The University of Franche-Comté

Master CompuPhys - Statistical exploitation of measurements

χ^2 test and nonlinear of Paris-Alger fly data

Chiari Even
Legrand Maxime
M1 Compuphys 2022

Supervisor : Fabrice Devaux

December 9, 2022

Contents

1 Introduction

The main objective of this practical work is to study a set of data by two approaches, one by a direct approach (χ^2 test) in a first time, and the other by taking the inverse problem, through a non-linear square fitting. We'll then compare both method to see if they lead to the same results.

2 Our data

We want to fit the data of Paris-Alger fly, we have :

T	1.90 - 1.95	1.95 - 2.00	2.00 - 2.05	2.05 - 2.10	2.10 - 2.15	2.15 - 2.20	2.20 - 2.25	2.25 - 2.30	2.30 - 2.35	2.35 - 2.40	2.40 - 2.45	2.45 - 2.50	2.50 - 2.55
N	19	19	39	48	87	94	104	92	57	44	28	26	13

Figure 1: Data of Paris-Alger fly

On this figure we can see the time classes in row T and the number of flights in those classes in row N.

3 χ^2 test

The aim of this first part is to perform a χ^2 test on a set of statistics about the duration of a Paris-Alger flight. In a first time one will compare the data to the Normal law, either by plotting their distributions or their CDF. Then we'll perform the χ^2 test and conclude about the nature of our data.

By processing the data with Python, one obtains the mean and STD :

- $\overline{T} = 2.216$ h
- $\sigma(T) = 0.135$

3.1 Comparison to the Normal law

By plotting the density distribution of the data and considering time slots as their average, one can compare it to the Normal law of same parameters as described above with respect to data bins.

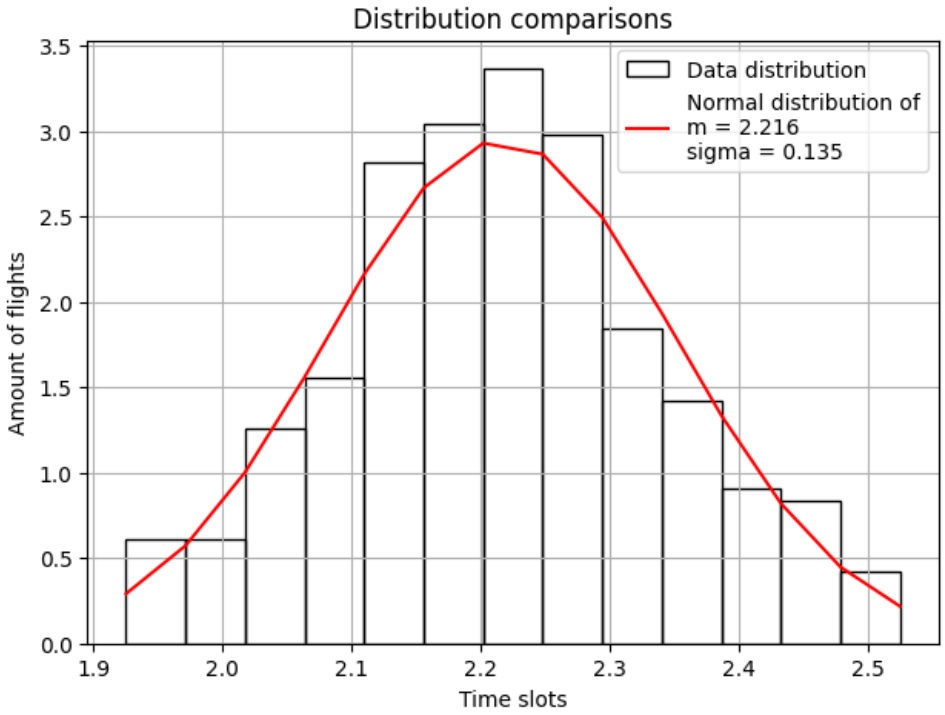

Figure 2: Comparison between Normal and data density distributions

We can also see it from another angle, by comparing their CDF :

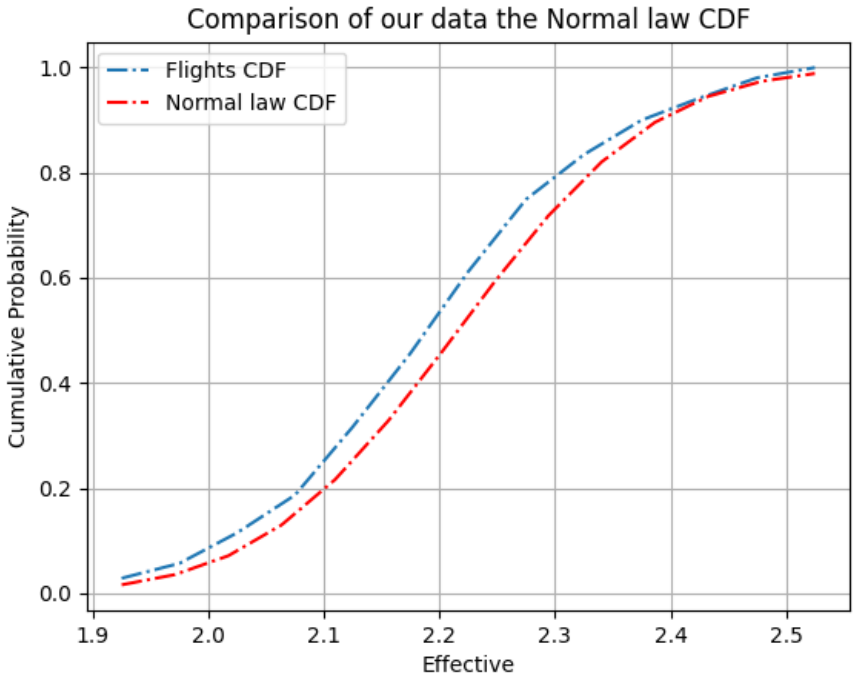

Figure 3: Comparison between Normal and data density distributions

By graphically analyzing these two graphs, one tends to say that the data may follow the Normal distribution, now let's compute the χ^2 to infirm or confirm the trend.

3.2 D^2 and conclusion on the nature of data

In order to compute the D^2 coefficient, one can proceed by normalizing the data from the Normal probability distribution $\mathcal{N}(T)$ to our data. It gives :

$$D^2 = \sum_{j=1}^{13} \frac{(N_j - NP_j^{exp})^2}{NP_j^{exp}} \tag{1}$$

With $N = 670$ the total amount of flights

After computation, we obtain $D^2 = 25.987$. By comparing it to a χ^2 table for 12 DDOF, one can conclude that the data doesn't follow the Normal law as it's outside the associated 95% confidence interval.

Degrees of freedom (*df*)	.99	.975	.95	.9	.1	.05	.025	.01
1	-------	0.001	0.004	0.016	2.706	3.841	5.024	6.635
2	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210
3	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345
4	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277
5	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086
6	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812
7	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475
8	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090
9	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666
10	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209
11	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725
12	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217
13	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688

Figure 4: Comparison between Normal and data density distributions

Now let's compare the results with a fit of the data before discussing it.

4 Non-linear fit of data

Fitting a curve is very useful in many domains as it allows us to predict the behavior of data. Starting from a scatter plot, one can model a curve on those points to get a fit, and after study it to do prediction on other results or understand the behavior of the scatter plot.

When you have a linear scatter plot this is easier to fit with a line because you have a simple type of equation and already known (or at least intuitive), for a non-linear curve it can quickly become a tricky process because many equations can fit your data (exponential, limited development, polynomial, Gaussian etc..). The more complex is the equation (amount of coefficients to adjust), the more the computation can become really long.

In our case we'll try to fit a Gaussian curve :

$$F(t) = a * \exp \frac{-(t-b)^2}{2c^2} \tag{2}$$

with:

- t is our variable
- a is the amplitude of our curve
- b is the average of our data
- c is the standard deviation

Here we need to compute a, b and c, we will get these coefficients through the algorithm of Gauss-Newton. This algorithm is based on the method of least-squares but expended to the case of non-linear curves.

The core of this algorithm is to minimise a sum of squared functions (the function that we choose in order to plot our data). After this we iterate our sum to compute it, this step of iteration is complex to calculate it as we need to define Jacobian matrix and normal equations.

When we get this iteration step we can minimise our sum to have a huge accuracy on our coefficients.

Fortunately we won't code this function here, instead we'll use a built-in function from the Scipy.optimise library.

4.1 Computation

The processed data here is the same than in part 1. Let's have a look at the plot this data :

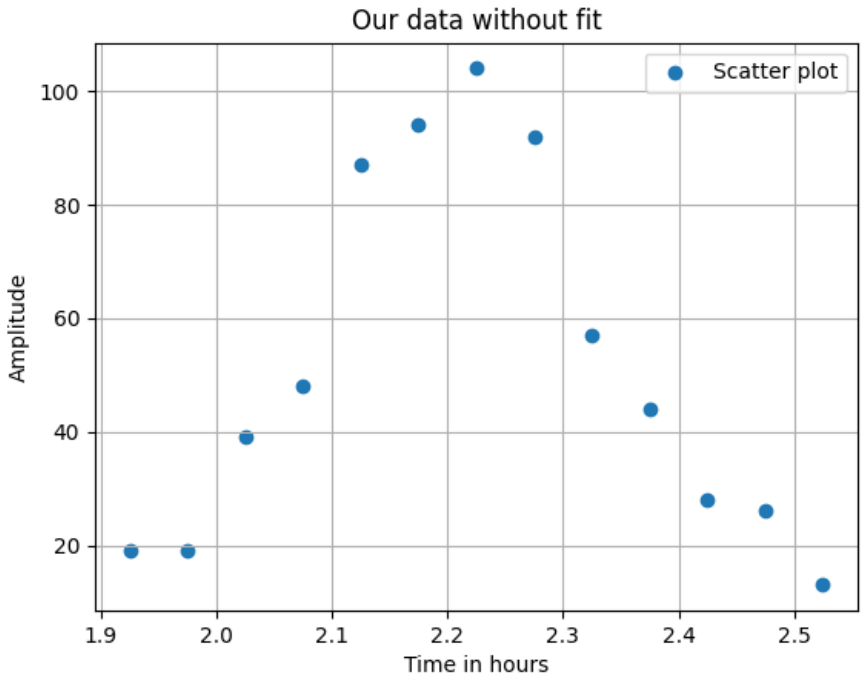

Figure 5: Scatter plot

On this graph we can see that the data seems to follows a Gaussian curve, so we want to build:

- the fit of this data with scipy.optimize based on Gauss-Newton method
- the theoretical curve, ie. the Normal distribution with our data STD and mean

This function takes in argument the function that we want to fit and the data that we want to fit:

```
fit_coef, pcov = curve_fit(GAUSS, T_DATA, Y_DATA) # use of curve.fit
```

It returns three coefficients for the Gaussian, a the amplitude, b the average and c the standard deviation.
This function also returns the uncertainty on these parameter called "pcov". In python, the function is defined as follows:

```
def GAUSS(x,a,b,c):
    return a*np.exp((-(x-b)**2)/(2*c**2))
```

Now we can compare the theoretical coefficients and the coefficients returned by curves.fit :

The theoretical coefficients : Amplitude $a = 104.0$, Average $b = 2.216$, Standard deviation $\sigma = 0.135$
The curves.fit coefficients : Amplitude $a = 98.579 \pm 4.47$, Average $b = 2.211 \pm 0.007$, Standard deviation $\sigma = 0.135 \pm 0.007$

Below is the computed coefficients correlation matrix for :

$$\begin{pmatrix} 1.99852930 \times 10^{1} & 1.35995614 \times 10^{-4} & -1.88213489 \times 10^{-2} \\ 1.35995614 \times 10^{-4} & 4.96448371 \times 10^{-5} & -4.54491018 \times 10^{-7} \\ -1.88213489 \times 10^{-2} & -4.54491018 \times 10^{-7} & 5.19064081 \times 10^{-5} \end{pmatrix}$$

We can see the curves.fit coefficients are close to the theoretical ones, through the uncertainty we can see the average and the standard deviation are in the interval of uncertainty around the estimated coefficients except for the amplitude. By looking at the computed coefficients covariance matrix, one can see that coefficients seem uncorrelated.

To compare with the fit we can build the theoretical curve, that means we have to plot the Gaussian with the coefficients computed for our data:

```
X=np.linspace(1.925,2.525,100)
th_coef=np.max(Y_DATA),np.mean(DATA),np.std(DATA) # computation of theoretical coefficient
```

X is a list with a regular step to evaluate the theoretical curve and plot it.

Now we can plot all the curves:

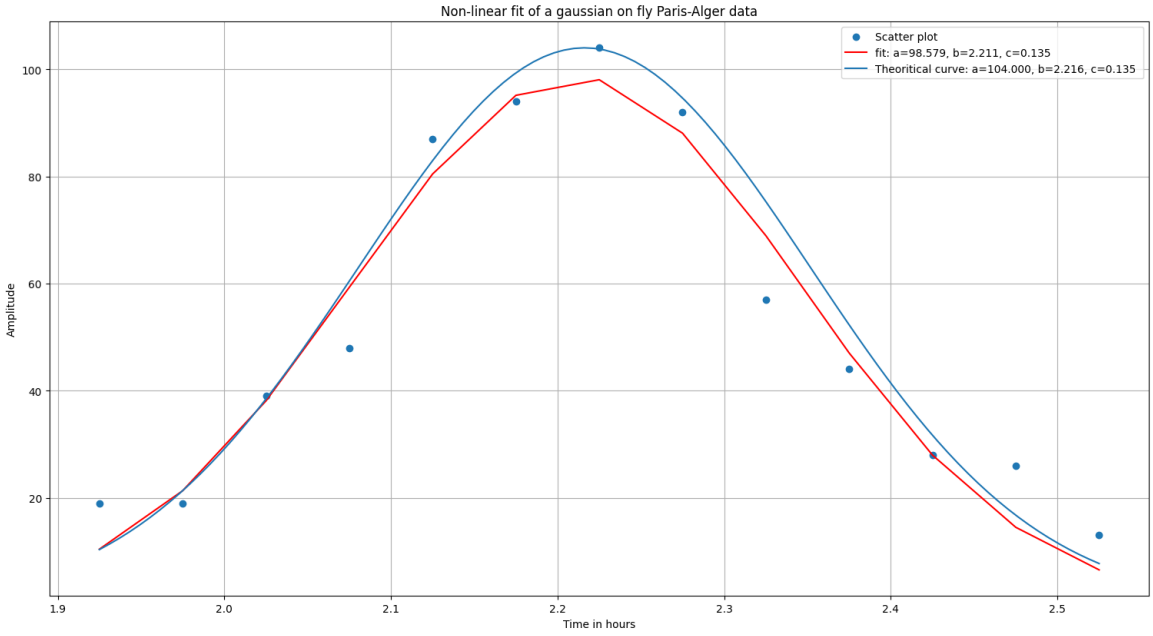

Figure 6: Fit of our data

We can see the data that we plot before and two curves on it, the theoretical curve seems to prove our data follows a Gaussian law.
Instead of the red curve that is our fit by using the function curves.fit, we can see the amplitude is too low like we have seen by the computation of our coefficients, but the fit seems correct.

5 Conclusion

In this practical work we have tested if our data followed the Normal distribution with a direct and an inverse problem approach which both lead to the conclusion that our data didn't follow the Normal distribution. Still, the χ^2 test seems to build some more accurate results towards the estimation.

We fitted our data with a Gaussian. In our case the fit seems to be correlated with our data but in the lectures an argument is to study the matrix of covariance.

The matrix of covariance is telling the degree of correlation between our data and the fitted curve, if the other parameters are very small than the diagonal ones, it means our data is not correlated with our Gaussian law.