[II.2313] Analyse de données – TP5

Rémi Biolley – Thierry Lincoln

A. Preliminary analysis

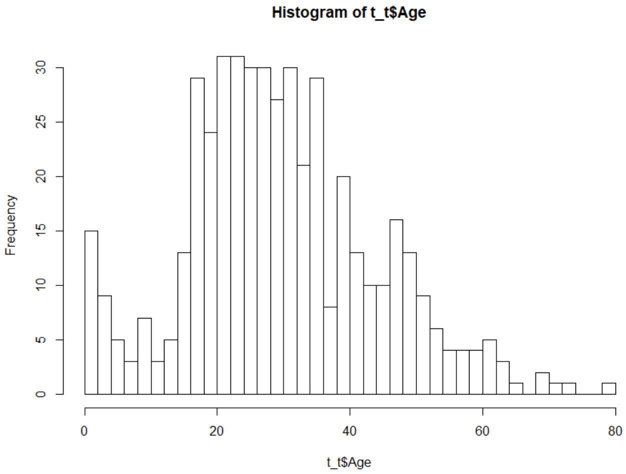
1. Open the "titanic_train.csv". Describe the different attributes

We open our training dataset. We'll use it later to predict the passengers' chances of survival in the test dataset. But, first, we want to understand our dataset in order to anticipate potential problems.

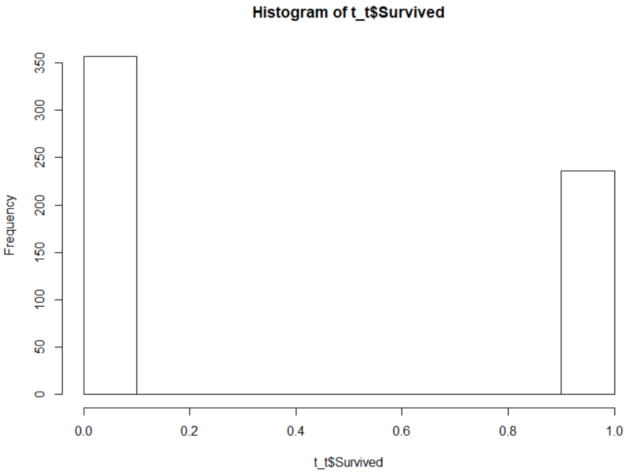
The *str()* function gives us a global vision of the dataset. It is composed of 13 columns. Some are containing numeric/integer values (*PassengerId*, *Survived*, *Pclass*, *Age*, *SibSp*, *Parch* and *Fare*) while the others are containing string values. We also see that there are several NA's values in the *Age* column.

/!\ We notice that some columns are presented as factors and some are not. This will have an importance later /!\

We pursue our study by plotting some histograms:



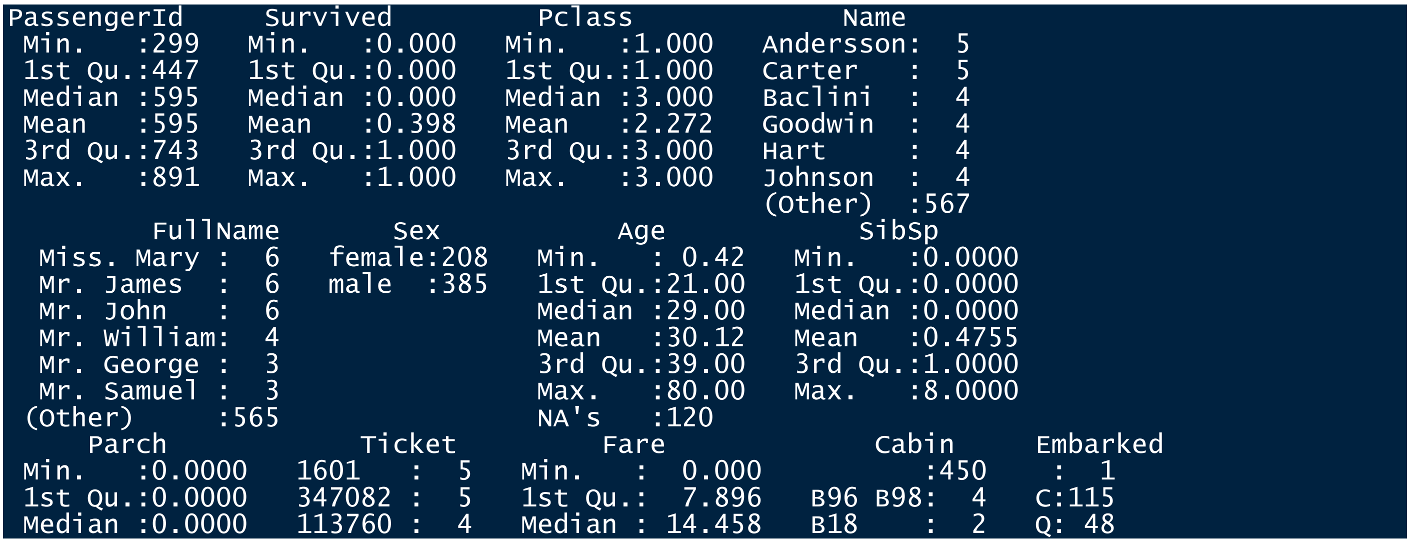
Histogram of t_t$Age



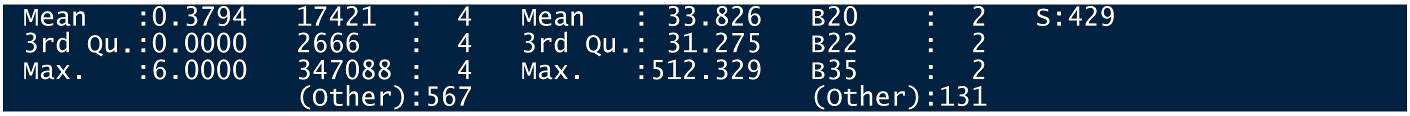
Histogram of t_t$Survived

This allows us to evaluate the consistency of our data. We see that there is a majority of people who are around 25-30 years old, while there are only a few children and elders. It is not shocking. Furthermore, a lot of people didn't survive the accident. That's what we were expecting.

In the end, we use the *summary()* function to have more precise idea of the columns' values:

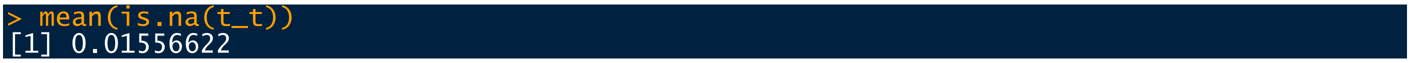


```
PassengerId     Survived          Pclass              Name
 Min.   :299   Min.   :0.000   Min.   :1.000   Andersson:  5
 1st Qu.:447   1st Qu.:0.000   1st Qu.:1.000   Carter   :  5
 Median :595   Median :0.000   Median :3.000   Baclini  :  4
 Mean   :595   Mean   :0.398   Mean   :2.272   Goodwin  :  4
 3rd Qu.:743   3rd Qu.:1.000   3rd Qu.:3.000   Hart     :  4
 Max.   :891   Max.   :1.000   Max.   :3.000   Johnson  :  4
                                               (Other)  :567
         FullName       Sex          Age               SibSp
  Miss. Mary :  6   female:208   Min.   : 0.42   Min.   :0.0000
  Mr. James  :  6   male  :385   1st Qu.:21.00   1st Qu.:0.0000
  Mr. John   :  6                Median :29.00   Median :0.0000
  Mr. William:  4                Mean   :30.12   Mean   :0.4755
  Mr. George :  3                3rd Qu.:39.00   3rd Qu.:1.0000
  Mr. Samuel :  3                Max.   :80.00   Max.   :8.0000
 (Other)     :565                NA's   :120
     Parch            Ticket          Fare             Cabin     Embarked
 Min.   :0.0000   1601   :  5   Min.   :  0.000          :450    :  1
 1st Qu.:0.0000   347082 :  5   1st Qu.:  7.896   B96 B98:  4   C:115
 Median :0.0000   113760 :  4   Median : 14.458   B18    :  2   Q: 48
```



```
 Mean   :0.3794   17421  :  4   Mean   : 33.826   B20    :  2   S:429
 3rd Qu.:0.0000   2666   :  4   3rd Qu.: 31.275   B22    :  2
 Max.   :6.0000   347088 :  4   Max.   :512.329   B35    :  2
                  (Other):567                     (Other):131
```

This function displays 3 interesting information. First, a lot of values are missing in the *Cabin* column (450 values, 76% of them). Second, there are 120 NA's in the *Age* column. Furthermore, 1 value is missing in the *Embarked* column.

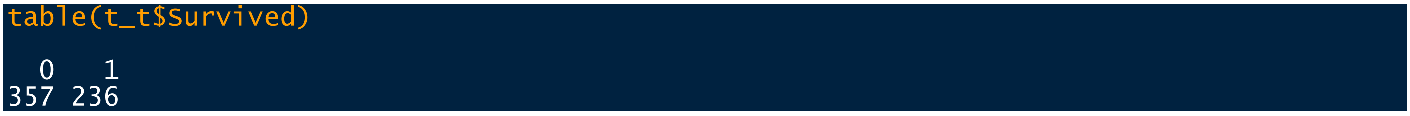


```
> mean(is.na(t_t))
[1] 0.01556622
```

In the end, only 1.5% of the values of the dataset are missing. That may sound a low percentage, but we'll have to deal with them later and will have to decide if we either leave the columns like that or give them new values. Giving a random value to *Embarked* will probably have no impact on our work, but we will have a lot more troubles with *Age* and *Cabin.*

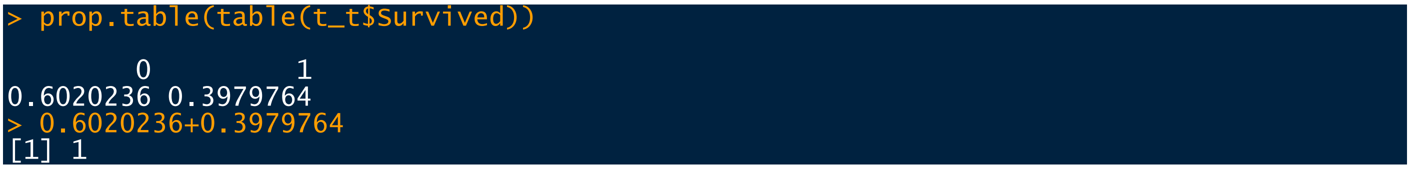
2. How many people died and survived in this data set?

We want to see the proportion of people who died during the accident. We can get the total number of dead and survivors using the *table()* function:



```
table(t_t$Survived)

  0   1 
357 236 
```

We also can directly get the percentages with the *prop.table()* function:



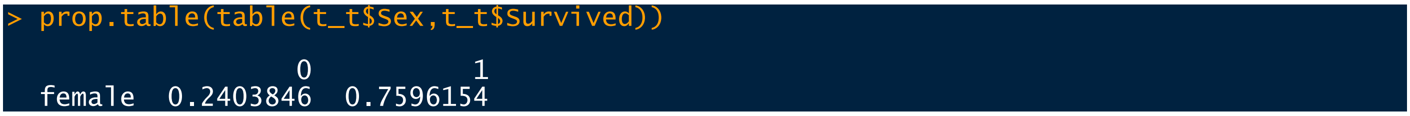
```
> prop.table(table(t_t$Survived))

        0         1 
0.6020236 0.3979764 
> 0.6020236+0.3979764
[1] 1
```

We take the time to verify that the sum is equal to 1.

B. Women and children First!

1. Compare the survival rates between men and women. Comment.

We now desire to evaluate if there is an important difference of chances of survival between women and men:



```
> prop.table(table(t_t$Sex,t_t$Survived))

                 0          1
  female  0.2403846  0.7596154
  male    0.7974026  0.2025974

> table(t_t$Sex,t_t$Survived)

           0   1
  female  50 158
  male   307  78
```

We conclude that it seems like the chances of survival are way higher when you are a female. Indeed, 75% of the women survived while 80% of the men died ...

2. We now want to assess the survival rates depending on the age of the passengers.

Let's try to see if the children had higher chances to survive the accident. First of all, we need to create a new column *Child*. Its lines will either take the value *adult,* if the passenger was above 18 years old, or *child*, if the passenger was under 18 years old.

```
t_t$Child <- ifelse(t_t$Age >18, "adult", "child")
```

We now can see if the children's survival rate compared to adults':

```
> prop.table(table(t_t$Child,t_t$Survived),1)

                0         1
  adult 0.5963542 0.4036458
  child 0.4606742 0.5393258
```

It seems like the children had slightly higher chances of survival. Nevertheless, the difference is way less important than the one we noticed between men and women.

3. What potentially biased hypothesis was considered in the previous question? Propose a solution to fix this problem and re-assess the survival rate accordingly. Are the results very different?

When we created our *Child* column, we didn't take care of the NA's present in the *Age* column. As a result, our column is containing NA's too.

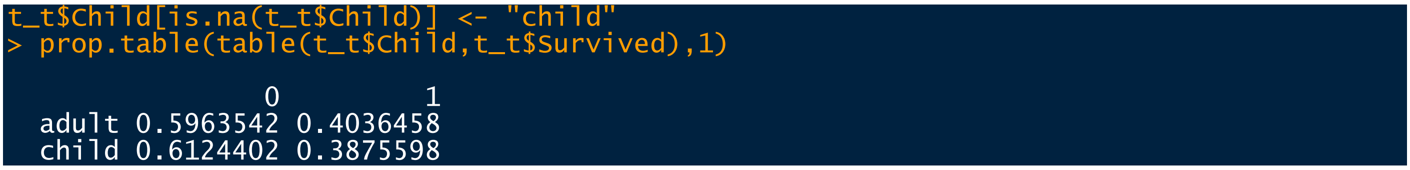
We can evaluate the proportion of missing data in the *Age* column:

```
mean(is.na(t_t$Age))
[1] 0.2023609
```

20% of the values are missing. This is a huge amount and can have a heavy repercussion on our final results. Now, we have to decide if we leave our column in its current state or if we fill in the missing values.

We decide to consider that the NA's are all adults. Then we consider that they are all children. Therefore, we may evaluate which case seems to be the more realistic.

```
t_t$Child[is.na(t_t$Child)] <- "adult"
prop.table(table(t_t$Child,t_t$Survived),1)

                0         1
  adult 0.6269841 0.3730159
  child 0.4606742 0.5393258
```



```
t_t$Child[is.na(t_t$Child)] <- "child"
> prop.table(table(t_t$Child,t_t$Survived),1)

                0         1
  adult 0.5963542 0.4036458
  child 0.6124402 0.3875598
```

Replacing NA's with *child* values has a way higher impact on our result. It even changes our interpretation: the children have a lower chance of survival than the adults.

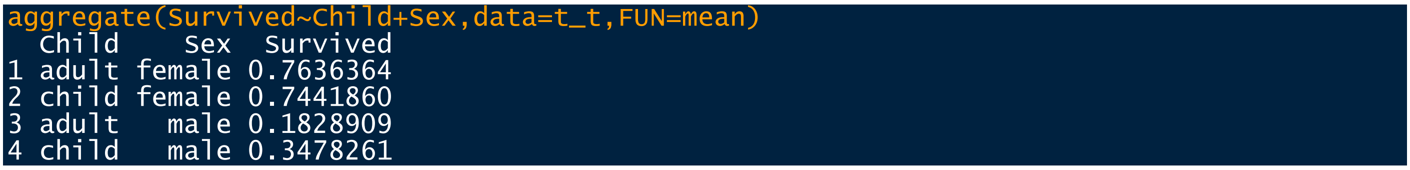
On the other side, replacing NA's with *adult* values gives us a result relatively close to the old one.

In the end, we decide to change the NA's in *adult* values. Indeed, it has a low impact (and realistic impact) on our results. Furthermore, we noticed earlier that there were more adults than children on the ship. Therefore, it is more probable that the NA's are mostly representing adults.

It could be even better to consider replacing NA's with the same repartition as the original children vs adult population on board. And trying to add the Parch (Parent-Children) variable into our new repartition model. Lacking time these need to be further explored.

4. Revert to the "Child" variable from question B.2. Use the fonction aggregate(·) to assess the survival rates of the passengers depending on their age and sex. What can you conclude on the effciency of the "Women and children first" policy during the evacuation of the Titanic ?

The *aggregate()* function allows us to see the probability of survival, depending on the sex **and** the age of the passenger:



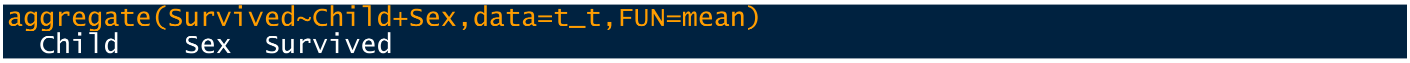
```
aggregate(Survived~Child+Sex,data=t_t,FUN=mean)
  Child    Sex  Survived
1 adult female 0.7636364
2 child female 0.7441860
3 adult   male 0.1828909
4 child   male 0.3478261
```

We get the following values:

- 76% of the adult women survived
- 74% of the girls survived
- 18% of the adult men survived
- 35% of the boys survived

We can draw several conclusions of this result. First of all, the sex of the passenger has a way more important impact on the chances of survival. Then, being a child doesn't seem to have any impact on your chances of survival if you are a woman. Nevertheless, being a child affects your chances of survival if you are a man.

Therefore, we can say that "Women first" policy is efficient. Unfortunately, this isn't the case for the "Children first" policy. Nevertheless, we can wonder if this inefficiency isn't due to threshold that we chose to create the *Child* column. Indeed, if we choose the age of 15 years old, instead of 18, we get the following result:



```
aggregate(Survived~Child+Sex,data=t_t,FUN=mean)
  Child    Sex  Survived
```

```
1 adult female 0.7692308
2 child female 0.6923077
3 adult   male 0.1750000
4 child   male 0.6000000
```

Unfortunately, this result is a little biased as we don’t have enough many children who are under 15.

5. Load the data from the file "titanic_test.csv" and add the "Child" variable as in question B.2. Convert this variable as well as the variable "Survived" into factors using the function as.factor(·). This conversion will prevent any type compatibility issue and should be applied to both data sets.

We now decide to apply the Naive Bayes classification to predict the chances of survival, using the training dataset. Then we will verify the accuracy of our prediction with a test dataset.

We make the same operations on the test dataset as the ones we made on the training dataset.

Also, the predict*()* function, that we will use later, needs to work with factors. Then, we convert our column *Child* into a factor (*Sex* already is a factor).

We apply the *naiveBayes()* function:

```
classifier=naiveBayes(Survived~Child+Sex,data=t_t_af)
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        0         1 
0.6020236 0.3979764 

Conditional probabilities:
   Child
Y       adult     child
  0 0.8481481 0.1518519
  1 0.7635468 0.2364532

   Sex
Y      female      male
  0 0.1400560 0.8599440
  1 0.6694915 0.3305085
```

This function returns the probabilities to be either a child or an adult given that we survived or not. It does the same for the sex of the passenger. We also get the probability to be dead or alive without condition.

We see that we have a probability of 0.6 to die in the accident.

The probability of being a child or an adult, given that we survived or not, has to be interpreted carefully. Indeed, there are way more adults than children on the boat. It translates into a high chance of being an adult if we are either dead (85%) or alive (76%). We mustn’t get this probability mixed up with it as the probability of being dead **AND** an adult ...

In addition, we see that we have a probability of 0.86 to be a male if we died, and a probability of 0.67 to be a woman if we survived. Those results are close to what we were expecting.

7. Compare the resulting vector with the real solution and compute the accuracy of your classifier.

Now, we can try to predict if the passengers of the test dataset survived or died, according to their sex and if they are adults or not.

```
pred=predict(classifier,t_test_af,type="class")
```

The function returns a factor, which says if the passengers died or not. To see if our prediction is accurate, we compare the predicted factor to the one of the test dataset:

```
table(pred,t_test_af$Survived)

pred   0   1
   0 161  31
   1  31  75
```

This gives us an accuracy of 79.1% ($\frac{161+75}{298} = 0.791$), which is pretty impressive considering the fact that we are only using as parameter the sex and the age variables. This confirms that these parameters are the most important when evaluating the odds of death onboard.

C. Survival rates depending on the social class

1. Display the survival rates for all 3 classes of passengers.

We want to see if the class of the passenger may have an impact on his chances of survival:

```
prop.table(table(t_t$Pclass,t_t$Survived),1)

            0         1
  1 0.3116883 0.6883117
  2 0.5000000 0.5000000
  3 0.7841270 0.2158730
```

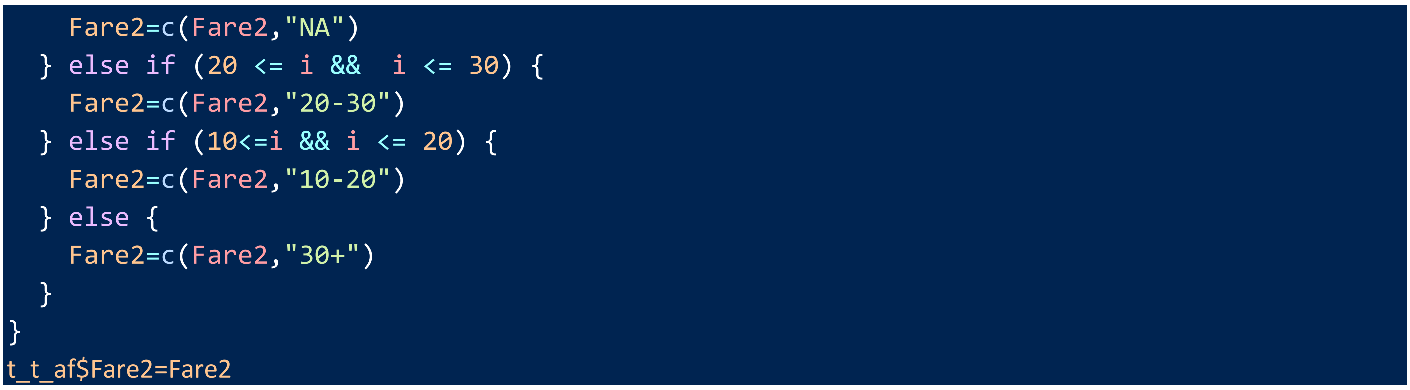
It seems that the passengers had more important chances to survive if they had a lower class. The passengers who were in 1st class died at 31%, while the ones who were in 3rd class died at 78%.

2. The Naive Bayes Classifier

Before we continue our analysis, we need to modify our training dataset. First, we convert the *Pclass* column into a factor (to compute our prediction later). We also need to create a new variable which regroups people who paid less than 10$, more than 30$ …

To do so, we use the following code:

```
Fare2=NULL
for(i in t_t_af$Fare){
  if (i<10) {
    Fare2=c(Fare2,"<10")
  } else if (is.na(i)) {
```

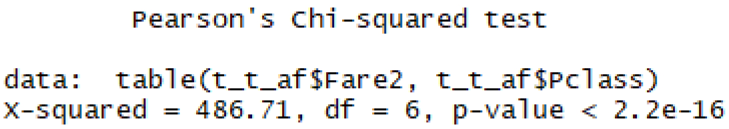


```
    Fare2=c(Fare2,"NA")
  } else if (20 <= i &&  i <= 30) {
    Fare2=c(Fare2,"20-30")
  } else if (10<=i && i <= 20) {
    Fare2=c(Fare2,"10-20")
  } else {
    Fare2=c(Fare2,"30+")
  }
}
t_t_af$Fare2=Fare2
```

We now have a brand-new column named *Fare2*. We don't forget to convert it into a factor as well and to apply the same changes on the test dataset.

3. Are the new attributes "Pclass" and "Fare2" independant? Justify.

We use a chi-squared test to determine if *Pclass* and *Fare2 are independent.* We get the following result:

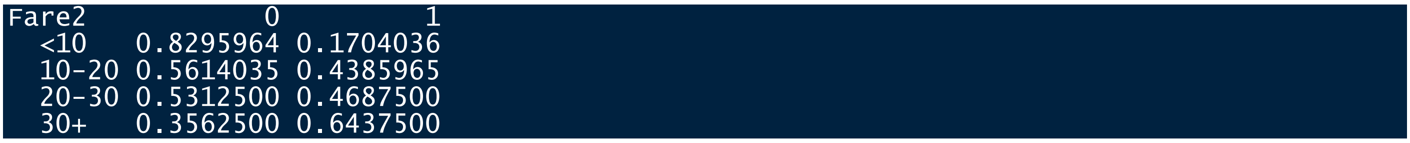


```
                Pearson's Chi-squared test

data:  table(t_t_af$Fare2, t_t_af$Pclass)
X-squared = 486.71, df = 6, p-value < 2.2e-16
```

The chi-squared value isn't equal to 0 so we need to use the p-value. The p-value is way smaller than 0.01 so we can say that *Fare2* and *Pclass* aren't independent. Furthermore, the Cramer's V is equal to *sqrt(486.71/(593*2)) = 0.64.* There is around 64% of correlation between the two variables.

This isn't surprising as a higher class has to be more expensive.

4. Display the survival rates depending on the value of "Fare2". Comment.

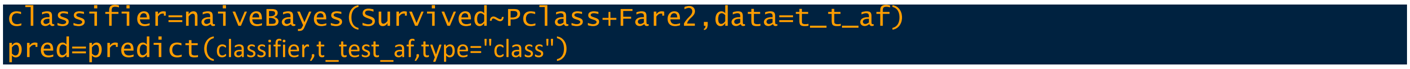


```
Fare2           0         1
  <10   0.8295964 0.1704036
  10-20 0.5614035 0.4385965
  20-30 0.5312500 0.4687500
  30+   0.3562500 0.6437500
```

We see that the more you pay, the more you're likely to survive the accident. The passengers who paid the least died at 82%, while the ones who paid more than 30$ died at 36% ...
The fare of the travel seems to have a high impact on your chances of survival.

5. Using questions B.6 et B.7, train a Naive Bayes classier with these 2 attributes and assess its accuracy on the "titanic_test.csv" validation set.

We train a Naive Bayes classifier with the attributes *Pclass* and *Fare2.* We proceed as we did previously:



```
classifier=naiveBayes(Survived~Pclass+Fare2,data=t_t_af)
pred=predict(classifier,t_test_af,type="class")
```

```
table(t_test_af$Survived,pred)
   pred
      0   1
  0 127  65
  1  51  55
```

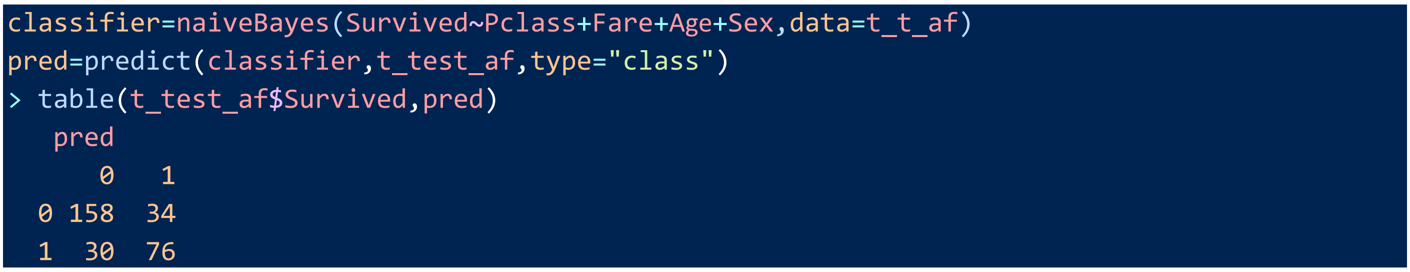
We successfully predicted 61.07 % of the values. It seems like the variables *Pclass* and *Fare2* aren't as good as *Child* and *Sex* to predict who is likely to die in an accident. This can be partially explained as *Fare2* and *Pclass* aren't independent. This involves that we may use several times the same information.

This shows that not any variable should be used to train our classifier.

D. Mixed model and decision trees

1. **Train another bayesian classifier using all the variables studied (child/adult, sex, class and fare range). Compare the results on the validation set with the two previous models. Comment.**

What if we used all the information we saw previously at the same time? We train a Naive Bayes classifier and we apply it on our test dataset:

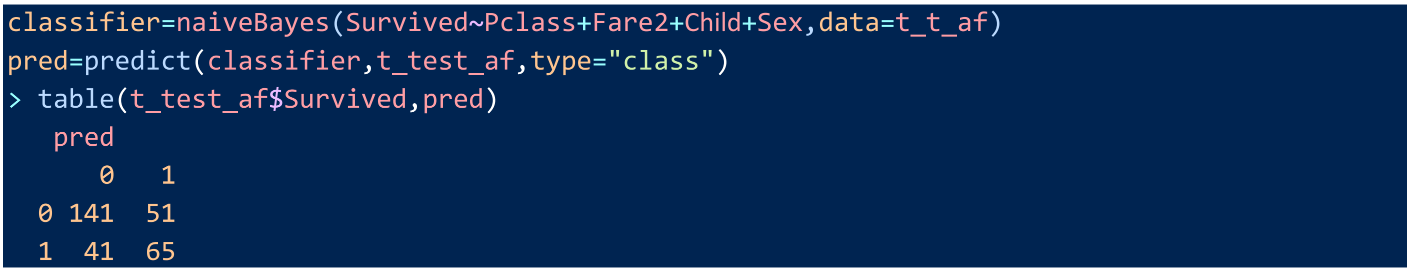


```
classifier=naiveBayes(Survived~Pclass+Fare+Age+Sex,data=t_t_af)
pred=predict(classifier,t_test_af,type="class")
> table(t_test_af$Survived,pred)
   pred
      0   1
  0 158  34
  1  30  76
```

We get a 77.52% accuracy. This is better than the prediction using *Pclass* and *Fare2* (61%), but still not as good as the 79.1% of our first prediction.

However, we used *Age* and *Fare* instead of *Child* and *Fare2*, which means we used more precise information. It would be even more interesting to use those 2 last variables to note if a combination of several variables is recommended.

With *Fare2* and *Child* we get the following result:



```
classifier=naiveBayes(Survived~Pclass+Fare2+Child+Sex,data=t_t_af)
pred=predict(classifier,t_test_af,type="class")
> table(t_test_af$Survived,pred)
   pred
      0   1
  0 141  51
  1  41  65
```

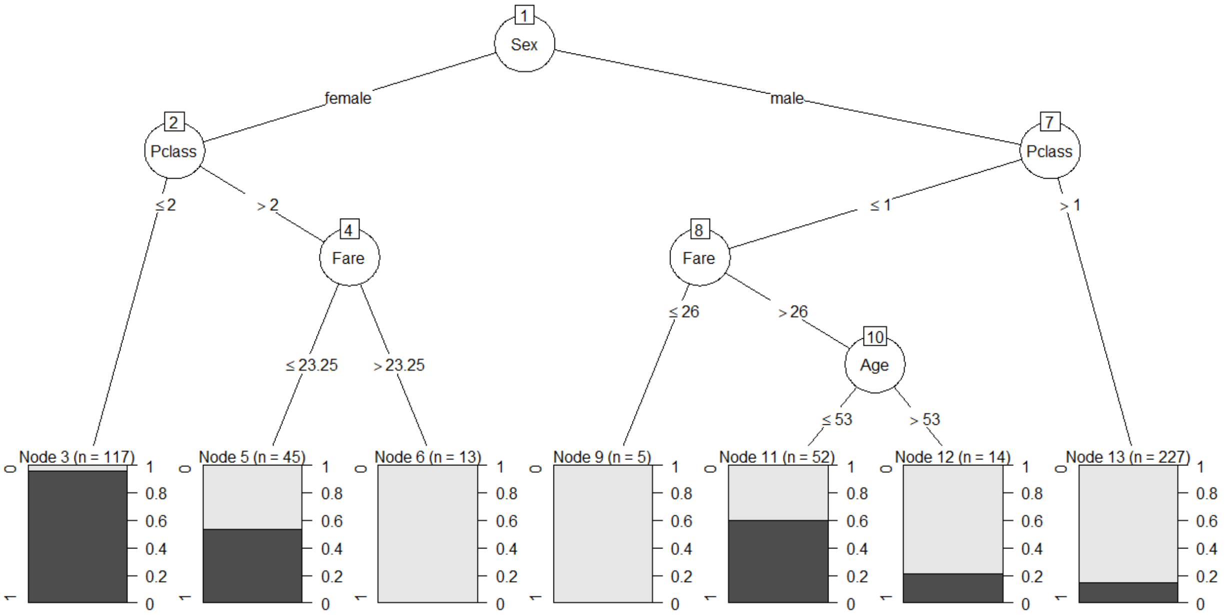
An accuracy of 69.13%. We see that this is 10% away from the result we obtained with *Sex* and *Child*. That shows that to cumulate the variables isn't the right choice. We should rather choose our variables carefully.

2. **We now want to try another classifier with the goal of outperforming naive bayes. We propose to use the C5.0 algorithm to create decision trees.**

We now want to see if we C50 algorithm may be a better solution than Naive Bayes to predict our values. This algorithm can be called by the function c5.0, which is included in the package e1071.

Before we compute this algorithm, we need to make sure that *Pclass, Fare* and *Age* are not factors (*Sex* and *Survived* have to remain factors). The variables which need not to be factors are the ones which contain numeric values.

```
t_t_af$Fare=as.numeric(as.character(t_t_af$Fare))
t_t_af$Pclass=as.numeric(as.character(t_t_af$Pclass))
t_t_af$Age=as.numeric(as.character(t_t_af$Age))
mod=C5.0(Survived~Pclass+Fare+Age+Sex, t_t_af, method='class')
plot(mod)
```



When we plot the value returned by the *c5.0()* function, we get this tree. The first variable to be checked is the sex of the passenger as it is the variable that has the biggest impact on the chances of survival. Then we'll check if the passenger was in a class higher or lower than 2. In this last case, we'll also check the fare paid by the passenger. And so on ...

We can see that we consider that the women who are in the 1st or the 2nd class are very likely to survive an accident. In the contrary, the man all have very low chances to survive, except the ones who are in 1st class and who paid an important fare (chance to survive: ~60%).

Two cases are considered as 100% lethal: the men who paid less than 26$ in 1st class and the women who paid more than 23.25$ in 3rd class. This is probably an anomaly due to the very low concerned population (respectively 13 and 5 people).

We apply this model on our validation set and get the following result:

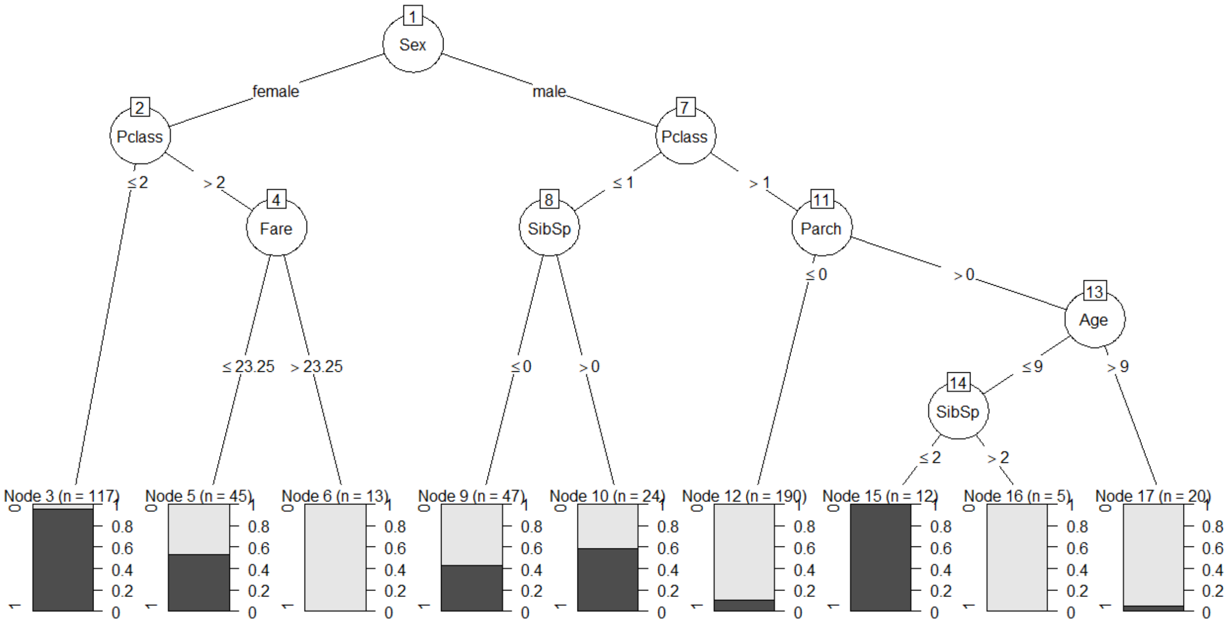
```
  pred
    0   1
0 151  41
1  28  78
```

We get an accuracy of 76.85%. Our prediction isn't better than the one we got with the bayesian classifier. It is pretty much equivalent.

3. **Try adding other attributes such as the boarding port (S=Southampton, C=Cherbourg, Q=Queenstown), the number of siblings on board (SibSp), the number of parents and children on board (Parch), or the cabin number (Cabin), and see if you can get a better tree. Comment on the resulting trees and their performances.**

We want to see if me can get a better prediction by including other variables.

We first add the variables *SibSp* and *Parch* as they have no missing values. We get the following result:



The Accuracy of this model is 79.53% which is the best one we got so far.

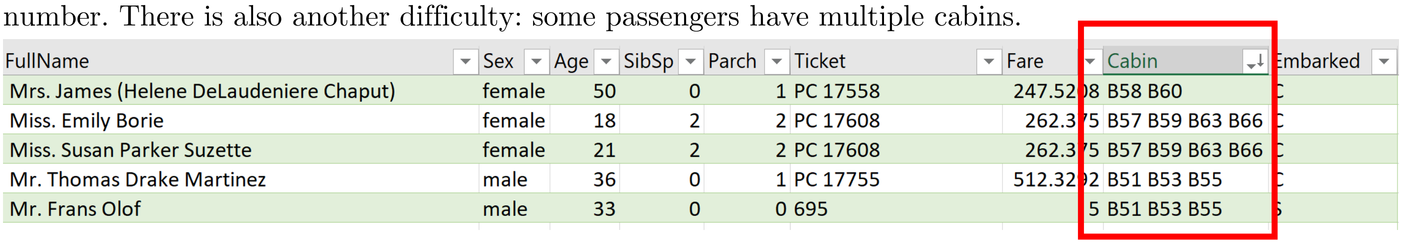
Let's now evaluate the « potential » of other variables:

° Cabin:

```
length(t_t_af$Cabin[t_t_af$Cabin == ""])
[1] 450
```

$$\frac{\mathbf{450}}{\mathbf{594}} = \mathbf{0.75}$$

This means that 75% of the rows are missing in the C*abin* variable. This is too much of a significant number. There is also another difficulty: some passengers have multiple cabins.



FullName	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
Mrs. James (Helene DeLaudeniere Chaput)	female	50	0	1	PC 17558	247.5208	B58 B60	C
Miss. Emily Borie	female	18	2	2	PC 17608	262.375	B57 B59 B63 B66	C
Miss. Susan Parker Suzette	female	21	2	2	PC 17608	262.375	B57 B59 B63 B66	C
Mr. Thomas Drake Martinez	male	36	0	1	PC 17755	512.3292	B51 B53 B55	C
Mr. Frans Olof	male	33	0	0	695	5	B51 B53 B55	S

Let's try to overcome these difficulties. We need to clean a bit the variable.

- First, we create a sub-table with only rows containing values in cabin

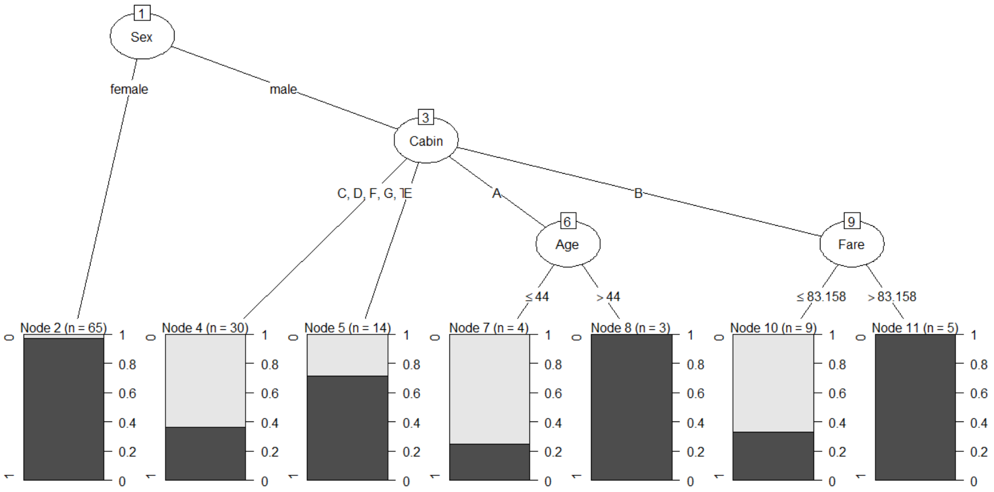
- Second, we only keep the first char of each string.
- Last, we convert Cabin into a factor

Here is the code that does so:

```
my_new_t_t <- t_t_af[!(t_t_af$Cabin == ""),]
my_new_t_t$Cabin=substr(my_new_t_t$Cabin, start = 1, stop = 1)
my_new_t_t$Cabin=as.factor(my_new_t_t$Cabin)

my_new_test_t <- t_test_af[!(t_test_af$Cabin == ""),]
my_new_test_t$Cabin=substr(my_new_test_t$Cabin, start = 1, stop = 1)
my_new_test_t$Cabin=as.factor(my_new_test_t$Cabin)
```

We now have **143 rows** in the titanic_train table and **61 rows** in the titanic_test table.
> Let's try to make predictions!

```
mod=C5.0(Survived~Pclass+Fare+Age+Sex+SibSp+Parch+Cabin,my_new_t_t, method='class')
```



```
table(my_new_test_t$Survived,pred)
   pred
     0  1
  0 18 10
  1  9 24
```

The accuracy of the model adding cabin in the model, lowers* the accuracy to 68% on the subset with 143 values.

* Removing *Cabin* in the model on the 143 values subset results in an 81% accuracy. Keeping (*Pclass+Fare+Age+Sex+SibSp+Parch*)

Conclusion:

This means that Cabin adds no values whatsoever to the model, it worsens the prediction. This is, of course, highly due to the fact that our clean dataset is too small because of the missing data ...

Let's try to reverse and to predict the Cabin class: A,B,C,...,T

```
mod=C5.0(Cabin~Fare+Pclass+Age,my_new_t_t, method='class')

> pred=predict(mod,my_new_test_t,type="class")
> table(my_new_test_t$Cabin,pred)
  pred
     A  B  C  D  E  F  G  T
  A  1  0  4  0  1  0  0  0
  B  0  2  7  0  2  0  0  0
  C  1  1 12  1  4  0  0  0
  D  0  1  1  5  2  2  0  0
  E  0  2  1  0  1  1  0  0
  F  0  0  0  0  1  5  0  0
  G  0  0  0  0  2  1  0  0
```

We cannot get a better accuracy than 42% by doing so, it still confirms the uselessness of Cabin.

° Embarked:

```
length(t_t_af$Embarked[t_t_af$Embarked == ""])
[1] 1
```

There is only one missing value in the *Embarked*.
Let's see quickly the repartition of the different values in *Embarked*:

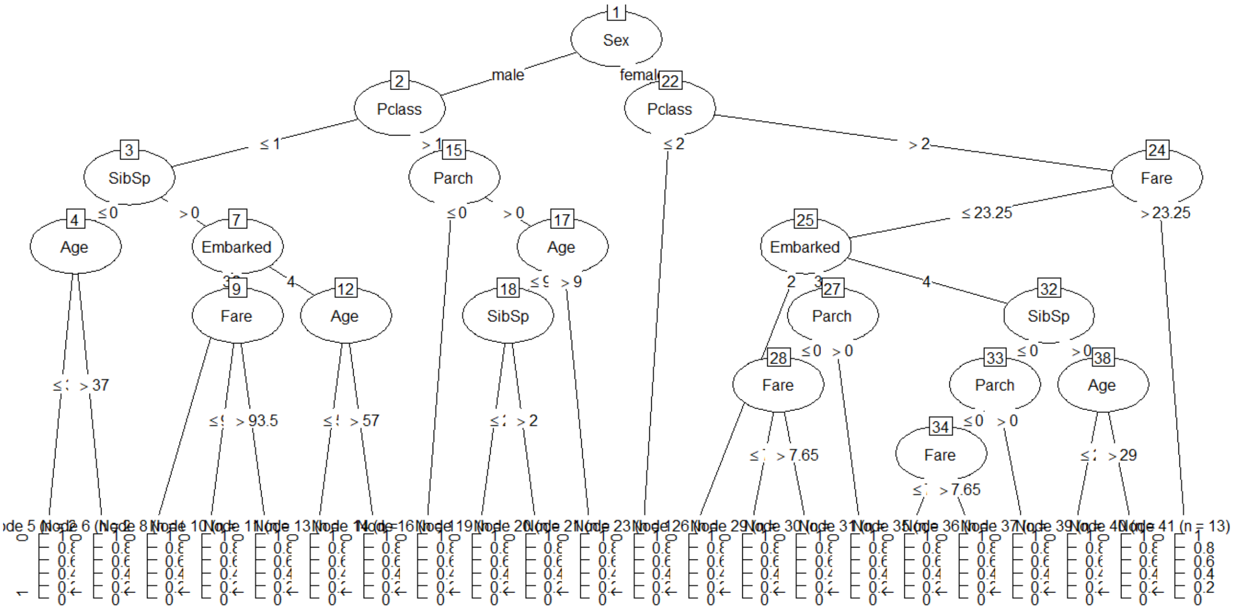
C	Q	S
115	48	429

We decide to replace the missing value with an S.

```
t_t$Embarked[532]= "S"
```

We process the decision tree again, adding the *Embarked* variable:

```
mod=C5.0(Survived~Pclass+Fare+Age+Sex+SibSp+Parch+Embarked,t_t_af, method='class')
```



Although it's unreadable, let's apply the model to our dataset and compare it to the real values:

```
table(t_test_af$Survived,pred)
   pred
      0   1
  0 164  28
  1  35  71
```

This is giving us an accuracy of 78.85%, thus adding the variable « Embarked » yields in a slightly less precise model (without counting that it takes more time to process).

In the end, we see that adding plenty of variables is a bad solution when you want to build a predictive classification model. You may end up with a worse accuracy and a very high complexity. This is perfectly illustrated by the fact that the 2nd best model was the bayesian classifier which was using only *Child* and *Sex.*