Titanic Data Analysis

Bahram Khanlarov

2022-08-08

Introduction

The sinking of the Titanic is one of the most infamous shipwrecks in history.

On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

We get the dataset from Kaggle platfoem through the link: https://www.kaggle.com/competitions/titanic/data We do basic data preparation and data exploration on this dataset.

Basic understanding of the data

We start reading our training and test set:

```
titanic_train <-read.csv("train.csv")
titanic_test <- read.csv("test.csv")
```

We can check the structure of the data using str():

```
str(titanic_train)
```

```
str(titanic_test)
```

The training set has 891 observations and 12 variables and the testing set has 418 observations and 11 variables. The traning set has 1 extra varible. Check which which one we are missing. I know we could see that in a very small dataset like this, but if its larger we want two compare them.

```
colnames_check <- colnames(titanic_train) %in% colnames(titanic_test)
colnames(titanic_train[colnames_check==FALSE])
```

```
## [1] "Survived"
```

As we can see we are missing the Survived in the test set. Which is correct because thats our challenge, we must predict this by creating a model.

```
#Use sapply(#object, class) to check the class of every column.
sapply(titanic_train, class)
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##   "integer"   "integer"   "integer" "character" "character"   "numeric"
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##   "integer"   "integer" "character"   "numeric" "character" "character"
```

We can see that the Survived and Pclass column are integers and Sex character. But they are actually categorical variables. To convert them into categorical variables (or factors), use the factor() function.Survived is a nominal categorical variable, whereas Pclass is an ordinal categorical variable. For an ordinal variable,

we provide the order=TRUE and levels argument in the ascending order of the values(Pclass 3 < Pclass 2 < Pclass 1).

```
#change columns class
#Survived: from integer into factor
titanic_train$Survived = as.factor(titanic_train$Survived)
titanic_train$Sex = as.factor(titanic_train$Sex)
titanic_train$Pclass=factor(titanic_train$Pclass,order=TRUE, levels = c(3, 2, 1))
```

Let's look deeper into the training set, and check how many passengers that survived vs did not make it.

```
table(titanic_train$Survived)
```

```
##
##   0   1
## 549 342
```

Out of the 891 there are only 342 who survived it. Check also as proportions.

```
prop.table(table(titanic_train$Survived))
```

```
##
##         0         1
## 0.6161616 0.3838384
```

A little more than one-third of the passengers survived the disaster. Now see if there is a difference between males and females that survived vs males that passed away.

```
table(titanic_train$Sex, titanic_train$Survived)
```

```
##
##            0   1
##   female  81 233
##   male   468 109
```

```
prop.table(table(titanic_train$Sex, titanic_train$Survived),margin = 1)
```

```
##
##                   0         1
##   female 0.2579618 0.7420382
##   male   0.8110919 0.1889081
```

As we can see most of the female survived and most of the male did not make it.

Data Preparation

Now we need to clean the dataset to create our models. Note that it is important to explore the data so that we understand what elements need to be cleaned.

```
#missing data

is.na(titanic_train)
sum(is.na(titanic_train))
```

```
#This function shows us exactly how much values are missing in each column.
apply(titanic_train, MARGIN = 2, FUN = function(x) {sum(is.na(x))})
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

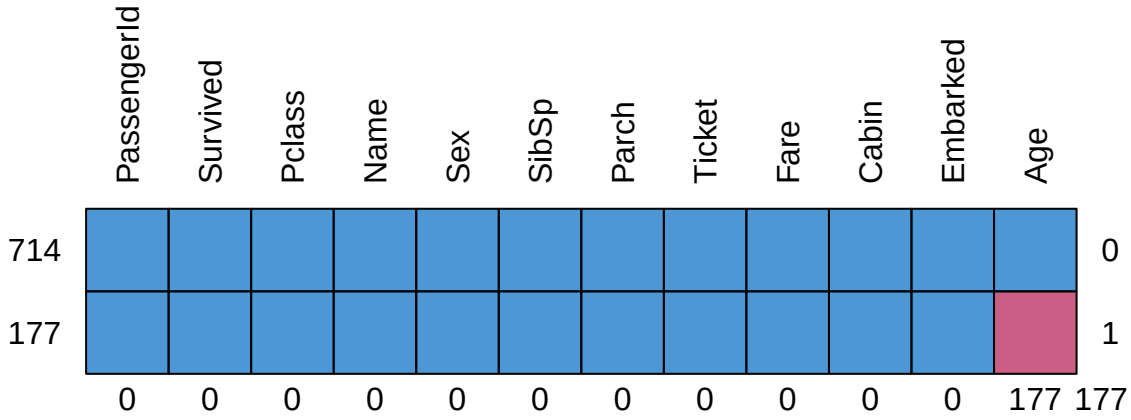
```
# Graphically check the missing data
library("mice")
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
missing_pattern <- md.pattern(titanic_train, rotate.names = TRUE)
```




```
colSums(is.na(titanic_test))
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0          86           0
##       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           1           0           0
```

As we can see we have missing values in Age in the training set and Age, Fare in the test set.

First we tackle the missing Fare, because this is only one value. Let see in wich row it's missing.

```
titanic_test[!complete.cases(titanic_test$Fare),]
```

```
##     PassengerId Pclass                Name  Sex  Age SibSp Parch Ticket Fare
## 153        1044      3 Storey, Mr. Thomas male 60.5     0     0   3701   NA
##     Cabin Embarked
## 153              S
```

As we can see the passenger on row 1044 has an NA Fare value. Now, we need to deal with the NA values in Age column. We will drop these rows using na.omit. Since the PassengerID is a unique identifier for the records, we will drop it. Intuitively the Name, Fare, Embarked and Ticket columns will not decide the survival, so we will drop them as well. So we will select the remaining columns using the select() function from dplyr library:

```
#drop missing data
titanic_test_dropedna <- na.omit(titanic_test)
titanic_train_dropedna <- na.omit(titanic_train)
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
```

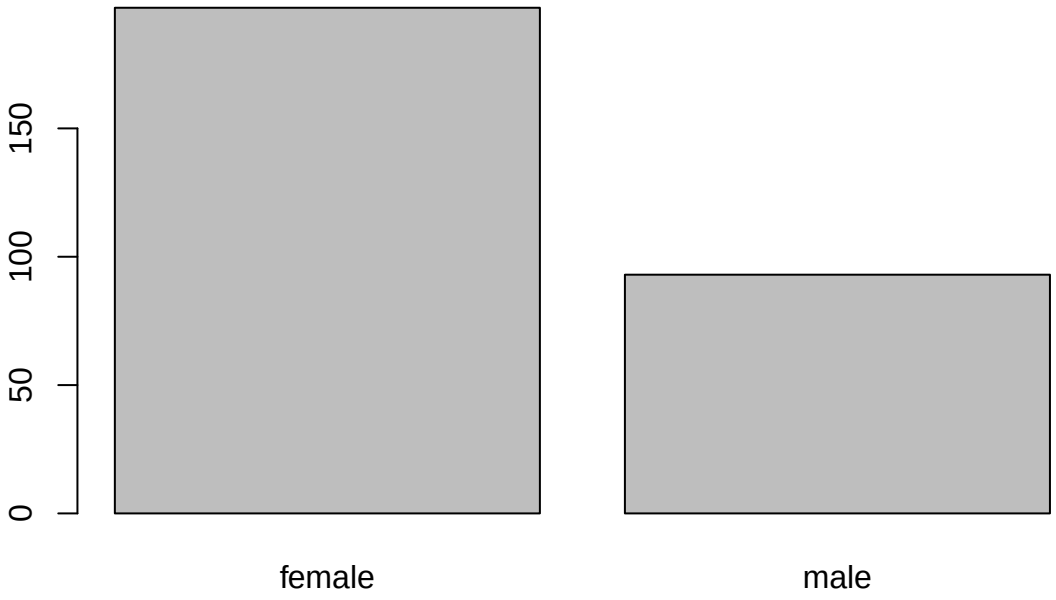
```
titanic_train_1 <-  select(titanic_train_dropedna, Survived, Pclass, Age, Sex, SibSp, Parch)
titanic_test_1 <- select(titanic_train_dropedna, Survived, Pclass, Age, Sex, SibSp, Parch)
```

```
#Alternatively we can use dropping NAs by:

titanic_train[rowSums(is.na(titanic_train)) <= 0,]
# or
library(tidyr)
titanic_train %>% drop_na()
```

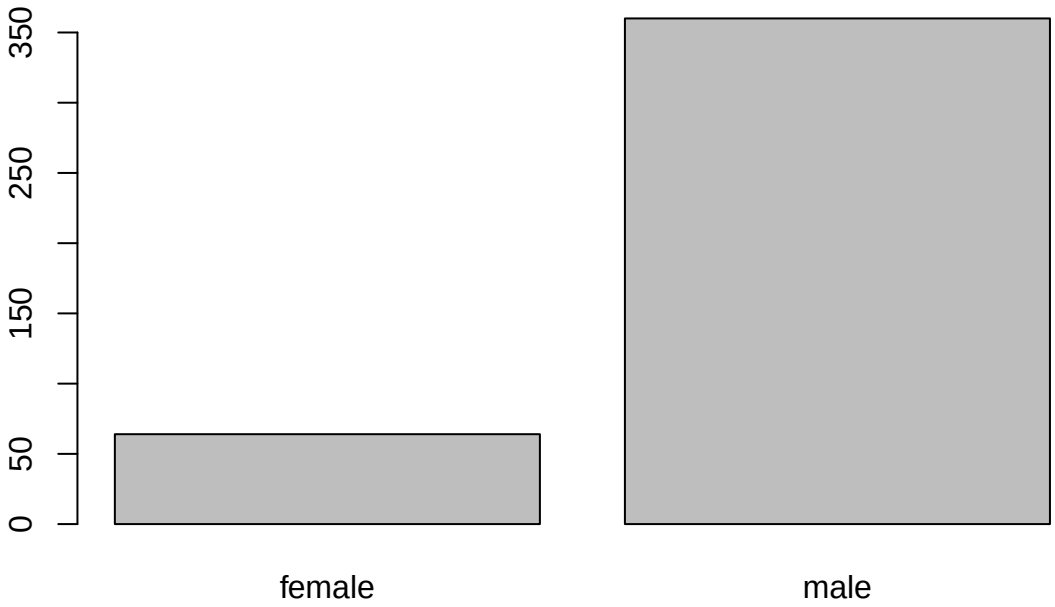
```
#separate data
titanic_survivor = titanic_train_dropedna[titanic_train_dropedna$Survived == 1, ]
titanic_nonsurvivor = titanic_train_dropedna[titanic_train_dropedna$Survived == 0, ]
```

Data Visualization

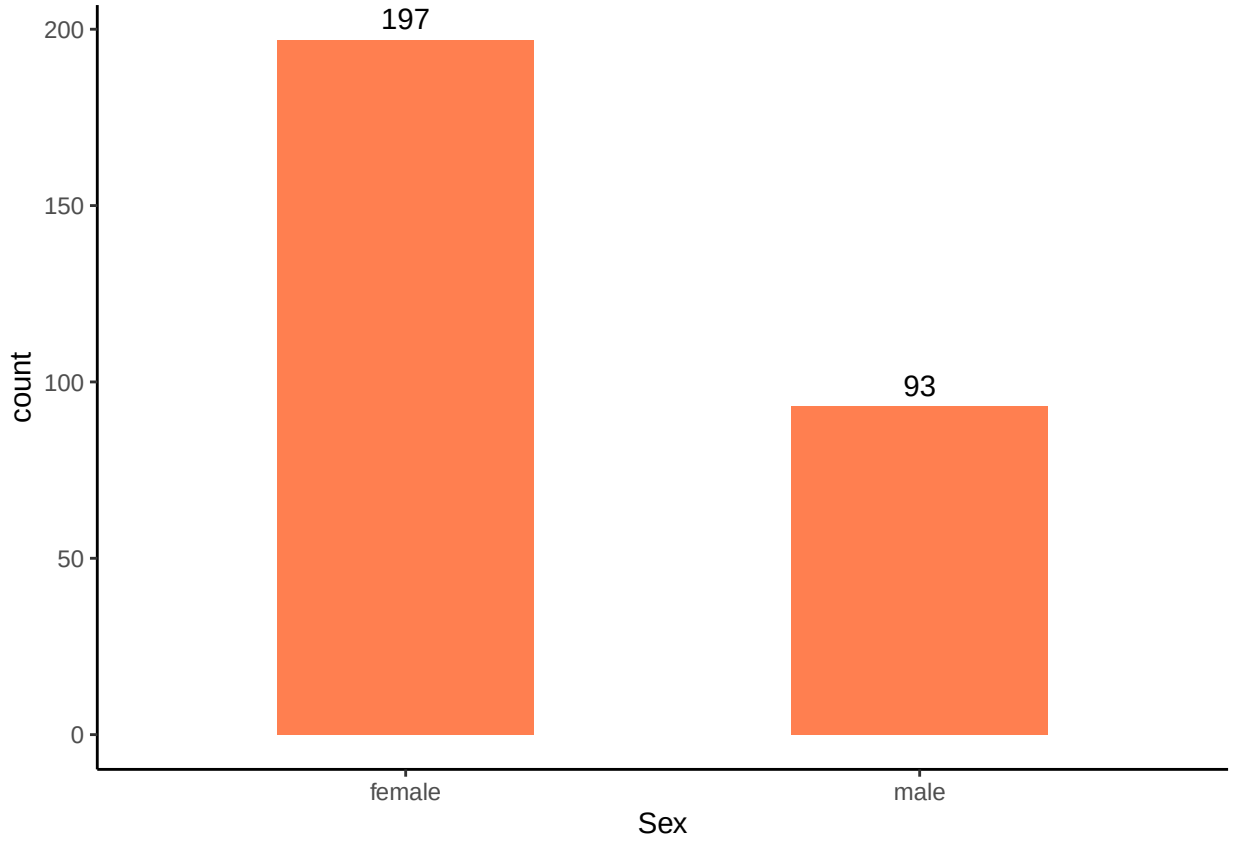
```
#barchart
barplot(table(titanic_survivor$Sex))
```



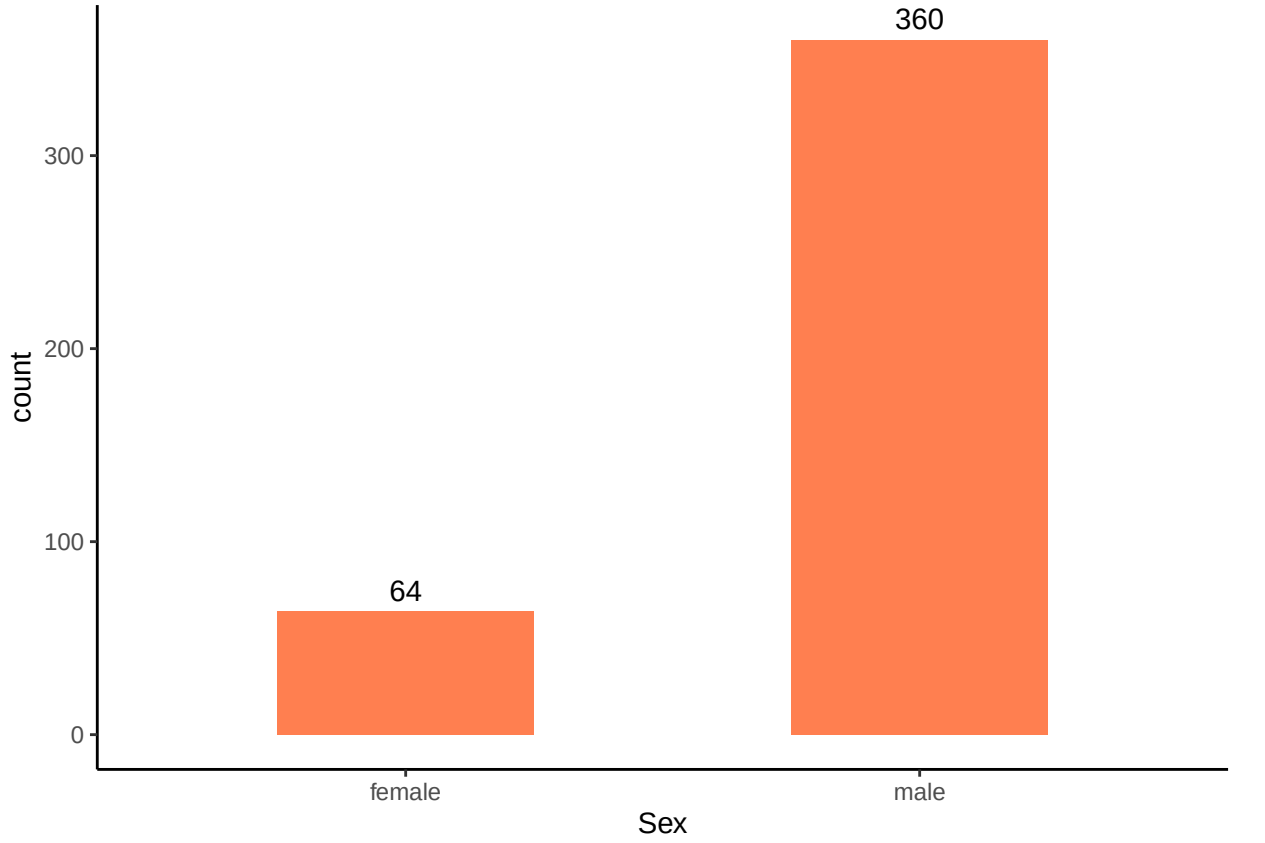

```
barplot(table(titanic_nonsurvivor$Sex))
```



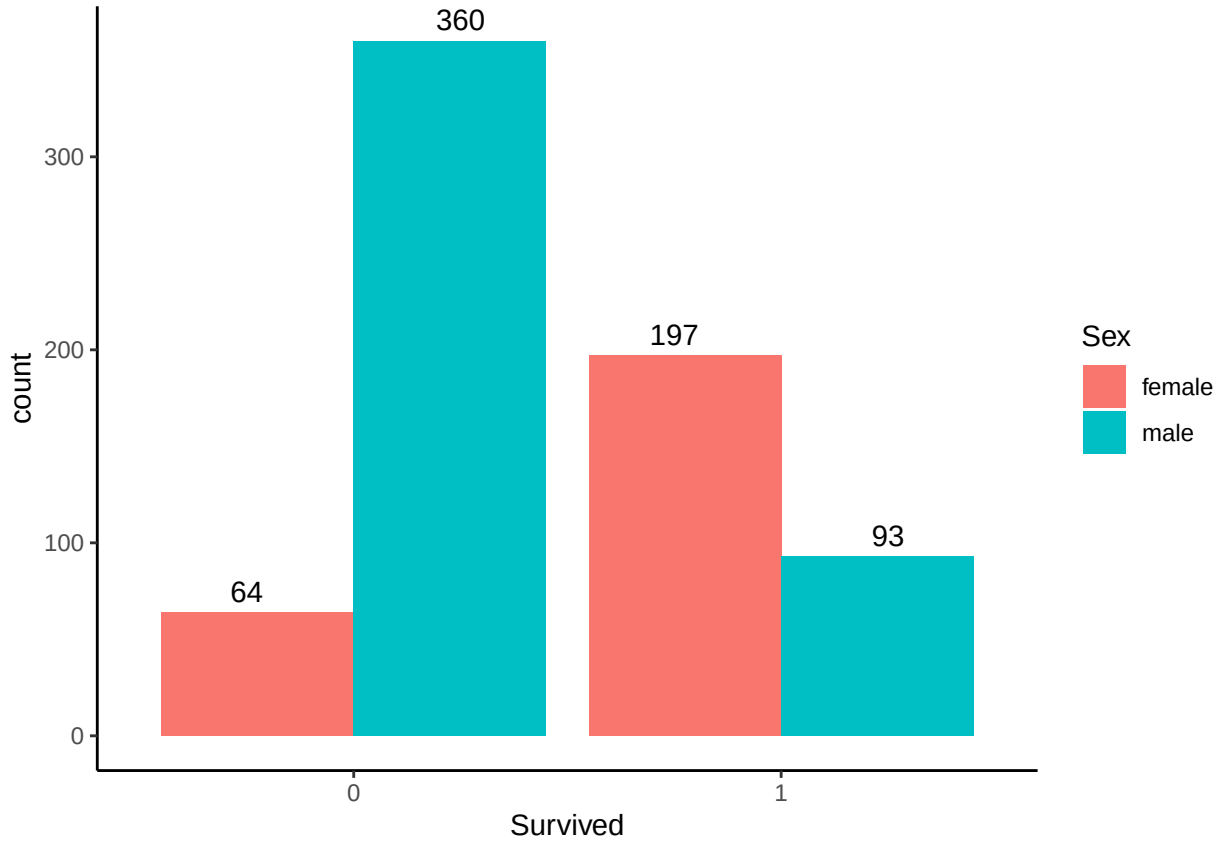

```
## number of  survivals by Sex

library(ggplot2)
ggplot(titanic_survivor, aes(x = Sex)) +
  geom_bar(width=0.5, fill = "coral") +
  geom_text(stat='count', aes(label=stat(count)), vjust=-0.5) +
  theme_classic()
```



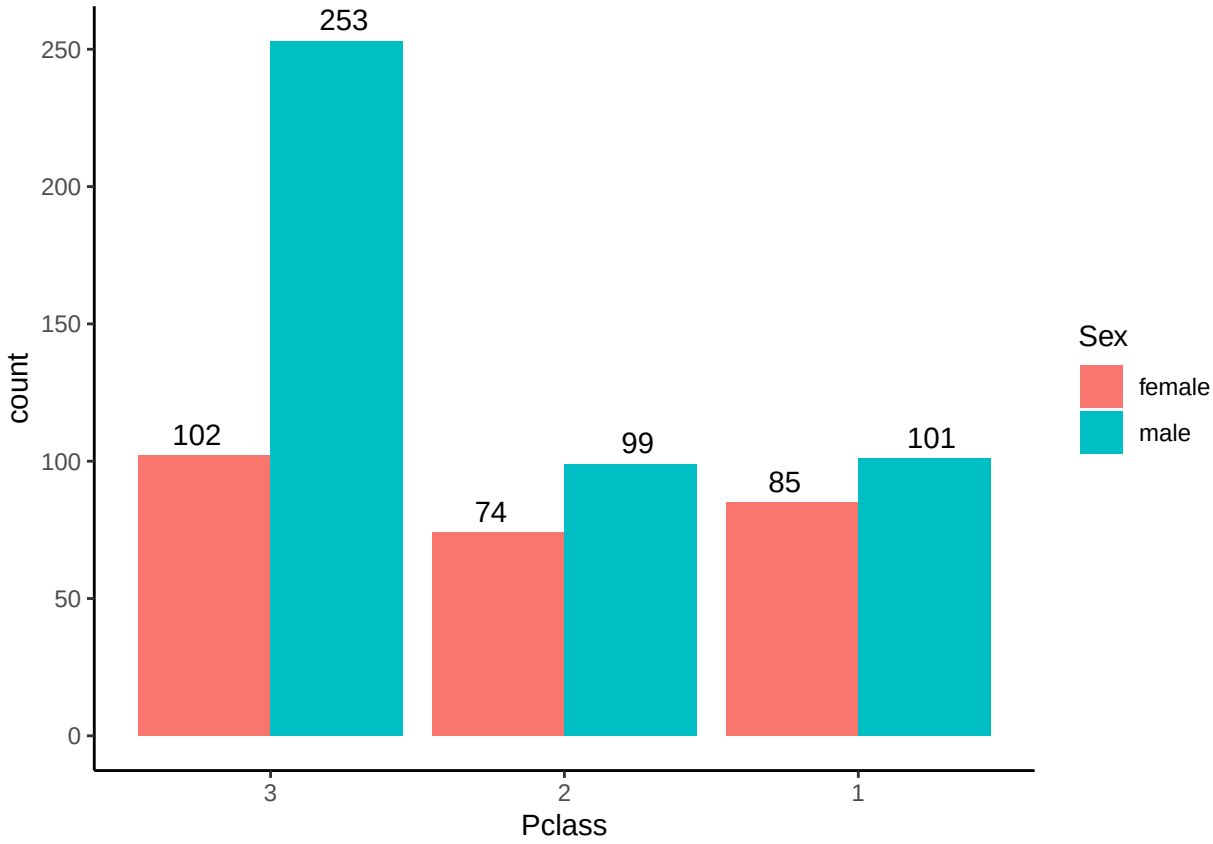
```
## number of  Non survivals by Sex

library(ggplot2)
ggplot(titanic_nonsurvivor, aes(x = Sex)) +
  geom_bar(width=0.5, fill = "coral") +
  geom_text(stat='count', aes(label=stat(count)), vjust=-0.5) +
  theme_classic()
```




```
ggplot(titanic_train_dropedna, aes(x = Survived, fill=Sex)) +
 geom_bar(position = position_dodge()) +
 geom_text(stat='count',
           aes(label=stat(count)),
           position = position_dodge(width=1), vjust=-0.5)+
 theme_classic()
```

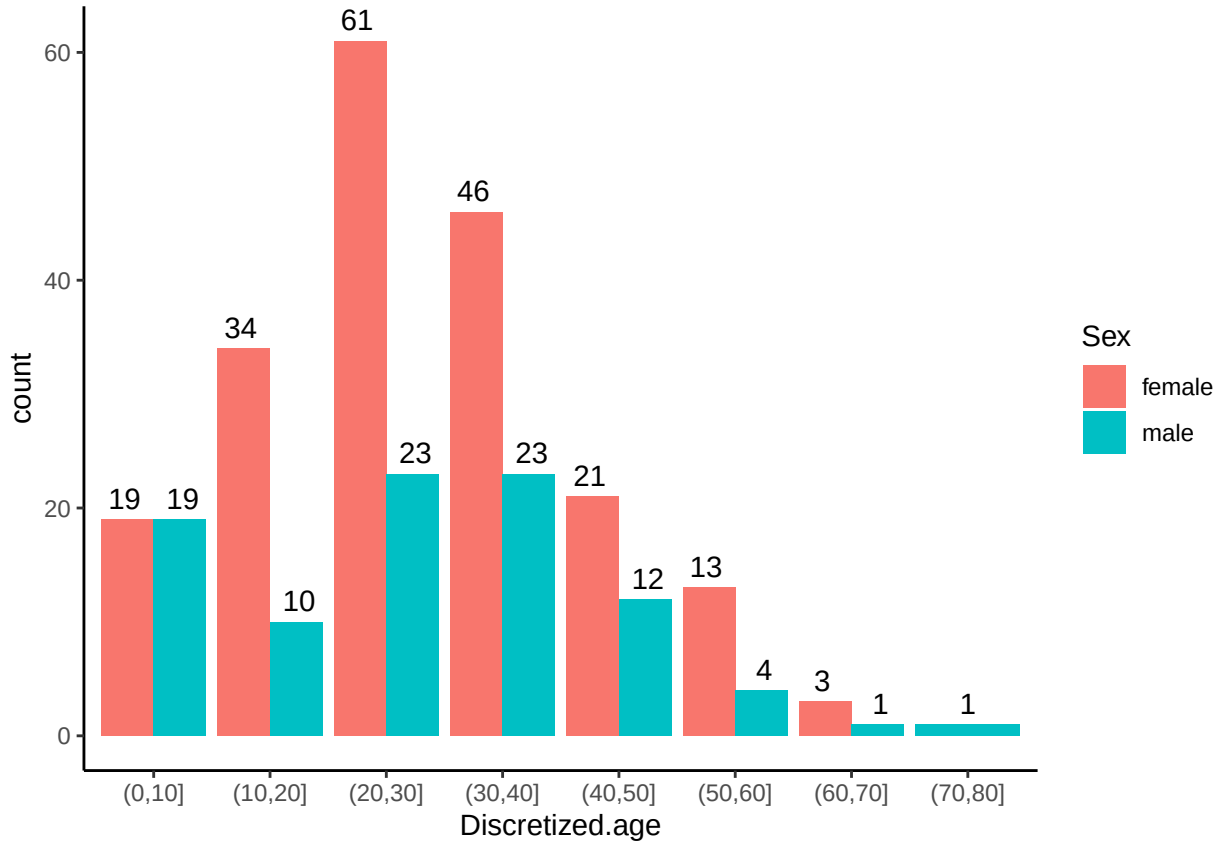



```
ggplot(titanic_train_dropedna, aes(x = Pclass, fill=Sex)) +
 geom_bar(position = position_dodge()) +
 geom_text(stat='count',
           aes(label=stat(count)),
           position = position_dodge(width=1), vjust=-0.5)+
 theme_classic()
```



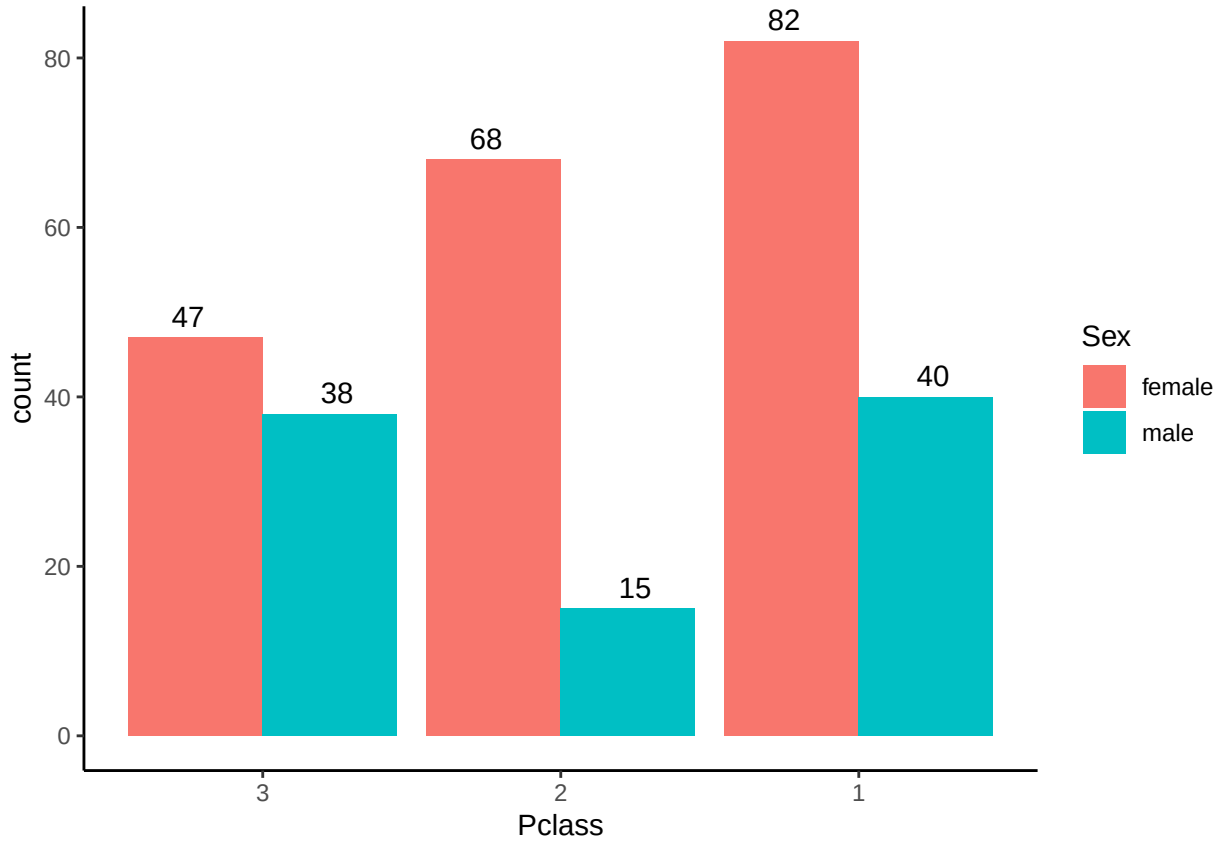
Here we have created a temporary attribute called Discretized.age which groups the ages with a span of 10 years. We discretize the age using the cut() function and specify the cuts in a vector. The temporary attribute it discarded after plotting. Most of the patients that died during hospitalization are in the age range from 70-80 years old.

```
#Discretize age to plot survival
titanic_survivor$Discretized.age = cut(titanic_survivor$Age, c(0,10,20,30,40,50,60,70,80,100))
# Plot discretized age
ggplot(titanic_survivor, aes(x = Discretized.age, fill=Sex)) +
  geom_bar(position = position_dodge()) +
  geom_text(stat='count', aes(label=stat(count)), position = position_dodge(width=1), vjust=-0.5)+
  theme_classic()
```

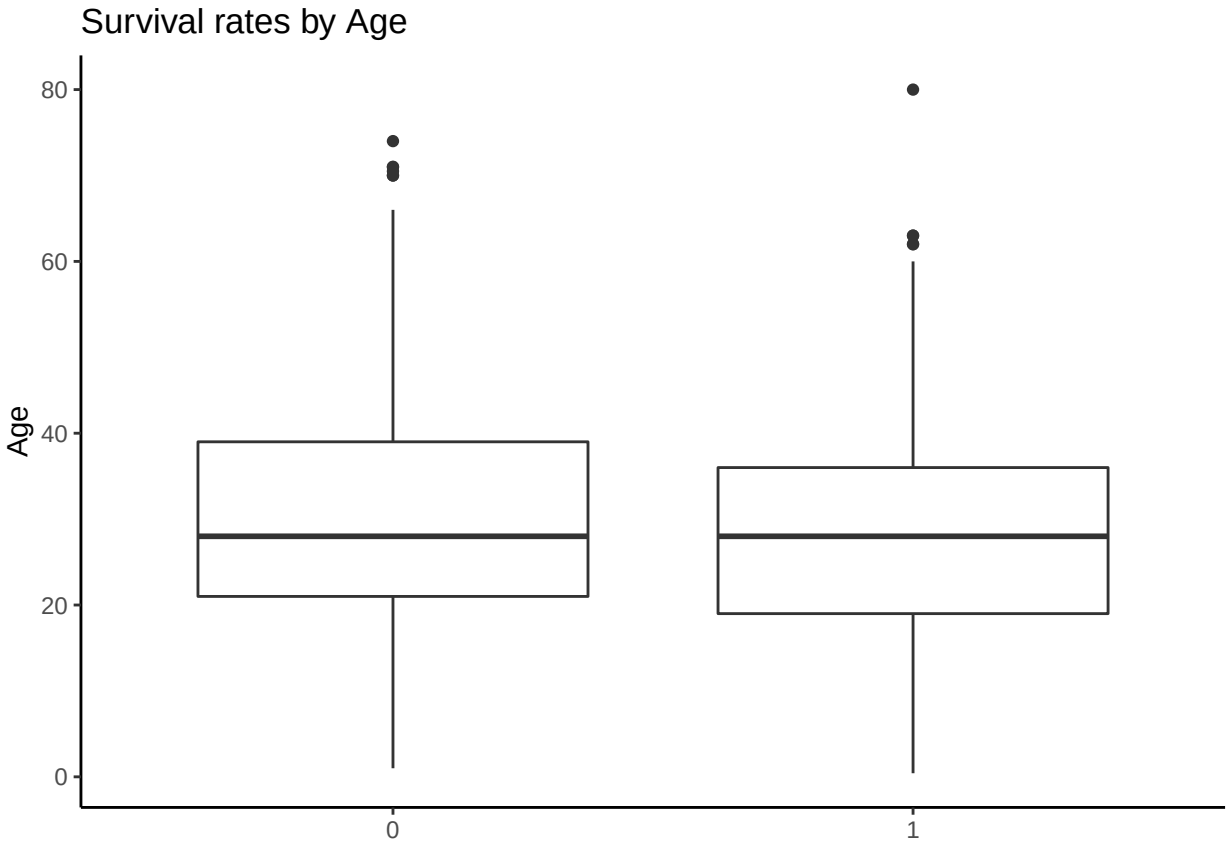



```
#data.frame$Discretized.age = NULL
```

```
ggplot(titanic_survivor, aes(x = Pclass, fill=Sex)) +
 geom_bar(position = position_dodge()) +
 geom_text(stat='count',
           aes(label=stat(count)),
           position = position_dodge(width=1), vjust=-0.5)+
 theme_classic()
```




```
# Boxplot
titanic_train_dropedna %>%
  ggplot(aes(x = Survived, y = Age)) +
  geom_boxplot() +
  theme_classic() +
  labs(title = "Survival rates by Age", x = NULL)
```



Passengers who survived seems to have a lower median age.