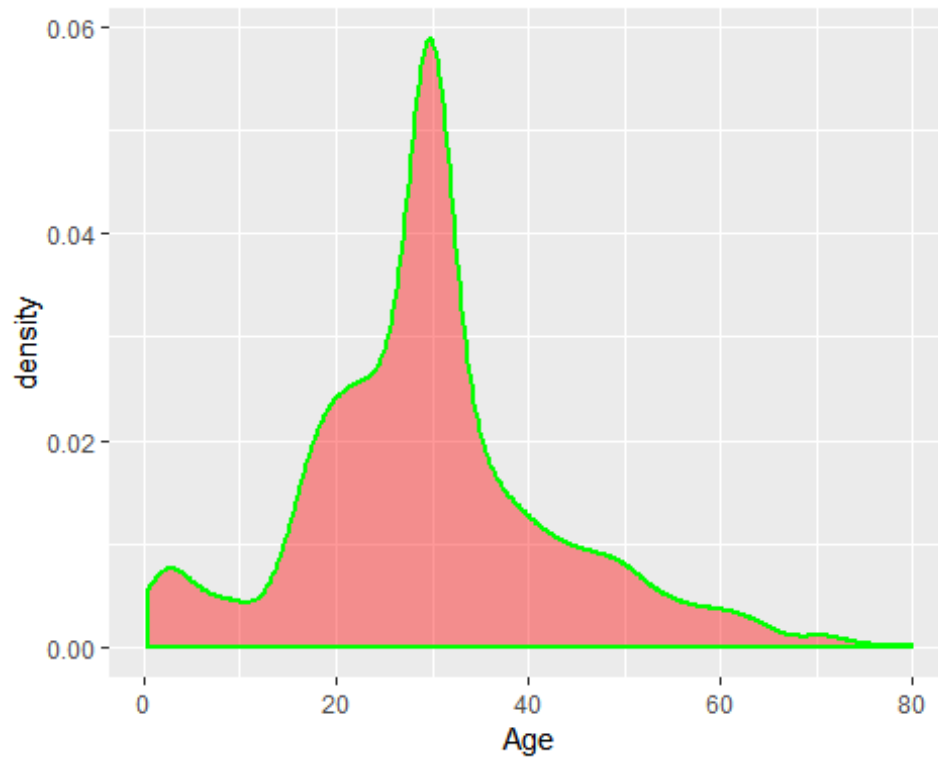
```
## Group.1      x  ## 1      C 0.5535714  ## 2      Q 0.3896104  ##
3      S 0.3390093
```

r ## there is sufficient difference in percentage, so we can use this also for prediction **Observations**

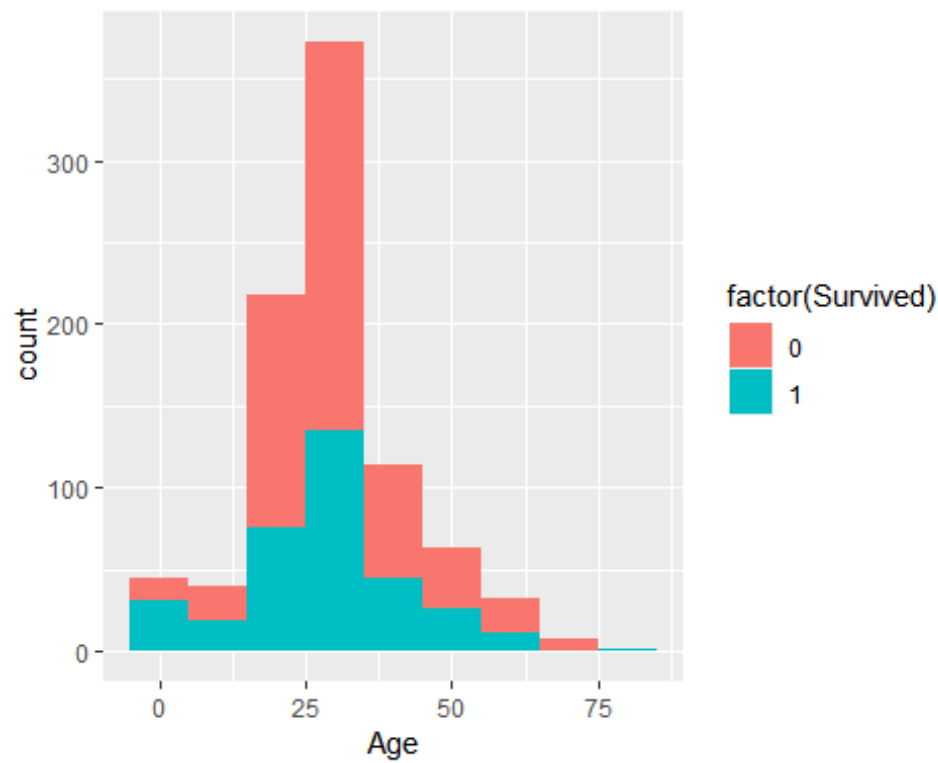
- In the table above we can see that only C has 55% survival rate. Which can also be seen in the graph given after the table above.

survival vs age Let's see how does difference in age can make a difference in the survival of a passenger.

```
ggplot(df,aes(x=Age)) + geom_density(col="green",fill="red",alpha=.4,size=1)
```



```
ggplot(df, aes(x=Age, fill=factor(Survived))) + geom_histogram(binwidth = 10)
```



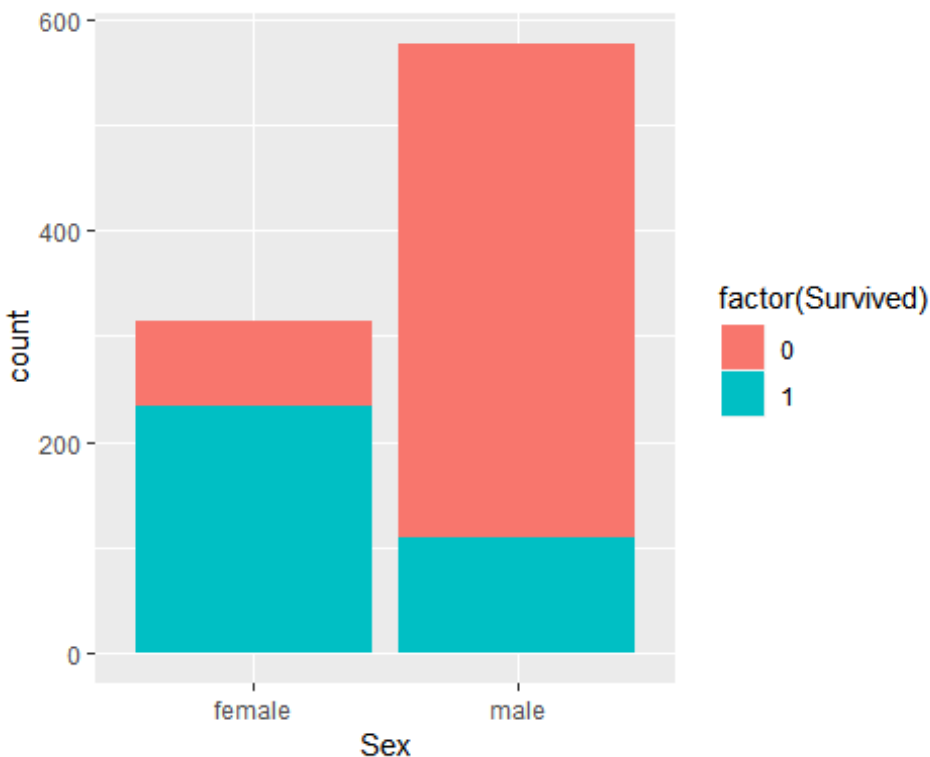
Observations

- Most number of people on the ship was around 30 years of age. It can be seen in the plot above. There is a sudden spike in the density plot at age 30.
- In the next plot we can see that as the age increased the survival rate also increased. Meaning people that were either old or child, were given more preferences than middle age people during emergency.

Sex vs survival

```
ggplot(df, aes(x=Sex, fill=factor(Survived))) + geom_histogram(stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



```
## Women had higher survival rate
```

Observations

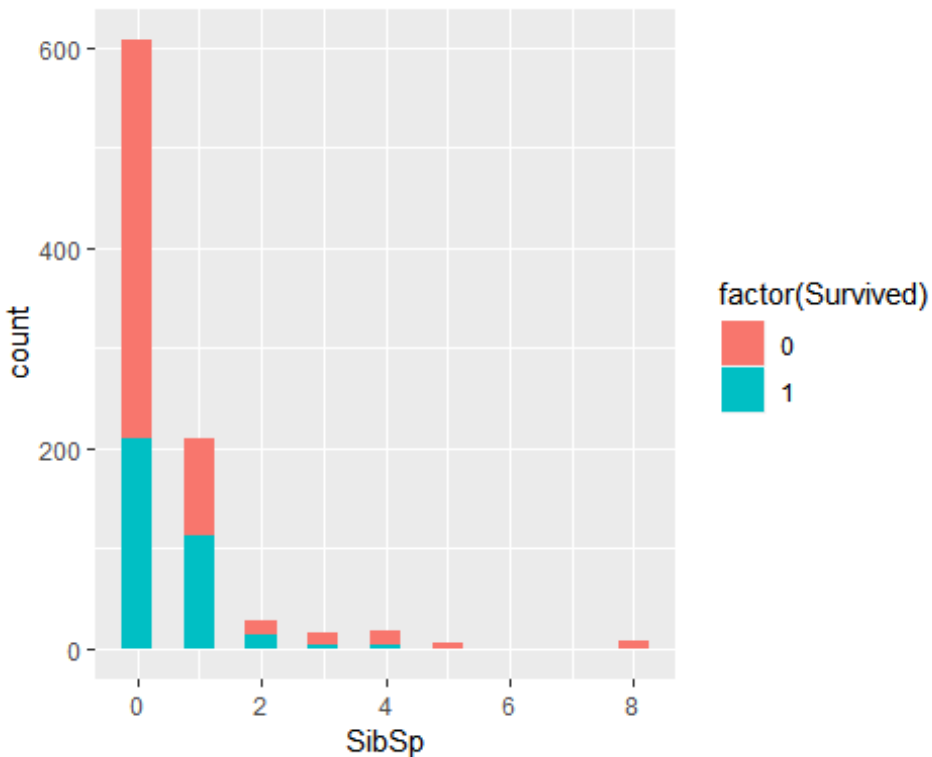
In the plot above we can see that women had higher survival rate, meaning women were given preference during emergency.

SibSp vs Survival

Can having more number of siblings effects the chances of survival?

```
ggplot(df, aes(x=SibSp, fill=factor(Survived))) + geom_bar(binwidth = .5)
```

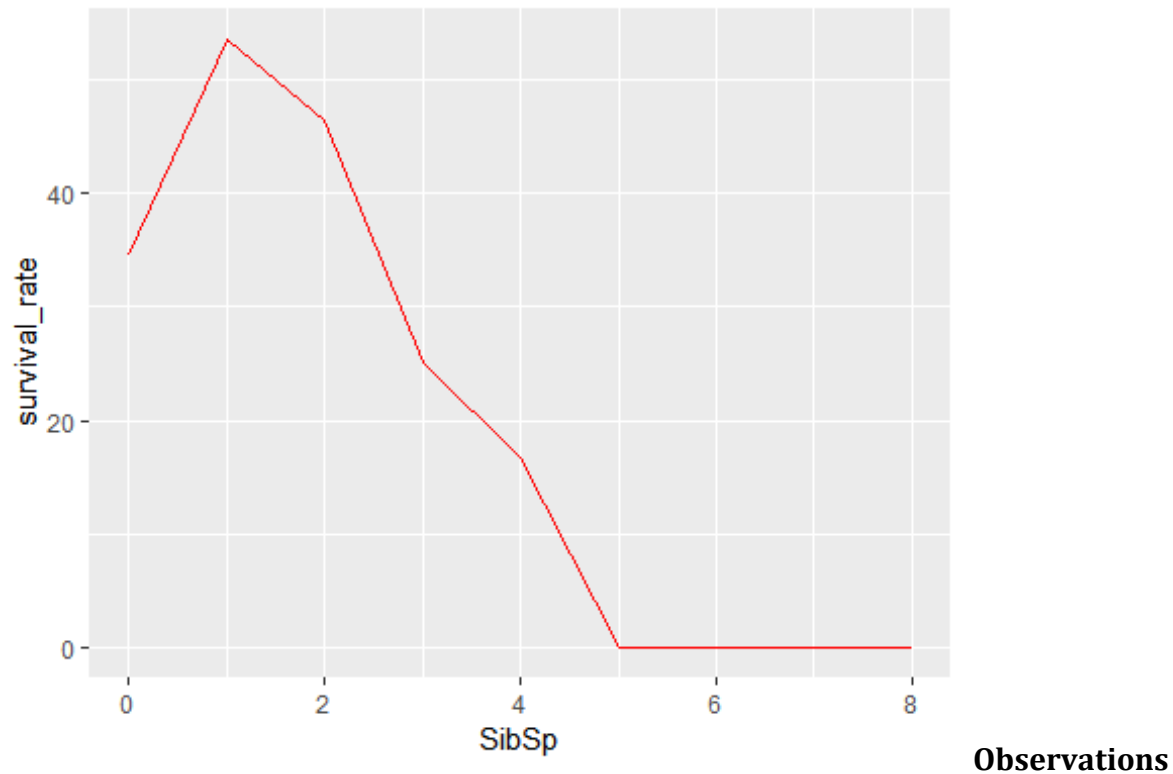
```
## Warning: `geom_bar()` no longer has a `binwidth` parameter. Please use
## `geom_histogram()` instead.
```



```
##SURVIAL RATE DECREASEED AS THE RESPONSIBILITY INCREASED.
percent <- df %>% group_by(SibSp) %>% summarise(lived=sum(Survived==1),died=
sum(Survived==0))
survival_rate <- (percent$lived)*100/(percent$lived + percent$died )
percent <- cbind(percent,survival_rate)
percent
```

```
## SibSp lived died survival_rate
## 1 0 210 398 34.53947
## 2 1 112 97 53.58852
## 3 2 13 15 46.42857
## 4 3 4 12 25.00000
## 5 4 3 15 16.66667
## 6 5 0 5 0.00000
## 7 8 0 7 0.00000
```

```
ggplot(percent,aes(x=SibSp,y=survival_rate,col=I("red")))+ geom_line()
```



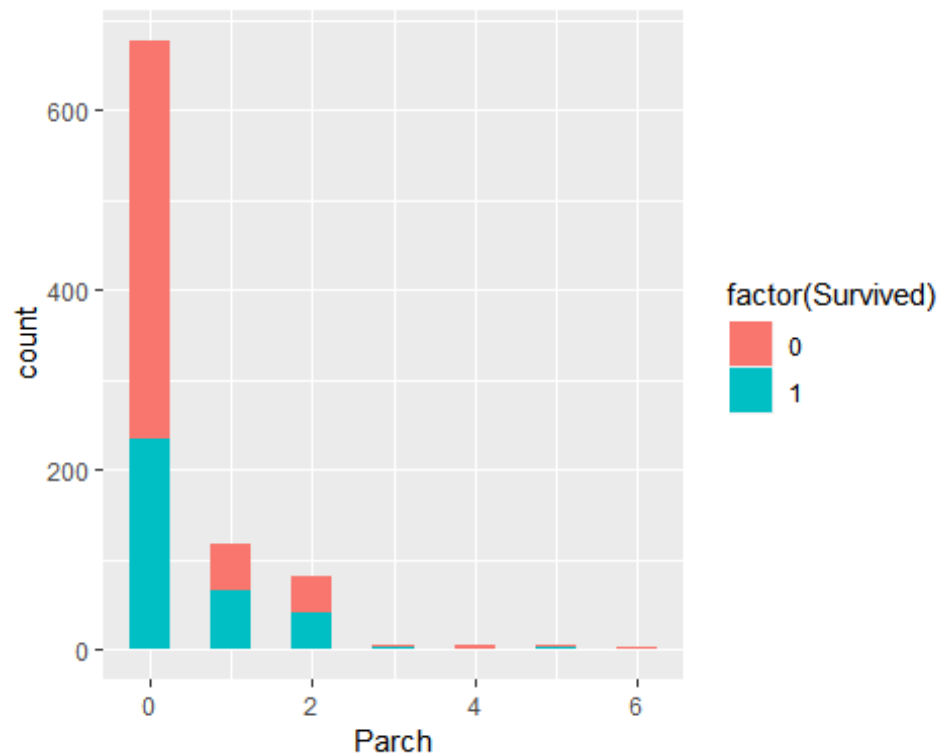
- In percent table we can see that the survival rate is decreasing as the responsibility increase, by responsibility I mean as the number of Siblings increased, in addition to spouses, the survival rate decreased.

Parch vs Survival

Can having more number of parents or children means they had more responsibility and less chances of survival.

```
#having a parent or child does not seem to effect the survival.
ggplot(df, aes(x=Parch, fill=factor(Survived))) + geom_bar(binwidth = .5)

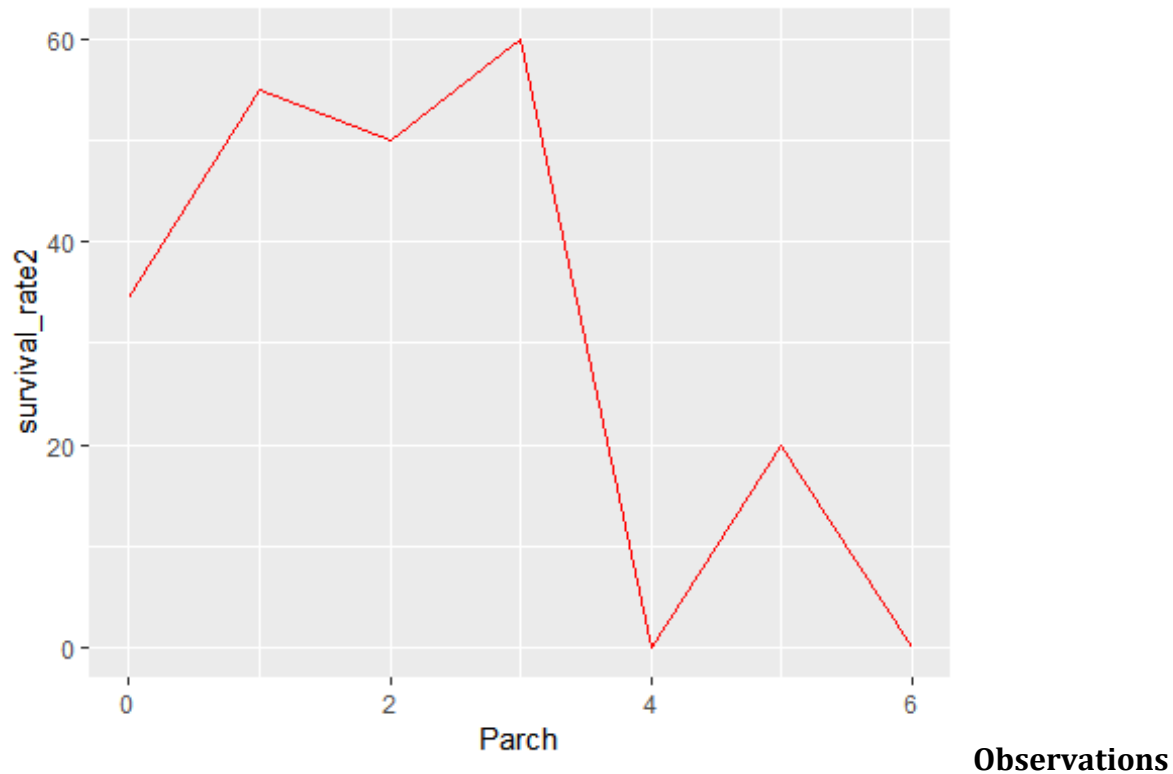
## Warning: `geom_bar()` no longer has a `binwidth` parameter. Please use
## `geom_histogram()` instead.
```

```
percent2 <- df %>% group_by(Parch) %>% summarise(lived=sum(Survived==1),died=
sum(Survived==0))
survival_rate2 <- (percent2$lived)*100/(percent2$lived + percent2$died )
percent2 <- cbind(percent2,survival_rate2)
percent2
```

##	Parch	lived	died	survival_rate2
## 1	0	233	445	34.36578
## 2	1	65	53	55.08475
## 3	2	40	40	50.00000
## 4	3	3	2	60.00000
## 5	4	0	4	0.00000
## 6	5	1	4	20.00000
## 7	6	0	1	0.00000

```
ggplot(percent2,aes(x=Parch,y=survival_rate2,col=I("red"))) + geom_line()
```



- In the percent2 table above, we can see that there is not much variation in survival rate depending upon the number of parent or child that individual have.
- But, we can get a more clear picture if we combine SibSp and Parch and name it the whole family size. So let us do that.

Introducing New Feature

Family size

Let's create a new variable called family size based on previous variables.

```
family_size <- df$SibSp + df$Parch + 1    ##family size for training
df <- cbind(df,family_size)

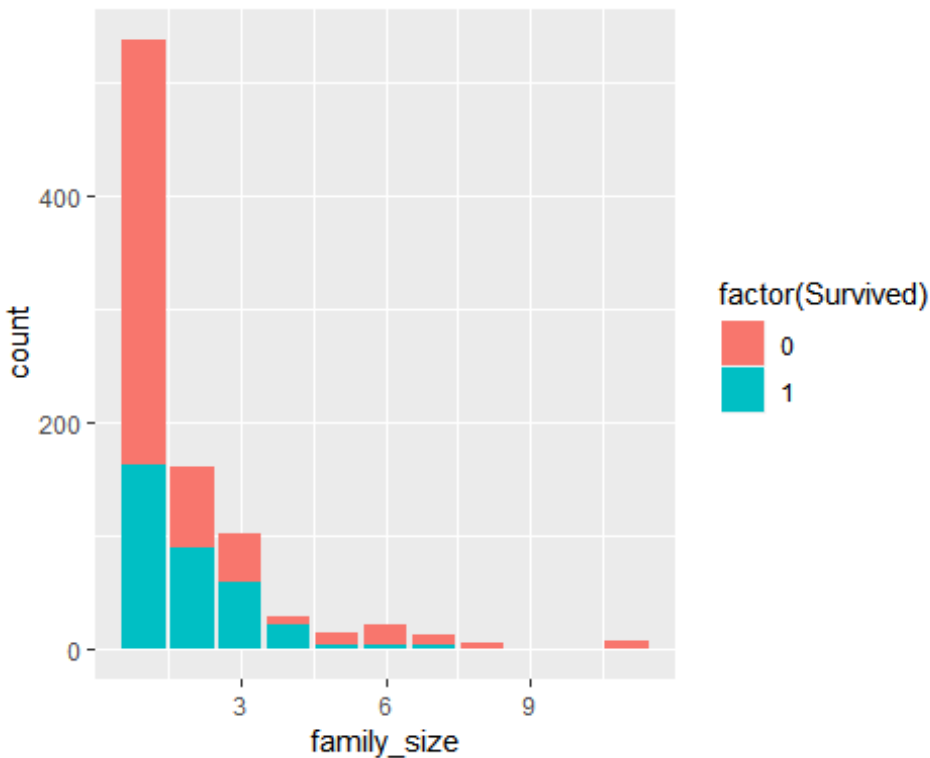
family_size <- test$SibSp + test$Parch + 1    ##family size for test
test <- cbind(test,family_size)
```

Observations

I have added the number of sibling and spouse to number of parents and childrens and 1 for the person itself. This can be a new feature called family.

Creating the variable for testing dataset too.

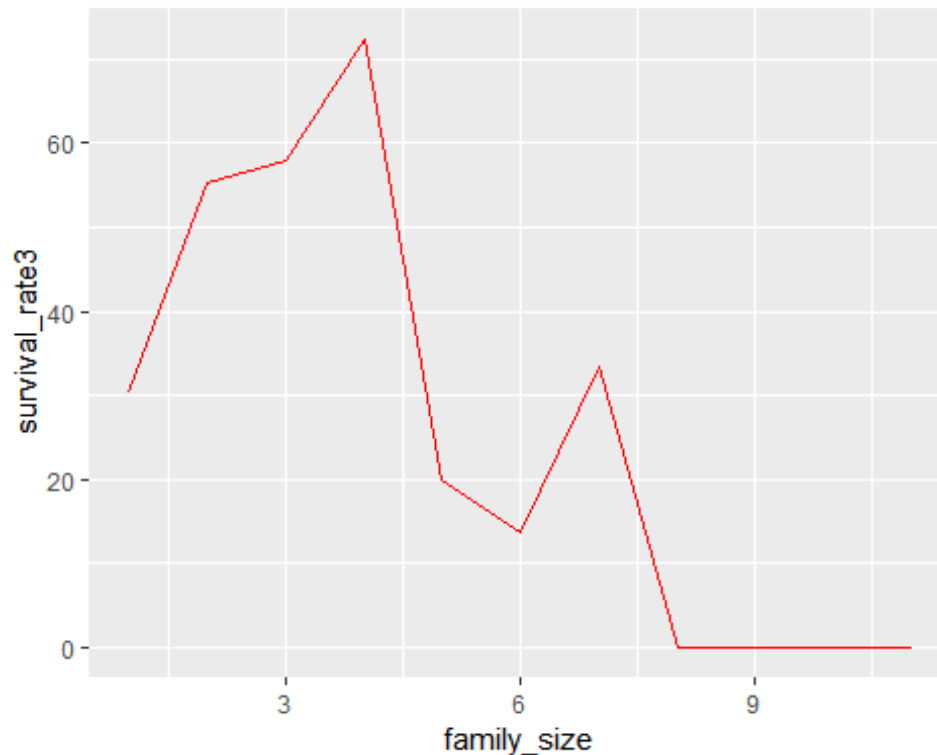
```
##family size for training set
ggplot(df,aes(x=family_size,fill=factor(Survived))) + geom_bar(stat =
"count")
```



```
percent3 <- df %>% group_by(family_size) %>%
summarise(lived=sum(Survived==1),died= sum(Survived==0))
survival_rate3 <- (percent3$lived)*100/(percent3$lived + percent3$died )
percent3 <- cbind(percent3,survival_rate3)
percent3
```

```
##  family_size lived died survival_rate3
## 1          1   163  374      30.35382
## 2          2    89   72      55.27950
## 3          3    59   43      57.84314
## 4          4    21    8      72.41379
## 5          5     3   12      20.00000
## 6          6     3   19      13.63636
## 7          7     4    8      33.33333
## 8          8     0    6       0.00000
## 9         11     0    7       0.00000
```

```
ggplot(percent3,aes(x=family_size,y=survival_rate3,col=I("red"))) +
geom_line()
```

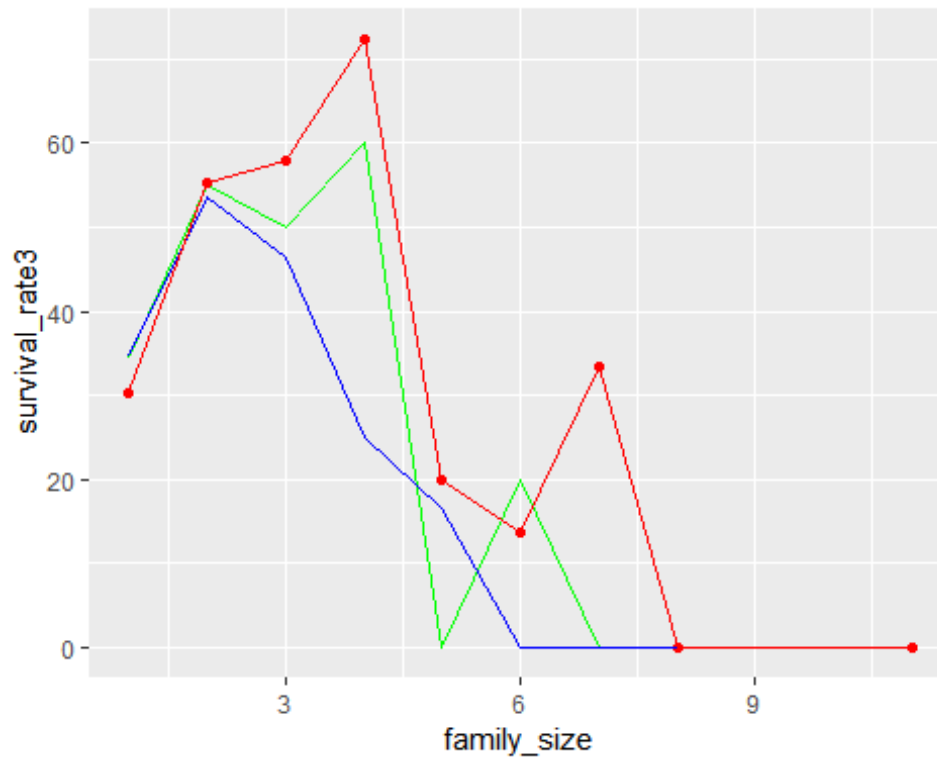


Observations *

After calculating the total family size, I have shown the survival rate in with respect to each category of family size. * In second chart I have shown the trend of survival rate based on the percentage of people survived belonging to each category.

Now to compare them all I have created a combined chart down below. * (a) percent line chart that showed SibSp * (b) percent2 line chart that showed Parch * (c) percent3 line chart that showed SibSp + Parch + 1

```
empty1 <- data.frame(matrix(c(0,0,0,0,0,0,0,0),nrow=2,ncol=4))
empty2 <- data.frame(matrix(c(0,0,0,0,0,0,0,0),nrow=2,ncol=4))
colnames(empty1) <- c("SibSp", "lived", "died", "survival_rate")
colnames(empty2) <- c("Parch", "lived", "died", "survival_rate2")
percent <- rbind(percent,empty1)
percent2 <- rbind(percent2,empty2)
dummy <- cbind(percent[,c(1,4)],percent2[,c(1,4)], percent3[,c(1,4)])
g <- ggplot(dummy,aes(x=family_size,y=survival_rate3,color=I("red"),group=2))
+ geom_line() ## after adding family
g <- g + geom_line(aes(x=family_size,y=survival_rate2,color="green")) +
geom_point(shape=16)## only parent and child
g <- g + geom_line(aes(x=family_size,y=survival_rate,color="blue")) +
geom_line() ## only spouse and siblings
g
```



Changing features

Below we can see that there are too many titles. So it is better to merge some of them.

```
## survival vs title
table(df$title)
```

```
##
##      Capt      Col      Don      Dr Jonkheer      Lady      Major      Master
##       1         2         1         7         1         1         2         40
##      Miss      Mlle      Mme      Mr      Mrs      Ms      Rev      Sir
##     182         2         1     517     125         1         6         1
##       th
##       1
```

Let's see how much percentage of people survived in each category of title.

```
survival_by_title <- df %>% group_by(title) %>%
  summarise(value=mean(Survived)*100)
survival_by_title

## # A tibble: 17 x 2
##   title      value
##   <chr>    <dbl>
## 1 Capt         0
## 2 Col        50
```

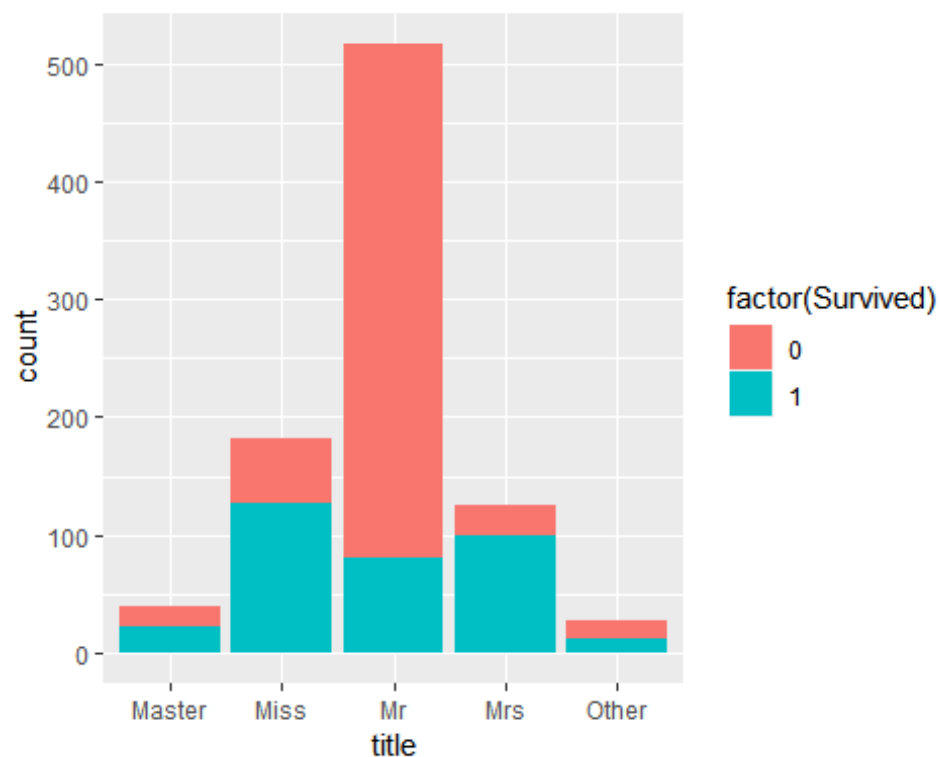
```
## 3 Don      0
## 4 Dr       42.9
## 5 Jonkheer 0
## 6 Lady     100
## 7 Major    50
## 8 Master   57.5
## 9 Miss     69.8
## 10 Mlle    100
## 11 Mme     100
## 12 Mr      15.7
## 13 Mrs     79.2
## 14 Ms      100
## 15 Rev     0
## 16 Sir     100
## 17 th      100
```

Below is the visualization of title vs survived after merging the less frequent titles.

```
df$title[df$title!="Master" & df$title != "Miss" & df$title!= "Mr" &
df$title!= "Mrs"] <- "Other"
table(df$title)
```

```
##
## Master   Miss    Mr    Mrs   Other
##      40    182   517   125    27
```

```
ggplot(df,aes(x=title,fill=factor(Survived))) + geom_bar()
```



Change title for test also.

```
test$title[test$title!="Master" & test$title != "Miss" & test$title!= "Mr" &
test$title!= "Mrs"] <- "Other"
```

Observations

- I have changed the feature named title. We can see that there are many different kind of title, then it makes sense to combine the less frequent ones together and let the more frequent ones as they are.
 - Then in the table we can see the title as well as their respective survival rate. We see that Women are given most preference during emergency with Miss having 70% survival and Mrs having 80% survival rate.
 - Men with the title Mr., meaning an average man on the ship was given least preference.
 - We can also see it in the barplot above, where each title being presented with respective survival rate in blue.
 - Changing the title for test dataset also because the training and testing dataset must have the consistency.
-

Normalization

In classification problem, we need to normalize the continuous result so that we can accommodate those continuous variables between 0 and 1. To do that, I have created a function called norm and then called fare and age variable because they were continuous. Both for testing and training dataset.

```
##normalize continuous variables
norm <- function(x){(x-min(x))/(max(x)-min(x))}

df$Age<- norm(df$Age)
df$Fare<- norm(df$Fare)
test$Age<- norm(test$Age)
test$Fare <- norm(test$Fare)

##subset only valueable columns
train <- df[,c(1,2,4,5,9,10,11,12)]
test <- test[,c(1,2,4,5,9,10,11,12)]
```

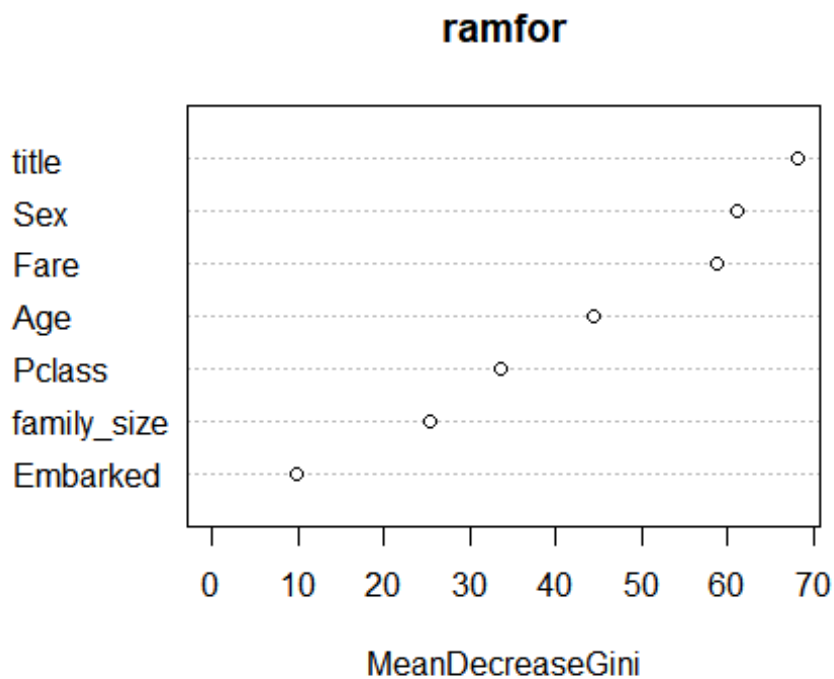
Since we know we have a classification problem at hand, it is reasonable to convert everything into factor. Both for training and testing dataset.

```
train$Pclass <- as.factor(train$Pclass)
train$title<- as.factor(train$title)
train$Embarked <- as.factor(df$Embarked)
test$Pclass <- as.factor(test$Pclass)
test$title<- as.factor(test$title)
```

Prediction

Use the random forest.

```
ramfor <- randomForest( factor(Survived) ~ Pclass + Sex + Age + Fare +  
Embarked + title + family_size, data = train)  
ramfor  
  
##  
## Call:  
## randomForest(formula = factor(Survived) ~ Pclass + Sex + Age +      Fare  
+ Embarked + title + family_size, data = train)  
##              Type of random forest: classification  
##              Number of trees: 500  
## No. of variables tried at each split: 2  
##  
##              OOB estimate of  error rate: 16.27%  
## Confusion matrix:  
##      0   1 class.error  
## 0 502  47  0.0856102  
## 1  98 244  0.2865497  
  
varImpPlot(ramfor)
```



```
pr <- predict(ramfor, test)
```

By creating the confusion matrix we can determine how many false positive and false negative we have. Moreover, we can also check the accuracy of the result.

```
confusion_matrix <- ramfor[5]
confusion_matrix <- as.data.frame(confusion_matrix)
Accuracy <- (confusion_matrix[1,1] + confusion_matrix[2,2]
)/(confusion_matrix[1,2]+confusion_matrix[2,2] +confusion_matrix[2,1]
+confusion_matrix[1,1] )
Accuracy <- round(Accuracy,2)
print(paste("Accuracy is", Accuracy*100,"%"))
## [1] "Accuracy is 84 %"
```