Titanic Analysis

S Wajahat Ali

5/7/2020

This analysis attempts to predict the survival of the Titanic passengers. In order to do this, I will use the different features available about the passengers, use a subset of the data to train an algorithm and then run the algorithm on the rest of the data set to get a prediction.

First all the missing values are found and all irrelavent variables are removed from the dataset. After cleaning the data visual analysis is done to find out the relationship between different features and Survival.Then, different forecating techniques are used to predict the survival of a passenger.

Data loading and cleaning

Reading Data

```
train = read.csv("train.csv", stringsAsFactors = FALSE)
test = read.csv("test.csv", stringsAsFactors = FALSE)
```

matching columns number on both sets of data

```
test$Survived = NA
```

creating a new dataset 'full' by combining both test and train

```
full = rbind(test, train)
```

summary of the data

```
summary(full)
```

```
##   PassengerId       Pclass          Name               Sex
##  Min.   :   1   Min.   :1.000   Length:1309        Length:1309
##  1st Qu.: 328   1st Qu.:2.000   Class :character   Class :character
##  Median : 655   Median :3.000   Mode  :character   Mode  :character
##  Mean   : 655   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:3.000
##  Max.   :1309   Max.   :3.000
##
##       Age            SibSp            Parch          Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.000   Length:1309
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000   Class :character
```

```
##  Median :28.00   Median :0.0000   Median :0.000   Mode  :character
##  Mean   :29.88   Mean   :0.4989   Mean   :0.385
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##  Max.   :80.00   Max.   :8.0000   Max.   :9.000
##  NA's   :263
##       Fare            Cabin             Embarked            Survived
##  Min.   :  0.000   Length:1309        Length:1309        Min.   :0.0000
##  1st Qu.:  7.896   Class :character   Class :character   1st Qu.:0.0000
##  Median : 14.454   Mode  :character   Mode  :character   Median :0.0000
##  Mean   : 33.295                                         Mean   :0.3838
##  3rd Qu.: 31.275                                         3rd Qu.:1.0000
##  Max.   :512.329                                         Max.   :1.0000
##  NA's   :1                                               NA's   :418
```

looking at possibe features which can be converted to factors.

```r
apply(full,2, function(x) length(unique(x)))
```

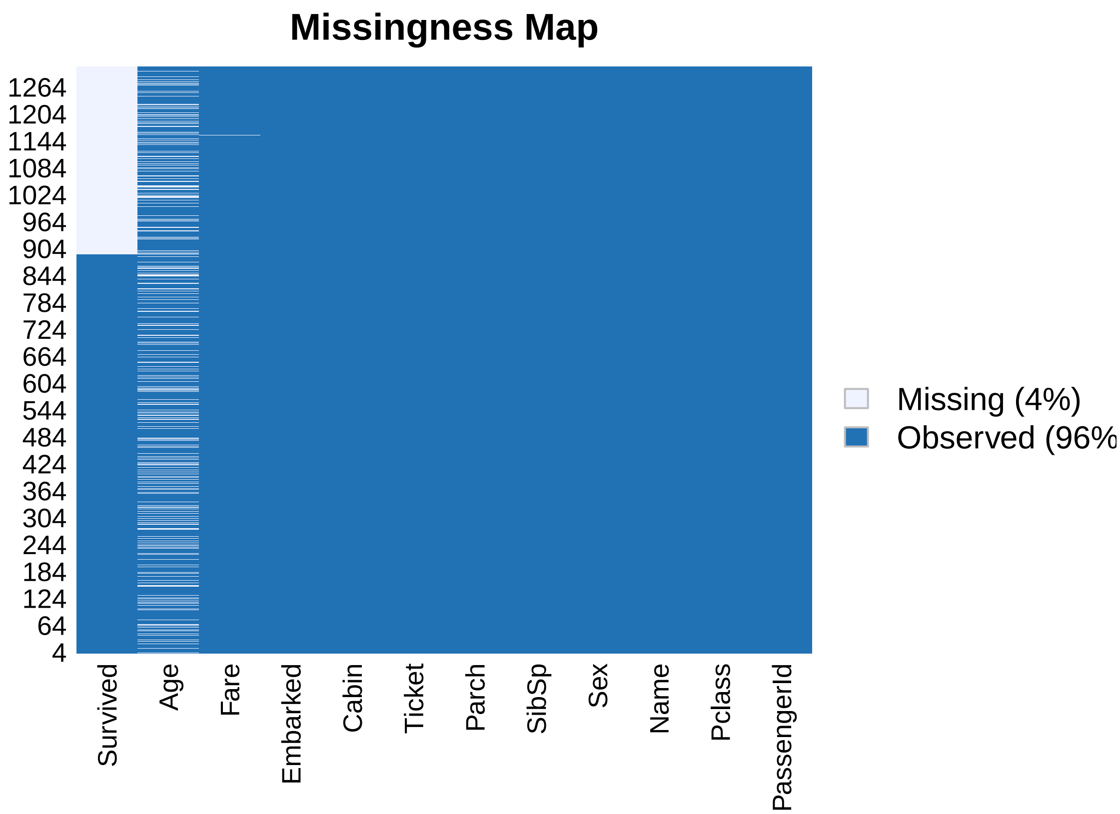
```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##        1309           3        1307           2          99           7
##       Parch      Ticket        Fare       Cabin    Embarked    Survived
##           8         929         282         187           4           3
```

Converting the features Survived, Pclass, Sex and Embarked to factors

```r
cols<-c("Survived","Pclass","Sex","Embarked")
for (i in cols){
  full[,i] <- as.factor(full[,i])
}
```

looking for any Missing values

```r
missmap(full)
```



Age and Fare have NAs

```
colSums(is.na(full))
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0         263           0
##       Parch      Ticket        Fare       Cabin    Embarked    Survived
##           0           0           1           0           0         418
```

Cabin and Embarked have empty strings

```
colSums(full=="")
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0          NA           0
##       Parch      Ticket        Fare       Cabin    Embarked    Survived
##           0           0          NA        1014           2          NA
```

##Cleaning Data

Ticket seems to have random aplpha numeric code so it will be removed Cabin has a lot of missing values so we will remove it too Name and PassengerId will also be removed, as they dont have any significant effect on Survived.

Removing Unwanted Variable

```
full = subset( full, select = -c(Cabin,Ticket,Name, PassengerId))
```

Filling out NAs and other missing values

```
 assigning the mode of emabarked to missing embarked
```

```
full[full$Embarked == '',"Embarked"] = "S"
```

```
assigning mean of fare to the missing values
```

```
full[is.na(full$Fare),"Fare"] = mean(full$Fare, na.rm = TRUE)
```

```
finding out missing age through SVM
```

```
# splitting the data into two data sets
have_age = subset(full,is.na(Age) == FALSE)
predict_age = subset(full, is.na(Age) == TRUE )

smp_size <- floor(0.80 * nrow(have_age))
train_ind <- sample(seq_len(nrow(have_age)), size = smp_size)
train_age <- have_age[train_ind, ]
test_age <- have_age[-train_ind,]

# since Age has NAs we will not pass it in our train data set
svm_model_age = svm(Age~Pclass+Sex+SibSp+Parch+Fare+Embarked, data = subset(train_age, select = -Surviv
                             type = "eps-regression", kernel = "radial")


test_age$age_predicted = predict(svm_model_age, subset(test_age, select = -Survived ))
accuracy(test_age$Age, test_age$age_predicted)
```

```
##                    ME     RMSE      MAE       MPE     MAPE
## Test set -0.9844547 12.66188 9.813791 -4.291838 35.90337
```

```
predicting age
```

```
predict_age$Age = predict(svm_model_age, subset(predict_age, select = -c(Age,Survived) ))
```

```
combining the two data, full1 doesnt have any missing value.
```

```
full1 = rbind(have_age, predict_age)
```

```
looking for any Missing values
```

```
colSums(is.na(full1))
```

```
##   Pclass      Sex      Age    SibSp    Parch     Fare Embarked Survived
##        0        0        0        0        0        0        0      418
```

```
# only Age has NAs as expected
```

```
colSums(full1=="")
```

```
##    Pclass       Sex       Age     SibSp     Parch      Fare Embarked Survived
##         0         0         0         0         0         0        0       NA
```

```
# no empty strings found
```

we have a clean data set now

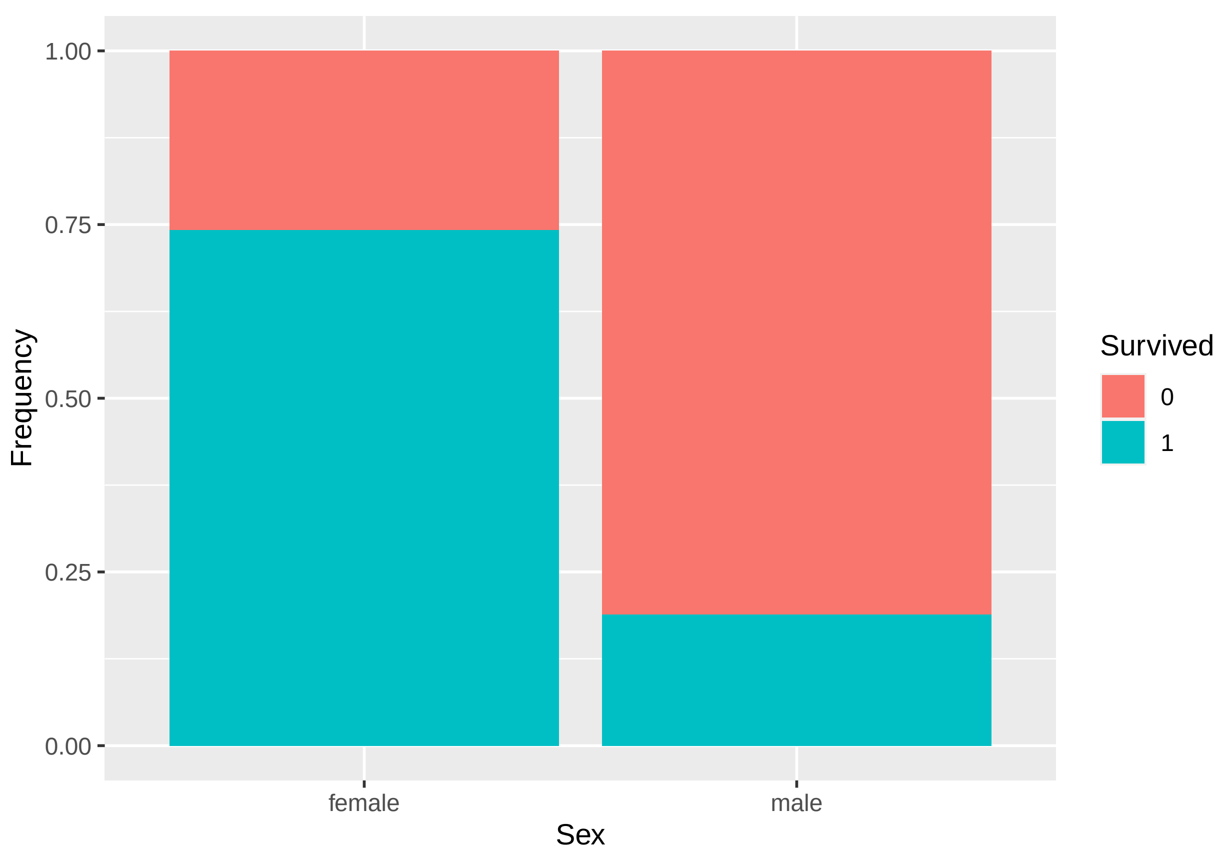
dividing the data into two sets

```
have_survived = subset(full1,is.na(Survived) == FALSE)
predict_survived = subset(full1, is.na(Survived) == TRUE )
```

Visual Analysis

Analyzing the role of gender in Survival

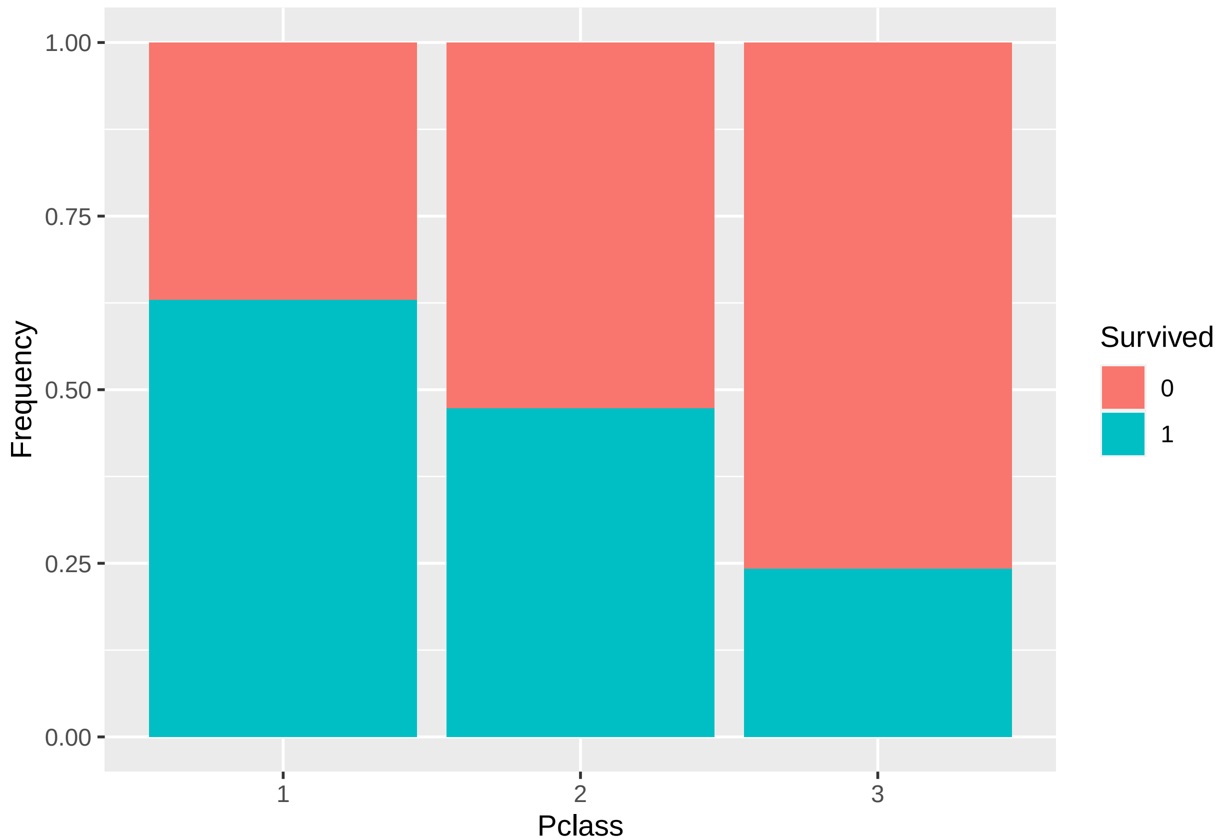
```
ggplot(have_survived,aes(x=Sex,fill=Survived))+
  geom_bar(position = "fill")+
  ylab("Frequency")
```



```
# a female has more chances of surviving compare to a male
```

Analyzing the role of Pclass in Survival

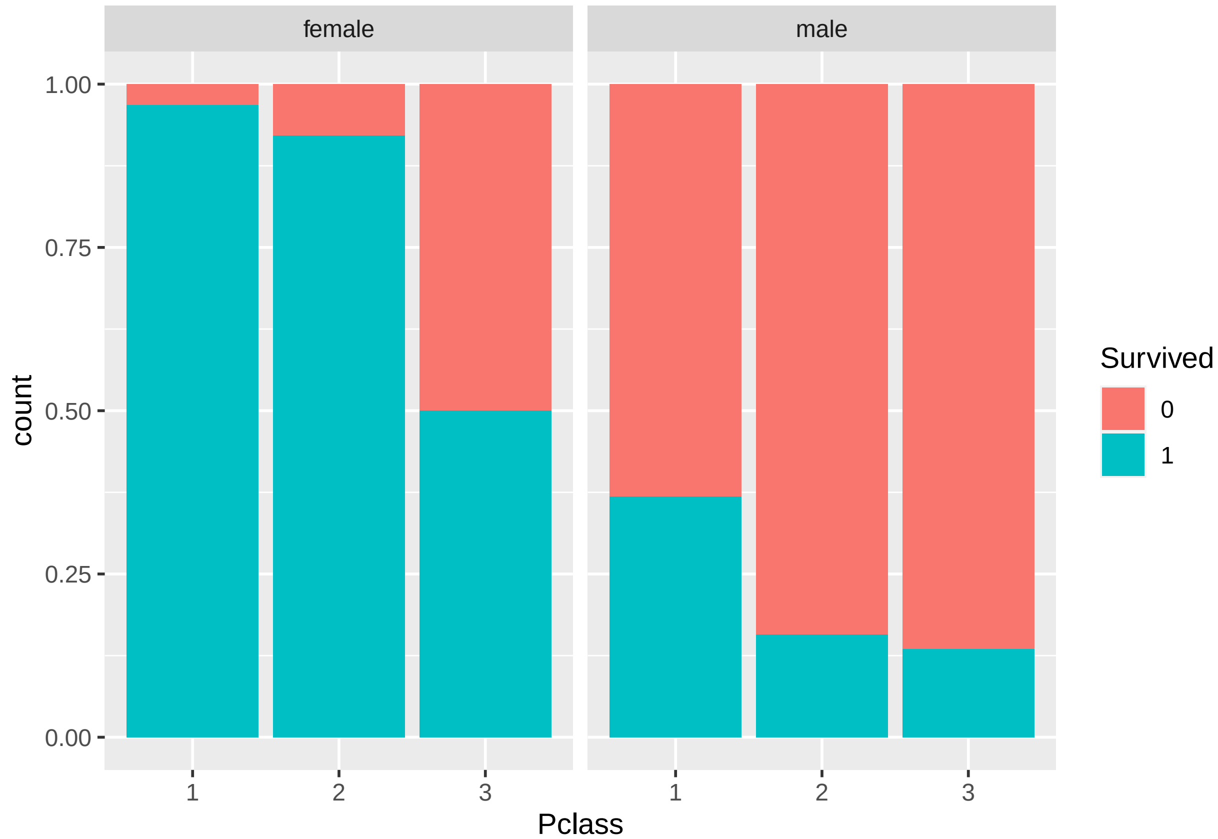
```
ggplot(have_survived,aes(x=Pclass,fill=Survived))+
  geom_bar(position = "fill")+
  ylab("Frequency")
```




```
# chances of survival are higher in class 1 and least in class 3
```

looking at gender classwise

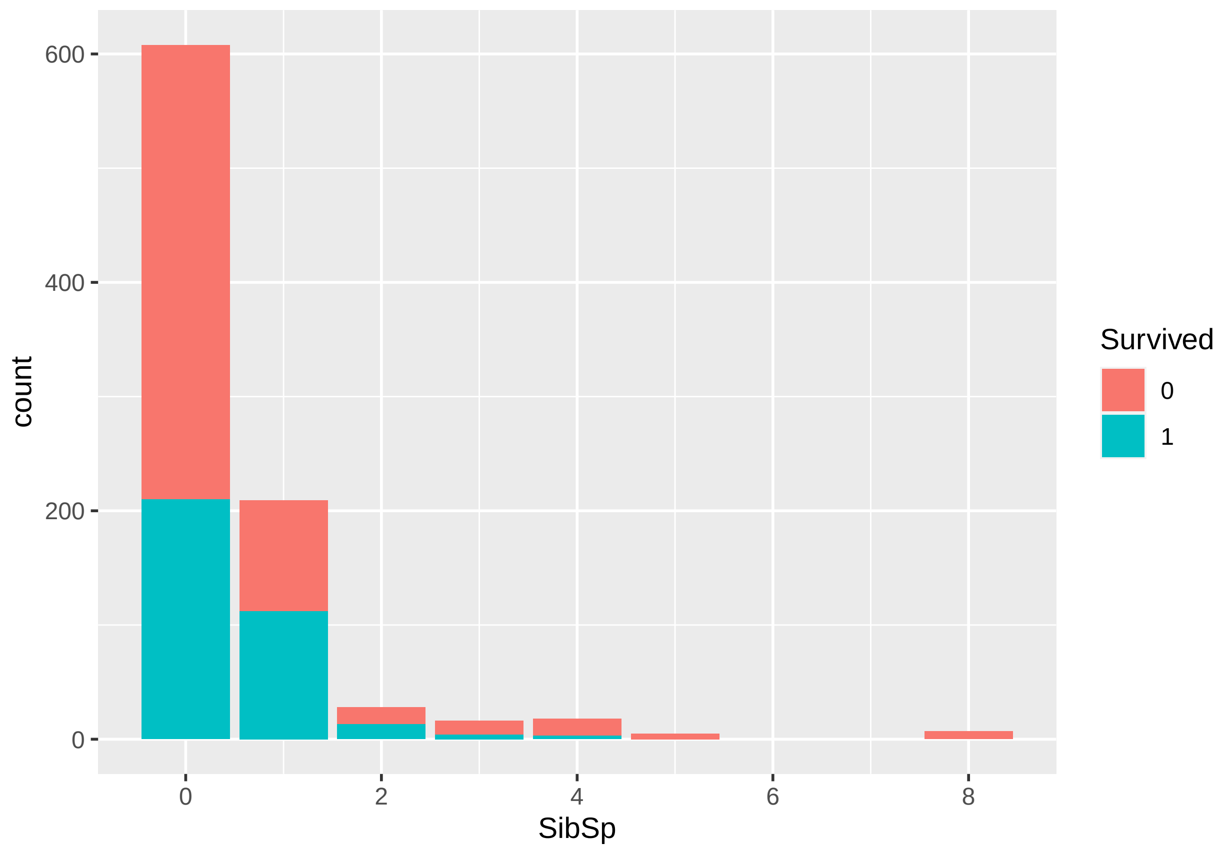
```
ggplot(data = have_survived,aes(x=Pclass,fill=Survived))+
  geom_bar(position="fill")+
  facet_wrap(~Sex)
```




```
# a female has higher chances of survival compared to a man regardless of class
```

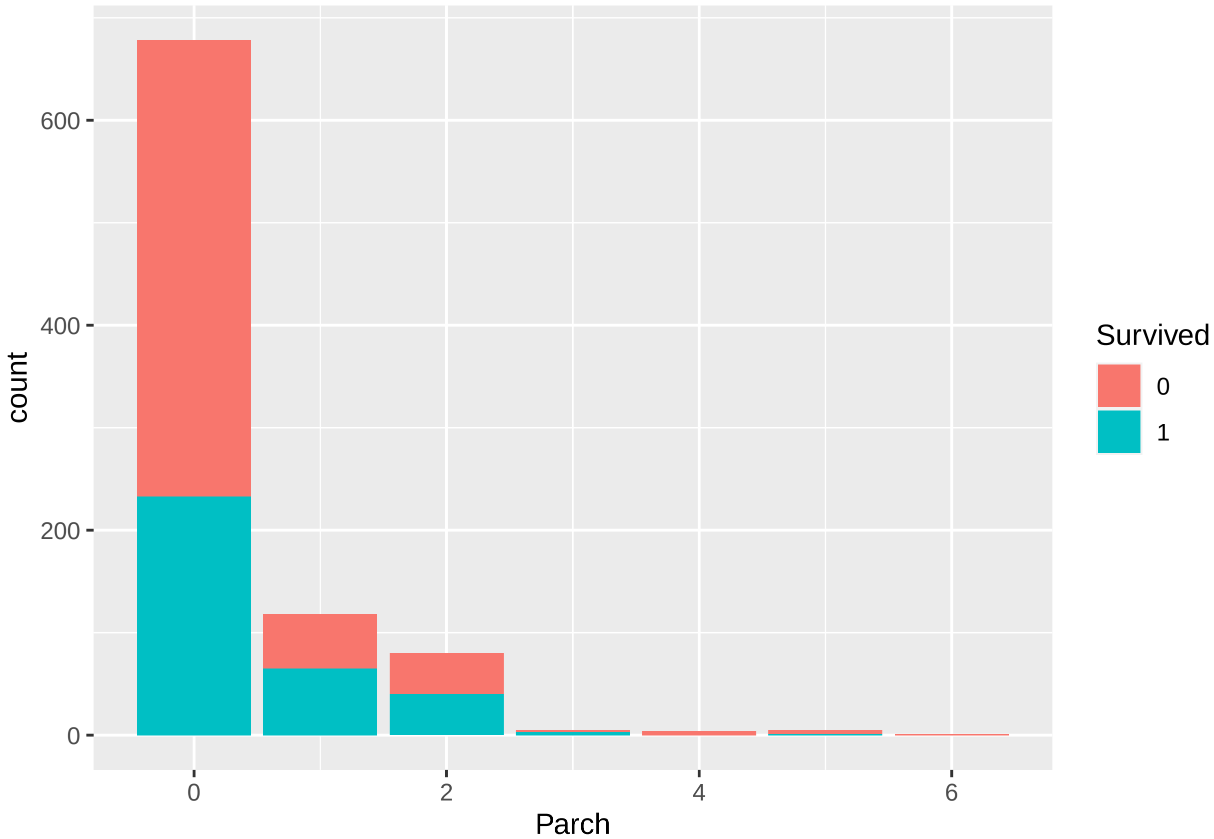
Analyzing the role of Sibsp in Survival

```
ggplot(have_survived,aes(x=SibSp,fill=Survived))+geom_bar()
```



Analyzing the role of Parch in Survival

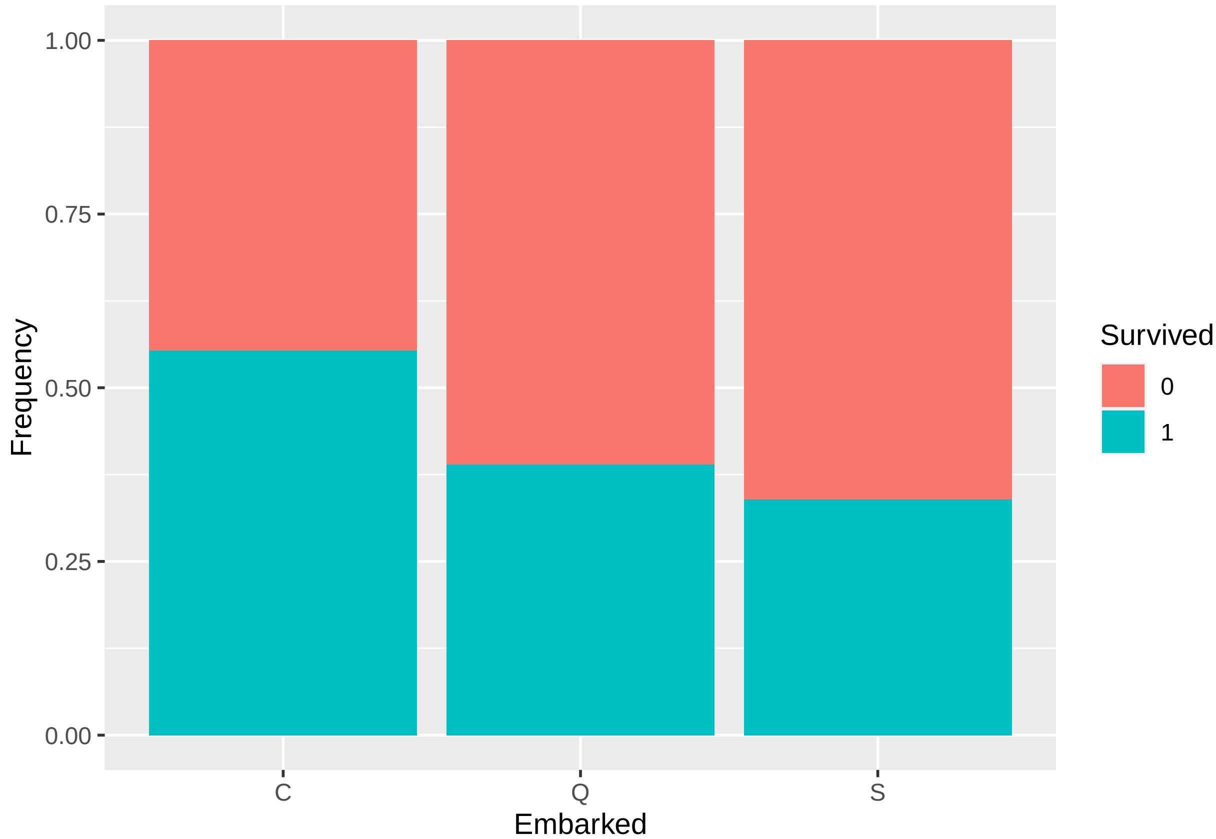
```
ggplot(have_survived,aes(x=Parch,fill=Survived))+geom_bar()
```



parch and SibSp seems to have similar impact on survivor but we are not sure if SibSp 0 corresponds to same passenger in Parch 0

```
Analyzing the role of Embarked in Survival
```

```r
ggplot(have_survived,aes(x=Embarked,fill=Survived))+
  geom_bar(position = "fill")+
  ylab("Frequency")
```




```r
# S and Q have little below 50% survived
# C has a little above 50% survived
```

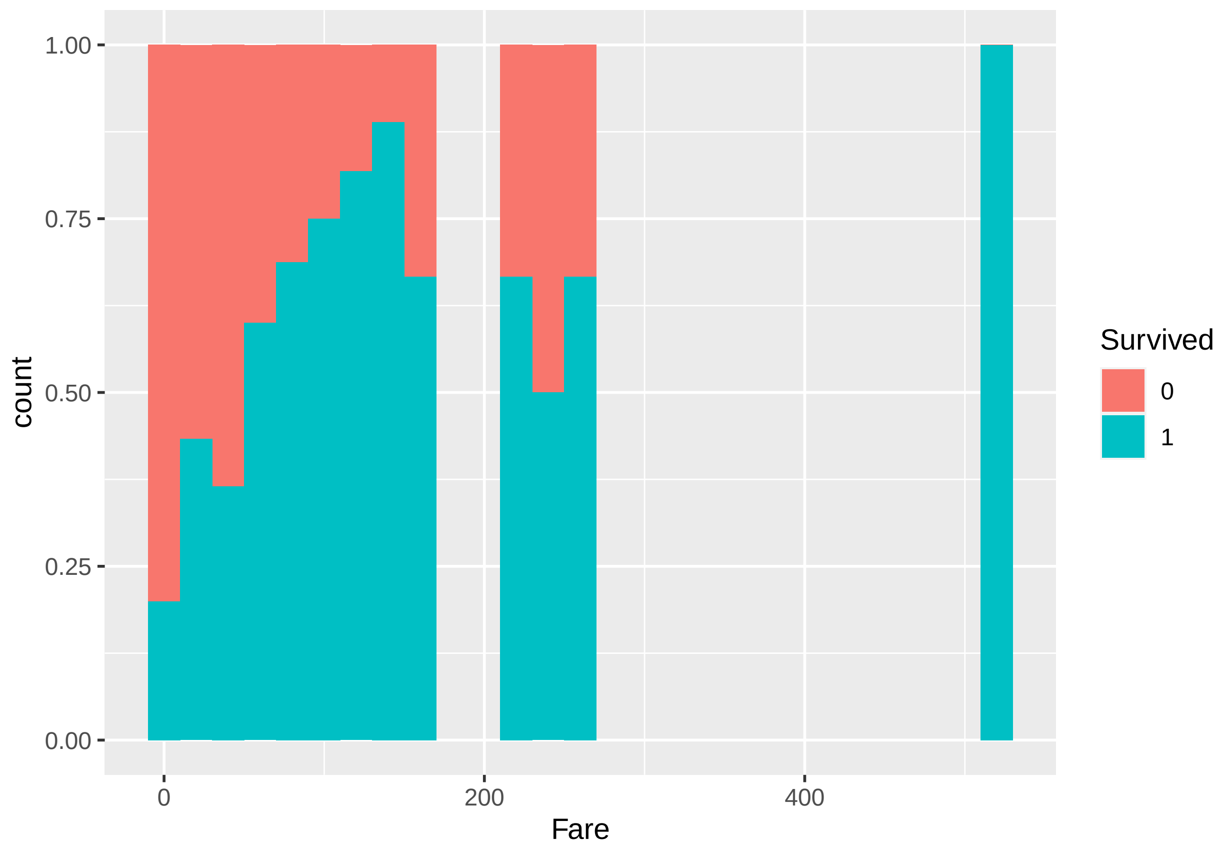
Analyzing the role of Age in Survival

```r
ggplot(data = have_survived,aes(x=Age,fill=Survived))+
  geom_histogram(binwidth = 3,position="fill")+
  ylab("Frequency")
```


```
# Children aged below 15 and old people aged above 80 have more chances of survival
```

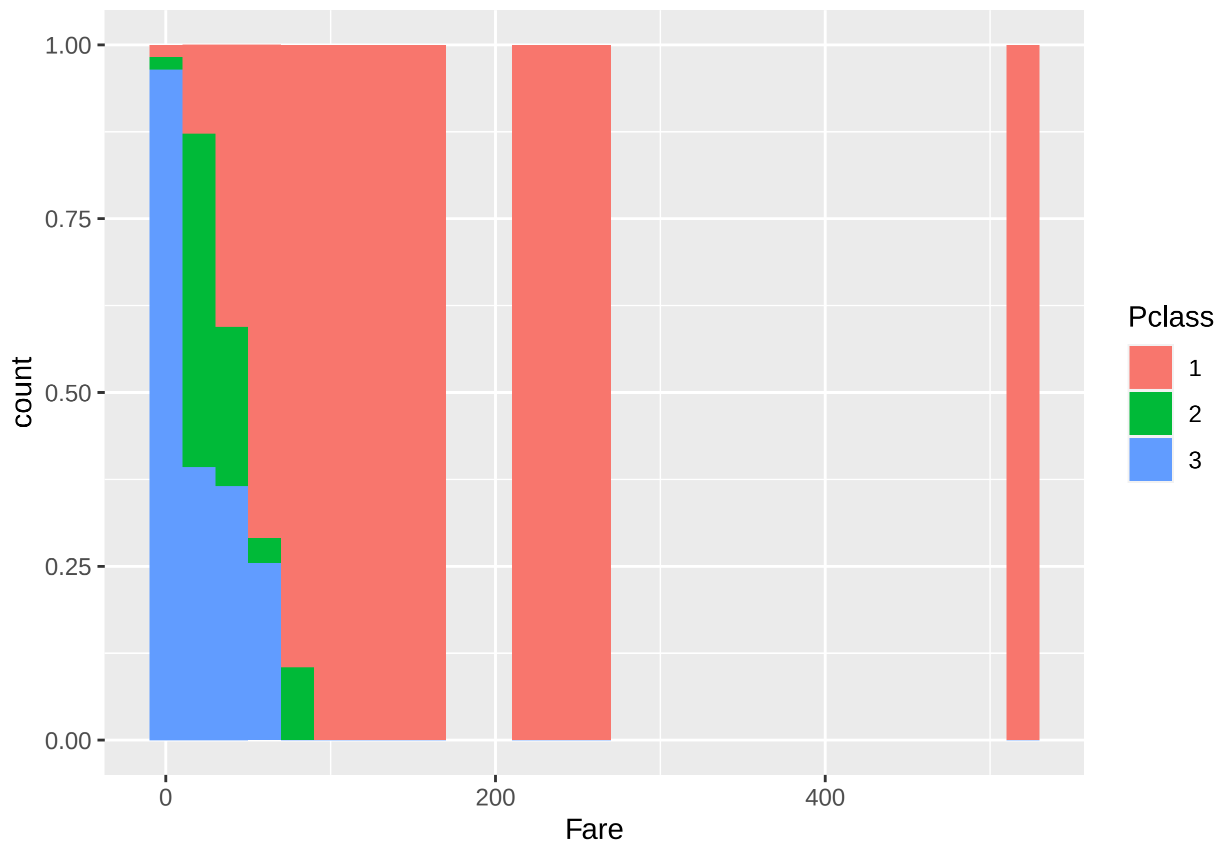
Analyzing the role of Fare in Survival

```
ggplot(data = have_survived,aes(x=Fare,fill=Survived))+
  geom_histogram(binwidth =20, position="fill")
```




```
# chances of survival increase with increasing fare
```

```
ggplot(data = have_survived,aes(x=Fare,fill=Pclass))+
  geom_histogram(binwidth =20, position="fill")
```




```
# class one has the highest fare and, class 3 has least
```

Predicting

```
dividing have_survived into test and train
```

```
smp_size <- floor(0.80 * nrow(have_survived))
train_ind <- sample(seq_len(nrow(have_survived)), size = smp_size)
train_survived <- have_survived[train_ind, ]
test_survived <- have_survived[-train_ind,]
```

Predicting with glm

```
glm_model_survived = glm(Survived~.,family = "binomial",
                         data = train_survived)
test_survived$predicted_survived = predict(glm_model_survived,test_survived)
test_survived$predicted_survived = ifelse(test_survived$predicted_survived > 0.5,1,0)
test_survived$predicted_survived = as.factor(test_survived$predicted_survived)
confusionMatrix(test_survived$predicted_survived,test_survived$Survived)
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction  0  1
##          0 95 30
##          1  7 47
##
##                Accuracy : 0.7933
##                  95% CI : (0.7265, 0.8501)
##     No Information Rate : 0.5698
##     P-Value [Acc > NIR] : 2.615e-10
##
##                   Kappa : 0.5623
##
##  Mcnemar's Test P-Value : 0.0002983
##
##             Sensitivity : 0.9314
##             Specificity : 0.6104
##          Pos Pred Value : 0.7600
##          Neg Pred Value : 0.8704
##              Prevalence : 0.5698
##          Detection Rate : 0.5307
##    Detection Prevalence : 0.6983
##       Balanced Accuracy : 0.7709
##
##        'Positive' Class : 0
##
```

```
# Accuracy = 84.92%
```

Predictiing with SVM

```
svm_model_survived = svm(Survived~., data = train_survived,
                    type = "C-classification", kernel = "radial")

test_survived$predicted_survived = predict(svm_model_survived,test_survived)
confusionMatrix(test_survived$predicted_survived,test_survived$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 90 21
##          1 12 56
##
##                Accuracy : 0.8156
##                  95% CI : (0.751, 0.8696)
##     No Information Rate : 0.5698
##     P-Value [Acc > NIR] : 2.746e-12
##
##                   Kappa : 0.6185
##
##  Mcnemar's Test P-Value : 0.1637
```

```
##
##             Sensitivity : 0.8824
##             Specificity : 0.7273
##          Pos Pred Value : 0.8108
##          Neg Pred Value : 0.8235
##              Prevalence : 0.5698
##          Detection Rate : 0.5028
##    Detection Prevalence : 0.6201
##       Balanced Accuracy : 0.8048
##
##        'Positive' Class : 0
##
```

```
# Accuracy = 86.03%
```

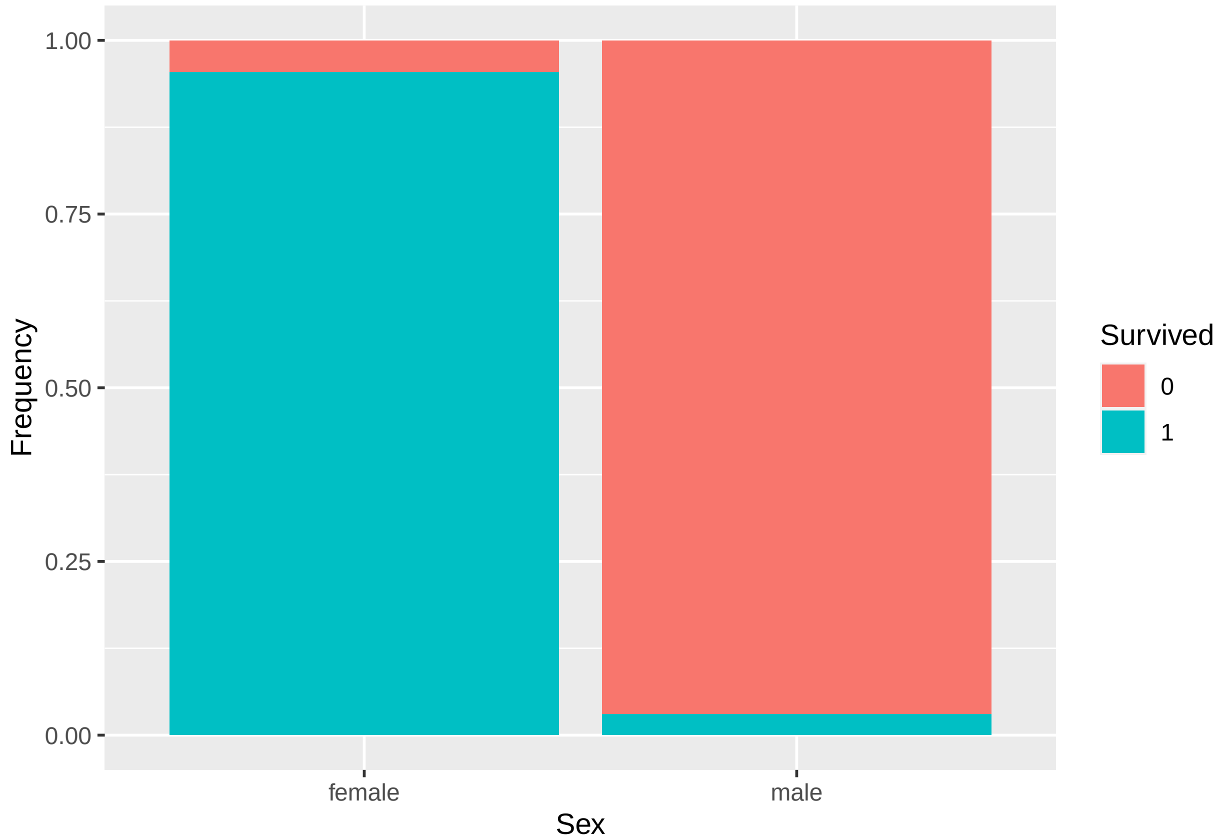
using the most accurate of the models above to predict

```
# we will not pass Survived as it has NAs
predict_survived$Survived = predict(svm_model_survived,subset(predict_survived,
                                                              select = -Survived ))
```

Visual Analysis of the Predicted Data

Analyzing the role of gender wrt predicted Survived

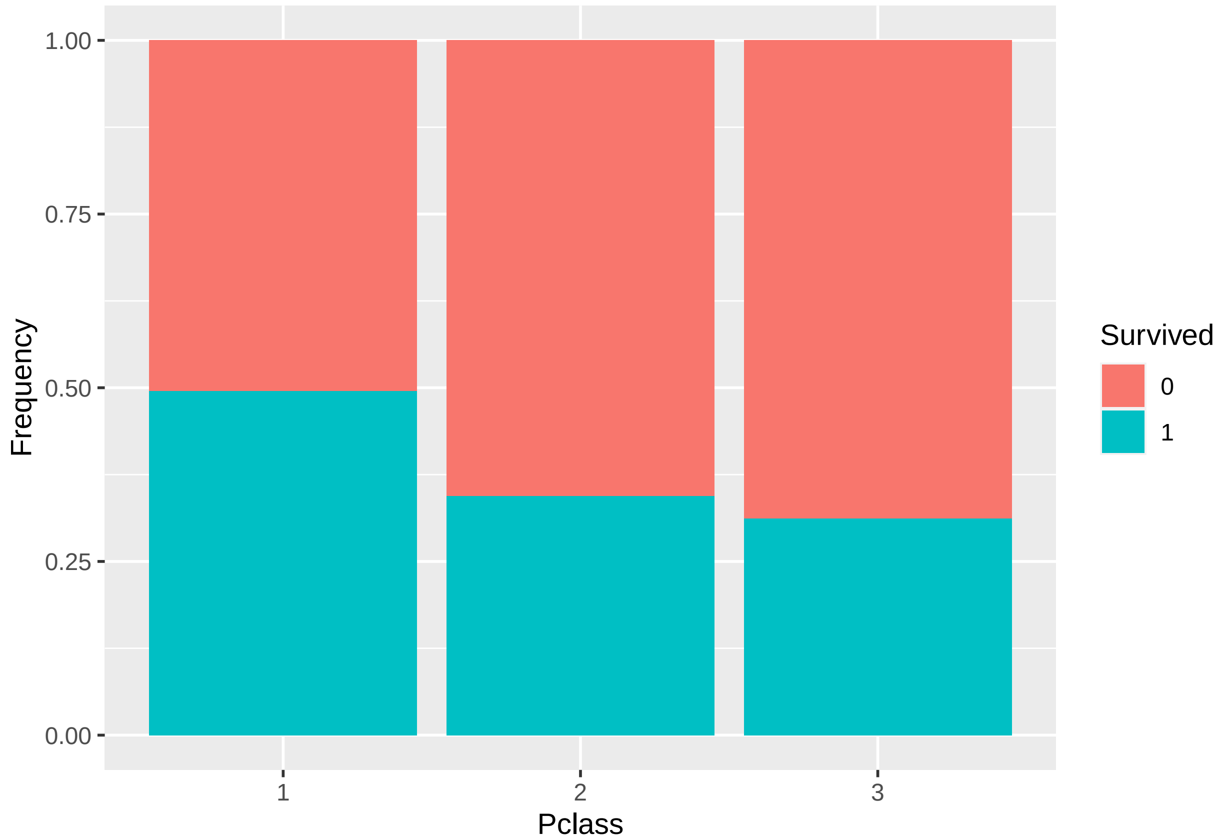
```
ggplot(predict_survived,aes(x=Sex,fill=Survived))+
  geom_bar(position="fill")+
  ylab("Frequency")
```




```
# as expected a female has more chances of survival compare to a male
```

Analyzing the role of Pclass wrt predicted Survived

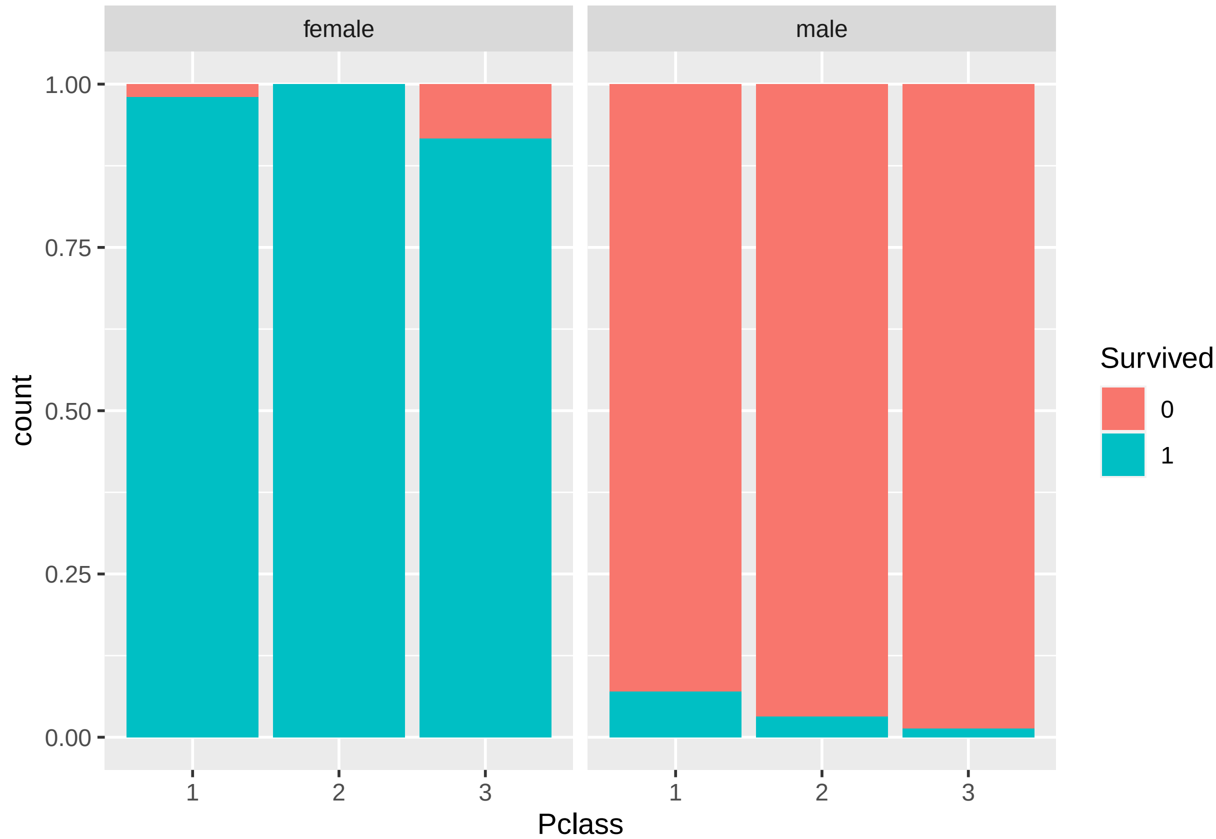
```
ggplot(predict_survived,aes(x=Pclass,fill=Survived))+
  geom_bar(position = "fill")+
  ylab("Frequency")
```




```
# as expected chances of survival are higher in class 1 and least in class 3
```

looking at gender classwise

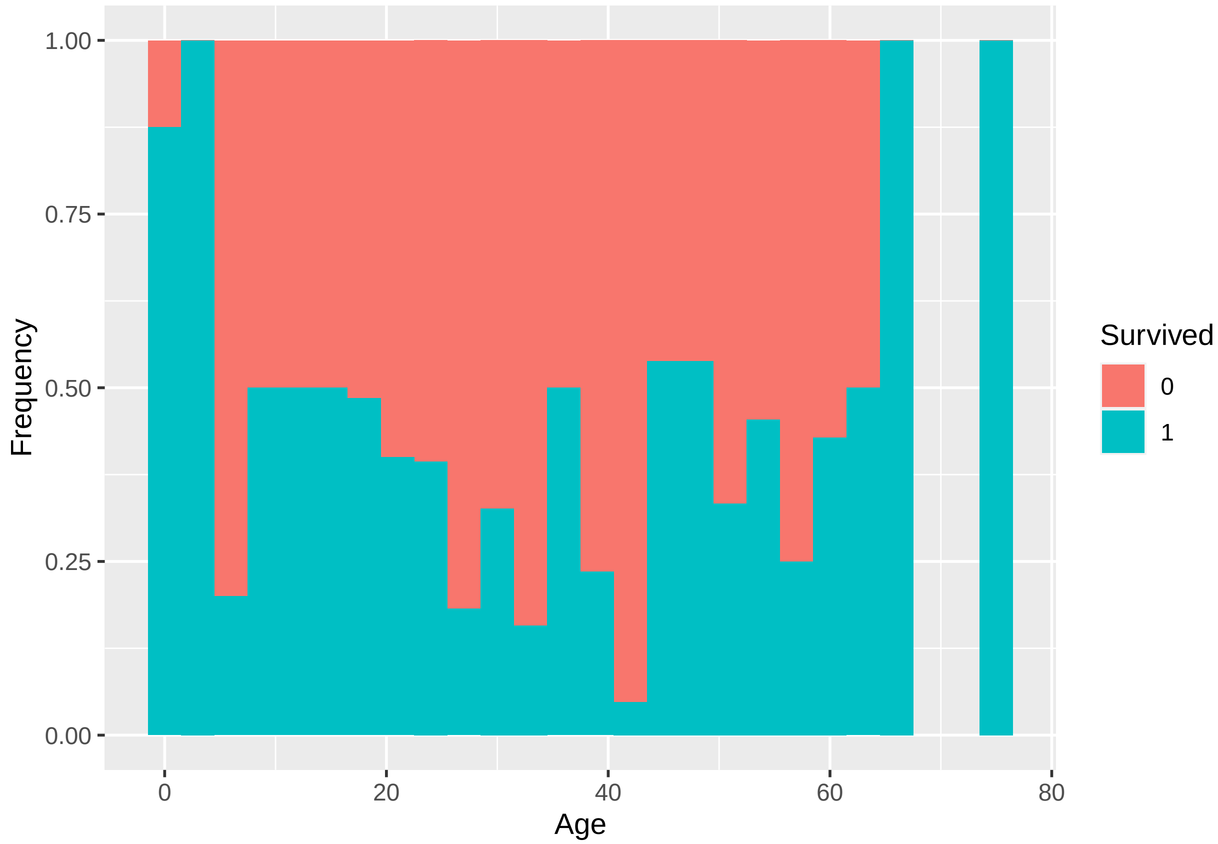
```
ggplot(data = predict_survived,aes(x=Pclass,fill=Survived))+
  geom_bar(position="fill")+
  facet_wrap(~Sex)
```




```
# as expected a female has higher chances of survival compared to a man regardless of class
```

Analyzing the role of Age wrt predicted Survived

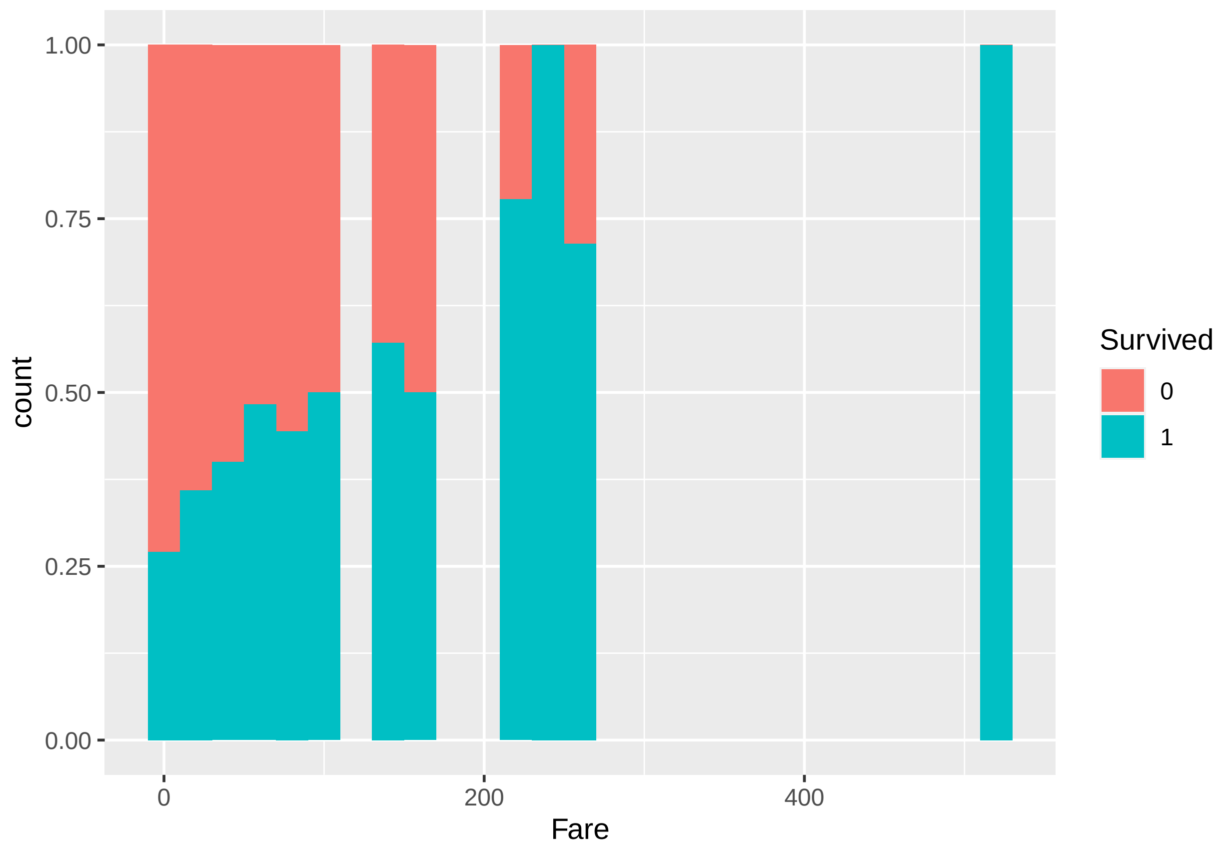
```
ggplot(data = predict_survived,aes(x=Age,fill=Survived))+
  geom_histogram(binwidth = 3,position="fill")+
  ylab("Frequency")
```




```
# As expected Children and old people have higher chances of survival
```

Analyzing the role of Fare wrt predicted Survived

```
ggplot(data = predict_survived,aes(x=Fare,fill=Survived))+
  geom_histogram(binwidth =20, position="fill")
```




```
# as expected chances of survival are higher for higher fare
```