Titanic

Nicole Lai

04/01/2021

Contents

Introduction

```
# Loading packages
library(tidyverse)
library(visdat)
library(car)
library(MASS)
library(randomForest)
library(knitr)
```

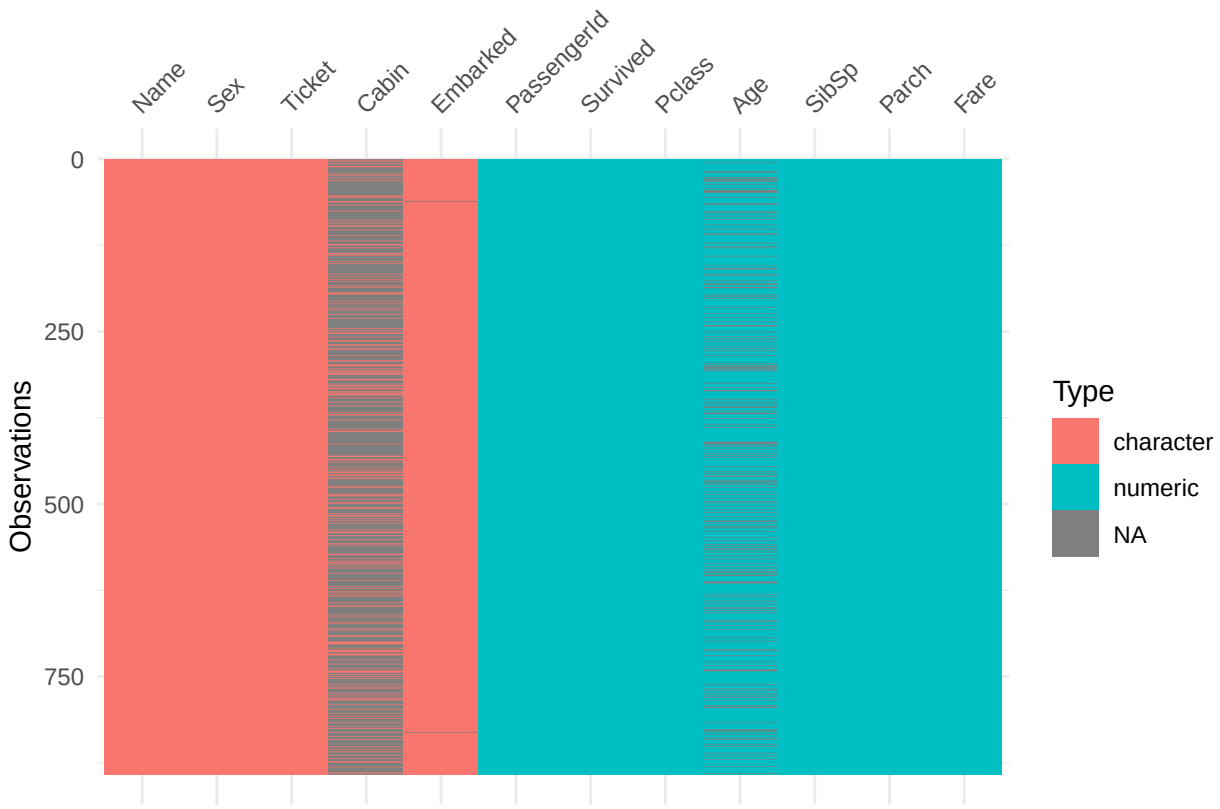
```
# Loading Dataset
train <- read_csv("~/Documents/Projects/Titanic/train.csv")
test <- read_csv("~/Documents/Projects/Titanic/test.csv")
gender_submission <- read_csv("~/Documents/Projects/Titanic/gender_submission.csv") %>% data.frame()
```

Data Dictionary

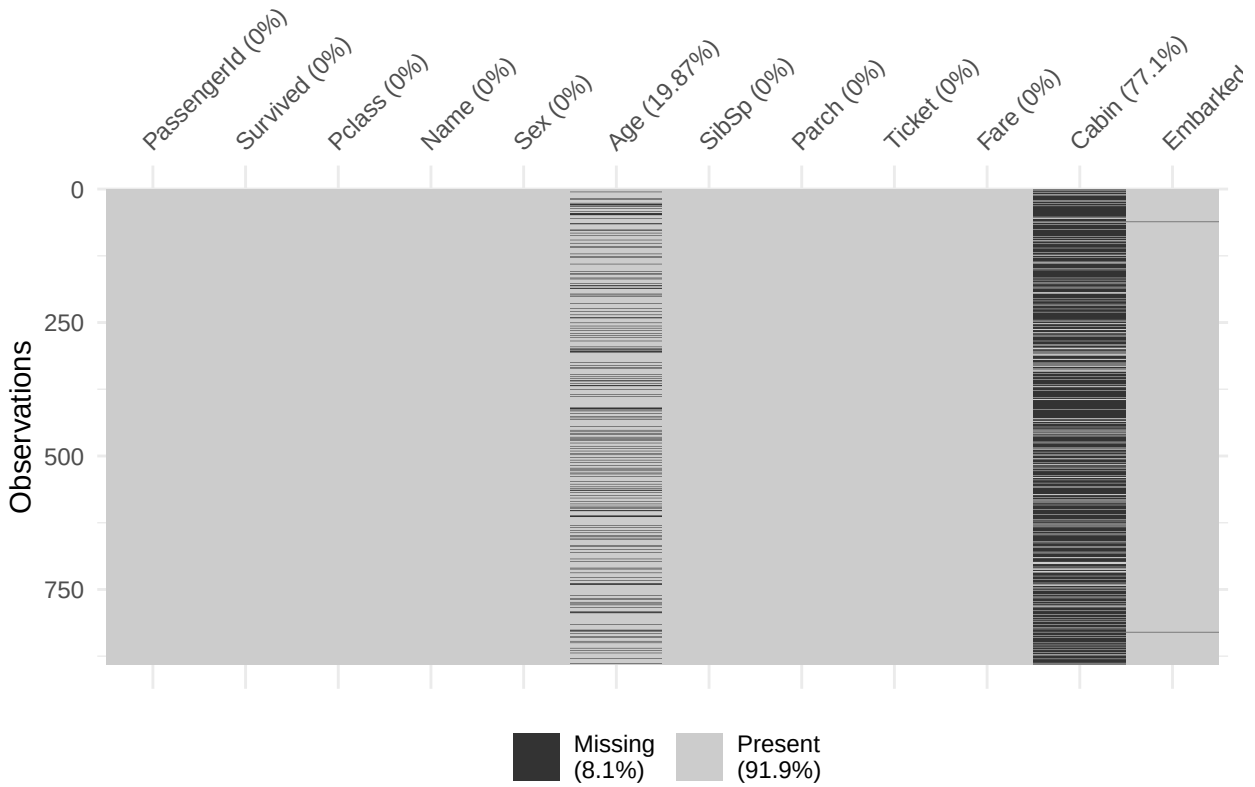
Variable Name	Descriptions
Survive	Survival: 0 = No, 1 = Yes
Pclass	Ticket class; 1 = 1st, 2 = 2nd, 3 = 3rd
Sex	Sex
Age	Age in years
Sibsp	# of siblings / spouses aboard the Titanic
Parch	# of parents / children aboard the Titanic
Ticket	Ticket number
Fare	Passenger fare
Cabin	Cabin number
Embarked	Port of Embarkation; C = Cherbourg, Q = Queenstown, S = Southampton

Overview of the Dataset

```
# Explore the data types and the percentage of missing value
vis_dat(train)
```




```
vis_miss(train)
```



- There are two **types of variable** in the raw dataset, including **Categorical** and **Numeric**.
  - **Categorical variables** include Name, Sex, Ticket, Cabin and Embarked.
  - **Numerical variables** include Age & Fare (continuous), and PassengerId, Survived, Pclass, SibSp & Parch (discrete).
- Some of the variable types does not make sense in this case. Thus, we have to do some type transformation before exploring the relationship between each variable and survived.
- From the second graph, we can also know that Cabin is the variable with largest amount of missing value. Then followed by Age and Embarked. Since **Cabin** contains 77.1% of the missing value, it is not that reasonable to analyze the relationship between cabin and survive in this case. We may **drop/ignore** this variable. As for Age, since it contains nearly 20% of the missing values, we may try to impute those NAs later.

```
# Type transformation

# Convert Survivied and Pclass into categorical variable
# Convert Sex, Cabin and Embarked to factor
clean_data <- train %>% data.frame()
clean_data$Survived <- factor(clean_data$Survived)
clean_data$Pclass <- factor(clean_data$Pclass)
clean_data$Sex <- factor(clean_data$Sex)
clean_data$Cabin <- factor(clean_data$Cabin)
clean_data$Embarked <- factor(clean_data$Embarked)
```

Summary of the dataset

```
# Summary of the dataset
head(clean_data)
```

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

```
summary(clean_data)
```

```
##   PassengerId    Survived Pclass      Name              Sex
##  Min.   :  1.0   0:549    1:216   Length:891         female:314
##  1st Qu.:223.5   1:342    2:184   Class :character   male  :577
##  Median :446.0            3:491   Mode  :character
##  Mean   :446.0
##  3rd Qu.:668.5
##  Max.   :891.0
##
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       Fare               Cabin     Embarked
##  Min.   :  0.00   B96 B98    :  4   C   :168
##  1st Qu.:  7.91   C23 C25 C27:  4   Q   : 77
##  Median : 14.45   G6         :  4   S   :644
##  Mean   : 32.20   C22 C26    :  3   NA's:  2
##  3rd Qu.: 31.00   D          :  3
##  Max.   :512.33   (Other)    :186
##                   NA's       :687
```

Categorical variable

- **Survived:** In this sample dataset, there are 891 observations. Among them, 549 did not survive while 342 survived. The survival rate of this sample is 38.38%.
- **Pclass:** There were 3 ticket classes. Within this sample, 216 of the observations were from the first class, 184 of the observations were from the second class. 491 of the observations were from the third

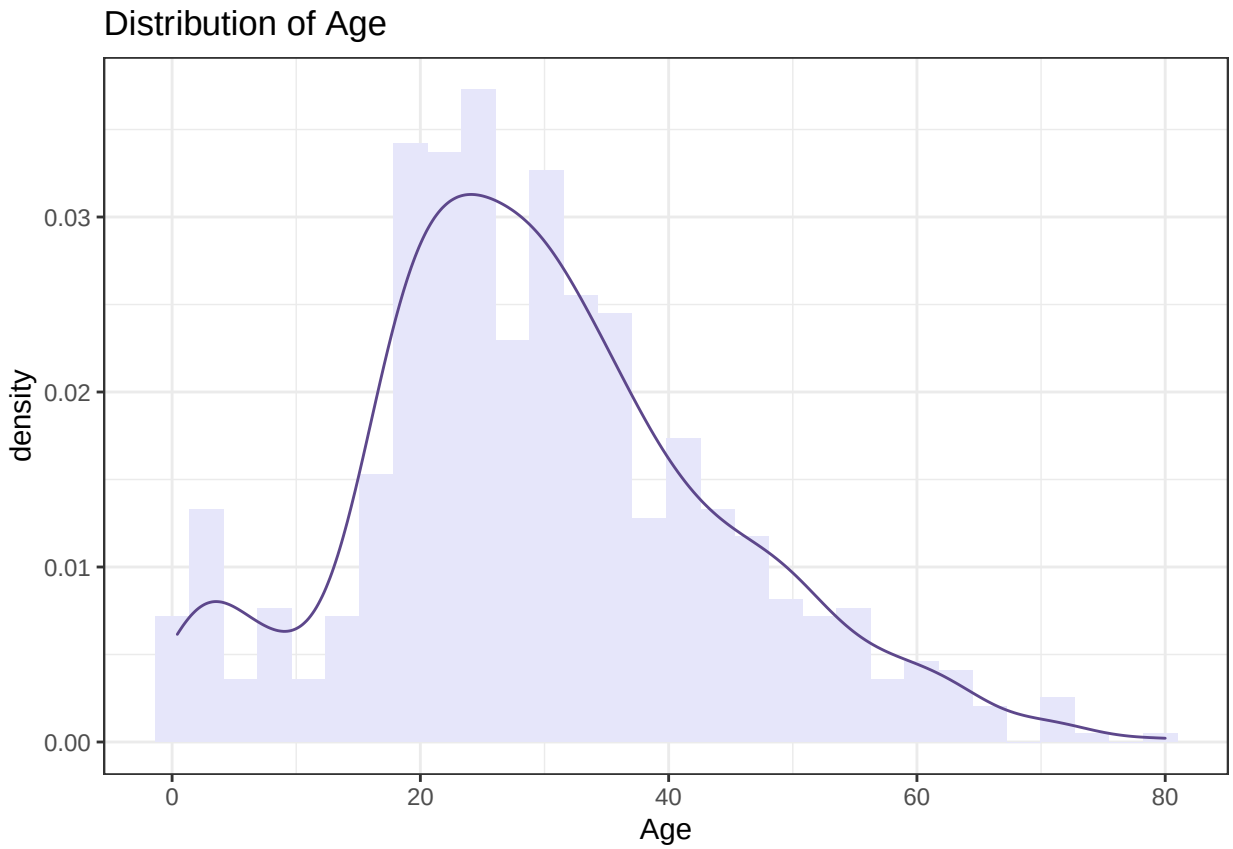
class.

- **Name:** This variable is a bit mess at this stage. Title information may be useful in further analysis and can be extracted.
- **Sex:** 314 of the observations in this sample were female and 577 of them were male.
- **Ticket:** Contains letters and numbers. Based on our common sense, it may not affect the survival. Thus can be **dropped**.
- **Embarked:** 168 of the observations departed from Cherbourg. 77 of the observations departed from Queenstown. 644 of the observations departed from Southampton. There are two observations whose port of embarkation did not be recorded.

Numerical variable

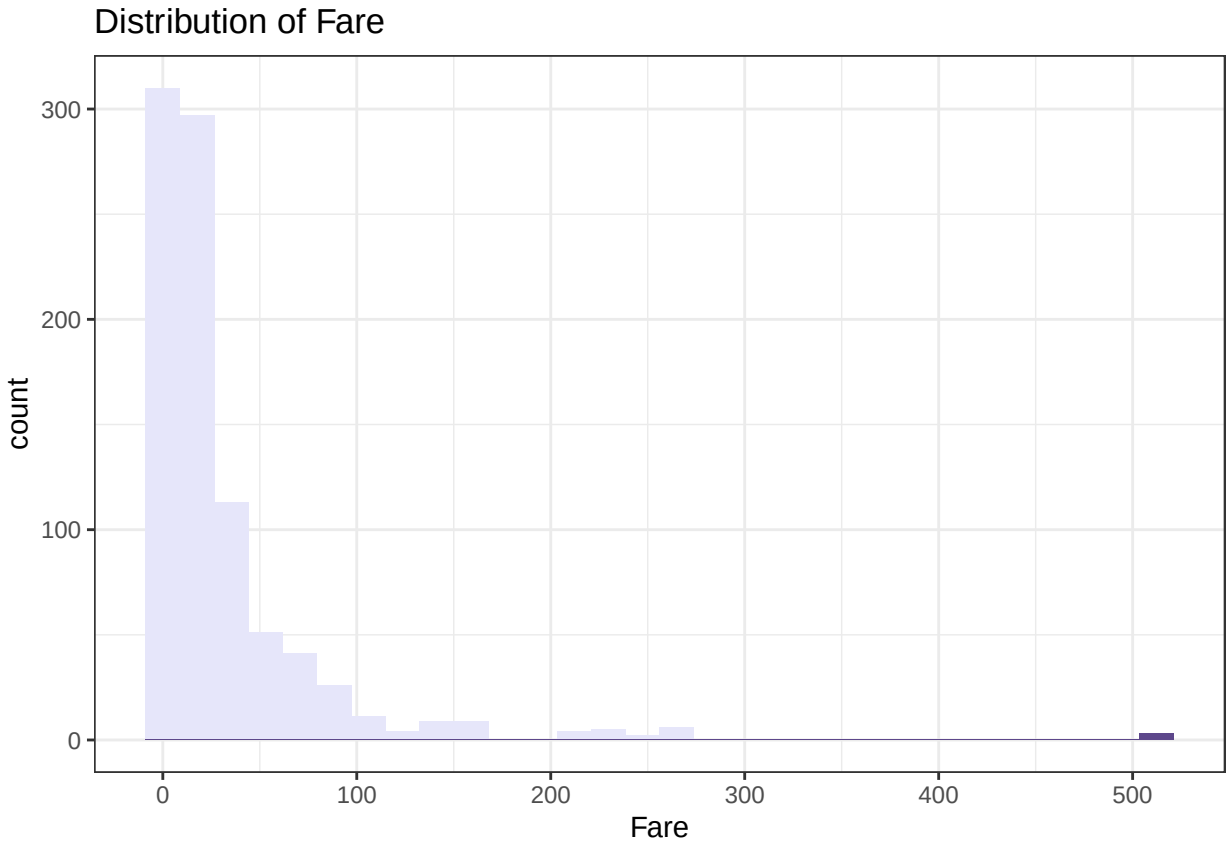
- **PassengerId:** A sequence of numbers, may not bring much information about the survival. According to common sense, it can be dropped. But this can be confirmed later.
- **Age:** The youngest observation in this sample was 0.42 years old. The oldest observation was 80 years old. The median of age is 28. May need to check whether there are some outliers in this variable.
- **SibSp:** 8 is the maximum amount of siblings / spouses that the observations in this sample travelled with. On average, each of them would travel with 0.523 siblings / spouses.
- **Parch:** 6 is maximum amount of parents / children that an observation had aboard the Titanic. At 3 quarters of them did not go with parents or children. On average, each observation may travel with 0.3816 children or parents.
- **Fare:** The minimum fare is 0 in this sample, while the maximum fare is 512.33. The median of the fare is 14.45. May need to check whether there are some outliers in this variable.

```
# Check whether Age contains some outliers
clean_data %>% ggplot(aes(x = Age, y = ..density..)) +
  geom_histogram(fill = "lavender") +
  geom_density(colour = "mediumpurple4") +
  ggtitle("Distribution of Age") +
  theme_bw()
```



This seems reasonable. Most of the observations are young adults.

```
# Check whether Fare contains some outliers
clean_data %>% ggplot(aes(x = Fare)) +
  geom_histogram(fill = "lavender") +
  geom_histogram(data=subset(clean_data,clean_data$Fare ==
                             max(clean_data$Fare)), fill="mediumpurple4") +
  ggtitle("Distribution of Fare") +
  theme_bw()
```



There are few observations with fare greater than 500. All the others are below 300.

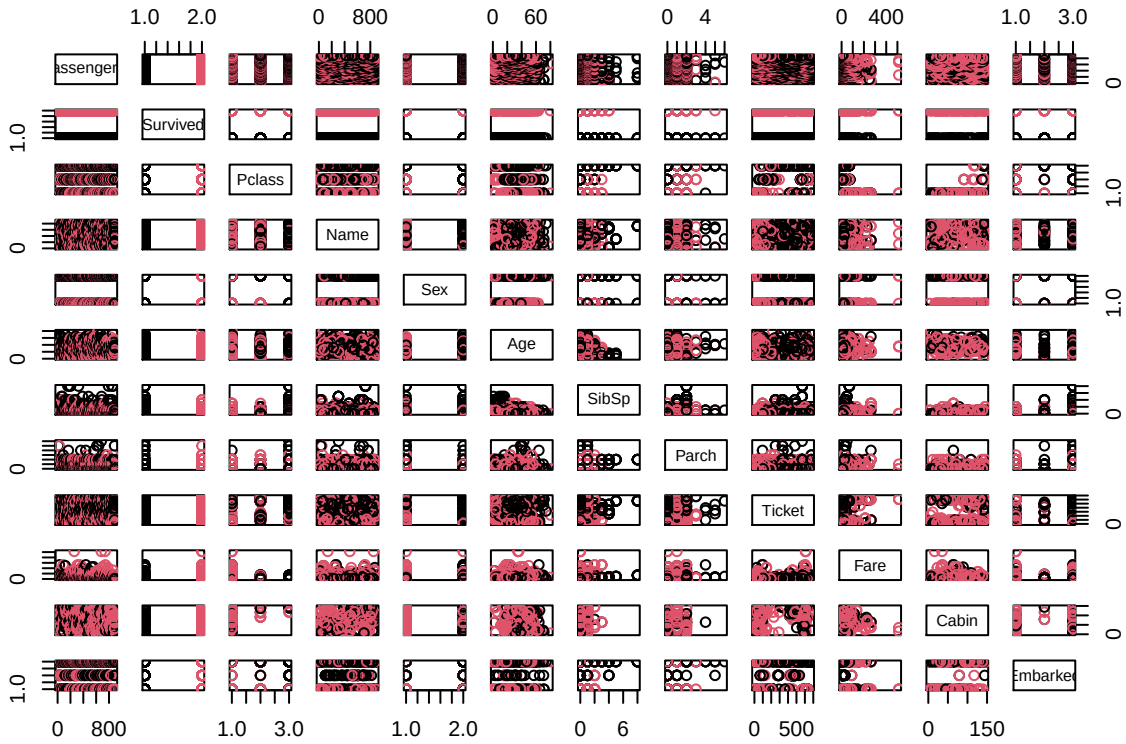
```
# More details about those had the maximum fare
max_fare <- clean_data %>% filter(clean_data$Fare == max(clean_data$Fare)) %>% data.frame()
max_fare %>% kable()
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
259	1	1	Ward, Miss. Anna	female	35	0	0	PC 17755	512.3292	NA	C
680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36	0	1	PC 17755	512.3292	B51 B53 B55	C
738	1	1	Lesurer, Mr. Gustave J	male	35	0	0	PC 17755	512.3292	B101	C

It seems that these may be considered as outliers.

Relationship Exploration

```
# Relationship overview
plot(clean_data, col = clean_data$Survived)
```

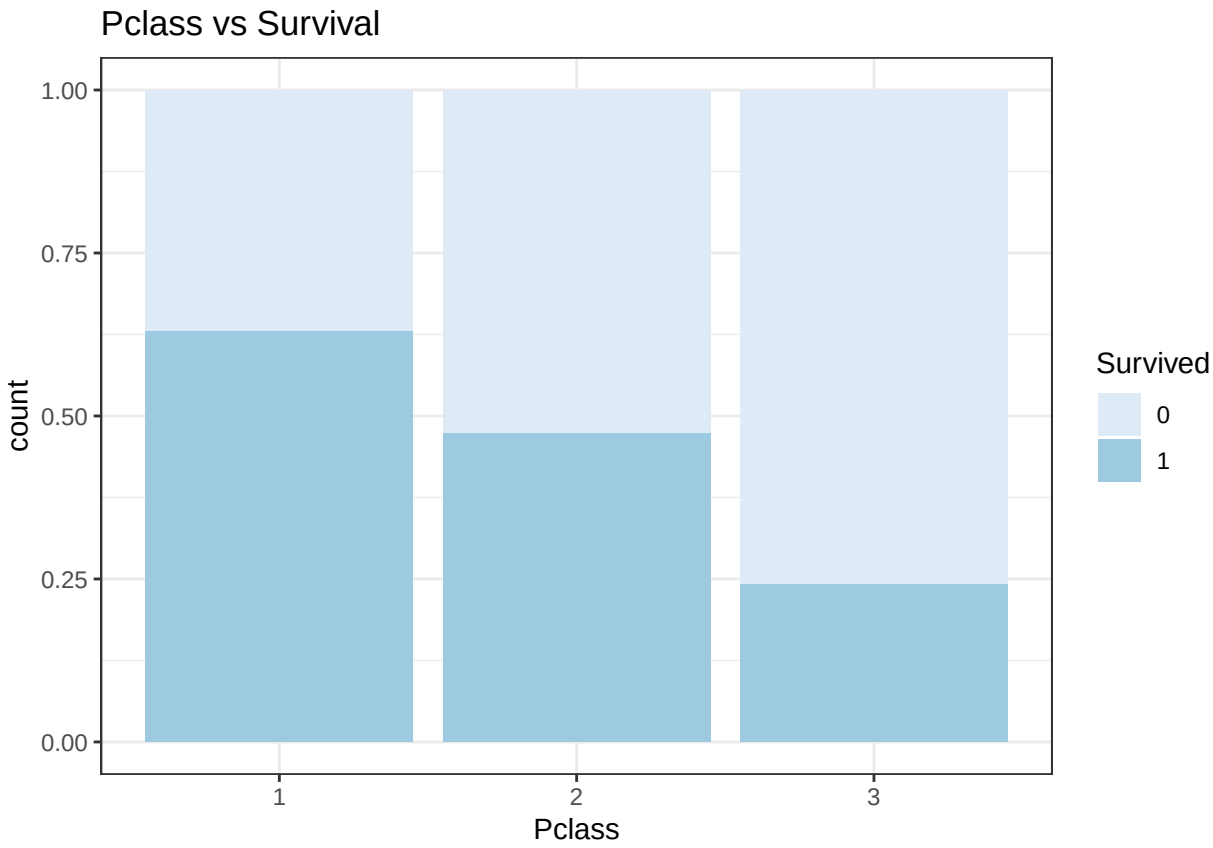


Hard to figure out the relationship.

We can explore the one-to-one relationship between each variable and Survival. For **Categorical variable**, we can visualize them by using **Bar Charts**. For **Numerical variable**, we can visualize them by using **Histograms**.

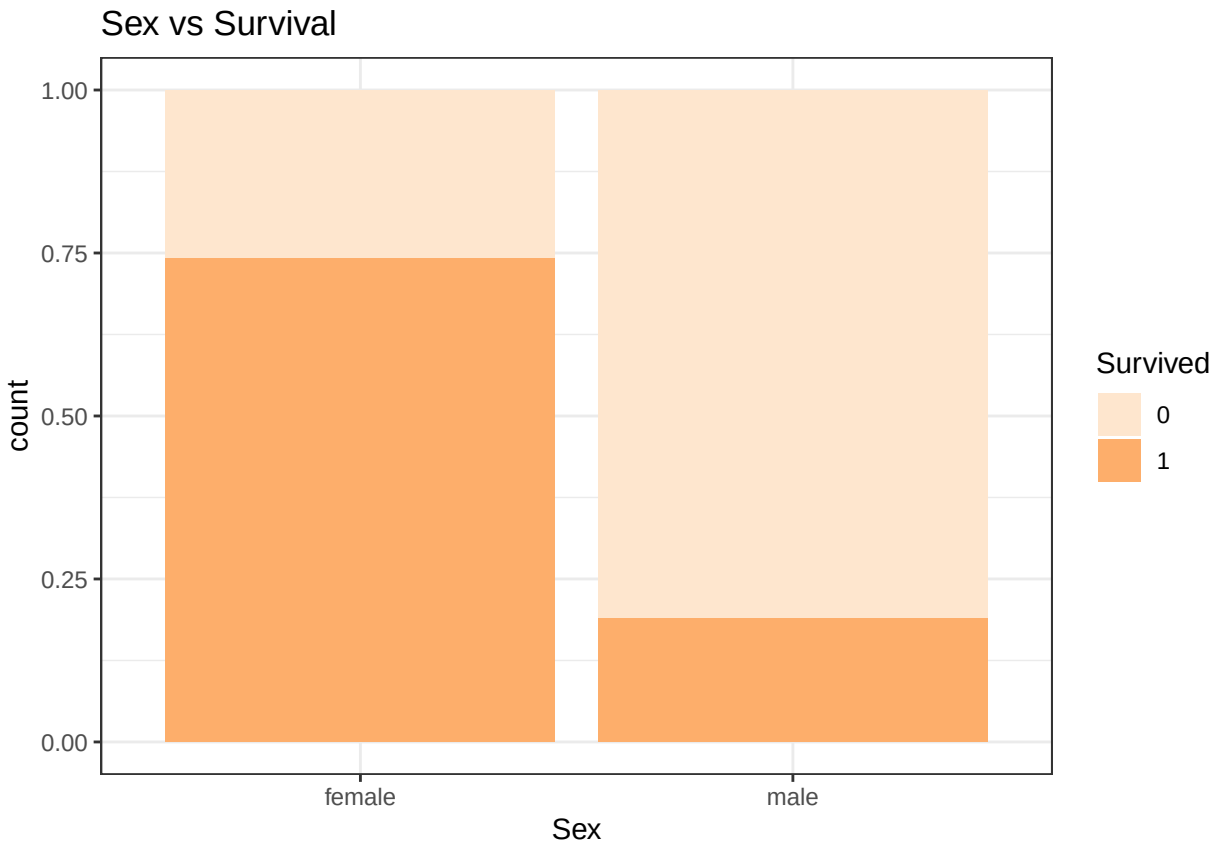
Categorical vs Survival

```
# Pclass
ggplot(clean_data[1:891,], aes(x = Pclass, fill = Survived)) +
  geom_bar(stat = "count", position = "fill") +
  scale_fill_brewer(palette = "Blues") +
  ggtitle("Pclass vs Survival") +
  theme_bw()
```



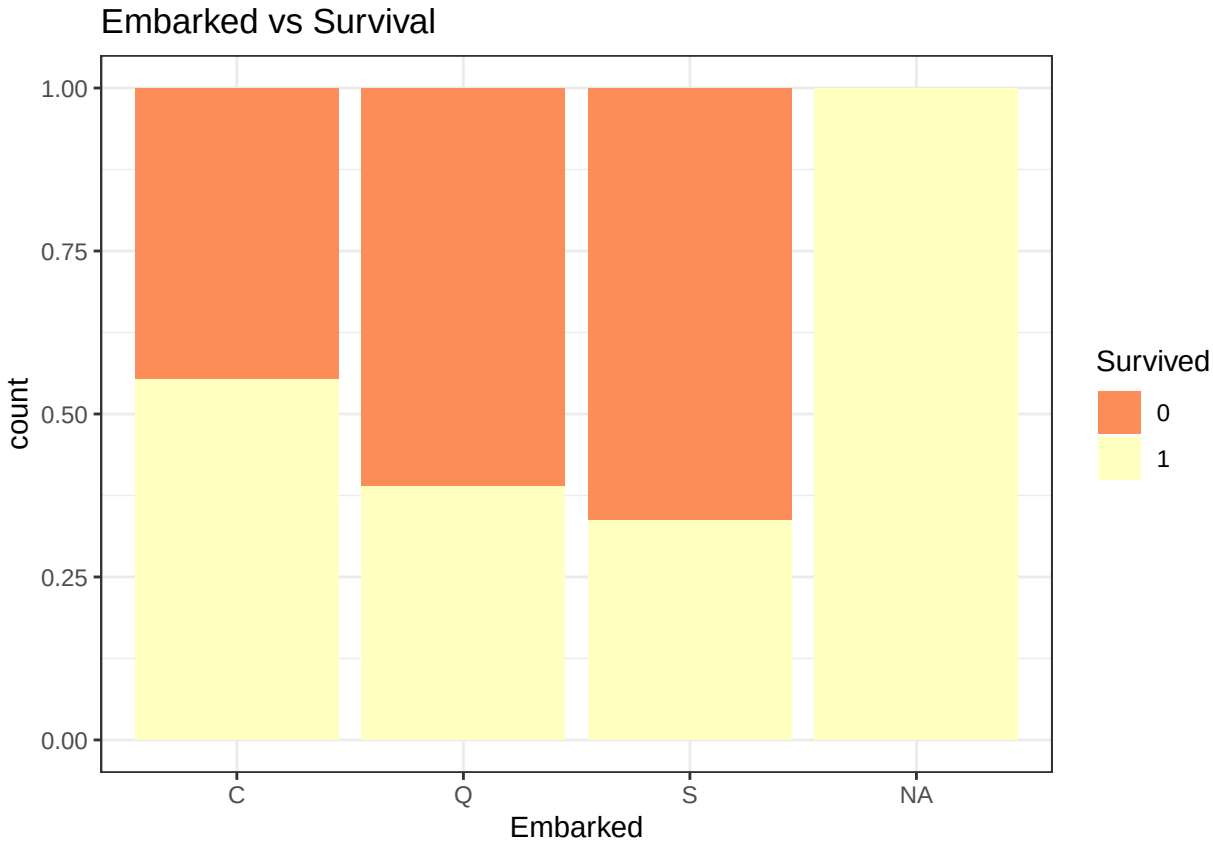
There is a relationship between ticket classes and survival. People from the first class were more likely to survive, while people from the third class were less likely to survive.

```
# Sex
ggplot(clean_data[1:891,], aes(x = Sex, fill = Survived)) +
  geom_bar(stat = "count", position = "fill") +
  scale_fill_brewer(palette = "Oranges") +
  ggtitle("Sex vs Survival") +
  theme_bw()
```



Sex may also influence the survival. Female were more likely to survive then male in this disaster.

```
# Embarked
ggplot(clean_data[1:891,], aes(x = Embarked, fill = Survived)) +
  geom_bar(stat = "count", position = "fill") +
  scale_fill_brewer(palette = "Spectral") +
  ggtitle("Embarked vs Survival") +
  theme_bw()
```



It seems that port of embarkation may influence the survival as well. More than 50% of the people departed from Cherbourg survived. However, people departed from Queenstown and Southampton had smaller survival rate, with around 38% and 35% respectively.

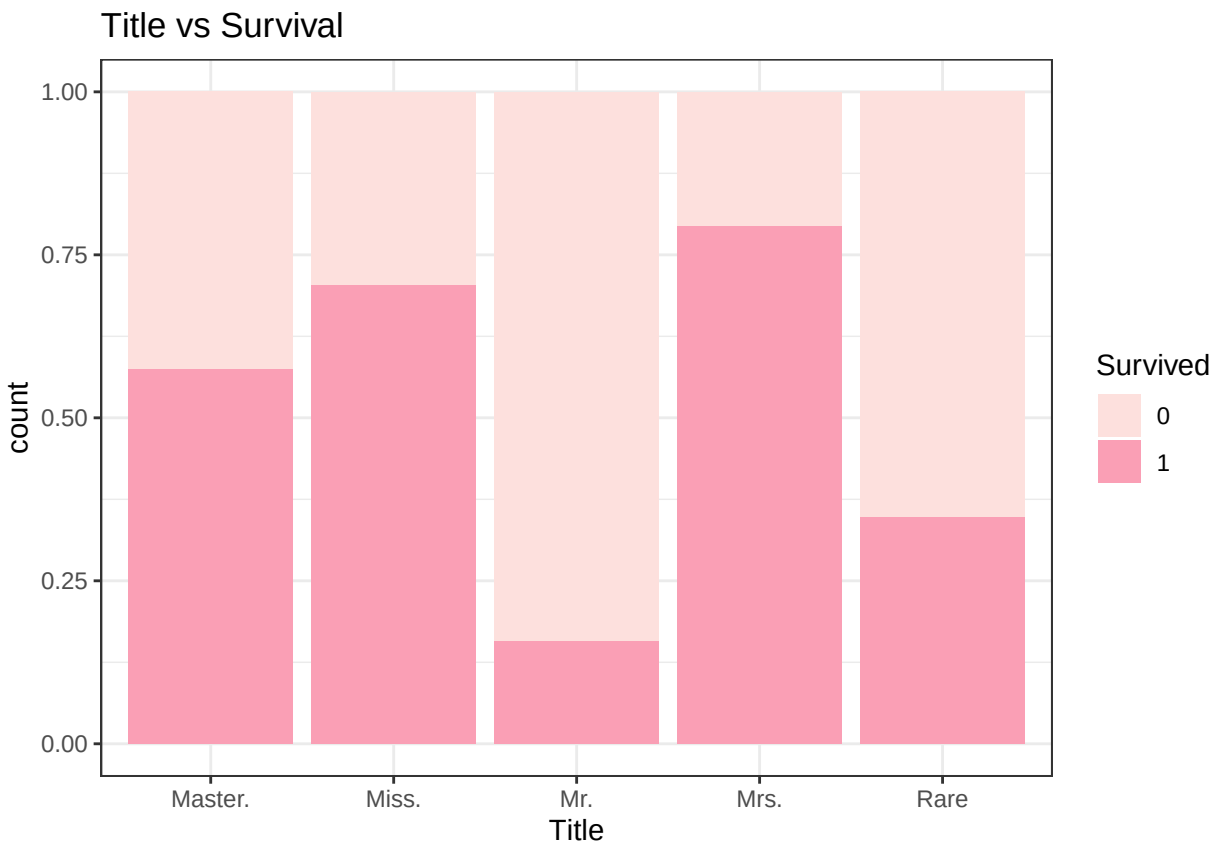
```
# Extract title information from variable "Name"
clean_data <- clean_data %>% mutate(Title = str_extract(clean_data$Name, '([A-Za-z]+)\\.'))
table(clean_data$Title)
```

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

```
clean_data$Title <- recode(clean_data$Title,
                           "c('Don.', 'Rev.', 'Dr.', 'Major.',
                           'Lady.','Sir.', 'Col.', 'Capt.', 'Countess.', 'Jonkheer.') = 'Rare'")
clean_data$Title <- recode(clean_data$Title, "'Mme.' = 'Mrs.'")
clean_data$Title <- recode(clean_data$Title, "'Ms.' = 'Miss.'")
clean_data$Title <- recode(clean_data$Title, "'Mlle.' = 'Miss.'")
clean_data$Title <- as.factor(clean_data$Title)
```

```
# Title
ggplot(clean_data[1:891,], aes(x = Title, fill = Survived)) +
  geom_bar(stat = "count", position = "fill") +
  scale_fill_brewer(palette = "RdPu") +
  ggtitle("Title vs Survival") +
```

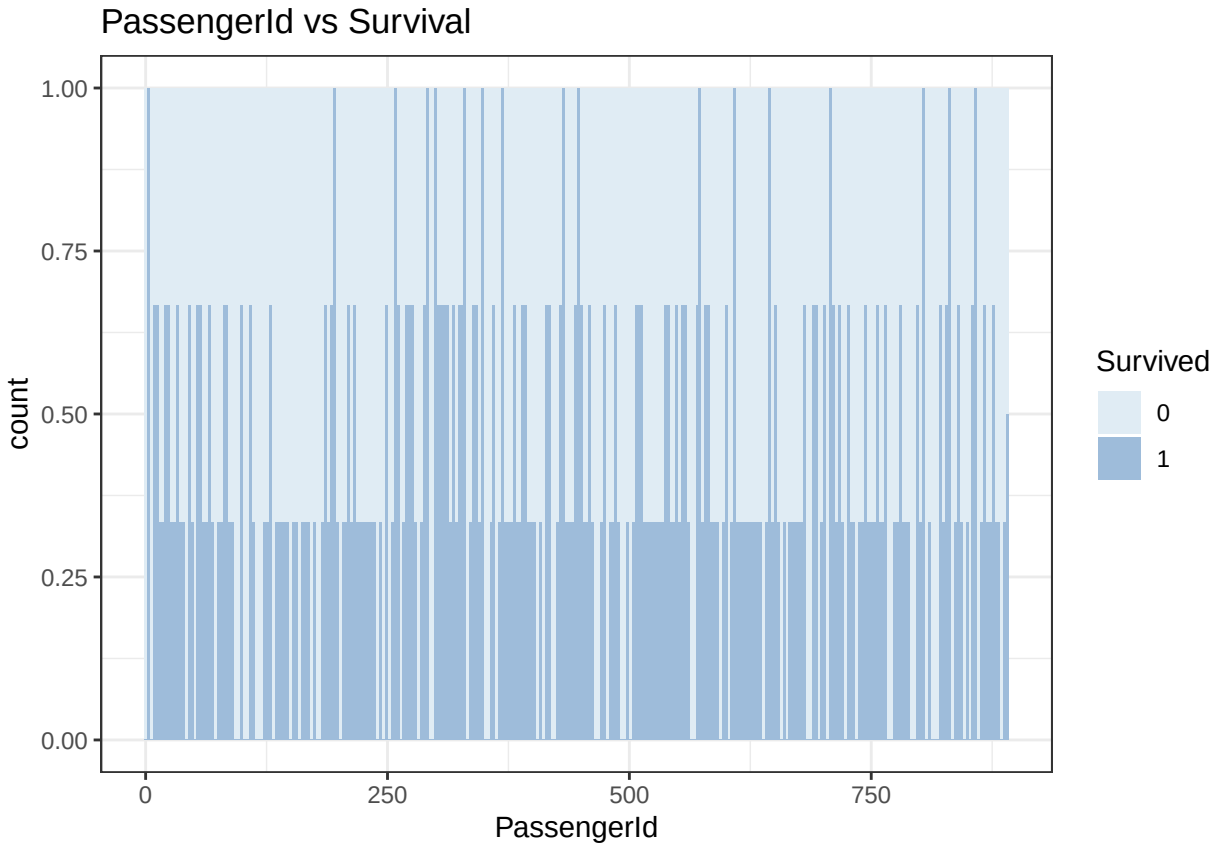
```
  theme_bw()
```



Title may affect the survival.

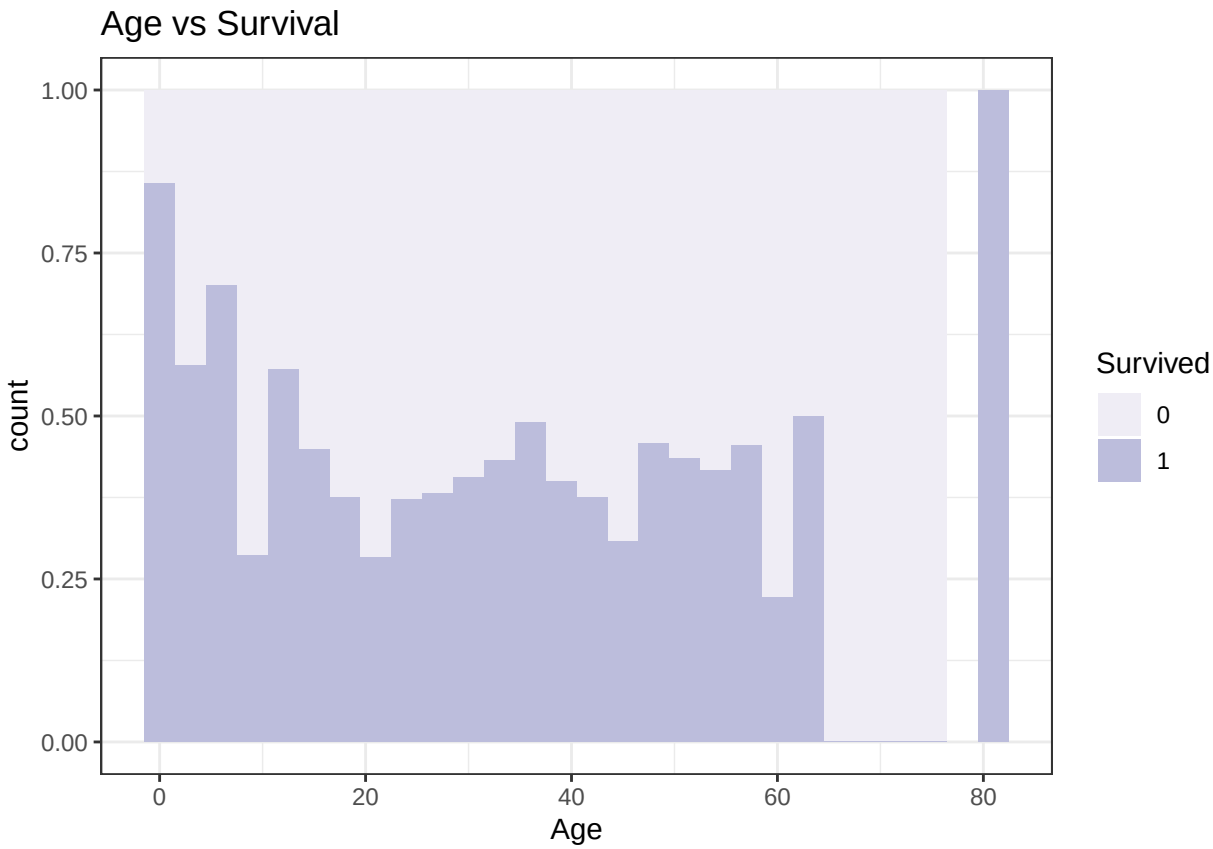
Numerical vs Survival

```
# PassengerId
ggplot(clean_data[1:891,], aes(x = PassengerId, fill = Survived)) +
  geom_histogram(binwidth = 3, position = "fill") +
  scale_fill_brewer(palette = "BuPu") +
  ggtitle("PassengerId vs Survival") +
  theme_bw()
```



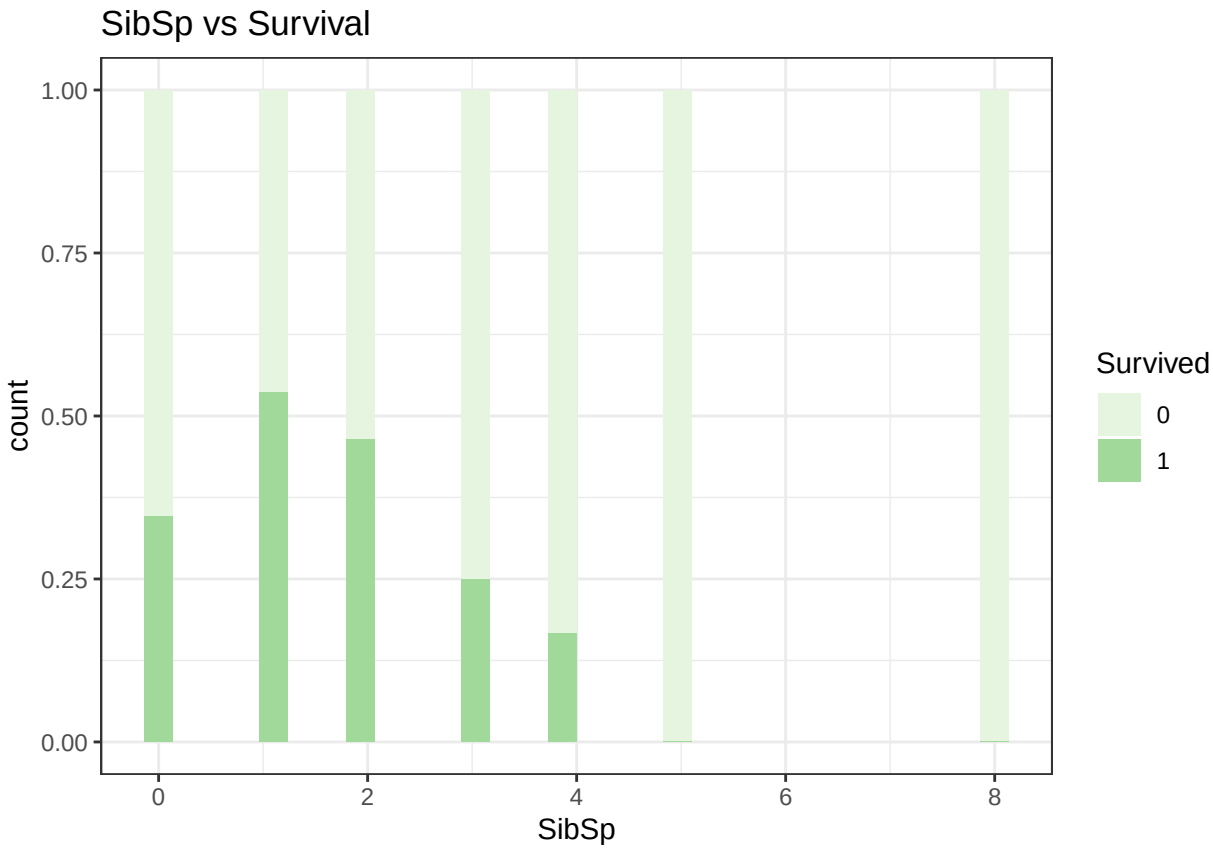
It is obvious that there is no pattern in this graph. Thus we may conclude that PassengerId may not influence survival. **PassengerId** can be dropped in further analysis.

```
# Age
ggplot(clean_data[1:891,], aes(x = Age, fill = Survived)) +
  geom_histogram(binwidth = 3, position = "fill") +
  scale_fill_brewer(palette = "Purples") +
  ggtitle("Age vs Survival") +
  theme_bw()
```

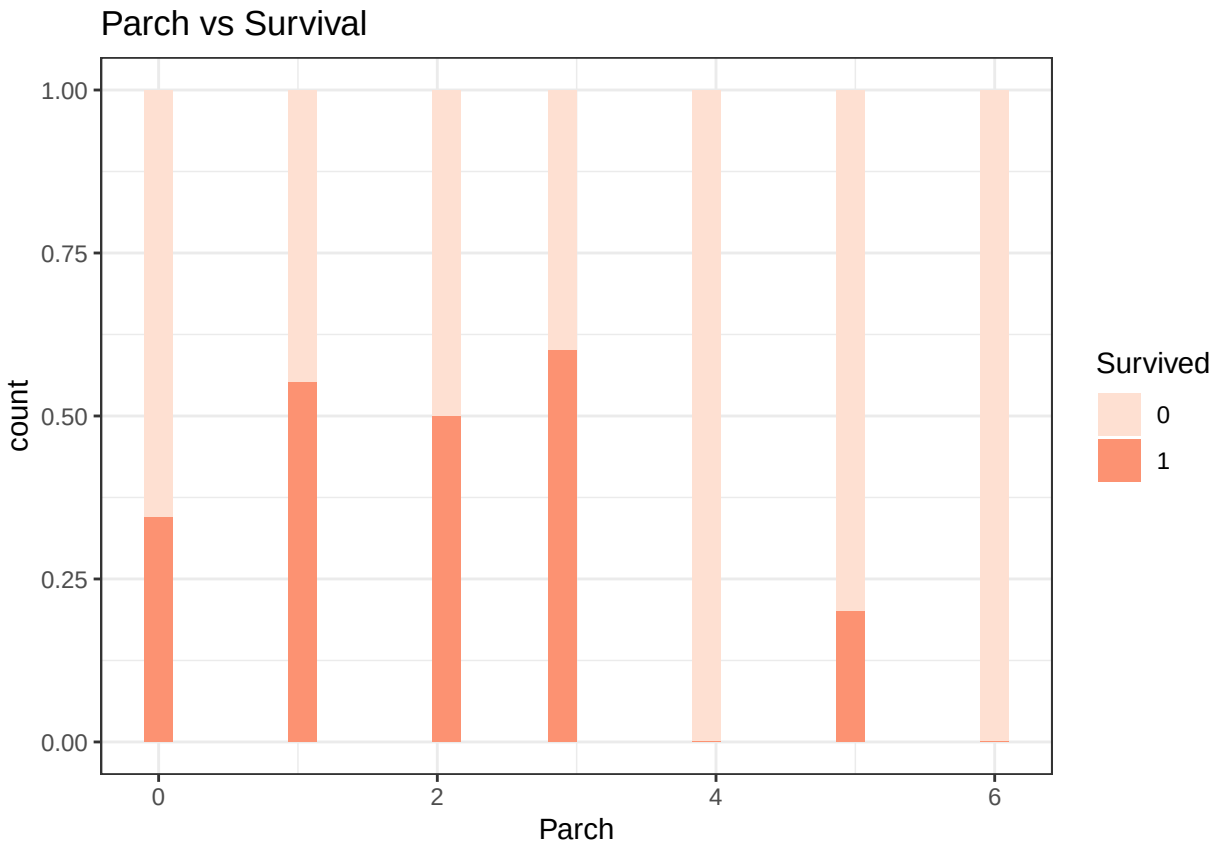


Relationship exists. Young people whose age is below 20 were more likely to survive.

```
# SibSp
ggplot(clean_data[1:891,], aes(x = SibSp, fill = Survived)) +
  geom_histogram(position = "fill") +
  scale_fill_brewer(palette = "Greens") +
  ggtitle("SibSp vs Survival") +
  theme_bw()
```



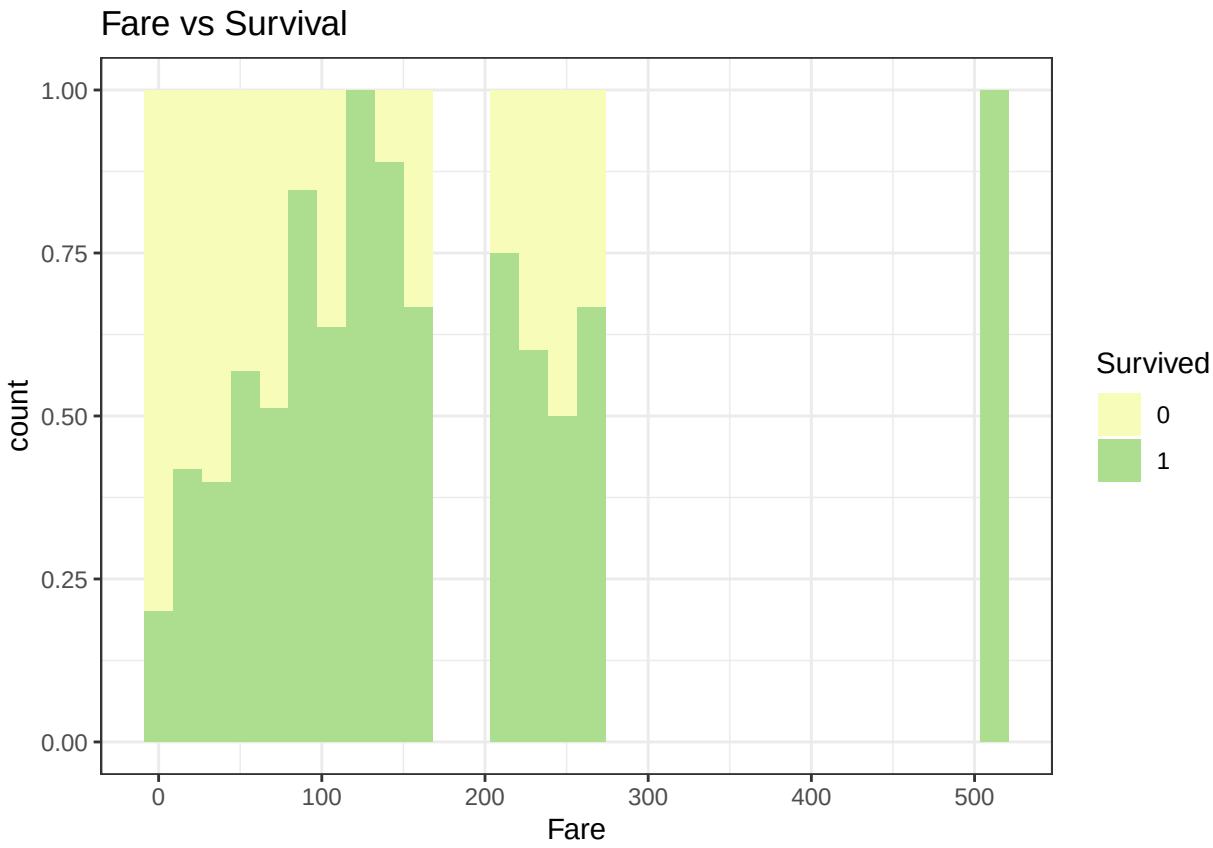

```
# Parch
ggplot(clean_data[1:891,], aes(x = Parch, fill = Survived)) +
  geom_histogram(position = "fill") +
  scale_fill_brewer(palette = "Reds") +
  ggtitle("Parch vs Survival") +
  theme_bw()
```



It seems SibSp has similar effects as Parch which is people had fewer relatives were more likely to survive. Thus, we may combine these two variables and create a new variable "Family"

```
# Combine SibSp and Parch to create variable Family
clean_data <- clean_data %>% mutate(Family = SibSp + Parch + 1)
```

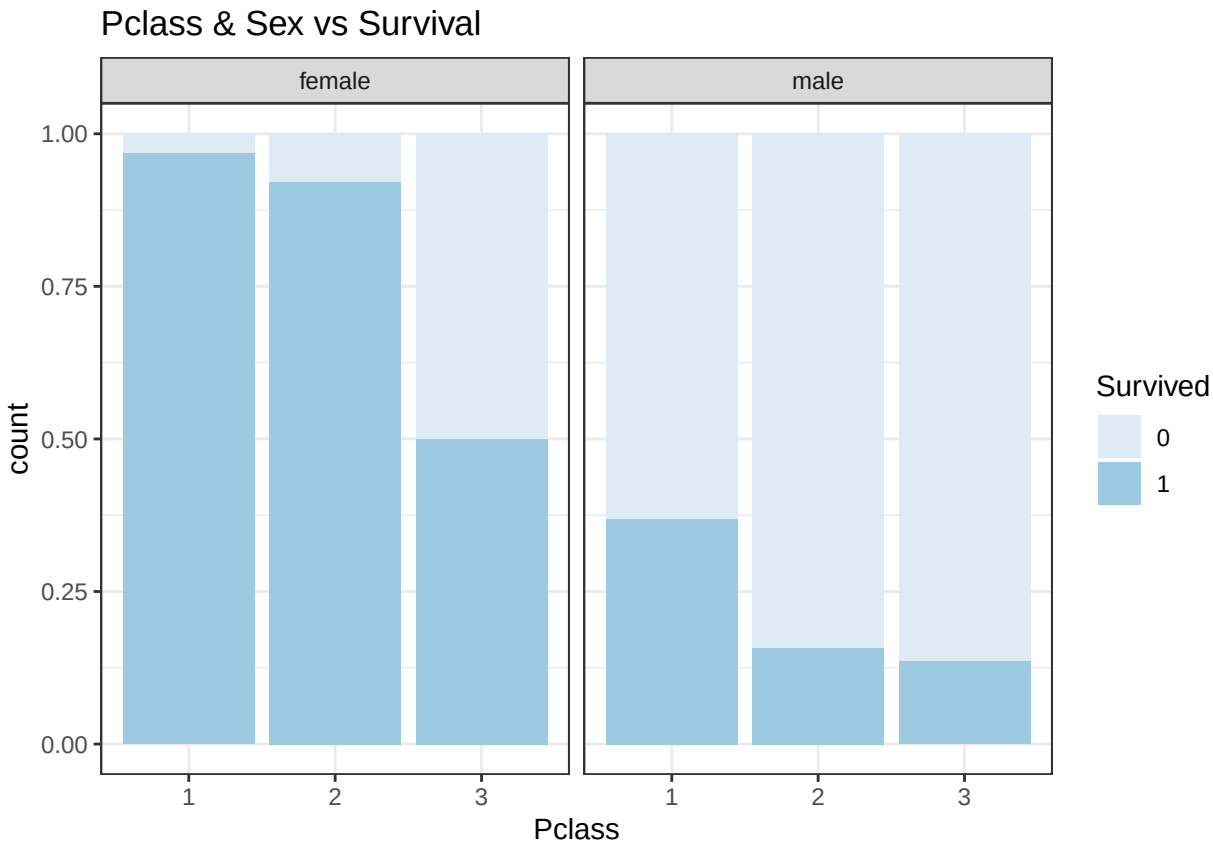
```
# Fare
ggplot(clean_data[1:891,], aes(x = Fare, fill = Survived)) +
  geom_histogram(position = "fill") +
  scale_fill_brewer(palette = "YlGn") +
  ggtitle("Fare vs Survival") +
  theme_bw()
```



It seems that passengers who paid for higher fare (greater than 100) were more likely to survive.

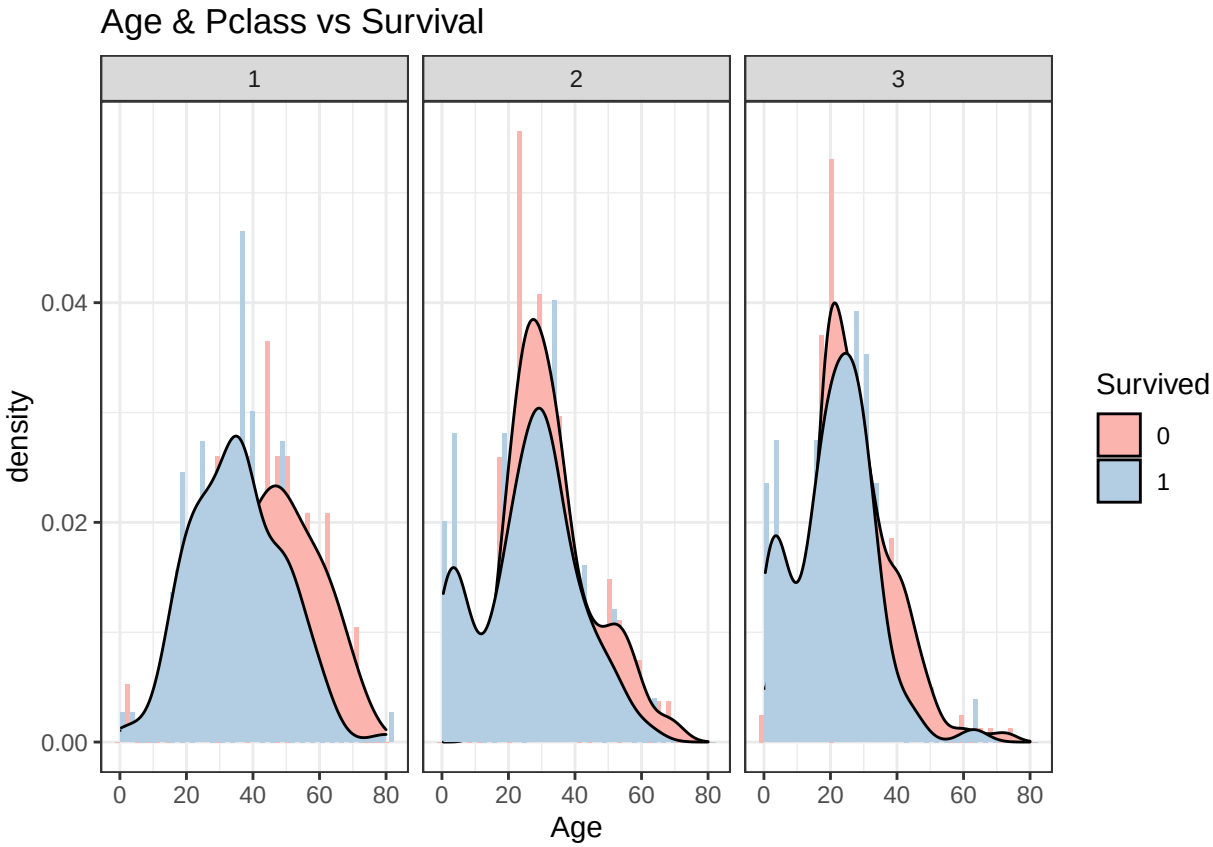
Interaction between variables

```
# Interaction between Sex and Pclass
ggplot(clean_data[1:891,], aes(x = Pclass, fill = Survived)) +
  geom_bar(stat = "count", position = "fill") +
  facet_grid(~Sex) +
  ggtitle("Pclass & Sex vs Survival") +
  scale_fill_brewer(palette = "Blues") +
  theme_bw()
```



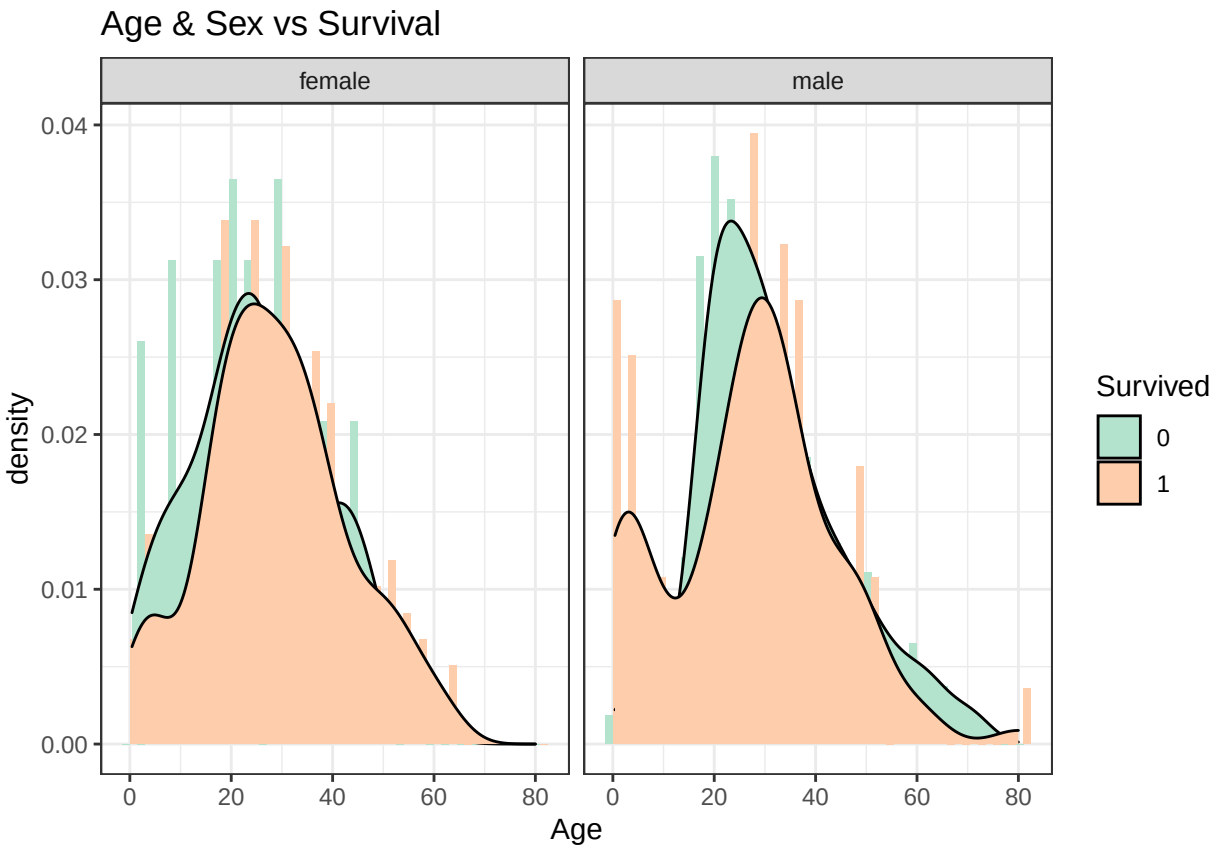
No matter in which class, female were still more likely to survive than male.

```
ggplot(clean_data[1:891,], aes(x = Age, y = ..density.., fill = Survived)) +
  geom_histogram(binwidth = 3, position = "dodge") +
  facet_grid(~Pclass) +
  geom_density() +
  ggtitle("Age & Pclass vs Survival") +
  scale_fill_brewer(palette = "Pastel1") +
  theme_bw()
```



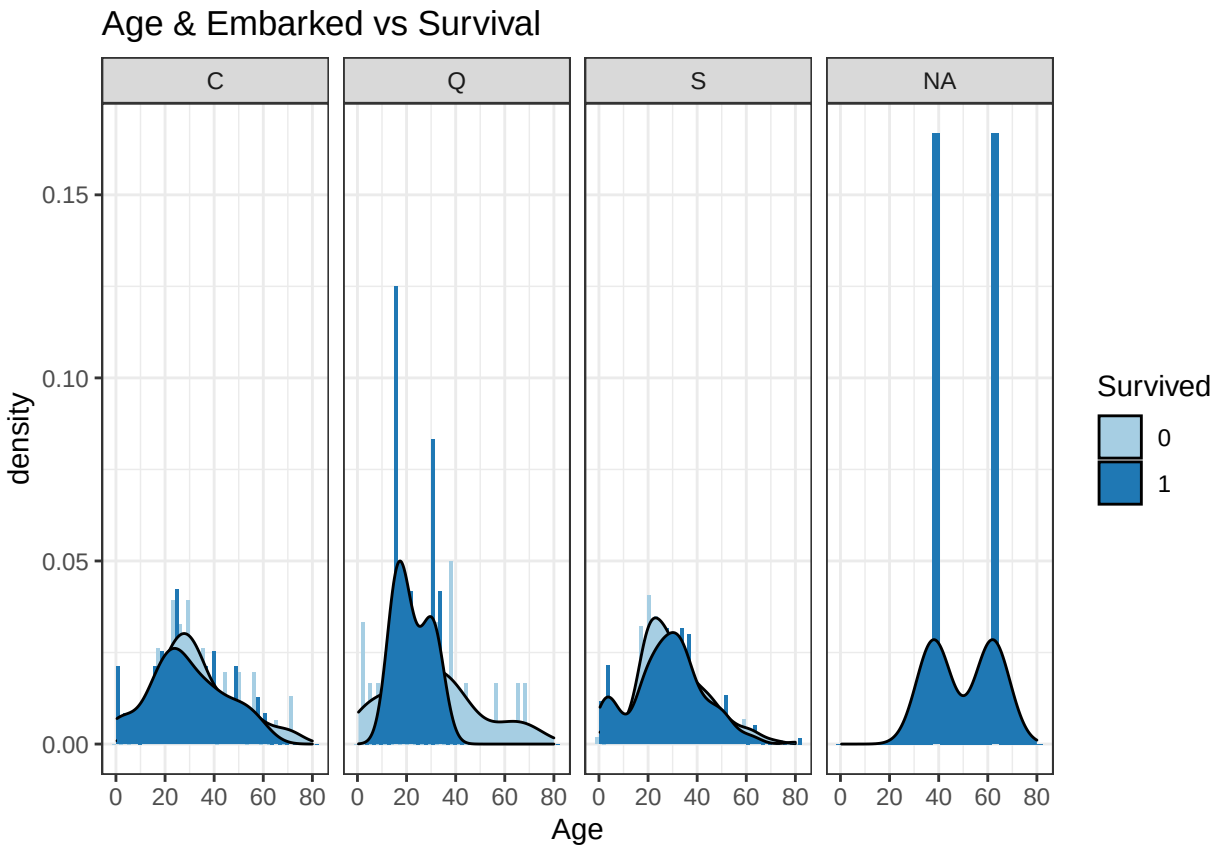
In first class, passengers who were less than 40 years old had higher survival rate. However, for class 2 and class 3, survival rates were higher for passengers younger than 20 years old. Thus, may consider the **interaction between Age and Pclass** when constructing the model.

```
ggplot(clean_data[1:891,], aes(x = Age, y = ..density.., fill = Survived)) +
  geom_histogram(binwidth = 3, position = "dodge") +
  geom_density() +
  facet_grid(~Sex) +
  ggtitle("Age & Sex vs Survival") +
  scale_fill_brewer(palette = "Pastel2") +
  theme_bw()
```



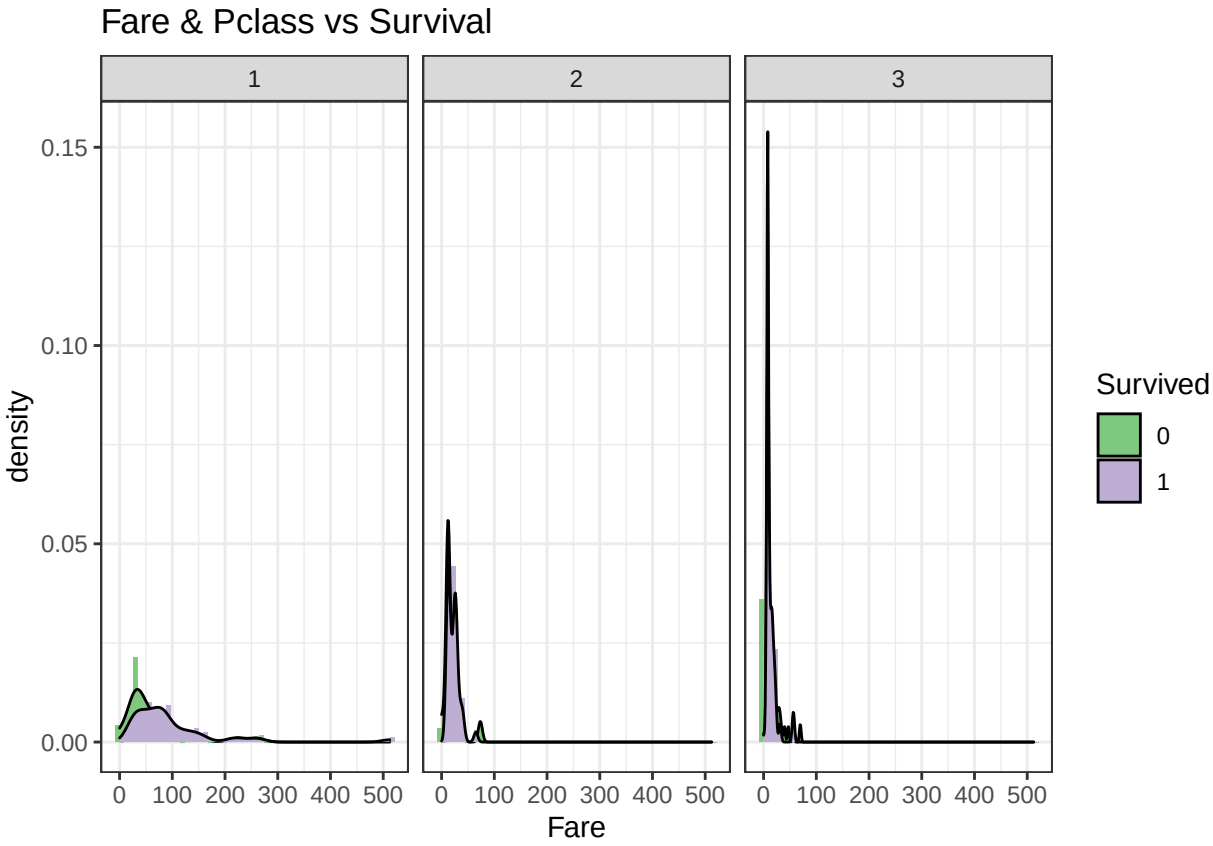
From these graphs, we can see that interaction exists between sex and age. It seems that male had higher survival rate if they were less than 15 years old. Thus, **interaction between Sex and Age** can be considered when doing further ananlysis.

```
ggplot(clean_data[1:891,], aes(x = Age, y = ..density.., fill = Survived)) +
  geom_histogram(binwidth = 3, position = "dodge") +
  geom_density() +
  facet_grid(~Embarked) +
  ggtitle("Age & Embarked vs Survival") +
  scale_fill_brewer(palette = "Paired") +
  theme_bw()
```



It seems that the partterns are a little bit different for different Embarked. Thus the interaction between age and embarked be considered.

```
ggplot(clean_data[1:891,], aes(x = Fare, y = ..density.., fill = Survived)) +
  geom_histogram(position = "dodge") +
  geom_density() +
  facet_grid(~Pclass) +
  ggtitle("Fare & Pclass vs Survival") +
  scale_fill_brewer(palette = "Accent") +
  theme_bw()
```



Interaction may also exists between Fare and Pclass. For first class, passengers were likely to survive if they paid more than 50 while for class 2, passengers paid 5 to 50 were more likely to survive. Passengers from class 3 paid 1 to 15 had higher survival rate. Thus, **interaction between Fare and Pclass** can be considered.

Conclusion for Relationship Exploration

Most variables in this dataset had influence on survival. These include:

Categorical: Pclass, Sex, Embarked, Title

Numerical: Age, Family, Fare

Interaction between these variables may be considered as well.

Data Wrangling

Training set

```
# Drop those unused variables
drops <- c("PassengerId", "Cabin", "SibSp", "Parch", "Name", "Ticket")
clean_data <- clean_data[ , !(names(clean_data) %in% drops)]
```

Since there were still 2 missing values in variable embarked, we may consider impute the variable age by using information from those related variables. In this way, we may use the median of age from different combination of class and sex.

```
# Complete the variables with missing values

# For Age
```

```
# When sex == female and class == 1
Female_1 <- clean_data[clean_data$Pclass == "1" & clean_data$Sex == "female", ]$Age %>%
  na.omit() %>% data.frame()
Female_1 <- median(Female_1[, 1])

clean_data[clean_data$Pclass == "1" & clean_data$Sex == "female" & clean_data$Age %in% NA, 'Age'] <- Fe

# When sex == female and class == 2
Female_2 <- clean_data[clean_data$Pclass == "2" & clean_data$Sex == "female", ]$Age %>%
  na.omit() %>% data.frame()
Female_2 <- median(Female_2[, 1])

clean_data[clean_data$Pclass == "2" & clean_data$Sex == "female" & clean_data$Age %in% NA, 'Age'] <- Fe

# When sex == female and class == 3
Female_3 <- clean_data[clean_data$Pclass == "3" & clean_data$Sex == "female", ]$Age %>%
  na.omit() %>% data.frame()
Female_3 <- median(Female_3[, 1])

clean_data[clean_data$Pclass == '3' & clean_data$Sex == "female" & clean_data$Age %in% NA, 'Age'] <- Fe

# When sex == male and class == 1
Male_1 <- clean_data[clean_data$Pclass == "1" & clean_data$Sex == "male", ]$Age %>%
  na.omit() %>% data.frame()
Male_1 <- median(Male_1[, 1])

clean_data[clean_data$Pclass == "1" & clean_data$Sex == "male" & clean_data$Age %in% NA, 'Age'] <- Male

# When sex == male and class == 2
Male_2 <- clean_data[clean_data$Pclass == "2" & clean_data$Sex == "male", ]$Age %>%
  na.omit() %>% data.frame()
Male_2 <- median(Male_2[, 1])

clean_data[clean_data$Pclass == "2" & clean_data$Sex == "male" & clean_data$Age %in% NA, 'Age'] <- Male

# When sex == male and class == 3
Male_3 <- clean_data[clean_data$Pclass == "3" & clean_data$Sex == "male", ]$Age %>%
  na.omit() %>% data.frame()
Male_3 <- median(Male_3[, 1])

clean_data[clean_data$Pclass == "3" & clean_data$Sex == "male" & clean_data$Age %in% NA, 'Age'] <- Male
```

Since there are only two missing value, we can just fill them with the mode.

```
# For embarked

# Find the mode
table(clean_data$Embarked)
```

```
##
##   C   Q   S
## 168  77 644
```

```
clean_data[clean_data$Embarked %in% NA, 'Embarked'] <- 'S'
```

Test set

```
gender_submission$Survived <- as.factor(gender_submission$Survived)
```

```
clean_test <- test %>% data.frame()

# Types transformation
clean_test$Pclass <- factor(clean_test$Pclass)
clean_test$Sex <- factor(clean_test$Sex)
clean_test$Embarked <- factor(clean_test$Embarked)

# Create Title variable
clean_test <- clean_test %>% mutate(Title = str_extract(test$Name, '([A-Za-z]+)\\.'))
clean_test$Title <- recode(clean_test$Title,
                           "c('Dona.', 'Dr.', 'Rev.', 'Col.') = 'Rare'")
clean_test$Title <- recode(clean_test$Title, "'Ms.' = 'Miss.'")

clean_test$Title <- as.factor(clean_test$Title)

# Create Family variable
clean_test <- clean_test %>% mutate(Family = SibSp + Parch + 1)
```

```
# Drop unused variables
drops_test <- c("SibSp", "Parch", "Cabin", "Ticket", "Name")
clean_test <- clean_test[ , !(names(clean_test) %in% drops_test)]
```

```
summary(clean_test)
```

```
##   PassengerId     Pclass      Sex           Age             Fare
##  Min.   : 892.0   1:107   female:152   Min.   : 0.17   Min.   :  0.000
##  1st Qu.: 996.2   2: 93   male  :266   1st Qu.:21.00   1st Qu.:  7.896
##  Median :1100.5   3:218                 Median :27.00   Median : 14.454
##  Mean   :1100.5                         Mean   :30.27   Mean   : 35.627
##  3rd Qu.:1204.8                         3rd Qu.:39.00   3rd Qu.: 31.500
##  Max.   :1309.0                         Max.   :76.00   Max.   :512.329
##                                         NA's   :86      NA's   :1
##  Embarked     Title         Family
##  C:102    Master.: 21   Min.   : 1.00
##  Q: 46    Miss.  : 79   1st Qu.: 1.00
##  S:270    Mr.    :240   Median : 1.00
##           Mrs.   : 72   Mean   : 1.84
##           Rare   :  6   3rd Qu.: 2.00
##                         Max.   :11.00
##
```

```
# Impute missing values

# When sex == female and class == 1
Test_Female_1 <- clean_test[clean_test$Pclass == "1" & clean_test$Sex == "female", ]$Age %>%
  na.omit() %>% data.frame()
Test_Female_1 <- median(Test_Female_1[, 1])

clean_test[clean_test$Pclass == "1" & clean_test$Sex == "female" & clean_test$Age %in% NA, 'Age'] <- Tes

# When sex == female and class == 2
```

```r
Test_Female_2 <- clean_test[clean_test$Pclass == "2" & clean_test$Sex == "female", ]$Age %>%
  na.omit() %>% data.frame()
Test_Female_2 <- median(Test_Female_2[, 1])

clean_test[clean_test$Pclass == "2" & clean_test$Sex == "female" & clean_test$Age %in% NA, 'Age'] <- Tes

# When sex == female and class == 3
Test_Female_3 <- clean_test[clean_test$Pclass == "3" & clean_test$Sex == "female", ]$Age %>%
  na.omit() %>% data.frame()
Test_Female_3 <- median(Test_Female_3[, 1])

clean_test[clean_test$Pclass == '3' & clean_test$Sex == "female" & clean_test$Age %in% NA, 'Age'] <- Tes

# When sex == male and class == 1
Test_Male_1 <- clean_test[clean_test$Pclass == "1" & clean_test$Sex == "male", ]$Age %>%
  na.omit() %>% data.frame()
Test_Male_1 <- median(Test_Male_1[, 1])

clean_test[clean_test$Pclass == "1" & clean_test$Sex == "male" & clean_test$Age %in% NA, 'Age'] <-Test_M

# When sex == male and class == 2
Test_Male_2 <- clean_test[clean_test$Pclass == "2" & clean_test$Sex == "male", ]$Age %>%
  na.omit() %>% data.frame()
Test_Male_2 <- median(Test_Male_2[, 1])

clean_test[clean_test$Pclass == "2" & clean_test$Sex == "male" & clean_test$Age %in% NA, 'Age'] <- Male_

# When sex == male and class == 3
Test_Male_3 <- clean_test[clean_test$Pclass == "3" & clean_test$Sex == "male", ]$Age %>%
  na.omit() %>% data.frame()
Test_Male_3 <- median(Test_Male_3[, 1])

clean_test[clean_test$Pclass == "3" & clean_test$Sex == "male" & clean_test$Age %in% NA, 'Age'] <- Test_
```

```r
# Impute the only missing value with the mean
clean_test[clean_test$Fare %in% NA, 'Fare'] <- mean(na.omit(clean_test$Fare))
```

Modelling and Prediction

Logistic Model

```r
# CV errors of Logistic Model with main effects

n <- dim(clean_data)[1]
CVLog <- 0
for (i in 1:n) {
  logistic <- glm(Survived~., data = clean_data[-i,], family = binomial(link = "logit"))
  glmprobs <- predict(logistic, newdata = clean_data[i,], type = "response")
  glmpred <- ifelse(glmprobs > 0.5, 1, 0)
  CVLog <- CVLog + sum(glmpred != clean_data[i,1])
}

data.frame(CVLog/n) %>% kable(col.names = NULL, caption = "LogMain")
```

Table 3: LogMain

0.1717172

```
# CV errors of Logistic Model with some interactions
CVLoginter <- 0
for (i in 1:n) {
  logistic_inter <- glm(Survived~ Pclass + Sex + Age + Fare + Embarked + Title + Family + Age*Sex*Pclass
  glminterprobs <- predict(logistic_inter, newdata = clean_data[i,], type = "response")
  glminterpred <- ifelse(glminterprobs > 0.5, 1, 0)
  CVLoginter <- CVLoginter + sum(glminterpred != clean_data[i,1])
}

data.frame(CVLoginter/n) %>% kable(col.names = NULL, caption = "LogInter1")
```

Table 4: LogInter1

0.1683502

```
# CV errors of Logistic Model with some interactions
CVLoginter <- 0
for (i in 1:n) {
  logistic_inter <- glm(Survived~ Pclass + Sex + Age + Fare + Embarked + Title + Family + Age*Sex*Pclass
  glminterprobs <- predict(logistic_inter, newdata = clean_data[i,], type = "response")
  glminterpred <- ifelse(glminterprobs > 0.5, 1, 0)
  CVLoginter <- CVLoginter + sum(glminterpred != clean_data[i,1])
}

data.frame(CVLoginter/n) %>% kable(col.names = NULL, caption = "LogInter2")
```

Table 5: LogInter2

0.1661055

It seems that the third logistic model is a bit better than the first two in the training set.

Apply to the test set

```
logistic_inter <- glm(Survived~., data = clean_data, family = binomial(link = "logit"))

glminterprobs <- predict(logistic_inter, newdata = clean_test[,], type = "response")
glminterpred <- ifelse(glminterprobs > 0.5, 1, 0)

table(glminterpred, gender_submission$Survived)
```

```
##
## glminterpred   0   1
##            0 242   9
##            1  24 143
```

```
n_test <- dim(clean_test)[1]
LogMainTest <- (24 + 9)/n_test
data.frame(LogMainTest) %>% kable(col.names = NULL, caption = "LogMainTest")
```

Table 6: LogMainTest

0.0789474

```
logistic_inter <- glm(Survived~ Pclass + Sex + Age + Fare + Embarked + Title + Family + Age*Sex*Pclass +

glminterprobs <- predict(logistic_inter, newdata = clean_test[,], type = "response")
glminterpred <- ifelse(glminterprobs > 0.5, 1, 0)

table(glminterpred, gender_submission$Survived)
```

```
##
## glminterpred   0   1
##            0 242  19
##            1  24 133
```

```
LogInterTest <- (24 + 19)/n_test
data.frame(LogInterTest) %>% kable(col.names = NULL, caption = "LogInterTest1")
```

Table 7: LogInterTest1

0.1028708

```
logistic_inter <- glm(Survived~ Pclass + Sex + Age + Fare + Embarked + Title + Family + Age*Sex*Pclass,

glminterprobs <- predict(logistic_inter, newdata = clean_test[,], type = "response")
glminterpred <- ifelse(glminterprobs > 0.5, 1, 0)

table(glminterpred, gender_submission$Survived)
```

```
##
## glminterpred   0   1
##            0 241  18
##            1  25 134
```

```
LogInterTest2 <- (25 + 18)/n_test
data.frame(LogInterTest2) %>% kable(col.names = NULL, caption = "LogInterTest2")
```

Table 8: LogInterTest2

0.1028708

When apply to the test set, it seems that the first logistic model (model only include the main effects or model without any interaction) is the best logistic model.

LDA

```
# CV errors of LDA
CVlda <- 0
for (i in 1:n) {
  LDA <- lda(Survived~., data = clean_data[-i,])
  LDApred <- predict(LDA, newdata = clean_data[i,])$class
  CVlda <- CVlda + sum(LDApred != clean_data[i,1])
}

data.frame(CVlda/n) %>% kable(col.names = NULL, caption = "LDACV")
```

Table 9: LDACV

0.1672278

Apply to test set

```
LDA <- lda(Survived~., data = clean_data)
LDApred <- predict(LDA, newdata = clean_test[,])$class
table(LDApred, gender_submission$Survived)
```

```
##
## LDApred   0   1
##       0 250   4
##       1  16 148
```

```
LDATest <- (4 + 16)/n_test
data.frame(LDATest) %>% kable(col.names = NULL, caption = "LDATest")
```

Table 10: LDATest

0.0478469

Radom Forest

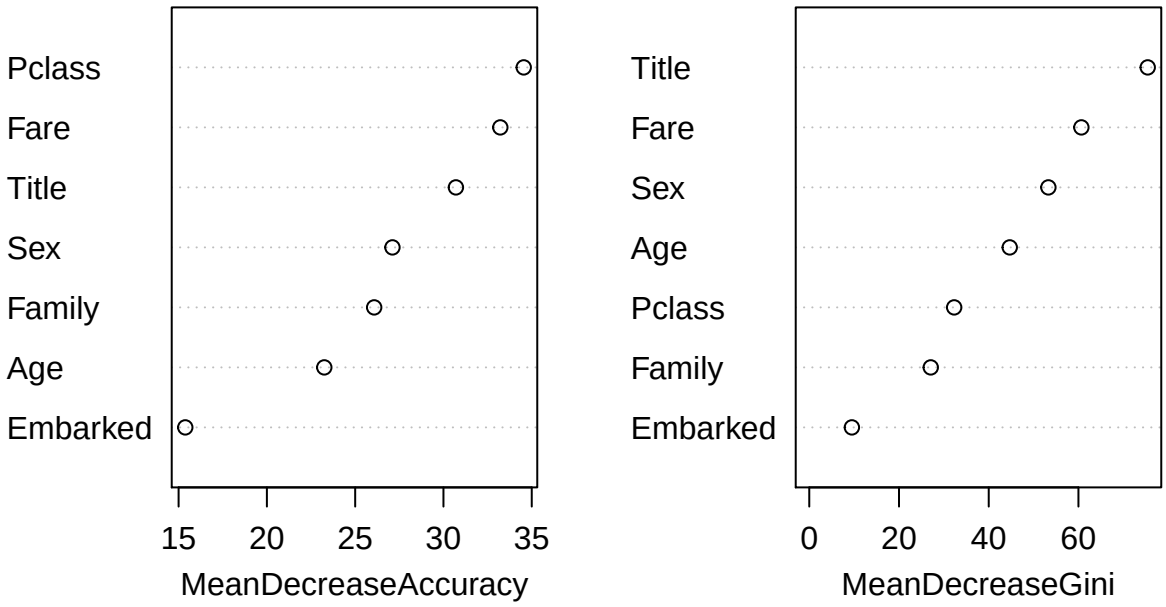
```
set.seed(12121021)
m.rf <- randomForest(formula = Survived~., data = clean_data, importance = TRUE)
OOBpred <- predict(m.rf)

# Random Forest Out-of-Bag Errors
data.frame(sum(OOBpred != clean_data$Survived)/n) %>% kable(col.names = NULL, caption = "RFOOB")
```

Table 11: RFOOB

0.1627385

```
# Importance of the variables
varImpPlot(m.rf)
```



Apply to the test set

```
pred_test <- predict(m.rf, clean_test)
RFTest <- sum(pred_test != gender_submission$Survived)/n_test
data.frame(RFTest) %>% kable(col.names = NULL, caption = "RFTest")
```

Table 12: RFTest

0.1100478

Conclusion

```
data.frame(LDATest, LogMainTest, LogInterTest, LogInterTest2, RFTest) %>% kable()
```

LDATest	LogMainTest	LogInterTest	LogInterTest2	RFTest
0.0478469	0.0789474	0.1028708	0.1028708	0.1100478

We can see that the **test errors** is **minimized** when using the **LDA model**.

```
# Save the prediction results
TestResult <- data.frame(PassengerID = clean_test$PassengerId, Survived = LDApred)
write.csv(TestResult, file = "Titanic_Test_Result.csv", row.names = F)
```