STA 5224: Final Project - Titanic Dataset

Huong Tran

3/10/2022

I. Project Proposal:

Objective:

This project will predict which kind of people are likely to survive in the disaster of Titanic. Multiple machine learning models will be taken in to account and the comparison of their performance will be derived.

About the dataset;

The data is obtained from the Titanic competition from Kaggle. While the test.csv and gender_submission.csv will be used for model training, the train.csv will be used to evaluate model performance.

The dependence variable is "Survived", which has value 0 or 1, indicates that the person survived after the disaster or not. The others are exploratory variables, with their meaning can be find at the website.

```
test <- read.csv2(
  "/Users/huongtran/OU /Course Work/SES 4/STA5224/Final Project 2/data/test.csv",
  header = T, sep = ","
  )

survive <- read.csv(
  "/Users/huongtran/OU /Course Work/SES 4/STA5224/Final Project 2/data/gender_submission.csv")

train <- read.csv2(
  "/Users/huongtran/OU /Course Work/SES 4/STA5224/Final Project 2/data/train.csv",
  header = T, sep = ","
  )

survive$Survived <- as.numeric(survive$Survived)
train$Survived <- as.character(train$Survived)
summary(train)
```

```
##   PassengerId      Survived             Pclass          Name
##  Min.   :  1.0   Length:891         Min.   :1.000   Length:891
##  1st Qu.:223.5   Class :character   1st Qu.:2.000   Class :character
##  Median :446.0   Mode  :character   Median :3.000   Mode  :character
##  Mean   :446.0                      Mean   :2.309
##  3rd Qu.:668.5                      3rd Qu.:3.000
##  Max.   :891.0                      Max.   :3.000
##      Sex                Age                SibSp           Parch
##  Length:891         Length:891         Min.   :0.000   Min.   :0.0000
##  Class :character   Class :character   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :0.000   Median :0.0000
##                                        Mean   :0.523   Mean   :0.3816
```

```
##                                         3rd Qu.:1.000   3rd Qu.:0.0000
##                                         Max.   :8.000   Max.   :6.0000
##     Ticket              Fare               Cabin              Embarked
##  Length:891         Length:891         Length:891         Length:891
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

```
colnames(train)
```

```
##  [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
##  [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
## [11] "Cabin"       "Embarked"
```

```
nrow(train)
```

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

II. EDA (Exploratory Data Analysis):

```
library(dplyr)
library(tidyr)
library(stringr)
library(ggplot2)
library(gmodels)
library(vcd)
```

```
colSums(is.na(train))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

At first, it seems that there is no missing value in this dataset, but in fact, there are some cells having value of double quote mark, and containing no infomation, those are considered as missing value.

```
anyDuplicated(train)
```

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

There is no duplicate rows in this dataset.

1. What about name?

Name can represent for the passenger race and ethnic, which can affect their survival chance. All of value in column *Name* are different from each other, which are not very meaningful. Therefore, I will split this column into 3 others columns: *First.name, Last.name, suffix*

```
length(unique(train$Name))
```

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

```
#train["First.Name"] <- str_split_fixed(train$Name, ", ", n = 3)[, 1 ]
train["Last.Name"] <- str_split_fixed(train$Name, " ", n = 3)[, 3]
train["Last.Name"] <- gsub("Mr.", "", train[,"Last.Name"], fixed = T)
```

```r
train["Last.Name"] <- gsub("Mrs.", "", train[,"Last.Name"], fixed = T)
train["Last.Name"] <- gsub("Miss.", "", train[,"Last.Name"], fixed = T)
train["Last.Name"] <- gsub("(", "", train[,"Last.Name"], fixed = T)
train[,"Last.Name"] <- trimws(train[,"Last.Name"], which = "left")

train["Last.Name"] <- str_split_fixed(train$Last.Name, " ", n = 2)[,1]
length(unique(train$Last.Name))
```

```
## [1] 437
```

```r
cat("The total nunber of unique last name is: ",
    length(unique(train$Last.Name)), "\n ")
```

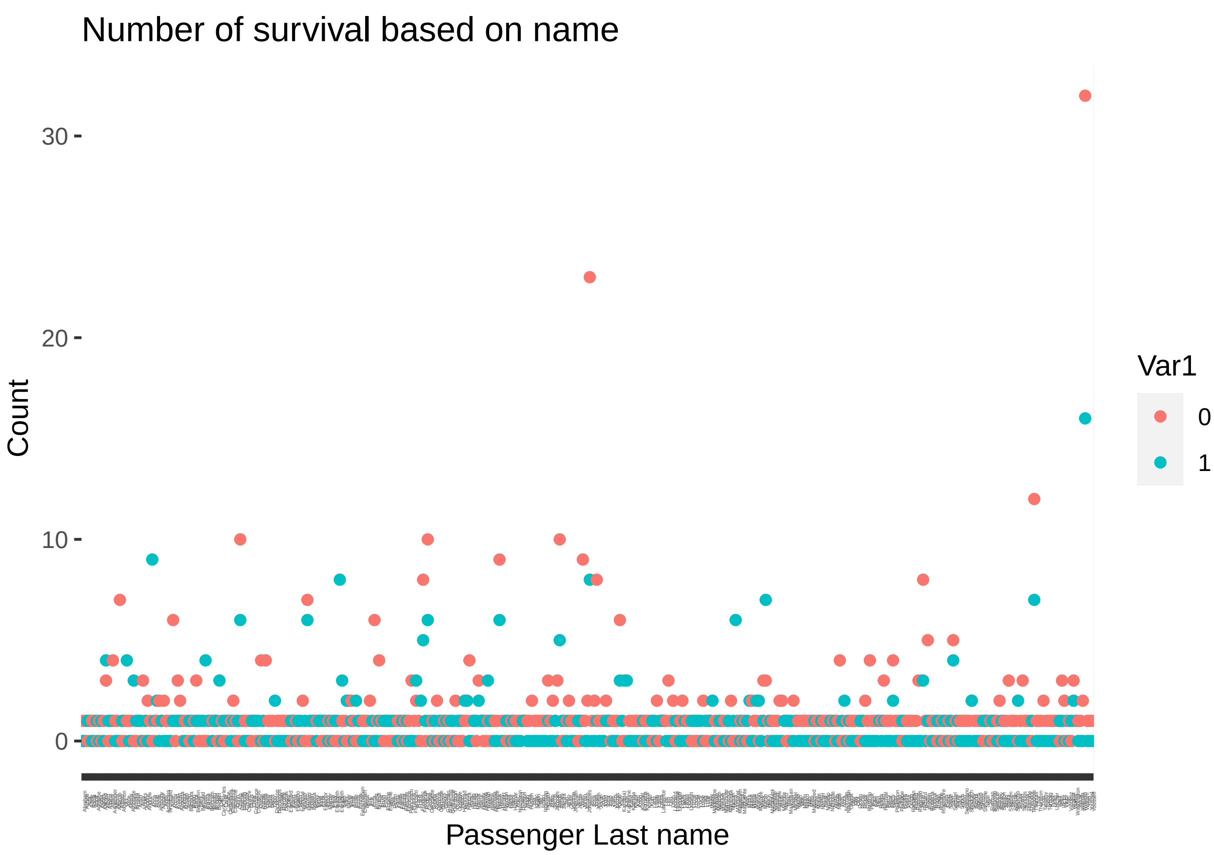
```
## The total nunber of unique last name is:  437
##
```

```r
tail(sort(unique(train$Last.Name)), 10)
```

```
##  [1] "Virginia"   "Walter"     "Washington" "Wazli"      "Wendla"
##  [6] "Wilhelm"    "William"    "Yoto"       "Youssef"    "Youssef f"
```

Looking some example of unique last name, there are many last name with "William". This mean that they might contain same information about their race and ethnic. Therefore, we can merge these last name in to one groups of "William", and so are the other last names, which contain the same information.

```r
df.name <- as.data.frame(table(train$Survived, train$new.Last.Name))
ggplot(df.name, aes(x = Var2, y = Freq, colour = Var1)) + geom_point() +
  theme(axis.text.x = element_text(angle = 90, size = 2)) +
  labs(title = "Number of survival based on name") +
  xlab("Passenger Last name") +
  ylab("Count")
```



From the plot, people with last name "Andrew", "Martin",... are likely to have more chance of surviving through the disaster.

```r
# meo <- train[order(train$new.Last.Name, train$Last.Name),]
summary(xtabs(Freq ~ Var1 + Var2, data = df.name))
```

```
## Call: xtabs(formula = Freq ~ Var1 + Var2, data = df.name)
## Number of cases in table: 891
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 515.6, df = 436, p-value = 0.005115
##  Chi-squared approximation may be incorrect
```

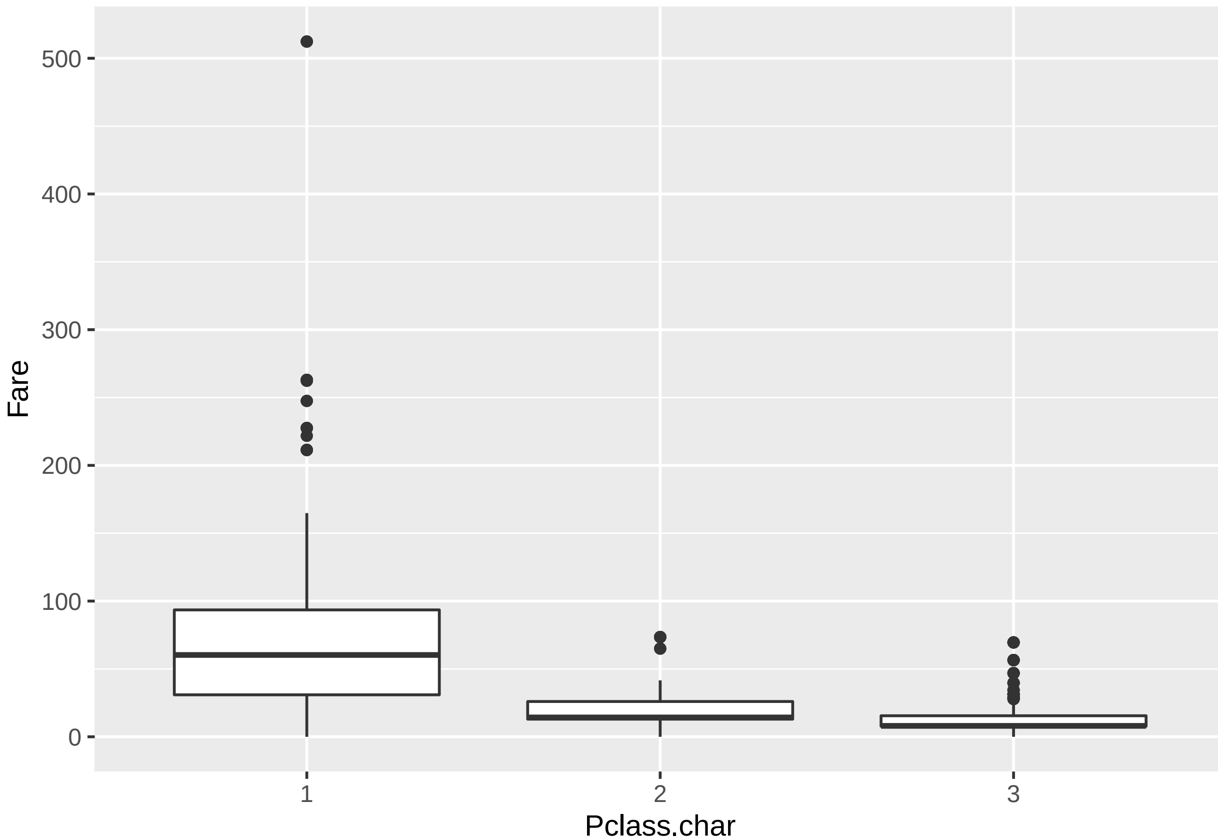
```r
df.name <- df.name[which(df.name$Freq >= 2),]
```

The F-test actually shows that last name is really helpful to predict the chance of survived. And this makes sense, since it represents for the class of passenger, which means they if they are close to the escape door or not.

2. What can we Ticket class tell us?

At first, we will look at the relation of Ticket class (*Pclass*) and Passenger fare (*fare*):

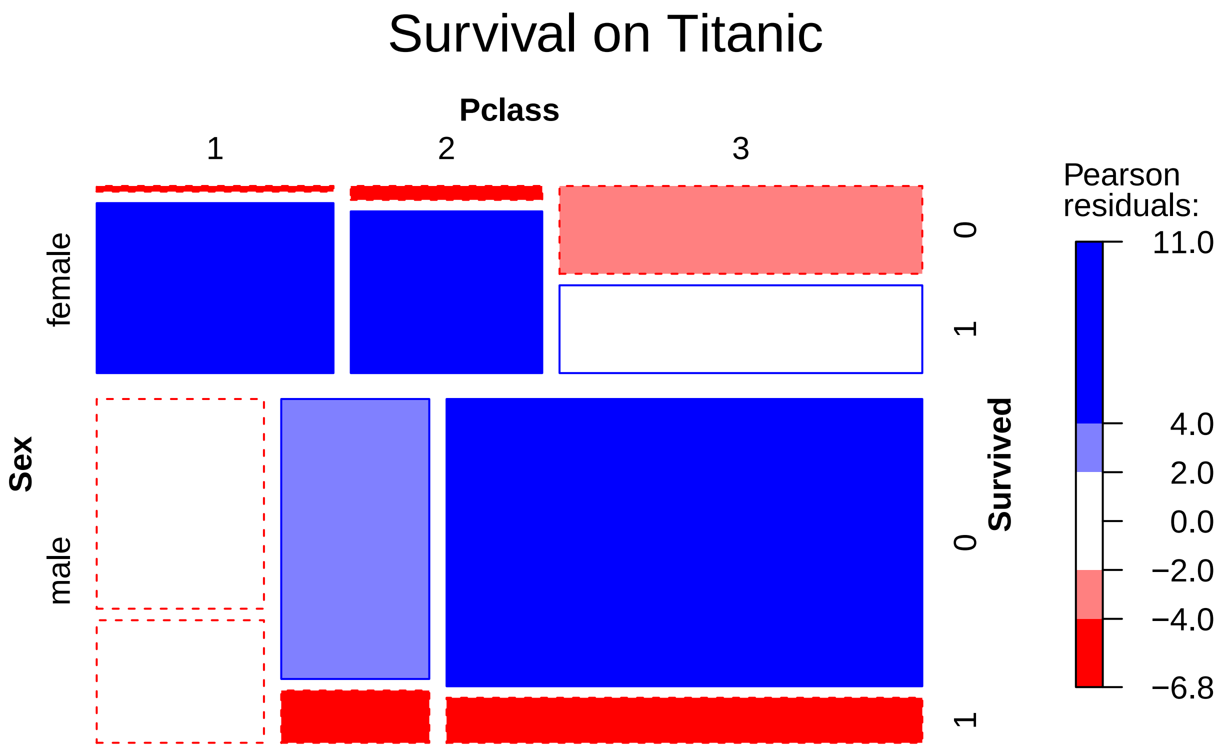
```r
train$Fare <- as.numeric(train$Fare)
train["Pclass.char"] <- as.character(train$Pclass)
ggplot(train, aes(Pclass.char, Fare)) + geom_boxplot()
```



Obviously, there is a significant positive correlation of the two variables, the upper class giving the most fare while the lower class giving the least fare.

Keep that in mind, we continue with the difference in their sex:

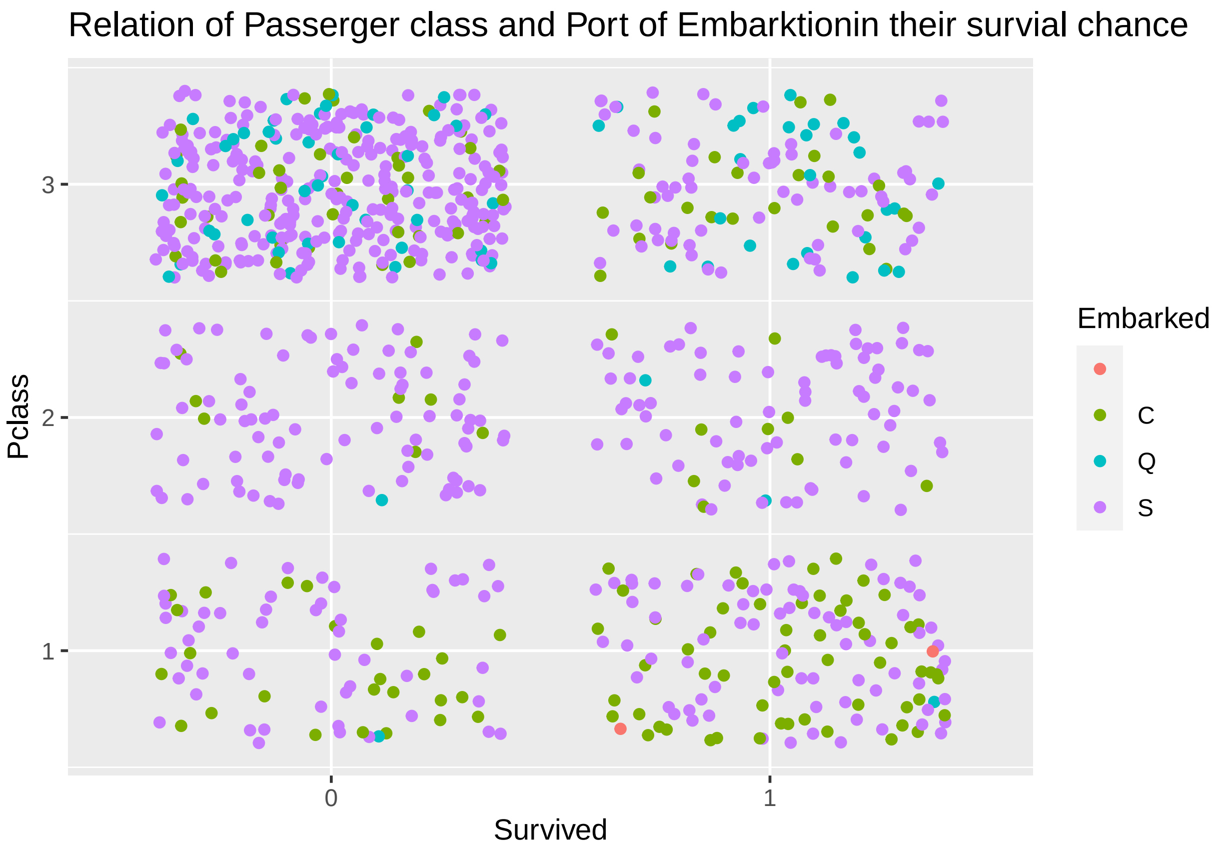
```
mosaic(~ Sex + Pclass + Survived, data = train,  main = "Survival on Titanic",
       gp = shading_Friendly, legend= T )
```



The mosaic plots shows that a women of upper class has the highest chance of survival while a men the the Lower class the lowest chance of survival. Also, although the total number of male is three times the total number of female, but male has the lower probability of survival. In fact, when the disaster hit, women and children were the first priority to go to the rescue boat.

3. Where did they embark?

```
ggplot(train, aes(Survived, Pclass, colour = Embarked)) + geom_jitter() +
  labs(title = "Relation of Passerger class and Port of Embarktionin their survial chance")
```



The majority of passenger used Southampton (S) to embark. However, Queenstown (Q) was used by the lower class ($Pclass = 3$).

What is the two red dot?

```
unique(train$Embarked)
```

```
## [1] "S" "C" "Q" ""
```

It turns out, there are 2 missing value a this column, but they are represented as the " ", that is why R could not detect any missing value. It turns out their personal information is different, but the others are the same. In fact, this cabin belongs to Mrs. George Nelson, and Miss Amelie is her maid.

```
train[which(train$Embarked == ""),]
```

```
##     PassengerId Survived Pclass                                      Name
## 62           62        1      1                       Icard, Miss. Amelie
## 830         830        1      1 Stone, Mrs. George Nelson (Martha Evelyn)
##        Sex Age SibSp Parch Ticket Fare Cabin Embarked Last.Name new.Last.Name
## 62  female  38     0     0 113572   80   B28             Amelie        Amelie
## 830 female  62     0     0 113572   80   B28             George        George
##     Pclass.char
## 62            1
## 830           1
```

Since these missing values are from cabin *B28*, let see other variables in deck B:

```
B <- train[grep("B", train$Cabin),]
CrossTable(B$Survived, B$Embarked, prop.c = T, prop.r = F, prop.t = F, prop.chisq = F)
```

```
##
```

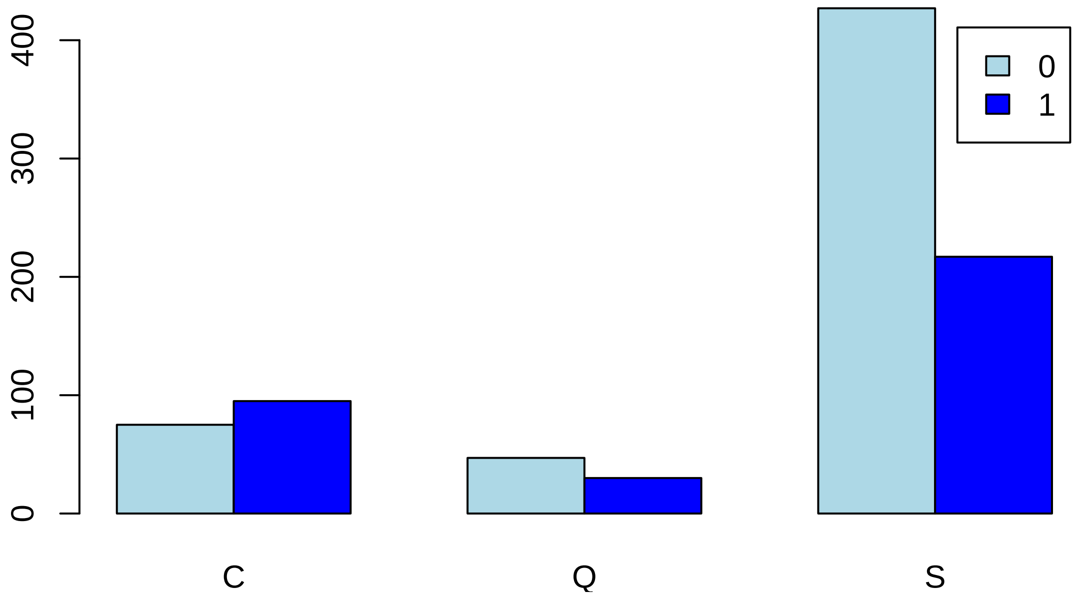
```
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  47
##
##
##              | B$Embarked
##   B$Survived |           |         C |         S | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##            0 |         0 |         5 |         7 |        12 |
##              |     0.000 |     0.227 |     0.304 |           |
## -------------|-----------|-----------|-----------|-----------|
##            1 |         2 |        17 |        16 |        35 |
##              |     1.000 |     0.773 |     0.696 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |         2 |        22 |        23 |        47 |
##              |     0.043 |     0.468 |     0.489 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
```

In general, passenger with Cabin in deck B used Cherbourg (C) and Southampton (S) as their embarkation. The percentage of survival of port Cherbourg (C) is higher, therefore, we can impute the missing data above by C.

```
train$Embarked[train$Embarked == ""] <- "C"
train[c(62, 830),]
```

```
##     PassengerId Survived Pclass                                      Name
## 62           62        1      1                       Icard, Miss. Amelie
## 830         830        1      1 Stone, Mrs. George Nelson (Martha Evelyn)
##        Sex Age SibSp Parch Ticket Fare Cabin Embarked Last.Name new.Last.Name
## 62  female  38     0     0 113572   80   B28        C    Amelie        Amelie
## 830 female  62     0     0 113572   80   B28        C    George        George
##     Pclass.char
## 62            1
## 830           1
```

```
table.embarked <- with(train, table(Survived, Embarked))
barplot(table.embarked, beside = T, legend= T, col = c("Lightblue", "blue"))
```



Only at port Cherbourge (C), the percentage of survival is higher.

4. Cabin number?

In fact, there are 162 cabins in total and their first labels was from A to G, but our data has only 148 different values. And it actually has some typo, since there are some passenger having more than one cabin number in their row. Fortunately, the values entered in that cell were from same deck. Also, the label "T" must be wrong.

```
cat("Number of unique value in Cabin: ", length(unique(train$Cabin)))
```

```
## Number of unique value in Cabin:  148
```

```
tail(sort(unique(train$Cabin)))
```

```
## [1] "F2"  "F33" "F38" "F4"  "G6"  "T"
```

```
cat("Number of missing value: " , nrow(train[which(train$Cabin == ""), ]), "\n",
    "They account for ",
    round( nrow(train[which(train$Cabin == ""), ])/nrow(train) * 100, 2),
    "% in total observation")
```

```
## Number of missing value:  687
##  They account for  77.1 % in total observation
```

Therefore, I will drop this variable in my model.

5. Age:

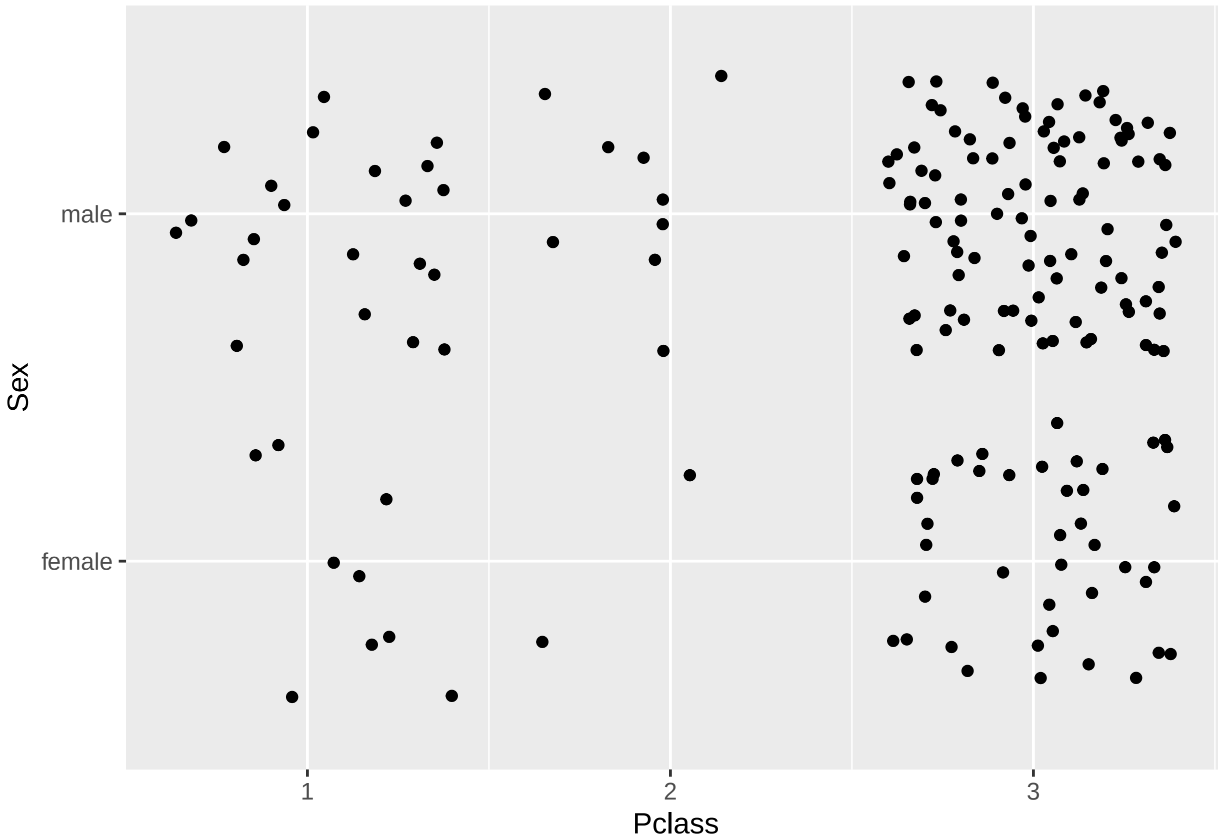
```
train[,"Age"] <- as.numeric(train[,"Age"])
cat("The number of missing value in variable Age: ",
    length(train[is.na(train$Age), "Age"]))
```

```
## The number of missing value in variable Age:  177
```

```
test[,"Age"] <- as.numeric(test[, "Age"])
length(test[is.na(test$Age), "Age"])
```

```
## [1] 86
```

```r
age.missing <- train[is.na(train$Age), ]
ggplot(age.missing, aes(Pclass, Sex)) + geom_jitter()
```

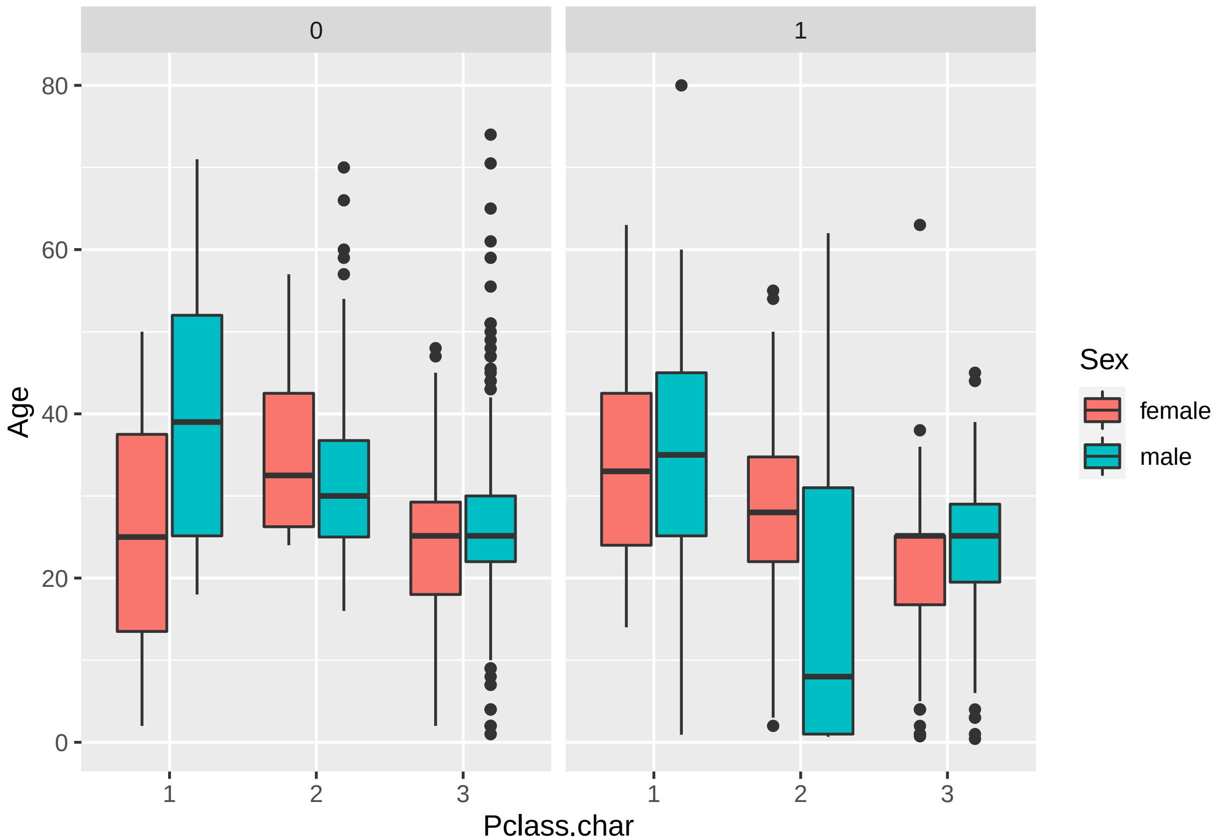


Those missing value are mainly in group 3, therefore, we can impute the missing data by mean of age in lower class.

```r
m <- mean(train[which(train$Pclass == 3), "Age"], na.rm =  T)
train$Age <- replace_na(train$Age, m)
```

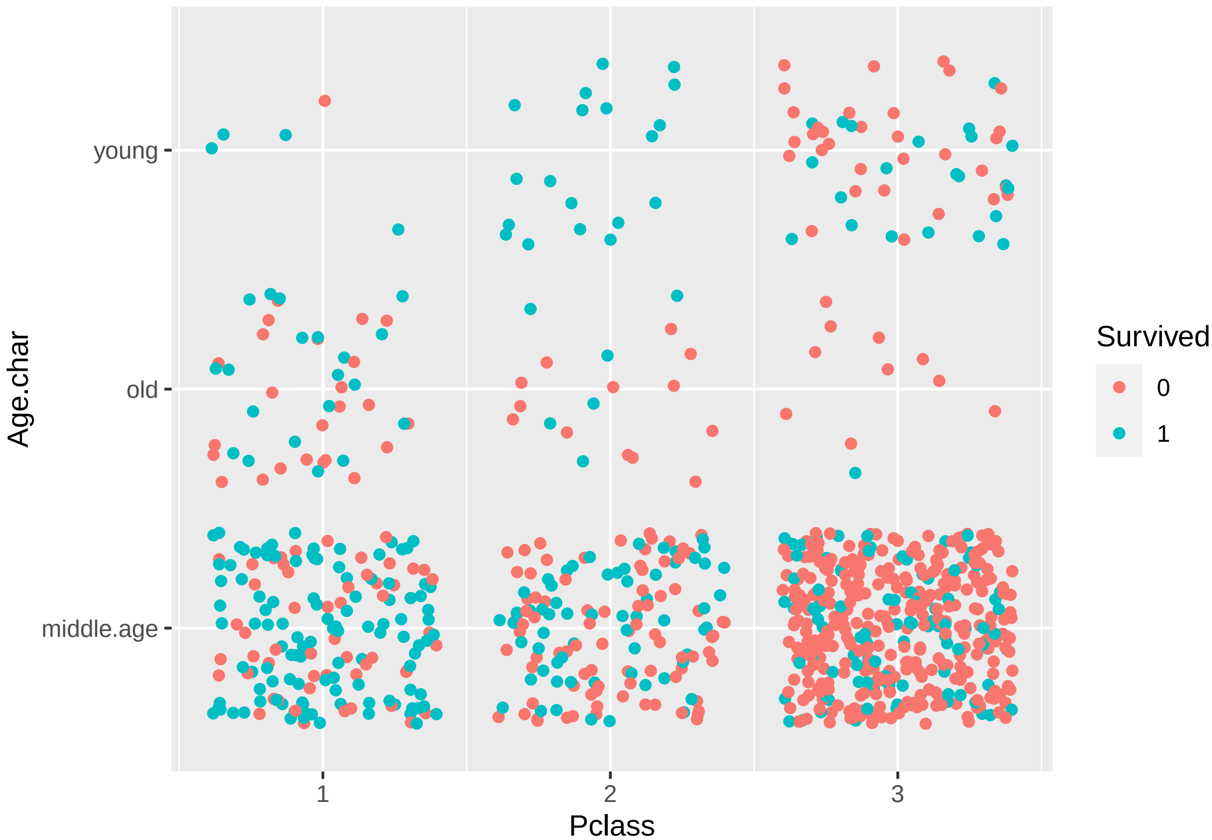
At first, we temporarily omit the missing data here.

```r
age <- train[which(train$Age != ""),]
ggplot(age, aes(Pclass.char, Age, fill = Sex)) + geom_boxplot() + facet_grid(cols = vars(Survived))
```



Surprisingly, 50% of survival female in middle class was less than 5 years old. Also, male in the middle class has more chance to survive when their age is younger than 20. Also, more than 50-year-old man is the least likely to survive through the disaster. This suggests a way to devide age into 3 smaller groups:

```
train <- train %>% mutate(Age.char = case_when(Age < 15 ~ "young",
                              Age < 50 ~ "middle.age",
                              Age < 100 ~ "old"))
ggplot(train, aes(Pclass, Age.char, colour = Survived)) + geom_jitter()
```

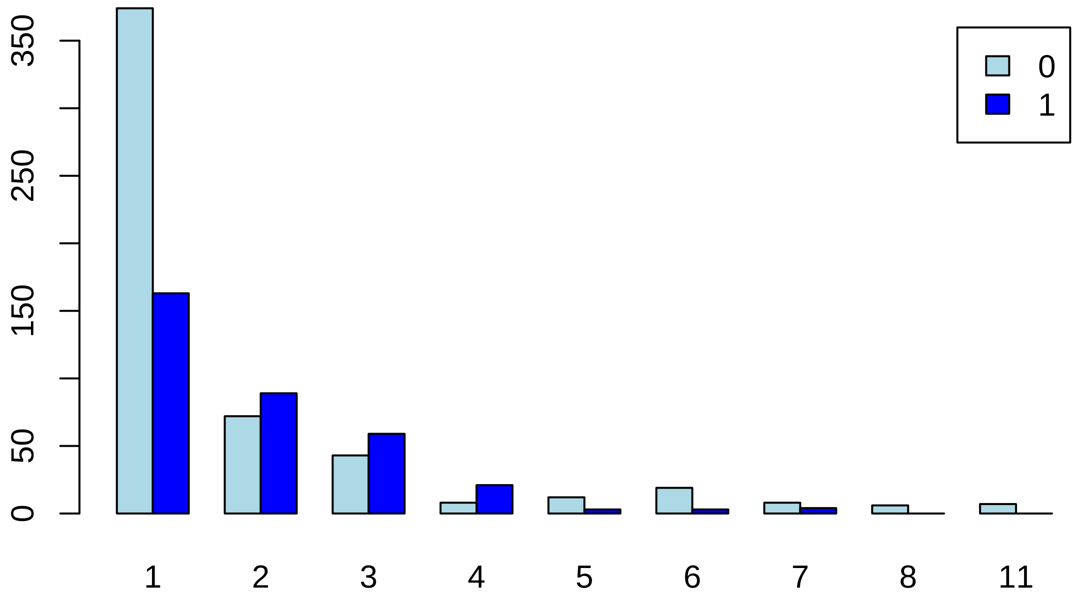


People in the first class has the highest chance to survive, especially when they are in middle age group. In contrast, most of men from lower class and middle age died during the disaster.

Family size:

At first, both variable *SibSp* and *Parch* contain information about family size, we can create a new variable as *family.size* to obtain information about passenger's family

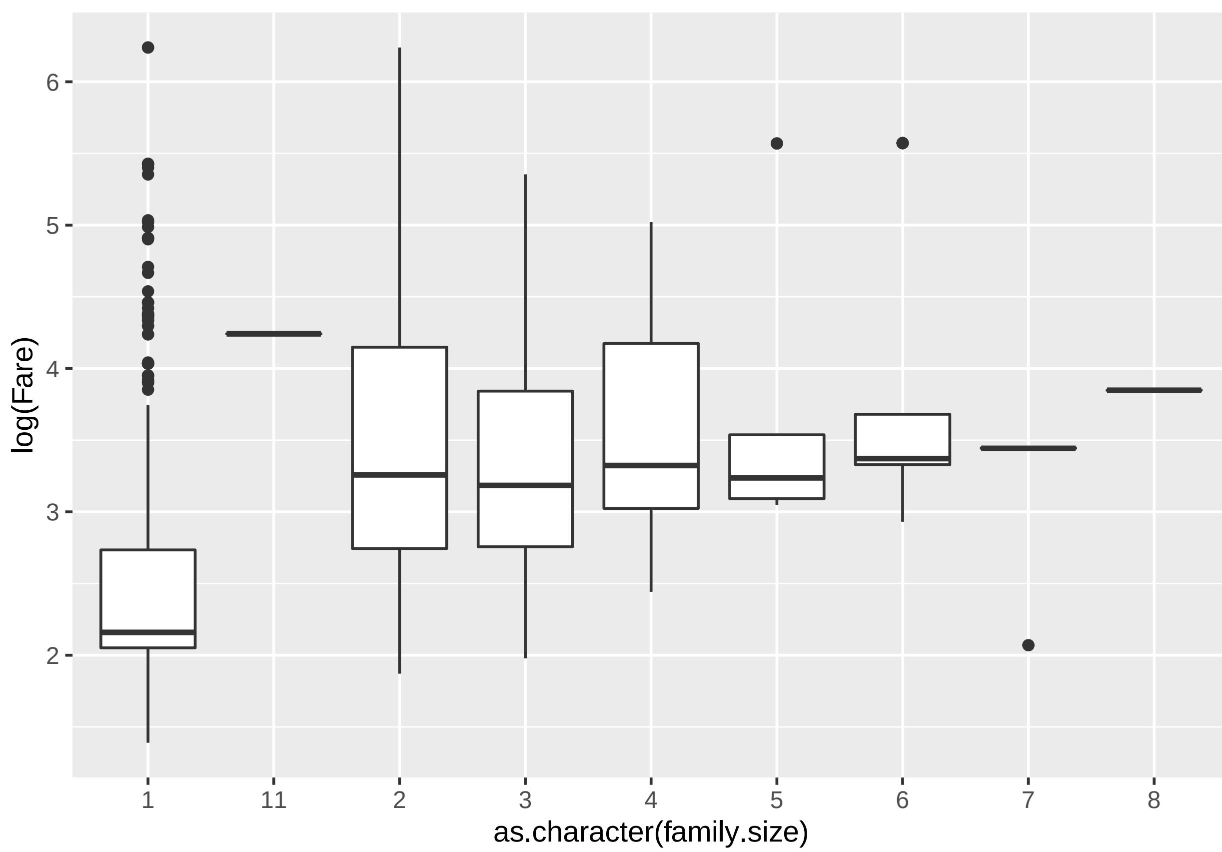
```
train <- train %>% mutate(family.size = SibSp + Parch  +  1 )
table.sib <- with(train, table(Survived, family.size))
barplot(table.sib, beside = T, legend = T, col = c("Lightblue", "Blue"))
```



Surprisingly, the number of survival in family size from 2 to 4 is higher.

```
ggplot(train, aes(as.character(family.size), log(Fare))) + geom_boxplot()
```

```
## Warning: Removed 15 rows containing non-finite values (stat_boxplot).
```



There is positive trend of ln(Fare) and size of family, i.e, the large family size the more fare that ticket they paid. Possibly, it was because the Fare was given the same for all member in a family, not individually different. Remember, when we discuss about missing value of Cabin B28, the two people there had the same fare value. Let's check this logic:

```r
fare <- train[which(train$family.size == 2),
              c("Ticket", "Fare", "family.size",
                "Pclass","Name", "Cabin")] %>% arrange(Ticket)
head(fare)
```

```
##   Ticket    Fare family.size Pclass
## 1 110813 75.2500           2      1
## 2 111361 57.9792           2      1
## 3 111361 57.9792           2      1
## 4 113505 55.0000           2      1
## 5 113505 55.0000           2      1
## 6 113509 61.9792           2      1
##                                               Name Cabin
## 1 Warren, Mrs. Frank Manley (Anna Sophia Atkinson)   D37
## 2                     Hippach, Miss. Jean Gertrude   B18
## 3  Hippach, Mrs. Louis Albert (Ida Sophia Fischer)   B18
## 4           Chibnall, Mrs. (Edith Martha Bowerman)   E33
## 5                      Bowerman, Miss. Elsie Edith   E33
## 6                    Ostby, Mr. Engelhart Cornelius   B30
```

It seems that family members had the same ticket number would have the the same Fare. Now, we will write a function to check how correct this assumption is:

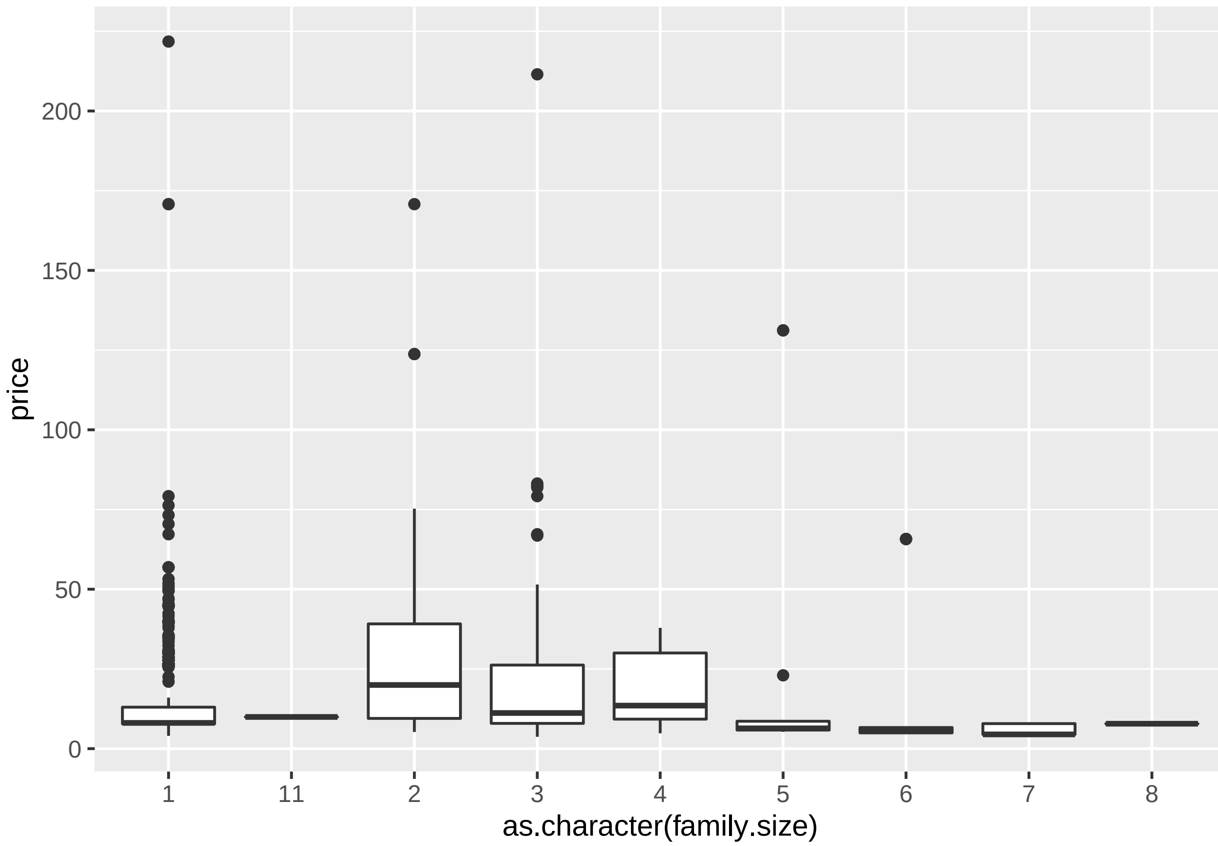
```r
train <- train %>% group_by(Ticket) %>% add_count() %>% mutate(mean.fare = mean(Fare))
nrow(train[which(train$Fare != train$mean.fare),])
```

```
## [1] 2
```

As we expected, there are only 2 cases that does not agree with our assumption. Therefore, it is actually a correlation between the family size and fare. To get rid of it, I will find the price that each person has to pay for their ticket and also fill 0 in price by the mean based on their class.

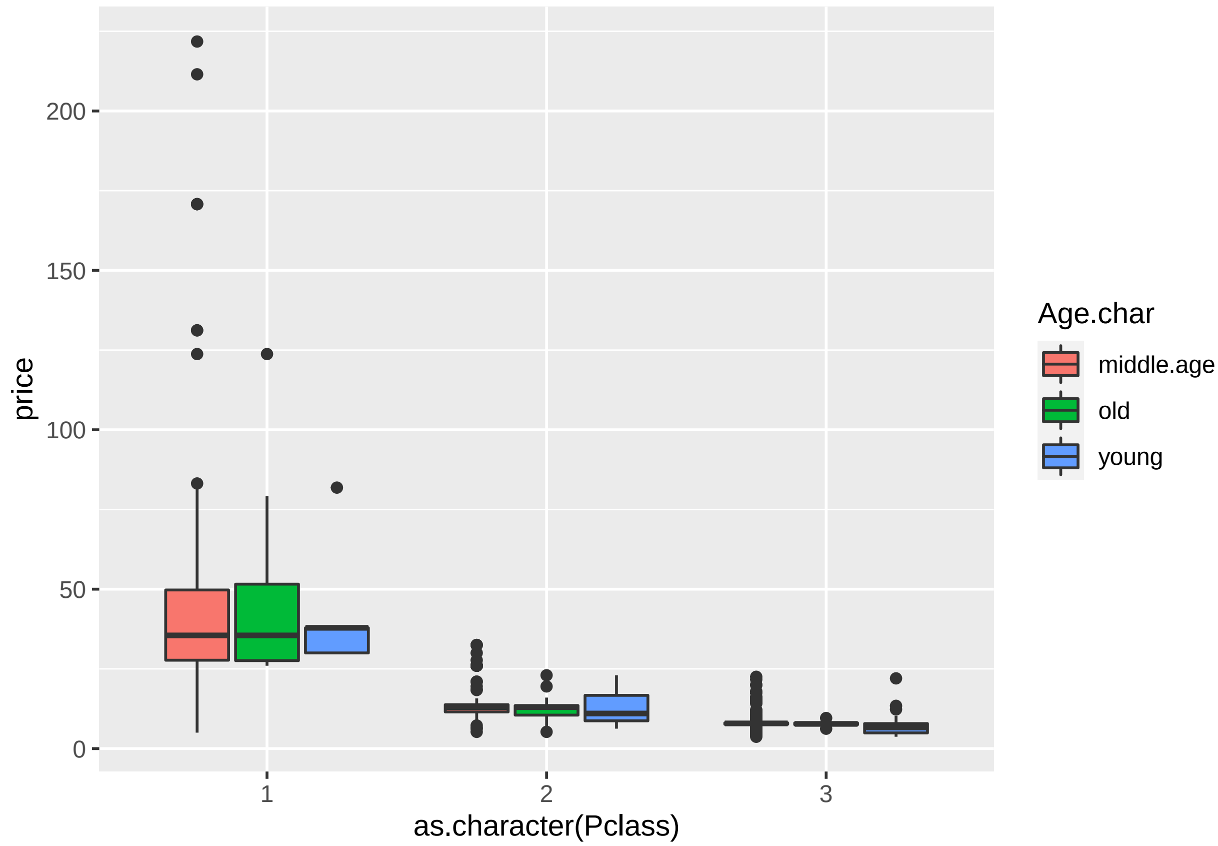
```r
train <- train %>% mutate(price = Fare / n )
for (i in 1:3){
  m <- mean(train[which(train$Pclass == i & train$price != 0), "price"]$price)
  train[which(train$Pclass == i & train$price == 0), "price"] <- m
}

ggplot(train, aes(as.character(family.size), price)) + geom_boxplot()
```



Until this point, there is no dependence of ticket fare or price. But the variable "price" is actully dependent on *Pclass*, and this happends in practice when you have to pay more to get the best service.

```
ggplot(train, aes(as.character(Pclass), price, fill = Age.char)) + geom_boxplot()
```



#

Model Preparation:

```
keep <- c("Survived", "price", "Sex", "Age", "Embarked", "family.size")

train <- train[, keep]
train$Survived <- as.numeric(train$Survived)
summary(train)
```

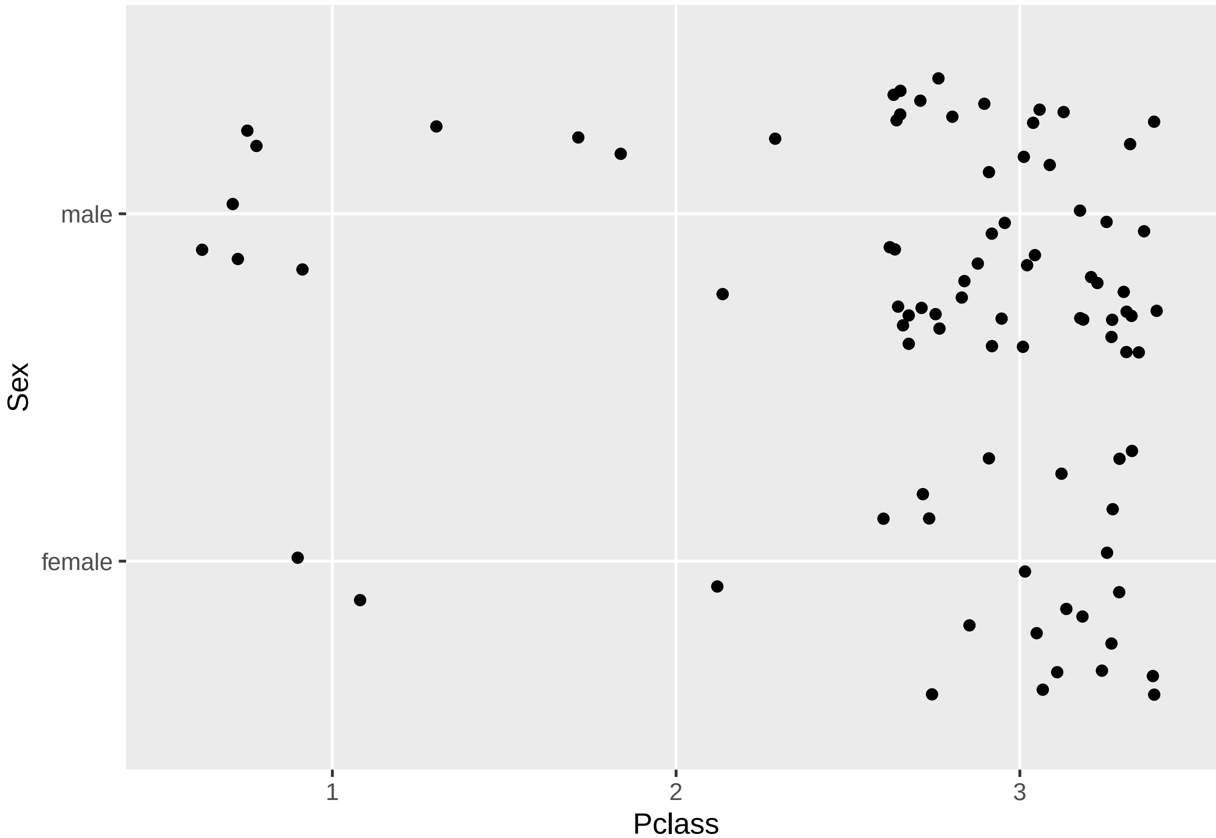
```
##     Survived          price              Sex                 Age
##  Min.   :0.0000   Min.   :  3.711   Length:891         Min.   : 0.42
##  1st Qu.:0.0000   1st Qu.:  7.775   Class :character   1st Qu.:22.00
##  Median :0.0000   Median :  9.250   Mode  :character   Median :25.14
##  Mean   :0.3838   Mean   : 18.169                      Mean   :28.79
##  3rd Qu.:1.0000   3rd Qu.: 25.927                      3rd Qu.:35.00
##  Max.   :1.0000   Max.   :221.779                      Max.   :80.00
##    Embarked          family.size
##  Length:891         Min.   : 1.000
##  Class :character   1st Qu.: 1.000
##  Mode  :character   Median : 1.000
##                     Mean   : 1.905
##                     3rd Qu.: 2.000
##                     Max.   :11.000
```

```
names(train)
```

```
## [1] "Survived"    "price"       "Sex"         "Age"         "Embarked"
## [6] "family.size"
```

The same way of data cleaning for test set:

```
test$Pclass <- as.factor(test$Pclass)
test$Fare <- as.numeric(test$Fare)
test$Age <- as.numeric(test$Age)
age.missing.test <- test[is.na(test$Age), ]
ggplot(age.missing.test, aes(Pclass, Sex)) + geom_jitter()
```




```
m.test <- mean(test[which(test$Pclass == 3), "Age"], na.rm =  T)
test$Age <- replace_na(test$Age, m.test)

test <- test %>% mutate(family.size = SibSp + Parch  +  1 )
test <- test %>% group_by(Ticket) %>% add_count() %>% mutate(mean.fare = mean(Fare))

test <- test %>% mutate(price = Fare / n )
for (i in 1:3){
  m <- mean(test[which(test$Pclass == i & test$price != 0), "price"]$price)
  test[which(test$Pclass == i & test$price == 0), "price"] <- m
}

keep.test <- keep[-1]
test <- test[, keep.test]
names(test)
```

```
## [1] "price"       "Sex"         "Age"         "Embarked"    "family.size"
```

Models:

1. Logistic Model:

```r
mod.reg <- glm(Survived ~., data = train, family = binomial)
summary(mod.reg)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0619  -0.6383  -0.5320   0.7306   2.2651
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.055257   0.362231   5.674 1.40e-08 ***
## price        0.029428   0.006129   4.801 1.58e-06 ***
## Sexmale     -2.671147   0.188260 -14.189  < 2e-16 ***
## Age         -0.020812   0.007274  -2.861 0.004219 **
## EmbarkedQ   -0.670857   0.365567  -1.835 0.066489 .
## EmbarkedS   -0.487832   0.227105  -2.148 0.031710 *
## family.size -0.221449   0.061595  -3.595 0.000324 ***
## ---
## 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:  851.35  on 884  degrees of freedom
## AIC: 865.35
##
## Number of Fisher Scoring iterations: 5
```

```r
reg <-  as.numeric(predict(mod.reg, test) > 0.5)

reg.tab <- xtabs(~ survive$Survived + reg)
reg.correct  <- (reg.tab[1,1] + reg.tab[2,2]) / sum(reg.tab)
cat("Logistic Regestion predict ", reg.correct *100, "% of correct case" )
```

```
## Logistic Regestion predict  96.64269 % of correct case
```

2. Classification:

Losgistic regression:

Classification:

Model Esampling:

Comparsions: