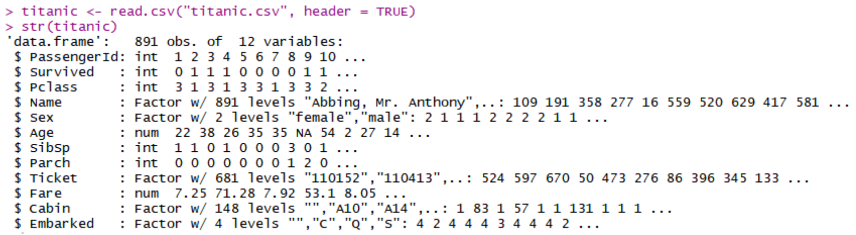
Data Analysis & Data Mining

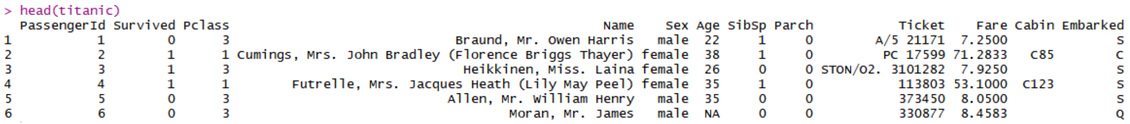
Assignment 3

Submission Date	January 4th, 2019
Subject	Data Analysis & Data Mining
Professor	Seokwoo Song Prof.
department	Industrial Management Engineering
Student ID	201401210
name	Hyeongwon Kang

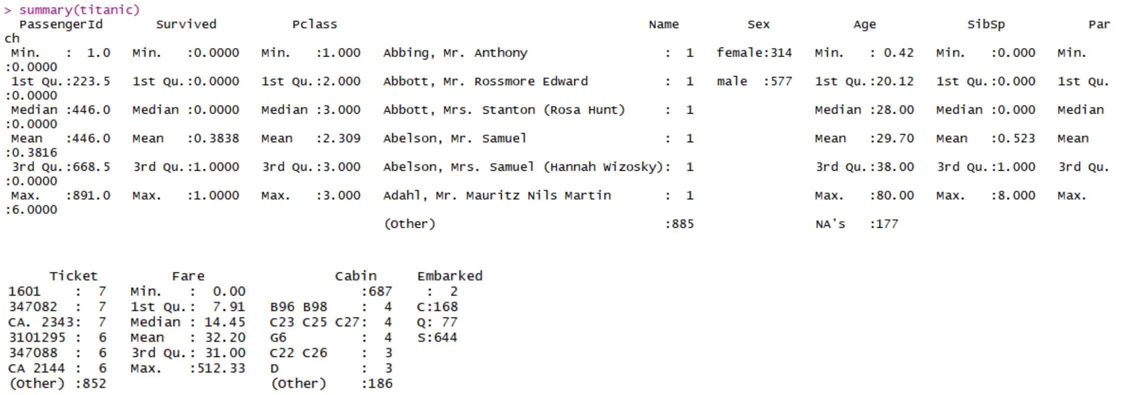
1. **To load Titanic data and look at basic information of the data**



```
> titanic <- read.csv("titanic.csv", header = TRUE)
> 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 ...
```

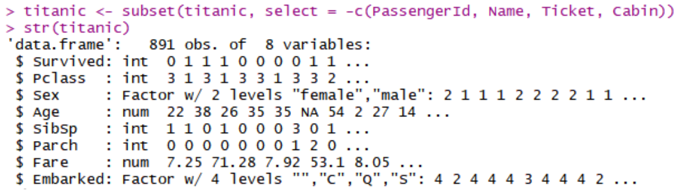


```
> head(titanic)
  PassengerId Survived Pclass                                                Name    Sex Age SibSp Parch           Ticket    Fare Cabin Embarked
1           1        0      3                             Braund, Mr. Owen Harris   male  22     1     0        A/5 21171  7.2500                S
2           2        1      1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0         PC 17599 71.2833   C85          C
3           3        1      3                              Heikkinen, Miss. Laina female  26     0     0 STON/O2. 3101282  7.9250                S
4           4        1      1        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0           113803 53.1000  C123          S
5           5        0      3                            Allen, Mr. William Henry   male  35     0     0           373450  8.0500                S
6           6        0      3                                    Moran, Mr. James   male  NA     0     0           330877  8.4583                Q
```



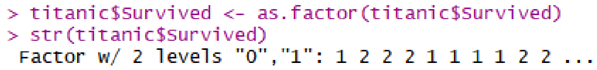
```
> summary(titanic)
  PassengerId       Survived          Pclass                                         Name         Sex           Age            SibSp           Par
ch
 Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Abbing, Mr. Anthony                  :  1   female:314   Min.   : 0.42   Min.   :0.000   Min.
:0.0000
 1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Abbott, Mr. Rossmore Edward          :  1   male  :577   1st Qu.:20.12   1st Qu.:0.000   1st Qu.
:0.0000
 Median :446.0   Median :0.0000   Median :3.000   Abbott, Mrs. Stanton (Rosa Hunt)     :  1                Median :28.00   Median :0.000   Median
:0.0000
 Mean   :446.0   Mean   :0.3838   Mean   :2.309   Abelson, Mr. Samuel                  :  1                Mean   :29.70   Mean   :0.523   Mean
:0.3816
 3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000   Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.
:0.0000
 Max.   :891.0   Max.   :1.0000   Max.   :3.000   Adahl, Mr. Mauritz Nils Martin       :  1                Max.   :80.00   Max.   :8.000   Max.
:6.0000
                                                  (Other)                              :885                NA's   :177


      Ticket          Fare                Cabin     Embarked
 1601    :  7   Min.   :  0.00              :687     :  2
 347082  :  7   1st Qu.:  7.91   B96 B98    :  4   C:168
 CA. 2343:  7   Median : 14.45   C23 C25 C27:  4   Q: 77
 3101295 :  6   Mean   : 32.20   G6         :  4   S:644
 347088  :  6   3rd Qu.: 31.00   C22 C26    :  3
 CA 2144 :  6   Max.   :512.33   D          :  3
 (Other) :852                    (Other)    :186
```

2. **To create a new dataset without having the fields (passenger Id, name, ticket, and cabin)**



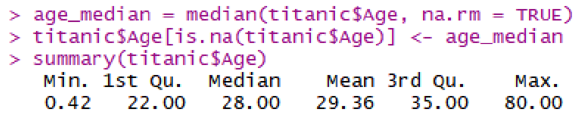
```
> titanic <- subset(titanic, select = -c(PassengerId, Name, Ticket, Cabin))
> str(titanic)
'data.frame':   891 obs. of  8 variables:
 $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age     : num  22 38 26 35 35 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 ...
 $ 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 ...
```

3. **To change the current data type of “Survived” into a categorical data**



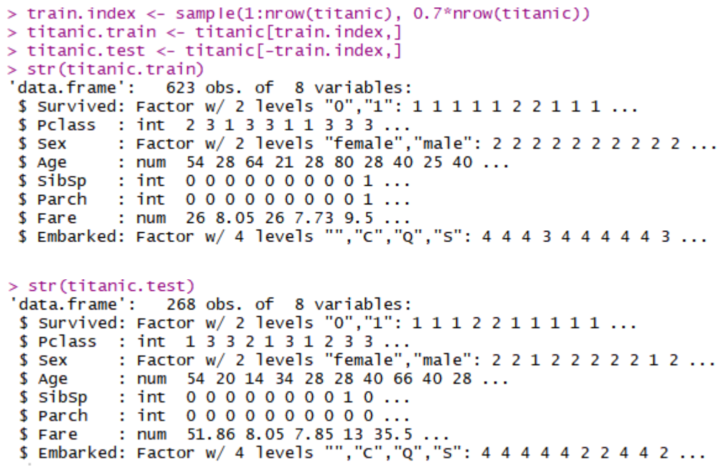
```
> titanic$Survived <- as.factor(titanic$Survived)
> str(titanic$Survived)
 Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
```

4. **To find the missing values in Age, and replace those missing values with “median” value**



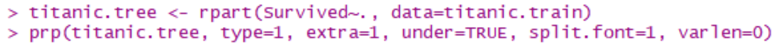
```
> age_median = median(titanic$Age, na.rm = TRUE)
> titanic$Age[is.na(titanic$Age)] <- age_median
> summary(titanic$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.42   22.00   28.00   29.36   35.00   80.00 
```

5. **To split the data into train and test sets with 70/30 rule**

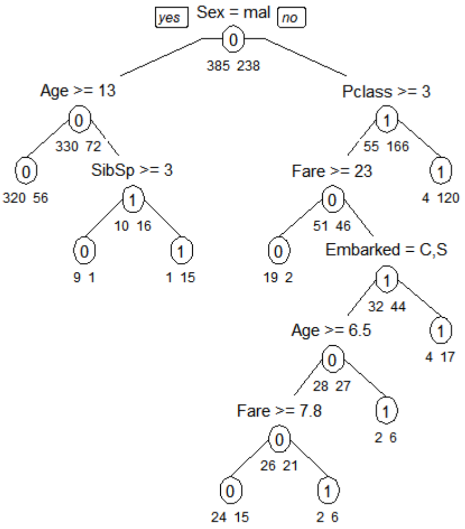


```
> train.index <- sample(1:nrow(titanic), 0.7*nrow(titanic))
> titanic.train <- titanic[train.index,]
> titanic.test <- titanic[-train.index,]
> str(titanic.train)
'data.frame':	623 obs. of  8 variables:
 $ Survived: Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 1 1 ...
 $ Pclass  : int  2 3 1 3 3 1 1 3 3 3 ...
 $ Sex     : Factor w/ 2 levels "female","male": 2 2 2 2 2 2 2 2 2 2 ...
 $ Age     : num  54 28 64 21 28 80 28 40 25 40 ...
 $ SibSp   : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Parch   : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Fare    : num  26 8.05 26 7.73 9.5 ...
 $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 4 4 3 4 4 4 4 4 3 ...

> str(titanic.test)
'data.frame':	268 obs. of  8 variables:
 $ Survived: Factor w/ 2 levels "0","1": 1 1 1 2 2 1 1 1 1 1 ...
 $ Pclass  : int  1 3 3 2 1 3 1 2 3 3 ...
 $ Sex     : Factor w/ 2 levels "female","male": 2 2 1 2 2 2 2 2 1 2 ...
 $ Age     : num  54 20 14 34 28 28 40 66 40 28 ...
 $ SibSp   : int  0 0 0 0 0 0 0 0 1 0 ...
 $ Parch   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Fare    : num  51.86 8.05 7.85 13 35.5 ...
 $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 4 4 4 4 2 2 4 4 2 ...
```

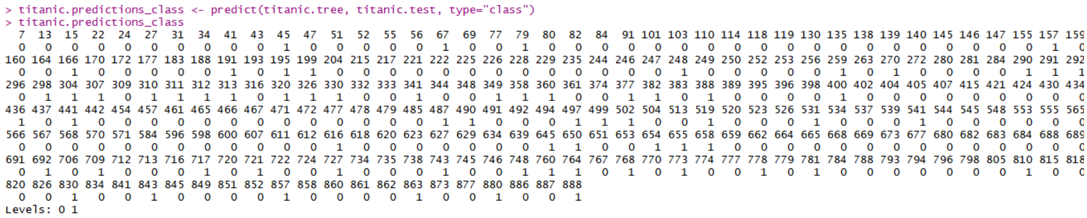
6. **To make a decision tree by using a train set and display the tree information**



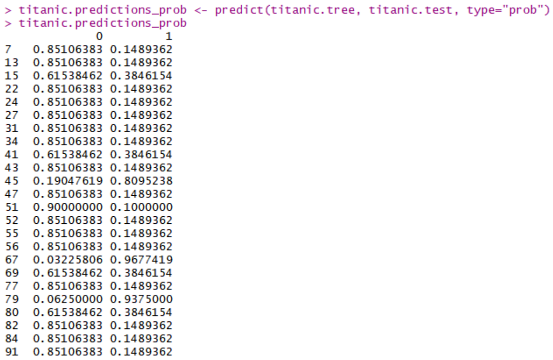
```
> titanic.tree <- rpart(Survived~., data=titanic.train)
> prp(titanic.tree, type=1, extra=1, under=TRUE, split.font=1, varlen=0)
```



7. To make a separate prediction with “class” and “prob” types, respectively

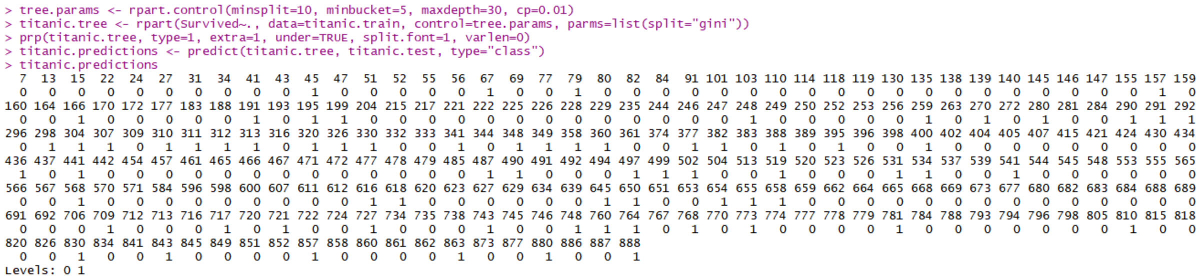


```
> titanic.predictions_class <- predict(titanic.tree, titanic.test, type="class")
> titanic.predictions_class
  7  13  15  22  24  27  31  34  41  43  45  47  51  52  55  56  67  69  77  79  80  82  84  91 101 103 110 114 118 119 130 135 138 139 140 145 146 147 155 157 159
  0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
160 164 166 170 172 177 183 188 191 193 195 199 204 215 217 221 222 225 226 228 229 235 244 246 247 248 249 250 252 253 256 259 263 270 272 280 281 284 290 291 292
  0   0   1   0   0   0   0   0   1   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   1   0   0   0   0   1   1   1
296 298 304 307 309 310 311 312 313 316 320 326 330 332 333 341 344 348 349 358 360 361 374 377 382 383 388 389 395 396 398 400 402 404 405 407 415 421 424 430 434
  0   1   1   1   0   1   1   1   1   0   1   1   1   0   0   1   0   0   1   1   1   0   0   1   1   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0
436 437 441 442 454 457 461 465 466 467 471 472 477 478 479 485 487 490 491 492 494 497 499 502 504 513 519 520 523 526 531 534 537 539 541 544 545 548 553 555 565
  1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0   1   1   1   0   0   1   0   0   0   1   0   0   0   1   0   0   0   0   0   0
566 567 568 570 571 584 596 598 600 607 611 612 616 618 620 623 627 629 634 639 645 650 651 653 654 655 658 659 662 664 665 668 669 673 677 680 682 683 684 688 689
  0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   1   1   0   0   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
691 692 706 709 712 713 716 717 720 721 722 724 727 734 735 738 743 745 746 748 760 764 767 768 770 773 774 777 778 779 781 784 788 793 794 796 798 805 810 815 818
  0   1   0   1   0   0   0   1   0   1   0   0   1   0   0   0   1   0   0   1   1   1   0   1   0   1   0   0   1   0   1   0   0   0   0   0   0   0   1   0   0
820 826 830 834 841 843 845 849 851 852 857 858 860 861 862 863 873 877 880 886 887 888
  0   0   1   0   0   1   0   0   0   0   1   0   0   0   0   1   0   0   1   0   0   1
Levels: 0 1
```



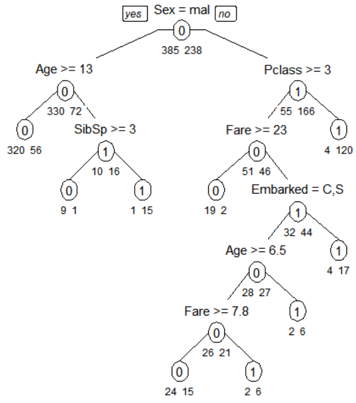
```
> titanic.predictions_prob <- predict(titanic.tree, titanic.test, type="prob")
> titanic.predictions_prob
             0         1
7   0.85106383 0.1489362
13  0.85106383 0.1489362
15  0.61538462 0.3846154
22  0.85106383 0.1489362
24  0.85106383 0.1489362
27  0.85106383 0.1489362
31  0.85106383 0.1489362
34  0.85106383 0.1489362
41  0.61538462 0.3846154
43  0.85106383 0.1489362
45  0.19047619 0.8095238
47  0.85106383 0.1489362
51  0.90000000 0.1000000
52  0.85106383 0.1489362
55  0.85106383 0.1489362
56  0.85106383 0.1489362
67  0.03225806 0.9677419
69  0.61538462 0.3846154
77  0.85106383 0.1489362
79  0.06250000 0.9375000
80  0.61538462 0.3846154
82  0.85106383 0.1489362
84  0.85106383 0.1489362
91  0.85106383 0.1489362
```

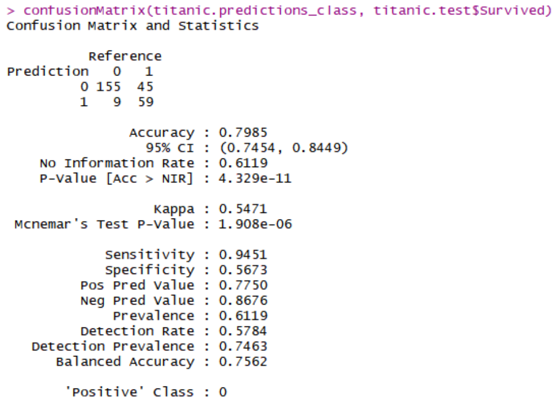
8. To use some controls in the decision tree, such as minimum split and depth, and make a new decision tree and a prediction again.



```
> tree.params <- rpart.control(minsplit=10, minbucket=5, maxdepth=30, cp=0.01)
> titanic.tree <- rpart(Survived~., data=titanic.train, control=tree.params, parms=list(split="gini"))
> prp(titanic.tree, type=1, extra=1, under=TRUE, split.font=1, varlen=0)
> titanic.predictions <- predict(titanic.tree, titanic.test, type="class")
> titanic.predictions
  7  13  15  22  24  27  31  34  41  43  45  47  51  52  55  56  67  69  77  79  80  82  84  91 101 103 110 114 118 119 130 135 138 139 140 145 146 147 155 157 159
  0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
160 164 166 170 172 177 183 188 191 193 195 199 204 215 217 221 222 225 226 228 229 235 244 246 247 248 249 250 252 253 256 259 263 270 272 280 281 284 290 291 292
  0   0   1   0   0   0   0   0   1   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   1   0   1   0   0   1   1   1
296 298 304 307 309 310 311 312 313 316 320 326 330 332 333 341 344 348 349 358 360 361 374 377 382 383 388 389 395 396 398 400 402 404 405 407 415 421 424 430 434
  0   1   1   1   0   1   1   1   1   0   1   1   1   0   0   1   0   1   1   1   1   0   0   1   1   0   1   0   1   0   0   1   0   0   0   0   0   0   0   0   0
436 437 441 442 454 457 461 465 466 467 471 472 477 478 479 485 487 490 491 492 494 497 499 502 504 513 519 520 523 526 531 534 537 539 541 544 545 548 553 555 565
  1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0   1   1   1   0   0   1   0   0   0   1   1   0   0   1   0   0   0   0   0   0
566 567 568 570 571 584 596 598 600 607 611 612 616 618 620 623 627 629 634 639 645 650 651 653 654 655 658 659 662 664 665 668 669 673 677 680 682 683 684 688 689
  0   0   1   0   0   0   0   0   0   0   0   0   1   1   0   0   0   0   0   0   1   1   0   0   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
691 692 706 709 712 713 716 717 720 721 722 724 727 734 735 738 743 745 746 748 760 764 767 768 770 773 774 777 778 779 781 784 788 793 794 796 798 805 810 815 818
  0   0   0   1   0   0   0   1   0   1   0   0   1   0   0   0   1   0   0   1   1   1   0   1   0   1   0   0   0   0   1   0   0   0   0   0   0   0   1   0   0
820 826 830 834 841 843 845 849 851 852 857 858 860 861 862 863 873 877 880 886 887 888
  0   0   1   0   0   1   0   0   0   0   1   0   0   0   0   1   0   0   1   0   0   1
Levels: 0 1
```

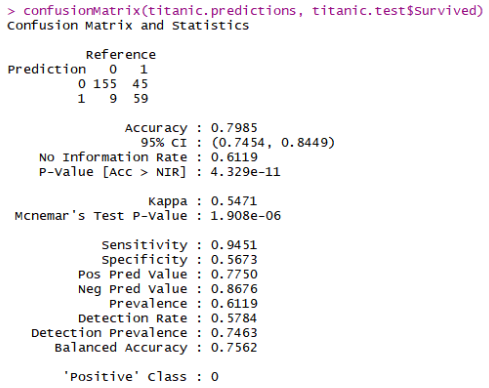
1) Not control parameter



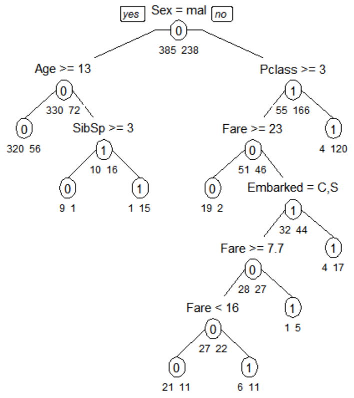


```
> confusionMatrix(titanic.predictions_class, titanic.test$Survived)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 155  45
         1   9  59
                                          
               Accuracy : 0.7985          
                 95% CI : (0.7454, 0.8449)
    No Information Rate : 0.6119          
    P-Value [Acc > NIR] : 4.329e-11       
                                          
                  Kappa : 0.5471          
 Mcnemar's Test P-Value : 1.908e-06       
                                          
            Sensitivity : 0.9451          
            Specificity : 0.5673          
         Pos Pred Value : 0.7750          
         Neg Pred Value : 0.8676          
             Prevalence : 0.6119          
         Detection Rate : 0.5784          
   Detection Prevalence : 0.7463          
      Balanced Accuracy : 0.7562          
                                          
       'Positive' Class : 0               
```

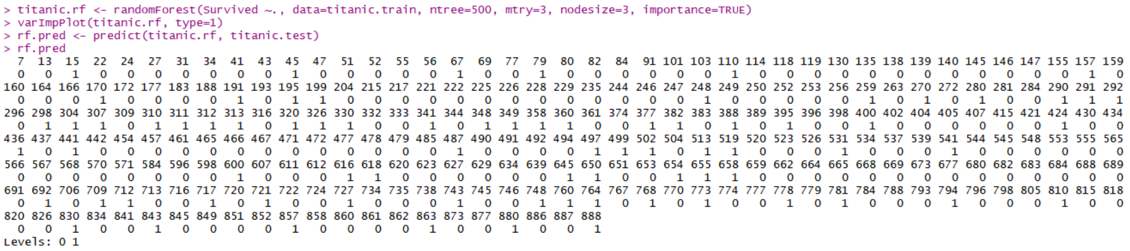
2) Control parameter



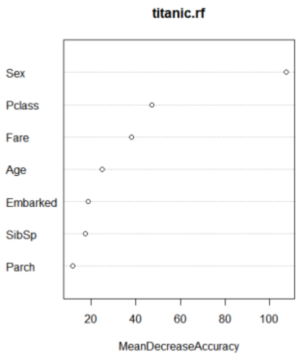
```
> confusionMatrix(titanic.predictions, titanic.test$Survived)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 155  45
         1   9  59
                                          
               Accuracy : 0.7985          
                 95% CI : (0.7454, 0.8449)
    No Information Rate : 0.6119          
    P-Value [Acc > NIR] : 4.329e-11       
                                          
                  Kappa : 0.5471          
 Mcnemar's Test P-Value : 1.908e-06       
                                          
            Sensitivity : 0.9451          
            Specificity : 0.5673          
         Pos Pred Value : 0.7750          
         Neg Pred Value : 0.8676          
             Prevalence : 0.6119          
         Detection Rate : 0.5784          
   Detection Prevalence : 0.7463          
      Balanced Accuracy : 0.7562          
                                          
       'Positive' Class : 0               
```



9. To apply a random forest with number of tree = 500 and mtry = 3

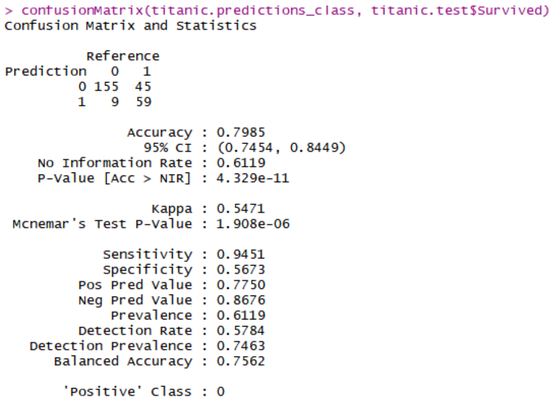


```
> titanic.rf <- randomForest(Survived ~., data=titanic.train, ntree=500, mtry=3, nodesize=3, importance=TRUE)
> varImpPlot(titanic.rf, type=1)
> rf.pred <- predict(titanic.rf, titanic.test)
> rf.pred
  7  13  15  22  24  27  31  34  41  43  45  47  51  52  55  56  67  69  77  79  80  82  84  91 101 103 110 114 118 119 130 135 138 139 140 145 146 147 155 157 159 
  0   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   1   0 
160 164 166 170 172 177 183 188 191 193 195 199 204 215 217 221 222 225 226 228 229 235 244 246 247 248 249 250 252 253 256 259 263 270 272 280 281 284 290 291 292 
  0   0   0   1   0   0   0   0   1   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   1   0   1   0   0   1   1   1 
296 298 304 307 309 310 311 312 313 316 320 326 330 332 333 341 344 348 349 358 360 361 374 377 382 383 388 389 395 396 398 400 402 404 405 407 415 421 424 430 434 
  0   1   1   1   0   1   1   1   1   0   1   1   1   0   0   1   0   1   1   1   1   0   0   1   1   0   1   0   1   0   0   1   0   0   0   0   0   0   1   0   0 
436 437 441 442 454 457 461 465 466 467 471 472 477 478 479 485 487 490 491 492 494 497 499 502 504 513 519 520 523 526 531 534 537 539 541 544 545 548 553 555 565 
  1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   1   1   1   0   1   1   0   0   0   1   0   0   0   1   0   0   0   0   0   0 
566 567 568 570 571 584 596 598 600 607 611 612 616 618 620 623 627 629 634 639 645 650 651 653 654 655 658 659 662 664 665 668 669 673 677 680 682 683 684 688 689 
  0   0   0   0   0   0   0   0   1   0   0   0   1   1   0   0   0   0   0   0   1   1   0   0   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0 
691 692 706 709 712 713 716 717 720 721 722 724 727 734 735 738 743 745 746 748 760 764 767 768 770 773 774 777 778 779 781 784 788 793 794 796 798 805 810 815 818 
  0   1   0   1   1   0   0   1   0   1   0   0   1   0   0   0   1   0   0   1   1   1   0   1   0   1   0   0   1   0   1   0   0   0   1   0   0   0   1   0   0 
820 826 830 834 841 843 845 849 851 852 857 858 860 861 862 863 873 877 880 886 887 888 
  0   0   1   0   0   1   0   0   0   0   1   0   0   0   0   1   0   0   1   0   0   1 
Levels: 0 1
```

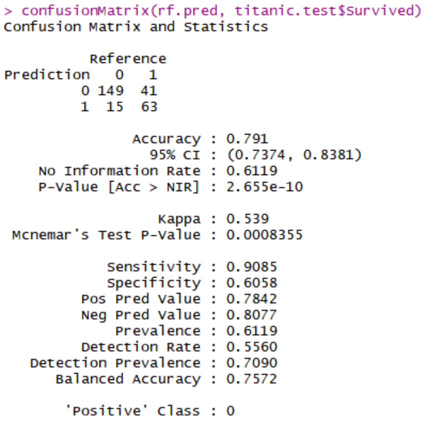


1) Decision tree (not control parameter)

2) Random forest



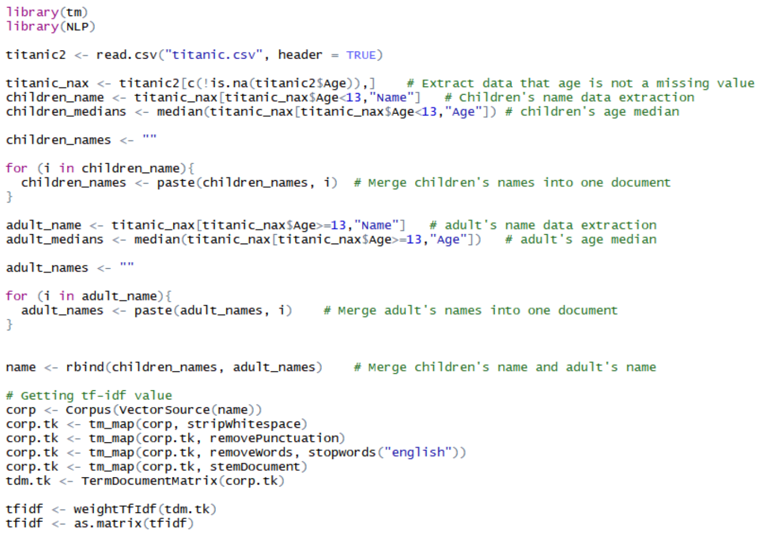
```
> confusionMatrix(titanic.predictions_class, titanic.test$Survived)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 155  45
         1   9  59

               Accuracy : 0.7985
                 95% CI : (0.7454, 0.8449)
    No Information Rate : 0.6119
    P-Value [Acc > NIR] : 4.329e-11

                  Kappa : 0.5471
 Mcnemar's Test P-Value : 1.908e-06

            Sensitivity : 0.9451
            Specificity : 0.5673
         Pos Pred Value : 0.7750
         Neg Pred Value : 0.8676
             Prevalence : 0.6119
         Detection Rate : 0.5784
   Detection Prevalence : 0.7463
      Balanced Accuracy : 0.7562

       'Positive' Class : 0
```



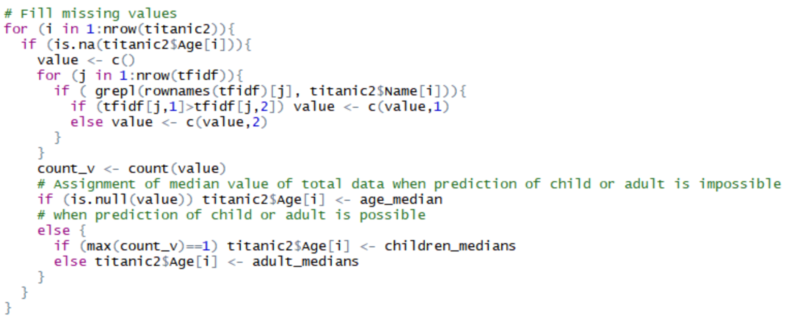
```
> confusionMatrix(rf.pred, titanic.test$Survived)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 149  41
         1  15  63

               Accuracy : 0.791
                 95% CI : (0.7374, 0.8381)
    No Information Rate : 0.6119
    P-Value [Acc > NIR] : 2.655e-10

                  Kappa : 0.539
 Mcnemar's Test P-Value : 0.0008355

            Sensitivity : 0.9085
            Specificity : 0.6058
         Pos Pred Value : 0.7842
         Neg Pred Value : 0.8077
             Prevalence : 0.6119
         Detection Rate : 0.5560
   Detection Prevalence : 0.7090
      Balanced Accuracy : 0.7572

       'Positive' Class : 0
```

10. EXTRA: to replace the missing value with median, it would be better to ignore the children (particularly, boys) before the replacement. How do you manage this issue? (hint: boys' names contain "Master.")

Children and adults were categorized to fill the missing values of age. And the tf-idf value was calculated to find the name often mentioned in the names of children and adults. Then, the name of the person with the missing value is compared and the value of age is filled in.

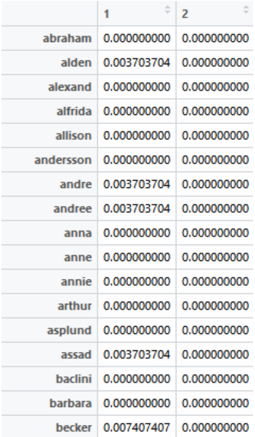


```
library(tm)
library(NLP)

titanic2 <- read.csv("titanic.csv", header = TRUE)

titanic_nax <- titanic2[c(!is.na(titanic2$Age)),]    # Extract data that age is not a missing value
children_name <- titanic_nax[titanic_nax$Age<13,"Name"]   # Children's name data extraction
children_medians <- median(titanic_nax[titanic_nax$Age<13,"Age"]) # children's age median

children_names <- ""

for (i in children_name){
  children_names <- paste(children_names, i)  # Merge children's names into one document
}

adult_name <- titanic_nax[titanic_nax$Age>=13,"Name"]   # adult's name data extraction
adult_medians <- median(titanic_nax[titanic_nax$Age>=13,"Age"])   # adult's age median

adult_names <- ""

for (i in adult_name){
  adult_names <- paste(adult_names, i)    # Merge adult's names into one document
}


name <- rbind(children_names, adult_names)    # Merge children's name and adult's name

# Getting tf-idf value
corp <- Corpus(VectorSource(name))
corp.tk <- tm_map(corp, stripWhitespace)
corp.tk <- tm_map(corp.tk, removePunctuation)
corp.tk <- tm_map(corp.tk, removeWords, stopwords("english"))
corp.tk <- tm_map(corp.tk, stemDocument)
tdm.tk <- TermDocumentMatrix(corp.tk)

tfidf <- weightTfIdf(tdm.tk)
tfidf <- as.matrix(tfidf)
```



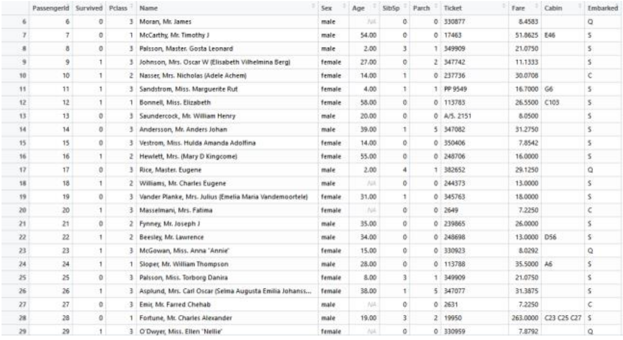
```
# Fill missing values
for (i in 1:nrow(titanic2)){
  if (is.na(titanic2$Age[i])){
    value <- c()
    for (j in 1:nrow(tfidf)){
      if ( grepl(rownames(tfidf)[j], titanic2$Name[i])){
        if (tfidf[j,1]>tfidf[j,2]) value <- c(value,1)
        else value <- c(value,2)
      }
    }
    count_v <- count(value)
    # Assignment of median value of total data when prediction of child or adult is impossible
    if (is.null(value)) titanic2$Age[i] <- age_median
    # when prediction of child or adult is possible
    else {
      if (max(count_v)==1) titanic2$Age[i] <- children_medians
      else titanic2$Age[i] <- adult_medians
    }
  }
}
```

✓ Result of TF-IDF

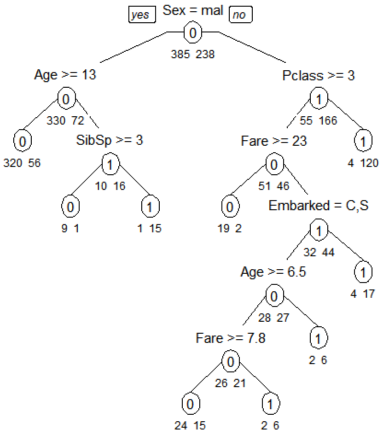


	1	2
abraham	0.000000000	0.000000000
alden	0.003703704	0.000000000
alexand	0.000000000	0.000000000
alfrida	0.000000000	0.000000000
allison	0.000000000	0.000000000
andersson	0.000000000	0.000000000
andre	0.003703704	0.000000000
andree	0.003703704	0.000000000
anna	0.000000000	0.000000000
anne	0.000000000	0.000000000
annie	0.000000000	0.000000000
arthur	0.000000000	0.000000000
asplund	0.000000000	0.000000000
assad	0.003703704	0.000000000
baclini	0.000000000	0.000000000
barbara	0.000000000	0.000000000
becker	0.007407407	0.000000000

1) Before

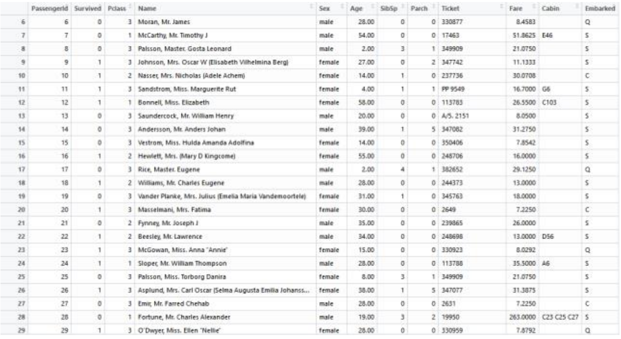


	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
6	6	0	3	Moran, Mr. James	male	NA	0	0	330877	8.4583		Q
7	7	0	1	McCarthy, Mr. Timothy J	male	54.00	0	0	17463	51.8625	E46	S
8	8	0	3	Palsson, Master. Gosta Leonard	male	2.00	3	1	349909	21.0750		S
9	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.00	0	2	347742	11.1333		S
10	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.00	1	0	237736	30.0708		C
11	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.00	1	1	PP 9549	16.7000	G6	S
12	12	1	1	Bonnell, Miss. Elizabeth	female	58.00	0	0	113783	26.5500	C103	S
13	13	0	3	Saundercock, Mr. William Henry	male	20.00	0	0	A/5. 2151	8.0500		S
14	14	0	3	Andersson, Mr. Anders Johan	male	39.00	1	5	347082	31.2750		S
15	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.00	0	0	350406	7.8542		S
16	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.00	0	0	248706	16.0000		S
17	17	0	3	Rice, Master. Eugene	male	2.00	4	1	382652	29.1250		Q
18	18	1	2	Williams, Mr. Charles Eugene	male	NA	0	0	244373	13.0000		S
19	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31.00	1	0	345763	18.0000		S
20	20	1	3	Masselmani, Mrs. Fatima	female	NA	0	0	2649	7.2250		C
21	21	0	2	Fynney, Mr. Joseph J	male	35.00	0	0	239865	26.0000		S
22	22	1	2	Beesley, Mr. Lawrence	male	34.00	0	0	248698	13.0000	D56	S
23	23	1	3	McGowan, Miss. Anna "Annie"	female	15.00	0	0	330923	8.0292		Q
24	24	1	1	Sloper, Mr. William Thompson	male	28.00	0	0	113788	35.5000	A6	S
25	25	0	3	Palsson, Miss. Torborg Danira	female	8.00	3	1	349909	21.0750		S
26	26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johanss...	female	38.00	1	5	347077	31.3875		S
27	27	0	3	Emir, Mr. Farred Chehab	male	NA	0	0	2631	7.2250		C
28	28	0	1	Fortune, Mr. Charles Alexander	male	19.00	3	2	19950	263.0000	C23 C25 C27	S
29	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NA	0	0	330959	7.8792		Q

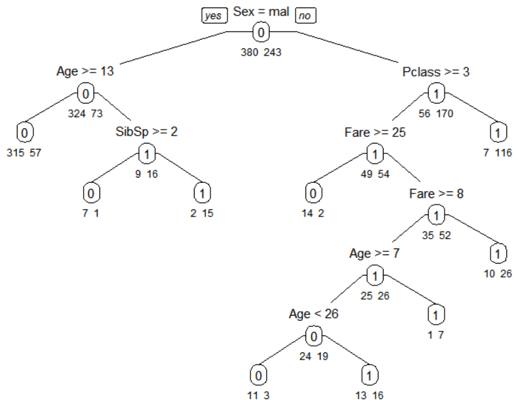


```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 155  45
         1   9  59
                                          
               Accuracy : 0.7985          
                 95% CI : (0.7454, 0.8449)
    No Information Rate : 0.6119          
    P-Value [Acc > NIR] : 4.329e-11       
                                          
                  Kappa : 0.5471          
 Mcnemar's Test P-Value : 1.908e-06       
                                          
            Sensitivity : 0.9451          
            Specificity : 0.5673          
         Pos Pred Value : 0.7750          
         Neg Pred Value : 0.8676          
             Prevalence : 0.6119          
         Detection Rate : 0.5784          
   Detection Prevalence : 0.7463          
      Balanced Accuracy : 0.7562          
                                          
       'Positive' Class : 0               
```

2) After



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
6	6	0	3	Moran, Mr. James	male	28.00	0	0	330877	8.4583		Q
7	7	0	1	McCarthy, Mr. Timothy J	male	54.00	0	0	17463	51.8625	E46	S
8	8	0	3	Palsson, Master. Gosta Leonard	male	2.00	3	1	349909	21.0750		S
9	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.00	0	2	347742	11.1333		S
10	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.00	1	0	237736	30.0708		C
11	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.00	1	1	PP 9549	16.7000	G6	S
12	12	1	1	Bonnell, Miss. Elizabeth	female	58.00	0	0	113783	26.5500	C103	S
13	13	0	3	Saundercock, Mr. William Henry	male	20.00	0	0	A/5. 2151	8.0500		S
14	14	0	3	Andersson, Mr. Anders Johan	male	39.00	1	5	347082	31.2750		S
15	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.00	0	0	350406	7.8542		S
16	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.00	0	0	248706	16.0000		S
17	17	0	3	Rice, Master. Eugene	male	2.00	4	1	382652	29.1250		Q
18	18	1	2	Williams, Mr. Charles Eugene	male	28.00	0	0	244373	13.0000		S
19	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31.00	1	0	345763	18.0000		S
20	20	1	3	Masselmani, Mrs. Fatima	female	30.00	0	0	2649	7.2250		C
21	21	0	2	Fynney, Mr. Joseph J	male	35.00	0	0	239865	26.0000		S
22	22	1	2	Beesley, Mr. Lawrence	male	34.00	0	0	248698	13.0000	D56	S
23	23	1	3	McGowan, Miss. Anna "Annie"	female	15.00	0	0	330923	8.0292		Q
24	24	1	1	Sloper, Mr. William Thompson	male	28.00	0	0	113788	35.5000	A6	S
25	25	0	3	Palsson, Miss. Torborg Danira	female	8.00	3	1	349909	21.0750		S
26	26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johanss...	female	38.00	1	5	347077	31.3875		S
27	27	0	3	Emir, Mr. Farred Chehab	male	28.00	0	0	2631	7.2250		C
28	28	0	1	Fortune, Mr. Charles Alexander	male	19.00	3	2	19950	263.0000	C23 C25 C27	S
29	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	28.00	0	0	330959	7.8792		Q



```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 154  36
         1  15  63
                                          
               Accuracy : 0.8097          
                 95% CI : (0.7575, 0.8549)
    No Information Rate : 0.6306          
    P-Value [Acc > NIR] : 1.381e-10       
                                          
                  Kappa : 0.5728          
 Mcnemar's Test P-Value : 0.005101        
                                          
            Sensitivity : 0.9112          
            Specificity : 0.6364          
         Pos Pred Value : 0.8105          
         Neg Pred Value : 0.8077          
             Prevalence : 0.6306          
         Detection Rate : 0.5746          
   Detection Prevalence : 0.7090          
      Balanced Accuracy : 0.7738          
                                          
       'Positive' Class : 0               
```