LOGISTIC REGRESSION

1. SELECTION OF DATASET:

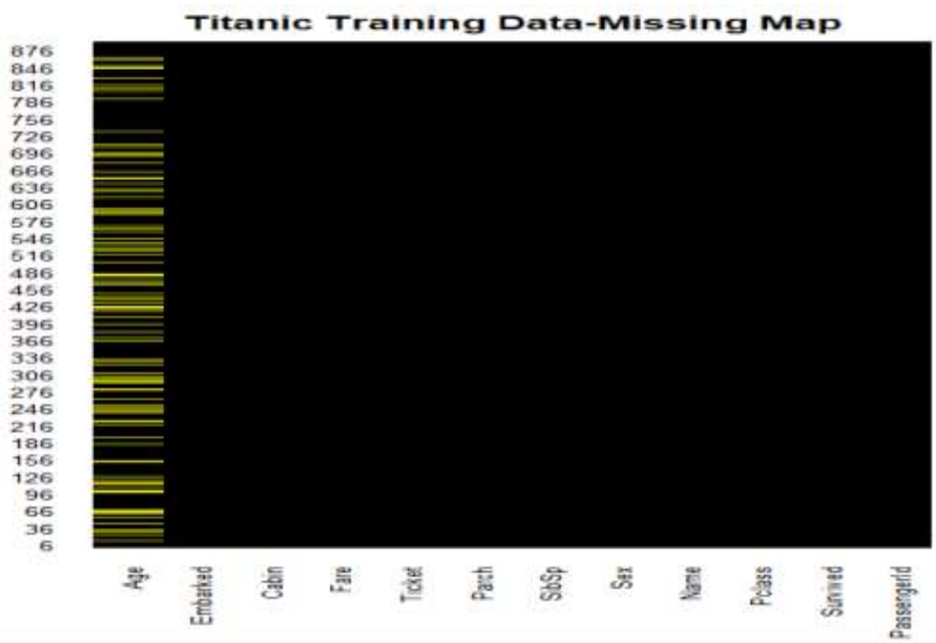
The dataset chosen here is ‘Titanic Dataset from Kaggle’. We are trying to predict a classification i.e. survived or not survived.

Dimension: 891 rows and 12 columns

```
> df.train<- read.csv('C:/Users/aishi/Desktop/Logistic_Regression/Titanic/Titanic_Dataset.csv')    #loading our training data into dataframe
> head(df.train)          #shows first six rows of the dataset
  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
1                             Braund, Mr. Owen Harris   male  22     1
2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
3                              Heikkinen, Miss. Laina female  26     0
4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1
5                            Allen, Mr. William Henry   male  35     0
6                                    Moran, Mr. James   male  NA     0
  Parch           Ticket    Fare Cabin Embarked
1     0        A/5 21171  7.2500              S
2     0         PC 17599 71.2833   C85        C
3     0 STON/O2. 3101282  7.9250              S
4     0           113803 53.1000  C123        S
5     0           373450  8.0500              S
6     0           330877  8.4583              Q
```

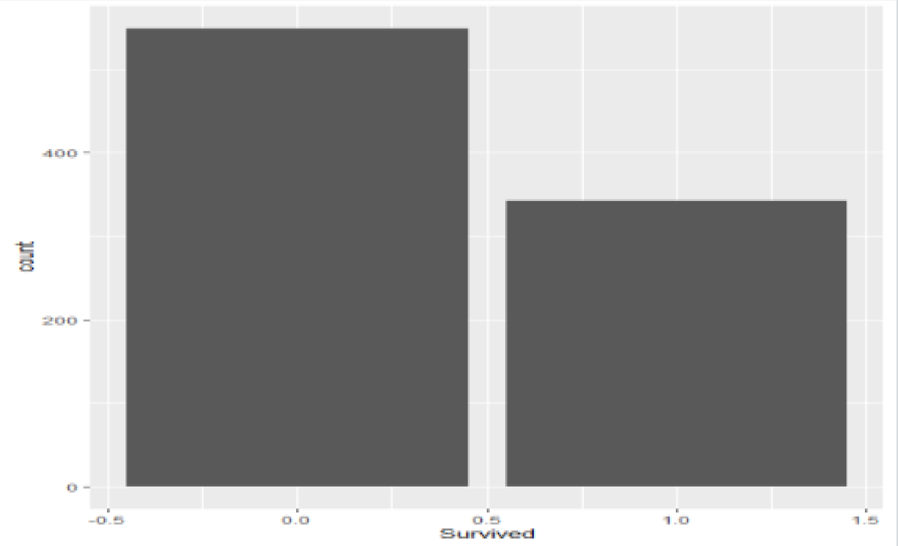
2. EXPLORATORY DATA ANALYSIS:

```
> library(Amelia)  #to explore how much missing data we have we use this package
> missmap(df.train, main='Titanic Training Data-Missing Map', col=c("yellow","black"),legend=FALSE) #yellow is missing and black means existing
```



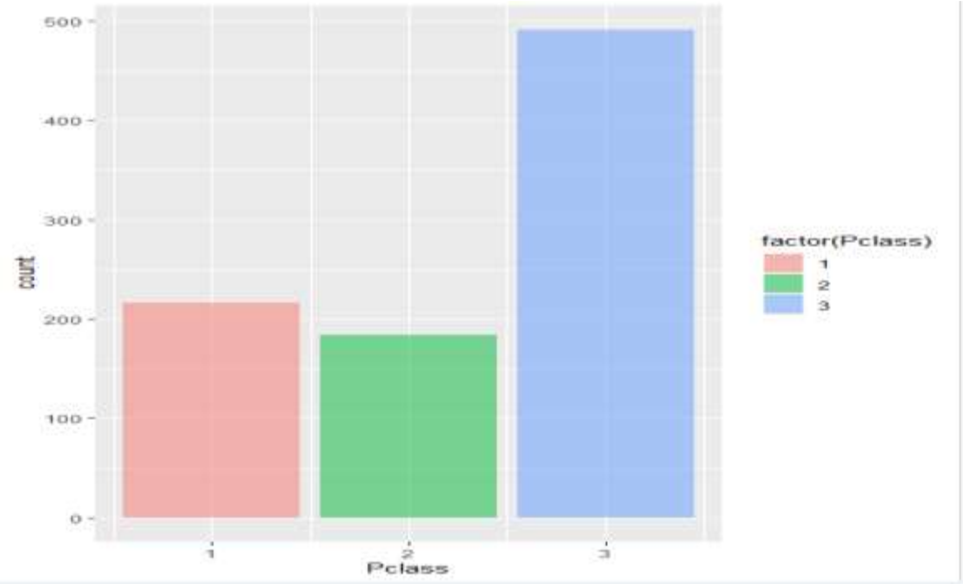
Here yellow represents missing data. We can see that roughly some percent of the ‘Age’ data is missing. Now we have to replace the missing ‘Age’ data with some imputations.

```
> #Data visualization using ggplot2
> library(ggplot2)
> ggplot(df.train, aes(Survived)) + geom_bar()
```



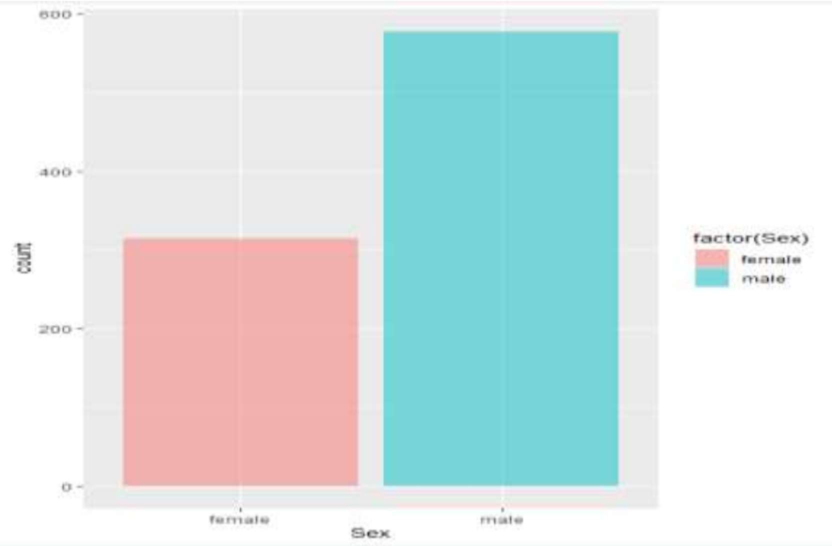
We can see from the graph that from the whole population, more people have died and less have survived.

```
> ggplot(df.train,aes(Pclass)) + geom_bar(aes(fill=factor(Pclass)),alpha=0.5)
```



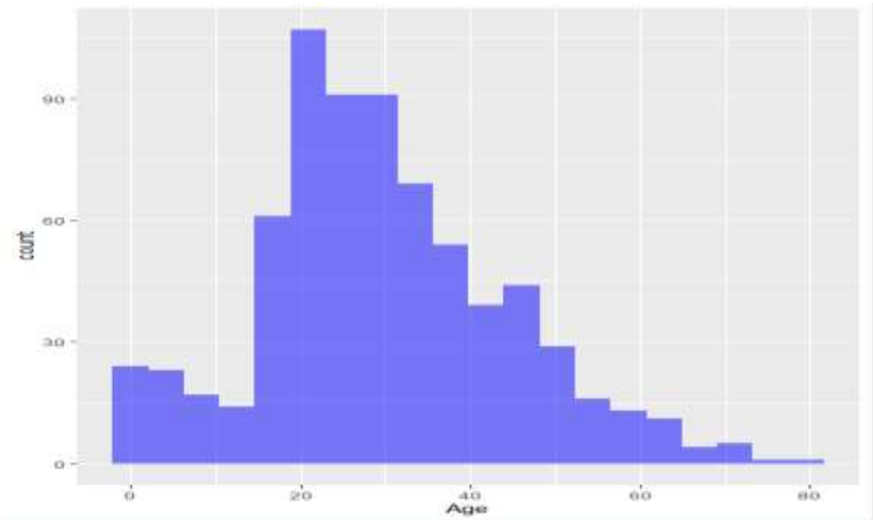
We can see from this graph that third class people were more in number than first and second class.

```
> ggplot(df.train,aes(Sex)) + geom_bar(aes(fill=factor(Sex)), alpha=0.5)
```



We can see from this graph that there were more number of males rather than females in the Titanic.

```
> ggplot(df.train,aes(Age)) + geom_histogram(fill='blue',bins=20,alpha=0.5)
Warning message:
Removed 177 rows containing non-finite values (stat_bin).
```

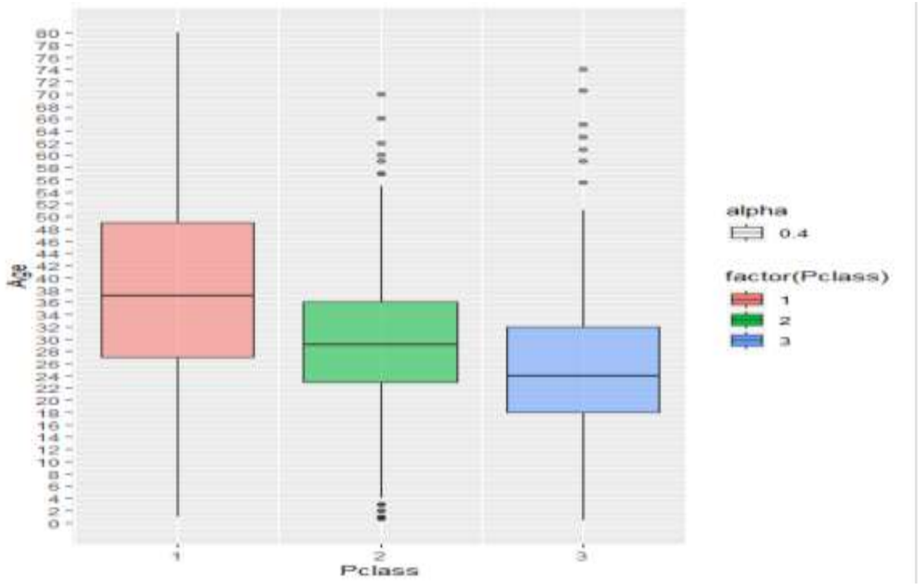


We can see form the graph that there were more number of young people in the Titanic rather than children or aged people.

3. DATA CLEANING:

We want to fill in the missing 'Age' data. So we do this is by filling in the mean age of all the passengers. We check the average age by 'Pclass'.

```
> #Data Cleaning
> #we want to fill in missing age data instead of just dropping the missing age rows
> p1<- ggplot(df.train,aes(Pclass,Age)) + geom_boxplot(aes(group=Pclass,fill=factor(Pclass),alpha=0.4))
> p1 + scale_y_continuous(breaks= seq(min(0), max(80),by=2))
Warning message:
Removed 177 rows containing non-finite values (stat_boxplot).
```

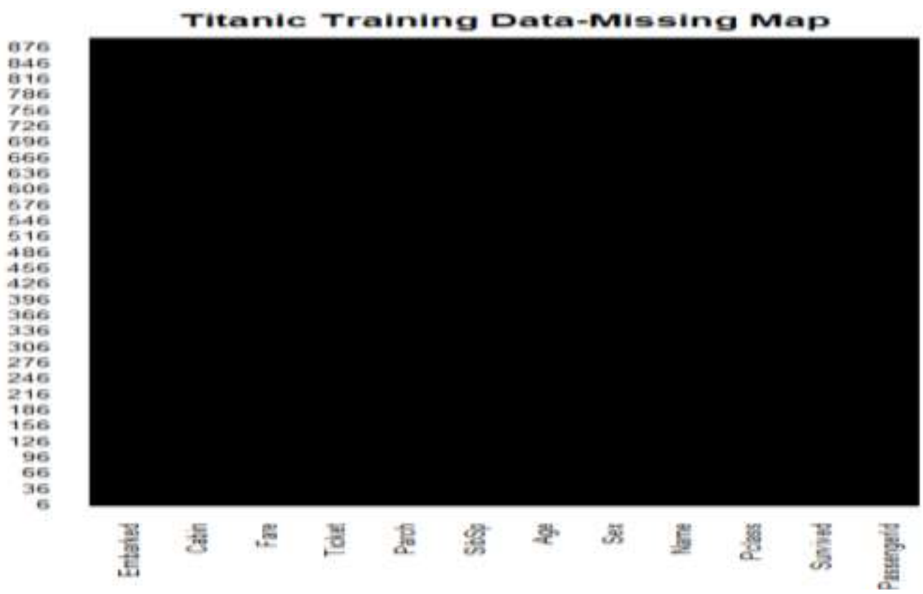


We can see that the passengers in the class 1 are mostly older and in the class 3 more people are younger in age. We will use these average age values to impute based on 'Pclass' for 'Age'.

```
> impute_age <- function(age,class){
+   out<- age
+   for (i in 1:length(age)){
+     if(is.na(age[i])) {
+       if(class[i]==1){
+         out[i] <- 37
+       }else if (class[i]==2){
+         out[i]<- 29
+       }else{
+         out[i] <- 24
+       }
+     }else{
+       out[i]<- age[i]
+     }
+   }
+   return(out)
+ }
>
> fixed.ages<- impute_age(df.train$Age, df.train$Pclass)
> df.train$Age<- fixed.ages
```

Now we will see whether the missing values got replaced or not.

```
> missmap(df.train, main='Titanic Training Data-Missing Map', col=c("yellow","black"),legend=FALSE)
```



4. BUILDING A LOGISTIC REGRESSION MODEL:

We have to build the model. We will remove the variables we won't be using and will make the features which we would be using of the correct data type.

```
> str(df.train)
'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 24 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 ...
```

We will select only the required columns for training.

```
> library(dplyr)
> df.train<- select(df.train,-PassengerId, -Name, -Ticket, -Cabin)
> head(df.train,3)
  Survived Pclass    Sex Age SibSp Parch    Fare Embarked
1        0      3   male  22     1     0  7.2500        S
2        1      1 female  38     1     0 71.2833        C
3        1      3 female  26     0     0  7.9250        S
```

We will set factor columns.

```
> str(df.train)
'data.frame':	891 obs. of  8 variables:
 $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age     : num  22 38 26 35 35 24 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 ...
 $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
> df.train$Survived <- factor(df.train$Survived)   #converting into factor
> df.train$Pclass <- factor(df.train$Pclass)
> df.train$Parch <- factor(df.train$Parch)
> df.train$SibSp <- factor(df.train$SibSp)
```

```
> str(df.train)
'data.frame':	891 obs. of  8 variables:
 $ 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 ...
 $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age     : num  22 38 26 35 35 24 54 2 27 14 ...
 $ SibSp   : Factor w/ 7 levels "0","1","2","3",..: 2 2 1 2 1 1 1 4 1 2 ...
 $ Parch   : Factor w/ 7 levels "0","1","2","3",..: 1 1 1 1 1 1 1 2 3 1 ...
 $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

```
> #Training the model
> library(caTools)
> set.seed(101)
> split=sample.split(df.train$Survived, SplitRatio=0.70)
> final.train=subset(df.train,split==TRUE)
> final.test=subset(df.train,split==FALSE)
> final.log.model <- glm(formula=Survived ~ . , family=binomial(link='logit'), data=final.train)
> summary(final.log.model)
```

```
Call:
glm(formula = Survived ~ ., family = binomial(link = "logit"),
    data = final.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8288  -0.5607  -0.4096   0.6174   2.4898

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.777e+01  2.400e+03   0.007 0.994091
Pclass2     -1.230e+00  3.814e-01  -3.225 0.001261 **
Pclass3     -2.160e+00  3.841e-01  -5.624 1.87e-08 ***
Sexmale     -2.660e+00  2.467e-01 -10.782  < 2e-16 ***
Age         -3.831e-02  1.034e-02  -3.705 0.000212 ***
SibSp1      -2.114e-02  2.755e-01  -0.077 0.938836
SibSp2      -4.000e-01  6.463e-01  -0.619 0.536028
SibSp3      -2.324e+00  8.994e-01  -2.584 0.009765 **
SibSp4      -1.196e+00  8.302e-01  -1.440 0.149839
SibSp5      -1.603e+01  9.592e+02  -0.017 0.986666
SibSp8      -1.633e+01  1.004e+03  -0.016 0.987019
Parch1       7.290e-01  3.545e-01   2.056 0.039771 *
Parch2       1.406e-01  4.504e-01   0.312 0.754892
Parch3       7.919e-01  1.229e+00   0.645 0.519226
Parch4      -1.498e+01  1.552e+03  -0.010 0.992300
Parch5      -9.772e-03  1.378e+00  -0.007 0.994343
Parch6      -1.635e+01  2.400e+03  -0.007 0.994563
Fare         3.128e-03  3.091e-03   1.012 0.311605
EmbarkedC   -1.398e+01  2.400e+03  -0.006 0.995353
EmbarkedQ   -1.387e+01  2.400e+03  -0.006 0.995386
EmbarkedS   -1.431e+01  2.400e+03  -0.006 0.995243
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 829.60  on 622  degrees of freedom
Residual deviance: 530.63  on 602  degrees of freedom
AIC: 572.63

Number of Fisher Scoring iterations: 15
```

We can clearly see that Pclass3, Sexmale and Age have the most significant features. Pclass2 and Parch1 have less significant feature. The other features have no significant features in the prediction.

```
> #Check the Prediction Accuracy
> fitted.probabilities <- predict(final.log.model,newdata=final.test, type='response')
> #Calculate from predicted values
> fitted.results<- ifelse(fitted.probabilities >0.5,1,0)
> misClasificError <- mean(fitted.results != final.test$Survived)
> print(paste('Accuracy:',1-misClasificError))
[1] "Accuracy: 0.798507462686567"
```

We can say that the model has achieved 79.85% accuracy.

```
> #Creating the confusion matrix
> table(final.test$Survived, fitted.probabilities > 0.5)

    FALSE TRUE
  0   140   25
  1    29   74
```

From the confusion matrix we can infer that:

- 140 people were predicted to die and they have actually died.
- 25 people were predicted to survive and they have actually died.
- 29 people were predicted to die and they have survived.
- 74 people were predicted to survive and they have actually survived.