# **Project** titanic-train.csv

titanic-train data is in the form of csv file. This data comes under classification data. The data we are predicting is survived and it is in the form of 0 and 1.

To read this file we need to do following steps in R

| **Commands** | **Description** |
|---|---|
| getwd() | To know about the position of the directory |
| Setwd() | To change the current directory |
| read.csv() | To read the file. |

**[In] : setwd("C://Users//ADMIN//Desktop")**

In windows we need to change the forward slash to backward slash in order to overcome the error.

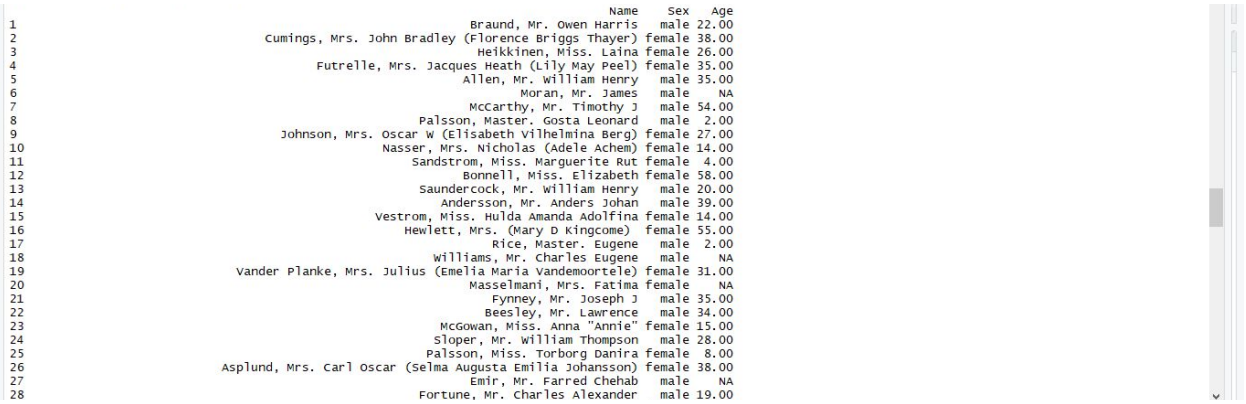
**[In]: data<-read.csv("titanic_train.csv")**

data is used to give other name for titanic data.

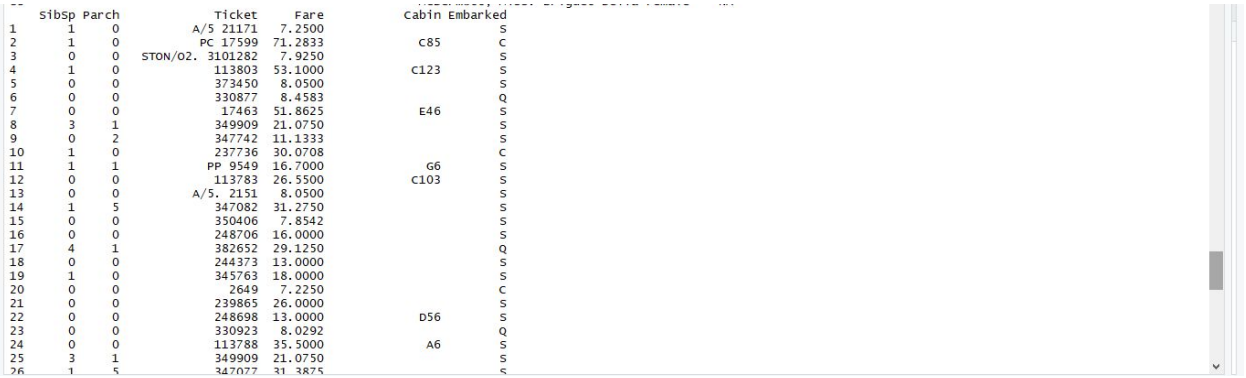
**[In]: data**

**[Out]:**

```
> setwd("C://Users//ADMIN//Desktop")
> data<-read.csv("titanic_train.csv")
> 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
7             7        0      1
8             8        0      3
9             9        1      3
10           10        1      2
11           11        1      3
12           12        1      1
13           13        0      3
14           14        0      3
15           15        0      3
16           16        1      2
17           17        0      3
18           18        1      2
19           19        0      3
20           20        1      3
21           21        0      2
22           22        1      2
23           23        1      3
24           24        1      1
25           25        0      3
26           26        1      3
27           27        0      3
28           28        0      1
29           29        1      3
30           30        0      3
```



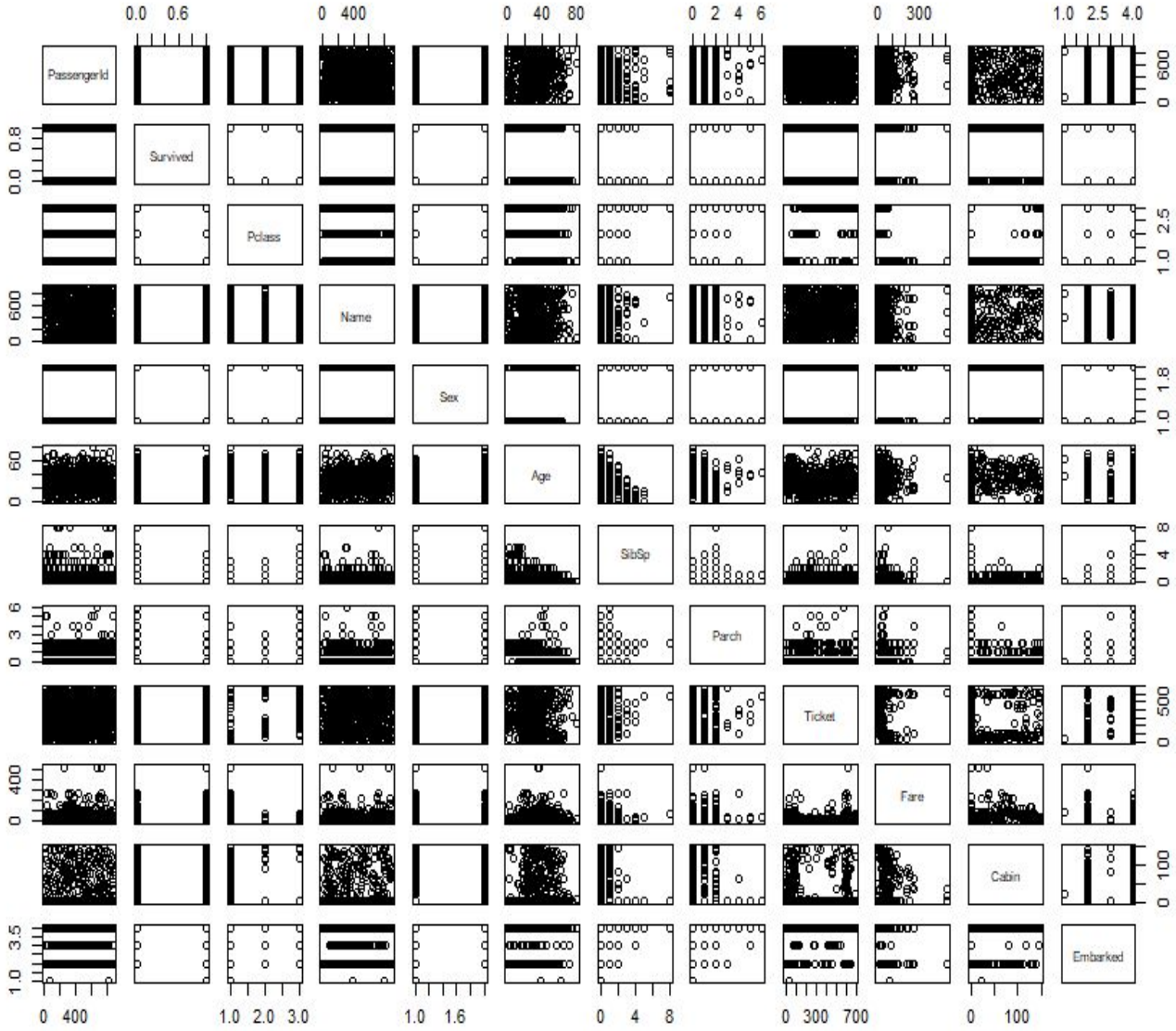
| | Name | Sex | Age |
|---|---|---|---|
| 1 | Braund, Mr. Owen Harris | male | 22.00 |
| 2 | Cumings, Mrs. John Bradley (Florence Briggs Thayer) | female | 38.00 |
| 3 | Heikkinen, Miss. Laina | female | 26.00 |
| 4 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.00 |
| 5 | Allen, Mr. William Henry | male | 35.00 |
| 6 | Moran, Mr. James | male | NA |
| 7 | McCarthy, Mr. Timothy J | male | 54.00 |
| 8 | Palsson, Master. Gosta Leonard | male | 2.00 |
| 9 | Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) | female | 27.00 |
| 10 | Nasser, Mrs. Nicholas (Adele Achem) | female | 14.00 |
| 11 | Sandstrom, Miss. Marguerite Rut | female | 4.00 |
| 12 | Bonnell, Miss. Elizabeth | female | 58.00 |
| 13 | Saundercock, Mr. William Henry | male | 20.00 |
| 14 | Andersson, Mr. Anders Johan | male | 39.00 |
| 15 | Vestrom, Miss. Hulda Amanda Adolfina | female | 14.00 |
| 16 | Hewlett, Mrs. (Mary D Kingcome) | female | 55.00 |
| 17 | Rice, Master. Eugene | male | 2.00 |
| 18 | Williams, Mr. Charles Eugene | male | NA |
| 19 | Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele) | female | 31.00 |
| 20 | Masselmani, Mrs. Fatima | female | NA |
| 21 | Fynney, Mr. Joseph J | male | 35.00 |
| 22 | Beesley, Mr. Lawrence | male | 34.00 |
| 23 | McGowan, Miss. Anna "Annie" | female | 15.00 |
| 24 | Sloper, Mr. William Thompson | male | 28.00 |
| 25 | Palsson, Miss. Torborg Danira | female | 8.00 |
| 26 | Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson) | female | 38.00 |
| 27 | Emir, Mr. Farred Chehab | male | NA |
| 28 | Fortune, Mr. Charles Alexander | male | 19.00 |



| | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | A/5 21171 | 7.2500 | | S |
| 2 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 0 | 0 | STON/O2. 3101282 | 7.9250 | | S |
| 4 | 1 | 0 | 113803 | 53.1000 | C123 | S |
| 5 | 0 | 0 | 373450 | 8.0500 | | S |
| 6 | 0 | 0 | 330877 | 8.4583 | | Q |
| 7 | 0 | 0 | 17463 | 51.8625 | E46 | S |
| 8 | 3 | 1 | 349909 | 21.0750 | | S |
| 9 | 0 | 2 | 347742 | 11.1333 | | S |
| 10 | 1 | 0 | 237736 | 30.0708 | | C |
| 11 | 1 | 1 | PP 9549 | 16.7000 | G6 | S |
| 12 | 0 | 0 | 113783 | 26.5500 | C103 | S |
| 13 | 0 | 0 | A/5. 2151 | 8.0500 | | S |
| 14 | 1 | 5 | 347082 | 31.2750 | | S |
| 15 | 0 | 0 | 350406 | 7.8542 | | S |
| 16 | 0 | 0 | 248706 | 16.0000 | | S |
| 17 | 4 | 1 | 382652 | 29.1250 | | Q |
| 18 | 0 | 0 | 244373 | 13.0000 | | S |
| 19 | 1 | 0 | 345763 | 18.0000 | | S |
| 20 | 0 | 0 | 2649 | 7.2250 | | C |
| 21 | 0 | 0 | 239865 | 26.0000 | | S |
| 22 | 0 | 0 | 248698 | 13.0000 | D56 | S |
| 23 | 0 | 0 | 330923 | 8.0292 | | Q |
| 24 | 0 | 0 | 113788 | 35.5000 | A6 | S |
| 25 | 3 | 1 | 349909 | 21.0750 | | S |
| 26 | 1 | 5 | 347077 | 31.3875 | | S |

**[In] plot(data)**

I




Above graphs describes the entire data. By seeing graphs we can conclude that there is so much unnecessary data in titanic file and also we can say that it is not comes under linear and logistic format.

So now remove the unnecessary data.

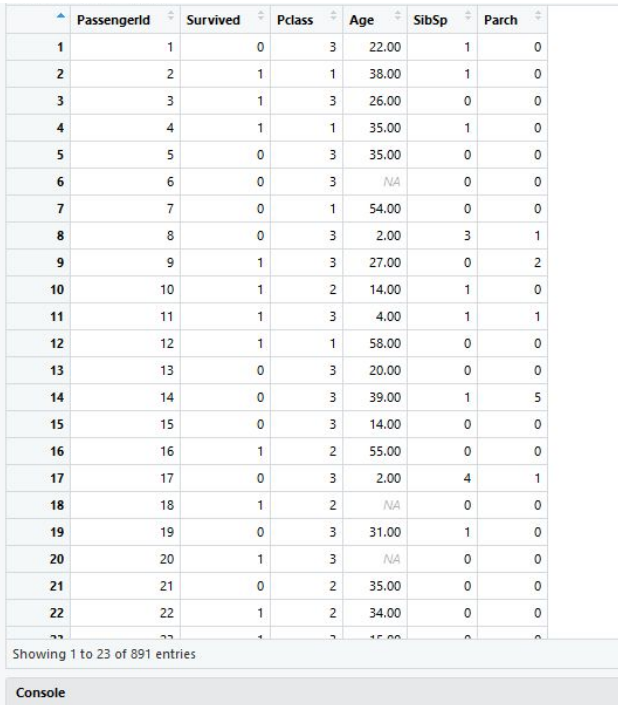
**[In]: data<-data[,-(4:5)]**

**[In]: data<-data[,-(7:12)]**

In R indexing starts from 1 so we are removing Names, Sex  and other columns

**[In] :View(data)**

**[Out]:**



| | PassengerId | Survived | Pclass | Age | SibSp | Parch |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 3 | 22.00 | 1 | 0 |
| 2 | 2 | 1 | 1 | 38.00 | 1 | 0 |
| 3 | 3 | 1 | 3 | 26.00 | 0 | 0 |
| 4 | 4 | 1 | 1 | 35.00 | 1 | 0 |
| 5 | 5 | 0 | 3 | 35.00 | 0 | 0 |
| 6 | 6 | 0 | 3 | NA | 0 | 0 |
| 7 | 7 | 0 | 1 | 54.00 | 0 | 0 |
| 8 | 8 | 0 | 3 | 2.00 | 3 | 1 |
| 9 | 9 | 1 | 3 | 27.00 | 0 | 2 |
| 10 | 10 | 1 | 2 | 14.00 | 1 | 0 |
| 11 | 11 | 1 | 3 | 4.00 | 1 | 1 |
| 12 | 12 | 1 | 1 | 58.00 | 0 | 0 |
| 13 | 13 | 0 | 3 | 20.00 | 0 | 0 |
| 14 | 14 | 0 | 3 | 39.00 | 1 | 5 |
| 15 | 15 | 0 | 3 | 14.00 | 0 | 0 |
| 16 | 16 | 1 | 2 | 55.00 | 0 | 0 |
| 17 | 17 | 0 | 3 | 2.00 | 4 | 1 |
| 18 | 18 | 1 | 2 | NA | 0 | 0 |
| 19 | 19 | 0 | 3 | 31.00 | 1 | 0 |
| 20 | 20 | 1 | 3 | NA | 0 | 0 |
| 21 | 21 | 0 | 2 | 35.00 | 0 | 0 |
| 22 | 22 | 1 | 2 | 34.00 | 0 | 0 |

Showing 1 to 23 of 891 entries

Console

In the above output there are NA in the Age . So that must be replaced by other data in order to get good prediction.

**[In]: data$Age=ifelse(is.na(data$Age),ave(data$Age,FUN = function(x)mean(x,na.rm = TRUE)),data$Age)**

**[In]:library(caTools)**

CaTools is the one of the package in the R which is used for training and testing the data.

**[In]: split=sample.split(data$Survived,SplitRatio = 0.88)**

*Here we are going to consider Survived as the dependent data and Age as the independent data.*

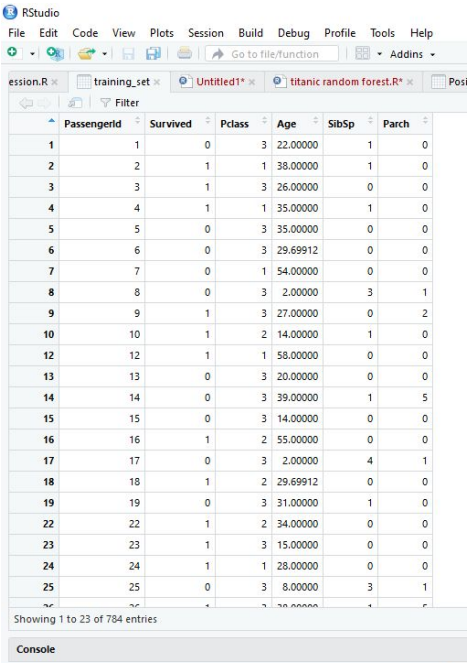
Split ration must be considered in such a way that training data must contain more data than the testing data.

**[In]:training_set=subset(data,split==TRUE)**

**[In]: testing_set=subset(data,split==FALSE)**

**[In]:View(training_set)**

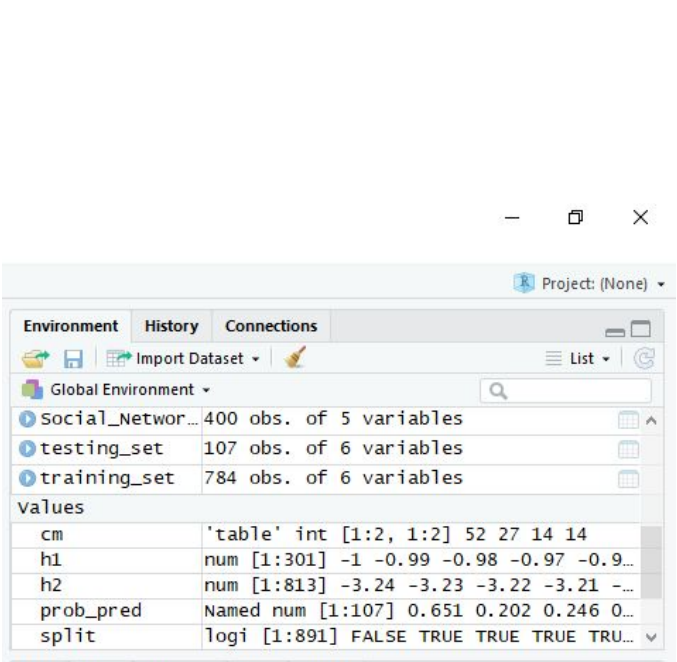
**[Out]:**

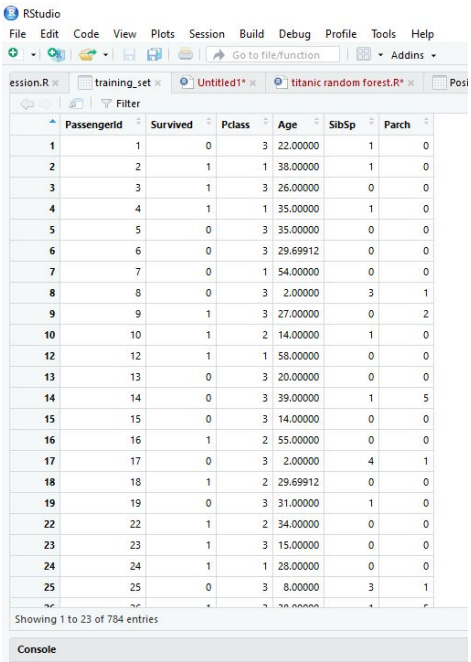



| | PassengerId | Survived | Pclass | Age | SibSp | Parch |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 3 | 22.00000 | 1 | 0 |
| 2 | 2 | 1 | 1 | 38.00000 | 1 | 0 |
| 3 | 3 | 1 | 3 | 26.00000 | 0 | 0 |
| 4 | 4 | 1 | 1 | 35.00000 | 1 | 0 |
| 5 | 5 | 0 | 3 | 35.00000 | 0 | 0 |
| 6 | 6 | 0 | 3 | 29.69912 | 0 | 0 |
| 7 | 7 | 0 | 1 | 54.00000 | 0 | 0 |
| 8 | 8 | 0 | 3 | 2.00000 | 3 | 1 |
| 9 | 9 | 1 | 3 | 27.00000 | 0 | 2 |
| 10 | 10 | 1 | 2 | 14.00000 | 1 | 0 |
| 12 | 12 | 1 | 1 | 58.00000 | 0 | 0 |
| 13 | 13 | 0 | 3 | 20.00000 | 0 | 0 |
| 14 | 14 | 0 | 3 | 39.00000 | 1 | 5 |
| 15 | 15 | 0 | 3 | 14.00000 | 0 | 0 |
| 16 | 16 | 1 | 2 | 55.00000 | 0 | 0 |
| 17 | 17 | 0 | 3 | 2.00000 | 4 | 1 |
| 18 | 18 | 1 | 2 | 29.69912 | 0 | 0 |
| 19 | 19 | 0 | 3 | 31.00000 | 1 | 0 |
| 22 | 22 | 1 | 2 | 34.00000 | 0 | 0 |
| 23 | 23 | 1 | 3 | 15.00000 | 0 | 0 |
| 24 | 24 | 1 | 1 | 28.00000 | 0 | 0 |
| 25 | 25 | 0 | 3 | 8.00000 | 3 | 1 |

**[In]:View(testing_set)**

**[Out]:**




| | PassengerId | Survived | Pclass | Age | SibSp | Parch |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 3 | 22.00000 | 1 | 0 |
| 2 | 2 | 1 | 1 | 38.00000 | 1 | 0 |
| 3 | 3 | 1 | 3 | 26.00000 | 0 | 0 |
| 4 | 4 | 1 | 1 | 35.00000 | 1 | 0 |
| 5 | 5 | 0 | 3 | 35.00000 | 0 | 0 |
| 6 | 6 | 0 | 3 | 29.69912 | 0 | 0 |
| 7 | 7 | 0 | 1 | 54.00000 | 0 | 0 |
| 8 | 8 | 0 | 3 | 2.00000 | 3 | 1 |
| 9 | 9 | 1 | 3 | 27.00000 | 0 | 2 |
| 10 | 10 | 1 | 2 | 14.00000 | 1 | 0 |
| 12 | 12 | 1 | 1 | 58.00000 | 0 | 0 |
| 13 | 13 | 0 | 3 | 20.00000 | 0 | 0 |
| 14 | 14 | 0 | 3 | 39.00000 | 1 | 5 |
| 15 | 15 | 0 | 3 | 14.00000 | 0 | 0 |
| 16 | 16 | 1 | 2 | 55.00000 | 0 | 0 |
| 17 | 17 | 0 | 3 | 2.00000 | 4 | 1 |
| 18 | 18 | 1 | 2 | 29.69912 | 0 | 0 |
| 19 | 19 | 0 | 3 | 31.00000 | 1 | 0 |
| 22 | 22 | 1 | 2 | 34.00000 | 0 | 0 |
| 23 | 23 | 1 | 3 | 15.00000 | 0 | 0 |
| 24 | 24 | 1 | 1 | 28.00000 | 0 | 0 |
| 25 | 25 | 0 | 3 | 8.00000 | 3 | 1 |

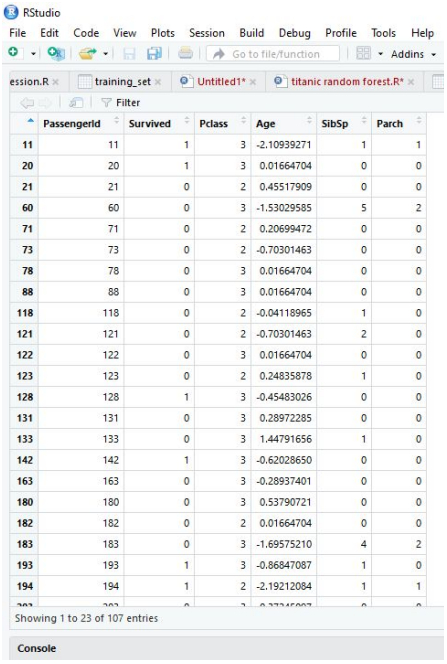
In above output we can see that testing set having 107 obs., and training set having 784 obs. It's just because of split ratio.

Also we can see that range of Age is not proper format so we need to scale them both in training and testing set.

**[In]: training_set[,4]=scale(training_set[,4])**

**[In]:testing_set[,4]=scale(testing_set[,4])**

**[out]:**




| PassengerId | Survived | Pclass | Age | SibSp | Parch |
|---|---|---|---|---|---|
| 11 | 1 | 3 | -2.10939271 | 1 | 1 |
| 20 | 1 | 3 | 0.01664704 | 0 | 0 |
| 21 | 0 | 2 | 0.45517909 | 0 | 0 |
| 60 | 0 | 3 | -1.53029585 | 5 | 2 |
| 71 | 0 | 2 | 0.20699472 | 0 | 0 |
| 73 | 0 | 2 | -0.70301463 | 0 | 0 |
| 78 | 0 | 3 | 0.01664704 | 0 | 0 |
| 88 | 0 | 3 | 0.01664704 | 0 | 0 |
| 118 | 0 | 2 | -0.04118965 | 1 | 0 |
| 121 | 0 | 2 | -0.70301463 | 2 | 0 |
| 122 | 0 | 3 | 0.01664704 | 0 | 0 |
| 123 | 0 | 2 | 0.24835878 | 1 | 0 |
| 128 | 1 | 3 | -0.45483026 | 0 | 0 |
| 131 | 0 | 3 | 0.28972285 | 0 | 0 |
| 133 | 0 | 3 | 1.44791656 | 1 | 0 |
| 142 | 1 | 3 | -0.62028650 | 0 | 0 |
| 163 | 0 | 3 | -0.28937401 | 0 | 0 |
| 180 | 0 | 3 | 0.53790721 | 0 | 0 |
| 182 | 0 | 2 | 0.01664704 | 0 | 0 |
| 183 | 0 | 3 | -1.69575210 | 4 | 2 |
| 193 | 1 | 3 | -0.86847087 | 1 | 0 |
| 194 | 1 | 2 | -2.19212084 | 1 | 1 |

We can see in the above o/p the age is scaled.

**Note:** *Above steps are similar for all the models.*

| **Random Forest** | **Decision Tree** |
|---|---|
| **[In]:install.packages("randomForest")**<br><br>For random forest model we need to install the random forest package and then give the command library to select the randomForest package.<br><br>**[In]:library(randomForest)**<br><br>**[In]:classifier<-randomForest(x=training_set[-4],y =training_set$Survived,ntree = 10)** | **[In]:install.packages("rpart")**<br><br>For Decision Tree model we need to install the rpart package and then give the command library to select the Decision package.<br><br>**[In]:library(rpart)**<br><br>**[In]:classifier=rpart(formula = Survived~.,data = training_set)** |

Here ntree means its forms the number of tree.

**[In]:y_pred=predict(classifier,newdata = testing_set[,4])**

We need to predict the test set.

**[In]:y_pred**

**[Out]:**

```
> y_pred
  2   5   9  14  27  35  36  44  50  74  76  77  79  97 102 107 115 124 127 148 153
3 216 235 238 245 249 254 257 260 261 283 286 293
  1   0   0   0   0   1   1   1   0   0   0   0   1   0   0   0   0   0   0   0   0
0   1   0   1   0   1   0   1   0   0   0   0   0   0
311 315 326 338 344 356 358 362 373 377 381 388 392 420 436 442 445 447 448 454 456
4 502 523 536 554 558 581 590 599 626 627 635 639
  1   0   1   1   0   0   0   0   0   0   1   0   0   0   1   0   0   1   1   1   0
0   0   0   1   0   1   0   0   0   0   0   0   0   0
657 665 672 674 675 679 683 688 692 697 699 711 740 745 753 759 781 797 807 810 838
8 870 876
  0   0   1   0   0   0   0   0   0   0   1   1   0   0   0   0   0   1   1   1   0
1   0   0
```

**[In]:cm=table(testing_set[,2],y_pred)**

CM means confusion matrix where we find the prediction value of testing set and predicted testing set.

**[In]:cm**
**[Out]:**

```
> cm=table(testing_set[,2],y_pred)
> cm
   y_pred
     0  1
  0 48 18
  1 31 10
```

**(52+14)/(52+14+27+14)**

```
[1] 0.6168224
```

So we got the 61% prediction value.

Here the formula always start with dependent value then followed by independent value. If there are multiple independent values we can use ~., instead of +.

**[In]:y_pred=predict(classifier,newdata = testing_set[,4],type = 'class')**

We need to predict the test set.
**[In]:y_pred**

**[Out]:**

```
> y_pred
  2   5   9  14  27  35  36  44  50  74  76  77  79  97 102 107 115 124 127 148 153
3 216 235 238 245 249 254 257 260 261 283 286 293
  1   0   0   0   0   1   1   1   0   0   0   0   1   0   0   0   0   0   0   0   0
0   1   0   1   0   1   0   1   0   0   0   0   0   0
311 315 326 338 344 356 358 362 373 377 381 388 392 420 436 442 445 447 448 454 456
4 502 523 536 554 558 581 590 599 626 627 635 639
  1   0   1   1   0   0   0   0   0   0   1   0   0   0   1   0   0   1   1   1   0
0   0   0   1   0   1   0   0   0   0   0   0   0   0
657 665 672 674 675 679 683 688 692 697 699 711 740 745 753 759 781 797 807 810 838
8 870 876
  0   0   1   0   0   0   0   0   0   0   1   1   0   0   0   0   0   1   1   1   0
1   0   0
```

**[In]:cm=table(testing_set[,2],y_pred)**

CM means confusion matrix where we find the prediction value of testing set and predicted testing set.

**[In]:cm**
**[Out]:**

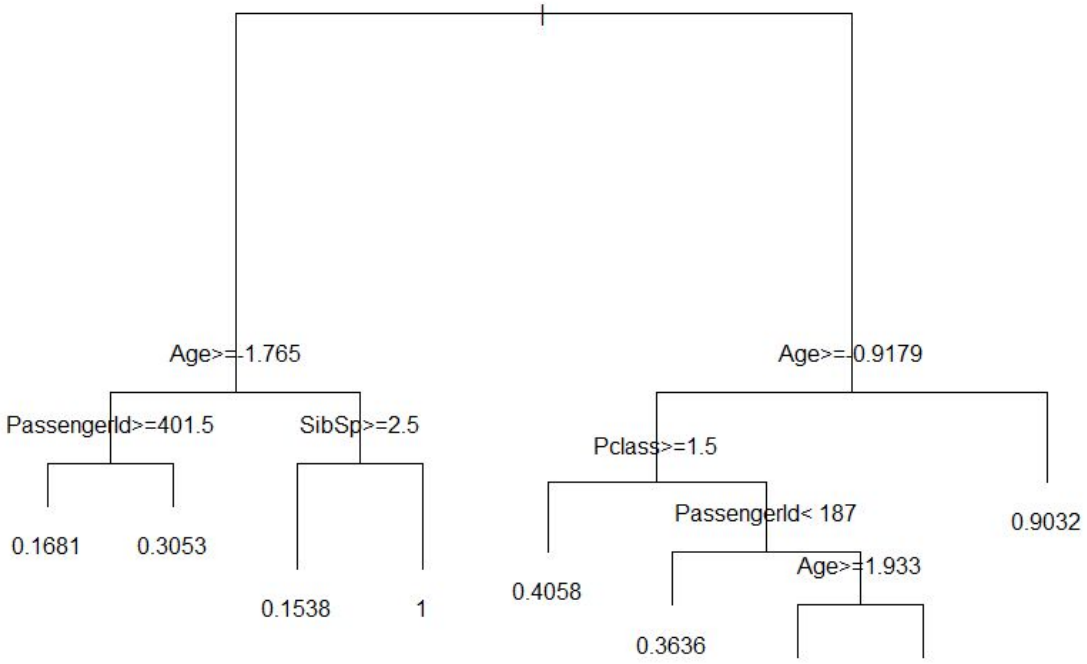
```
> cm=table(testing_set[,2],y_pred)
> cm
   y_pred
     0  1
  0 48 18
  1 31 10
```

**(48+10)/(48+10+31+18)**
**[1] 0.5420561**

So we got the 54% prediction value

**Graph for Decision Tree:**

```
> plot(classifier)
> text(classifier)
```




**Note***: So comparative both models the best predictive value we got for the random forest model. So we can conclude that random forest model is the best model for the titanic data.*