LEARN SIMPLE R IMPLIMENTATIONS ON DATA WITH THE TITANIC DATA SET

MUTHAMA KELVIN MUTUKU

2024-07-02

Contents

WHAT IS EXPECTED OF THE LEARNER.

At the end of the study the learner should be able to:

1. Call data in csv form.
   - view the data set.
   - show and explain the structure of a given data set.
2. Restore all variables to their required data types.
3. Draw insights from the data set given.
4. Visualize the data and draw insights in various plots.
5. model data accordingly; fitting the data to the correct model.

lets begin.

In this study we will use the **TIRANIC DATA SET**. The Titanic dataset, often used for machine learning and data analysis projects, typically includes various attributes about the passengers and details about their journey. The most commonly used version is available on Kaggle, and it usually includes the following columns:

1. **PassengerId**: Unique identifier for each passenger.
2. **Survived**: Survival status (0 = No, 1 = Yes).
3. **Pclass**: Ticket class (1 = 1st, 2 = 2nd, 3 = 3rd).
4. **Name**: Passenger's name.
5. **Sex**: Gender of the passenger.
6. **Age**: Age of the passenger.
7. **SibSp**: Number of siblings or spouses aboard the Titanic.
8. **Parch**: Number of parents or children aboard the Titanic.
9. **Ticket**: Ticket number.
10. **Fare**: Fare paid for the ticket.
11. **Cabin**: Cabin number (if available).

12. **Embarked**: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton).

This data set allows for various types of analyses and modeling, such as predicting survival, exploring correlations, and performing feature engineering. Is there a specific ana lysis or task you have in mind with the Titanic dataset?

1. Predicting Survival (Classification Task)

- **Goal**: Build a model to predict whether a passenger survived or not.
- **Expected Result**: A classification model (e.g., logistic regression, decision tree, random forest, etc.) with performance metrics such as accuracy, precision, recall, and F1-score. An accuracy around 70-80% is often considered reasonable for this dataset.

2. Data Exploration and Visualization

- **Goal**: Understand the dataset through exploratory data analysis (EDA).
- **Expected Result**: Insights and visualizations such as:
  - Distribution of passengers by class, gender, and age.
  - Survival rates by different features (e.g., gender, class, age).
  - Correlations between different features and survival.

3. Feature Engineering

- **Goal**: Create new features that could improve model performance.
- **Expected Result**: New features such as:
  - Family size (combining SibSp and Parch).
  - Title extracted from the Name column.
  - Age group bins.
  - Fare per person (Fare divided by the number of people in the same ticket).

4. Model Evaluation and Comparison

- **Goal**: Compare different models to find the best performing one.
- **Expected Result**: Performance metrics (accuracy, precision, recall, F1-score, ROC-AUC) for different models and a discussion of the best model based on these metrics.

5. Deployment

- **Goal**: Deploy the model for practical use.
- **Expected Result**: A deployed model that can take new passenger data as input and predict the survival outcome. This could be done using a web application, API, or other means.

Example Workflow:

1. **Data Cleaning**: Handle missing values, correct data types, etc.
2. **EDA**: Visualize and summarize key patterns and relationships.
3. **Feature Engineering**: Create and select features that improve model performance.
4. **Model Building**: Train various models and tune hyperparameters.
5. **Model Evaluation**: Compare models and select the best one.
6. **Conclusion**: Summarize findings and insights.

Calling the required libraries.

```
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("datasets")
library("ggplot2")
library("graphics")
library("stats")
library("ggeffects")
library("randomForest")
```

```
## randomForest 4.7-1.1

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

##
## Attaching package: 'randomForest'

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

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

creating the dataframes.

```
#calling the dataset
titanic_train_data <- read.csv('C:/Users/EliteBook/OneDrive/Desktop/DATA SCIENCE PERSONAL PROJECTS/TITA
titanic_test_data <- read.csv('C:/Users/EliteBook/OneDrive/Desktop/DATA SCIENCE PERSONAL PROJECTS/TITAN
```

We can have a view of the datasets for some clear understanding and knowledge of the workflow.

```
#viewing the dataset
head(titanic_train_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              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500              S
## 6           330877  8.4583              Q
```

```
head(titanic_test_data)
```

```
##   PassengerId Pclass                                          Name    Sex  Age
## 1         892      3                              Kelly, Mr. James   male 34.5
## 2         893      3              Wilkes, Mrs. James (Ellen Needs) female 47.0
## 3         894      2                     Myles, Mr. Thomas Francis   male 62.0
## 4         895      3                              Wirz, Mr. Albert   male 27.0
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0
## 6         897      3                    Svensson, Mr. Johan Cervin   male 14.0
##   SibSp Parch  Ticket    Fare Cabin Embarked
## 1     0     0  330911  7.8292              Q
## 2     1     0  363272  7.0000              S
## 3     0     0  240276  9.6875              Q
## 4     0     0  315154  8.6625              S
## 5     1     1 3101298 12.2875              S
## 6     0     0    7538  9.2250              S
```

Viewing the data gives you a general look on what you are dealin with but with the help of **str()** you can clearly understand the datta's structure.

```
str(titanic_train_data)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

You can now tell some essential features of the data according to:

- The dimensions (891 rows and 12 columns).
- The data types (int:integer, chr:character, num:numeric).

```
str(titanic_test_data)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis
##  $ Sex        : chr  "male" "female" "male" "male" ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
```

```
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

Same to the **titanic_test_data** you also can now tell some essential features of the data according to:

- The dimensions (891 rows and 11 columns),
- The data types (int:integer, chr:character, num:numeric).

In addition you can use the **summary()** function to tell features of the dataset.

```
summary(titanic_train_data)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     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
##    Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

```
summary(titanic_test_data)
```

```
##   PassengerId         Pclass          Name               Sex
##  Min.   : 892.0   Min.   :1.000   Length:418         Length:418
##  1st Qu.: 996.2   1st Qu.:1.000   Class :character   Class :character
##  Median :1100.5   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :30.27   Mean   :0.4474   Mean   :0.3923
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000
```

```
##  NA's   :86
##       Fare            Cabin             Embarked
##  Min.   :  0.000   Length:418         Length:418
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 35.627
##  3rd Qu.: 31.500
##  Max.   :512.329
##  NA's   :1
```

With the help of this function we can check the:

- length of a given column(variable).
- Measure of spread of the data:
  - Mean.
  - Median.
  - Quantiles.
  - Range.
  - min.
  - max.

Cleaning the data.

In this process we will have to concider variables with: - missinng values. - outliers. - Wrong datatypes.

1. Transforming the data with levels to factor.

Using the **as.factor()** function we can convert a variable to bieng in levles. eg:

The **Pclass** has three levels which are:

- class 1
- class 2
- class 3

```
#setting factors to categorical data
titanic_train_data$Pclass<-as.factor(titanic_train_data$Pclass)
titanic_train_data$Embarked<-as.factor(titanic_train_data$Embarked)
titanic_train_data$Sex<-as.factor(titanic_train_data$Sex)

titanic_test_data$Pclass<-as.factor(titanic_test_data$Pclass)
titanic_test_data$Embarked<-as.factor(titanic_test_data$Embarked)
titanic_test_data$Sex<-as.factor(titanic_test_data$Sex)
```

If you were to call the **str()** of the data you will see that the type of some variables has changed to factor.

```
str(titanic_train_data)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
```

```
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

You will notice that even though the **survived** varible in **titanic_train_data** has two levels(0 and 1), we have not factored it. This is because we will use it for some analysis and errors call if it is factored.

2. missing values.

For the missing values there various ways of dealing with them, you can choose to:

- Ignore the missing values.
- Impute the missing values.
- Remove the missing values.

All this depends on the dataset you are working on and the insights you need from the data.

Identifying the missing values. We will first check for missing values in all the variables.

```
titanic_train_data %>%
  is.na() %>%
  table()
```

```
## .
## FALSE  TRUE
## 10515   177
```

We can see that we have 177 null values, but we can't tell their specific variables. To do so we can;

```
filter_all(titanic_train_data, any_vars(is.na(.))) %>%
  head()
```

```
##   PassengerId Survived Pclass                          Name    Sex Age SibSp
## 1           6        0      3              Moran, Mr. James   male  NA     0
## 2          18        1      2  Williams, Mr. Charles Eugene   male  NA     0
## 3          20        1      3       Masselmani, Mrs. Fatima female  NA     0
## 4          27        0      3       Emir, Mr. Farred Chehab   male  NA     0
## 5          29        1      3 O'Dwyer, Miss. Ellen "Nellie" female  NA     0
## 6          30        0      3           Todoroff, Mr. Lalio   male  NA     0
##   Parch Ticket    Fare Cabin Embarked
## 1     0 330877  8.4583              Q
## 2     0 244373 13.0000              S
## 3     0   2649  7.2250              C
## 4     0   2631  7.2250              C
## 5     0 330959  7.8792              Q
## 6     0 349216  7.8958              S
```

According to our data the only variable with null values is the **Age** variable. It contains 177 null values.

In addition there is a varable with **empty characters**, and this is the **embarked** variable.

```
titanic_train_data$Embarked %>%
  table()
```

```
## .
##       C   Q   S
##   2 168  77 644
```

According to our data there are two values that have empty characters.

dealing with the missing values. We will impute our data(fill the data using mean, median or mode). Most precisely we will have to use the mean of the data.

Lets begin with variable **Age**.

```
#calculating the mean.
mean.titanic_train_data<- mean(titanic_train_data$Age, na.rm = T)

#Imputing the data.
titanic_train_data$Age<- titanic_train_data%>%
  select(Age) %>%
  apply(c(2), . %>% {ifelse(is.na(.), mean.titanic_train_data, .)})
titanic_test_data$Age<- titanic_test_data%>%
  select(Age) %>%
  apply(c(2), . %>% {ifelse(is.na(.), 29.70, .)})
```

Next, we can work on the missing values in **Embarked**.

```
titanic_train_data[titanic_train_data$Embarked == '', "Embarked"] <- 'S'

titanic_test_data[titanic_test_data$Embarked == '', "Embarked"] <- 'S'
```

Before concluding on the missing values, you will notice that we have left out the missing values in **Cabin**. This is heavy duty but you can draw your conclusions from the internet and other sources. Now that we are done with the missing values, we can proceed to the next part.

Clussering our data into understandable and meaningful subgroups.

For better understanding of the data clustering can be done, where you devide a variable set into small subsets. Eg: the variable **age** is large, but we can devide it into: The elderly, Non-youth, Youth and Children.

```
#creating an age cluser
titanic_train_data<- titanic_train_data %>%
  mutate(age_groups = ifelse(0<=Age & Age<=13, 1, ifelse(14<=Age & Age<=35, 2, ifelse(36<=Age & Age<=60
```

In the data:

- 1 stands for children.
- 2 stands for youth.
- 3 stands for adults.
- 4 stands for elderly.

Other additional clusters are as follows.

```
#Additional age clusers
titanic_train_data<- titanic_train_data %>%
  mutate(age_groups_young = ifelse(0<=Age & Age<=13, "child",ifelse(13<=Age & Age<=35, "youth", "adult"

titanic_train_data<- titanic_train_data %>%
  mutate(age_groups_children = ifelse(0<=Age & Age<=5, "infant",ifelse(5<=Age & Age<=7, "child",ifelse(
```

Grouping the data to help with analysis.

With the help of **groupby()** function, we can get insights from the data as follows.

```
#analysing the data by grouping
titanic_train_data%>%
  group_by(Pclass)%>%
  summarise(mean(Survived))
```

```
## # A tibble: 3 x 2
##   Pclass `mean(Survived)`
##   <fct>             <dbl>
## 1 1                 0.630
## 2 2                 0.473
## 3 3                 0.242
```

using the mean we can tell chances of people surviving according to their **Pclass**. In class one, the chances of survival were .6296296 and those of class two were 0.4728261 while class three had 0.2423625. This interpratation is logic as we expect the first class to be well equiped incase of any damages or accident.

```
titanic_train_data%>%
  group_by(Sex)%>%
  summarise(mean(Survived))
```

```
## # A tibble: 2 x 2
##   Sex    `mean(Survived)`
##   <fct>             <dbl>
## 1 female            0.742
## 2 male              0.189
```

According to the analysis, the females had higher chances of surviving than men. In the incident women were highly concidered and were offered more saving boats than men because men were believed to swim and survive hardships better.

```
titanic_train_data%>%
  group_by(SibSp)%>%
  summarise(mean(Survived))
```

```
## # A tibble: 7 x 2
##   SibSp `mean(Survived)`
##   <int>            <dbl>
## 1     0            0.345
## 2     1            0.536
## 3     2            0.464
## 4     3            0.25
## 5     4            0.167
## 6     5            0
## 7     8            0
```

According to the analysis, it seems there were people with up to 8 siblings and that the chances of surval are not well defined but bigger families never survived. Maybe as they tried to save each other the more they died.

```
titanic_train_data%>%
  group_by(Parch)%>%
  summarise(mean(Survived))
```

```
## # A tibble: 7 x 2
##   Parch `mean(Survived)`
##   <int>            <dbl>
## 1     0            0.344
## 2     1            0.551
## 3     2            0.5
## 4     3            0.6
## 5     4            0
## 6     5            0.2
## 7     6            0
```

Same as the **SibSp**, it seems there were families with up to 6 children and that the chances of surval are not well defined but bigger families never survived. Maybe as they tried to save each other the more they died.

```
titanic_train_data %>%
  group_by(age_groups_children)%>%
  summarise( mean(Survived))
```

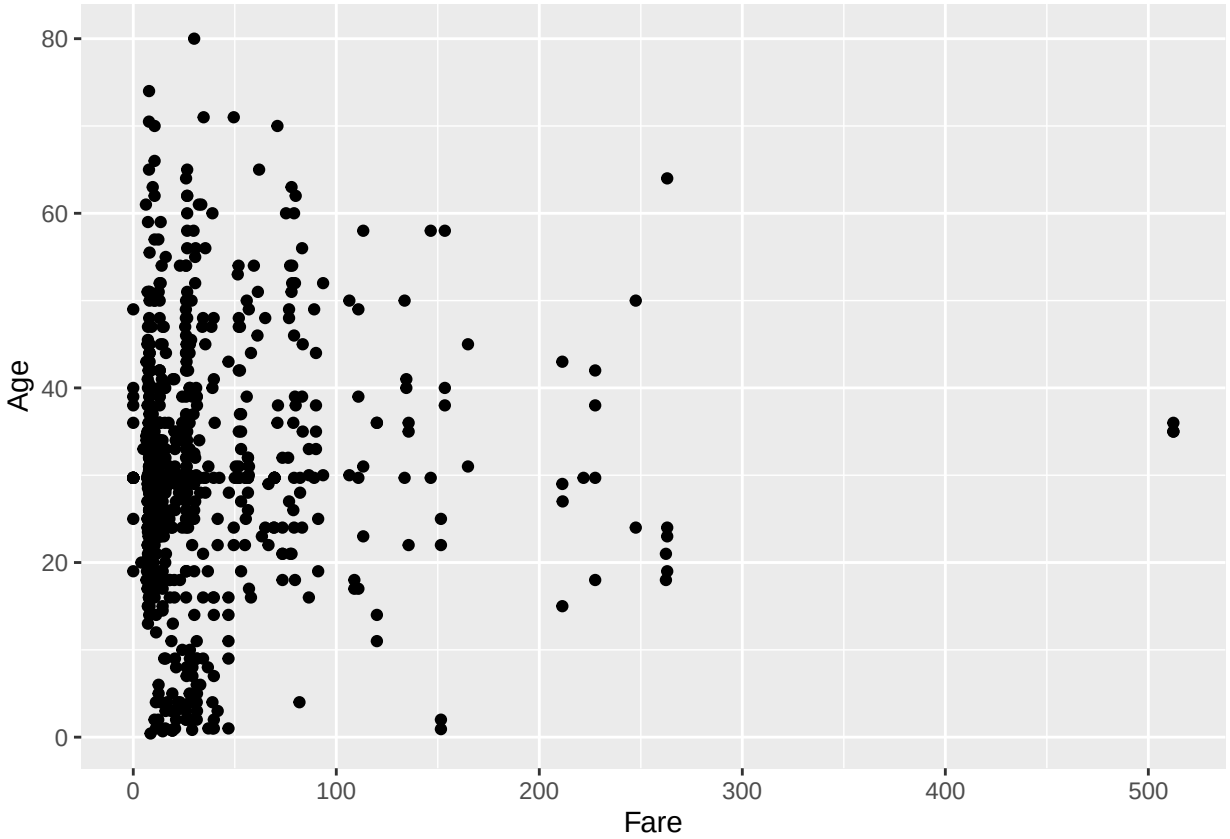
```
## # A tibble: 4 x 2
##   age_groups_children[,"Age"] `mean(Survived)`
##   <chr>                                 <dbl>
## 1 child                                 0.5
## 2 infant                                0.705
## 3 non-child                             0.366
## 4 teen                                  0.381
```

```
#setting factors to categorical data
titanic_train_data$Survived<- as.factor(titanic_train_data$Survived)
```

Visualising the data for insights

In this part we will ensure that we plot variables to see their relationships and give insights

```
titanic_train_data %>%
  select(Survived, Pclass,, Fare, Ticket ,age_groups, Sex,Age) %>%
  ggplot(data=) +
  geom_point(mapping =  aes(x=Fare, y= Age), stat = "identity")
```



According to the plot, we can say that the most paid fare ranges between 0-100 and the **age** between 15-50 had high numbers aboard. Two outlers are detected at the far end of the **x-axis**, measures should be carried

inorder to decide on how to deal with them.

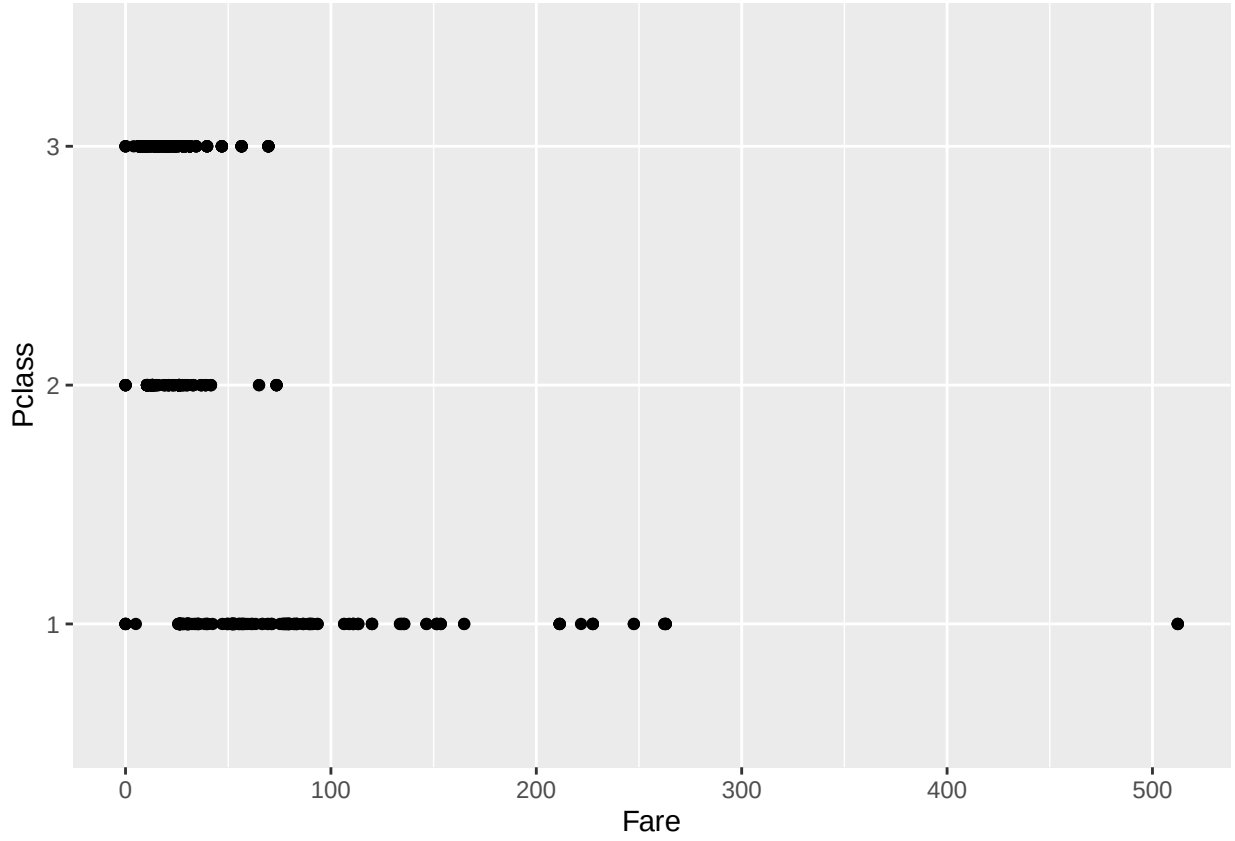
To check for the values we can do the following.

```
filter_all(titanic_train_data, any_vars(titanic_train_data$Fare>=500))
```

```
##   PassengerId Survived Pclass                                  Name    Sex Age
## 1         259        1      1                    Ward, Miss. Anna female  35
## 2         680        1      1 Cardeza, Mr. Thomas Drake Martinez   male  36
## 3         738        1      1              Lesurer, Mr. Gustave J   male  35
##   SibSp Parch   Ticket     Fare       Cabin Embarked Age   Age       Age
## 1     0     0 PC 17755 512.3292                   C   2 youth non-child
## 2     0     1 PC 17755 512.3292 B51 B53 B55       C   3 adult non-child
## 3     0     0 PC 17755 512.3292        B101       C   2 youth non-child
```

You will notice that they are three points with the same fare of 512.3292, embarked destination(c), Pclass(1) and ticket number(PC 17755).

```
titanic_train_data %>%
  select(Survived, Pclass,, Fare, Ticket ,age_groups, Sex,Age) %>%
  ggplot(data=) +
  geom_point(mapping =  aes(x=Fare, y= Pclass), stat = "identity")
```



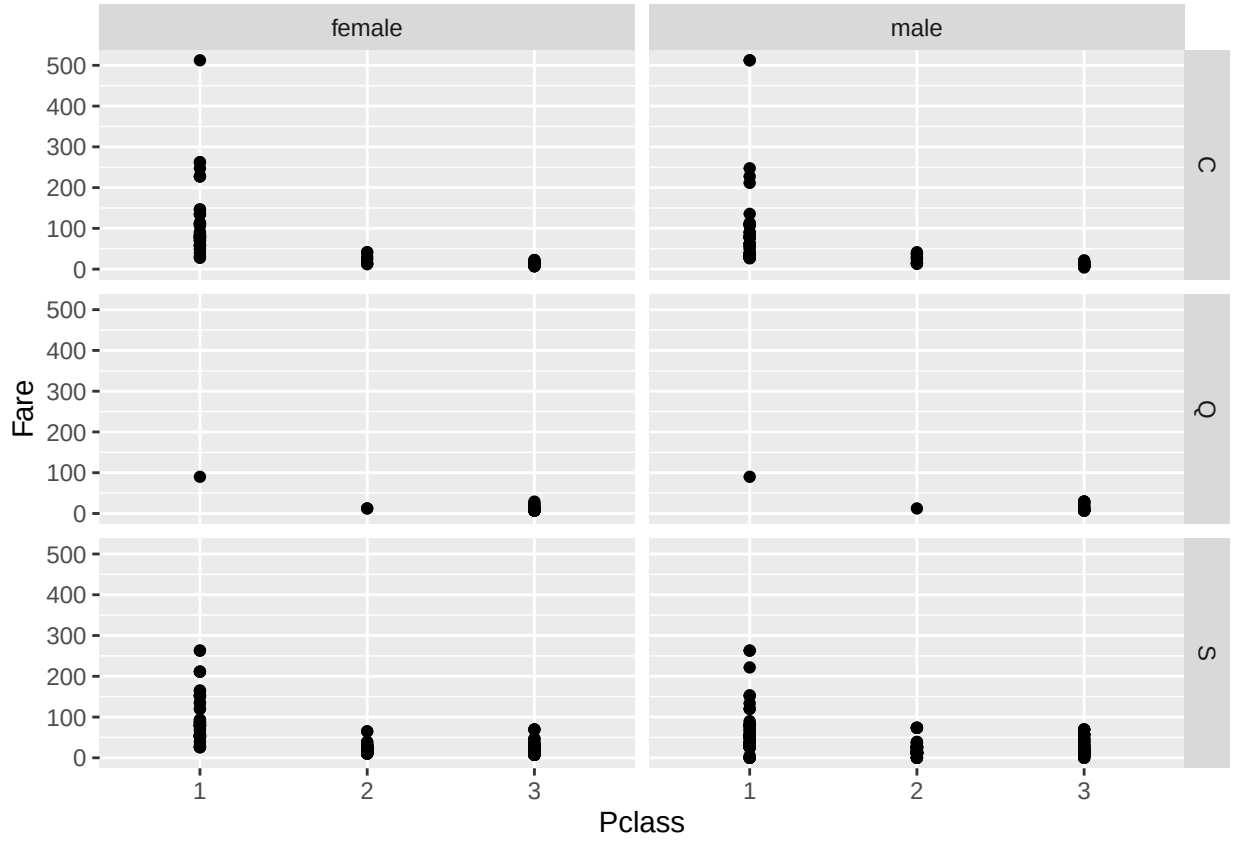
Further investigation can be done as we see people in different **Pclasses** paying the same fare.

```
filter_all(titanic_train_data, any_vars(titanic_train_data$Pclass==1 & titanic_train_data$Fare<500)) %>%
  head()
```

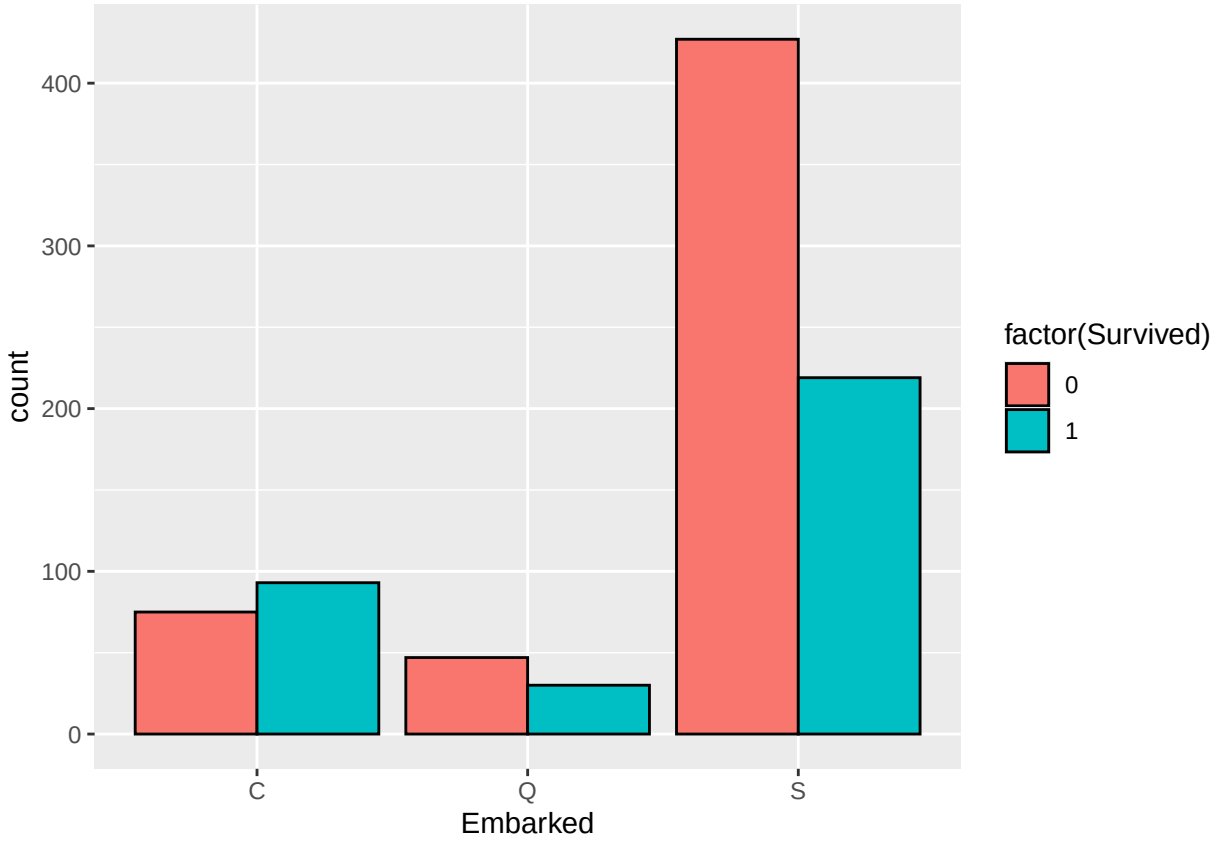
```
##   PassengerId Survived Pclass
## 1           2        1      1
```

```
## 2           4        1      1
## 3           7        0      1
## 4          12        1      1
## 5          24        1      1
## 6          28        0      1
##                                                 Name    Sex Age SibSp Parch
## 1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 2        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 3                             McCarthy, Mr. Timothy J   male  54     0     0
## 4                           Bonnell, Miss. Elizabeth female  58     0     0
## 5                     Sloper, Mr. William Thompson    male  28     0     0
## 6                    Fortune, Mr. Charles Alexander   male  19     3     2
##     Ticket     Fare       Cabin Embarked Age   Age       Age
## 1 PC 17599  71.2833         C85        C   3 adult non-child
## 2   113803  53.1000        C123        S   2 youth non-child
## 3    17463  51.8625         E46        S   3 adult non-child
## 4   113783  26.5500        C103        S   3 adult non-child
## 5   113788  35.5000          A6        S   2 youth non-child
## 6    19950 263.0000 C23 C25 C27        S   2 youth non-child
```

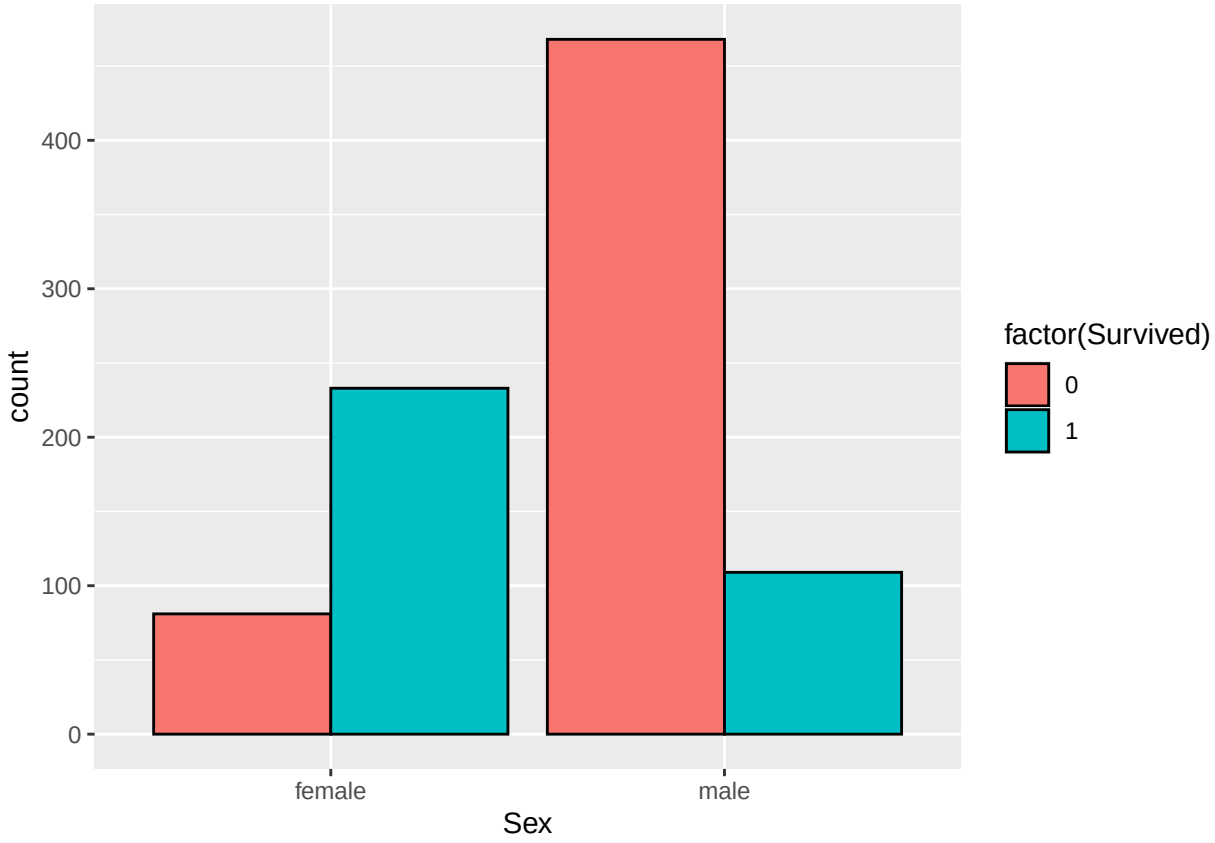
```
titanic_train_data %>%
  select(Survived, Embarked, Pclass, Fare, Ticket ,age_groups, Sex,Age) %>%
  ggplot(data=) +
  geom_point(mapping =  aes(x=Pclass, y = Fare), stat = "identity") +
  facet_grid( Embarked ~ Sex)
```



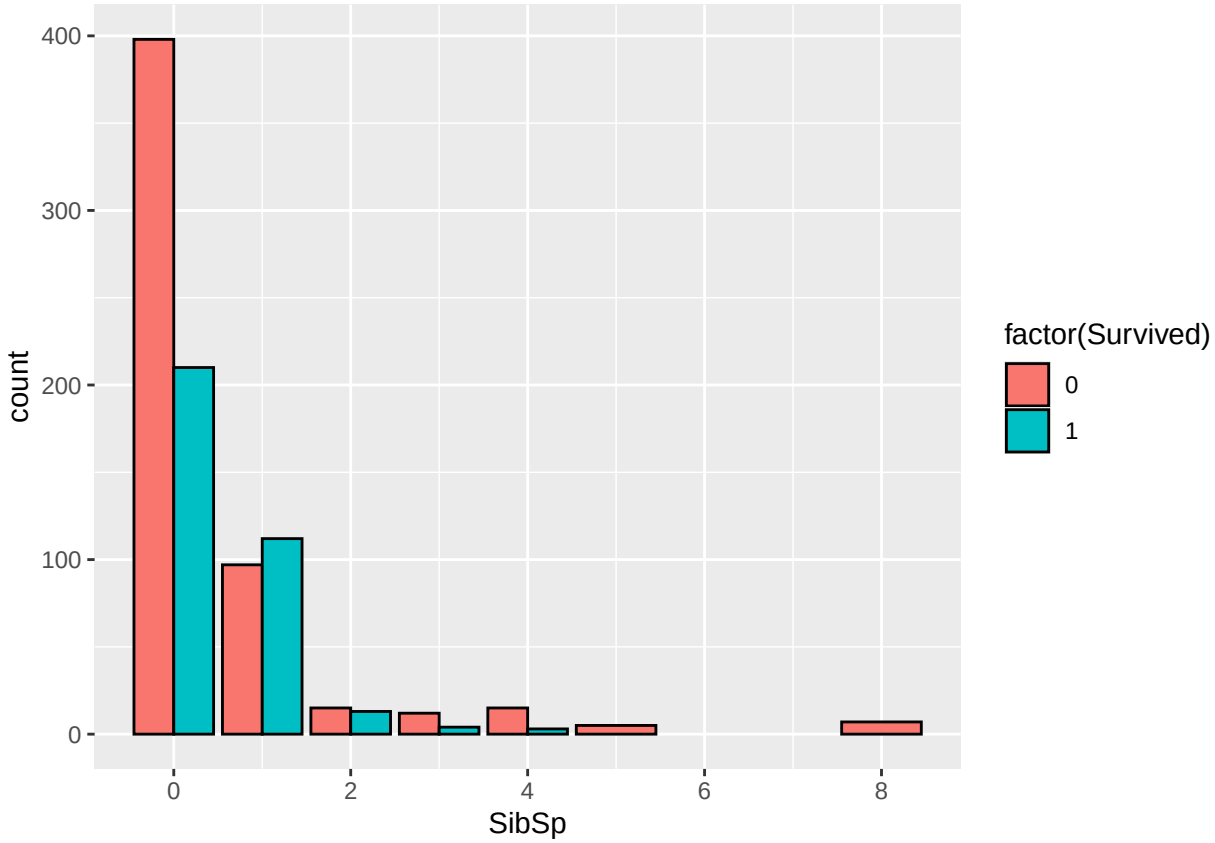
```
titanic_train_data %>%
  select(Survived, Embarked, Pclass, age_groups, Sex,Age) %>%
  ggplot(data=) +
  geom_bar(mapping =  aes(x=Embarked, fill= factor(Survived)), color = "black", position = position_dod
```



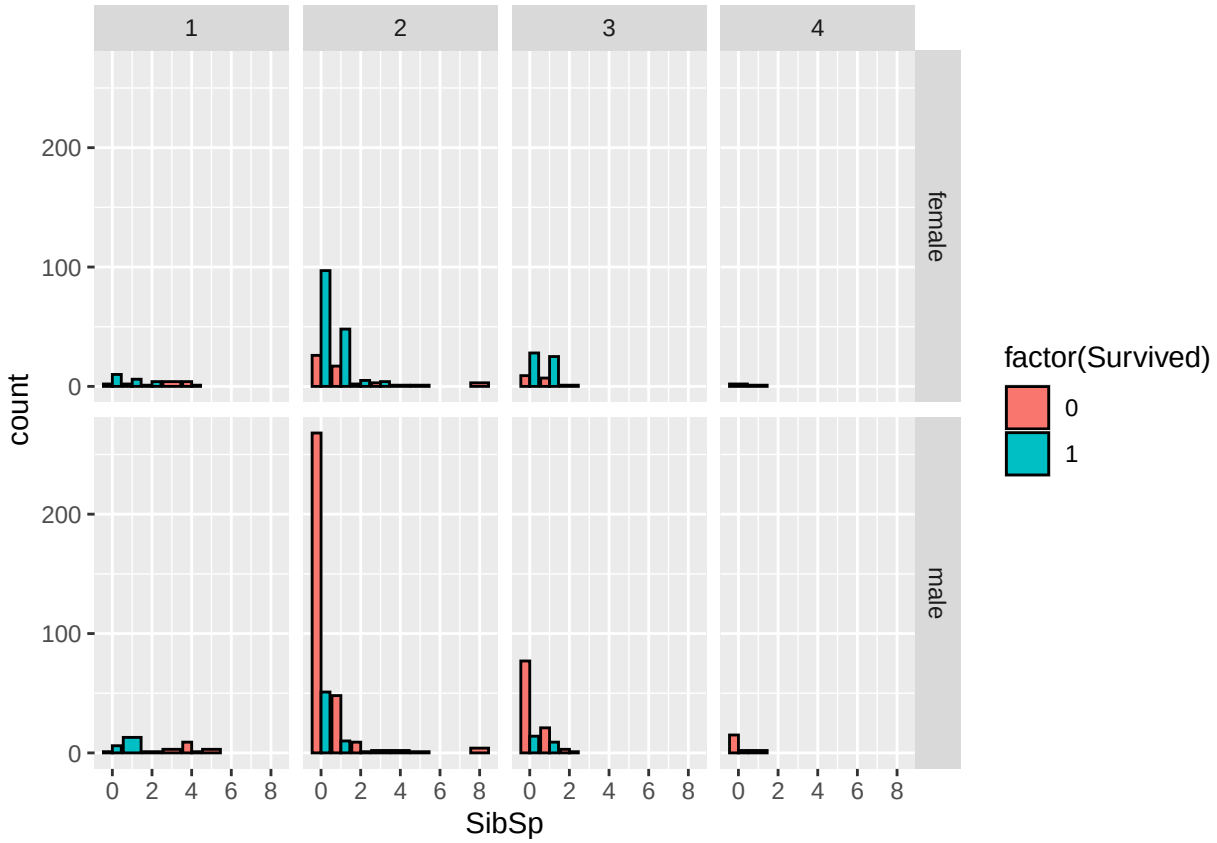

```
titanic_train_data %>%
  select(Survived, Embarked, Pclass, age_groups, Sex,Age) %>%
  ggplot(data=) +
  geom_bar(mapping =  aes(x=Sex, fill= factor(Survived)), color = "black", position = position_dodge())
```



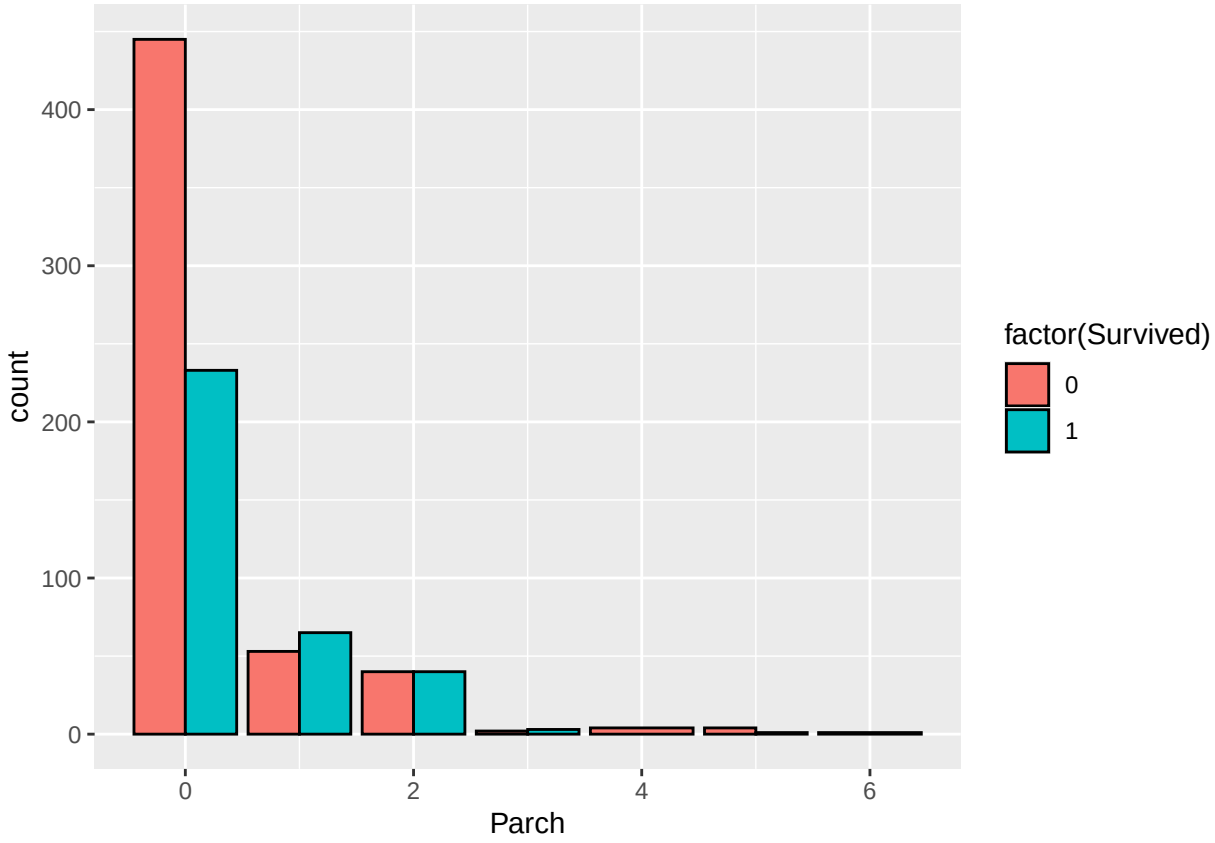

```
titanic_train_data %>%
  select(Survived,SibSp) %>%
  ggplot(data=) +
  geom_bar(mapping =  aes(x=SibSp, fill= factor(Survived)), color = "black", position = position_dodge()
```



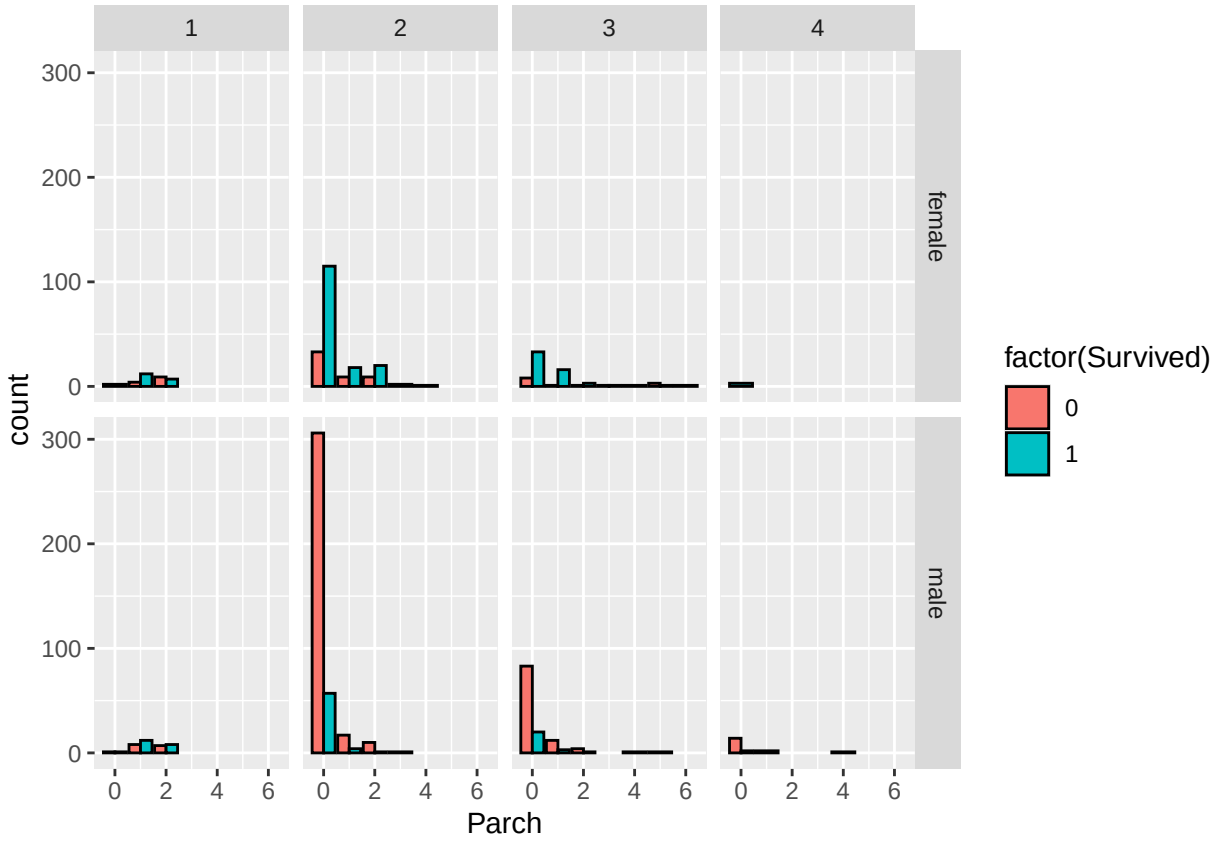

```
titanic_train_data %>%
  select(Survived, SibSp, Sex, age_groups) %>%
  ggplot(data=) +
  geom_bar(mapping =  aes(x=SibSp, fill= factor(Survived)), color = "black", position = position_dodge()
  facet_grid(Sex~age_groups)
```



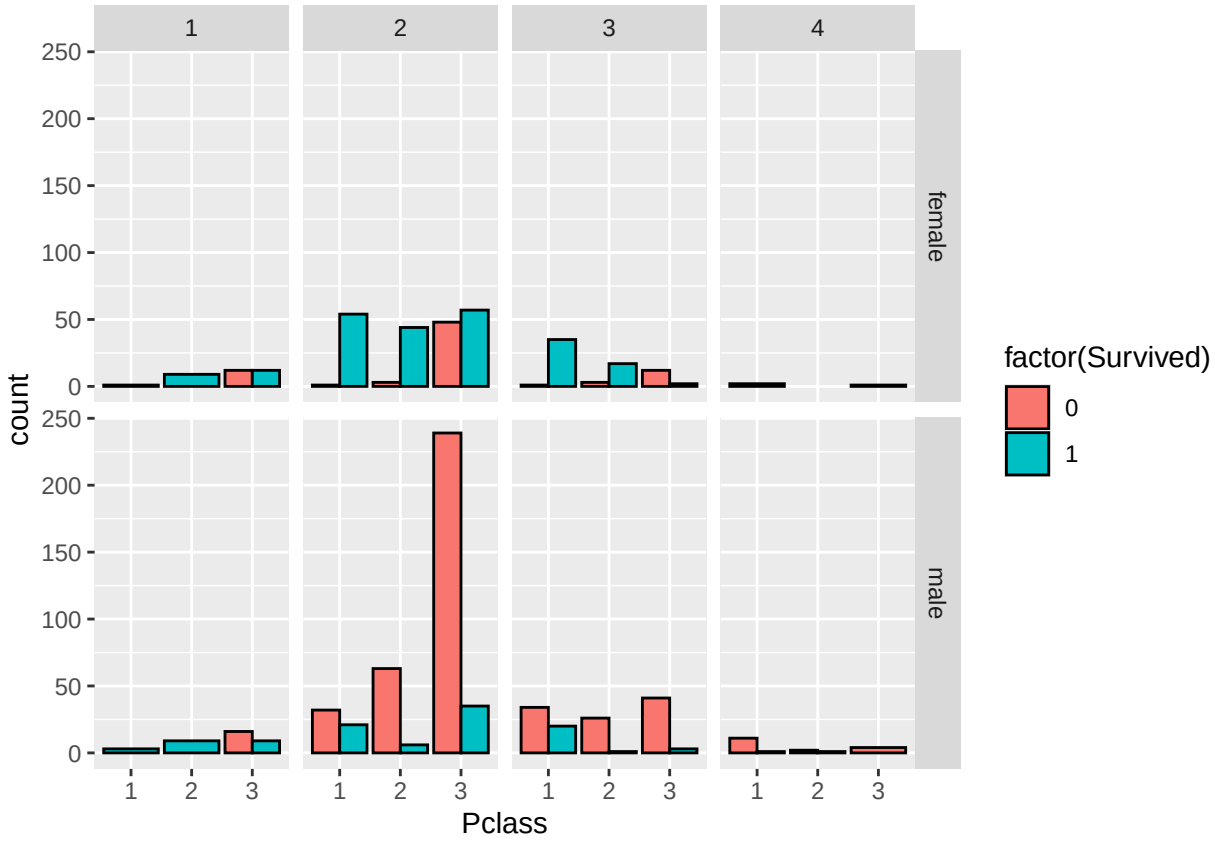

```
titanic_train_data %>%
  select(Survived, Parch) %>%
  ggplot(data=) +
  geom_bar(mapping =  aes(x=Parch, fill= factor(Survived)), color = "black", position = position_dodge()
```



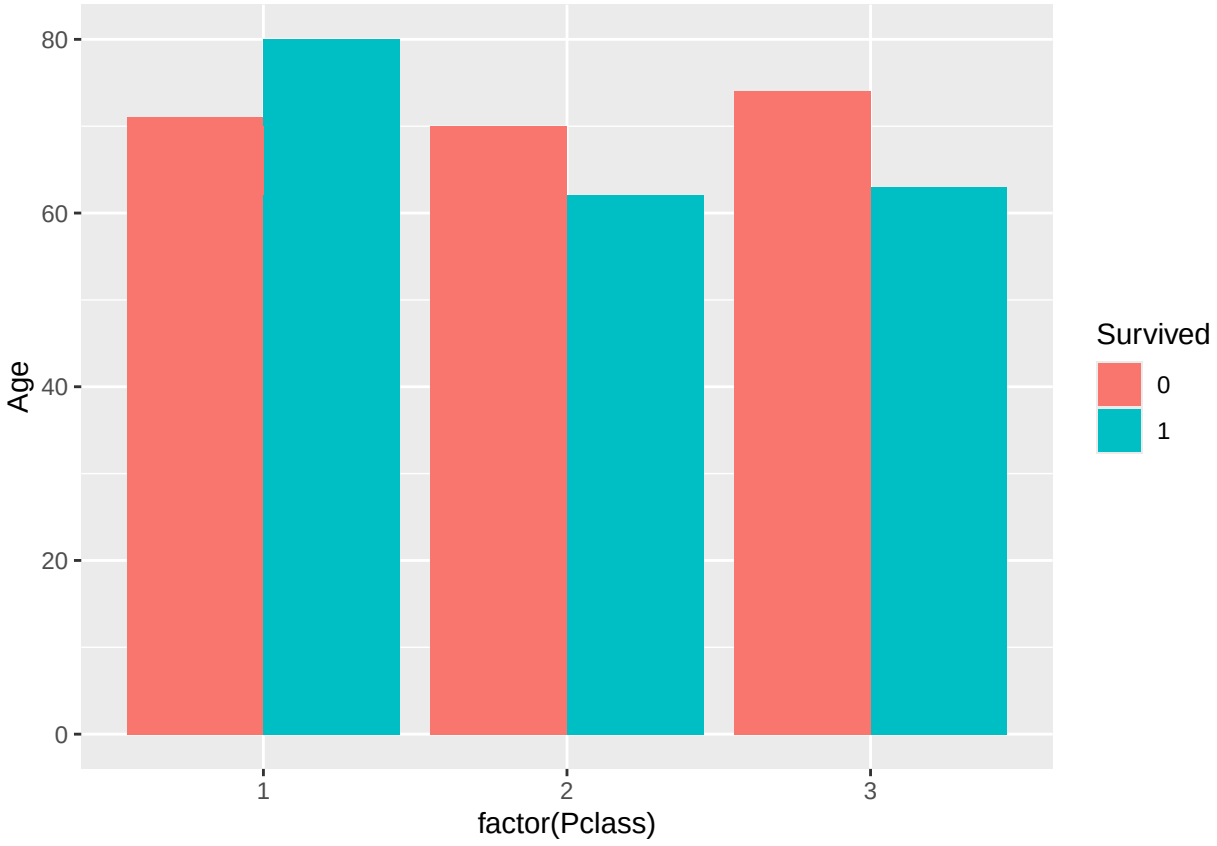

```
titanic_train_data %>%
  select(Survived, Parch, Sex, age_groups) %>%
  ggplot(data=) +
  geom_bar(mapping =  aes(x=Parch, fill= factor(Survived)), color = "black", position = position_dodge()
  facet_grid(Sex~age_groups)
```



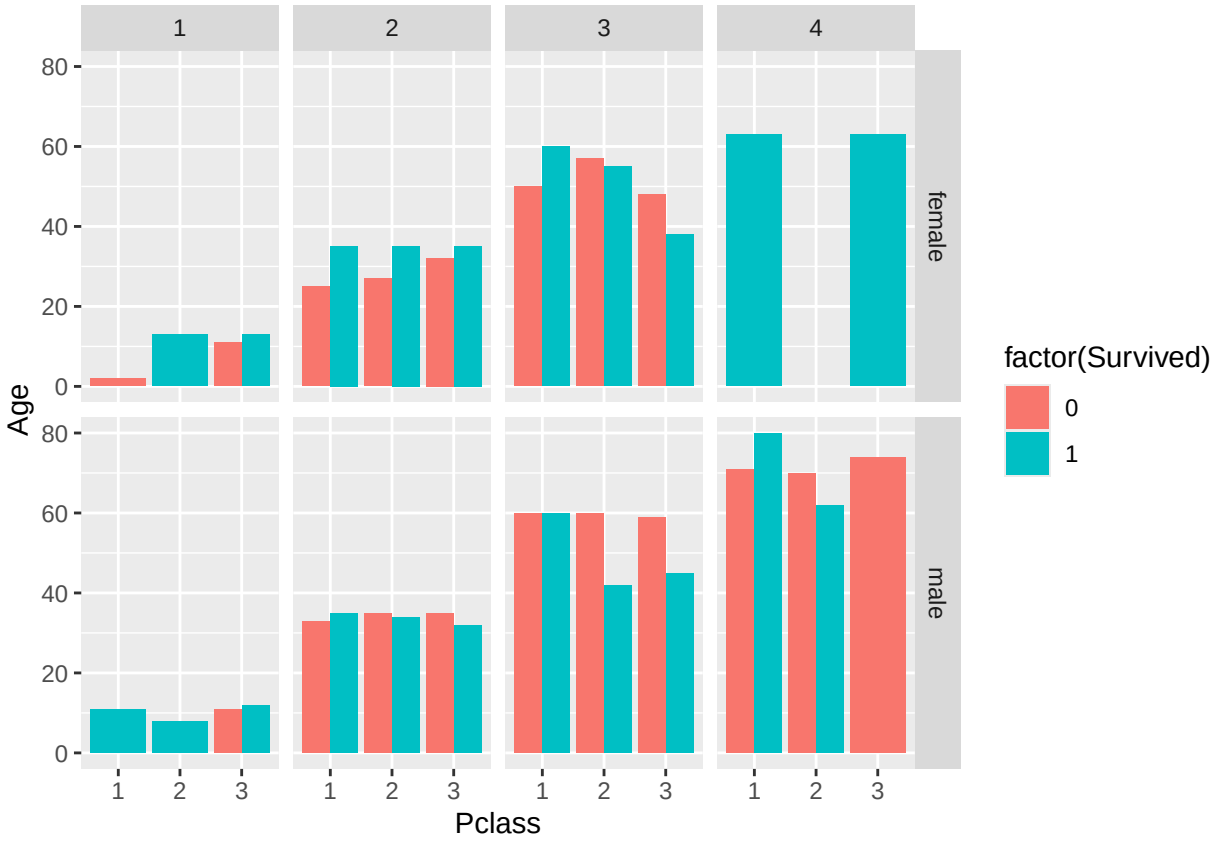

```
titanic_train_data %>%
  select(Survived, Pclass, age_groups, Sex,Age) %>%
  ggplot(data=) +
  geom_bar(mapping =  aes(x=Pclass, fill= factor(Survived)), color = "black", position = position_dodge
  facet_grid(Sex ~ age_groups)
```




```
titanic_train_data %>%
  select(Survived, Pclass, age_groups, Sex,Age) %>%
  ggplot(data=) +
  geom_bar(mapping =  aes(x=factor(Pclass), y = Age, fill= Survived), stat = "identity",  position=posi
```




```
titanic_train_data %>%
  select(Survived, Pclass, age_groups, Sex,Age) %>%
  ggplot(data=) +
  geom_bar(mapping =  aes(x=Pclass, y = Age, fill= factor(Survived)), stat = "identity",  position=posi
  facet_grid(Sex ~ age_groups)
```




```
#setting factors to categorical data
titanic_test_data$Survived <- NA
titanic_test_data$Survived<- as.factor(titanic_test_data$Survived)
```

Modelling the data.

Method one for doing our prediction

```
rf.model<-randomForest(factor(Survived) ~  age_groups + SibSp + Parch + Sex + Pclass + Embarked, data= 
rf.model %>%
  predict() %>%
  table()
```

```
## .
##   0   1
## 626 265
```

Method two for doing our prediction.

```
titanic_train_data$Age<- titanic_train_data%>%
  select(Age) %>%
  apply(c(2), . %>% {ifelse(is.na(.), 29.70, .)})

titanic_train_data.head <- titanic_train_data

titanic_test_data.head <- titanic_train_data %>%
```

```
  mutate(Age = NA)

str(titanic_test_data.head)
```

```
## 'data.frame':    891 obs. of  15 variables:
##  $ PassengerId        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived           : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass             : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name               : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs T
##  $ Sex                : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age                : logi  NA NA NA NA NA NA ...
##  $ SibSp              : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch              : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket             : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare               : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin              : chr  "" "C85" "" "C123" ...
##  $ Embarked           : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ age_groups         : num [1:891, 1] 2 3 2 2 2 2 3 1 2 2 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "Age"
##  $ age_groups_young   : chr [1:891, 1] "youth" "adult" "youth" "youth" ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "Age"
##  $ age_groups_children: chr [1:891, 1] "non-child" "non-child" "non-child" "non-child" ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "Age"
```

```
str(titanic_train_data.head)
```

```
## 'data.frame':    891 obs. of  15 variables:
##  $ PassengerId        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived           : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass             : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name               : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs T
##  $ Sex                : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age                : num [1:891, 1] 22 38 26 35 35 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "Age"
##  $ SibSp              : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch              : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket             : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare               : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin              : chr  "" "C85" "" "C123" ...
##  $ Embarked           : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ age_groups         : num [1:891, 1] 2 3 2 2 2 2 3 1 2 2 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "Age"
##  $ age_groups_young   : chr [1:891, 1] "youth" "adult" "youth" "youth" ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : NULL
##   .. ..$ : chr "Age"
##  $ age_groups_children: chr [1:891, 1] "non-child" "non-child" "non-child" "non-child" ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "Age"
```

```
modeleAge<-titanic_train_data.head %>%
  select(Age, Fare, Parch, Survived, SibSp, Pclass, Ticket, Sex) %>%
  lm(Age ~ Fare + Parch + Survived + SibSp + Pclass + Ticket +Sex, data= .)

predicted.Age<- predict(modeleAge , newdata =  titanic_test_data.head)
```

```
## Warning in predict.lm(modeleAge, newdata = titanic_test_data.head): prediction
## from a rank-deficient fit may be misleading
```

```
titanic_train_data$Age<- predicted.Age

Model.Survive<-randomForest(Survived ~ Age +  SibSp + Parch + Sex + Pclass + Embarked + Fare, data= tita

Model.Survive %>%
  predict() %>%
  table()
```

```
## .
##   0   1
## 609 282
```

```
rf.Predicted.Survive<-rf.model %>%
  predict()
rf.Predicted.Survive.table<-as.data.frame(rf.Predicted.Survive)
```