title: “Titanic Tragedy Dataset EDA and Training” output: html_notebook — # LOADING DATASET

```
titanic <- read.csv("/home/tanmay/Datasets/train.csv")
```

EXPLORATORY DATA ANALYSIS

```
#DATASET PREVIEW
head(titanic)
```

```
##   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
```

DATASET STRUCTURE

```
str(titanic)
```

```
## '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       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 581 ...
##  $ 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     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

DATASET SUMMARY

```
summary(titanic)
```

```
##   PassengerId       Survived          Pclass
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000
##  Median :446.0   Median :0.0000   Median :3.000
##  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
##
##                                     Name         Sex           Age
##  Abbing, Mr. Anthony                   :  1   female:314   Min.   : 0.42
##  Abbott, Mr. Rossmore Edward           :  1   male  :577   1st Qu.:20.12
##  Abbott, Mrs. Stanton (Rosa Hunt)      :  1                Median :28.00
##  Abelson, Mr. Samuel                   :  1                Mean   :29.70
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00
##  Adahl, Mr. Mauritz Nils Martin        :  1                Max.   :80.00
##  (Other)                               :885                NA's   :177
##      SibSp           Parch             Ticket         Fare
##  Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :  0.00
##  1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:  7.91
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45
##  Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   : 32.20
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.: 31.00
##  Max.   :8.000   Max.   :6.0000   CA 2144 :  6   Max.   :512.33
##                                   (Other) :852
##          Cabin     Embarked
##             :687    :  2
##  B96 B98    :  4   C:168
##  C23 C25 C27:  4   Q: 77
##  G6         :  4   S:644
##  C22 C26    :  3
##  D          :  3
##  (Other)    :186
```

CHECKING MISSING VALUES

```
checkNA <- function(x){sum(is.na(x))/length(x)*100}
sapply(titanic,checkNA)
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##     0.00000     0.00000     0.00000     0.00000     0.00000    19.86532
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##     0.00000     0.00000     0.00000     0.00000     0.00000     0.00000
```

```
print("MISSING VALUES WITHOUT NA")
```

```
## [1] "MISSING VALUES WITHOUT NA"
```

```
checkMissing <- function(x){sum(x=="")/length(x)*100}
sapply(titanic,checkMissing)
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##   0.0000000   0.0000000   0.0000000   0.0000000  77.1043771   0.2244669
```

Missing Value Treatment

```
#1. Age: Replacing NA values in Age with mean
#titanic[is.na(titanic$Age),6] <- mean(titanic$Age)
titanic$Age[is.na(titanic$Age)] <- round(mean(titanic$Age, na.rm = TRUE))

#2. Embarked: Replacing Empty Embarked with most common value 'S'
titanic$Embarked <- replace(titanic$Embarked, which(titanic$Embarked==""), 'S')
```

```
Title <-  gsub("^.*, (.*?)\\..*$", "\\1", titanic$Name)
titanic$Title <- as.factor(Title)
table(Title)
```

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

```
  titanic$FamilyCount <-titanic$SibSp + titanic$Parch + 1
titanic$FamilySize[titanic$FamilyCount == 1] <- 'Single'
titanic$FamilySize[titanic$FamilyCount < 5 & titanic$FamilyCount >= 2] <- 'Small'
titanic$FamilySize[titanic$FamilyCount >= 5] <- 'Big'
titanic$FamilySize=as.factor(titanic$FamilySize)
table(titanic$FamilySize)
```

```
##
##    Big Single  Small
##     62    537    292
```

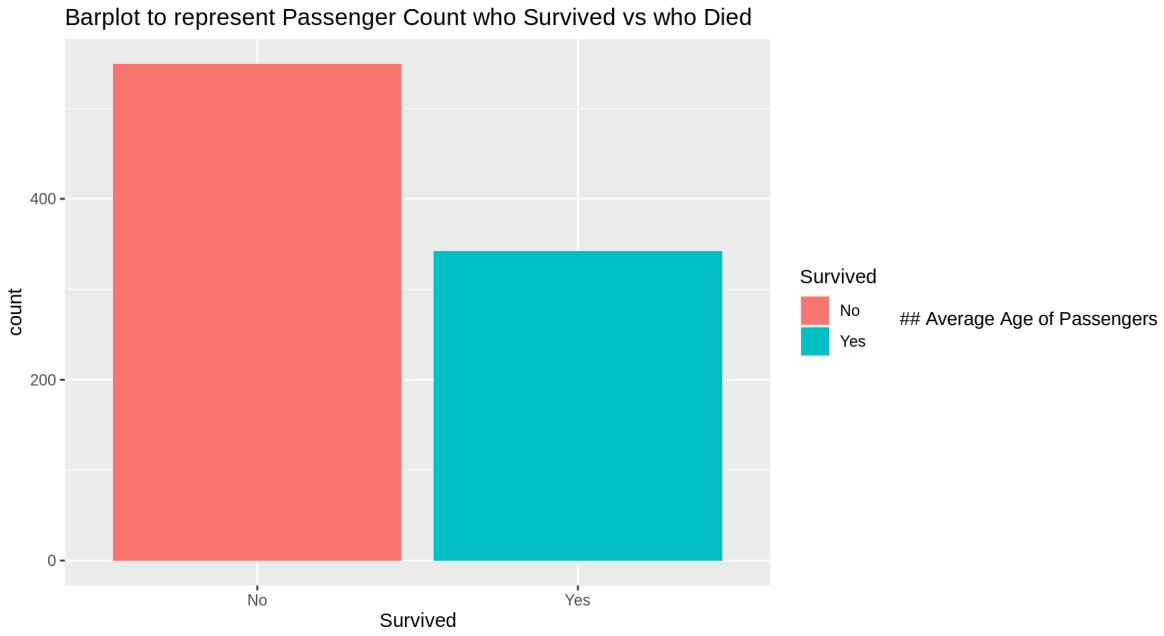
DATA PREPROCESSING

```
# 1.Changing names of few categorical variables for interpretability
titanic$Survived <- ifelse(titanic$Survived==1,"Yes","No")
titanic$Survived <- as.factor(titanic$Survived)

titanic$Embarked <- ifelse(titanic$Embarked=="S","Southampton",
                           ifelse(titanic$Embarked=="C","Cherbourg", "Queenstown"))
titanic$Embarked <- as.factor(titanic$Embarked)
#
# # 2.Converting categorical variables from int to factor
# i) Pclass
titanic$Pclass <- as.factor(titanic$Pclass)
#
# # ii) SibSp
titanic$SibSp <- as.factor(titanic$SibSp)
#
# # iii) Parch
titanic$Parch <- as.factor(titanic$Parch)

#Preview
head(titanic)
```

```
##   PassengerId Survived Pclass
## 1           1       No      3
## 2           2      Yes      1
## 3           3      Yes      3
## 4           4      Yes      1
## 5           5       No      3
## 6           6       No      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  30     0     0
##             Ticket    Fare Cabin    Embarked Title FamilyCount FamilySize
## 1        A/5 21171  7.2500       Southampton    Mr           2      Small
## 2         PC 17599 71.2833   C85   Cherbourg   Mrs           2      Small
## 3 STON/O2. 3101282  7.9250       Southampton  Miss           1     Single
## 4           113803 53.1000  C123 Southampton   Mrs           2      Small
## 5           373450  8.0500       Southampton    Mr           1     Single
## 6           330877  8.4583        Queenstown    Mr           1     Single
```

Survival Demographic

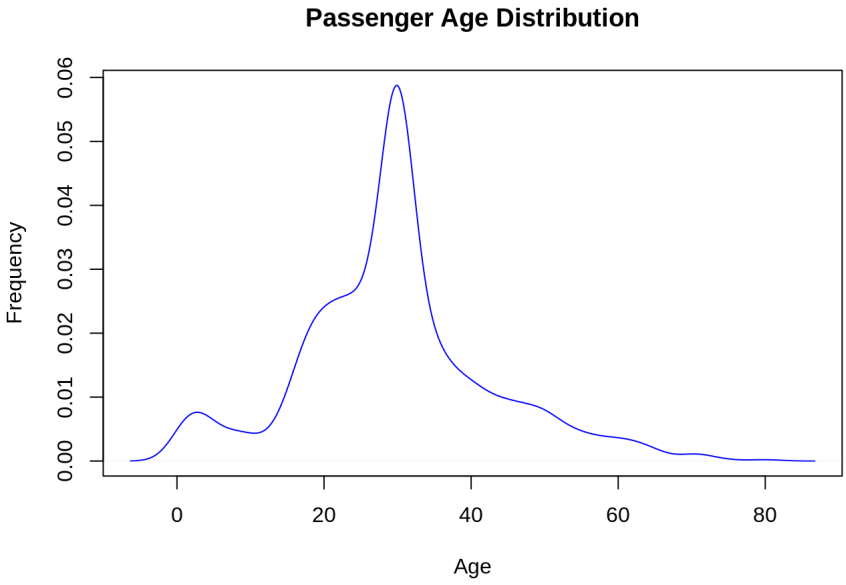
```
library(ggplot2)
ggplot(titanic, aes(Survived,fill = Survived))+
  geom_bar()+
  ggtitle("Barplot to represent Passenger Count who Survived vs who Died")
```




```
summary(titanic$Age)
```

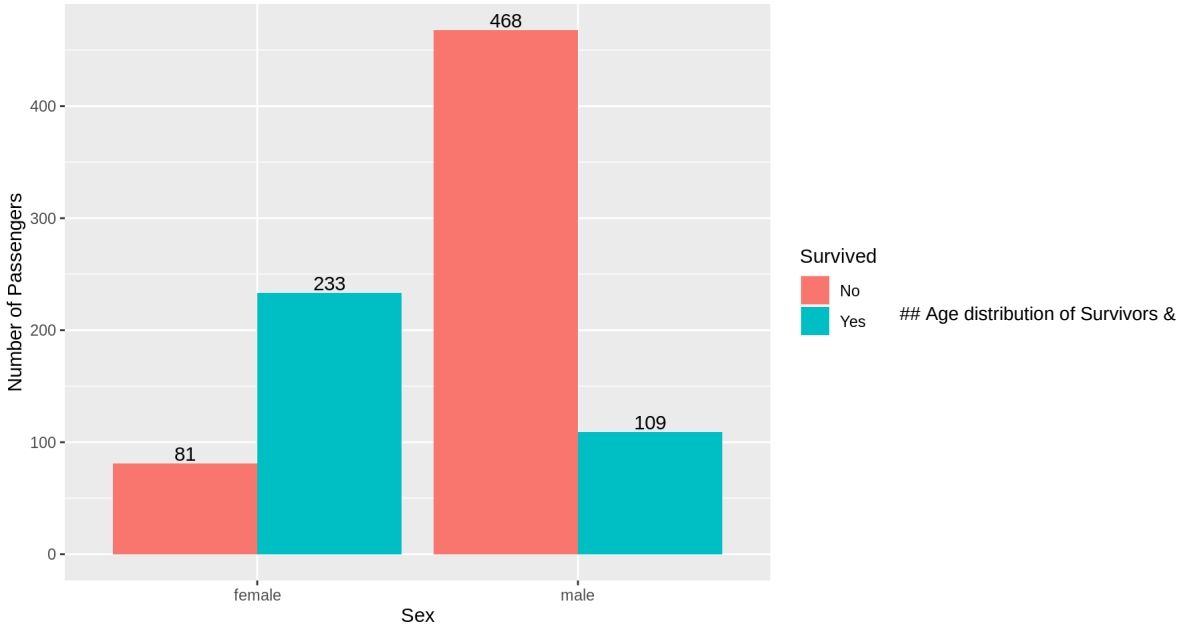
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   22.00   30.00   29.76   35.00   80.00
```

```
d <- density(titanic$Age)
plot(d,main="Passenger Age Distribution",xlab="Age",ylab="Frequency",col="blue")
```



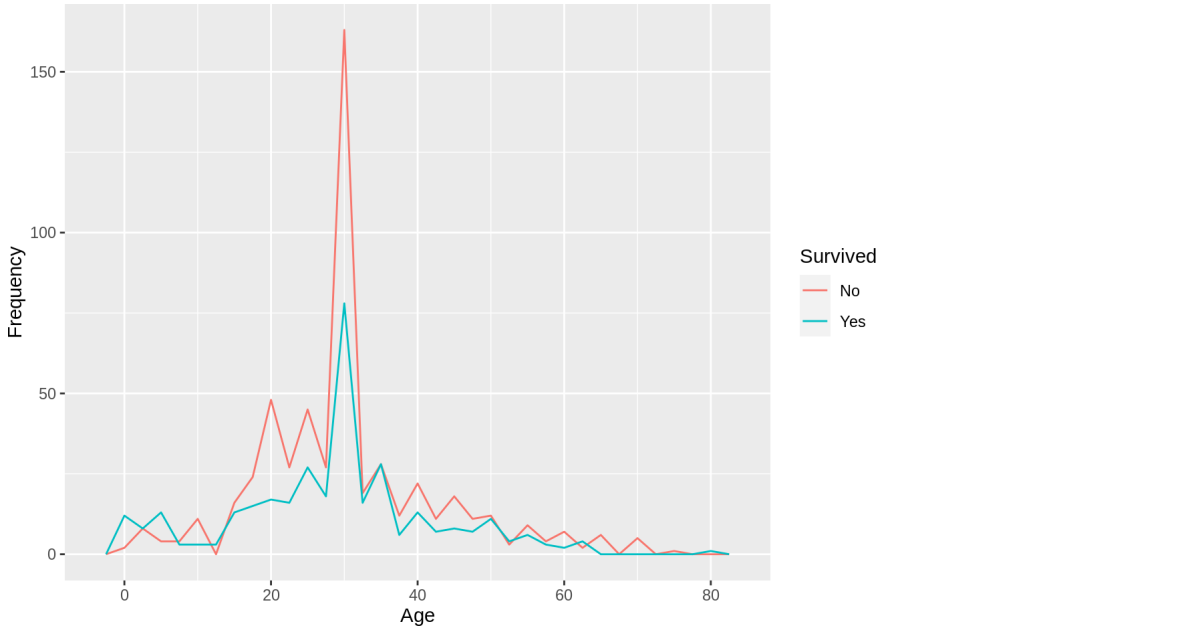
Proportion of survivors by gender

```
ggplot(titanic, aes(x=Sex,fill=Survived))+ geom_bar(position = "dodge") + geom_text(stat='count',aes(label=..coun
t..),position = position_dodge(0.9),vjust=-0.2) +
ylab("Number of Passengers")
```



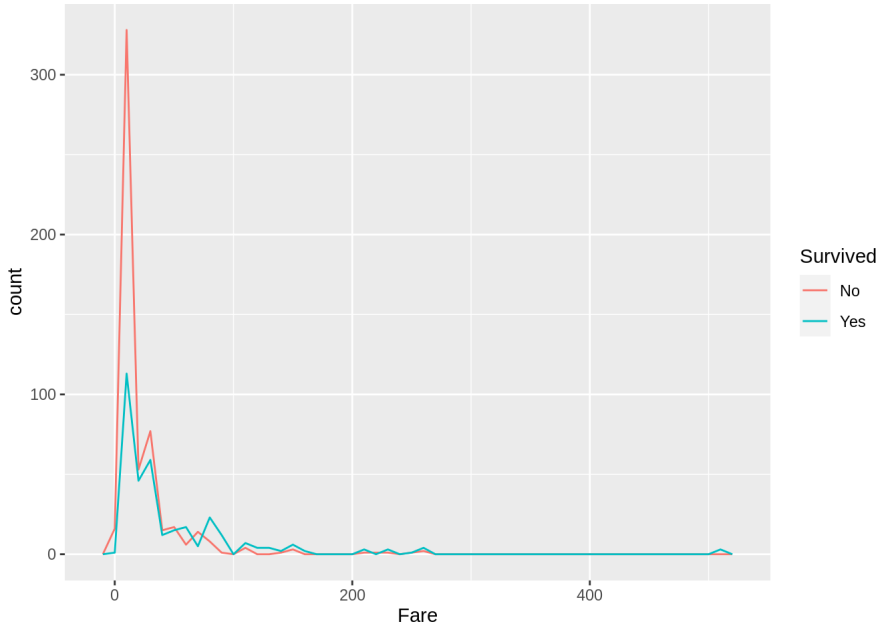
Age distribution of Survivors & Non-Survivors

```
ggplot(titanic) + geom_freqpoly(mapping = aes(x = Age, color = Survived), binwidth = 2.5) +
ylab("Frequency")
```



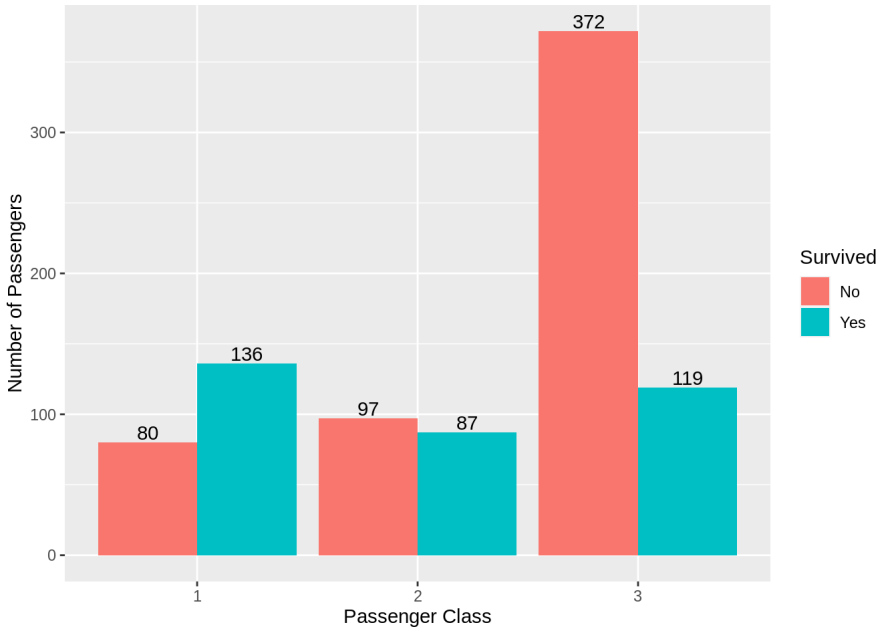
Distribution of Passenger Fare for Survivors & Non-Survivors

```
ggplot(titanic) + geom_freqpoly(mapping = aes(x = Fare, color = Survived), binwidth = 10)
```



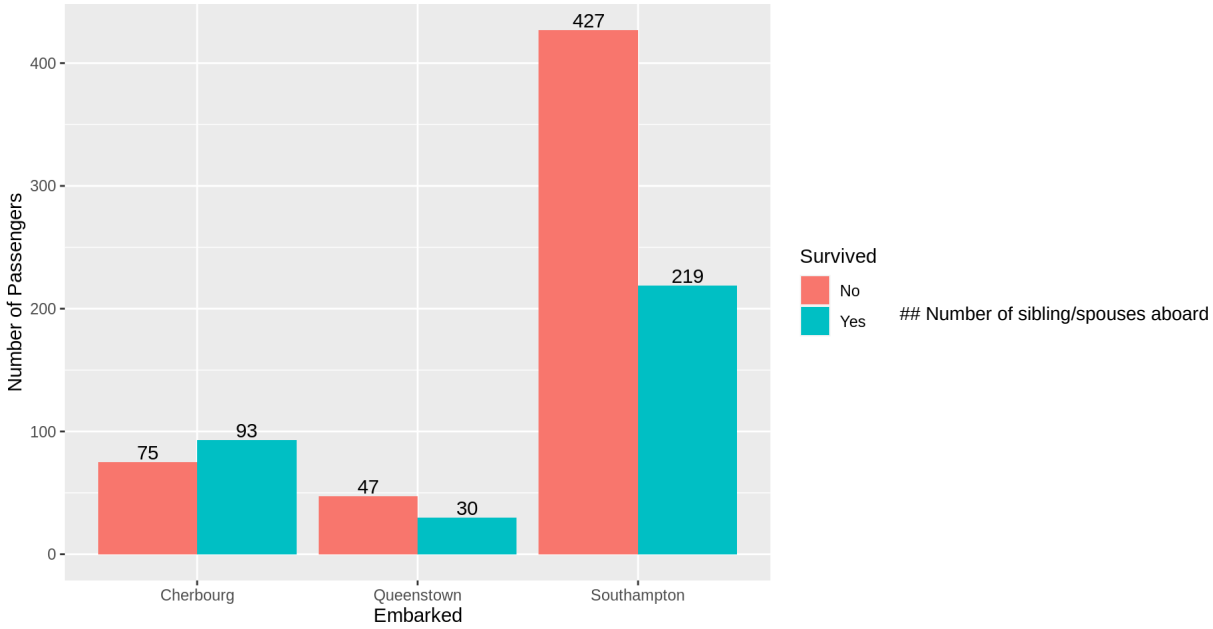
Passenger Class of most Non-Survivors

```
  ggplot(titanic, aes(x=Pclass,fill=Survived))+ geom_bar(position = "dodge") + geom_text(stat='count',aes(label
=..count..),position = position_dodge(0.9),vjust=-0.2) +
ylab("Number of Passengers") + xlab("Passenger Class")
```



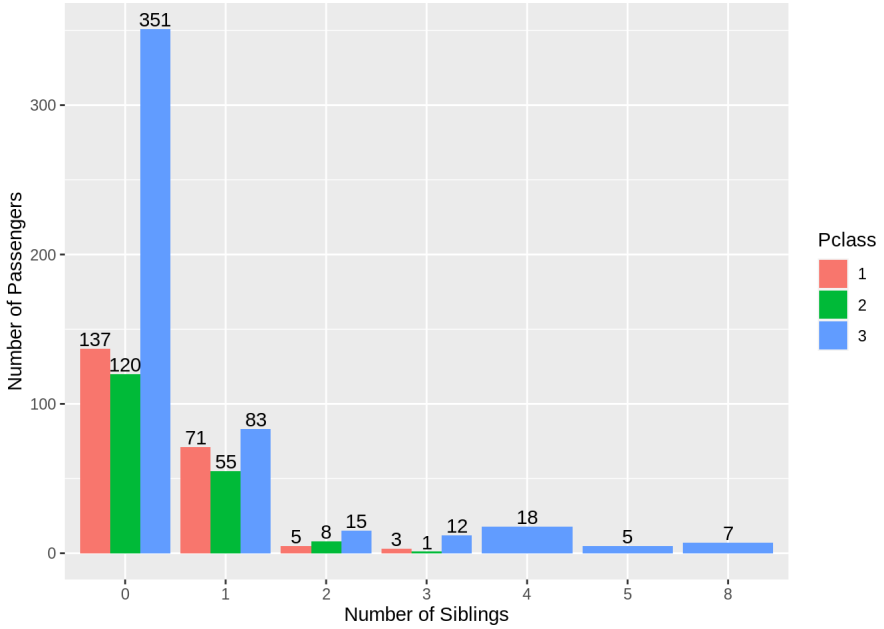
Proportion of survivors by place of Embarkment

```
ggplot(titanic, aes(x=Embarked,fill=Survived))+ geom_bar(position = "dodge") + geom_text(stat='count',aes(label
=..count..),position = position_dodge(0.9),vjust=-0.2) +
ylab("Number of Passengers")
```



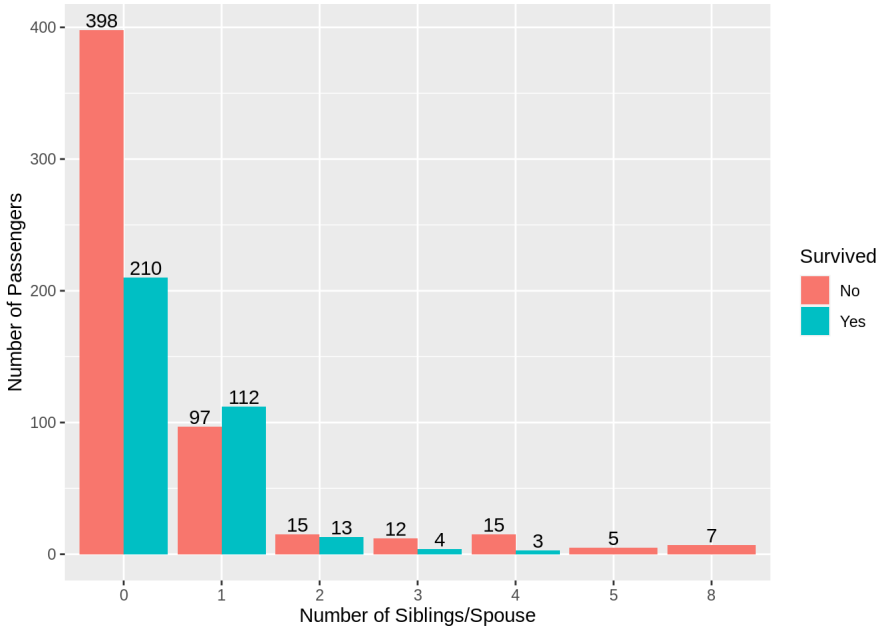
Titanic & Passenger Class

```
ggplot(titanic, aes(x=SibSp,fill=Pclass))+ geom_bar(position = "dodge") + geom_text(stat='count',aes(label=..coun
t..),position = position_dodge(0.9),vjust=-0.2) +
ylab("Number of Passengers") + xlab("Number of Siblings")
```



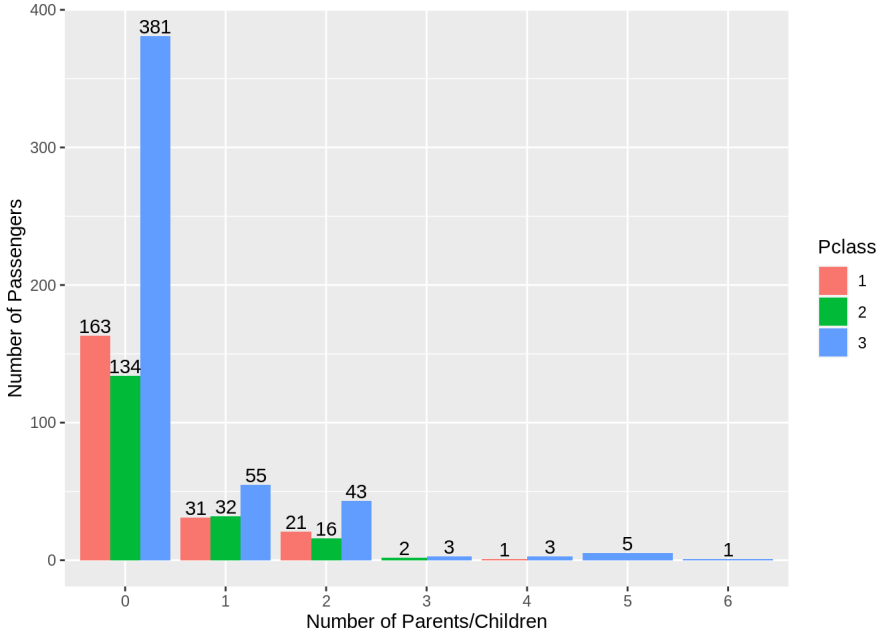
Number of sibling/spouses aboard Titanic related to Survival

```
ggplot(titanic, aes(x=SibSp,fill=Survived))+ geom_bar(position = "dodge") + geom_text(stat='count',aes(label=..co
unt..),position = position_dodge(0.9),vjust=-0.2) +
ylab("Number of Passengers")+xlab("Number of Siblings/Spouse")
```



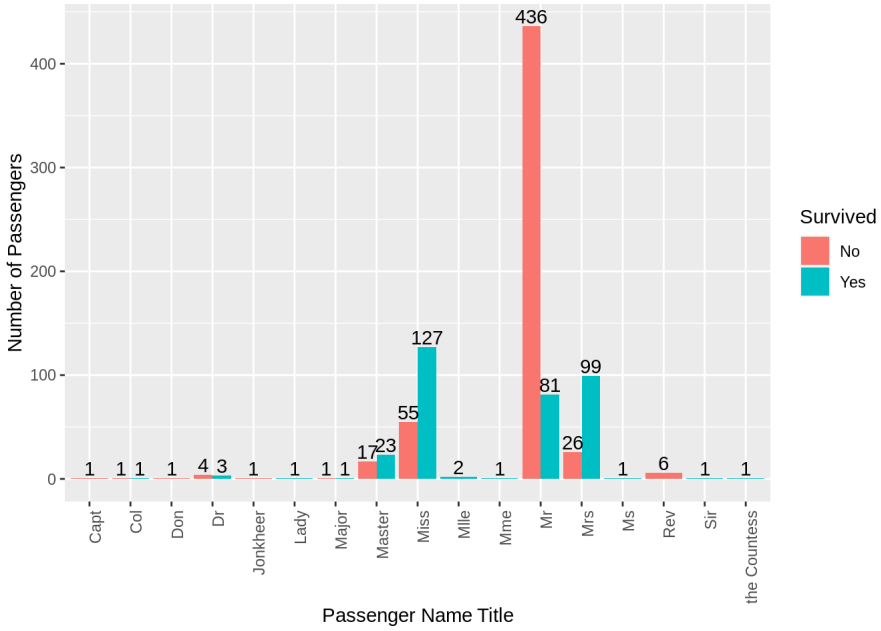
Number of parents/children aboard Titanic differ with Passenger Class

```
ggplot(titanic, aes(x=Parch,fill=Pclass))+ geom_bar(position = "dodge") + geom_text(stat='count',aes(label=..coun
t..),position = position_dodge(0.9),vjust=-0.2) +
ylab("Number of Passengers") + xlab("Number of Parents/Children")
```



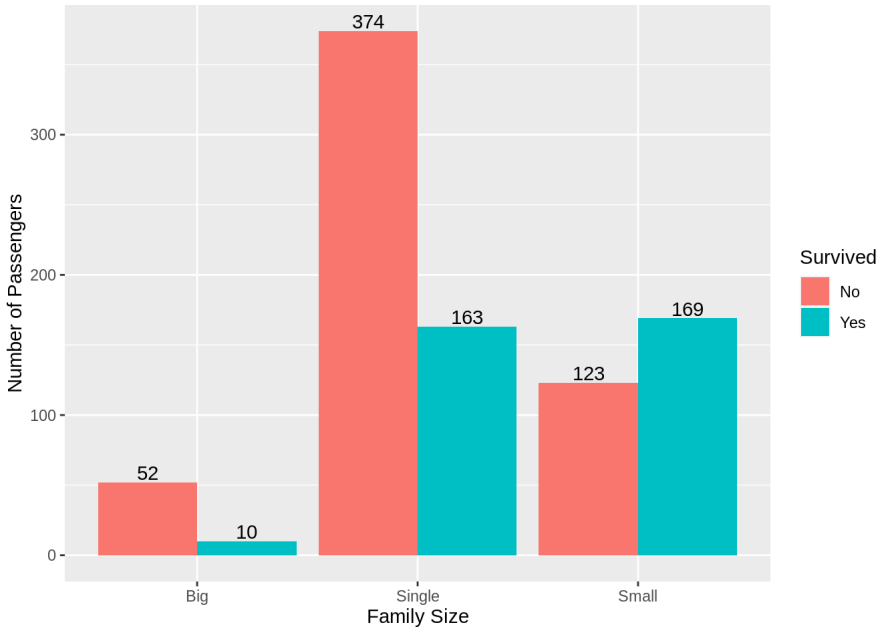
Relation between Passenger Name Title & Survival

```
ggplot(titanic, aes(x=Title,fill=Survived))+ geom_bar(position = "dodge") + geom_text(stat='count',aes(label=..co
unt..),position = position_dodge(0.9),vjust=-0.2) +
ylab("Number of Passengers") + xlab("Passenger Name Title") + theme(axis.text.x = element_text(angle = 90, hjust
 = 1))
```



Relation between Family Size & Survival

```
ggplot(titanic, aes(x=FamilySize,fill=Survived))+ geom_bar(position = "dodge") + geom_text(stat='count',aes(label
=..count..),position = position_dodge(0.9),vjust=-0.2) +
ylab("Number of Passengers") + xlab("Family Size")
```



EDA COMPLETE

USING LOGISTIC REGRESSION FOR TARGET VARIABLE (“SURVIVED”)

CREATING DUMMIES OF CATEGORICAL COLUMNS

```
library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
complete_data <- read.csv("/home/tanmay/Datasets/train.csv")

## Missing values imputation
complete_data$Embarked[complete_data$Embarked==""] <- "S"
complete_data$Age[is.na(complete_data$Age)] <- median(complete_data$Age,na.rm=T)

## Removing Cabin as it has very high missing values, passengerId, Ticket and Name are not required
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_data <- complete_data %>% select(-c(Cabin, PassengerId, Ticket, Name))

## Converting "Survived","Pclass","Sex","Embarked" to factors
for (i in c("Pclass","Sex","Embarked")){
  titanic_data[,i]=as.factor(titanic_data[,i])
}

## Create dummy variables for categorical variables
library(dummies)
titanic_data <- dummy.data.frame(titanic_data, names=c("Pclass","Sex","Embarked"), sep="_")
```

```
head(titanic_data)
```

```
##   Survived Pclass_1 Pclass_2 Pclass_3 Sex_female Sex_male Age SibSp Parch
## 1        0        0        0        1          0        1  22     1     0
## 2        1        1        0        0          1        0  38     1     0
## 3        1        0        0        1          1        0  26     0     0
## 4        1        1        0        0          1        0  35     1     0
## 5        0        0        0        1          0        1  35     0     0
## 6        0        0        0        1          0        1  28     0     0
##      Fare Embarked_C Embarked_Q Embarked_S
## 1  7.2500          0          0          1
## 2 71.2833          1          0          0
## 3  7.9250          0          0          1
## 4 53.1000          0          0          1
## 5  8.0500          0          0          1
## 6  8.4583          0          1          0
```

SPLITTING DATA INTO TRAINING AND TESTING

```
  training_data = titanic_data[1:790,]
  surv.x = training_data[,-1]
  surv.y = training_data[,1]
  testing_data = titanic_data[791:891,]
  test.x = testing_data[,-1]
  test.y = testing_data[,1]
```

Building Helper Functions

```
library(ggplot2)
library(dplyr)

#sigmoid function, inverse of logit
sigmoid <- function(z){1/(1+exp(-z))}

#cost function
cost <- function(theta, X, y){
  m <- length(y) # number of training examples
  h <- sigmoid(X %*% theta)
  J <- (t(-y)%*%log(h)-t(1-y)%*%log(1-h))/m
  J
}

#gradient function
grad <- function(theta, X, y){
  m <- length(y)

  h <- sigmoid(X%*%theta)
  grad <- (t(X)%*%(h - y))/m
  grad
}
```

```
# probability of getting 1
logisticProb <- function(theta, X){
  X <- na.omit(X)
  #add bias term and convert to matrix
  X <- mutate(X, bias =1)
  X <- as.matrix(X[,c(ncol(X), 1:(ncol(X)-1))])
  return(sigmoid(X%*%theta))
}

# y prediction
logisticPred <- function(prob){
  return(round(prob, 0))
}
```

Logistic Regression Code

```
logisticReg <- function(X, y){
  #remove NA rows
  X <- na.omit(X)
  y <- na.omit(y)
  #add bias term and convert to matrix
  X <- mutate(X, bias =1)
  #move the bias column to col1
  X <- as.matrix(X[, c(ncol(X), 1:(ncol(X)-1))])
  print(dim(X))
  # X <- as.matrix(X)
  y <- as.matrix(y)
  #initialize theta
  theta <- matrix(rep(0, ncol(X)), nrow = ncol(X))
  print(theta)
  #use the optim function to perform gradient descent
  costOpti <- optim(theta, fn = cost, gr = grad, X = X, y = y)
  #return coefficients
  return(costOpti$par)
}
```

Feed Data to Logistic Reg Function

```
mod <- logisticReg(surv.x, surv.y)
```

```
## [1] 790  13
##       [,1]
##  [1,]    0
##  [2,]    0
##  [3,]    0
##  [4,]    0
##  [5,]    0
##  [6,]    0
##  [7,]    0
##  [8,]    0
##  [9,]    0
## [10,]    0
## [11,]    0
## [12,]    0
## [13,]    0
```

```
mod
```

```
##              [,1]
##  [1,]  1.822397622
##  [2,]  1.054016653
##  [3,]  0.814370988
##  [4,] -0.131740476
##  [5,] -0.002607663
##  [6,] -2.790779911
##  [7,] -0.017867833
##  [8,] -0.373573817
##  [9,] -0.112454547
## [10,]  0.006588427
## [11,]  0.059334185
## [12,] -0.218448171
## [13,] -0.703119457
```

```
pre <- data.frame(matrix(c(1,0,0,0,1,24,0,0,60,1,0,0),nrow=1))
zpr <- logisticProb(mod,pre)
zpr
```

```
##           [,1]
## [1,] 0.5278327
```

```
ans <- logisticPred(zpr)
ans
```

```
##      [,1]
## [1,]    1
```

```
  grid <- test.x
  prob <- logisticProb(mod,grid)
  print(length(prob))
```

```
## [1] 101
```

```
  Z <- logisticPred(prob)
  print(length(Z))
```

```
## [1] 101
```

```
  gridPred = cbind(grid, Z)
```

```
results.table <- table(Z, test.y,dnn = c('Predicted','Actual'))
print(results.table)
```

```
##          Actual
## Predicted  0  1
##         0 57 12
##         1  8 24
```

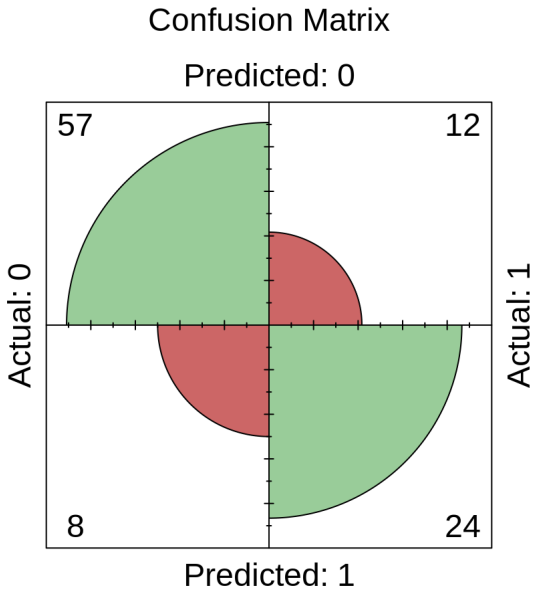
```
precision <- results.table[2,2] / (results.table[2,2] + results.table[2,1])
recal <- results.table[2,2] / (results.table[2,2] + results.table[1,2])
F1 <- 2 * precision * recal / (precision + recal)
print(paste('F1-score: ', F1))
```

```
## [1] "F1-score:  0.705882352941177"
```

```
#My Accuracy Function
accu <- function(tp,tn,tot){
  return((tp+tn)/tot)
}
acc <- accu(tp = results.table[1,1],tn = results.table[2,2],length(test.y))
print(paste('Accuracy: ', acc))
```

```
## [1] "Accuracy:  0.801980198019802"
```

```
fourfoldplot(results.table, color = c("#CC6666", "#99CC99"),
             conf.level = 0, margin = 1, main = "Confusion Matrix")
```


Confusion Matrix
Predicted: 0
57
12
Actual: 0
Actual: 1
8
24
Predicted: 1