Diagnosis of the model - Goodness of fit tests

Bachelor in Computer Science and Engineering

2020/21

1. Introduction

The aim of this practice is to assign a probability model to a sample dataset in such a way that the chosen model can represent the population from which the data was taken. The task of looking for the suitable model is denoted by **distribution fitting**. In order to select a good probability model for a given dataset it is necessary to make statistical tests. The task to execute these tests is called **diagnosis of the model**. Therefore we will say that a model **fits well** the data if our data sample will positively pass the tests of the **diagnosis**.

The usual way to perform the distribution fitting is the following. We start from a data sample and we compare its empirical distribution with the one of known models (Normal, Poisson, Exponential, etc). To evaluate the goodness-of-fit of a model we will make the Chi-squared test.

In the following we will use the data contained in the file `TiempoaccesoWeb.xlsx`. We start by analyzing the variable `Ordenador_Uni` in the file `TiempoAccesoWeb.xlsx`. This variable contains 55 measurements of times, measured in seconds, that are the times needed to access to the University UC3M's web page from a computer of its library. Starting from this set of data, we want to find a probability model that well describes the population of the accessing times necessary to access from a computer of the library the web page of the University UC3M. Afterwards we analyze the variable `tiempo` of the file `AlumnosIndustriales.xlsx` that contains measurements of the time spent by a group of students to get to the University.

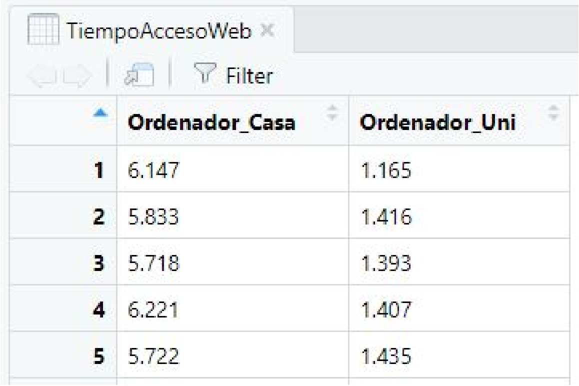
2. Model fitting. Variable `Ordenador_Uni`

2.1 Descriptive analysis of the data

The first thing to do is the descriptive analysis of the data (computing the characteristic measures and inspecting the histogram). In this way we could have a first idea of which model to use.

First we read and view the data file. The figure shows the first five observations of this datafile. Note that the line `View(TiempoAccesoWeb)` appears as a comment, to execute it, simply delete the symbol `#`.

```r
library(readxl)
TiempoAccesoWeb <- read_excel("TiempoAccesoWeb.xlsx")
#View(TiempoAccesoWeb)
```

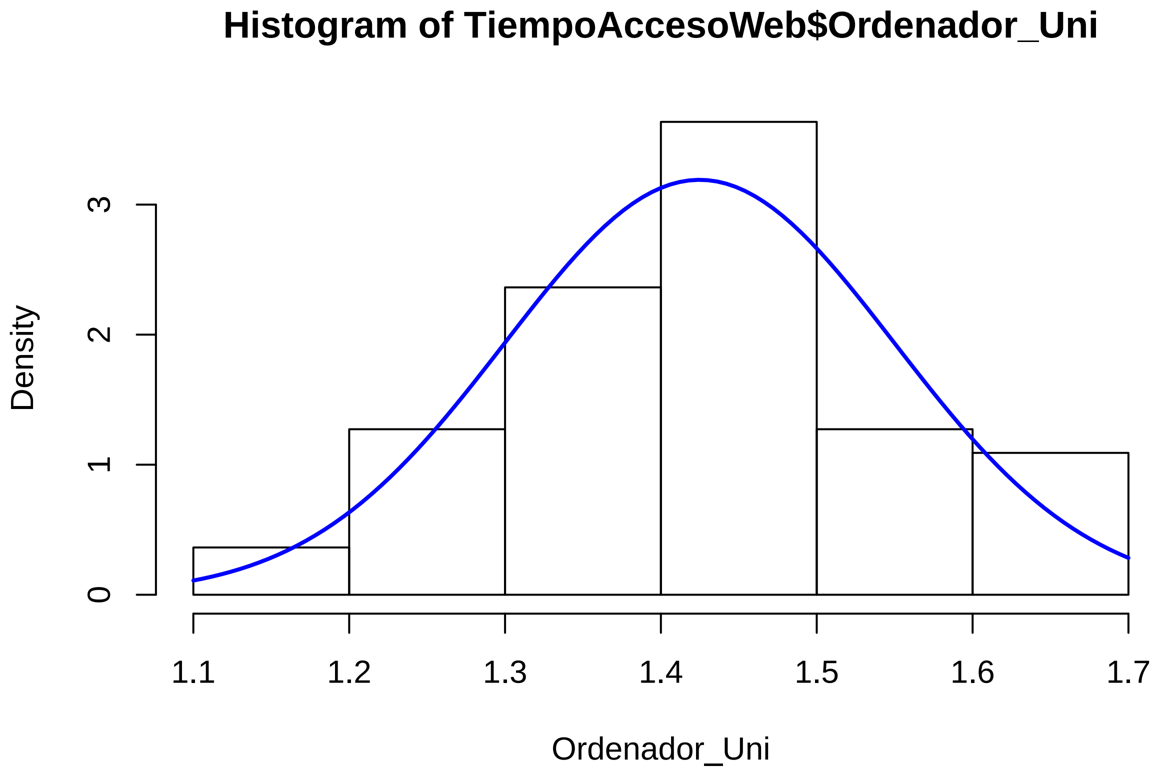


	Ordenador_Casa	Ordenador_Uni
1	6.147	1.165
2	5.833	1.416
3	5.718	1.393
4	6.221	1.407
5	5.722	1.435

```r
suppressWarnings(library(summarytools))
descr(TiempoAccesoWeb$Ordenador_Uni)
```

```
## Descriptive Statistics
## TiempoAccesoWeb$Ordenador_Uni
## N: 55
##
##                      Ordenador_Uni
## ----------------- ---------------
##              Mean            1.42
##           Std.Dev            0.13
##               Min            1.16
##                Q1            1.34
##            Median            1.42
##                Q3            1.50
##               Max            1.68
##               MAD            0.11
##               IQR            0.15
##                CV            0.09
##          Skewness            0.08
##       SE.Skewness            0.32
##          Kurtosis           -0.47
##           N.Valid           55.00
##         Pct.Valid          100.00
```

```r
hist(TiempoAccesoWeb$Ordenador_Uni,
     probability = TRUE, # histogram has a total area = 1
     xlab = "Ordenador_Uni")
curve(dnorm(x, mean(TiempoAccesoWeb$Ordenador_Uni), sd(TiempoAccesoWeb$Ordenador_Uni)),
      col="blue", lwd=2, add=TRUE, yaxt="n")
```



We can appreciate that the histogram looks like a Normal density function. Indeed it is unimodal and quite symmetric (`Skewness = 0.08`) but its bell is not exactly like the Gauss' one (`Kurtosis = -0.29`). From this we can deduce that a normal distribution could fit well our data and so it could be a good model for the population we are studying.

2.2 Diagnosis of the chosen model

To evaluate the goodness of the fitted model we can use the Chi-squared test. We should remember that the Chi-squared test is a discrepancy measure among the observed and expected number of observations in a given partition

$$\sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i},$$

where k is the number of intervals or cells in the partition, O_i is the number of observations that are in i-th cell and E_i is the expected number of observations in the same cell.

First, we must construct a partition of $\mathbb{R}$ and count how many values of `Ordenador_Uni` fall in each interval of the partition. An easy way is to use the partition obtained by the `hist` function

```
Partition <- hist(TiempoAccesoWeb$Ordenador_Uni, plot = FALSE)
Partition
```

```
## $breaks
## [1] 1.1 1.2 1.3 1.4 1.5 1.6 1.7
##
## $counts
## [1]  2  7 13 20  7  6
##
## $density
## [1] 0.3636364 1.2727273 2.3636364 3.6363636 1.2727273 1.0909091
##
## $mids
## [1] 1.15 1.25 1.35 1.45 1.55 1.65
##
## $xname
## [1] "TiempoAccesoWeb$Ordenador_Uni"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

The component `breaks` of `Partition` gives the points that define the intervals in the histogram. That is, the six intervals in the partition are $(1.1, 1.2]$, $(1.2, 1.3]$, $(1.3, 1.4]$, $(1.4, 1.5]$, $(1.5, 1.6]$ and $(1, 6, 1.7]$. The component `counts` gives the number of observations inside each interval or cell. This are the **observed**, O_i.

It should be noted that the above partition does not cover all $\mathbb{R}$ since intervals $(-\infty, 1.1]$ and $(1.7, +\infty)$ are not considered. We will assume that the first interval of the partition is $(-\infty, 1.2]$ and the last interval is $(1.6, +\infty)$.

Next, we fit the normal model to `Ordenador_Uni`

```
library(fitdistrplus)
normalfit <- fitdist(TiempoAccesoWeb$Ordenador_Uni, "norm")
normalfit
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##       estimate Std. Error
## mean 1.4248182 0.01670598
## sd   0.1238948 0.01180945
```

The estimated parameters for the Normal random variable are in our case $\widehat{\mu} = 1.42481818$ and $\widehat{\sigma} = 0.12389484$ that are equal to the corresponding values shown in the descriptive analysis of the variable. Therefore the fitted model is

$$X \sim \mathcal{N}(1.42481818, 0.12389484).$$

Finally, we perform a diagnosis test to appreciate the goodness of our fitting. We should calculate the expected number of observations under the *fitted* normal distribution

```
CummulativeProbabilities = pnorm(c(-Inf, Partition$breaks[c(-1,-7)], Inf),
                           normalfit$estimate[1], normalfit$estimate[2])
Probabilities = diff(CummulativeProbabilities)
Expected = length(TiempoAccesoWeb$Ordenador_Uni)*Probabilities
chisq.test(Partition$counts, p = Probabilities)
```

```
## Warning in chisq.test(Partition$counts, p = Probabilities): Chi-squared
## approximation may be incorrect

##
##  Chi-squared test for given probabilities
##
## data:  Partition$counts
## X-squared = 2.625, df = 5, p-value = 0.7576
```

The result of the Chi-squared test can be resumed in the following three quantities

- The calculated test statistic, `X-squared` $= \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$, where o_i is the number of observations in the sample that are in i-th cell and e_i is the expected number of observations in the same cell.

  This statistic summarizes the relation between the histogram and the continuous curve of the density function. The bigger is its value the worse is the goodness of the fit of the chosen theoretical model.

- `df` (degrees of freedom), represents the parameter of the selected Chi-squared distribution and it is used as a reference point to appreciate the quality of the fitting.

  - The degrees of freedom at the `chisq.test` function are computed as `df` $= k - 1$ since it does not takes into account the number of estimated parameters.

  - The degrees of freedom must be computed as `df` $= k - p - 1$, where p is the number of unknown parameters of the model that are estimated using the data sample, in this case it is equal to 2 (the mean and the variance).

- `p-value` is the probability that the test statistic takes a value higher than `X-squared`. In this case it is given by the value of the area of the right-tail starting from 2.625 calculated with the density function of a Chi-squared distribution with `df` degrees of freedom.

  - Notice that `df = 5` corresponds to number of cells minus one, $k - 1$, but we estimate two parameters, so we should to use a χ^2 distribution with `df = 3`, $k - p - 1$.

  ```
  pchisq(2.5646, 3, lower.tail = FALSE)
  ```

  ```
  ## [1] 0.4637294
  ```

  That is, the correct `p-value` $= 0.4637294$.

If the `p-value` is less than 0.05 we assume that it is quite improbable to obtain the resulting value of the test statistic if the model were good. Therefore we conclude that the test is unsatisfactory. On the other hand if the `p-value` is bigger than 0.05 we conclude that the fit is relatively good and that the chosen model can be considered reasonable to represent the population.

In our case the pvalue is equal to 0.4637294 and therefore we conclude that the normal model is a reasonable model to represent our population.

2.3 Other normality goodness-of-fit tests

The chi-square test is usually not recommended for testing the hypothesis of normality due to its inferior power properties compared to other tests. There are many functions in R to make various different goodness-of-fit tests. All of them may be interpreted by looking at the p-values in the same way we have done by looking at the Chi-squared test. In particular, the package `nortest` includes the following:

- `ad.test`: Anderson-Darling test
- `cvm.test`: Cramer-von Mises test
- `lillie.test`: Kolmogorov-Smirnov-Lilliefors test
- `pearson.test`: Pearson chi-square test for normality
- `sf.test`: Shapiro-Francia test

For example, it is possible to check that the p-values corresponding to these tests are bigger than 0.05 too, thus corroborating our selection of the Normal model.

```
library(nortest)
ad.test(TiempoAccesoWeb$Ordenador_Uni)
```

```
##
##  Anderson-Darling normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## A = 0.4312, p-value = 0.2958
```

```
cvm.test(TiempoAccesoWeb$Ordenador_Uni)
```

```
##
##  Cramer-von Mises normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## W = 0.073781, p-value = 0.2447
```

```
lillie.test(TiempoAccesoWeb$Ordenador_Uni)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## D = 0.088043, p-value = 0.3582
```

```
pearson.test(TiempoAccesoWeb$Ordenador_Uni)
```

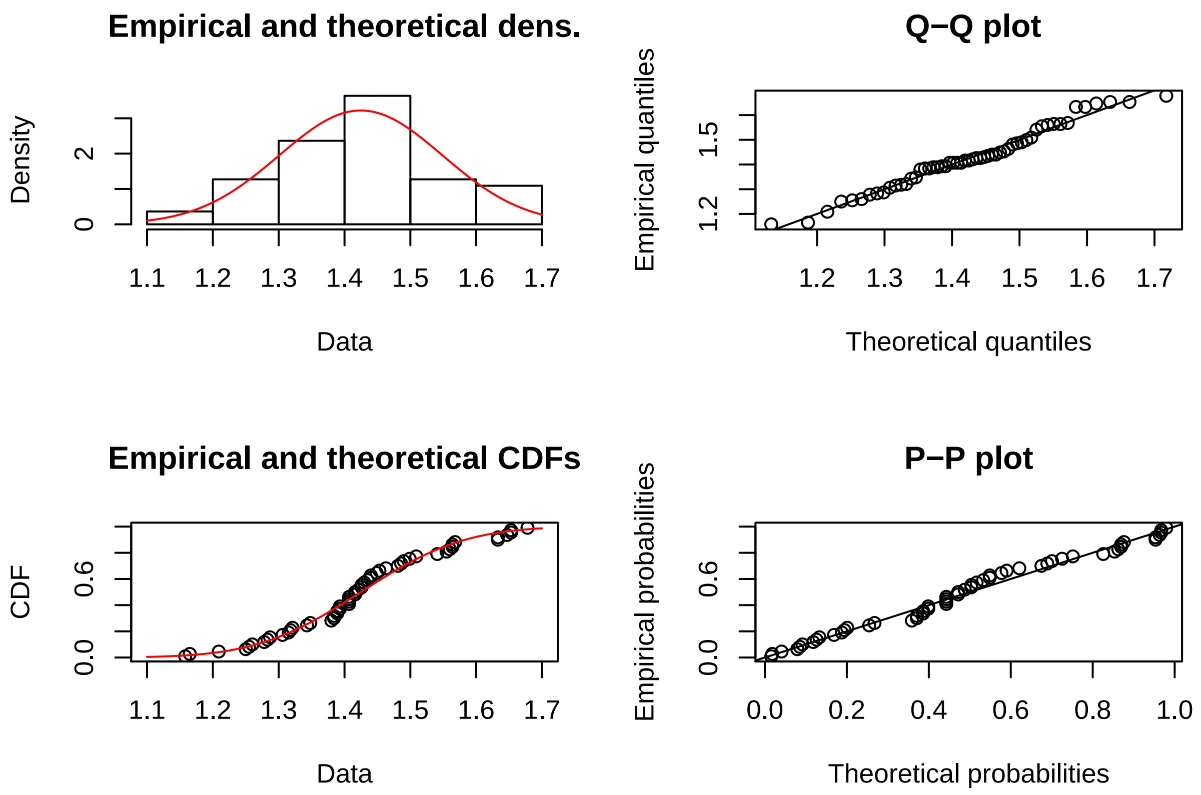
```
##
##  Pearson chi-square normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## P = 5.9091, p-value = 0.5504
```

```
sf.test(TiempoAccesoWeb$Ordenador_Uni)
```

```
##
##  Shapiro-Francia normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## W = 0.98159, p-value = 0.4749
```

Also, it is possible to obtain a graphical representation of the fitting by

```
plot(normalfit)
```



3. Model fitting for the variable `tiempo`

In this section we repeat the above analysis for the variable `tiempo` at file `AlumnosIndustriales.xlsx`. This variable contains measurements of the time spent by a group of students to get to the University. The sample size is equal to 95.

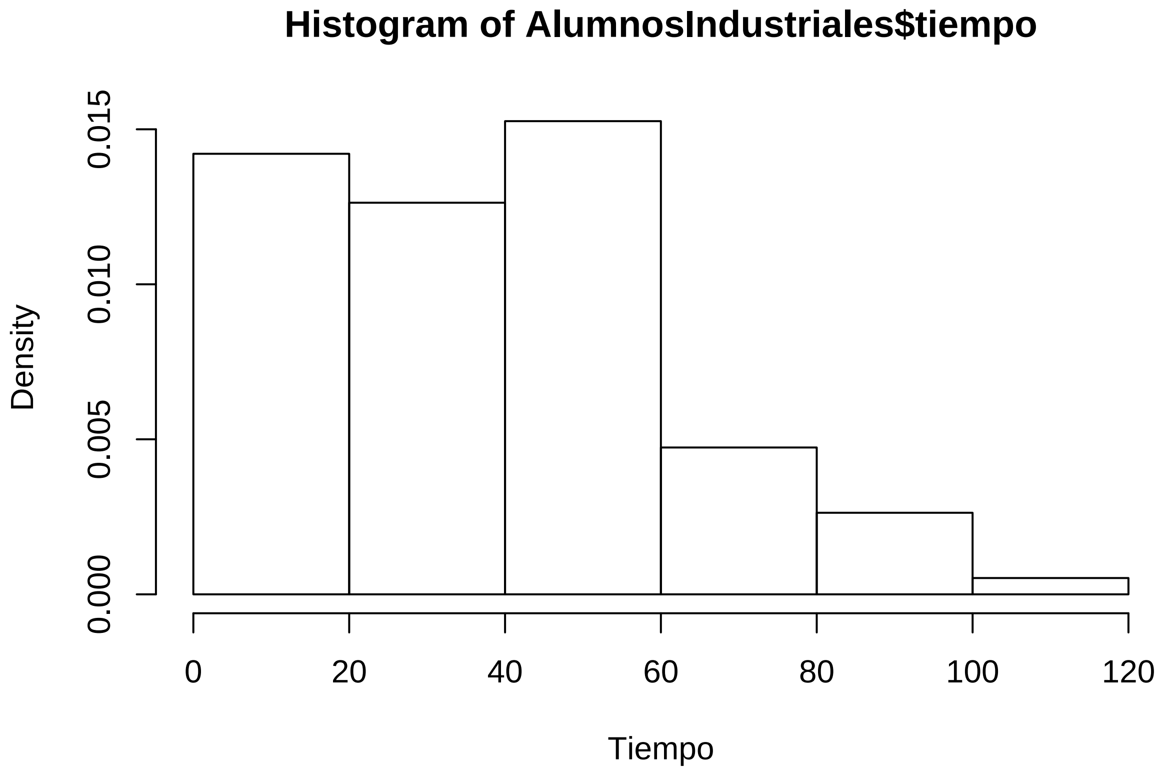
3.1 Descriptive analysis of data

After loading the file `AlumnosIndustriales.xlsx`, we perform the descriptive analysis of the variable `tiempo` (computing the characteristic measures and inspecting the histogram).

```
suppressWarnings(library(summarytools))
descr(AlumnosIndustriales$tiempo)
```

```
## Descriptive Statistics
## AlumnosIndustriales$tiempo
## N: 95
##
##                       tiempo
## ----------------- --------
##              Mean    41.42
##           Std.Dev    24.74
##               Min     1.00
##                Q1    20.00
##            Median    40.00
##                Q3    60.00
##                 Max   120.00
##                 MAD    29.65
##                 IQR    40.00
##                  CV     0.60
##            Skewness     0.63
##         SE.Skewness     0.25
##            Kurtosis    -0.04
##             N.Valid    95.00
##           Pct.Valid   100.00
```

```
hist(AlumnosIndustriales$tiempo,
     probability = TRUE, # histogram has a total area = 1
     xlab = "Tiempo")
```



The data looks unimodal and with positive asymmetry. We have two options to fit a model to these data. First we try to fit a model that has positive asymmetry, like for example the Weibull distribution or the Lognormal distribution. Next we will try to make a transformation of the data in order to correct the asymmetry and to try to fit a Normal distribution. For example, we could try to apply a square root operation (note that to fit a Normal to the logarithm of a variable is the same as to fit a Lognormal distribution to the variable with no transformation).

3.2 Fitting a Weibull distribution

As in the previous example, we fit the model

```
library(fitdistrplus)
weibullfit <- fitdist(AlumnosIndustriales$tiempo, "weibull")
weibullfit
```

```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## shape  1.708639  0.1393375
## scale 46.341096  2.9242445
```

Now, we will obtain the observed and the expected number of observations in the intervals defined by the default histogram.

```
Partition <- hist(AlumnosIndustriales$tiempo, plot = FALSE)
Partition
```

```
## $breaks
## [1]   0  20  40  60  80 100 120
##
## $counts
## [1] 27 24 29  9  5  1
##
## $density
## [1] 0.0142105263 0.0126315789 0.0152631579 0.0047368421 0.0026315789
## [6] 0.0005263158
##
## $mids
## [1]  10  30  50  70  90 110
##
## $xname
## [1] "AlumnosIndustriales$tiempo"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
CummulativeProbabilities = pweibull(c(Partition$breaks[-7], Inf),
                      weibullfit$estimate[1], weibullfit$estimate[2])
Probabilities = diff(CummulativeProbabilities)
Expected = length(AlumnosIndustriales$tiempo)*Probabilities
chisq.test(Partition$counts, p = Probabilities)
```

```
## Warning in chisq.test(Partition$counts, p = Probabilities): Chi-squared
## approximation may be incorrect
```

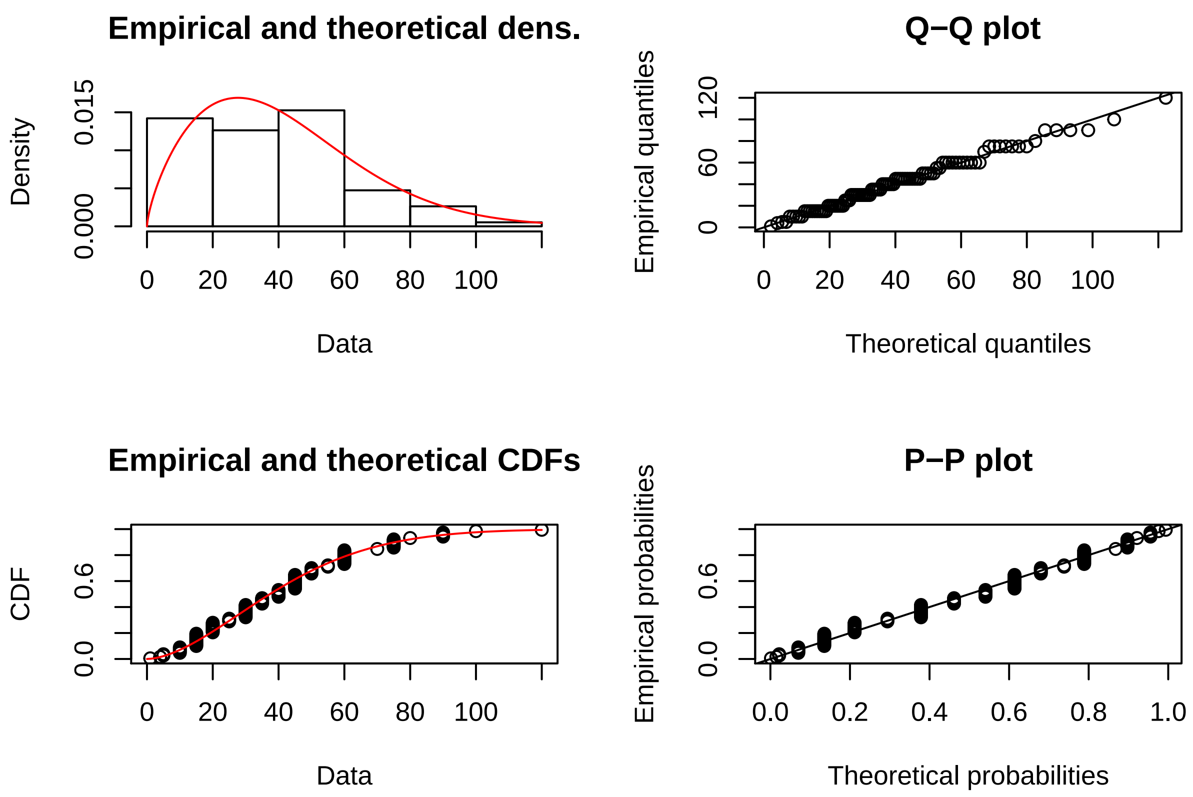
```
##
##  Chi-squared test for given probabilities
##
## data:  Partition$counts
## X-squared = 7.0387, df = 5, p-value = 0.2178
```

Here, again, we should to re-calculate the `p-value`since we estimate the two parameters of the Weibull distribution.

```
pchisq(7.0467, 3, lower.tail = FALSE)
```

```
## [1] 0.07042409
```

```
plot(weibullfit)
```



From a comparison of histogram with the Weibull density function and from looking at the p-value we realize that the fit is satisfactory. This means that we could use the Weibull probability model to describe the time spent by the students to get to the University.

3.3 Fitting a Lognormal distribution

We proceed as before: (i) model fitting; (ii) calculation of the observed and expected number of observations at each interval in the histogram and (iii) Chi-squared test.

```
library(fitdistrplus)
lognormalfit <- fitdist(AlumnosIndustriales$tiempo, "lnorm")
lognormalfit
```

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##          estimate Std. Error
## meanlog 3.4891976 0.08090337
## sdlog   0.7885485 0.05720691
```

```
Partition <- hist(AlumnosIndustriales$tiempo, plot = FALSE)
Partition
```

```
## $breaks
## [1]   0  20  40  60  80 100 120
##
## $counts
## [1] 27 24 29  9  5  1
##
```

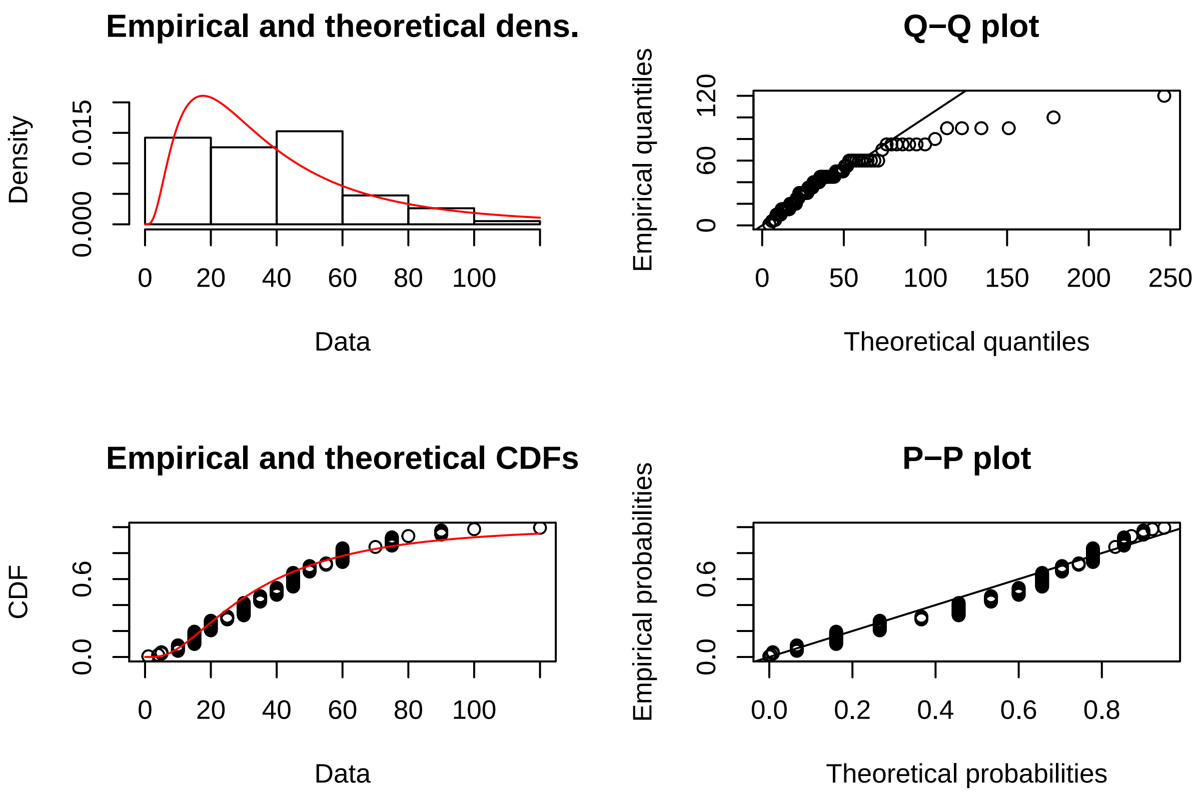
```
## $density
## [1] 0.0142105263 0.0126315789 0.0152631579 0.0047368421 0.0026315789
## [6] 0.0005263158
##
## $mids
## [1]  10  30  50  70  90 110
##
## $xname
## [1] "AlumnosIndustriales$tiempo"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
CummulativeProbabilities = plnorm(c(Partition$breaks[-7], Inf),
                      lognormalfit$estimate[1], lognormalfit$estimate[2])
Probabilities = diff(CummulativeProbabilities)
Expected = length(AlumnosIndustriales$tiempo)*Probabilities
chisq.test(Partition$counts, p = Probabilities)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Partition$counts
## X-squared = 16.15, df = 5, p-value = 0.00643
```

It looks clear that this fitting is no such good as the one before. The p-value obtained by the Chi-squared test is very low. In fact, the `p-value` is smaller since we should use `pchisq(16.15, 3, lower.tail = FALSE)`.

```
plot(lognormalfit)
```



The histogram gives us the reason of the bad fit; indeed the Lognormal distribution has a higher kurtosis than the dataset. In conclusion the Lognormal model is not good to represent our data.

3.4 Fitting a Normal distribution to a transformation of the dataset

The variable `tiempo` is positive asymmetric, however its square-root looks quite symmetric. If we fit a Normal distribution to the square-root of the data we get the following results:

```r
library(fitdistrplus)
normalfit <- fitdistr(sqrt(AlumnosIndustriales$tiempo), "normal")
normalfit
```

```
##      mean          sd    
##   6.1169314   2.0010506 
##  (0.2053035) (0.1451715)
```

```r
Partition <- hist(sqrt(AlumnosIndustriales$tiempo), plot = FALSE)
Partition
```

```
## $breaks
##  [1]  1  2  3  4  5  6  7  8  9 10 11
## 
## $counts
##  [1]  2  2 15 11 15 17 18  9  5  1
## 
## $density
##  [1] 0.02105263 0.02105263 0.15789474 0.11578947 0.15789474 0.17894737
##  [7] 0.18947368 0.09473684 0.05263158 0.01052632
```

```
##
## $mids
##  [1]  1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5  9.5 10.5
##
## $xname
## [1] "sqrt(AlumnosIndustriales$tiempo)"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
CummulativeProbabilities = pnorm(c(-Inf, Partition$breaks[c(-1,-11)], Inf),
                      normalfit$estimate[1], normalfit$estimate[2])
Probabilities = diff(CummulativeProbabilities)
Expected = length(AlumnosIndustriales$tiempo)*Probabilities
chisq.test(Partition$counts, p = Probabilities)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Partition$counts
## X-squared = 9.3823, df = 9, p-value = 0.4028
```

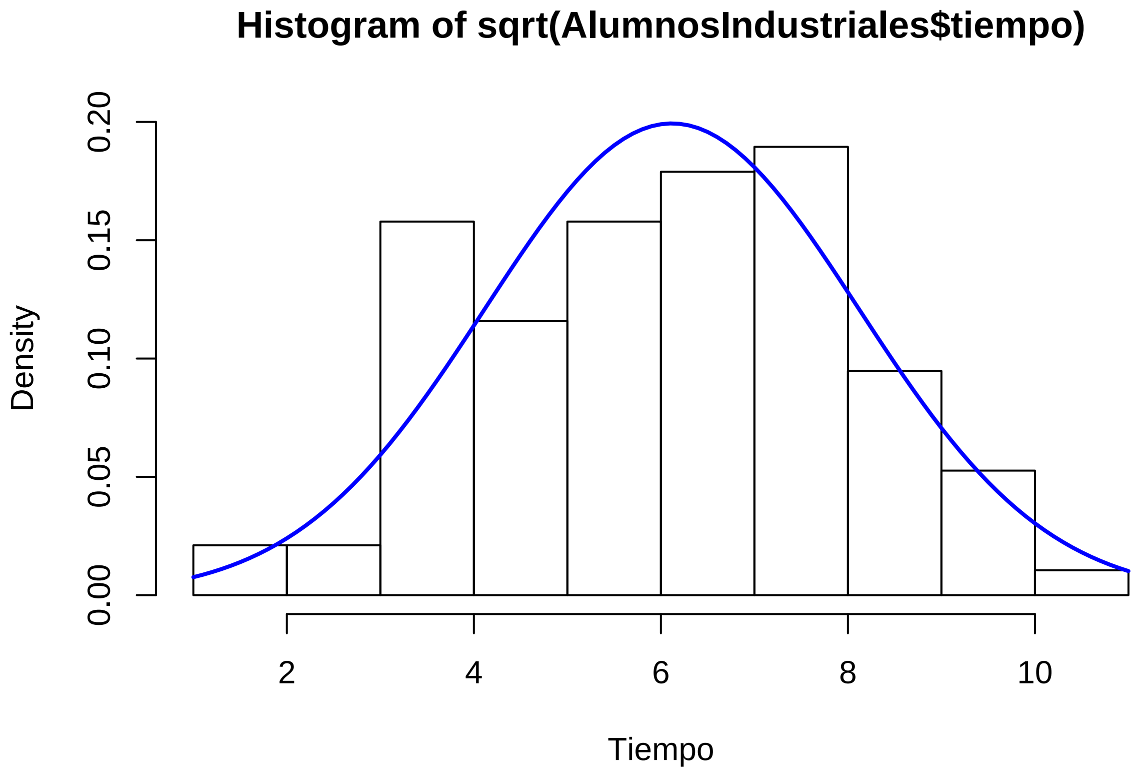
The `p-value` taking into account that two parameters were estimated is

```
pchisq(9.3823, 7, lower.tail = FALSE)
```

```
## [1] 0.226361
```

which is bigger than 0.05.

```
hist(sqrt(AlumnosIndustriales$tiempo),
          probability = TRUE, # histogram has a total area = 1
     xlab = "Tiempo", ylim = c(0,0.2))
curve(dnorm(x, normalfit$estimate[1], normalfit$estimate[2]),
      col="blue", lwd=2, add=TRUE, yaxt="n")
```



The fitting looks almost as good as the one done by using the Weibull distribution.

We can check the above results by the normality tests mentioned in section 2.3:

```
library(nortest)
ad.test(sqrt(AlumnosIndustriales$tiempo))
```

```
##
##  Anderson-Darling normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## A = 0.52436, p-value = 0.1773
```

```
cvm.test(sqrt(AlumnosIndustriales$tiempo))
```

```
##
##  Cramer-von Mises normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## W = 0.086902, p-value = 0.1664
```

```
lillie.test(sqrt(AlumnosIndustriales$tiempo))
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## D = 0.078749, p-value = 0.1562
```

```r
pearson.test(sqrt(AlumnosIndustriales$tiempo), n.classes = 10)
```

```
##
##  Pearson chi-square normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## P = 10.368, p-value = 0.1686
```

```r
sf.test(sqrt(AlumnosIndustriales$tiempo))
```

```
##
##  Shapiro-Francia normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## W = 0.98791, p-value = 0.458
```

4. Example of an application of the goodness-of-fit test

To have a good model that represents the population from which we may have obtained a data sample is very useful. It allows, among other things, to compute the probabilities of events in a way much more precise than using the observed relative frequencies of the sample dataset.

In this example we compute the probability that a student lives at a distance of more than one hour from the University. We can do this by using the Weibull model as well as by using the Normal model applied to the square root of the variable `tiempo`. These two models will give us two different results, however we expect them to be very close to each other.

4.1 Computation using the Weibull model

As we have seen above, the Weibull that better fits our data has the following parameters: `shape` = 1.7088275 and `scale` = 46.3508101. Then, we can calculate the required probability, $\Pr(Tiempo > 60)$, by

```r
pweibull(60, shape = 1.7088275, scale = 46.3508101, lower.tail = FALSE)
```

```
## [1] 0.2113264
```

We can conclude that the probability that a student lives at a distance of more than one hour from the University is approximately equal to 0.211.

4.2 Computation using the Normal model applied to the square-root of the variable

As seen above, the square root can be well fitted to a Normal distribution. To compute the probability that the student takes more than 60 minutes to get at the University it is equivalent to compute the probability that the square root of the spent time is more than $\sqrt{(60)} = 7.745967$ (measured as square-root of minutes). The Normal distribution that better fits our data has the following estimated parameters: `mean` = 6.1169314 and `sd` = 2.0010506.

We can then compute the required probability for this distribution by

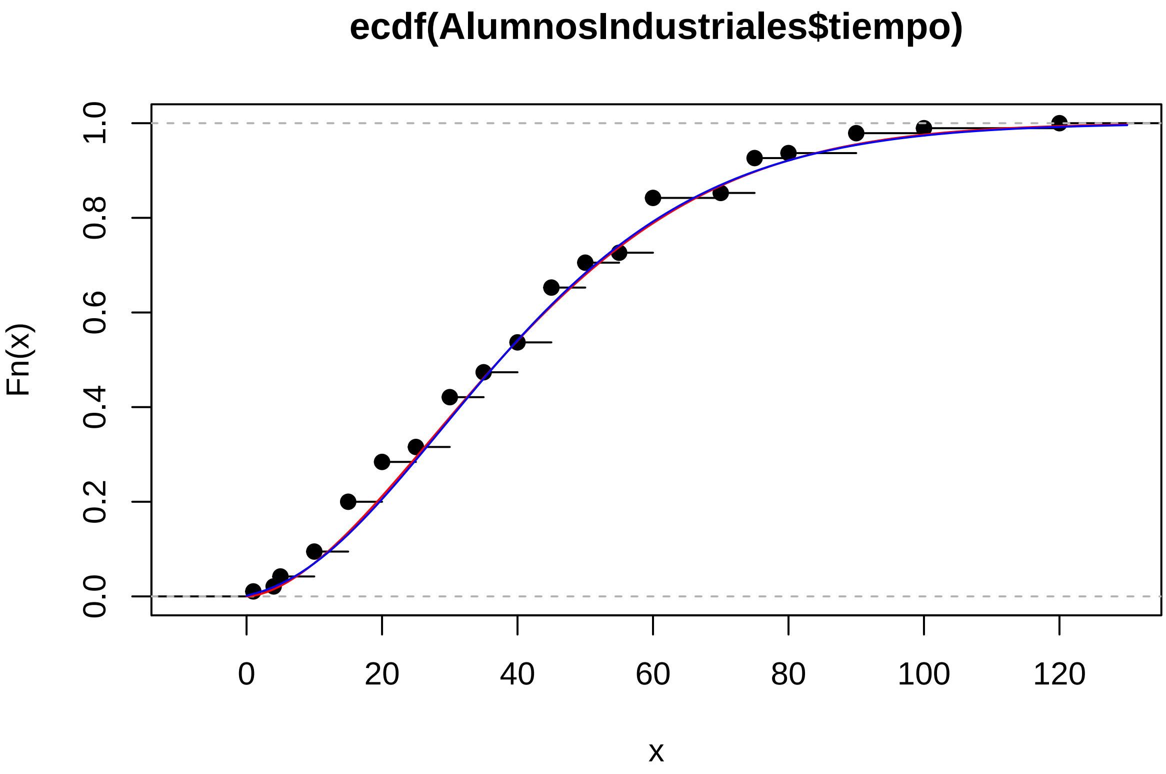
```r
pnorm(sqrt(60), mean = 6.1169314, sd = 2.0010506, lower.tail = FALSE)
```

```
## [1] 0.2077967
```

Therefore, by using this model, the probability that a student lives at a distance of more than one hour from the University is approximately equal to 0.208, and it is very close to the computed by using the Weibull model.

The following graph provides a comparison of the estimated distribution function using the Weibull model (in red) and the Normal model (in blue). It is clear that both models are very similar, and reasonably fit the empirical distribution function.

```r
plot(ecdf(AlumnosIndustriales$tiempo))
lines(0:130, pweibull(0:130, shape = 1.7088275, scale = 46.3508101), col="red")
lines(0:130, pnorm(sqrt(0:130),  mean = 6.1169314, sd = 2.0010506), col="blue")
```


ecdf(AlumnosIndustriales$tiempo)
Fn(x)
x