

NON-PARAMETRIC STATISTICS PROJECT
ON THE TOPIC

Non-parametric tests and density estimation on dataset Titanic

MADE BY
Cynthia CHEDRAWI
Marharyta TOMINA

PROFESSEUR
Marie Luce TAUPIN



March 6, 2023

Contents

1	Overview	2
2	Data description	2
3	Non-parametric Tests	2
3.1	Statistical tests to check the equality of the distributions	3
3.1.1	"Survived" and "Age" variables	3
3.1.2	"Survived" and "Fare" variables	6
3.2	Statistical tests to check the independence of variables	8
3.2.1	"PClass", "Sex", "Embarked" and "Survival"	8
3.2.2	"SibSp", "Parch" and "Survival"	10
4	Building the GLM	11
4.1	Evaluation of the quality of modelfit	12
5	Density Estimation	13
5.1	Estimation using histograms	13
5.2	Estimation of density using the kernel	17
5.3	Estimation using regression functions	19

1 Overview

2 Data description

We are going to work with the data of Titanic passengers. After removing all the information, that we do not want to consider in our survey, we are left with dataset, consisting of 9 columns and 1,309 rows.

Each row represents a single passenger, while each column gives us a certain piece of information about passenger.

Description: df [1,309 × 9]

PassengerId <int>	Survived <int>	Pclass <int>	Sex <chr>	Age <dbl>	SibSp <int>	Parch <int>	Fare <dbl>	Embarked <chr>
1	0	3	male	22.000000	1	0	7.2500	Southampton
2	1	1	female	38.000000	1	0	71.2833	Cherbourg
3	1	3	female	26.000000	0	0	7.9250	Southampton
4	1	1	female	35.000000	1	0	53.1000	Southampton
5	0	3	male	35.000000	0	0	8.0500	Southampton
6	0	3	male	31.776381	0	0	8.4583	Queenstown
7	0	1	male	54.000000	0	0	51.8625	Southampton
8	0	3	male	2.000000	3	1	21.0750	Southampton
9	1	3	female	27.000000	0	2	11.1333	Southampton
10	1	2	female	14.000000	1	0	30.0708	Cherbourg

1-10 of 1,309 rows

Previous 1 2 3 4 5 6 ... 100 Next

Figure 1: Dataset

The explanation of variables:

- "PassengerId" – id of the passanger.
- "Survived" – if person survived, or not (1 - yes, 0 - no).
- "Pclass" – class of the ticket (1 - 1st class, 2 - 2nd class, 3 - 3rd class).
- "Sex" – passanger's sex ("male", "female").
- "Age" – passanger's age (continuous data).
- "SibSp" – number of spouses and/or siblings the passenger had on board with him/her (0, 1, 2, 3, 4, 5, 8).
- "Parch" – number of parents and/or children the passenger had with him/her on board (0, 1, 2, 3, 4, 5, 6, 9).
- "Fare" – price, payed for the tickets by passanger (continuous variable).
- "Embarked" – port of embarkation that the passenger took ("Cherbourg", "Queenstown", "Southampton").

We can easily see, that we have two quantitative variables: "Age", "Fare"; and seven qualitative variables: "PassengerId", "Survived", "Pclass", "Sex", "SibSp", "Parch", "Embarked".

3 Non-parametric Tests

In this work we want to investigate mostly the "Survived" variable, and how the other variables are connected with it or if they have any kind of influence on the fact if person have survived or not. Thus, all the following tests will be aimed at researching this.

Let us first see, how the "Survived" variable is distributed.

We can see, that nearly twice more people did not survive after Titanic crashed. Let us hold the following tests.

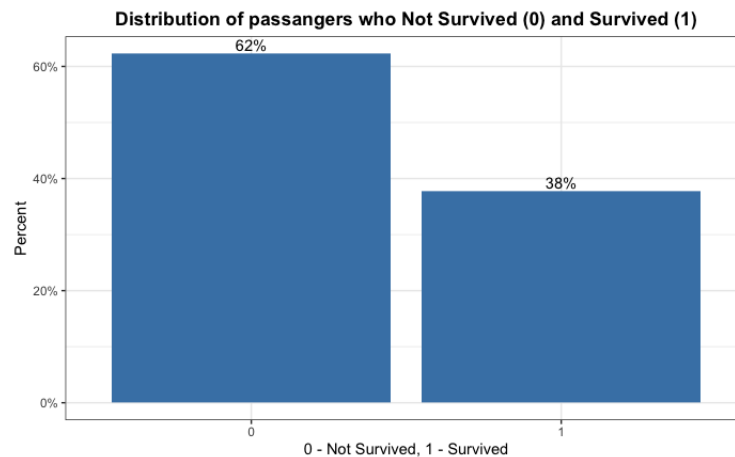


Figure 2: "Survived" variable distribution

3.1 Statistical tests to check the equality of the distributions

In this part of the work we want to separate a set of passengers on Titanic to the ones who survived and the ones, who did not survive and see if the continuous variables, "Age", and then "Fare", have the same distribution in both of the groups. If they do, it will mean, that age of passenger (the price they paid for the tickets) did not influence the survival of the person, if not - vice versa.

3.1.1 "Survived" and "Age" variables

First thing we want to do is to visualise the data, that we have to conduct the primary analysis. Here is a boxplot of the "Age" variable, divided considering "Survived" data. Analyzing this picture, we

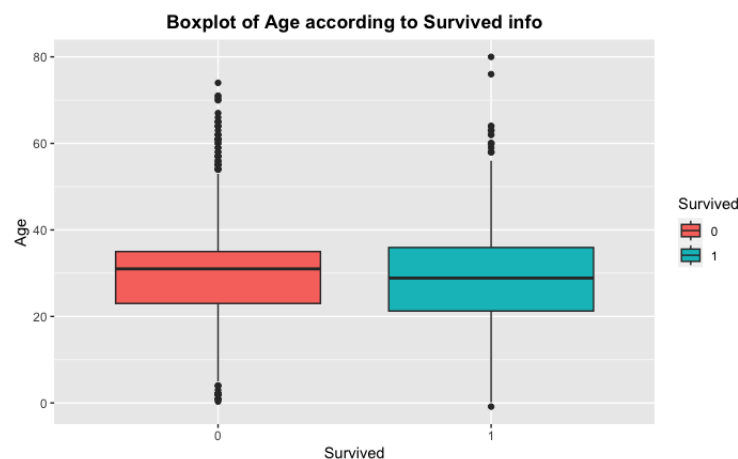


Figure 3: "Age" variable boxplot, divided by "Survived" (0 - No, 1 - Yes)

can tell, that medians are slightly different, the Interquartile range box of the survived group(1) is slightly bigger, as well as the whiskers. The group of the ones, who did not survive(0) has a lot of outliers.

Overall, from this picture it is hard to certainly say, if those groups have the same distribution or not.

The next this we build is the histogram of both groups, where we can see better the difference between the distributions, and slightly different mean. Considering the information, collected from the images, we can make a primary conclusion, that the groups have different distributions, as well, as different

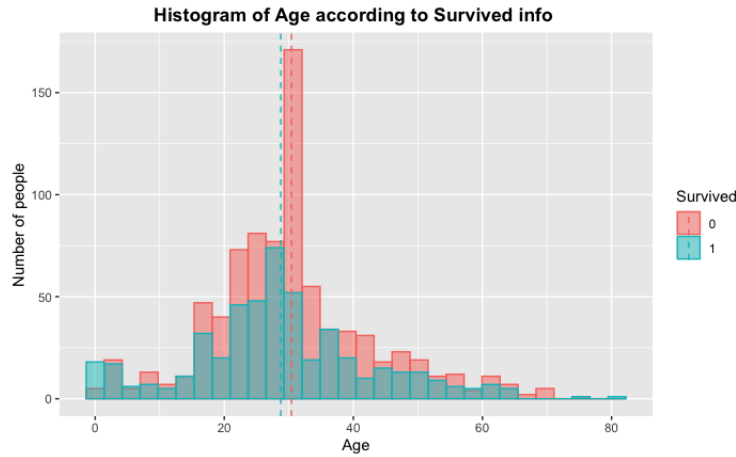


Figure 4: "Age" variable histogram, divided by "Survived" (0 - No, 1 - Yes)

means, but we want to have a numerical result, proving that and showing the degree of difference. For this we are going to hold the following tests.

Mean equality check

To compare the means of two distributions, we might use Student t - test. However, Student t - test requires testing data to have a standard Gaussian distribution. If this condition is not met, we might use the CLT theorem.

Student t - test The two-sample t-test is used to determine if two population means are equal. It assumes equal variances of the distributions and Normally distributed data.

For two-sample unpaired data case, t-test is defined:

- $H_0 : \mathbb{E}(X) = \mathbb{E}(Y)$
- $H_1 : \mathbb{E}(X) \neq \mathbb{E}(Y)$
- Test statistic: $T = \frac{\bar{X} - \bar{Y}}{s \sqrt{\frac{1}{n} + \frac{1}{m}}}$

where n and m are the sample sizes, \bar{X} and \bar{Y} are the sample means, and s is the sample variances.

CLT

The central limit theorem says that the sampling distribution of the mean will always follow a normal distribution when the sample size is sufficiently large. This theorem is applicable even for variables that are originally not normally distributed.

The test is defined:

- $H_0 : \mathbb{E}(X) = \mathbb{E}(Y)$
- $H_1 : \mathbb{E}(X) \neq \mathbb{E}(Y)$
- Test statistic: $T = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{s_1^2}{n} + \frac{s_2^2}{m}}}$

where s_1 and s_2 are the sample variances.

To understand, which one of the tests we should use, we have to check firstly, if our data has a Normal distribution or not. To do so, we will use Shapiro-Wilk test and a special case of Kolmogorov-Smirnov test (Lilliefors test of normality).

Normality distribution check

Shapiro–Wilk test

Shapiro–Wilk test is a correlation test, - tests based on the ratio of two weighted least-squares estimates of scale obtained from order statistics.

Shapiro and Wilk test was originally restricted for sample size of less than 50. This test was the first test that was able to detect departures from normality due to either skewness or kurtosis, or both. It has become the preferred test because of its good power properties.

Given an ordered random sample, $y_1 < y_2 < \dots < y_n$ the original Shapiro-Wilk test is defined as:

- H_0 : data has normal distribution
- H_1 : data does not have normal distribution
- Test statistic: $W = \frac{(\sum_{i=1}^n a_i y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$

where y_i is the i^{th} order statistic, \bar{y} is the sample mean, $a_i = (a_1, \dots, a_n) = \frac{m^T V^{-1}}{(m^T V^{-1} V^{-1} m)^{\frac{1}{2}}}$, and $m = (m_1, \dots, m_n)$ are the expected values of the order statistics of independent and identically distributed random variables sampled from the standard normal distribution and V is the covariance matrix of those order statistics.

The value of W lies between zero and one. Small values of W lead to the rejection of normality whereas a value of one indicates normality of the data

Lilliefors test

The KS test (and the Lilliefors) looks at the largest difference between the empirical CDF and the specified distribution.

Lilliefors (LF) test is a modification of the Kolmogorov-Smirnov test. The KS test is appropriate in a situation where the parameters of the hypothesized distribution are completely known. However, sometimes it is difficult to initially or completely specify the parameters as the distribution is unknown. In this case, the parameters need to be estimated based on the sample data. In contrast with the KS test, the parameters for LF test are estimated based on the sample.

Given a sample of n observations, Lilliefors test is defined as follows:

- $H_0 : F(x) = F^*(x)$
- $H_1 : F(x) \neq F^*(x)$
- Test statistic: $D = \sup_x |F^*(x) - F_n(x)|$

where $F^*(x)$ is the EDF of Gaussian distribution and $F_n(x)$ is the sample cumulative distribution function.

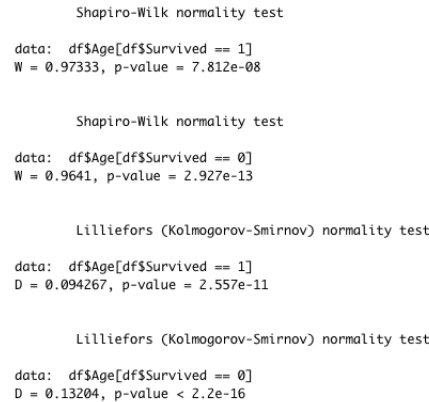
After conducting the tests, we see, that none of our data has a normal distribution, because the p-values are extremely small. Thus we will use CLT to check if the means of the distributions are equal. The CLT test result gives us a small p-value equal to

$$p = 0.03086523$$

which is smaller, than 0.05, so we reject the zero-hypothesis H_0 about the equality of the means and conclude, that the expectations are different.

Distributions equality check

Followingly, we will use Kolmogorov-Smirnov(having the same statistic and hypothesis, as defined in Lilliefors test before) and Wilcoxon tests to see if the distributions of the Age of people who survived, and who did not, are the same or not.



```

Shapiro-Wilk normality test

data:  df$Age[df$Survived == 1]
W = 0.97333, p-value = 7.812e-08

Shapiro-Wilk normality test

data:  df$Age[df$Survived == 0]
W = 0.9641, p-value = 2.927e-13

Lilliefors (Kolmogorov-Smirnov) normality test

data:  df$Age[df$Survived == 1]
D = 0.094267, p-value = 2.557e-11

Lilliefors (Kolmogorov-Smirnov) normality test

data:  df$Age[df$Survived == 0]
D = 0.13204, p-value < 2.2e-16

```

Figure 5: Shapiro–Wilk and Kolmogorov–Smirnov tests

Wilcoxon test

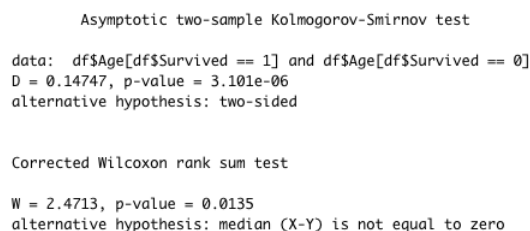
The Wilcoxon Rank Sum Test is often described as the non-parametric version of the two-sample t-test. The Wilcoxon test, makes the two assumptions of independence and equal variance. It does not assume our data have a known distribution. Since the Wilcoxon Rank Sum Test does not assume known distributions, it does not deal with parameters, and therefore we call it a non-parametric test.

Let us have a sample D_1, \dots, D_n of real r.v. of length n and assumed diffuse law. When the median of $D_1 + D_2$ is different from 0, the ranks of the positive D_i are not distributed uniformly on $1, \dots, n$. We want to test:

- $H_0 : Med(D_1 + D_2) = 0$
- $H_1 : Med(D_1 + D_2) \neq 0$
- Test statistic: $W_n^+ = \sum_{i=1}^n R_{|D|}(i) 1_{\{D_i > 0\}}$

where $R_{|D|}$ the rank vector associated with $(|D_1|, \dots, |D_n|)$.

Here both tests give us a small p-values, which makes us reject the null-hypothesis about equality of the distributions of two sets, and conclude, that they have different distributions.



```

Asymptotic two-sample Kolmogorov-Smirnov test

data:  df$Age[df$Survived == 1] and df$Age[df$Survived == 0]
D = 0.14747, p-value = 3.101e-06
alternative hypothesis: two-sided

Corrected Wilcoxon rank sum test

W = 2.4713, p-value = 0.0135
alternative hypothesis: median (X-Y) is not equal to zero

```

Figure 6: Kolmogorov–Smirnov and Wilcoxon tests

3.1.2 "Survived" and "Fare" variables

We want to conduct the same tests for the "Fare" variable, trying to understand if the price of a ticket could have influenced survival.

By the boxplot image we can conclude, that the distributions have different Interquartile range box, and the medians. They both have a lot of outliers. Histogram also proves, that mean and the distributions

of our data is different.

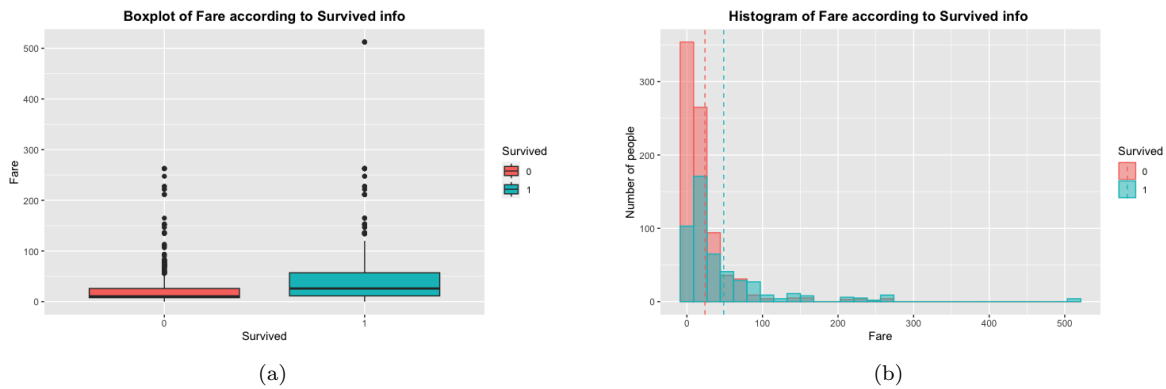


Figure 7: "Fare" variable boxplot and histogram, divided by "Survived" (0 - No, 1 - Yes)

Before conducting the tests on the equality of the means of two datasets, we check if the data is normally distributed, just like in the previous part. It turns out, that none of the data is normally distributed, so we will use TLC once again to compare the means.

```
Shapiro-Wilk normality test
data: df$Fare[df$Survived == 1]
W = 0.60017, p-value < 2.2e-16

Shapiro-Wilk normality test
data: df$Fare[df$Survived == 0]
W = 0.51546, p-value < 2.2e-16

Lilliefors (Kolmogorov-Smirnov) normality test
data: df$Fare[df$Survived == 1]
D = 0.26882, p-value < 2.2e-16

Lilliefors (Kolmogorov-Smirnov) normality test
data: df$Fare[df$Survived == 0]
D = 0.28377, p-value < 2.2e-16
```

Figure 8: Shapiro-Wilk and Kolmogorov-Smirnov tests

The TLC gives us the p-value equal to:

$$p = 6.283862e - 14$$

which is obviously less, than 0.05, which makes us reject null hypothesis about the equality of the means, and say that the means are not equal.

Now, we run the tests on the equality of the distributions and reassure our previous conjectures, that two sets do not come from the same distribution.

```

Asymptotic two-sample Kolmogorov-Smirnov test

data: df$Fare[df$Survived == 1] and df$Fare[df$Survived == 0]
D = 0.27839, p-value < 2.2e-16
alternative hypothesis: two-sided


Wilcoxon rank sum test with continuity correction

data: df$Fare[df$Survived == 1] and df$Fare[df$Survived == 0]
W = 269459, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0


Corrected Wilcoxon rank sum test

W = -11.001, p-value = 0
alternative hypothesis: median (X-Y) is not equal to zero

```

Figure 9: Kolmogorov-Smirnov and Wilcoxon tests

3.2 Statistical tests to check the independence of variables

3.2.1 "PClass", "Sex", "Embarked" and "Survival"

In this section we will work with qualitative variables and see if they have significant level of independence from the "Survived" variable (i.e. do not have any influence on the fact that person survived or not) or they are dependent.

Firstly, let us see the distribution of classes depending on the fact if person survived, or not:

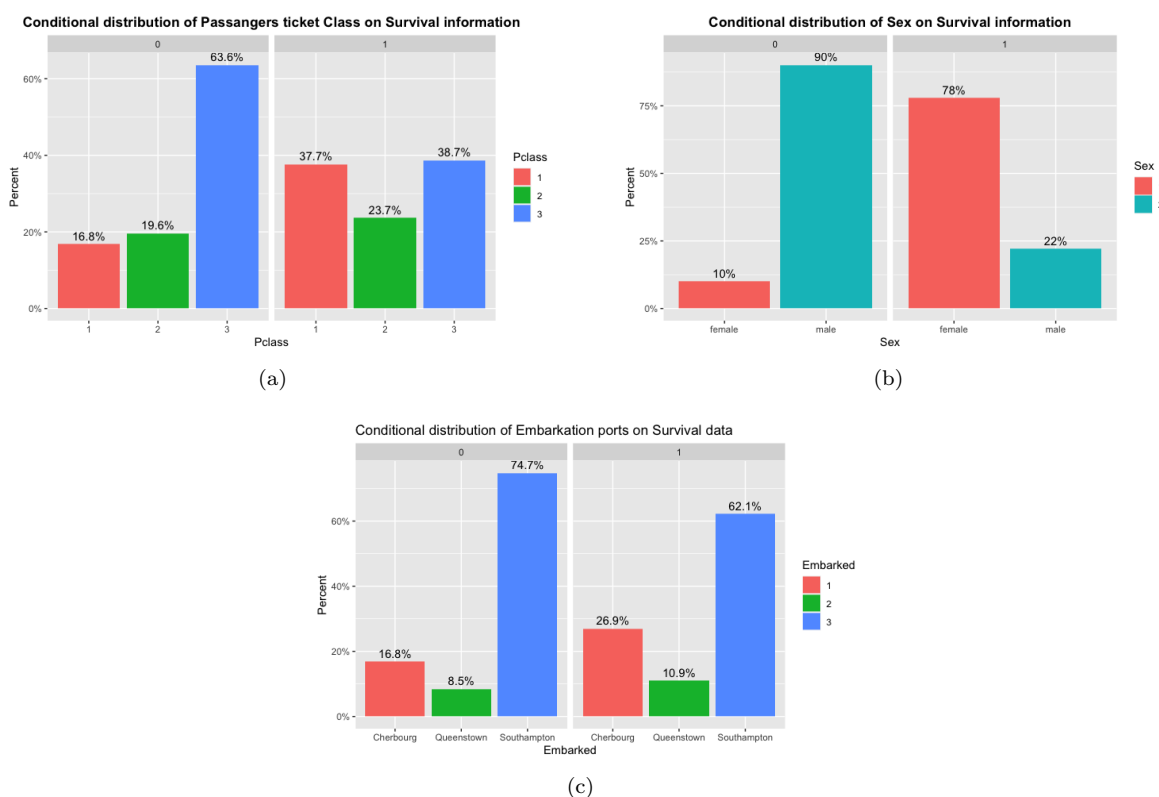


Figure 10: Conditional distribution of "PClass", "Sex", "Embarked" on Survival information

Now we will do a Chi-Square test, that is used to determine if there is a significant dependence between two categorical variables.

Chi-Square test The chi-square test of independence is used to test for a relationship between two categorical variables. The Chi-square test is used when the sample is large enough (in this case the

p-value is an approximation that becomes exact when the sample becomes infinite, which is the case for many statistical tests).

The test is:

- H_0 : $Variable_1$ is independent of $Variable_2$
- H_1 : $Variable_1$ is not independent of $Variable_2$
- Test statistic: $\chi^{2*} = \sum_{i=1}^{rc} \frac{(O_i E_i)^2}{E_i}$

where r is the number of rows and c is the number of columns in dataset, $O_1, O_2 \dots, O_{rc}$ denote the observed counts for each cell, $E_1, E_2 \dots, E_{rc}$ denote the respective expected counts for each cell, and

$$E = \frac{\text{row total} * \text{column total}}{n}$$

The Chi-squared test for all the variables gave high chi-squared values and really small p values, meaning that the class of the ticket, sex and port of embarkation are not independent with the "Survival" variable.

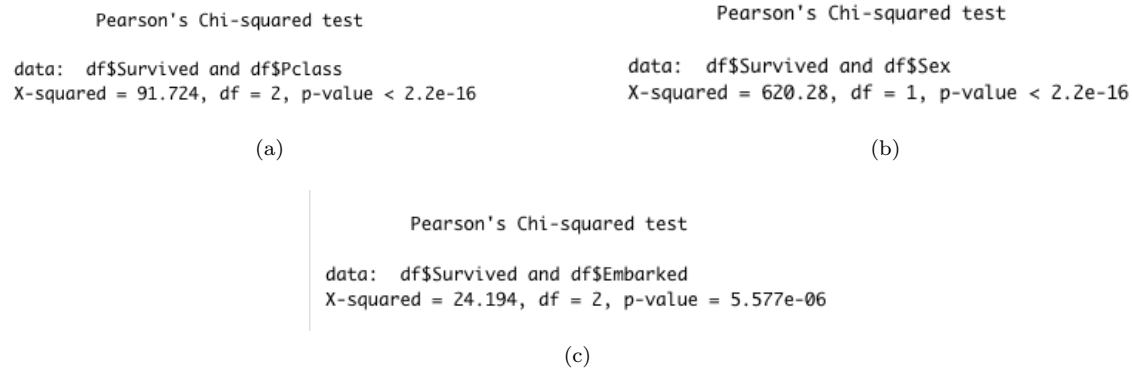


Figure 11: Chi-squared test for "PClass", "Sex", "Embarked" with "Survival"

Fisher exact test Fisher test is another independence test, used to determine if there is a significant relationship between two categorical variables. Fisher's exact test is used when the sample is small (and in this case the p-value is exact and is not an approximation).

The hypotheses of the Fisher's exact test are the same than for the Chi-square test, that is:

- H_0 : $Variable_1$ is independent of $Variable_2$
- H_1 : $Variable_1$ is not independent of $Variable_2$

After running Fisher's test, we have small p-values, that makes us reject the null-hypothesis and prove the result of the previous test of not independence of these values from "Survived".

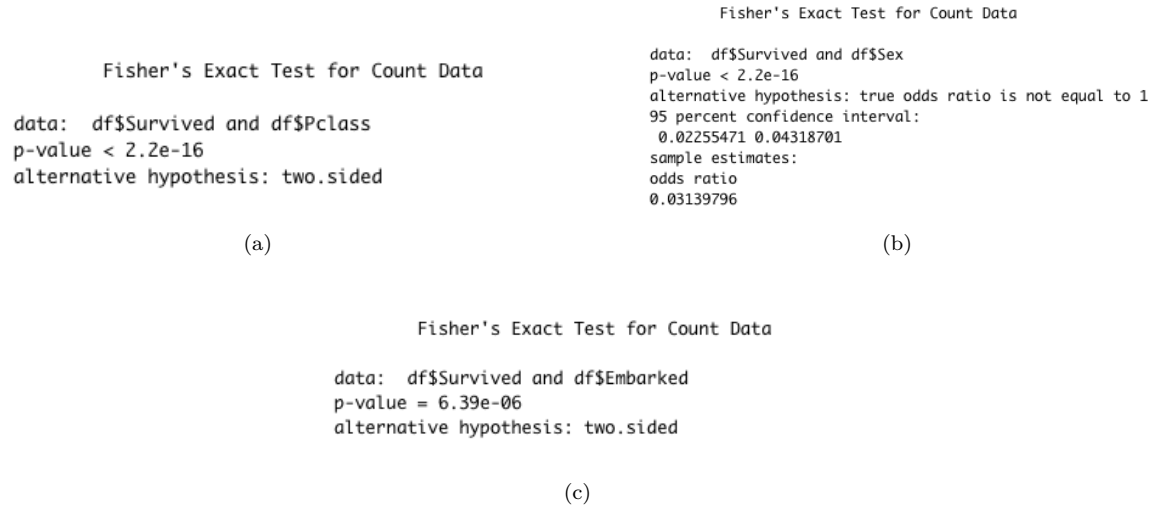


Figure 12: Fisher exact test for "PClass", "Sex", "Embarked" with "Survival"

3.2.2 "SibSp", "Parch" and "Survival"

As variables "SibSp" and "Parch" have a lot of classes (7 and 8), we thought, that it would be more reasonable to divide them into new classes to reduce its amount.

For example, "SibSp", number of spouses and/or siblings the passenger had on board with him/her takes the values 0, 1, 2, 3, 4, 5, 8. We group (0), (1, 2, 3) and (4, 5, 8) in three classes.

The same we do with the "Parch" variable, number of parents and/or children the passenger had with him/her on board, which takes the values 0, 1, 2, 3, 4, 5, 6, 9. We grouped (0), (1, 2, 3) and (4, 5, 6, 9) in three classes.

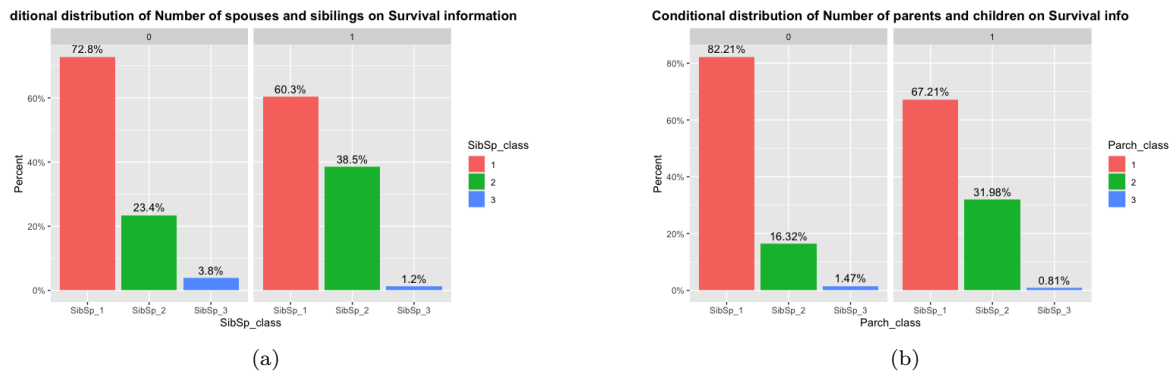


Figure 13: Conditional distribution of "SibSp", "Parch" on Survival information

Further we did the same tests, Chi-square and Fisher's exact test to check if there is a significant dependence.

Having small p-values, once again we reject null-hypothesis.

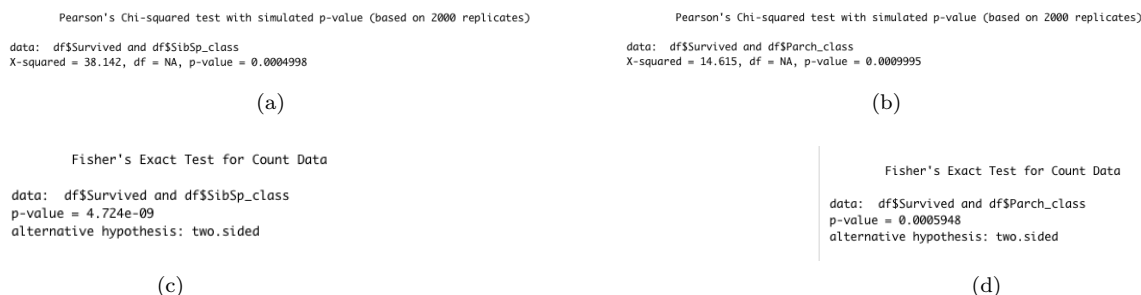


Figure 14: Chi-square and Fisher's exact test for "SibSp", "Parch" with "Survival"

4 Building the GLM

To build a proper model for predicting the "Survived" variable, we will use forward stepwise algorithm to choose the most significantly important predictors.

Firstly, we create a model with just an intersection. After that we want to make a new model, where we add one predictor, which is going to be significant and would predict better, than others.

After some tests, we notice that adding "Sex" variable (which is significant looking at the small p-value after the tests conducted) to our glm performs better, than adding the others, basing on the Residual deviance parameter, which tells us how well the response variable can be predicted by the specific model that we fit. The lower the value, the better the model is able to predict the value of the response variable.

```
Call:
glm(formula = Survived ~ Sex, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8707  -0.5262  -0.5262   0.6180   2.0227

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.5588     0.1222  12.75  <2e-16 ***
Sexmale     -3.4660     0.1596  -21.71  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1735.1  on 1308  degrees of freedom
Residual deviance: 1079.7  on 1307  degrees of freedom
AIC: 1083.7

Number of Fisher Scoring iterations: 4
```

Figure 15: GLM summary

Now we perform ANOVA test to compare the model without any predictors and with the one, that we added.

The `anova()` function will take the model objects as arguments, and return an ANOVA testing whether the more complex model is significantly better at capturing the data than the simpler model. If the resulting p-value is sufficiently low (usually less than 0.05), we conclude that the more complex model is significantly better than the simpler model, and thus favor the more complex model. If the p-value is not sufficiently low (usually greater than 0.05), we should favor the simpler model.

As we can see, the result shows a Df of 1 (indicating that the more complex model has one additional parameter), and a very small p-value ($< .001$). This means that adding the "Sex" to the model did lead to a significantly improved fit over the model.a. Further we will do the same procedure, adding new

Analysis of Deviance Table

```

Model 1: Survived ~ 1
Model 2: Survived ~ Sex
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      1308      1735.1
2      1307      1079.7  1    655.45 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figure 16: ANOVA for model without predictors and with one significant.

variable, that shows better results of prediction, ANOVA testing with the previous model and choose the best one.

We end up with the following model:

```

Call:
glm(formula = Survived ~ Sex + Pclass + Age + SibSp, family = binomial,
    data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8106  -0.4891  -0.3503   0.4661   2.5625

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.462572    0.458058  11.926 < 2e-16 ***
Sexmale     -3.761191    0.185586 -20.267 < 2e-16 ***
Pclass      -1.081222    0.113341  -9.540 < 2e-16 ***
Age         -0.037840    0.007273  -5.203 1.96e-07 ***
SibSp       -0.375245    0.088391  -4.245 2.18e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1735.13  on 1308  degrees of freedom
Residual deviance: 961.67  on 1304  degrees of freedom
AIC: 971.67

Number of Fisher Scoring iterations: 5

```

Figure 17: Final GLM summary

The model.e turns out to show the best results, comparing to the others. After comparing it with the model, having all variables as predictors, we also see, that model.e, the last one with 4 predictors still shows better results.

In R, stepAIC is one of the most commonly used search method for feature selection. We try to keep on minimizing the stepAIC value to come up with the final set of features. “stepAIC” does not necessarily mean to improve the model performance, however, it is used to simplify the model without impacting much on the performance. So AIC quantifies the amount of information loss due to this simplification. AIC stands for Akaike Information Criteria.

We want to choose the model with the smallest AIC value, which is model.e in our case once again.

4.1 Evaluation of the quality of modelfit

ROC curve

A ROC curve is constructed by plotting the true positive rate (TPR) against the false positive rate (FPR). The true positive rate is the proportion of observations that were correctly predicted to be positive out of all positive observations ($TP/(TP + FN)$). Similarly, the false positive rate is the proportion of observations that are incorrectly predicted to be positive out of all negative observations ($FP/(TN + FP)$).

Classifiers that give curves closer to the top-left corner indicate a better performance.

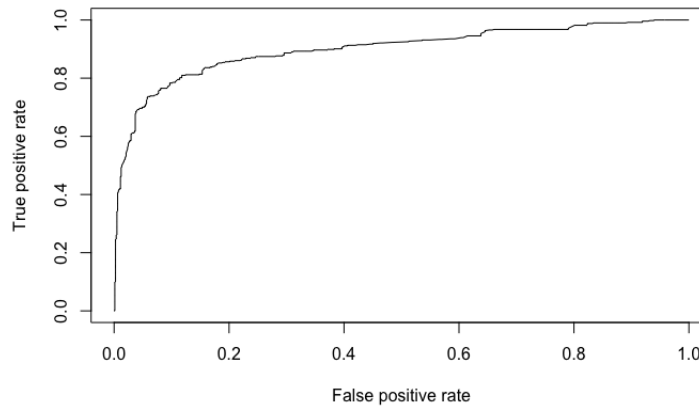


Figure 18: ROC curve for final model

We can also see the **AUC** value - the area under the ROC curve. In our case auc value is equal to 0.8991319. It is known, that the closer the AUC value is to the 1, the better the given model fits the data.

5 Density Estimation

In this part of the project, we are interested in estimating the density of the variables. We will estimate the density first by using histograms, second by kernel functions and then by using regression functions. We choose to work on the variable age from our dataset.

5.1 Estimation using histograms

In order to check the estimation of the variables, we will use the function `hist()`. We start by plotting two histograms one by frequency and the other with probabilities.

A histogram in frequency shows the count or number of observations falling in each bin of the histogram. It is calculated by dividing the range of the data into a set of intervals, called bins, and counting the number of observations that fall into each bin. While, a histogram with probabilities shows the proportion or probability of observations falling in each bin of the histogram. It is calculated by dividing the count in each bin by the total number of observations and normalizing by the bin width.

The difference is that the histogram in frequency shows the actual number of observations in each bin, while the histogram with probabilities shows the relative frequency or probability of observations in each bin.

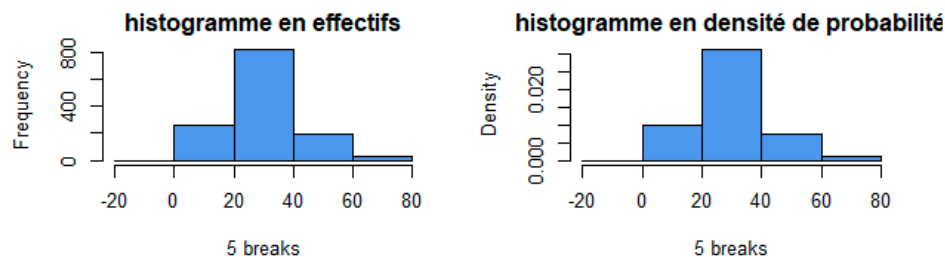


Figure 19: Histogram in frequency and with probabilities"

We can clearly see that we have the same results in the two histograms.

In the function `hist()`, we often use a `breaks` argument. This argument specifies how to divide the range of the data into intervals, or "bins", for the purpose of creating the plot. These bins are used to count the number of data points that fall within each bin, and the resulting counts are used to create the plot.

We will specify now the `breaks` argument to change the number and/or width of the bins used for the plot. We consider the following number of breaks: 5, 10, and 15.

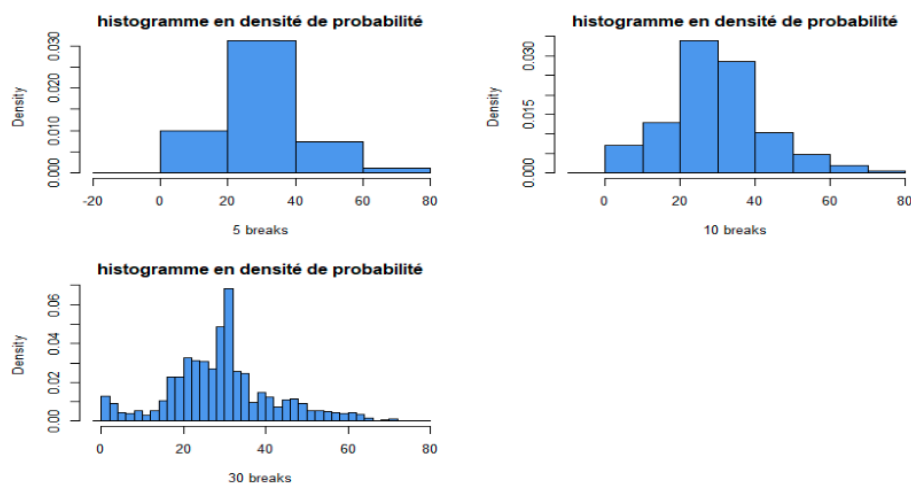


Figure 20: Histogram with different number of breaks

In figure 20, the histograms show the distribution of the variable "Age" using different numbers of breaks for grouping the data. The x-axis represents the range of ages, and the y-axis represents the frequency or density of the observations within each bin.

The first histogram with 5 breaks shows a relatively smooth distribution, with a peak around the age of 30 and a gradual decrease in density towards older ages. The second histogram with 10 breaks shows a similar pattern, but with more detail, revealing some gaps in the distribution (for example, around ages 28-32 and 50-54) that were not visible in the first histogram. And the third histogram with 30 breaks shows even more detail, with some small peaks and valleys in the distribution that were not visible before. However, it also shows some noise or fluctuations that may be due to sampling variability.

We will now try to plot the histogram with greater number of breaks.

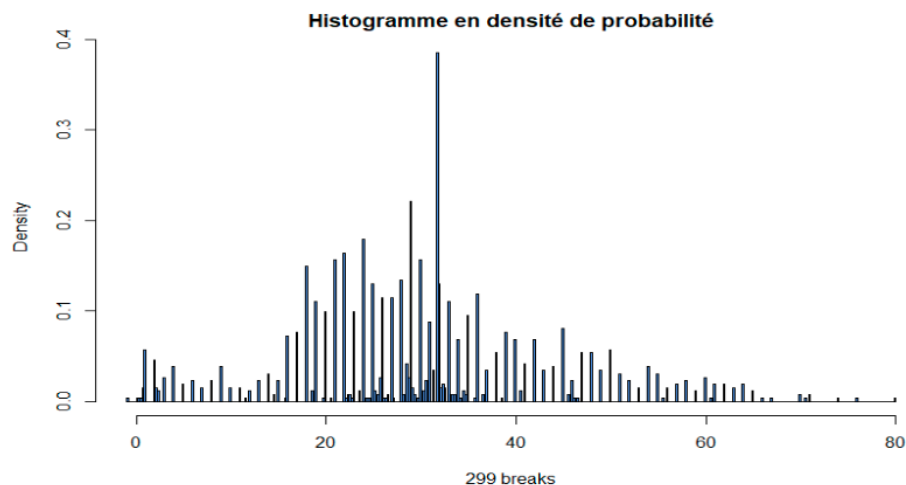


Figure 21: Histogram with 299 breaks

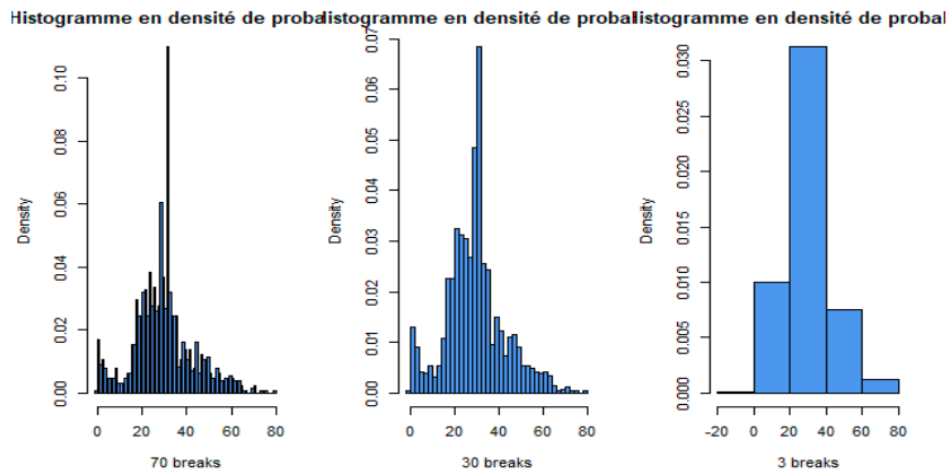


Figure 22: Histogram with higher number of breaks

In figure 21, histogram has 299 breaks, which means there is one bar for each observation. This histogram is not very informative since we cannot see any patterns in the data. The next three histograms have 70, 30, and 3 breaks respectively. In figure 22, the histogram with 70 breaks shows more information about the data distribution. We can see that the distribution is skewed to the right with most passengers being in their 20s and 30s. The histogram with 30 breaks provides even more information about the distribution of the data, and we can see more clearly that the distribution is skewed to the right. Finally, the histogram with only 3 breaks shows the general shape of the distribution, but it does not provide much detail about the distribution.

Overall, the choice of the number of breaks in a histogram depends on the amount of detail we want to see in the data and the overall shape of the distribution. By varying that argument, we can see that the choice of the number of classes affects the appearance of the histogram. A too small number of classes can mask characteristics of the distribution, while too many can introduce noise.

We will repeat now our work with the library ggplot2 and we get these two histograms with different bins.

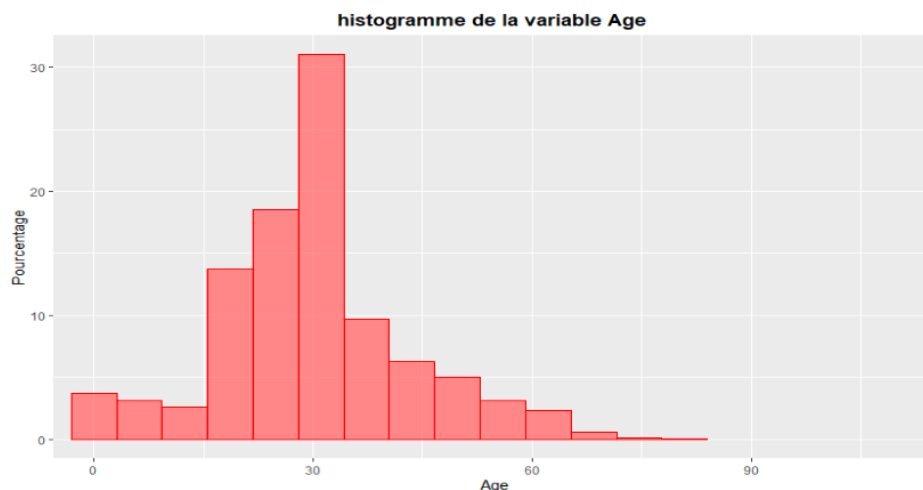


Figure 23: Histogram with 14 bins

In both Figures 23 and 24, the x-axis represents the age of the passengers and the y-axis represents the percentage of passengers in each age range.

The first histogram has 14 bins or breaks, while the second has 20 bins. The number of bins determines

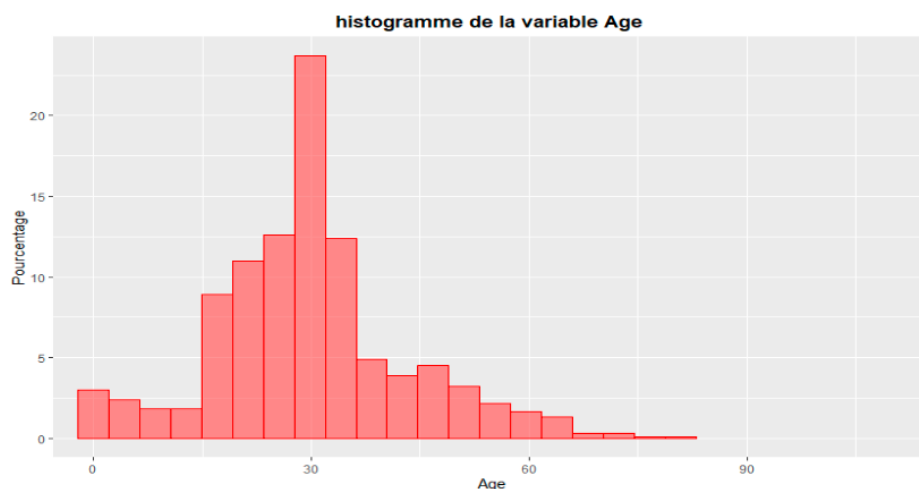


Figure 24: Histogram with 20 bins

the granularity of the histogram, with a higher number of bins resulting in a more detailed view of the distribution of the variable.

Both histograms show a peak in the number of passengers in their early 20s, with a gradual decrease in the number of passengers as age increases. The second histogram with more bins provides a more detailed view of the distribution and highlights some smaller peaks in the number of passengers in their 30s and 50s.

Our step now will be adding to our histogram a uni-dimensional representation of the observations with the function `rug()`. The rug plot shows the actual location of each observation as a tick mark along the

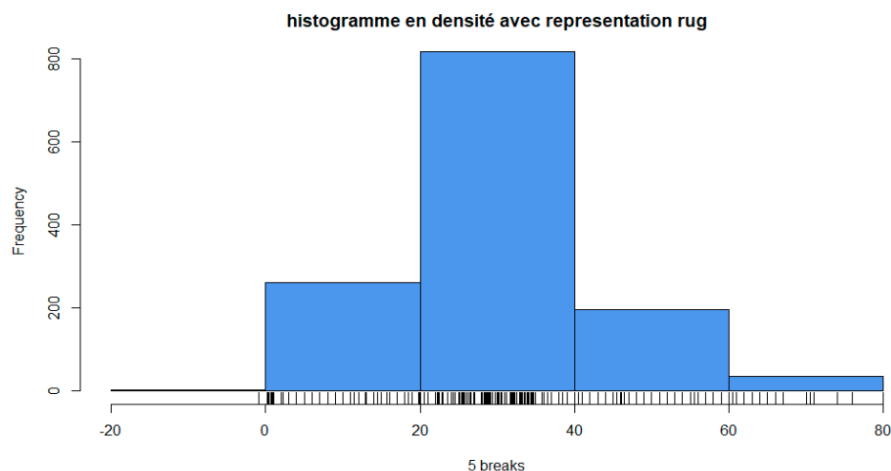


Figure 25: Histogram with the function rug

horizontal axis. This can be useful to get a sense of the overall distribution and to identify any outliers or unusual patterns in the data.

We compare now the function `truehist()` to `hist()`. The output of `truehist()` is similar to that of `hist()`, but with a few additional components. In figure 27, we can see the plot by the `truehist()` function while trying different options of it such as density which provides an estimate of the underlying density of the data, which is useful for visualizing the shape of the distribution and the border, the limit on x and `plot.n`. Overall, the `truehist()` function provides a more informative and visually appealing alternative to the basic `hist()` function.

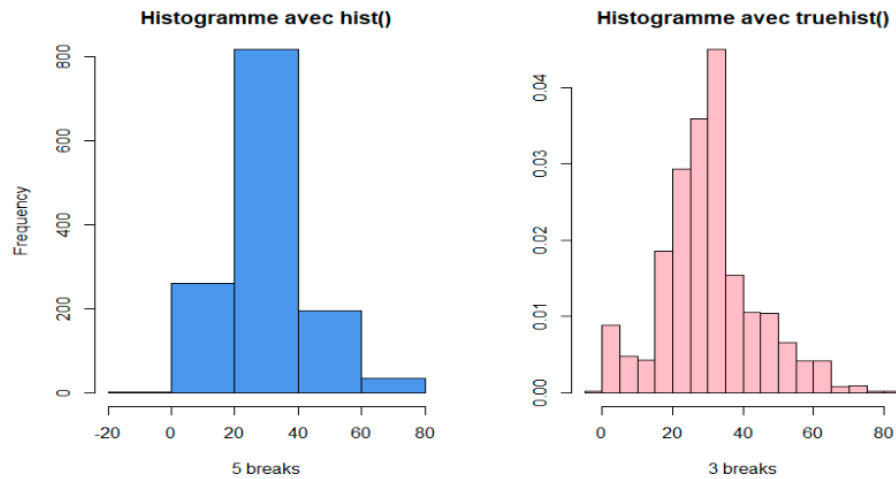


Figure 26: Histogram with hist() and truehist()

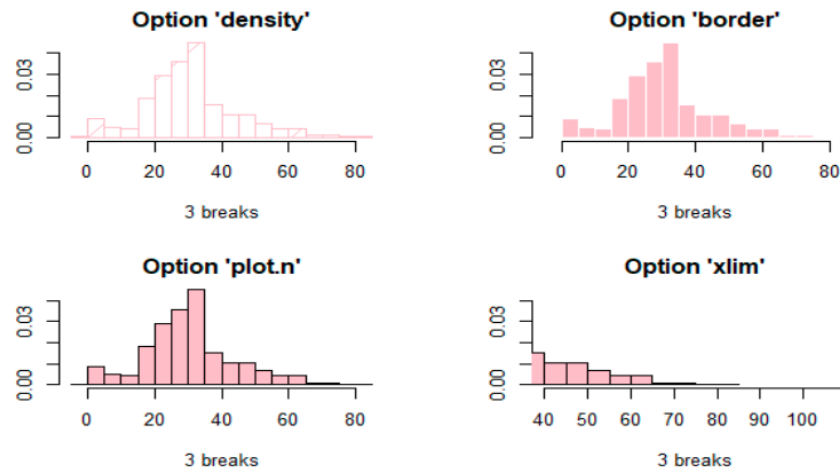


Figure 27: Histogram with varying options in truehist()

5.2 Estimation of density using the kernel

The density function provides a kernel density estimate of the density of observations. In other words, it estimates the probability density function of a random variable based on a set of observed data. It does this by placing a kernel function on each observation and summing these functions to estimate the underlying probability density function. The resulting density estimate can be plotted as a smooth curve that shows the shape of the underlying distribution.

The resulting plot shows the estimated density of the Age variable, which represents the probability density function of the variable. The x-axis represents the values of the variable, and the y-axis represents the estimated density values.

The code uses the default kernel function which is the Gaussian kernel, so We will vary the kernel argument now and get the following: The choice of kernel function in density estimation can have an impact on the shape and smoothness of the estimated density curve.

- Epanechnikov kernel: has a bell shape with a flat top. It assigns a higher weight to observations closer to the center of the kernel. It can produce a smoother estimate of the density compared to the rectangular kernel.
- Rectangular kernel: has a constant weight for observations within the bandwidth. It can produce

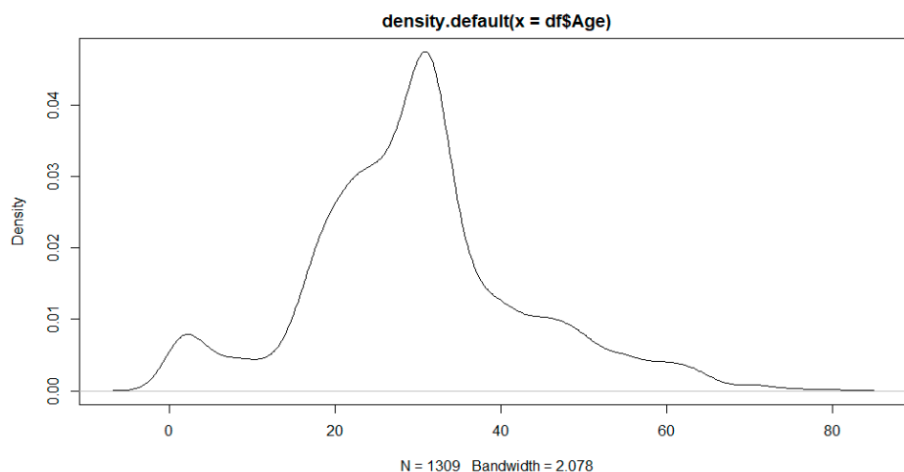


Figure 28: Distribution of the variable age with kernel function

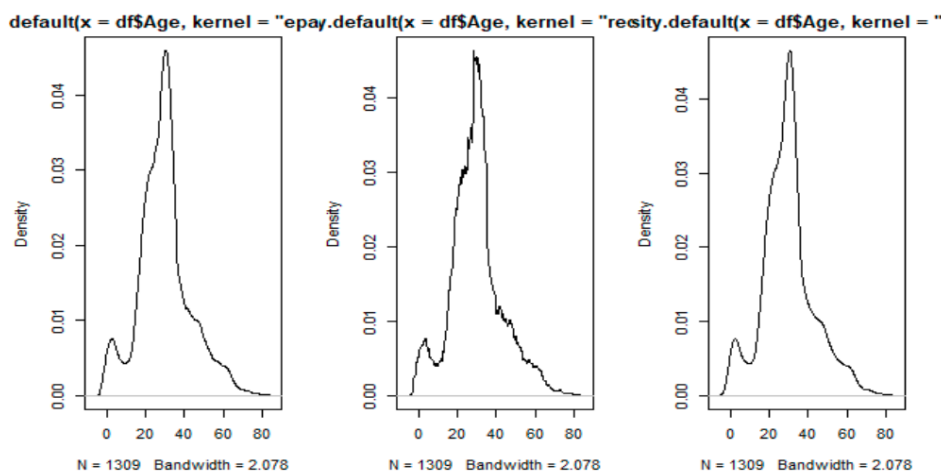


Figure 29: Distribution of the variable age with different kernel functions

a rough estimate of the density with high variability at the edges of the kernel.

- Cosine kernel: has a semi-circular shape with a maximum weight at the center of the kernel. It can produce a smoother estimate of the density than the rectangular kernel, but less smooth than the Epanechnikov kernel.

In Figure 28, the first plot uses the Epanechnikov kernel, which is a type of quadratic kernel that is more efficient than the Gaussian kernel when the sample size is large. The second uses the rectangular and the third uses the cosine kernel, which is a type of kernel that is symmetric and oscillating. Every shape of these density curves reflects the shape of the kernel function. The choice of kernel function ultimately depends on the desired level of smoothness in the estimated density.

Now, we will be varying instead the bandwidth (bw) argument which is used to control the smoothness of the density estimate. It determines the width of the kernel function and affects how much weight is given to nearby observations.

As we can see in figures 30 and 31, the density curve with a larger bw value 0.5 is smoother, while the one with a smaller bw value 0.2 is more jagged and has more detail. The choice of bw value depends on the data and the desired level of smoothness in the density estimate. A smaller bandwidth will result in a more detailed (less smooth) density estimate, which can be sensitive to local fluctuations in the data. A larger bandwidth will result in a smoother density estimate that is less sensitive to local fluctuations,

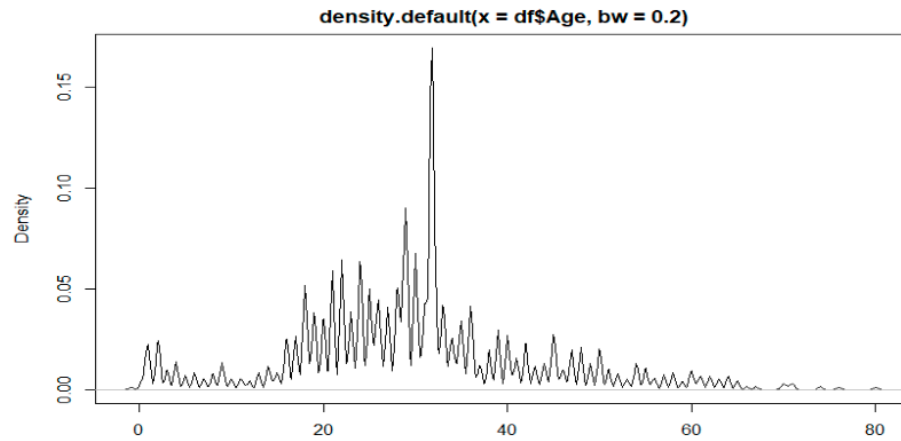


Figure 30: Distribution of the variable age with bw=0.2

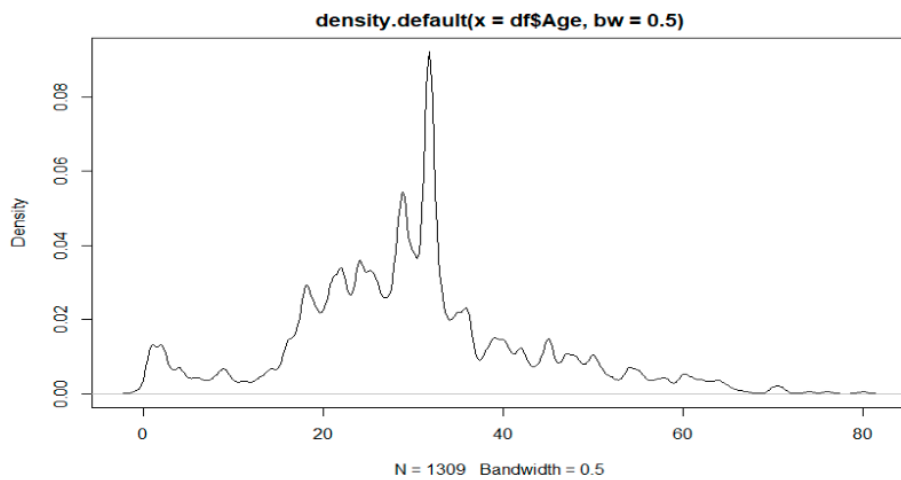


Figure 31: Distribution of the variable age with bw=0.5

but may miss important features of the data.

The optimal bandwidth value depends on the characteristics of the data and the purpose of the analysis. If the bandwidth is too small, the density estimate may overfit the data, while if it is too large, it may underfit the data. There are various methods for selecting the optimal bandwidth, such as the rule-of-thumb, cross-validation, and maximum likelihood. We considered the cross-validation in our project.

We consider a grid of values of $h \in [0, 8]$ and consider the sample as composed of 5 packets. For each value of h and each packet of observations, we want to calculate the Gaussian kernel estimator obtained for this value of h and using the observations outside the considered packet. Then, we want to estimate the error committed by this estimator on the observations of the considered packet. Next, we take the average these errors over the 5 packets. And finally, we select the value of h that gives the lowest average error.

5.3 Estimation using regression functions

In this part, we will estimate the density by regression. We will use different estimator; the first one is the Nadaraya-Watson estimator with different bandwidths using the `ksmooth` function using different bandwidths that are 0.1, 1, and 5.

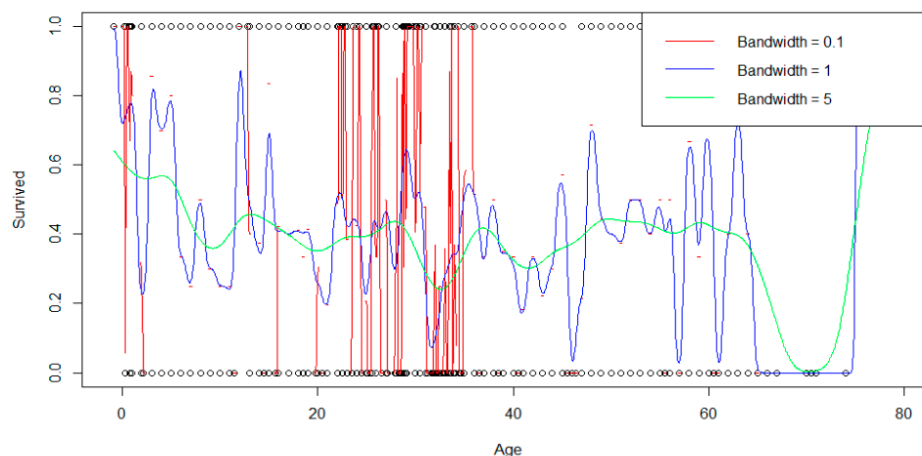


Figure 32: Nadaraya-Watson estimator with different bandwidths

These curves, in figure 33, show how the choice of bandwidth affects the smoothing of the data.

When the bandwidth is small 0.1, the red curve is very wiggly and follows the data points closely. This can lead to overfitting and capture noise in the data. When the bandwidth is larger 1, the blue curve is smoother and captures the trend in the data without fitting the noise. When the bandwidth is even larger 5, the green curve is even smoother and captures the overall trend in the data, but may miss some of the details and variations in the data.

Now, we will estimate the regression function using local polynomials. The resulting curve, in figure 34,

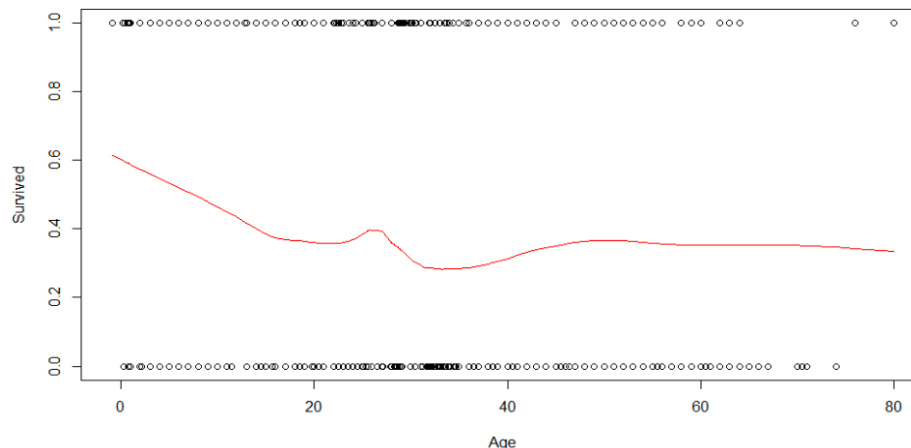


Figure 33: Estimate the regression function using local polynomials

shows a smooth fit to the data due to the fact that the local polynomial regression takes into account the curvature of the data, which results in a more flexible and accurate estimate. However, the choice of smoothing parameter f is important as too much smoothing can result in oversmoothing and too little smoothing can result in overfitting.

We now estimate the regression function using advanced local polynomials, with a smoothing parameter (span) of 0.75, which determines the degree of smoothing of the estimate.

We see a smooth curve that captures the general trend in the data, similar to the local polynomial estimator. The span parameter resulted in a moderate degree of smoothing.

Now, we use the cubic splines to estimate the regression function. The function `smooth.spline` is used

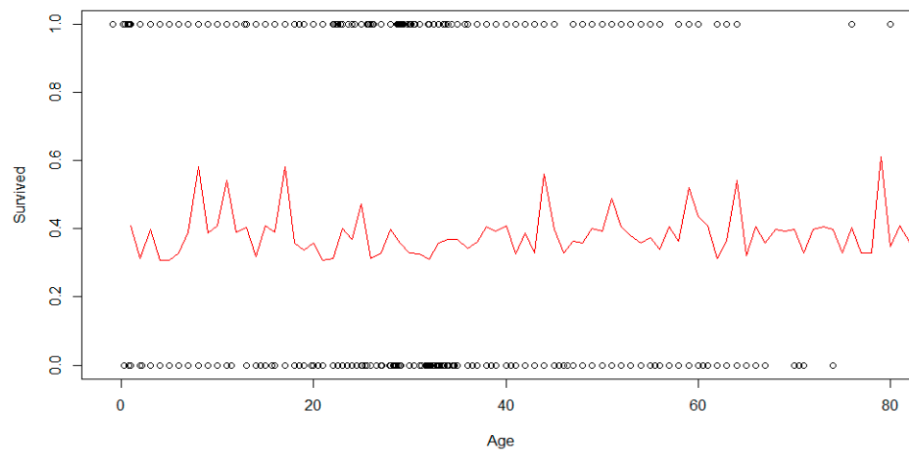


Figure 34: Estimate the regression function using advanced local polynomials

to fit the spline to the data, and the predict function is used to obtain the estimated values of the regression function on a grid of points. In this case, $df = 5$ specifies that the spline should have 5 degrees of freedom.

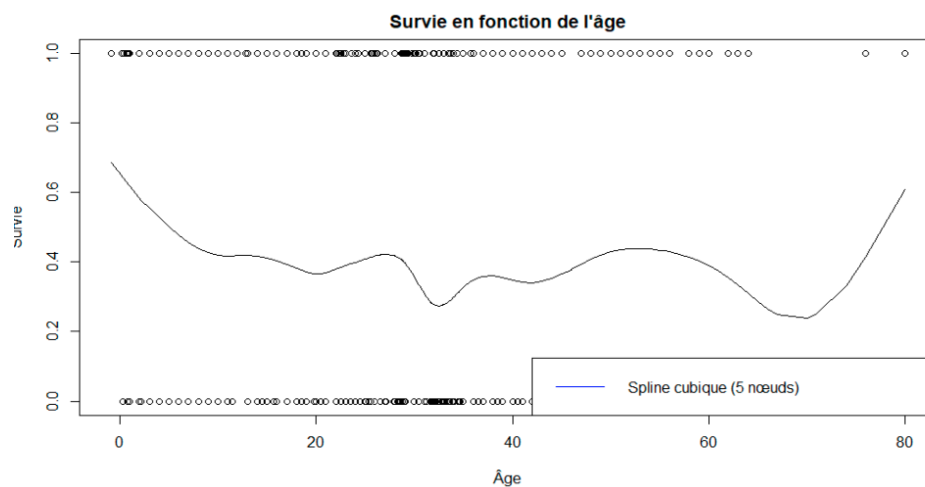


Figure 35: estimate the regression function using cubic splines

As we see in figure 36, the cubic spline with 5 knots fits the data well, capturing the general trend of the data without overfitting. As we increase the number of knots, the curve would become more flexible and fit the data more closely, but it may also start to capture noise in the data rather than the underlying trend. On the other hand, decreasing the number of knots could lead to an oversimplified model that does not capture important features of the data. In general, the choice of the number of knots depends on the complexity of the data.

We will compare the estimators by plotting each of the estimated curves on the same graph and compare their shapes and levels of smoothing.

The shapes of the curves differ slightly. The Nadaraya-Watson estimator with a normal kernel and bandwidth of 3 appears to have a smoother curve compared to the other three methods. The local polynomial regression using the lowess method seems to have more fluctuations and a less smooth curve. The local polynomial regression using the loess method with a span of 0.75 is smoother than the lowess method, but still shows some fluctuations. The cubic spline with 5 knots is also relatively smooth but has some sharp turns between the knots.

Overall, the choice of non-parametric regression method and tuning parameters should depend on the

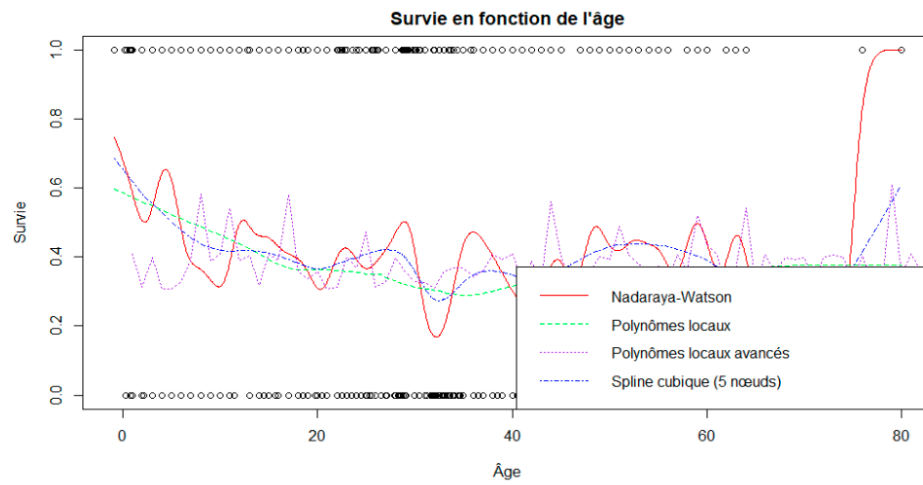


Figure 36: Comparison of the estimators

specific characteristics of the data and the research question.