1 Exercise 1

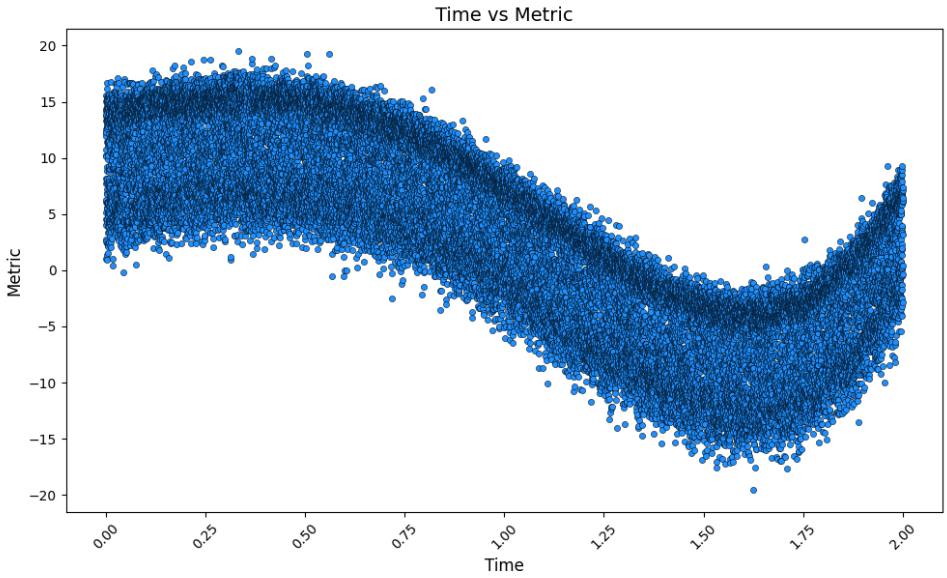
In this exercise, we were given a dataset that contained measurements of some quantity over time.

The data were stored in a CSV file where we have in every line two values: the first is the measurement time (that spans from 0 to 2), and the second is the measurement metric value (with values spanning from -20 to 20).
In total, we have 27199 entries.

The task was to model some statistics of the dataset and identify a trend in the data.

1.1 Part 1: identify a data trend

To begin with, we simply draw a scatter plot of the data (*time* against *metric*). This visualization helps to identify any apparent trends or patterns in the data.



The scatter plot reveals a noticeable trend in the data.

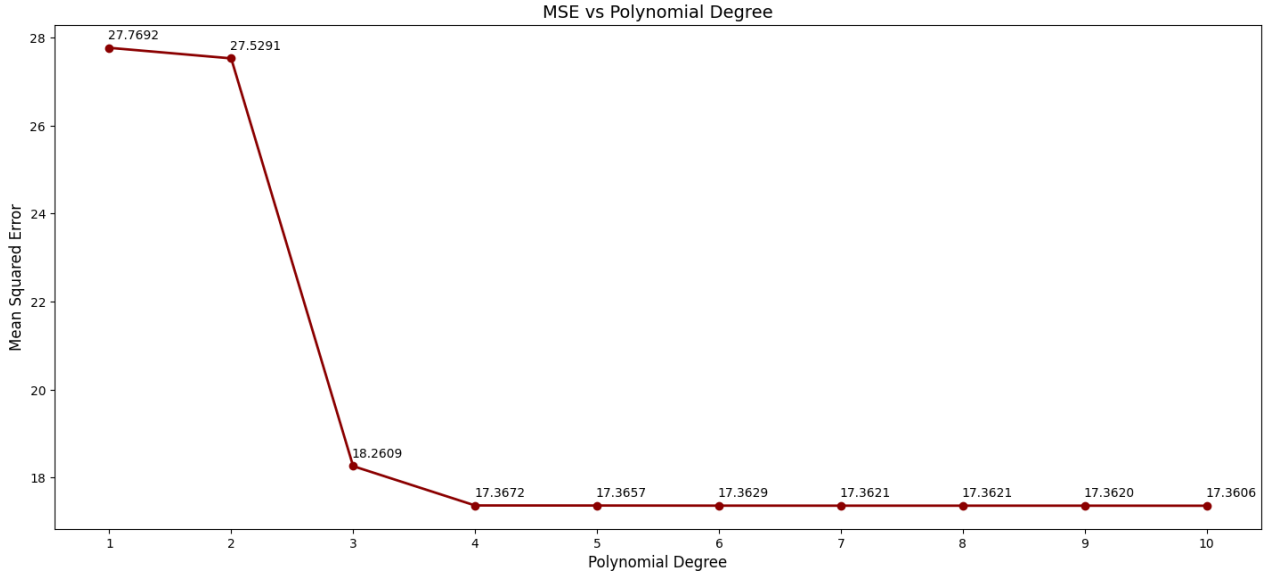
The metric value seems to follow a non-linear curve over time. This makes us think about the presence of a systematic pattern rather than random variation of the data.

1.2 Part 2: estimate the coefficients of polynomial trend function and remove the trend

After having confirmed the presence of a trend in our data, we used the **least squares** method to model it as polynomial.

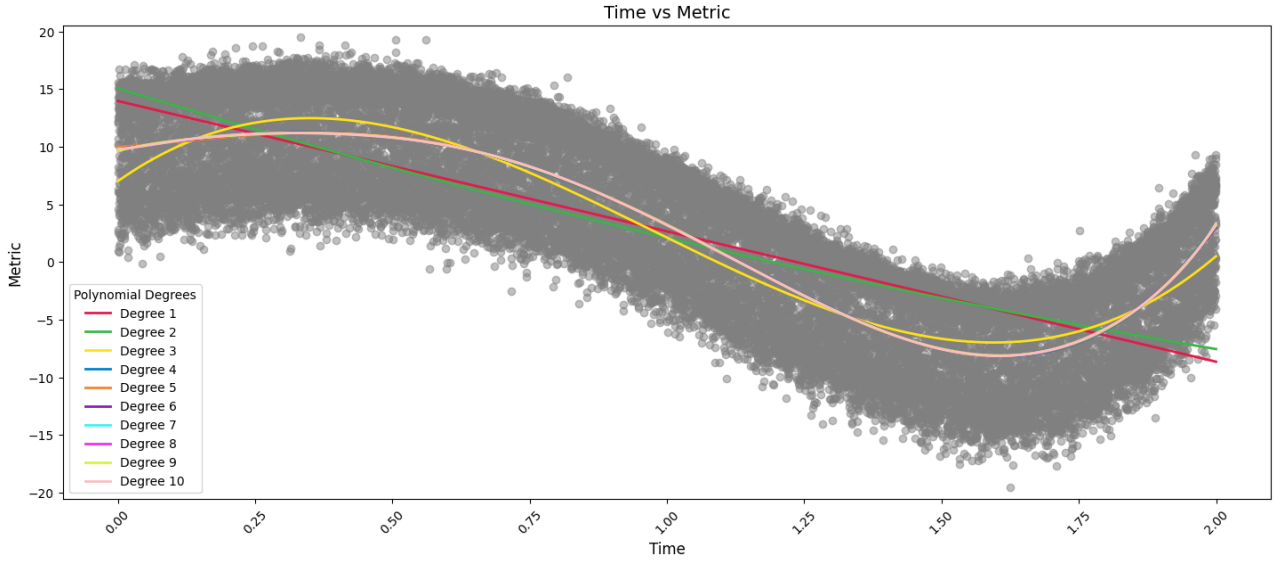
We tried to fit the trend with polynomials of degree up to 10 (by implementing from scratch the least squares method) and computed the following mean squared errors (shown in the plot on the next page).

As we can see in the graph, after degree 5 there is an elbow of the mean squared error. This means that by increasing the degree of our polynomial, we no longer get so many advantages in terms of modeling, but we are just overfitting our data.



We have also plotted the polynomials of various degrees on our trend, noticing that beyond degree 5, the polynomial curves became indistinguishable from one another within the relevant portion of the plot!

This strongly reinforces our conclusion that a degree-5 polnyomial is enough to accurately model the trend.



1.3 Part 3: trend removal and comparison

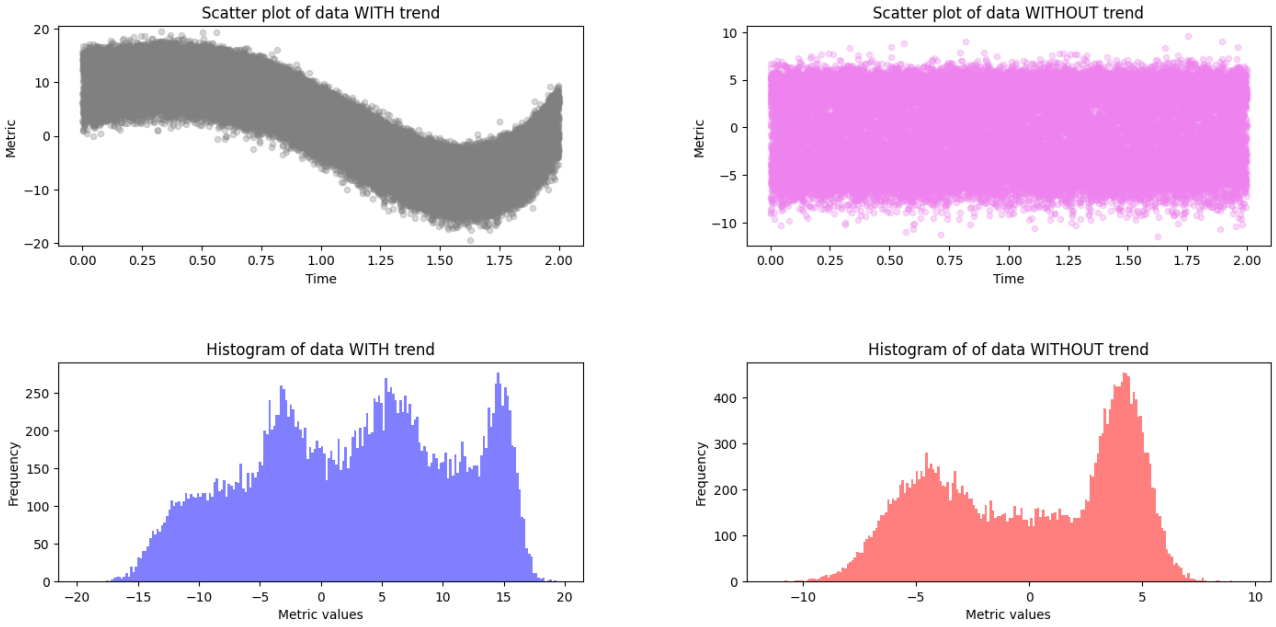
After having obtained the coefficients of the 5-degree polynomial function that allows us to model the trend, we simply remove it.

How?
Consider $P(x)$ our 5-degree polynomial with respect to the independent variable (the *time*).

We evaluated the trend by computing the estimated trend values at each point x, obtaining a vector $Y_{\text{trend}} = P(X)$.

Finally, subtracting the fitted trend from the original data, we obtained the detrended dataset, with the trend removed:

$$Y_{\text{residuals}} = Y - Y_{\text{trend}}$$



The results are quite self-explicative, and the scatter plot of the detrended dataset does not show a systematic pattern in the data, but just a random variation.

1.4 Part 4: fitting the dataset with a mixture of Gaussian

The next step of our work, was to try to fit a mixture of three Gaussian distributions to the detrended dataset. In order to do that, we have implemented the **Expectation-Maximization** algorithm.

We have implemented a general algorithm that can fit an arbitrary number of Gaussians on the data, passed as parameter of the function. It can be checked in file *exercise4.py*.

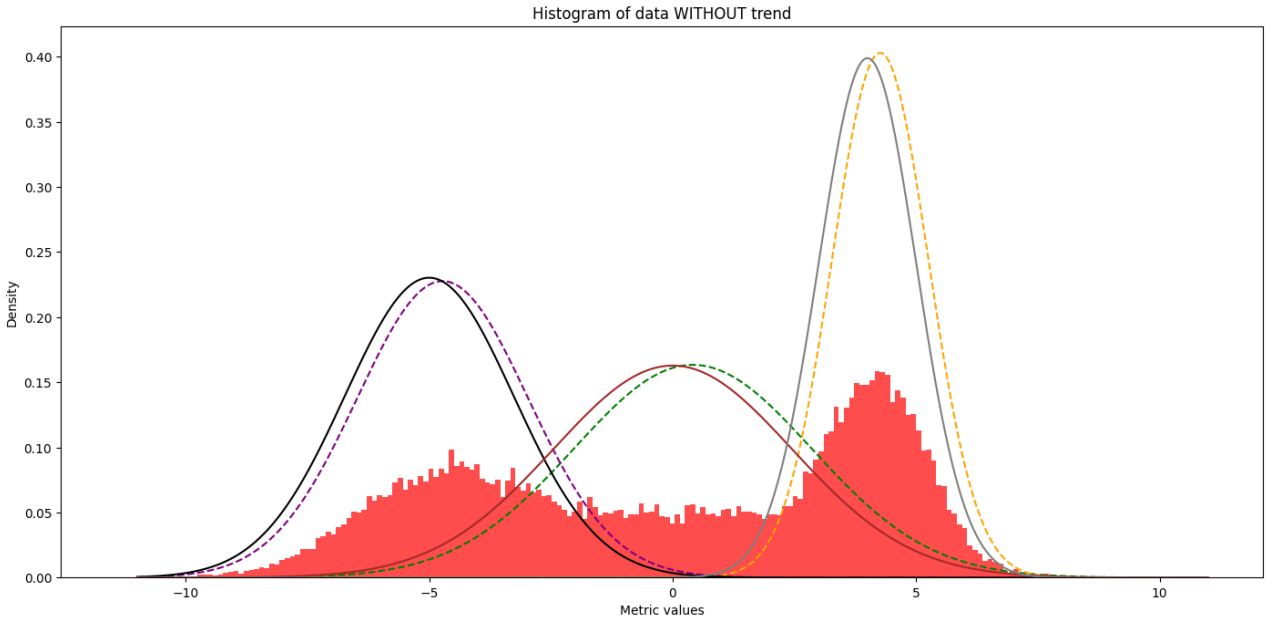
Our algorithm initializes the parameters to estimate by initially setting the standard deviation to match the standard deviation of the dataset and the mean as the dataset's mean shifted by a random value, different for each parameter.
This is not mandatory but allows to fasten the convergence of the algorithm.

In our simulation setting, we have fit a mixture of 3 Gaussians and run the algorithm for 150 epochs. That's the results we got:

	empirical (μ, σ^2)	real (μ, σ^2)
Gaussian distribution # 1	$(-4.725, 3.069)$	$(-5, 3)$
Gaussian distribution # 2	$(0.413, 5.960)$	$(0, 6)$
Gaussian distribution # 3	$(4.263, 0.980)$	$(4, 1)$

There are no significant discrepancies between empirical estimations and theoretical parameters, as shown also in the plot (*dashed* lines are the empirical Gaussians). However, correctly

"predict" the original Gaussian distribution that generated a point can be challenging, especially near Gaussian intersections (and that is the reason of the few discrepancies we have). Misclassifications in these cases may be useful, as such points could be interpreted as "outliers" better represented by a different Gaussian, the one actually estimated.



1.5 Facultative part: determining the optimal number of Gaussians

To automatically establish the best number of Gaussian distributions to fit the de-trended dataset, we came up with the following idea:

1. We considered $k = 1, 2, 3, 4, 5$ Gaussian components.
2. Then for each k we run the Expectation-Maximization algorithm and we compute two different metrics: **AIC** and **BIC**. AIC and BIC are interesting statistical metrics since they do not only take into account the likelihood of the data within the estimated distributions but also consider the number of parameters of the models (penalizing the ones more complex in order to avoid overfitting).
3. We plot those metrics to verify which number k minimizes them, considering also overfitting on the data.

By plotting those metrics, as we can see in the next page, we see that after 3 Gaussians we have a "knee" in AIC/BIC values: this means that use more Gaussians to fit the data does not allow us to have a great improvement in terms of dataset fitting.

These analytical results can be confirmed by another more intuitive method that simply consists of plotting the empirical PDF of the estimated Gaussians and visually observe that starting from 4 distributions we are overfitting our data.

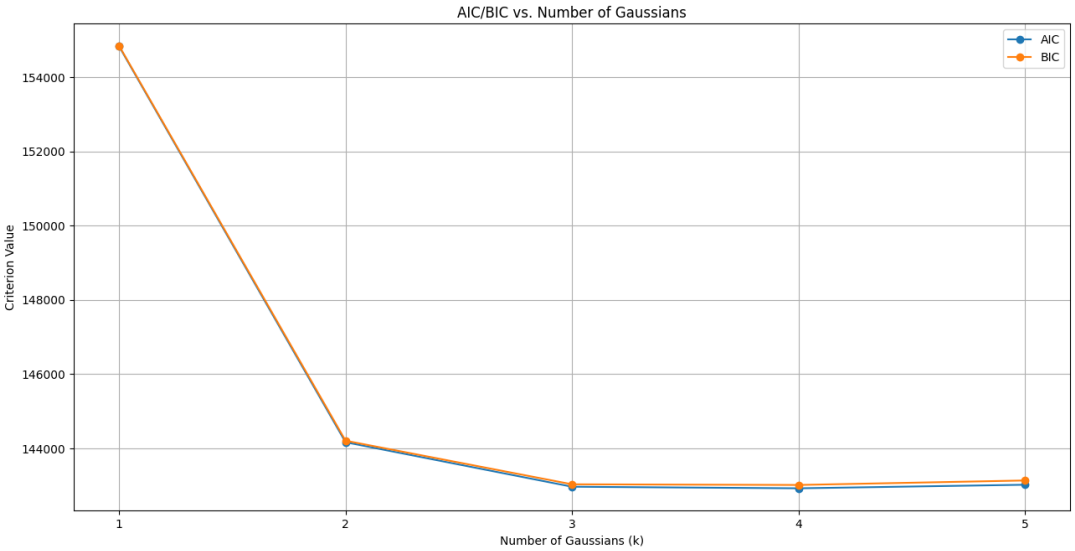


Figure 1: Comparison of the metrics with different components

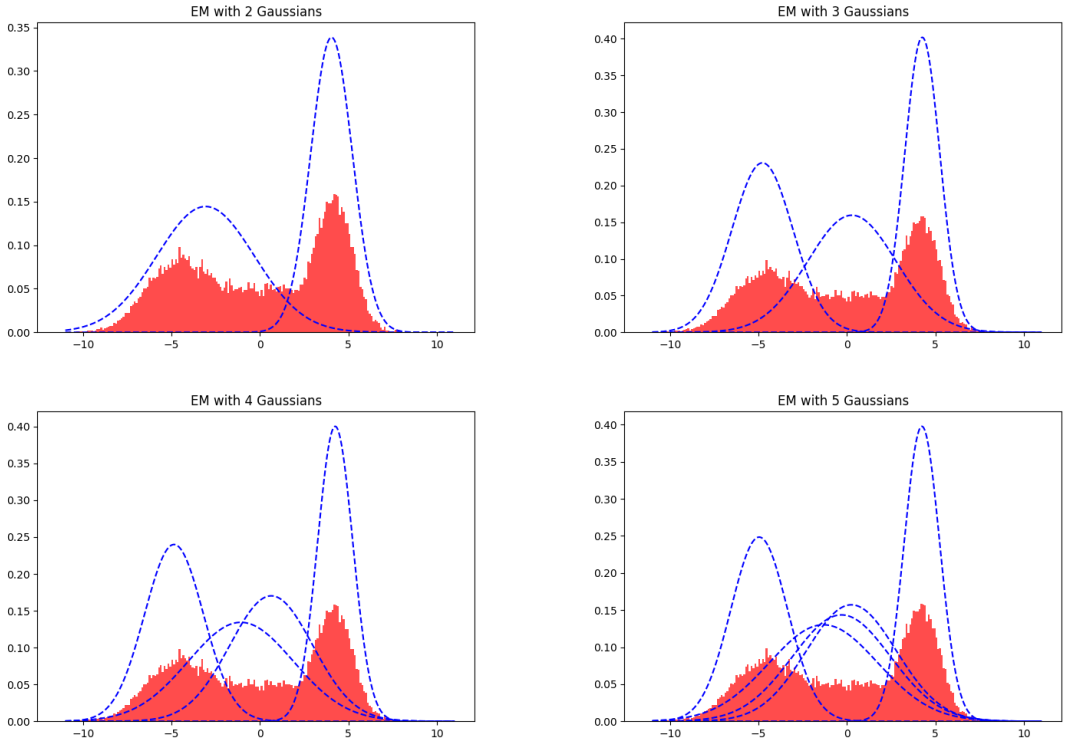


Figure 2: Empirical PDFs of 2, 3, 4 and 5 Gaussians