**2015/2016**

# DATA MINING PROJECT

## “Titanic Dataset”

Prof. Dino Pedreschi

Prof. Anna Monreale

PhD. Riccardo Guidotti

Simay Koçan - 525804

Hüseyin Varol Erdem – 525822

# 1. Data Understanding: Titanic Disaster Data Set

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

Titanic data set consist of 12 attributes and 891 instances that were collected from passenger's information for analysis.

## 1.1 Data Semantics

**PassengerId:** PassengerId assigns to each passenger. It provides only enough information to distinguish one object from another. Therefore it is a nominal and also discrete attribute.

**Survived:** Survived determines whether the passengers survived (1), or not (0). It is a categorical (nominal) attribute and also binary attribute that is a special case of discrete attributes.

**Pclass:** It is a nominal and also discrete attribute. It demonstrates the class of the passenger which was in first class (1), second class (2), or third class (3).

**Name:** Name column shows the name of a passenger. It is a nominal attribute.

**Sex:** Sex is the gender of the passenger. It can be male or female. The type of that attribute is categorical.

**Age:** The age of the passenger is illustrated by Age column. Age's type is numerical which is the type of continuous attribute. If the value of age is lower than 1, it will be fractional and also if the age is estimated, the form of that is xx.5.

**SibSp:** SibSp column shows the number of siblings and spouses the passenger had on board. It is a numerical attribute.

**Parch:** It assigns the number of parents and children the passenger had on board. The type of that attribute is numerical.

**Ticket:** It is a nominal attribute. It gives the ticket number of the passenger.

**Fare:** Fare column shows that how much the passenger paid for the ticket. It is a numerical and also continuous attribute.

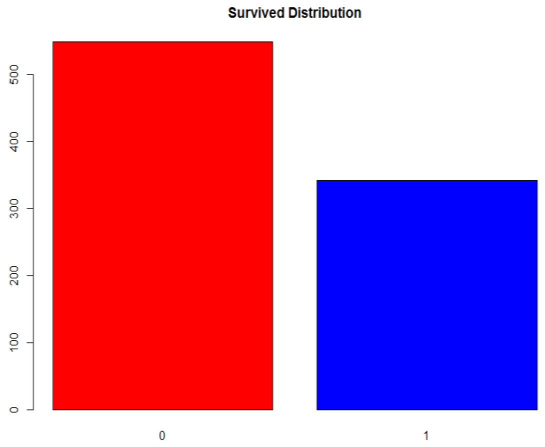
**Cabin:** It tells which cabin the passenger was in. The type of Cabin column is categorical.

**Embarked:** Embarked gives an information about where the passenger boarded the Titanic. It is a categorical attribute and the values of Embarked are C (Cherbourg), Q (Queenstown), and S (Southampton).

## 1.2 Distribution of the Variables and Statistics

### 1.2.1 Categorical Attributes Distribution

**Survived:** Survived attribute has no missing values.

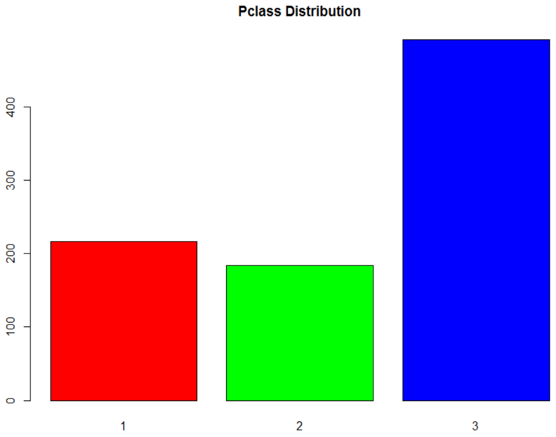


Survival rate from Titanic: 38%

Number of died passenger: 549

Number of survived passenger: 342

**Pclass:** Pclass attribute has no missing values.

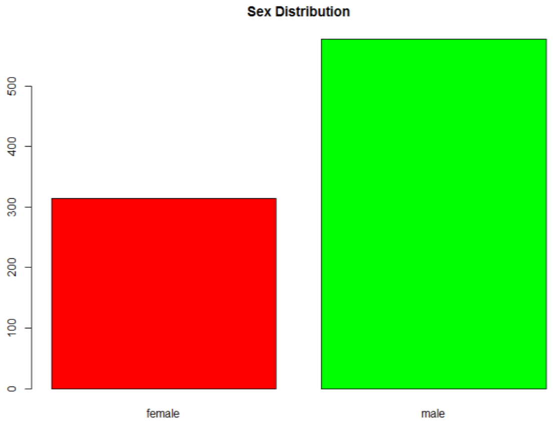


Number of First Class passenger: 216

Number of Second Class passenger: 184

Number of Third Class passenger: 491

**Sex:** Sex attribute has no missing values.

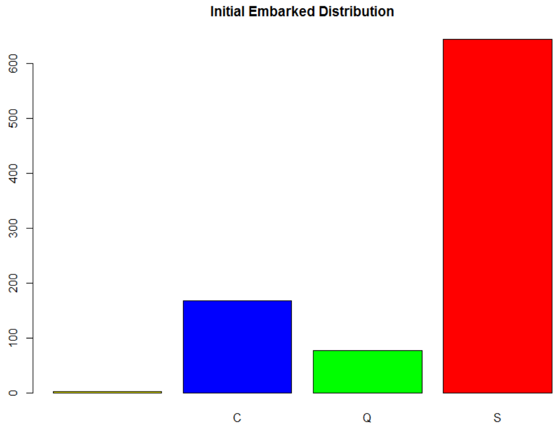


Female rate from Titanic: 35%

Number of female passenger: 314

Number of male passenger: 577

**Embarked:** Embarked attribute has only two missing values. Data distribution with two missing value.



Number of Cherbourg’s passenger: 168

Number of Queenstown’s passenger: 77

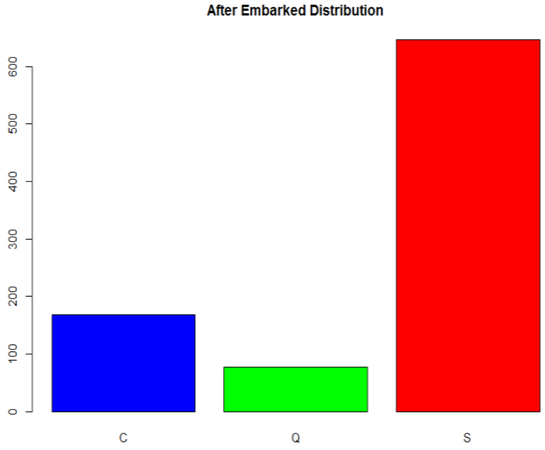
Number of Southampton’s passenger: 644

Number of missing embarked info: 2

### 1.2.2 Assessing Data Quality on Categorical Attributes

“PassengerId”, “Cabin”, “Ticket” and “Name” Attributes are discarded from Titanic data. They do not provide enough information for analysis. Most of the values in the “Cabin” column are missing (only 204 values out of 891 rows). The “Ticket” and “Name” columns are unlikely to tell much without some domain knowledge about what the ticket number mean, and about which names correlate with characteristics like large or rich families. Other categorical attributes “Survived”, “Pclass” and “Sex” have complete data. “Embarked” column has only two missing value.

**Filling Missing Embarked Records:** Missing records are filled with most frequent Embarked value which is “S” (Southampton). After this step, their data distribution seems like below:

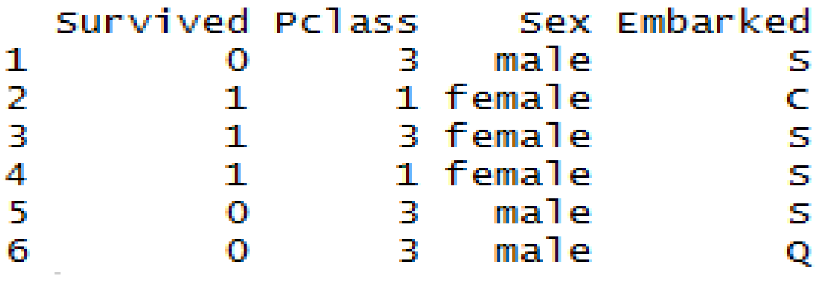


Number of Cherbourg’s passenger: 168

Number of Queenstown’s passenger: 77

Number of Southampton’s passenger: 646

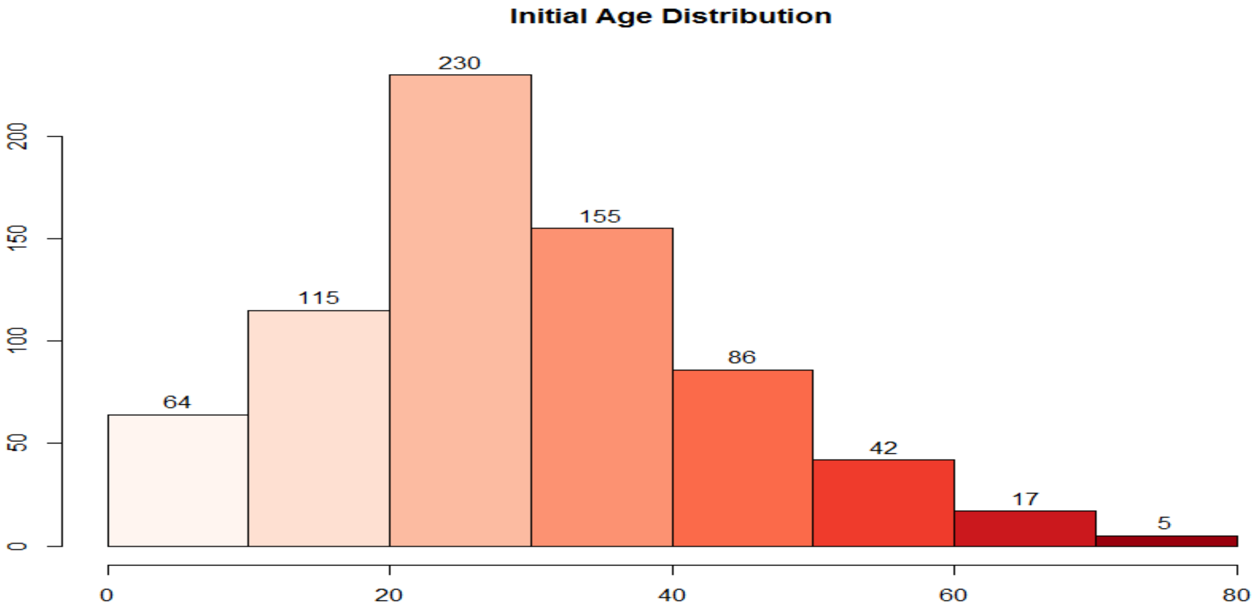
Finally, the first six row of the categorical attribute's view on Titanic Data Set:



```
  Survived Pclass    Sex Embarked
1        0      3   male        S
2        1      1 female        C
3        1      3 female        S
4        1      1 female        S
5        0      3   male        S
6        0      3   male        Q
```

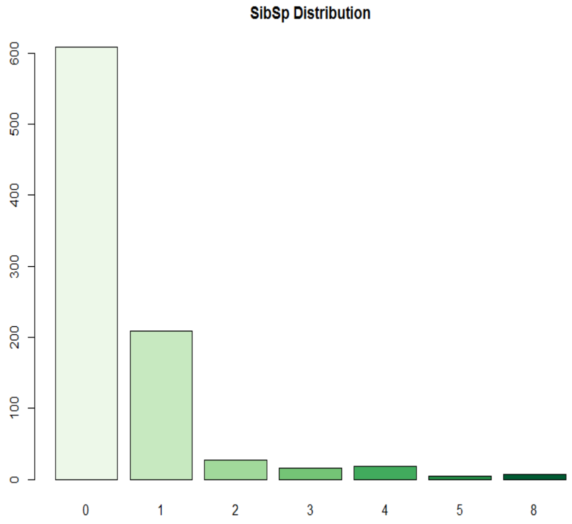
## 1.2.3 Numerical Attributes Distribution

**Age:** Age attribute has 177 missing values. It can be seen in General view of numerical attributes' statistics.



Values on bars show that the number of passenger's age is separated according to intervals.

**SibSp:** SibSp attribute has no missing value.



Number of passengers who have no sibling or spouse: 608

Number of passengers who have one sibling or spouse: 209

Number of passengers who have two siblings or spouses: 28

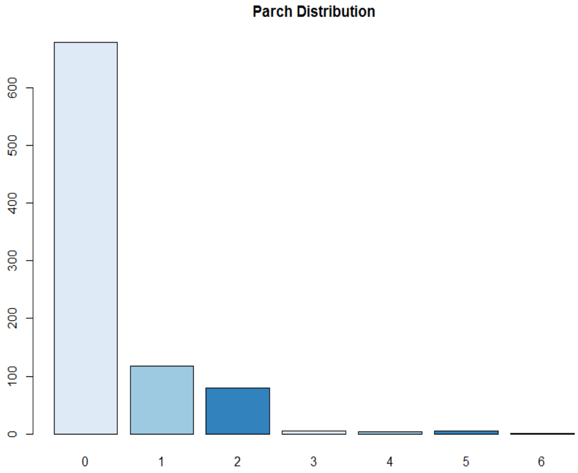
Number of passengers who have three siblings or spouse: 16

Number of passengers who have four siblings or spouse: 18

Number of passengers who have five siblings or spouse: 5

Number of passengers who have eight siblings or spouse: 7

**Parch:** Parch attribute has no missing value.



Number of passengers who have no parent or child:  678

Number of passengers who have one parent or child: 118

Number of passengers who have two parents or children: 80

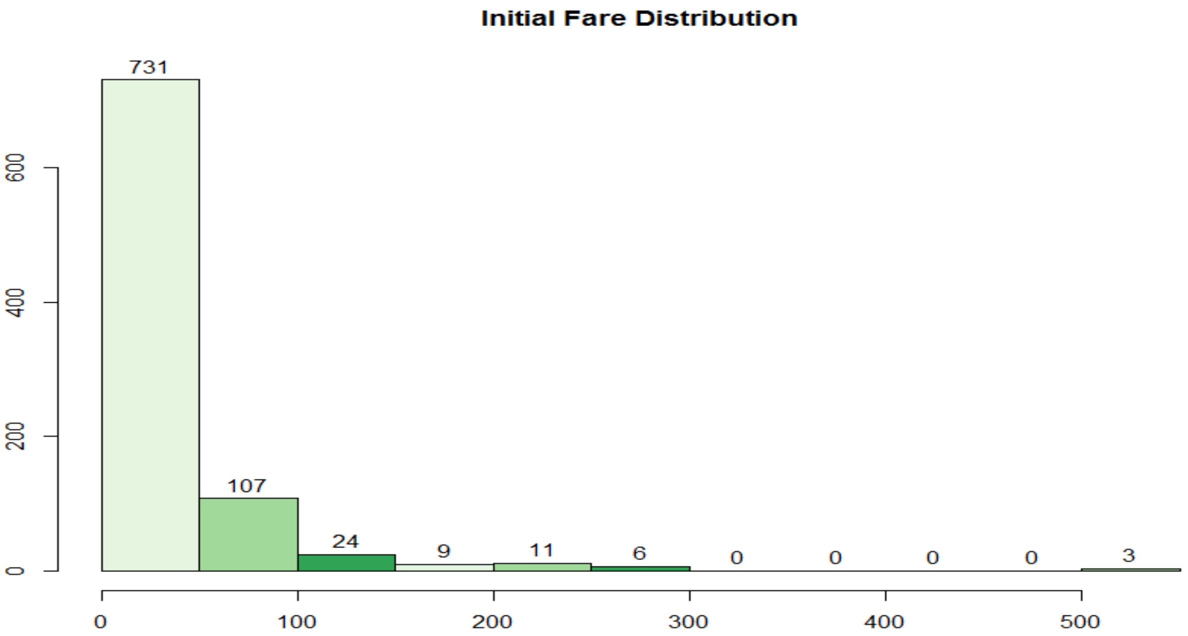
Number of passengers who have three parents or children: 5

Number of passengers who have four parents or children: 4

Number of passengers who have five parents or children: 5

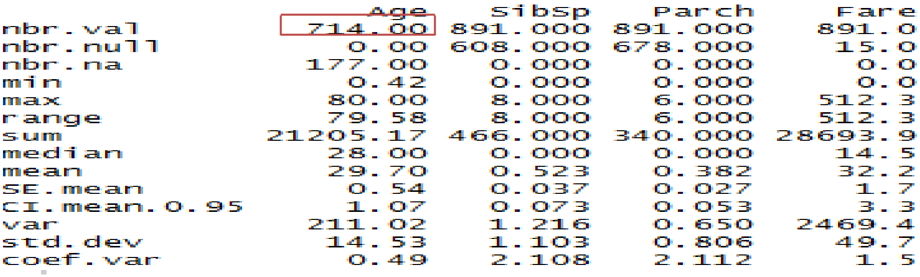
Number of passengers who have six parents or children: 1

**Fare:** Fare attribute has no missing value.



Values on bars show that the number of passengers whose ticket fares are separated according to intervals.

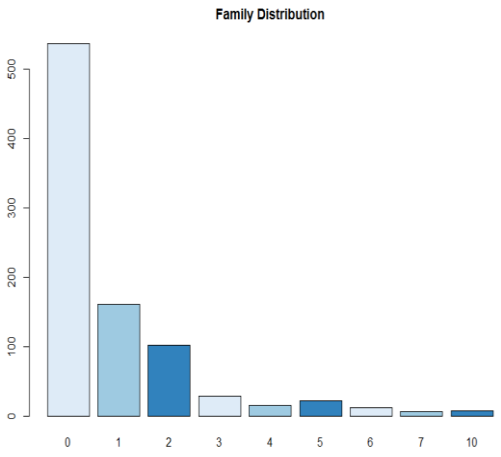
General view of numerical attribute’s statistics:



```
                  Age   Sibsp   Parch      Fare
nbr.val        714.00 891.000 891.000     891.0
nbr.null         0.00 608.000 678.000      15.0
nbr.na         177.00   0.000   0.000       0.0
min              0.42   0.000   0.000       0.0
max             80.00   8.000   6.000     512.3
range           79.58   8.000   6.000     512.3
sum          21205.17 466.000 340.000   28693.9
median          28.00   0.000   0.000      14.5
mean            29.70   0.523   0.382      32.2
SE.mean          0.54   0.037   0.027       1.7
CI.mean.0.95     1.07   0.073   0.053       3.3
var            211.02   1.216   0.650    2469.4
std.dev         14.53   1.103   0.806      49.7
coef.var         0.49   2.108   2.112       1.5
```

### 1.2.4 Assessing Data Quality on Numerical Attributes

**Merging "SibSp" and "Parch" Attributes:** "SibSp" and "Parch" columns have a similar meaning. Therefore, they are merged on the one column by creating a new which is "Family" column.



Number of passengers who have no family member: 537

Number of passengers who have one family member: 161

Number of passengers who have two family members: 102

Number of passengers who have three family members: 29

Number of passengers who have four family members: 15

Number of passengers who have five family members: 22

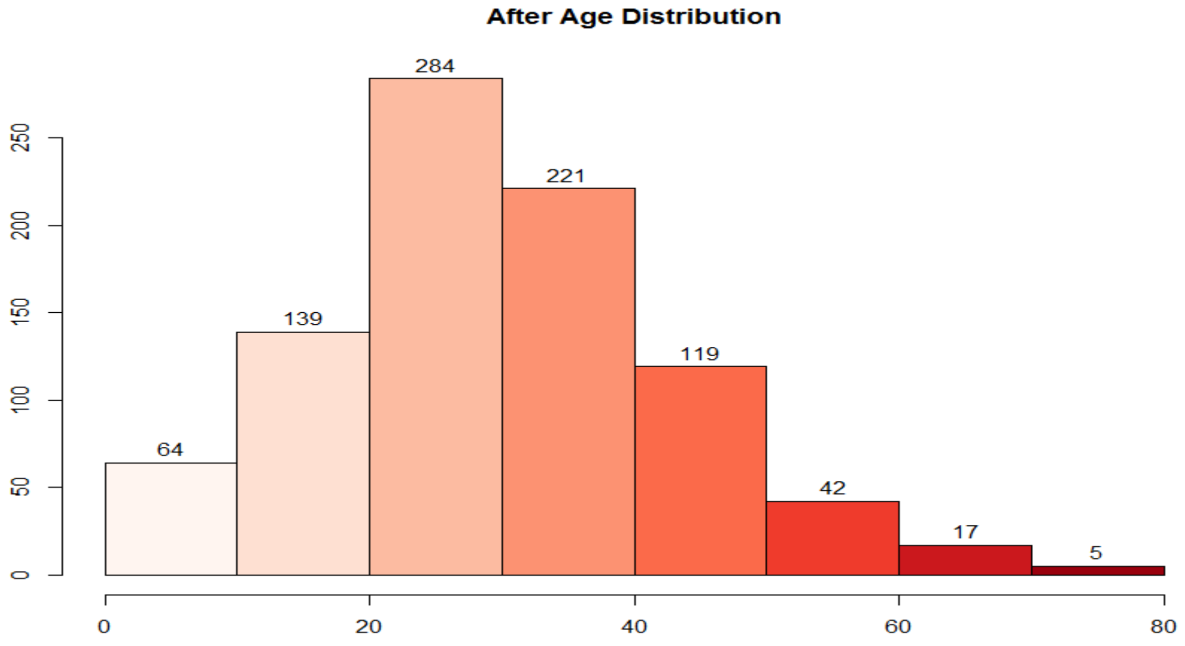
Number of passengers who have six family members: 12

Number of passengers who have seven family members: 6

Number of passengers who have ten family members: 7

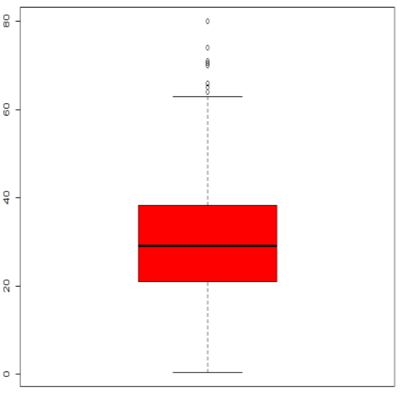
**Filling Missing Age Records:** When filling missing Age records, this technique is used for balanced distribution:

By taking Age's mean and standard deviation, generating random Age values between (Age's mean – Age's standard deviation) and (Age's mean + Age's standard deviation) and then they are assigned to missing Age's records. After calculating this, it's distribution:

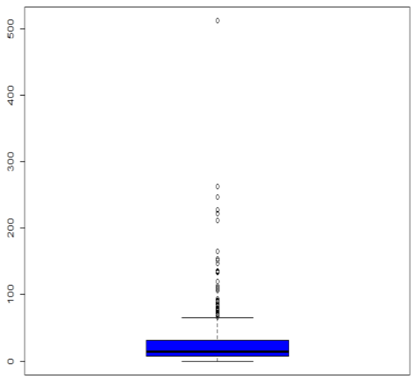


If the above distribution is compared with initial distribution of age, it can be seen that, missing age values are filled with balanced distribution.

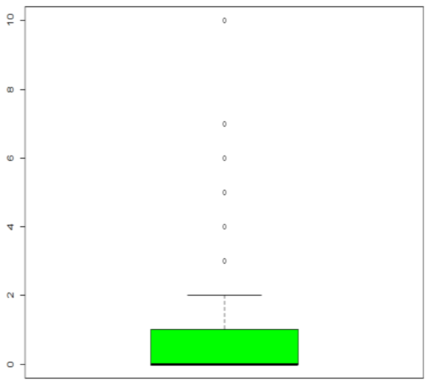
## Boxplots:



Age Boxplot

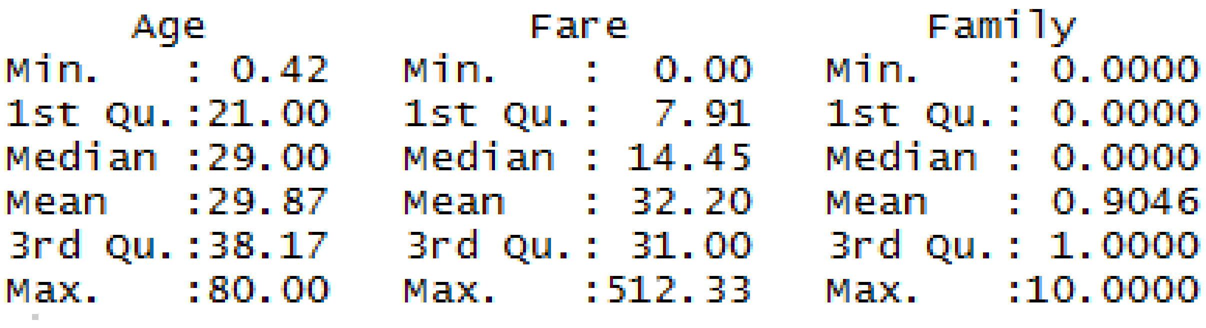


Fare Boxplot



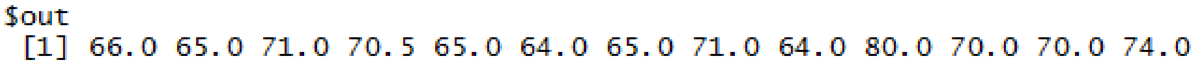
Family Boxplot

## Statistics:



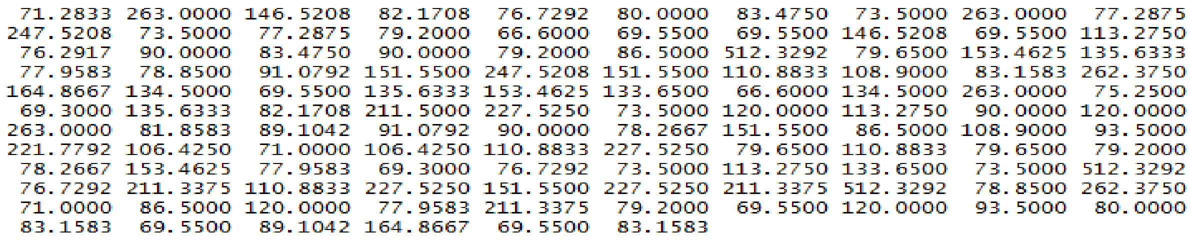
```
      Age              Fare             Family
Min.   : 0.42   Min.   :  0.00   Min.   : 0.0000
1st Qu.:21.00   1st Qu.:  7.91   1st Qu.: 0.0000
Median :29.00   Median : 14.45   Median : 0.0000
Mean   :29.87   Mean   : 32.20   Mean   : 0.9046
3rd Qu.:38.17   3rd Qu.: 31.00   3rd Qu.: 1.0000
Max.   :80.00   Max.   :512.33   Max.   :10.0000
```

## High Values for Age Attribute:



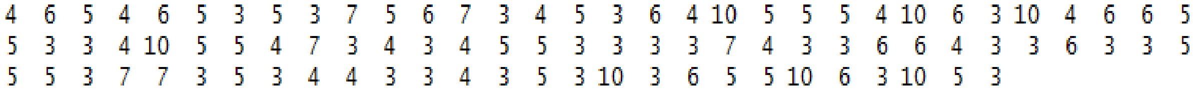
```
$out
 [1] 66.0 65.0 71.0 70.5 65.0 64.0 65.0 71.0 64.0 80.0 70.0 70.0 74.0
```

## High Values for Fare Attribute:



```
 71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750  73.5000 263.0000  77.2875
247.5208  73.5000  77.2875  79.2000  66.6000  69.5500  69.5500 146.5208  69.5500 113.2750
 76.2917  90.0000  83.4750  90.0000  79.2000  86.5000 512.3292  79.6500 153.4625 135.6333
 77.9583  78.8500  91.0792 151.5500 247.5208 151.5500 110.8833 108.9000  83.1583 262.3750
164.8667 134.5000  69.5500 135.6333 153.4625 133.6500  66.6000 134.5000 263.0000  75.2500
 69.3000 135.6333  82.1708 211.5000 227.5250  73.5000 120.0000 113.2750  90.0000 120.0000
263.0000  81.8583  89.1042  91.0792  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000
221.7792 106.4250  71.0000 106.4250 110.8833 227.5250  79.6500 110.8833  79.6500  79.2000
 78.2667 153.4625  77.9583  69.3000  76.7292  73.5000 113.2750 133.6500  73.5000 512.3292
 76.7292 211.3375 110.8833 227.5250 151.5500 227.5250 211.3375 512.3292  78.8500 262.3750
 71.0000  86.5000 120.0000  77.9583 211.3375  79.2000  69.5500 120.0000  93.5000  80.0000
 83.1583  69.5500  89.1042 164.8667  69.5500  83.1583
```

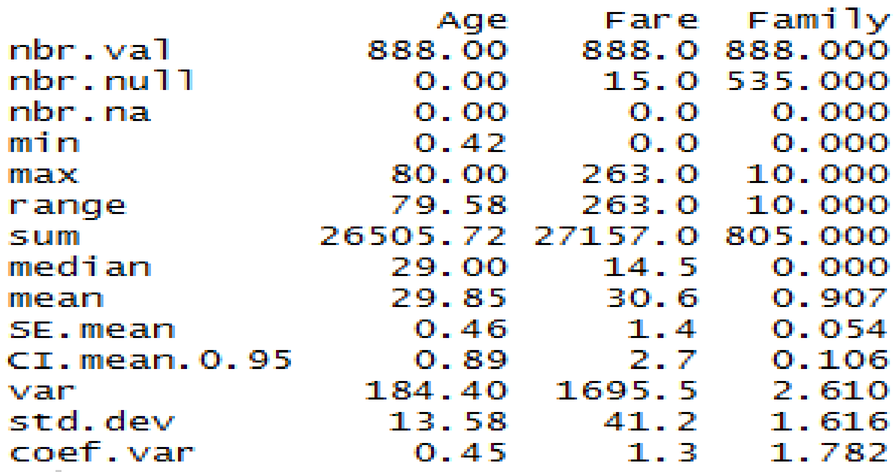
## High Values for Family Attribute:



```
 4  6  5  4  6  5  3  5  3  7  5  6  7  3  4  5  3  6  4 10  5  5  5  4 10  6  3 10  4  6  6  5
 5  3  3  4 10  5  5  4  7  3  4  3  4  5  5  3  3  3  3  7  4  3  3  6  6  4  3  3  6  3  3  5
 5  5  3  7  7  3  5  3  4  4  3  3  4  3  5  3 10  3  6  5  5 10  6  3 10  5  3
```

The higher values of Age and Family columns cannot be thought as outliers because; there is not any abnormal value for analysis. On the other hand, Fare column's top values (512.3292) are excessively isolated from their values. Therefore, they are decided to remove.

After assessing the data, statistics are:

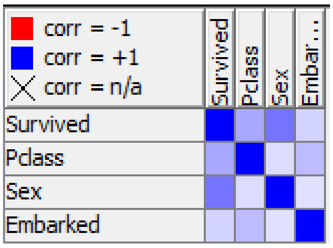


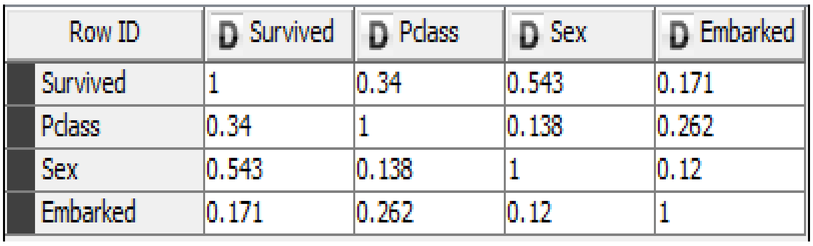
```
                      Age     Fare   Family
nbr.val            888.00    888.0  888.000
nbr.null             0.00     15.0  535.000
nbr.na               0.00      0.0    0.000
min                  0.42      0.0    0.000
max                 80.00    263.0   10.000
range               79.58    263.0   10.000
sum              26505.72  27157.0  805.000
median              29.00     14.5    0.000
mean                29.85     30.6    0.907
SE.mean              0.46      1.4    0.054
CI.mean.0.95         0.89      2.7    0.106
var                184.40   1695.5    2.610
std.dev             13.58     41.2    1.616
coef.var             0.45      1.3    1.782
```

## 1.3 Pairwise Correlations

For a more accurate analysis of data, it is looked for a correlation between pairs of attributes. The correlation can reduce the size of dataset as highlights presence of redundant attributes.

By using the tool "Linear Correlation" of KNIME and function "cor()", "cor.test()" of R, two different correlation matrices are obtained by separating attributes which are numeric and non-numeric.

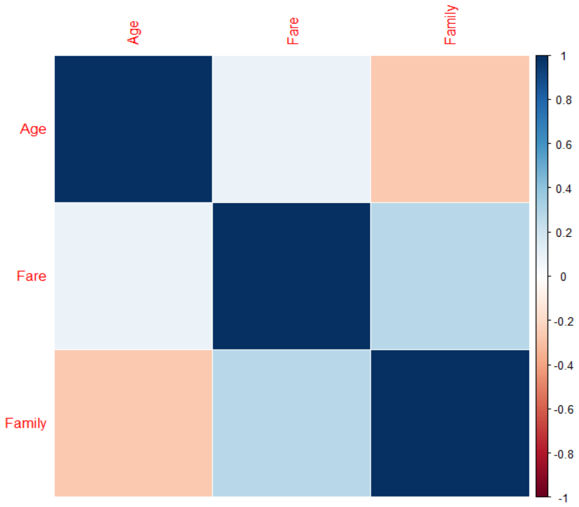


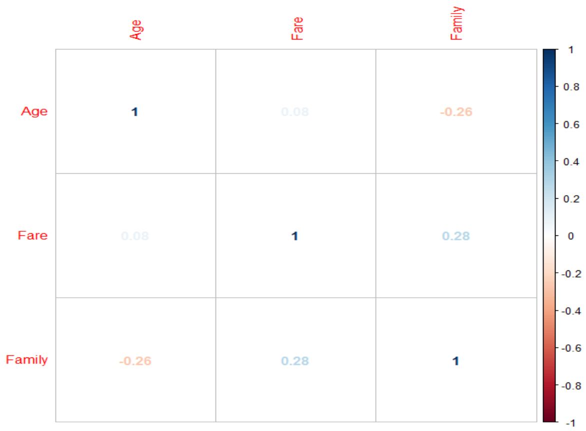


| Row ID | D Survived | D Pclass | D Sex | D Embarked |
|---|---|---|---|---|
| Survived | 1 | 0.34 | 0.543 | 0.171 |
| Pclass | 0.34 | 1 | 0.138 | 0.262 |
| Sex | 0.543 | 0.138 | 1 | 0.12 |
| Embarked | 0.171 | 0.262 | 0.12 | 1 |

Nominal <-> nominal: Pearson's chi square test on the contingency table. This value is then normalized to a range [0, 1] using Cramer's V, whereby 0 represents no correlation and 1 a strong correlation.

The highest correlation is between "Survived" and "Sex" attributes, but this value(0.543) is not enough to remove one of them.





Numeric <-> numeric: Pearson's product-moment coefficient. The value of this measure ranges from -1 (strong negative correlation) to 1 (strong positive correlation). A value of 0 represents no linear correlation (the columns might still be highly dependent on each other, though).

It can be seen that there are not strong negative or positive correlation between attributes. Therefore, the dataset's dimension is not reduced.

# 2. Clustering Analysis

Given a set of multi-dimensional objects, to cluster means creating groups (partitions) of similar (related), this allows you to discover in a data set any "natural" bonds between objects. In clustering there is no right answer: different algorithms can lead to different cluster. In this analysis, the techniques are used:

- K-means clustering
- DBSCAN clustering
- Hierarchical clustering

In this section, it is decided to;

- Exclude the attributes("SibSp" and "Parch") from the dataset, and use of new column "Family" obtained by merging "SibSp" and "Parch".
- Remove the records which are decided to be outlier on Data Understanding part.

## 2.1 K-means Clustering

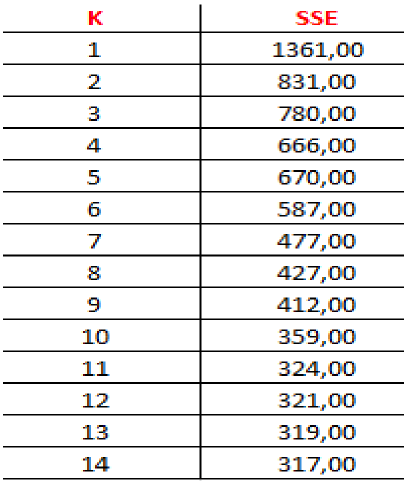
Major parts of the K-means algorithm are;

1. Select k points as initial centroids

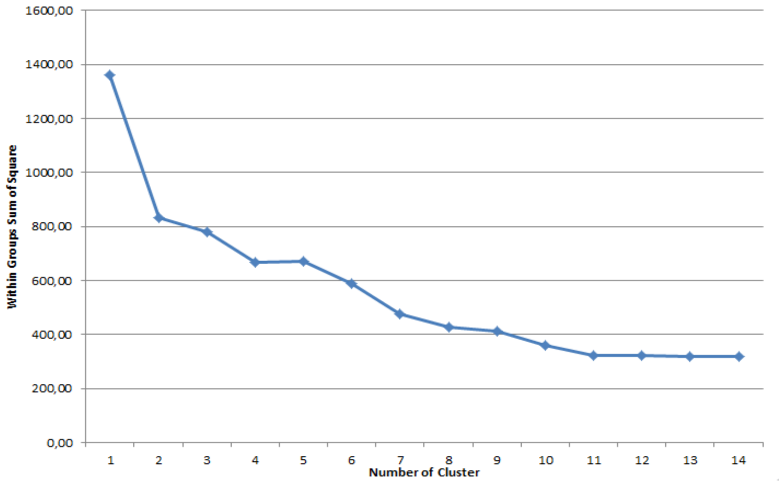
2. Creates k clusters associating each point closest to the centroid

3. Repeat from step 1 until the algorithm converges

The value k is given as input by the user; the algorithm is strongly influenced by the choice of this parameter. Therefore, determining best value of k is an initial part before running the algorithm. Below graphs help to select best k.



| K | SSE |
|---|---|
| 1 | 1361,00 |
| 2 | 831,00 |
| 3 | 780,00 |
| 4 | 666,00 |
| 5 | 670,00 |
| 6 | 587,00 |
| 7 | 477,00 |
| 8 | 427,00 |
| 9 | 412,00 |
| 10 | 359,00 |
| 11 | 324,00 |
| 12 | 321,00 |
| 13 | 319,00 |
| 14 | 317,00 |



SimpleKmeans function in R is applied with the following sets of parameter;

With the first configuration (k = 2) it is found that:

,

- Almost all died are found in Cluster 0. It can be seen in Figure 1.
- While denser for female are in Cluster 1, for male are in Cluster 0.(Figure 2)
- Quite dramatically visually in Figure 3 it seems that sex of the passenger shows significant clustering around survival chances.

Note that in the all graphs:
For “Survive”, 1=Died, 2=Survived passengers.
For “Sex”, female=1, male=2.
For “Embarked”, 1=C, 2=Q and 3=S

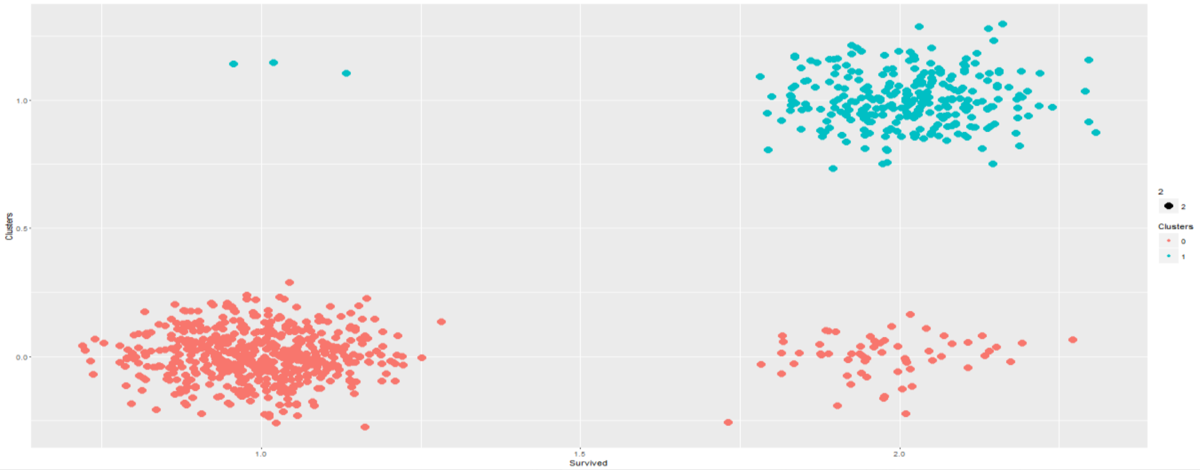


Figure 1-Survived-Cluster

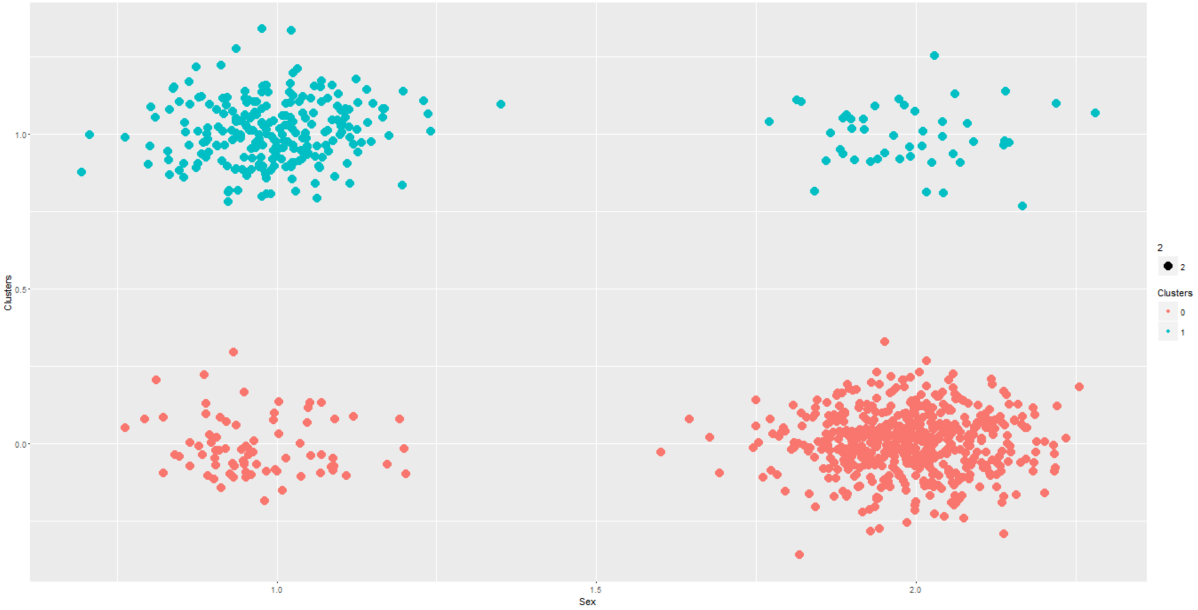


Figure 2-Sex-Cluster

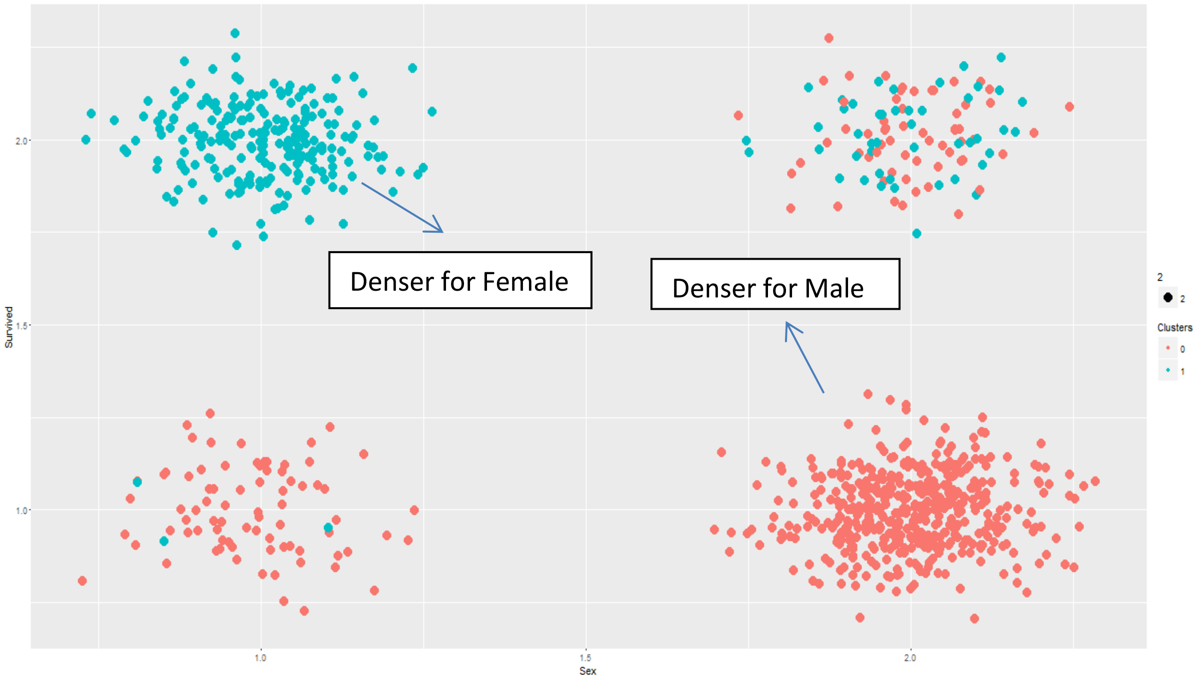


Figure 3- Sex-Survived

- In Figure 4, it can be seen that “Fare” and “Age” are not suitable for k-means method because of different density.
- All of died passengers who are in the 2nd and 3rd are obtained in Cluster 0. At the same time, all of survivors who are in the 1st are occurred in Cluster 1(Figure 5)

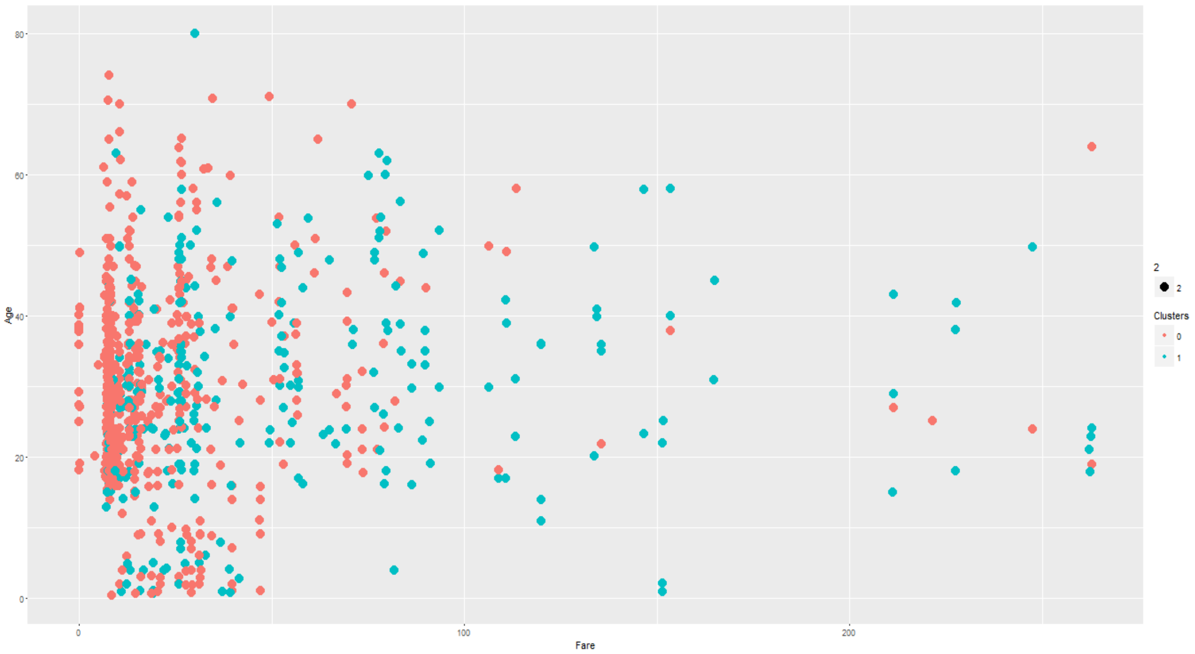


Figure 4-Fare-Age

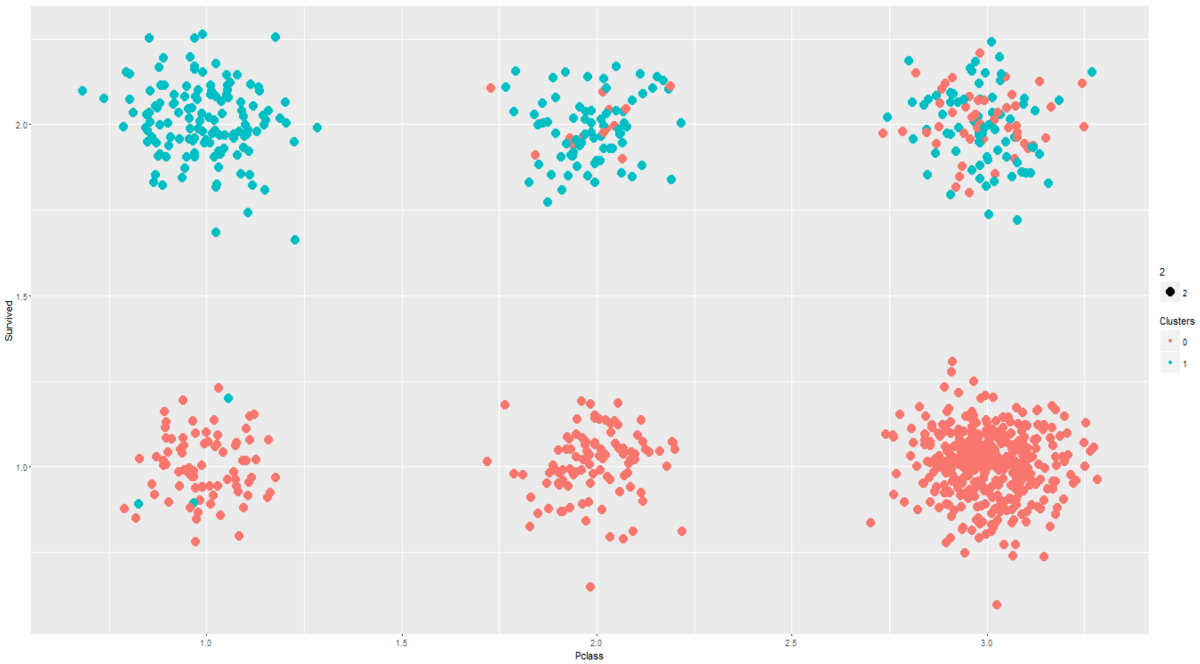


Figure 5- Pclass-Survived

With the second configuration (k=3);

- In Figure 6, If  “Fare” is more than 100, these values seem only cluster 1.
- Figure 7 says that dense for first class is occurred in cluster 1 and dense for third class are occurred in cluster 0.

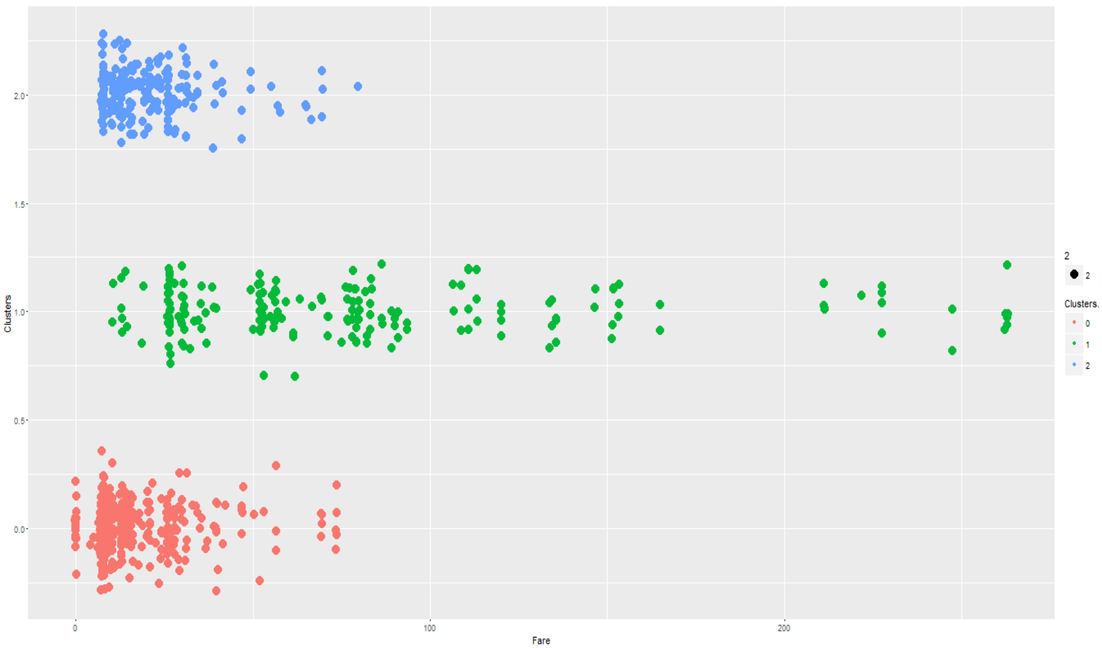


Figure 6-Fare-Age

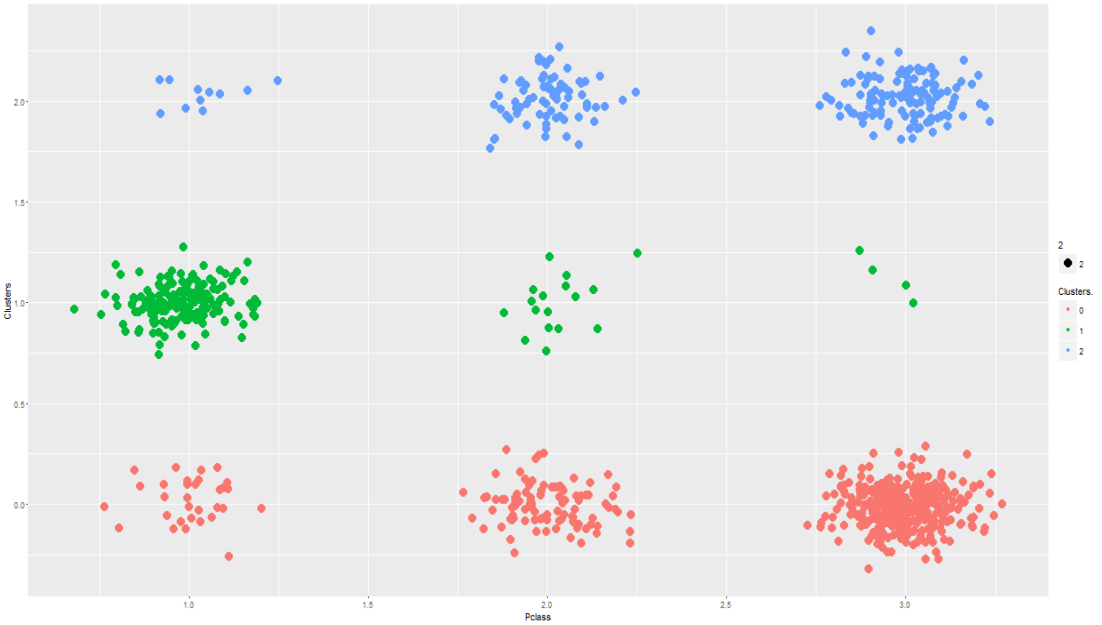


Figure 7-Pclass-Cluster

- In Figure 8, almost no passengers who embark from Queenstown are occurred in cluster 1.
- Figure 9 says that first class is assigned cluster 1, survivors who are in the second or third class are assigned cluster 2, died passengers who are in the second or third class are assigned cluster 0.

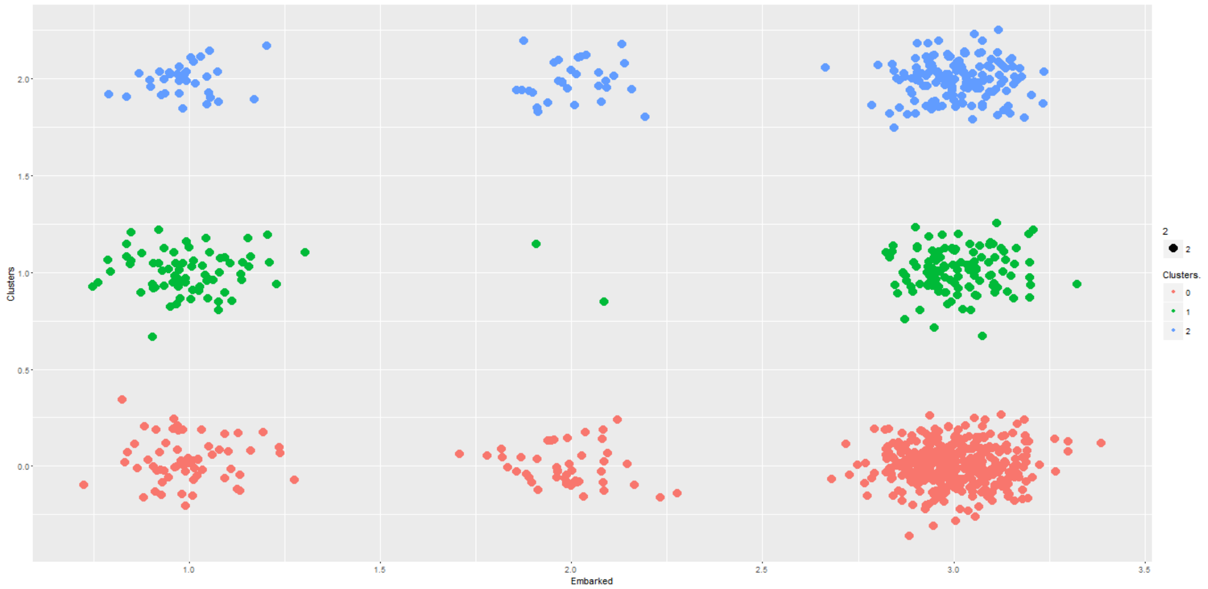


Figure 8- Embarked-Cluster

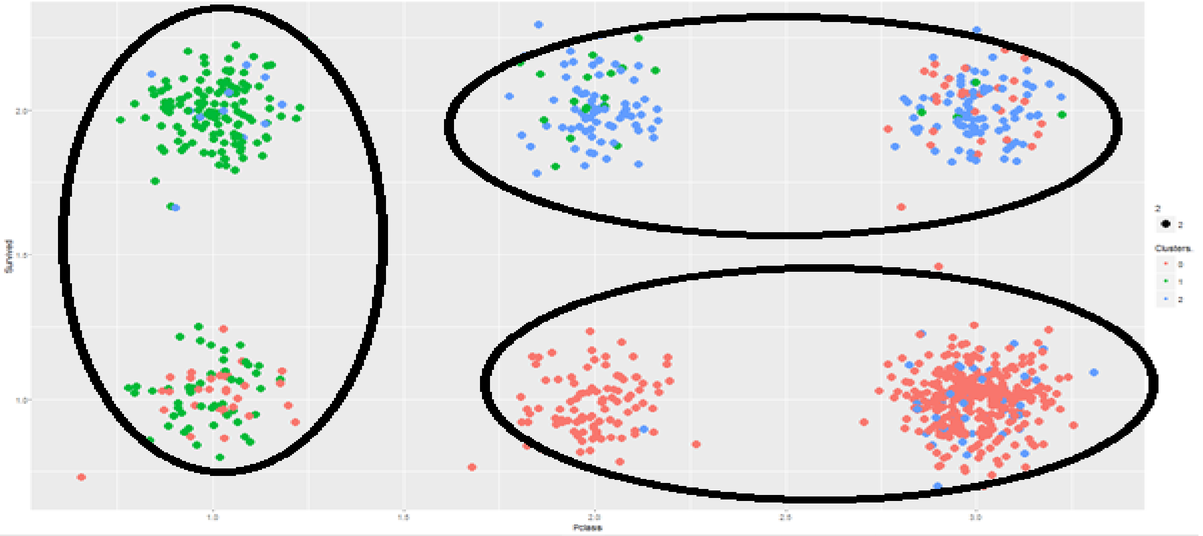


Figure 9- Pclass-Survived

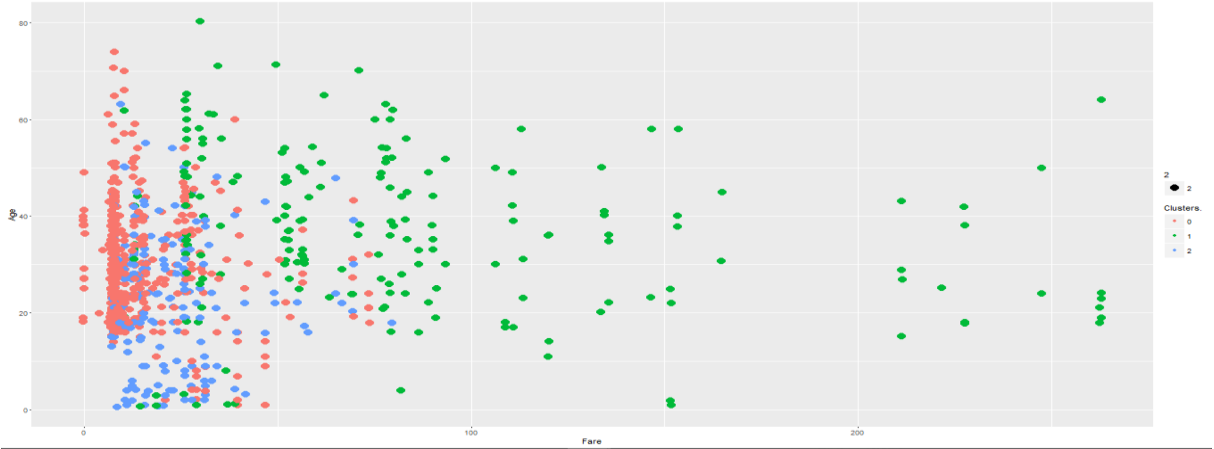
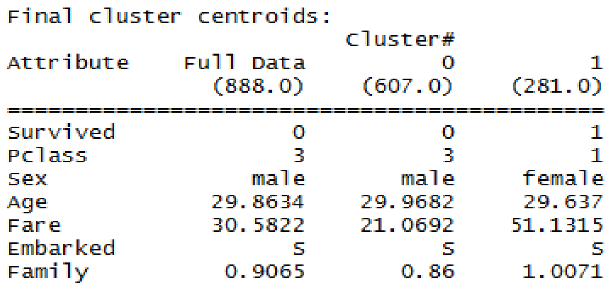
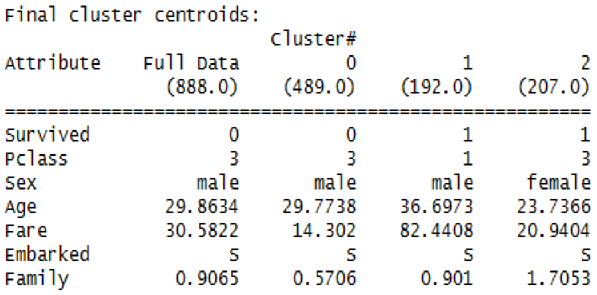


Figure 10-Fare-Age

Below figures show the summary of clusters.



```
Final cluster centroids:
                            Cluster#
Attribute      Full Data          0          1
                 (888.0)    (607.0)    (281.0)
===============================================
Survived               0          0          1
Pclass                 3          3          1
Sex                 male       male     female
Age              29.8634    29.9682     29.637
Fare             30.5822    21.0692    51.1315
Embarked               S          S          S
Family            0.9065       0.86     1.0071
```



```
Final cluster centroids:
                               Cluster#
Attribute      Full Data          0          1          2
                 (888.0)    (489.0)    (192.0)    (207.0)
==========================================================
Survived               0          0          1          1
Pclass                 3          3          1          3
Sex                 male       male       male     female
Age              29.8634    29.7738    36.6973    23.7366
Fare             30.5822     14.302    82.4408    20.9404
Embarked               S          S          S          S
Family            0.9065     0.5706      0.901     1.7053
```

## 2.2 DBSCAN Clustering

Density-based spatial clustering of applications with noise is a data clustering algorithm. For DBSCAN, the parameters ε and minPts are needed. The parameters must be specified by the user. Ideally, the value of ε is given by the problem to solve and minPts is then the desired minimum cluster size.

KNNdist and kNNdistplot functions İn R are used for determining the value eps and Minpts. In the first case MinPts=10, Eps=9(Figure 11) it showed a single cluster containing 100% of the points and a very small number of point noise (65). This means that at this level it is not possible distinguish any areas to higher density.

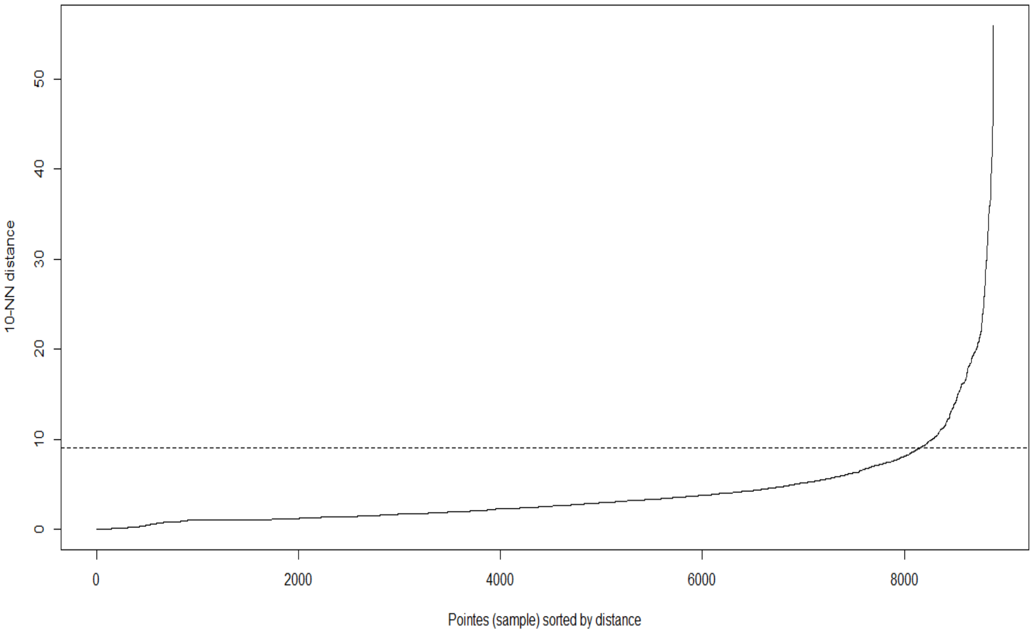


Figure 11- 10-NN Distance

In Figure 12 and 14, the pink points assign noise points.

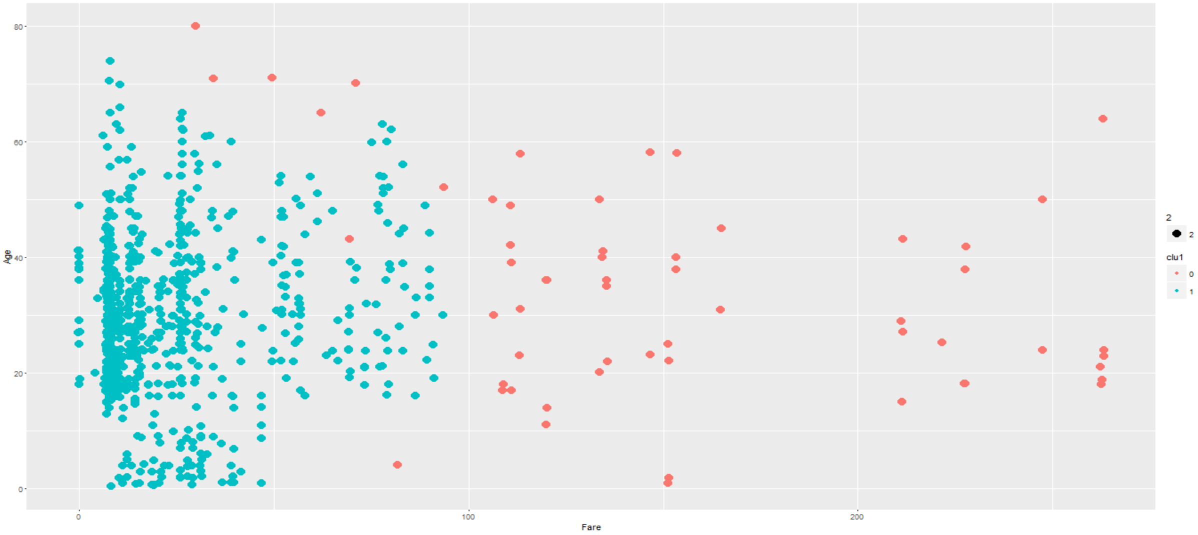


Figure 12- Fare-Age

Therefore, it proceeded to decrease the k to 6, after running DBSCAN with MinPts = 6 and Eps = 9 cannot further decrease and therefore, test is carried out with decreasing Eps. The results obtained show that the decrease of EPS corresponds to decreasing of the dominant cluster, increasing of the noise points and an increasing small cluster. As a result, running with MinPts=6 and Eps=7(Figure 13) below result are obtained. Three clusters occur with these parameters. This result is enough to distinguish areas that have higher density than others. (Figure 14)

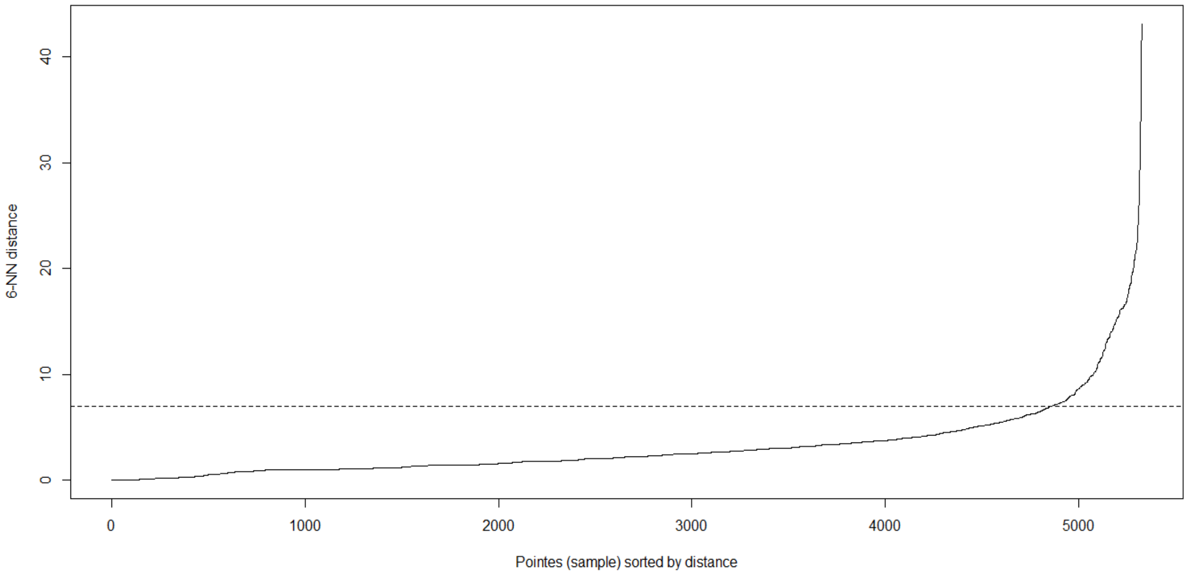


Figure 13- 6-NN Distance

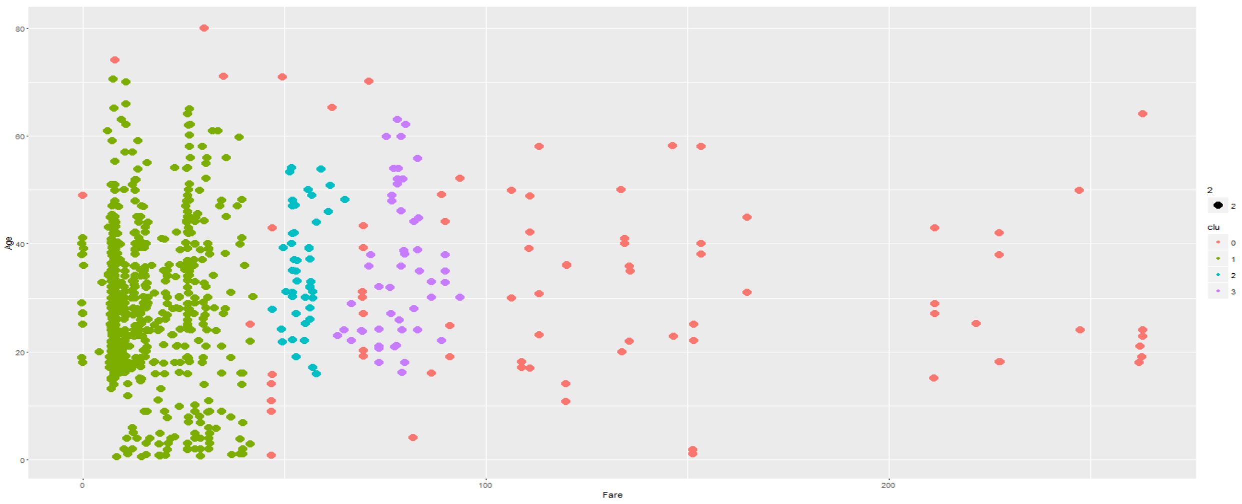


Figure 14- Fare-Age

## 2.2 Hierarchical Clustering

Hierarchical clustering allows you to create a hierarchy of clusters. It compares two clusters and measuring the distance between the two. Depending on the distance, it creates a dendrogram that blends (or divide) clusters in pairs until it is no longer possible merger (or division).

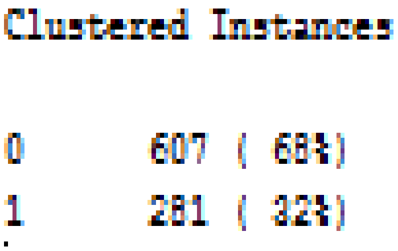
When applying hierarchical clustering, sampling is useful because of higher complexity. However, for this analysis sampling is not applied on the data.

**Ward-Method Hierarchical Clustering:**

For Ward's method, the proximity between two clusters is defined as the increase in the squared error that results when two clusters are merged. Thus, this method uses the same objective function as K-means clustering. After applying Ward-Linkage on the data, the results clusters are almost same with K-means clusters when k is equal to 2 and 3 respectively.

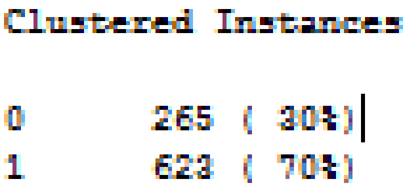
When k=2 the results are:

K-means:



```
Clustered Instances

0       607 ( 68%)
1       281 ( 32%)
```

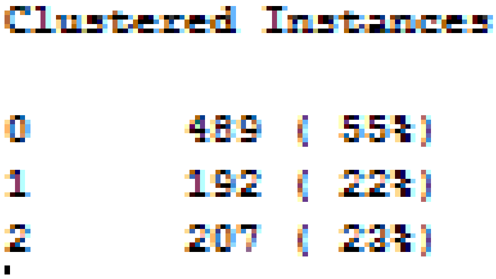
Ward-Linkage:



```
Clustered Instances

0       265 ( 30%)
1       623 ( 70%)
```

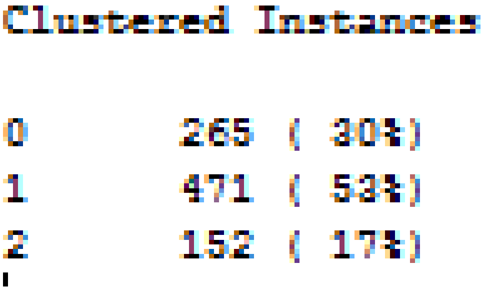
When k=3 the results are:

K-means:

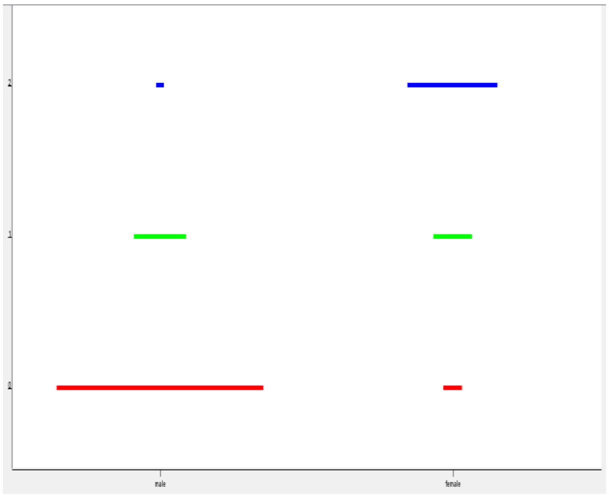


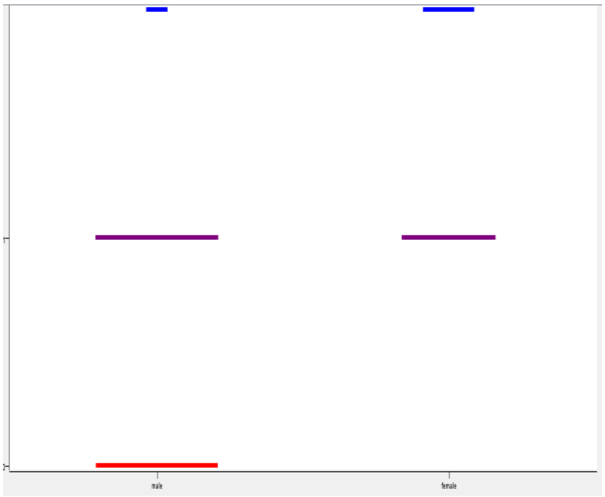
```
Clustered Instances

0       489 ( 55%)
1       192 ( 22%)
2       207 ( 23%)
```

Ward-Linkage:



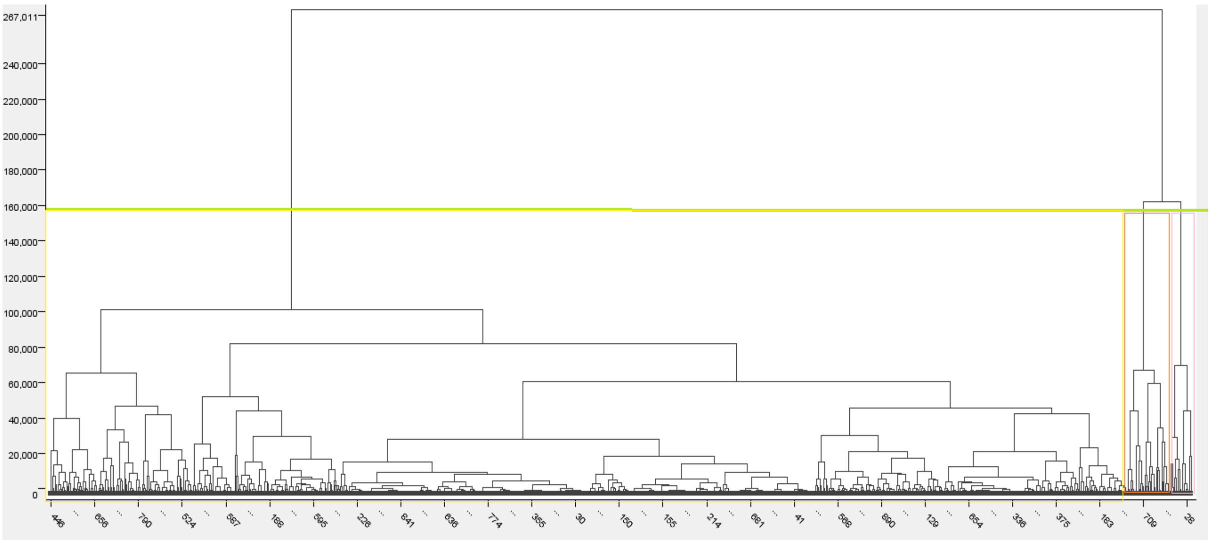
```
Clustered Instances

0       265 ( 30%)
1       471 ( 53%)
2       152 ( 17%)
```

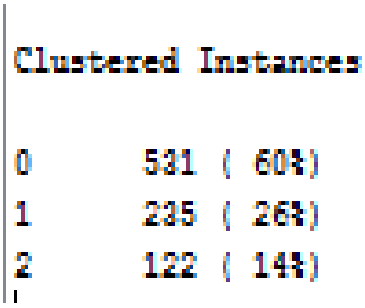




In the first cluster, men have bigger portion than women. In the second cluster, they are nearly same and in the third cluster number of women passenger are more than men passengers.

**Complete-Linkage Clustering:**

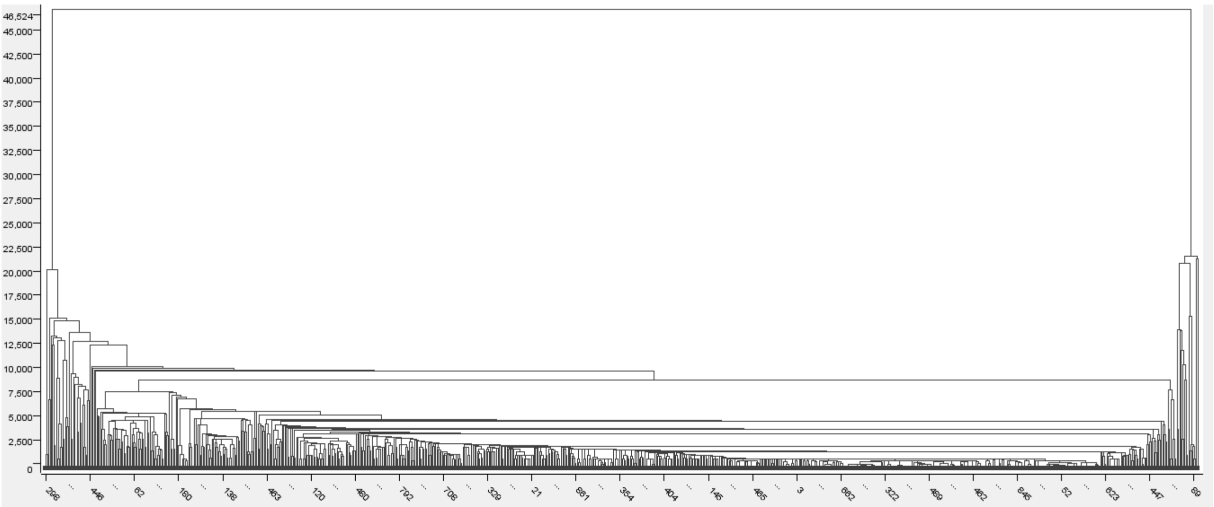


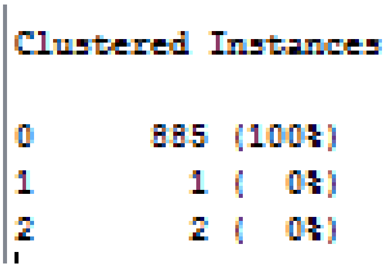


```
Clustered Instances

0       531 ( 60%)
1       235 ( 26%)
2       122 ( 14%)
```

**Single-Linkage Clustering:**

Dendrogram of Single-Linkage is not suitable to see the results.





```
Clustered Instances

0       885 (100%)
1         1 (  0%)
2         2 (  0%)
```

# 3. Association Rules Mining

Association rule learning is a method for discovering interesting relations between variables in datasets. In this section, the data is prepared to be feasible for methods in R, and association rule mining is applied on “Titanic Disaster Data Set”.

## 3.1 Pre-Processing

Five major attributes which are “**Sex**”, “**Survived**”, “**Age**”, “**Pclass**” and “**Family**” are considered to better analyze for Titanic Data Set. Quantitative attributes have many distinct values. Therefore, discretization techniques are used on these attributes.

### Age column

When the discretization technique “**Equal Frequency**” is applied on the Age Column, intervals split the data like this way:

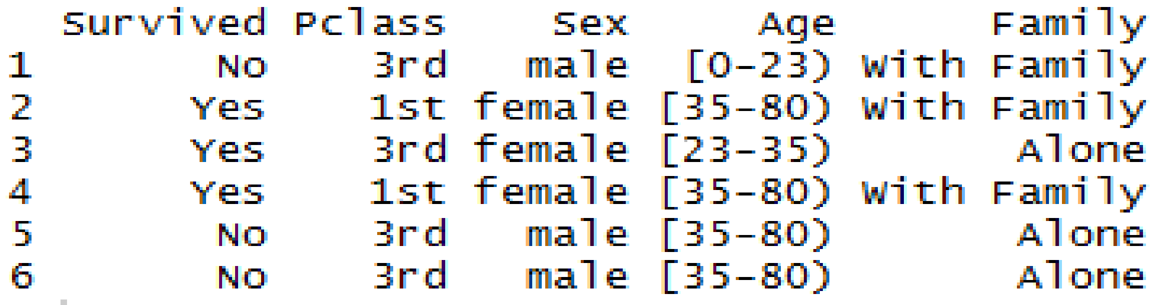
```
[ 0.42,23.5) [23.50,34.5) [34.50,80.0]
         300          297          294
```

First range consist of [0.42, 23.5) that contains 300 records, second range is [23.5, 34.5) contains 297 records and the last one is [34.5, 80] contains 294 records. Age is an integer value, so intervals are taken [0, 23), [23, 35) and [35, 80].

### Family Column

Family Column is created by adding other two columns “**SibSp**” and “**Parch**” It shows that any Passenger who had a member or lots of members from family in Titanic. Therefore, it is categorized with “**Alone**” (when the addition is equal to 0) or “**With Family**” (when the addition is greater than 0) records.

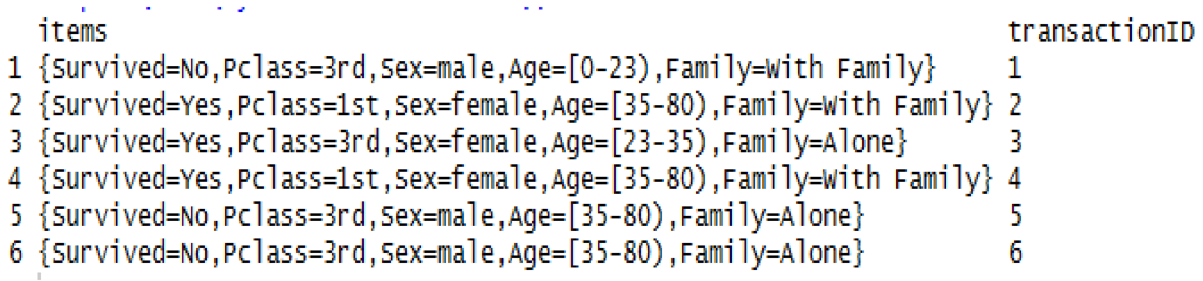
After the pre-processing step, the first 6 passenger’s example of remaining “Titanic Data” is shown in below.



```
  Survived Pclass    Sex     Age      Family
1       No    3rd   male  [0-23) With Family
2      Yes    1st female [35-80) With Family
3      Yes    3rd female [23-35)       Alone
4      Yes    1st female [35-80) With Family
5       No    3rd   male [35-80)       Alone
6       No    3rd   male [35-80)       Alone
```

## Transform to Transaction Data

After transforming the remaining data, example of first 6 transactions are shown in below.



```
  items                                                            transactionID
1 {Survived=No,Pclass=3rd,Sex=male,Age=[0-23),Family=With Family}     1
2 {Survived=Yes,Pclass=1st,Sex=female,Age=[35-80),Family=With Family} 2
3 {Survived=Yes,Pclass=3rd,Sex=female,Age=[23-35),Family=Alone}       3
4 {Survived=Yes,Pclass=1st,Sex=female,Age=[35-80),Family=With Family} 4
5 {Survived=No,Pclass=3rd,Sex=male,Age=[35-80),Family=Alone}          5
6 {Survived=No,Pclass=3rd,Sex=male,Age=[35-80),Family=Alone}          6
```

The graph of attributes distributions according to their frequencies in the transaction data are shown in Figure 15.

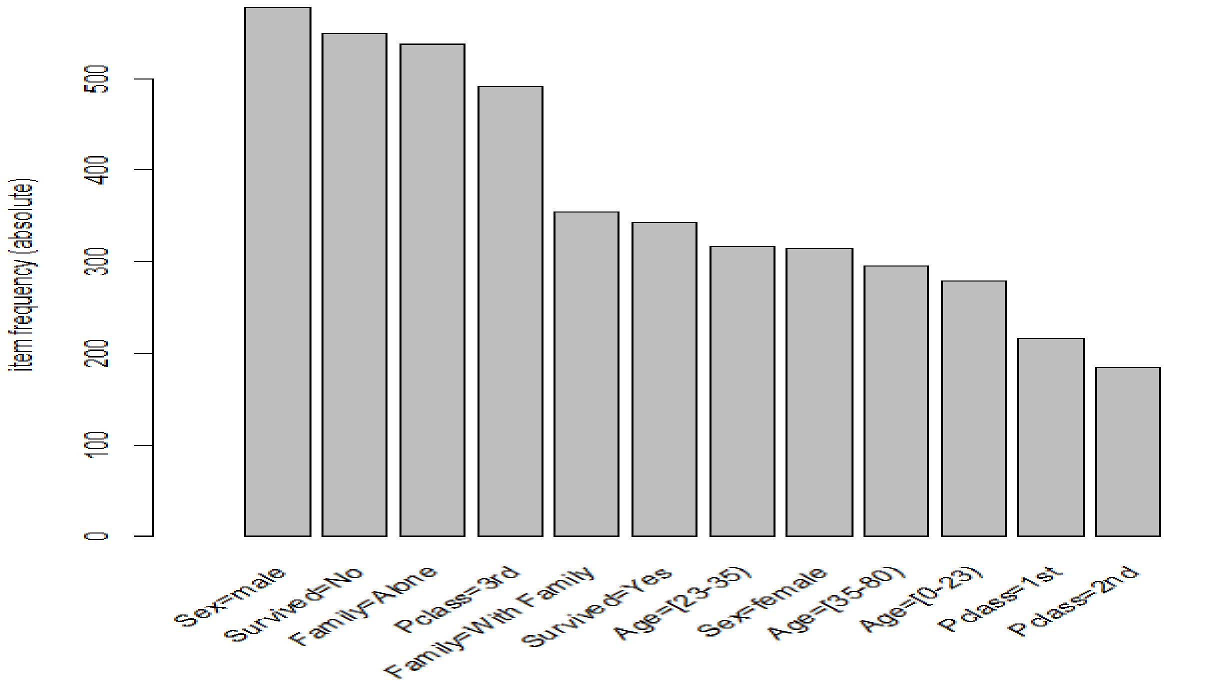


Figure 15- Frequency of items in the whole dataset

The most frequent items are:

```
most frequent items:
Sex=male   Survived=No   Family=Alone   Pclass=3rd   Family=With Family (Other)
577        549           537            491          354                  1947
```

## 3.2 Frequent Itemset Generation

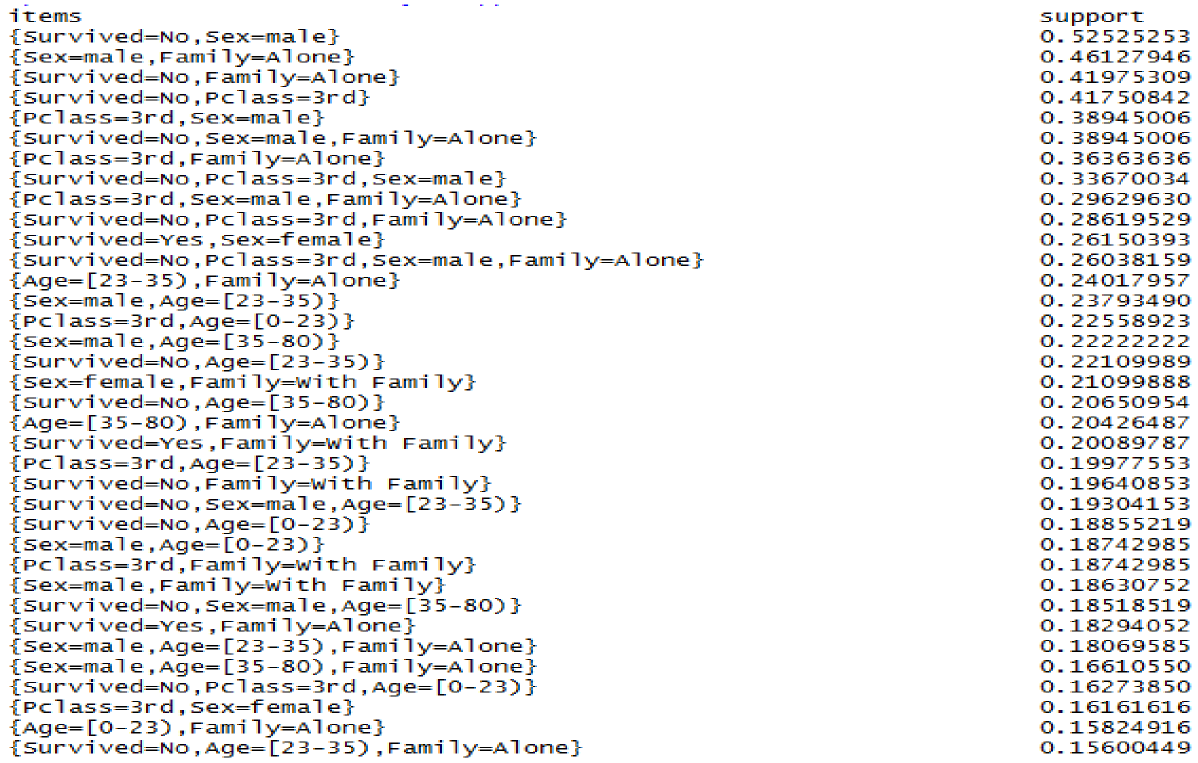
The R provides several algorithms for the detection of frequent itemset (Eclat, Apriori etc..). For this analysis, the Apriori algorithm is used. Parameters are 2 and 0.5% for minimum length (minlen), minimum support (supp) respectively. Minimum length is chosen 2 because, if it is selected lower than 2, there is some sets include only one item. They are meaningless. When the “Closed” or “Maximal” itemsets are wanted to find in R, it is easy to extract them by changing the target parameter (“closed” or “maximal”) in apriori function. For this analysis, the target parameter is set “frequent”.

Notice that, starting with parameter minimum support is selected 10% and by using apriori algorithm, most frequent items are determined:

```
most frequent items:
Survived=No  Sex=male  Family=Alone   Pclass=3rd  Age=[23-35)  (Other)
30           30        28             25           17            74
```

It is not prefered way because, lots of important frequent items are hidden.Finally, minimum support is decreased until 0.5%.

The Figure 16 shows the highest frequent itemset whose support is greater than 0.5% (for this table 15%) and length is greater than 2.



```
items                                                                  support
{Survived=No,Sex=male}                                                 0.52525253
{Sex=male,Family=Alone}                                                0.46127946
{Survived=No,Family=Alone}                                             0.41975309
{Survived=No,Pclass=3rd}                                               0.41750842
{Pclass=3rd,Sex=male}                                                  0.38945006
{Survived=No,Sex=male,Family=Alone}                                    0.38945006
{Pclass=3rd,Family=Alone}                                              0.36363636
{Survived=No,Pclass=3rd,Sex=male}                                      0.33670034
{Pclass=3rd,Sex=male,Family=Alone}                                     0.29629630
{Survived=No,Pclass=3rd,Family=Alone}                                  0.28619529
{Survived=Yes,Sex=female}                                              0.26150393
{Survived=No,Pclass=3rd,Sex=male,Family=Alone}                         0.26038159
{Age=[23-35),Family=Alone}                                             0.24017957
{Sex=male,Age=[23-35)}                                                 0.23793490
{Pclass=3rd,Age=[0-23)}                                                0.22558923
{Sex=male,Age=[35-80)}                                                 0.22222222
{Survived=No,Age=[23-35)}                                              0.22109989
{Sex=female,Family=With Family}                                        0.21099888
{Survived=No,Age=[35-80)}                                              0.20650954
{Age=[35-80),Family=Alone}                                             0.20426487
{Survived=Yes,Family=With Family}                                      0.20089787
{Pclass=3rd,Age=[23-35)}                                               0.19977553
{Survived=No,Family=With Family}                                       0.19640853
{Survived=No,Sex=male,Age=[23-35)}                                     0.19304153
{Survived=No,Age=[0-23)}                                               0.18855219
{Sex=male,Age=[0-23)}                                                  0.18742985
{Pclass=3rd,Family=With Family}                                        0.18742985
{Sex=male,Family=With Family}                                          0.18630752
{Survived=No,Sex=male,Age=[35-80)}                                     0.18518519
{Survived=Yes,Family=Alone}                                            0.18294052
{Sex=male,Age=[23-35),Family=Alone}                                    0.18069585
{Sex=male,Age=[35-80),Family=Alone}                                    0.16610550
{Survived=No,Pclass=3rd,Age=[0-23)}                                    0.16273850
{Pclass=3rd,Sex=female}                                                0.16161616
{Age=[0-23),Family=Alone}                                              0.15824916
{Survived=No,Age=[23-35),Family=Alone}                                 0.15600449
```

Figure 16- Frequent Itemset

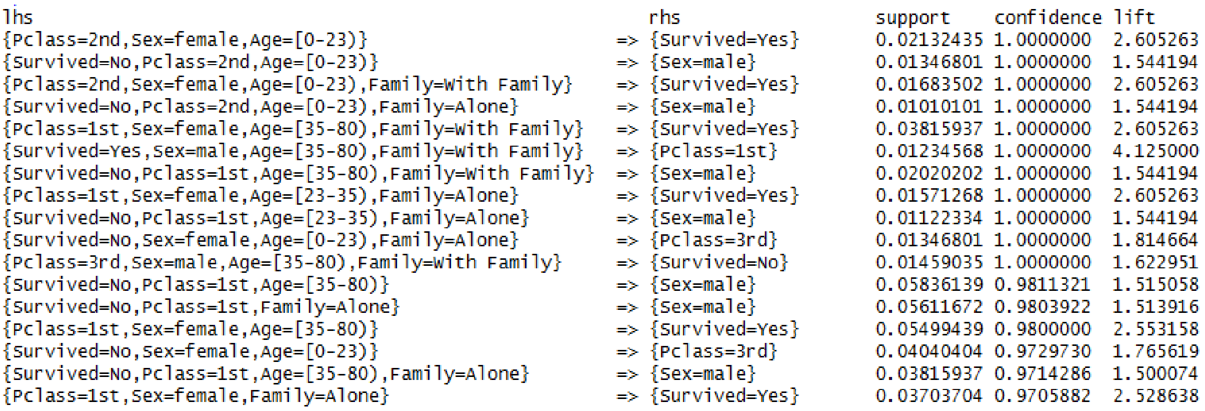
When comparing Figure 15 and 16, one of them shows the most frequent items in the whole dataset are “Survived=No”, “Sex=male”, “Family=Alone” , “Pclass=3$^{rd}$” and the other one illustrates obtained frequent itemsets with these items have high support. Moreover, when looking to frequencies of attributes in the frequent itemsets, the most frequent items are:

```
Sex=male Survived=Yes Family=With Family  Family=Alone  Survived=No (Other)
130      129          127                 125           116         675
```

So, for example “Survived=Yes” or “Family=With Family” might has a remerkable frequent, they only have low support, but important situation for extract rules.

## 3.3 Rule Generation

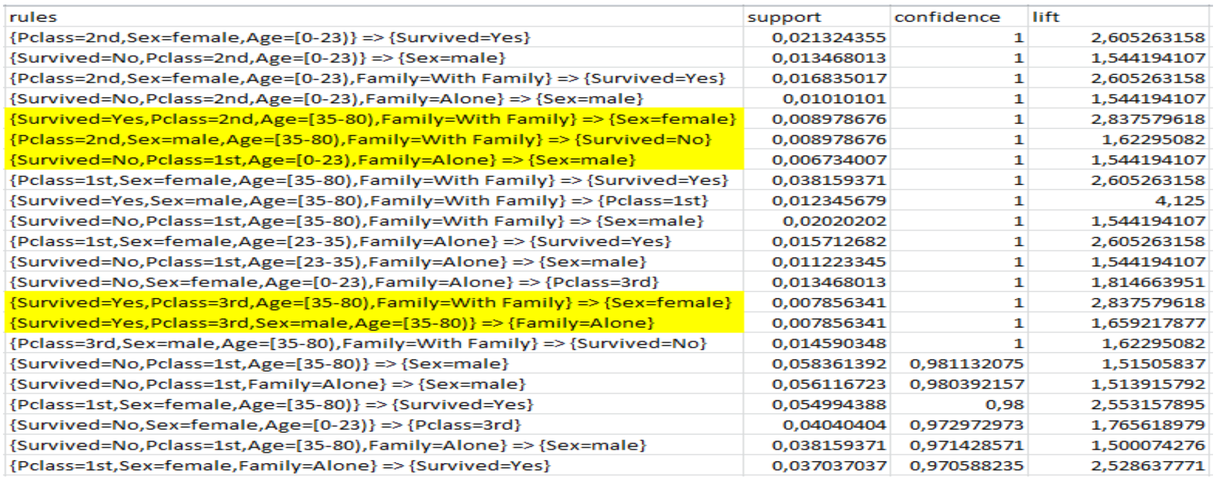
For extracting association rules, apriori function is used in R. Parameters are starting with minimum support(supp)=1%, confidence=80%, minlen=2. For getting the rules, target parameter is chosen “rules”.



```
lhs                                                       rhs              support    confidence lift
{Pclass=2nd,Sex=female,Age=[0-23)}                     => {Survived=Yes}   0.02132435 1.0000000  2.605263
{Survived=No,Pclass=2nd,Age=[0-23)}                    => {Sex=male}       0.01346801 1.0000000  1.544194
{Pclass=2nd,Sex=female,Age=[0-23),Family=With Family}  => {Survived=Yes}   0.01683502 1.0000000  2.605263
{Survived=No,Pclass=2nd,Age=[0-23),Family=Alone}       => {Sex=male}       0.01010101 1.0000000  1.544194
{Pclass=1st,Sex=female,Age=[35-80),Family=With Family} => {Survived=Yes}   0.03815937 1.0000000  2.605263
{Survived=Yes,Sex=male,Age=[35-80),Family=With Family} => {Pclass=1st}     0.01234568 1.0000000  4.125000
{Survived=No,Pclass=1st,Age=[35-80),Family=With Family} => {Sex=male}      0.02020202 1.0000000  1.544194
{Pclass=1st,Sex=female,Age=[23-35),Family=Alone}       => {Survived=Yes}   0.01571268 1.0000000  2.605263
{Survived=No,Pclass=1st,Age=[23-35),Family=Alone}      => {Sex=male}       0.01122334 1.0000000  1.544194
{Survived=No,Sex=female,Age=[0-23),Family=Alone}       => {Pclass=3rd}     0.01346801 1.0000000  1.814664
{Pclass=3rd,Sex=male,Age=[35-80),Family=With Family}   => {Survived=No}    0.01459035 1.0000000  1.622951
{Survived=No,Pclass=1st,Age=[35-80)}                   => {Sex=male}       0.05836139 0.9811321  1.515058
{Survived=No,Pclass=1st,Family=Alone}                  => {Sex=male}       0.05611672 0.9803922  1.513916
{Pclass=1st,Sex=female,Age=[35-80)}                    => {Survived=Yes}   0.05499439 0.9800000  2.553158
{Survived=No,Sex=female,Age=[0-23)}                    => {Pclass=3rd}     0.04040404 0.9729730  1.765619
{Survived=No,Pclass=1st,Age=[35-80),Family=Alone}      => {Sex=male}       0.03815937 0.9714286  1.500074
{Pclass=1st,Sex=female,Family=Alone}                   => {Survived=Yes}   0.03703704 0.9705882  2.528638
```

Figure 17-0.5% support 80% confidence

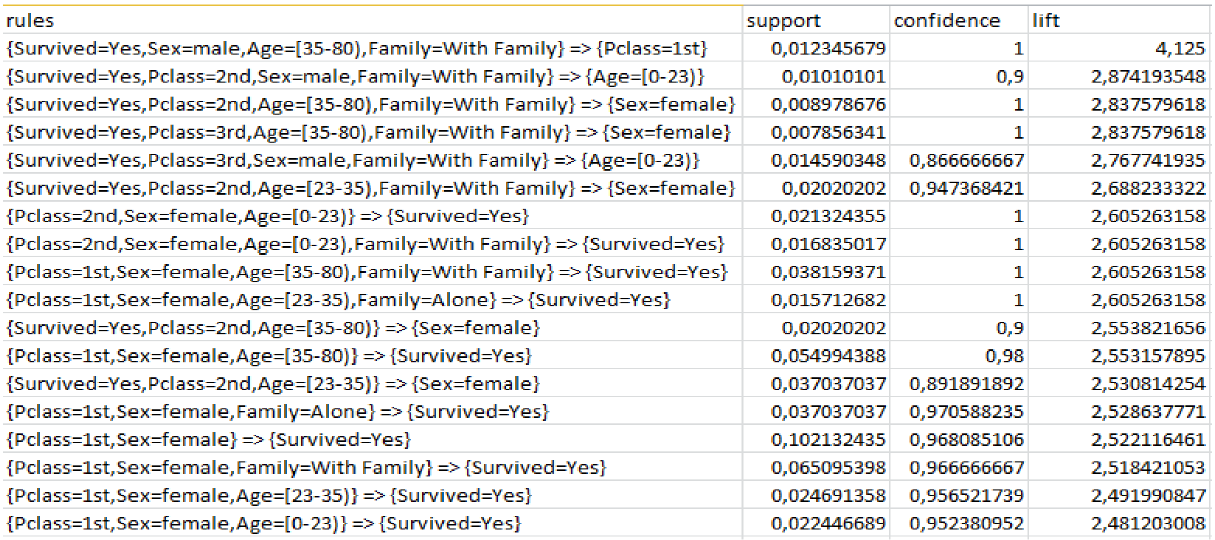
When parameters are selected 0.5%, 80% for minimum support and confidence respectively, it is shown in Figure 17.



| rules | support | confidence | lift |
|---|---|---|---|
| {Pclass=2nd,Sex=female,Age=[0-23)} => {Survived=Yes} | 0,021324355 | 1 | 2,605263158 |
| {Survived=No,Pclass=2nd,Age=[0-23)} => {Sex=male} | 0,013468013 | 1 | 1,544194107 |
| {Pclass=2nd,Sex=female,Age=[0-23),Family=With Family} => {Survived=Yes} | 0,016835017 | 1 | 2,605263158 |
| {Survived=No,Pclass=2nd,Age=[0-23),Family=Alone} => {Sex=male} | 0,01010101 | 1 | 1,544194107 |
| {Survived=Yes,Pclass=2nd,Age=[35-80),Family=With Family} => {Sex=female} | 0,008978676 | 1 | 2,837579618 |
| {Pclass=2nd,Sex=male,Age=[35-80),Family=With Family} => {Survived=No} | 0,008978676 | 1 | 1,62295082 |
| {Survived=No,Pclass=1st,Age=[0-23),Family=Alone} => {Sex=male} | 0,006734007 | 1 | 1,544194107 |
| {Pclass=1st,Sex=female,Age=[35-80),Family=With Family} => {Survived=Yes} | 0,038159371 | 1 | 2,605263158 |
| {Survived=Yes,Sex=male,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,012345679 | 1 | 4,125 |
| {Survived=No,Pclass=1st,Age=[35-80),Family=With Family} => {Sex=male} | 0,02020202 | 1 | 1,544194107 |
| {Pclass=1st,Sex=female,Age=[23-35),Family=Alone} => {Survived=Yes} | 0,015712682 | 1 | 2,605263158 |
| {Survived=No,Pclass=1st,Age=[23-35),Family=Alone} => {Sex=male} | 0,011223345 | 1 | 1,544194107 |
| {Survived=No,Sex=female,Age=[0-23),Family=Alone} => {Pclass=3rd} | 0,013468013 | 1 | 1,814663951 |
| {Survived=Yes,Pclass=3rd,Age=[35-80),Family=With Family} => {Sex=female} | 0,007856341 | 1 | 2,837579618 |
| {Survived=Yes,Pclass=3rd,Sex=male,Age=[35-80)} => {Family=Alone} | 0,007856341 | 1 | 1,659217877 |
| {Pclass=3rd,Sex=male,Age=[35-80),Family=With Family} => {Survived=No} | 0,014590348 | 1 | 1,62295082 |
| {Survived=No,Pclass=1st,Age=[35-80)} => {Sex=male} | 0,058361392 | 0,981132075 | 1,51505837 |
| {Survived=No,Pclass=1st,Family=Alone} => {Sex=male} | 0,056116723 | 0,980392157 | 1,513915792 |
| {Pclass=1st,Sex=female,Age=[35-80)} => {Survived=Yes} | 0,054994388 | 0,98 | 2,553157895 |
| {Survived=No,Sex=female,Age=[0-23)} => {Pclass=3rd} | 0,04040404 | 0,972972973 | 1,765618979 |
| {Survived=No,Pclass=1st,Age=[35-80),Family=Alone} => {Sex=male} | 0,038159371 | 0,971428571 | 1,500074276 |
| {Pclass=1st,Sex=female,Family=Alone} => {Survived=Yes} | 0,037037037 | 0,970588235 | 2,528637771 |

Figure 18-Rules with 80% confidence

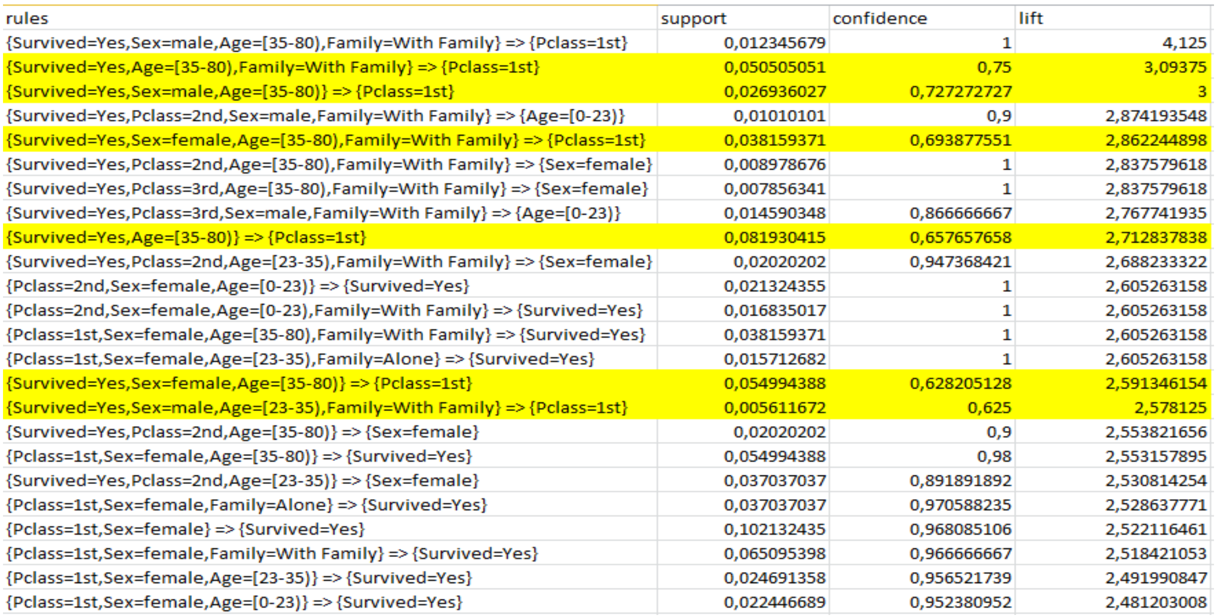
Figure 17, rules are sorted by confidence. It can be seen some rules that have high confidence are lost when comparing with Figure 18(yellow lines). When it is sorted by lift:



| rules | support | confidence | lift |
|---|---|---|---|
| {Survived=Yes,Sex=male,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,012345679 | 1 | 4,125 |
| {Survived=Yes,Pclass=2nd,Sex=male,Family=With Family} => {Age=[0-23)} | 0,01010101 | 0,9 | 2,874193548 |
| {Survived=Yes,Pclass=2nd,Age=[35-80),Family=With Family} => {Sex=female} | 0,008978676 | 1 | 2,837579618 |
| {Survived=Yes,Pclass=3rd,Age=[35-80),Family=With Family} => {Sex=female} | 0,007856341 | 1 | 2,837579618 |
| {Survived=Yes,Pclass=3rd,Sex=male,Family=With Family} => {Age=[0-23)} | 0,014590348 | 0,866666667 | 2,767741935 |
| {Survived=Yes,Pclass=2nd,Age=[23-35),Family=With Family} => {Sex=female} | 0,02020202 | 0,947368421 | 2,688233322 |
| {Pclass=2nd,Sex=female,Age=[0-23)} => {Survived=Yes} | 0,021324355 | 1 | 2,605263158 |
| {Pclass=2nd,Sex=female,Age=[0-23),Family=With Family} => {Survived=Yes} | 0,016835017 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[35-80),Family=With Family} => {Survived=Yes} | 0,038159371 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[23-35),Family=Alone} => {Survived=Yes} | 0,015712682 | 1 | 2,605263158 |
| {Survived=Yes,Pclass=2nd,Age=[35-80)} => {Sex=female} | 0,02020202 | 0,9 | 2,553821656 |
| {Pclass=1st,Sex=female,Age=[35-80)} => {Survived=Yes} | 0,054994388 | 0,98 | 2,553157895 |
| {Survived=Yes,Pclass=2nd,Age=[23-35)} => {Sex=female} | 0,037037037 | 0,891891892 | 2,530814254 |
| {Pclass=1st,Sex=female,Family=Alone} => {Survived=Yes} | 0,037037037 | 0,970588235 | 2,528637771 |
| {Pclass=1st,Sex=female} => {Survived=Yes} | 0,102132435 | 0,968085106 | 2,522116461 |
| {Pclass=1st,Sex=female,Family=With Family} => {Survived=Yes} | 0,065095398 | 0,966666667 | 2,518421053 |
| {Pclass=1st,Sex=female,Age=[23-35)} => {Survived=Yes} | 0,024691358 | 0,956521739 | 2,491990847 |
| {Pclass=1st,Sex=female,Age=[0-23)} => {Survived=Yes} | 0,022446689 | 0,952380952 | 2,481203008 |

Figure 19-Sorted By Lift

The figure 19 shows only rules which are greater than 80% confidence. When the confidence is taken 50%, the results of rules are shown below:

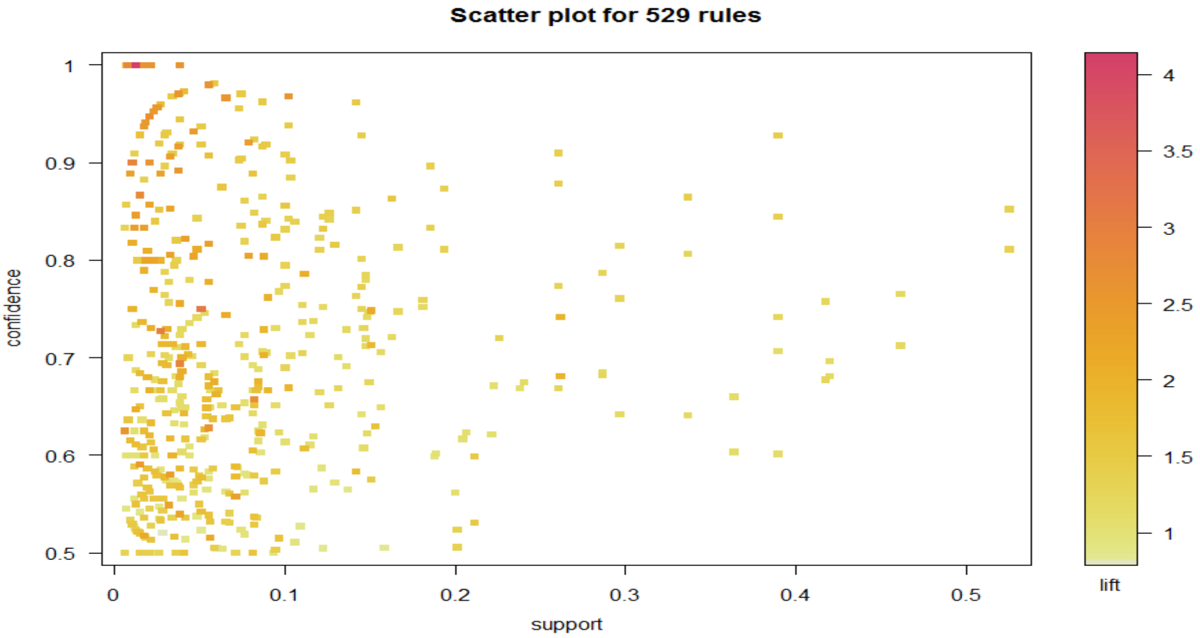


| rules | support | confidence | lift |
|---|---|---|---|
| {Survived=Yes,Sex=male,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,012345679 | 1 | 4,125 |
| {Survived=Yes,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,050505051 | 0,75 | 3,09375 |
| {Survived=Yes,Sex=male,Age=[35-80)} => {Pclass=1st} | 0,026936027 | 0,727272727 | 3 |
| {Survived=Yes,Pclass=2nd,Sex=male,Family=With Family} => {Age=[0-23)} | 0,01010101 | 0,9 | 2,874193548 |
| {Survived=Yes,Sex=female,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,038159371 | 0,693877551 | 2,862244898 |
| {Survived=Yes,Pclass=2nd,Age=[35-80),Family=With Family} => {Sex=female} | 0,008978676 | 1 | 2,837579618 |
| {Survived=Yes,Pclass=3rd,Age=[35-80),Family=With Family} => {Sex=female} | 0,007856341 | 1 | 2,837579618 |
| {Survived=Yes,Pclass=3rd,Sex=male,Family=With Family} => {Age=[0-23)} | 0,014590348 | 0,866666667 | 2,767741935 |
| {Survived=Yes,Age=[35-80)} => {Pclass=1st} | 0,081930415 | 0,657657658 | 2,712837838 |
| {Survived=Yes,Pclass=2nd,Age=[23-35),Family=With Family} => {Sex=female} | 0,02020202 | 0,947368421 | 2,688233322 |
| {Pclass=2nd,Sex=female,Age=[0-23)} => {Survived=Yes} | 0,021324355 | 1 | 2,605263158 |
| {Pclass=2nd,Sex=female,Age=[0-23),Family=With Family} => {Survived=Yes} | 0,016835017 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[35-80),Family=With Family} => {Survived=Yes} | 0,038159371 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[23-35),Family=Alone} => {Survived=Yes} | 0,015712682 | 1 | 2,605263158 |
| {Survived=Yes,Sex=female,Age=[35-80)} => {Pclass=1st} | 0,054994388 | 0,628205128 | 2,591346154 |
| {Survived=Yes,Sex=male,Age=[23-35),Family=With Family} => {Pclass=1st} | 0,005611672 | 0,625 | 2,578125 |
| {Survived=Yes,Pclass=2nd,Age=[35-80)} => {Sex=female} | 0,02020202 | 0,9 | 2,553821656 |
| {Pclass=1st,Sex=female,Age=[35-80)} => {Survived=Yes} | 0,054994388 | 0,98 | 2,553157895 |
| {Survived=Yes,Pclass=2nd,Age=[23-35)} => {Sex=female} | 0,037037037 | 0,891891892 | 2,530814254 |
| {Pclass=1st,Sex=female,Family=Alone} => {Survived=Yes} | 0,037037037 | 0,970588235 | 2,528637771 |
| {Pclass=1st,Sex=female} => {Survived=Yes} | 0,102132435 | 0,968085106 | 2,522116461 |
| {Pclass=1st,Sex=female,Family=With Family} => {Survived=Yes} | 0,065095398 | 0,966666667 | 2,518421053 |
| {Pclass=1st,Sex=female,Age=[23-35)} => {Survived=Yes} | 0,024691358 | 0,956521739 | 2,491990847 |
| {Pclass=1st,Sex=female,Age=[0-23)} => {Survived=Yes} | 0,022446689 | 0,952380952 | 2,481203008 |

Figure 20- 50% Confidence

Yellow lines in the Figure 20 demonstrates rules which have lift>1(positive correlated) and are not exist in the Figure 19.

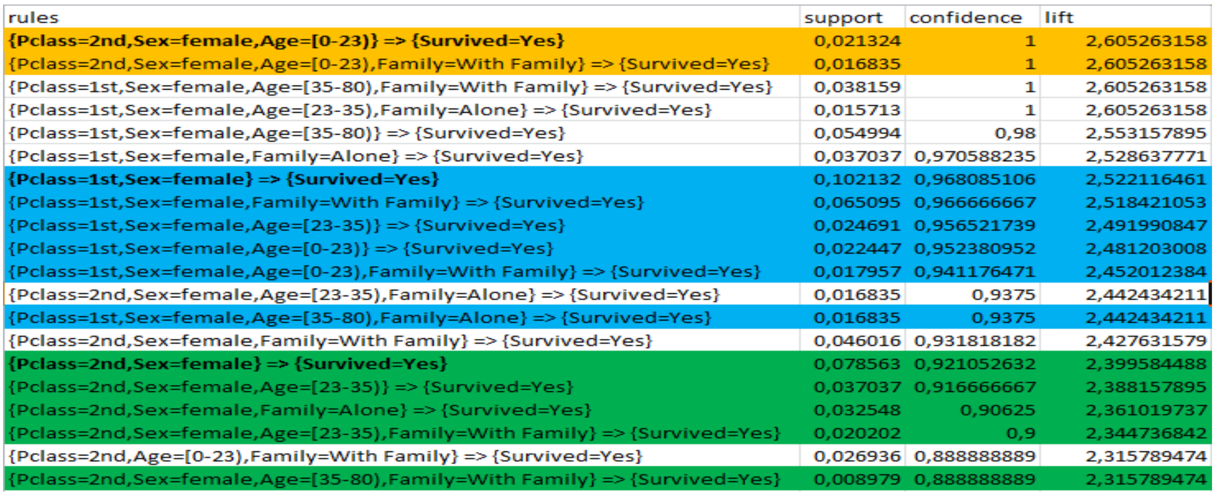
Therefore, best value of minimum support and confidence are considered 0.5% and 50% in turn.



## 3.3.1 Removing Redundancy

Some rules do not provide extra knowledge as other rules already contain the information. For example, if there is the rule in the {{Pclass=2nd,Sex=female,Age=[0-23)} => {Survived=Yes} then the rule{Pclass=2nd,Sex=female,Age=[0-23),Family=With Family} => {Survived=Yes} is not so informative. Generally speaking, when a rule is a super rule of another rule and the former has the same or a lower lift, the former rule is considered to be redundant. For this analysis it is thought that when the consequent is equal to “Survived=Yes/No” or “Pclass=1st/2nd/3rd”, the rules have more meaningful. Therefore, redundancy filter is applied for these consequents.

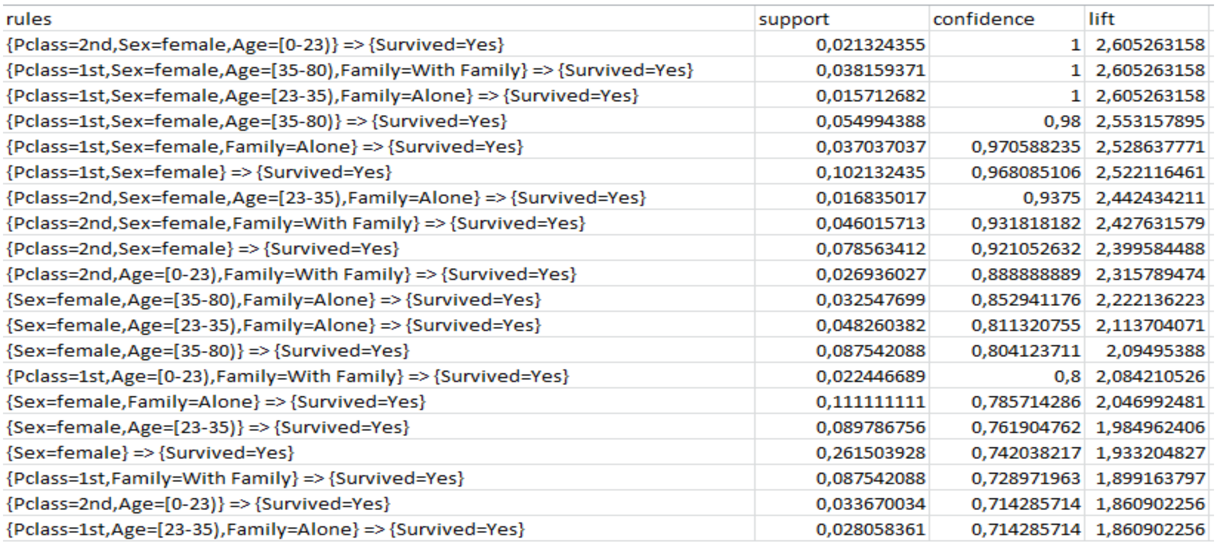
The Figure 21 shows first 20 rules when consequent is equal to “Survived=Yes/No”. Super rules are labeled with bold and former rules have same color line with their super rule. So, they will be redundant.



| rules | support | confidence | lift |
|---|---|---|---|
| **{Pclass=2nd,Sex=female,Age=[0-23)} => {Survived=Yes}** | 0,021324 | 1 | 2,605263158 |
| {Pclass=2nd,Sex=female,Age=[0-23),Family=With Family} => {Survived=Yes} | 0,016835 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[35-80),Family=With Family} => {Survived=Yes} | 0,038159 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[23-35),Family=Alone} => {Survived=Yes} | 0,015713 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[35-80)} => {Survived=Yes} | 0,054994 | 0,98 | 2,553157895 |
| {Pclass=1st,Sex=female,Family=Alone} => {Survived=Yes} | 0,037037 | 0,970588235 | 2,528637771 |
| **{Pclass=1st,Sex=female} => {Survived=Yes}** | 0,102132 | 0,968085106 | 2,522116461 |
| {Pclass=1st,Sex=female,Family=With Family} => {Survived=Yes} | 0,065095 | 0,966666667 | 2,518421053 |
| {Pclass=1st,Sex=female,Age=[23-35)} => {Survived=Yes} | 0,024691 | 0,956521739 | 2,491990847 |
| {Pclass=1st,Sex=female,Age=[0-23)} => {Survived=Yes} | 0,022447 | 0,952380952 | 2,481203008 |
| {Pclass=1st,Sex=female,Age=[0-23),Family=With Family} => {Survived=Yes} | 0,017957 | 0,941176471 | 2,452012384 |
| {Pclass=2nd,Sex=female,Age=[23-35),Family=Alone} => {Survived=Yes} | 0,016835 | 0,9375 | 2,442434211 |
| {Pclass=1st,Sex=female,Age=[35-80),Family=Alone} => {Survived=Yes} | 0,016835 | 0,9375 | 2,442434211 |
| {Pclass=2nd,Sex=female,Family=With Family} => {Survived=Yes} | 0,046016 | 0,931818182 | 2,427631579 |
| **{Pclass=2nd,Sex=female} => {Survived=Yes}** | 0,078563 | 0,921052632 | 2,399584488 |
| {Pclass=2nd,Sex=female,Age=[23-35)} => {Survived=Yes} | 0,037037 | 0,916666667 | 2,388157895 |
| {Pclass=2nd,Sex=female,Family=Alone} => {Survived=Yes} | 0,032548 | 0,90625 | 2,361019737 |
| {Pclass=2nd,Sex=female,Age=[23-35),Family=With Family} => {Survived=Yes} | 0,020202 | 0,9 | 2,344736842 |
| {Pclass=2nd,Age=[0-23),Family=With Family} => {Survived=Yes} | 0,026936 | 0,888888889 | 2,315789474 |
| {Pclass=2nd,Sex=female,Age=[35-80),Family=With Family} => {Survived=Yes} | 0,008979 | 0,888888889 | 2,315789474 |

Figure 21-Rules with Redundancy

The Figure 22 shows first 20 rules when redundant rules are extracted from the rules.



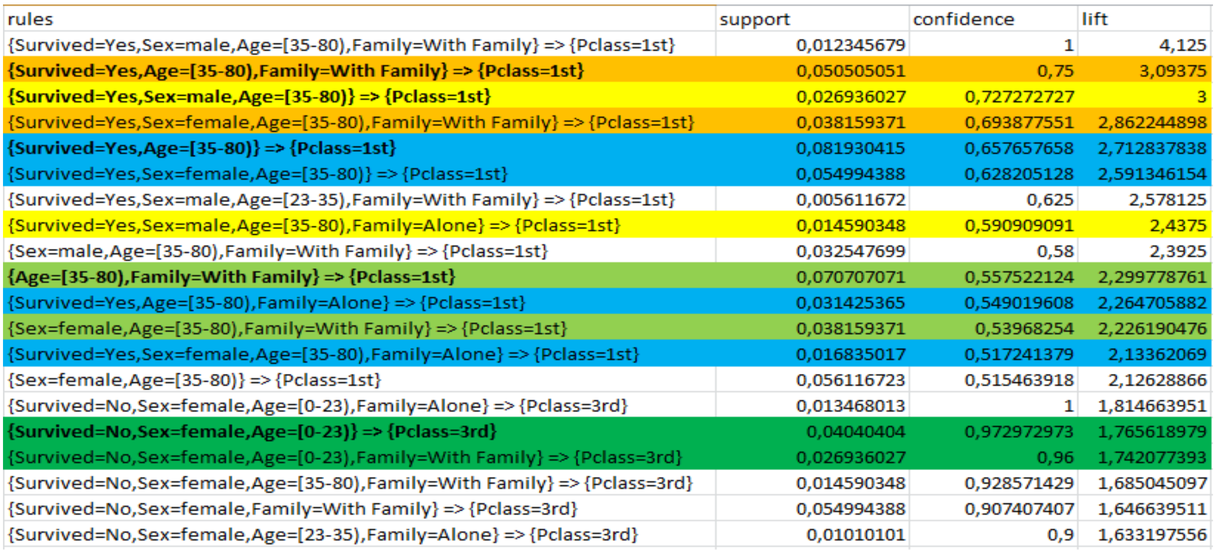
| rules | support | confidence | lift |
|---|---|---|---|
| {Pclass=2nd,Sex=female,Age=[0-23)} => {Survived=Yes} | 0,021324355 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[35-80),Family=With Family} => {Survived=Yes} | 0,038159371 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[23-35),Family=Alone} => {Survived=Yes} | 0,015712682 | 1 | 2,605263158 |
| {Pclass=1st,Sex=female,Age=[35-80)} => {Survived=Yes} | 0,054994388 | 0,98 | 2,553157895 |
| {Pclass=1st,Sex=female,Family=Alone} => {Survived=Yes} | 0,037037037 | 0,970588235 | 2,528637771 |
| {Pclass=1st,Sex=female} => {Survived=Yes} | 0,102132435 | 0,968085106 | 2,522116461 |
| {Pclass=2nd,Sex=female,Age=[23-35),Family=Alone} => {Survived=Yes} | 0,016835017 | 0,9375 | 2,442434211 |
| {Pclass=2nd,Sex=female,Family=With Family} => {Survived=Yes} | 0,046015713 | 0,931818182 | 2,427631579 |
| {Pclass=2nd,Sex=female} => {Survived=Yes} | 0,078563412 | 0,921052632 | 2,399584488 |
| {Pclass=2nd,Age=[0-23),Family=With Family} => {Survived=Yes} | 0,026936027 | 0,888888889 | 2,315789474 |
| {Sex=female,Age=[35-80),Family=Alone} => {Survived=Yes} | 0,032547699 | 0,852941176 | 2,222136223 |
| {Sex=female,Age=[23-35),Family=Alone} => {Survived=Yes} | 0,048260382 | 0,811320755 | 2,113704071 |
| {Sex=female,Age=[35-80)} => {Survived=Yes} | 0,087542088 | 0,804123711 | 2,09495388 |
| {Pclass=1st,Age=[0-23),Family=With Family} => {Survived=Yes} | 0,022446689 | 0,8 | 2,084210526 |
| {Sex=female,Family=Alone} => {Survived=Yes} | 0,111111111 | 0,785714286 | 2,046992481 |
| {Sex=female,Age=[23-35)} => {Survived=Yes} | 0,089786756 | 0,761904762 | 1,984962406 |
| {Sex=female} => {Survived=Yes} | 0,261503928 | 0,742038217 | 1,933204827 |
| {Pclass=1st,Family=With Family} => {Survived=Yes} | 0,087542088 | 0,728971963 | 1,899163797 |
| {Pclass=2nd,Age=[0-23)} => {Survived=Yes} | 0,033670034 | 0,714285714 | 1,860902256 |
| {Pclass=1st,Age=[23-35),Family=Alone} => {Survived=Yes} | 0,028058361 | 0,714285714 | 1,860902256 |

Figure 22-Rules without Redundancy

Meaningful knowledge extracted from the rules that based on first analysis:

1) {Pclass=2nd, Sex=female,Age=[0-23)} => {Survived=Yes} with rule confidence=100%. It has the highest lift value so the rule of antecedent and consequent are positive correlated. According to rule, they have 100% survival rate who are in 2nd class,female and child(or teenager). There are same situation in {Pclass=1st,Sex=female,Age=[35-80),Family=With Family} => {Survived=Yes} and {Pclass=1st,Sex=female,Age=[23-35),Family=Alone} => {Survived=Yes} with rule confidence=100%.
2) When the rule {Pclass=1st, Sex=female} => {Survived=Yes} comparing with {Pclass=2nd,Sex=female} => {Survived=Yes}, the first one have high confidence than the second. Therefore, it can be said that the women are in the first class have a more chance than the second class to be survival.
3) {Pclass=2nd,Age=[0-23)} => {Survived =Yes} with the confidence=71%. It has higher confidence than {Pclass=1st, Age=[0-23)} => {Survived=Yes} . It means that child (or teenager) in the second class has a higher survival rate than the first class. It is interesting because, the estimation is contrast.
4) The rules {Pclass=2nd,Sex=male,Age=[35-80),Family=With Family} => {Survived=No} and {Pclass=3rd,Sex=male,Age=[35-80),Family=With Family} => {Survived=No} have confidence=100%. It shows that nobody who provides these antecedents are survival.
5) The rule {Age=[0-23)} => {Survived=No} has a lift(0.97)<1, so there is slight negative correlation between two items.

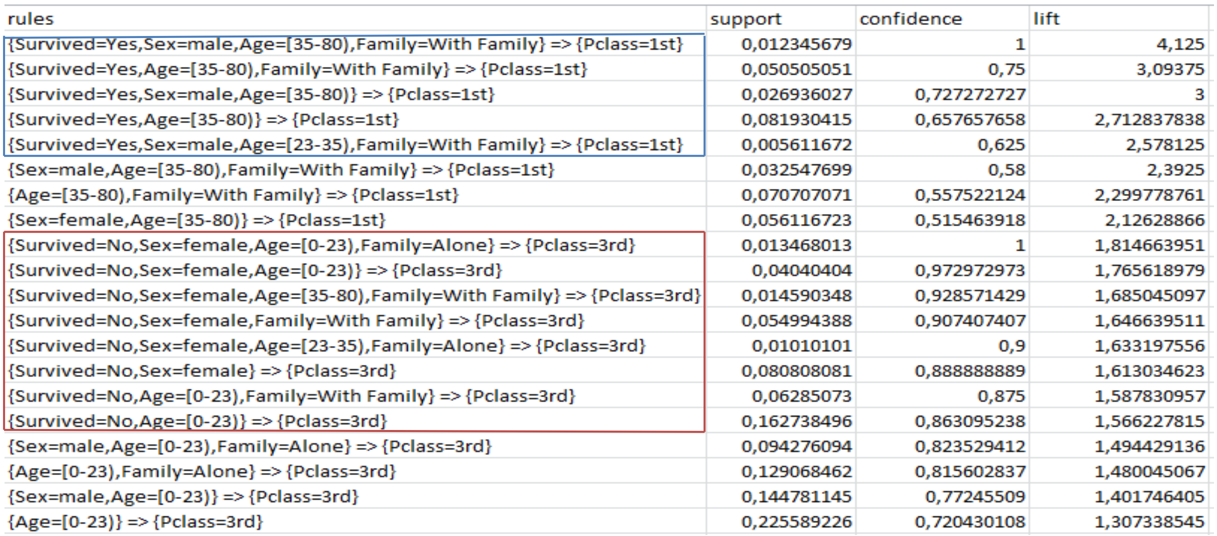
The Figure 23 demonstrates same analysis for the “Pclass=1st/2nd/3rd” that are at consequent side (rhs).



| rules | support | confidence | lift |
|---|---|---|---|
| {Survived=Yes,Sex=male,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,012345679 | 1 | 4,125 |
| **{Survived=Yes,Age=[35-80),Family=With Family} => {Pclass=1st}** | 0,050505051 | 0,75 | 3,09375 |
| **{Survived=Yes,Sex=male,Age=[35-80)} => {Pclass=1st}** | 0,026936027 | 0,727272727 | 3 |
| {Survived=Yes,Sex=female,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,038159371 | 0,693877551 | 2,862244898 |
| **{Survived=Yes,Age=[35-80)} => {Pclass=1st}** | 0,081930415 | 0,657657658 | 2,712837838 |
| {Survived=Yes,Sex=female,Age=[35-80)} => {Pclass=1st} | 0,054994388 | 0,628205128 | 2,591346154 |
| {Survived=Yes,Sex=male,Age=[23-35),Family=With Family} => {Pclass=1st} | 0,005611672 | 0,625 | 2,578125 |
| {Survived=Yes,Sex=male,Age=[35-80),Family=Alone} => {Pclass=1st} | 0,014590348 | 0,590909091 | 2,4375 |
| {Sex=male,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,032547699 | 0,58 | 2,3925 |
| **{Age=[35-80),Family=With Family} => {Pclass=1st}** | 0,070707071 | 0,557522124 | 2,299778761 |
| {Survived=Yes,Age=[35-80),Family=Alone} => {Pclass=1st} | 0,031425365 | 0,549019608 | 2,264705882 |
| {Sex=female,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,038159371 | 0,53968254 | 2,226190476 |
| {Survived=Yes,Sex=female,Age=[35-80),Family=Alone} => {Pclass=1st} | 0,016835017 | 0,517241379 | 2,13362069 |
| {Sex=female,Age=[35-80)} => {Pclass=1st} | 0,056116723 | 0,515463918 | 2,12628866 |
| {Survived=No,Sex=female,Age=[0-23),Family=Alone} => {Pclass=3rd} | 0,013468013 | 1 | 1,814663951 |
| **{Survived=No,Sex=female,Age=[0-23)} => {Pclass=3rd}** | 0,04040404 | 0,972972973 | 1,765618979 |
| {Survived=No,Sex=female,Age=[0-23),Family=With Family} => {Pclass=3rd} | 0,026936027 | 0,96 | 1,742077393 |
| {Survived=No,Sex=female,Age=[35-80),Family=With Family} => {Pclass=3rd} | 0,014590348 | 0,928571429 | 1,685045097 |
| {Survived=No,Sex=female,Family=With Family} => {Pclass=3rd} | 0,054994388 | 0,907407407 | 1,646639511 |
| {Survived=No,Sex=female,Age=[23-35),Family=Alone} => {Pclass=3rd} | 0,01010101 | 0,9 | 1,633197556 |

Figure 23-Rules with Redundancy (RHS = Pclass)

The Figure 24 shows first 20 rules when redundant rules are extracted from the rules.



| rules | support | confidence | lift |
|---|---|---|---|
| {Survived=Yes,Sex=male,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,012345679 | 1 | 4,125 |
| {Survived=Yes,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,050505051 | 0,75 | 3,09375 |
| {Survived=Yes,Sex=male,Age=[35-80)} => {Pclass=1st} | 0,026936027 | 0,727272727 | 3 |
| {Survived=Yes,Age=[35-80)} => {Pclass=1st} | 0,081930415 | 0,657657658 | 2,712837838 |
| {Survived=Yes,Sex=male,Age=[23-35),Family=With Family} => {Pclass=1st} | 0,005611672 | 0,625 | 2,578125 |
| {Sex=male,Age=[35-80),Family=With Family} => {Pclass=1st} | 0,032547699 | 0,58 | 2,3925 |
| {Age=[35-80),Family=With Family} => {Pclass=1st} | 0,070707071 | 0,557522124 | 2,299778761 |
| {Sex=female,Age=[35-80)} => {Pclass=1st} | 0,056116723 | 0,515463918 | 2,12628866 |
| {Survived=No,Sex=female,Age=[0-23),Family=Alone} => {Pclass=3rd} | 0,013468013 | 1 | 1,814663951 |
| {Survived=No,Sex=female,Age=[0-23)} => {Pclass=3rd} | 0,04040404 | 0,972972973 | 1,765618979 |
| {Survived=No,Sex=female,Age=[35-80),Family=With Family} => {Pclass=3rd} | 0,014590348 | 0,928571429 | 1,685045097 |
| {Survived=No,Sex=female,Family=With Family} => {Pclass=3rd} | 0,054994388 | 0,907407407 | 1,646639511 |
| {Survived=No,Sex=female,Age=[23-35),Family=Alone} => {Pclass=3rd} | 0,01010101 | 0,9 | 1,633197556 |
| {Survived=No,Sex=female} => {Pclass=3rd} | 0,080808081 | 0,888888889 | 1,613034623 |
| {Survived=No,Age=[0-23),Family=With Family} => {Pclass=3rd} | 0,06285073 | 0,875 | 1,587830957 |
| {Survived=No,Age=[0-23)} => {Pclass=3rd} | 0,162738496 | 0,863095238 | 1,566227815 |
| {Sex=male,Age=[0-23),Family=Alone} => {Pclass=3rd} | 0,094276094 | 0,823529412 | 1,494429136 |
| {Age=[0-23),Family=Alone} => {Pclass=3rd} | 0,129068462 | 0,815602837 | 1,480045067 |
| {Sex=male,Age=[0-23)} => {Pclass=3rd} | 0,144781145 | 0,77245509 | 1,401746405 |
| {Age=[0-23)} => {Pclass=3rd} | 0,225589226 | 0,720430108 | 1,307338545 |

Figure 24 Rules without Redundancy

Remarks from these rules:

1) When the rules {Survived=Yes,Sex=male,Age=[35-80),Family=With Family} => {Pclass=1st} with confidence=100% and {Survived=Yes, ,Sex=male, Age=[23-35),Family=With Family} => {Pclass=1st} with confidence=63% are considered, it can be seen that . the man who are older than 35, they tend to be first class than younger.
2) The borders on the figure show that “Survived” have an important impact on different “Pclass”. If a passenger is survived, he/she tends to be in first class.

On the other hand, unless a passenger is survived, he/she tends to be in third class.

3) The rule {Survived=No} => {Pclass=3rd} with confidence=68% said that 68% of the passenger who is not survived are in the third class.

# 4. Classification

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data.

## 4.1 Pre-Processing

For classification models, “Survived”, “Pclass”, ”Sex”, “Embarked”, “Family” and “Age” attributes are used when decision trees are created. “Fare“ column is caused overfitting on data so this column is eliminated. “Survived” column is used to make predicted class.

Titanic test set consist of 11 attributes without “Survived” attribute and 418 instances that was collected from passenger’s information for testing.

The “Name”, “Ticket”, “Cabin” and “Fare” attributes are deleted from the test data. For a certain analysis, missing values are filled like filling train data set. In the “Age” column there are some missing values. Therefore, balanced distribution technique is applied on it.

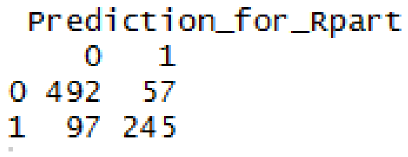
“SibSp” and “Parch” columns are merged again for making a right test. As a result, “Family” is created on test set.

## 4.2 Learning of Different Decision Trees

**Cart Algorithm:** Classification and regression trees (CART) are a non-parametric, binary decision tree learning technique that produces either classification or regression trees, depending on whether the dependent variable is categorical or numeric, respectively[Source: Wikipedia]. In R, Rpart function is used for Cart Modelling.

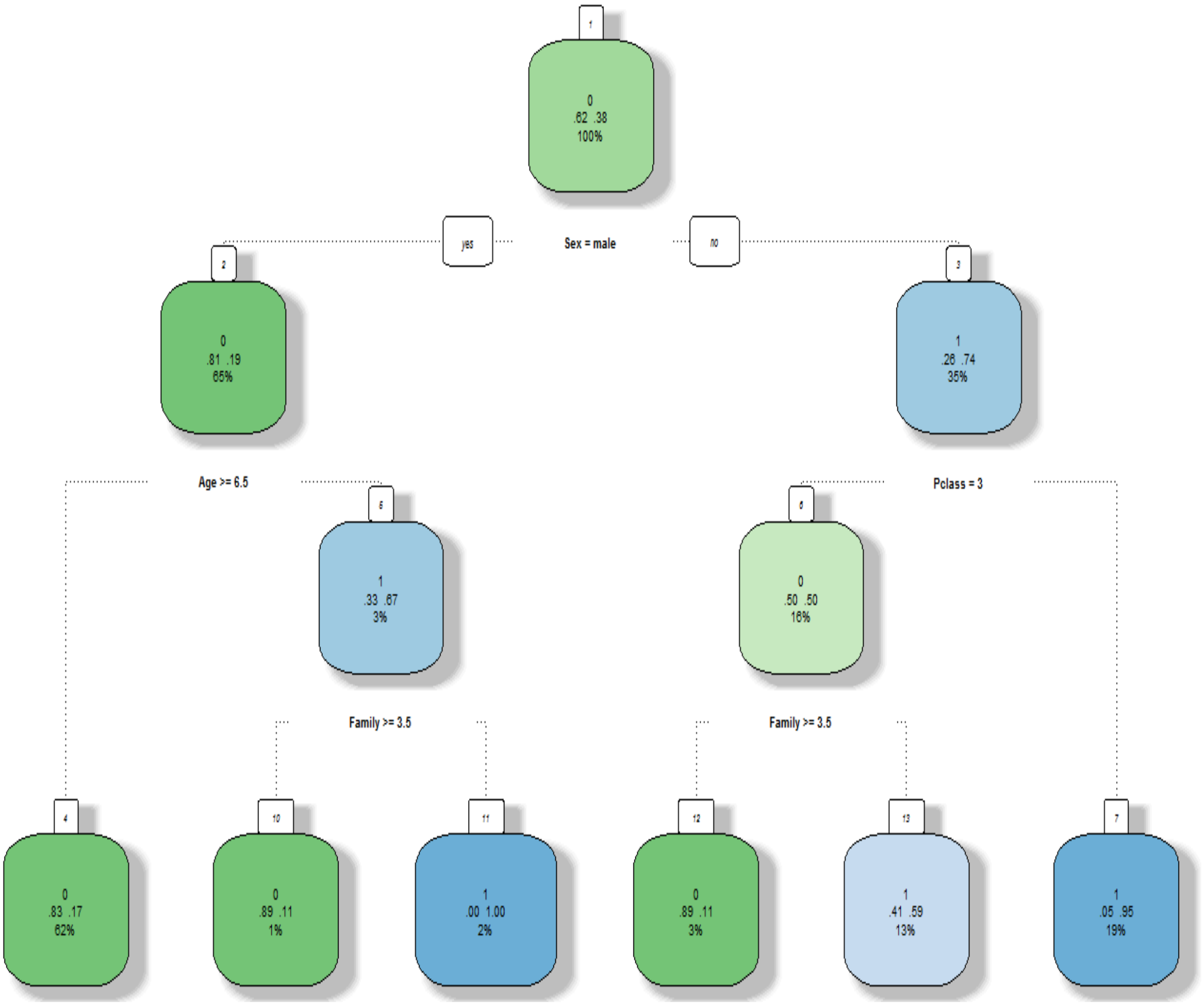
First of all, tree is built from the training set and test on this set, because of comparing the accuracy with cross-validation.

Confusion matrix on training set and decision tree;



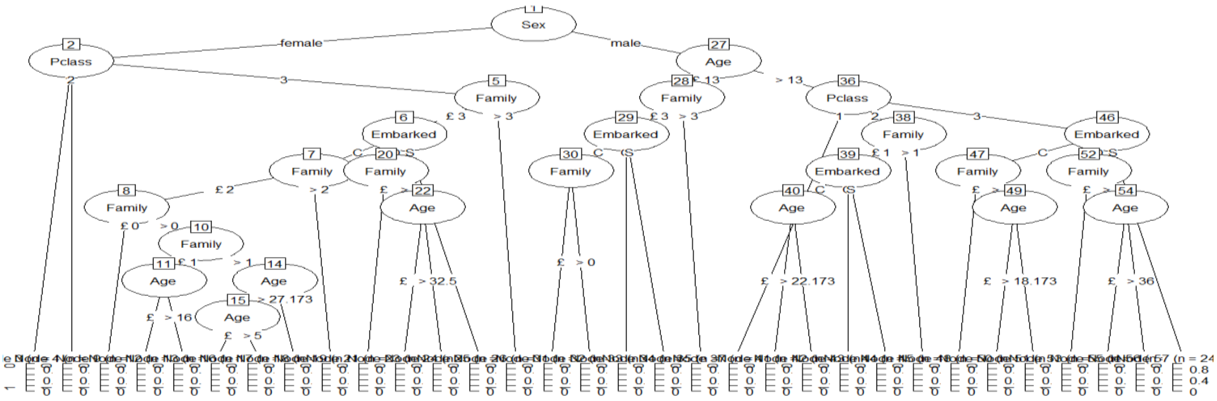
```
  Prediction_for_Rpart
      0   1
0 492  57
1  97 245
```

with accuracy= %82,72



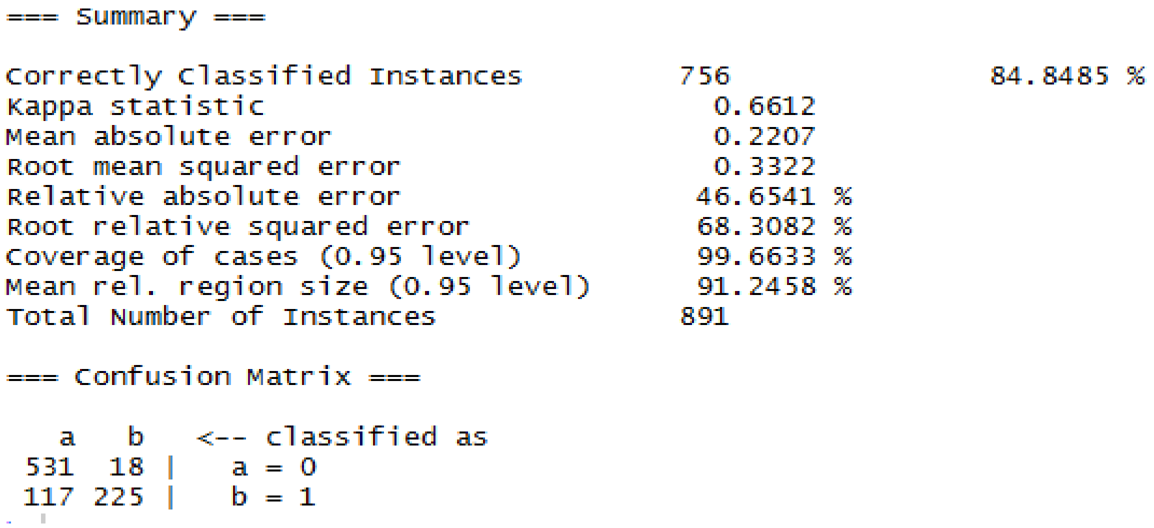
**C4.5 Algorithm:** C4.5 is an extension of Quinlan's earlier ID3 algorithm**.** It builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy [Source: Wikipedia]. For this model, J48 is used in R.

Decision tree and summary of J48 modeling;



The number of leaves are: 32

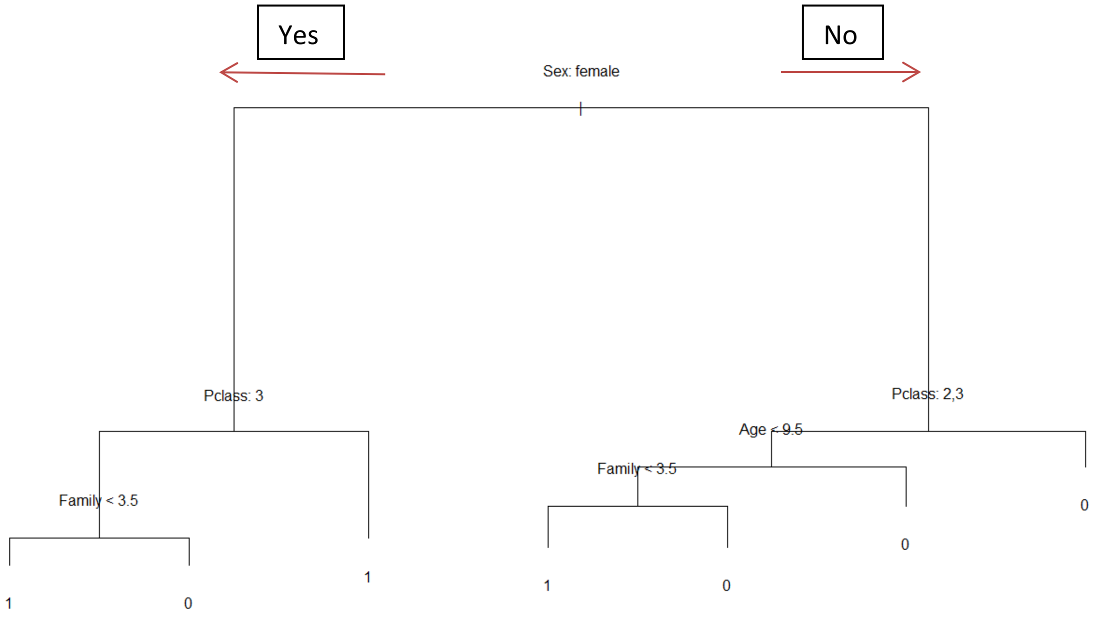
Size of the three is: 57

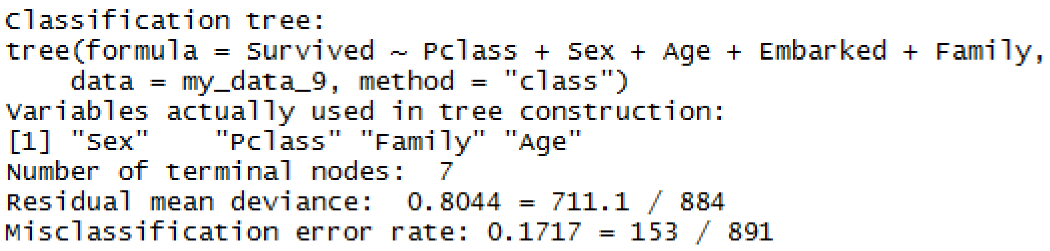


```
=== Summary ===

Correctly Classified Instances         756               84.8485 %
Kappa statistic                          0.6612
Mean absolute error                      0.2207
Root mean squared error                  0.3322
Relative absolute error                 46.6541 %
Root relative squared error             68.3082 %
Coverage of cases (0.95 level)          99.6633 %
Mean rel. region size (0.95 level)      91.2458 %
Total Number of Instances              891

=== Confusion Matrix ===

   a   b   <-- classified as
 531  18 |   a = 0
 117 225 |   b = 1
```

**Tree Function in R:** A tree is grown by binary recursive partitioning using the response in the specified formula and choosing splits from the terms of the right-hand-side. Numeric variables are divided into X<a and X>a; the levels of an unordered factor are divided into two non-empty groups. The split which maximizes the reduction in impurity is chosen, the data set split and the process repeated. Splitting continues until the terminal nodes are too small or too few to be split.

Decision tree, and summary;

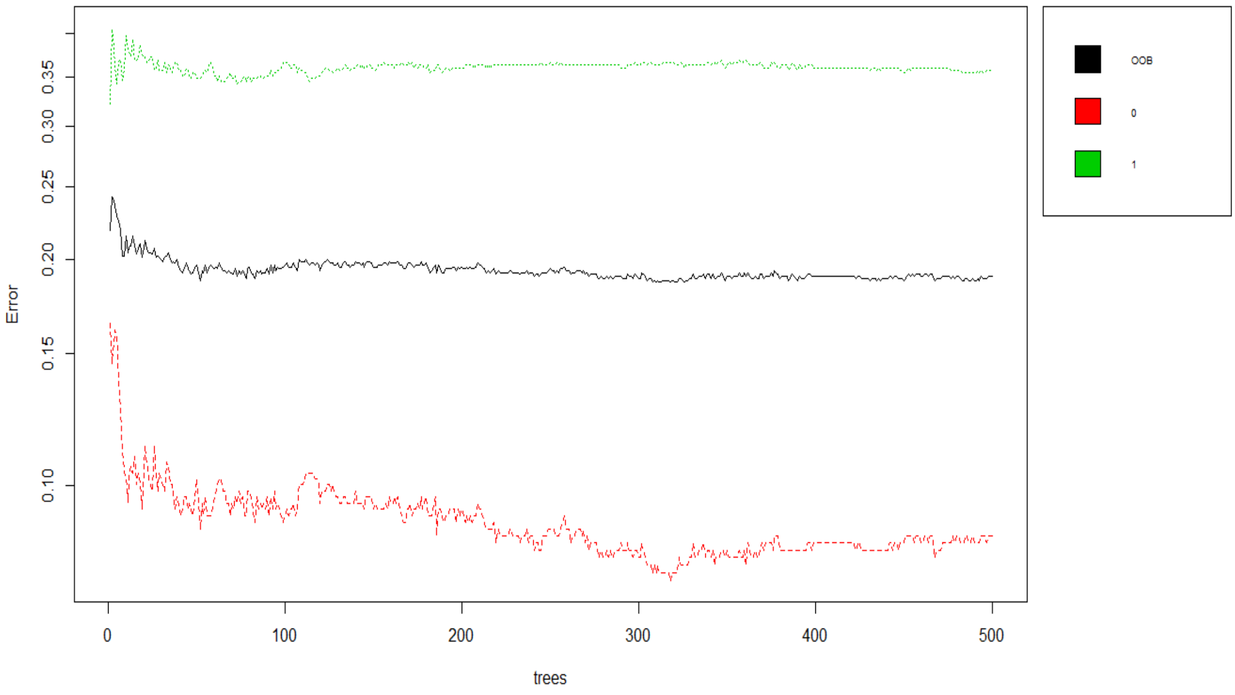




```
Classification tree:
tree(formula = Survived ~ Pclass + Sex + Age + Embarked + Family,
    data = my_data_9, method = "class")
Variables actually used in tree construction:
[1] "Sex"    "Pclass" "Family" "Age"
Number of terminal nodes:  7
Residual mean deviance:  0.8044 = 711.1 / 884
Misclassification error rate: 0.1717 = 153 / 891
```

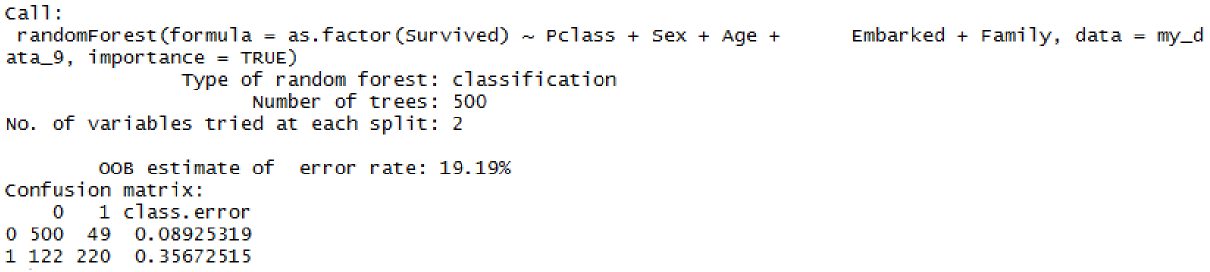
**Random Forest Algorithm:** Random forest is very good in that it is an ensemble learning method used for classification. It uses multiple models for better performance that just using a single tree model.  In addition because many sample are selected in the process a measure of variable importance can be obtain and this approach can be used for model selection and can be particularly useful when forward/backward stepwise selection is not appropriate and when working with an extremely high number of candidate variables that need to be reduced. For modelling with this approach, randomForest function is used in R.

Error of modeling trees obtained by randomForest;



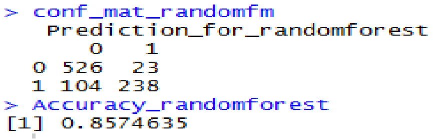
Out-of-bag estimation for the generalization error that is shown by black line is the error rate of the out-of-bag classifier on the training set.

Summary of random forest is shown in below, the more number of tree is (parameter of function like 500 in summary), the lower OBB error rate is.



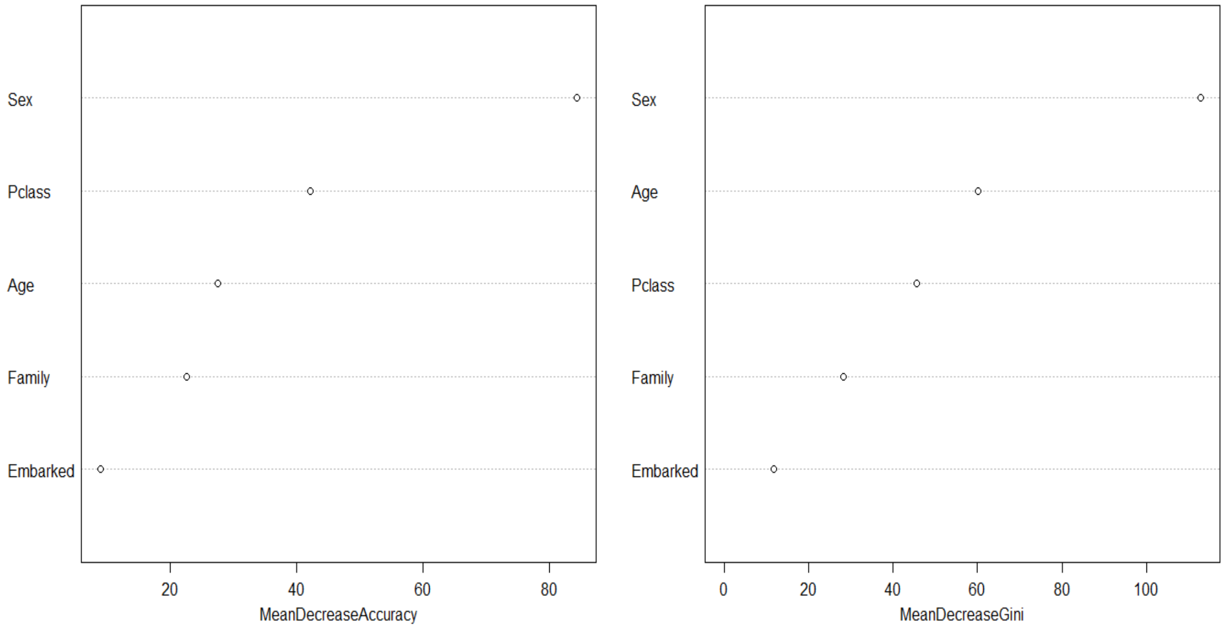
```
Call:
 randomForest(formula = as.factor(Survived) ~ Pclass + Sex + Age +       Embarked + Family, data = my_d
ata_9, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 19.19%
Confusion matrix:
    0   1 class.error
0 500  49  0.08925319
1 122 220  0.35672515
```

Prediction on all training data:



```
> conf_mat_randomfm
  Prediction_for_randomforest
      0   1
  0 526  23
  1 104 238
> Accuracy_randomforest
[1] 0.8574635
```

Importance of variables;



There are two types of importance measures shown above. The accuracy one tests to see how worse the model performs without each variable, so a high decrease in accuracy would be expected for very predictive variables. The Gini one digs into the mathematics behind decision trees, but essentially measures how pure the nodes are at the end of the tree.

## 4.3 Decision Trees Validation and Interpretation

### For the tree obtained by Rpart functions;

Rpart function uses prune method based on minimize the cross-validation error. In order to minimize this error, the complexity-xerror-size graph is analyzed by plotcp(). Also, printcp() provides a summary of this graph.

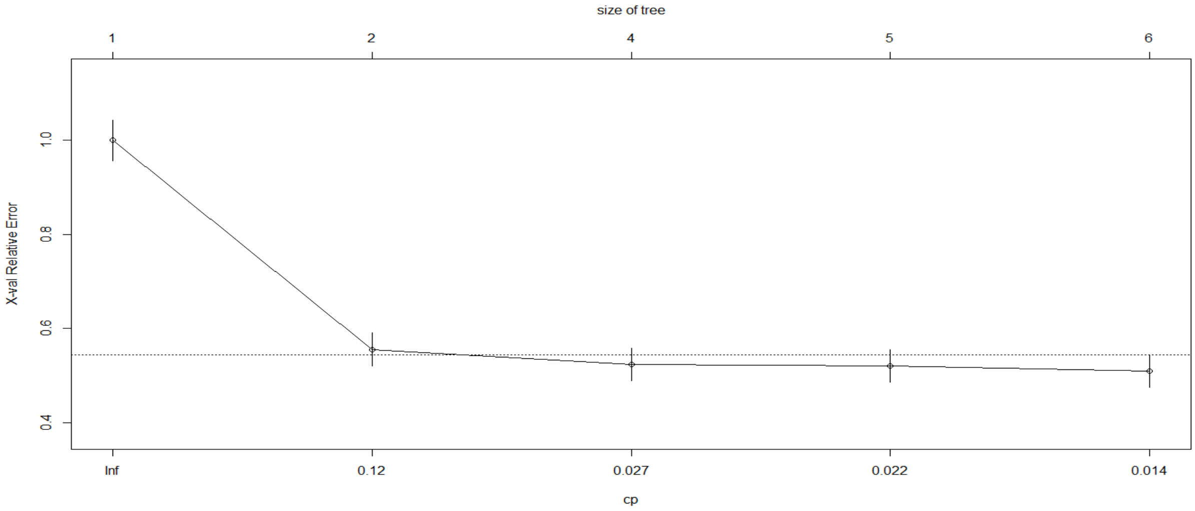


Figure 25-Complexity-Error-Size

According to the Figure 25, it is easily seem that the tree has to pruned by the complexity value 0,014. When making it, the pruned tree is same with unpruned.

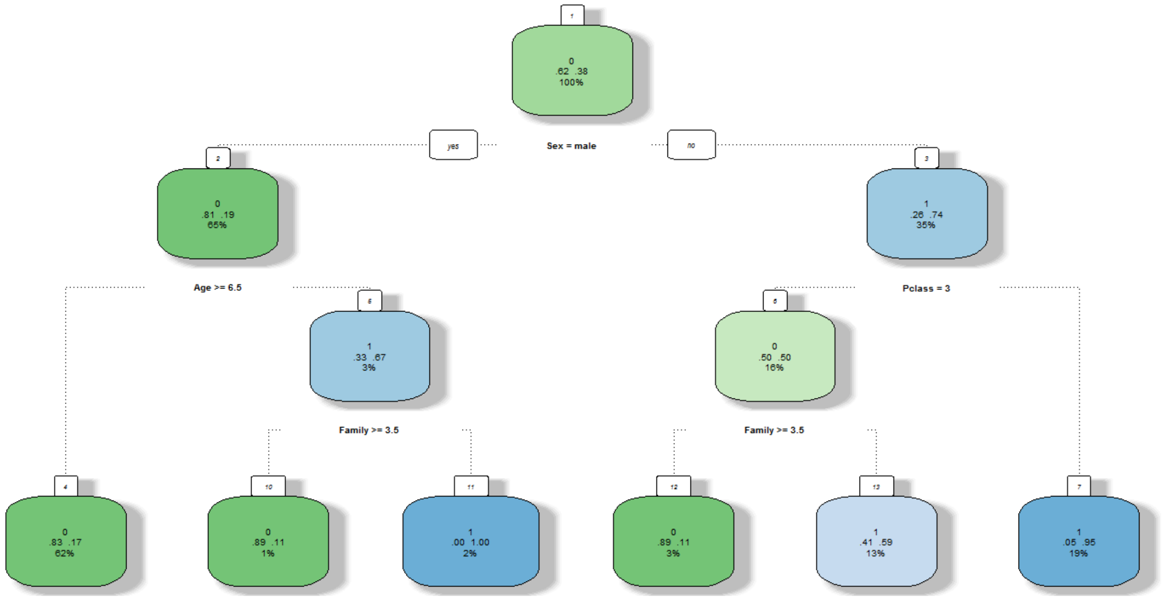
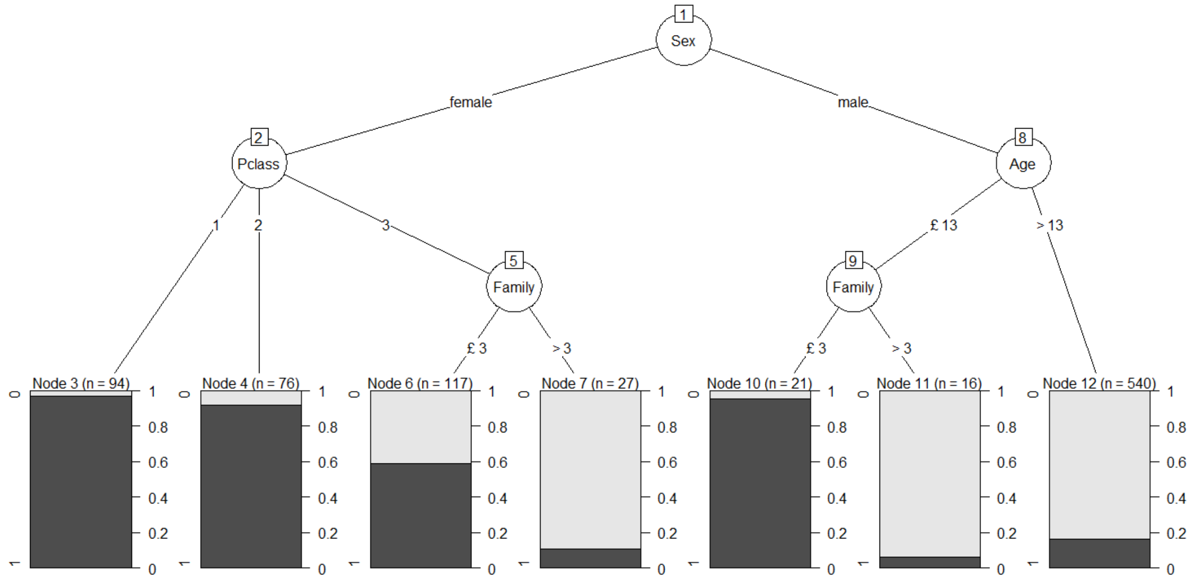


Figure 26-Decision Tree with Minimum Error

The Figure 26 shows that tree takes minimum error when it is split to 6 nodes.

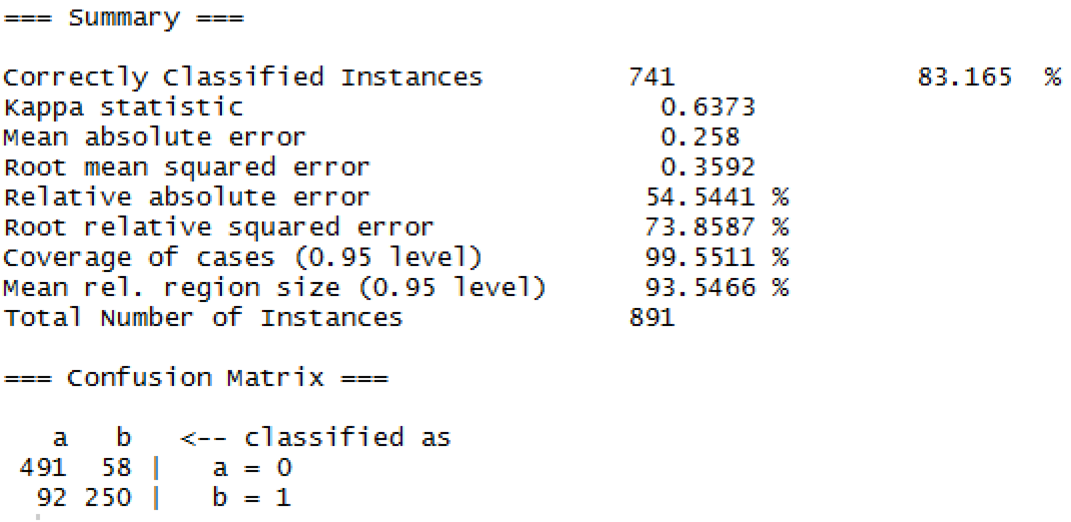
**Another tree obtained by J48;**

In the previous section, unpruned tree is obtained by parameter of J48(). It can be seen easily that the tree will be caused over-fitting so on the test data will have more error.  For pruning tree, the function's parameter is changed with the prune one. As a result, below tree is obtained;



The number of leaves are: 7

Size of the tree is: 12

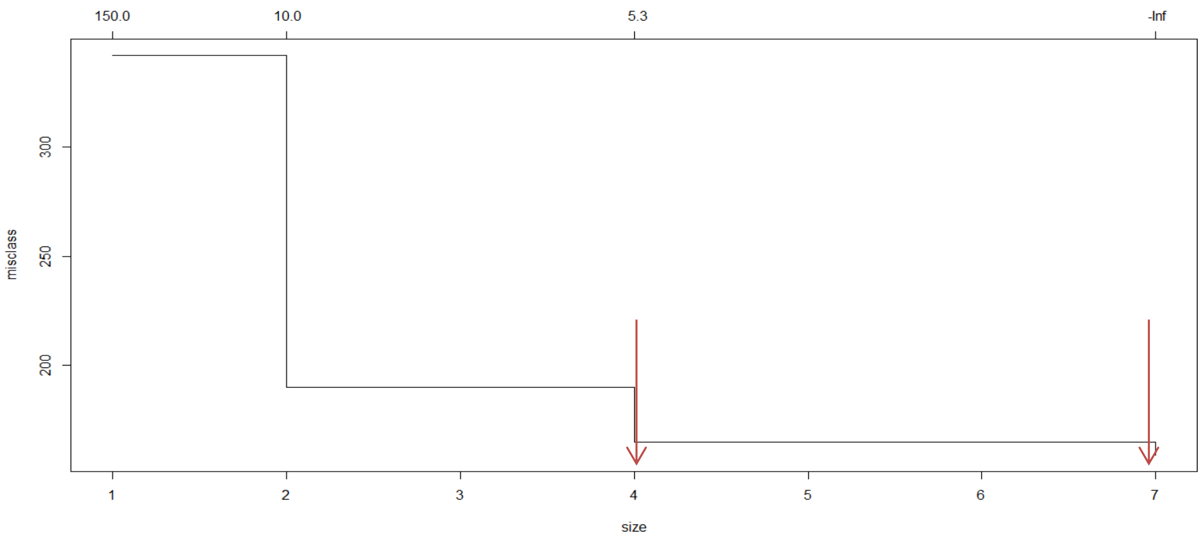


```
=== Summary ===

Correctly Classified Instances         741               83.165  %
Kappa statistic                          0.6373
Mean absolute error                      0.258
Root mean squared error                  0.3592
Relative absolute error                 54.5441 %
Root relative squared error             73.8587 %
Coverage of cases (0.95 level)          99.5511 %
Mean rel. region size (0.95 level)      93.5466 %
Total Number of Instances              891

=== Confusion Matrix ===

   a   b   <-- classified as
 491  58 |   a = 0
  92 250 |   b = 1
```

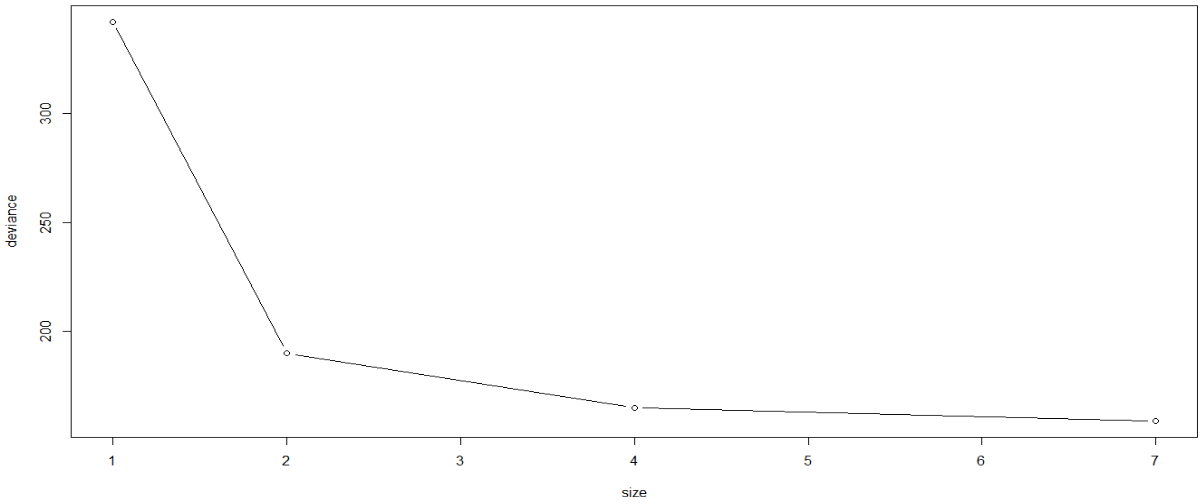
It seems that unpruned version of J48 tree has higher accuracy; however there is an over-fitting and it has 32 leaves and 51 nodes. Therefore, pruned tree is more robust.

**Another tree obtained by tree() in R;**

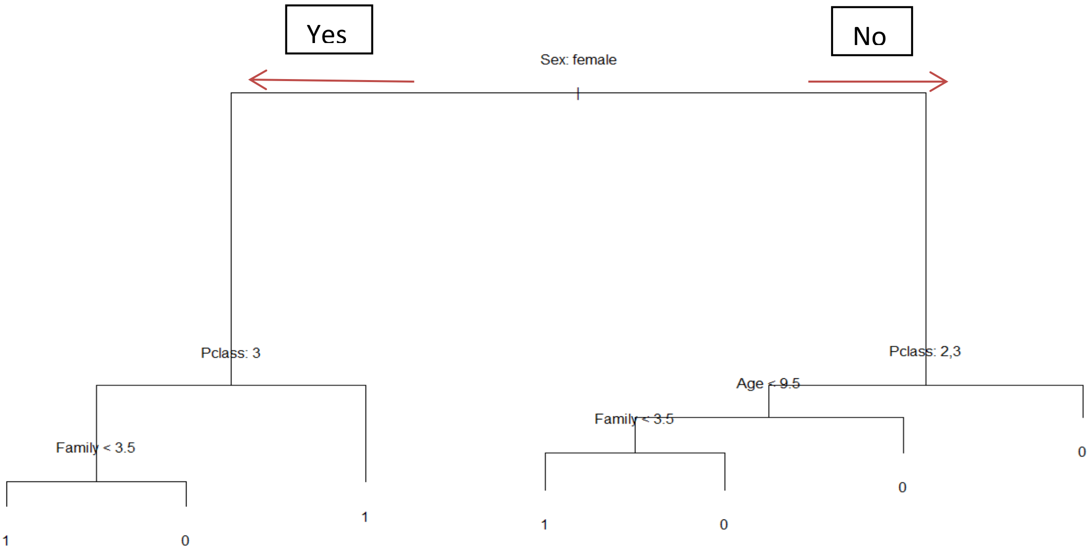
It may need “pruning” to avoid over-fitting on test data. For pruning, cv.tree() is used with the parameter “prune.missclass” and then plot() for result of cv.tree() provides in the below plot ;



The lower X axis is the number of terminal nodes and the upper X axis is the number of folds (# of pieces the data is split) in the cross validation. It shows how the misclassification error varies against these. So, this plot is very useful in determining the optimal number of terminal nodes at which the decision tree should be pruned. The two red lines mark the two options (# terminal nodes) to prune the data. Ideally, it is best keep the tree as simple as possible (lesser number of nodes) and the misclassification error as low as possible. The first choice is 4 but when comparing with 7, 7 is lower error than 4. At the same time, it can be seen that 7 is the best for deviance in the below plot.



As a result, pruned tree is generated. Comparing with unpruned version of tree, it is seen that they are same. Since, the lowest error is obtained when the tree reaches the biggest number of terminal nodes.



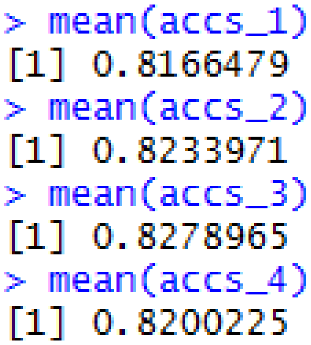
**Another tree obtained by randomForest() in R;**

For randomForest function, there is not pruning. Each tree is grown to the largest extent possible.

**K-Fold Cross Validation**

Cross Validation is one of the most important concepts in any type of data modeling. It simply says, try to leave a sample on which you do not train the model and test the model on this sample before finalizing the model.

For learning more robust accuracies, “7-Fold Cross Validation” is applied on the data. Each decision tree models with pruned one in the previous sections are used in the cross validation step. The accuracies are :

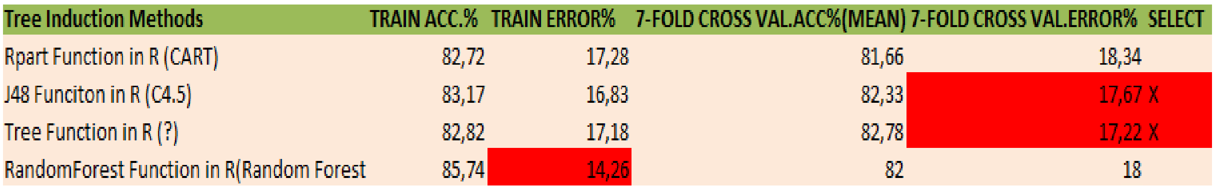


```
> mean(accs_1)
[1] 0.8166479
> mean(accs_2)
[1] 0.8233971
> mean(accs_3)
[1] 0.8278965
> mean(accs_4)
[1] 0.8200225
```

1. Rpart()
2. J48()
3. Tree()
4. RandomForest()

All of them have lower accuracies than on training. Predictions on all training data are used in the first part of classification and so error rates are generated. Now, cross-validation errors are obtained for each one. Cross-Validation is also used for selection algorithms. Therefore, training errors and cross validation errors are compared to choose best algorithm.

According to the Figure 27, the lowest training error is Random Forest but also it has not lowest cross-validation errors. Most important effect on selection of models or algorithms is that they have lowest cross-validation errors. As a result, J48() and tree() are chosen to generate best decision trees. But all of methods have only slight difference between each other. They may have same accuracies on real test set.

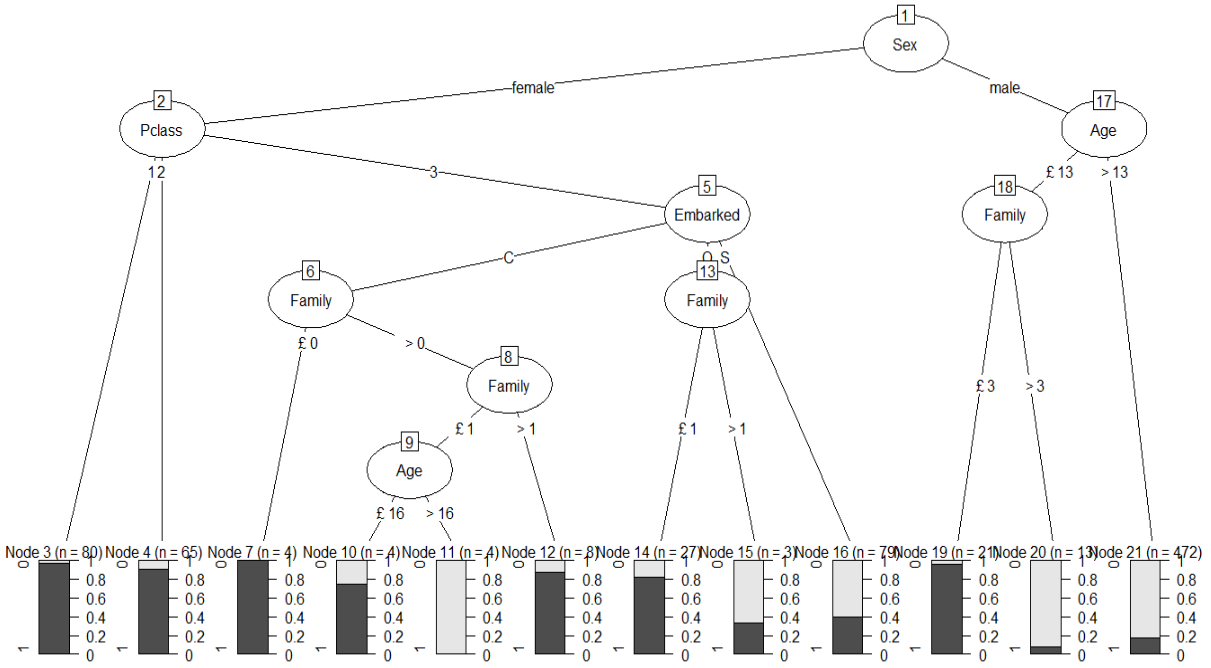


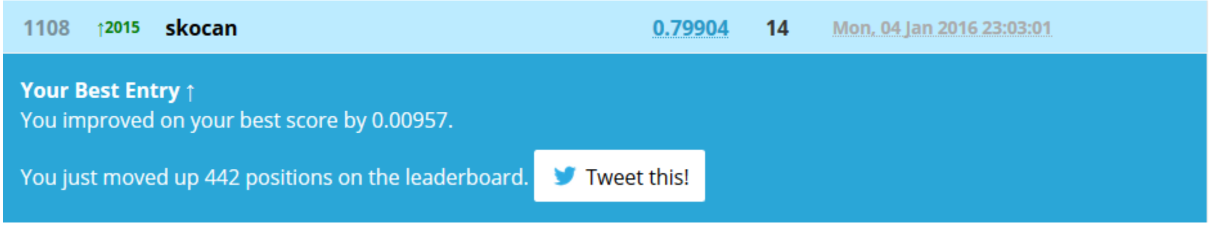
| Tree Induction Methods | TRAIN ACC.% | TRAIN ERROR% | 7-FOLD CROSS VAL.ACC%(MEAN) | 7-FOLD CROSS VAL.ERROR% | SELECT |
|---|---|---|---|---|---|
| Rpart Function in R (CART) | 82,72 | 17,28 | 81,66 | 18,34 | |
| J48 Funciton in R (C4.5) | 83,17 | 16,83 | 82,33 | 17,67 | X |
| Tree Function in R (?) | 82,82 | 17,18 | 82,78 | 17,22 | X |
| RandomForest Function in R(Random Forest | 85,74 | 14,26 | 82 | 18 | |

Figure 27-Accuracies-Errors

## 4.4 Best Decision Trees and Kaggle Submission

As a result of cross-validation, J48() and tree() generates best decision trees. Each model is applied on real test data and sent to Kaggle to make submission. The best score is taken from J48(). Therefore, the best decision tree is shown in below:





1108 ↑2015 skocan 0.79904 14 Mon, 04 Jan 2016 23:03:01

**Your Best Entry ↑**
You improved on your best score by 0.00957.

You just moved up 442 positions on the leaderboard.

Figure 28-Score of Kaggle

The Figure 28 shows that the accuracy of model on test data is almost %80.