Documentation for Dataset1

Construction of the M Matrix for Least Squares

To initiate the linear regression process, I constructed the M matrix. This involved stacking the x-values from the dataset alongside a column of ones. This resulted in a matrix, denoted as M , facilitating the least squares regression to estimate the coefficients of the linear equation.

For stacking I looked up for a function: “np.column_stack” on google

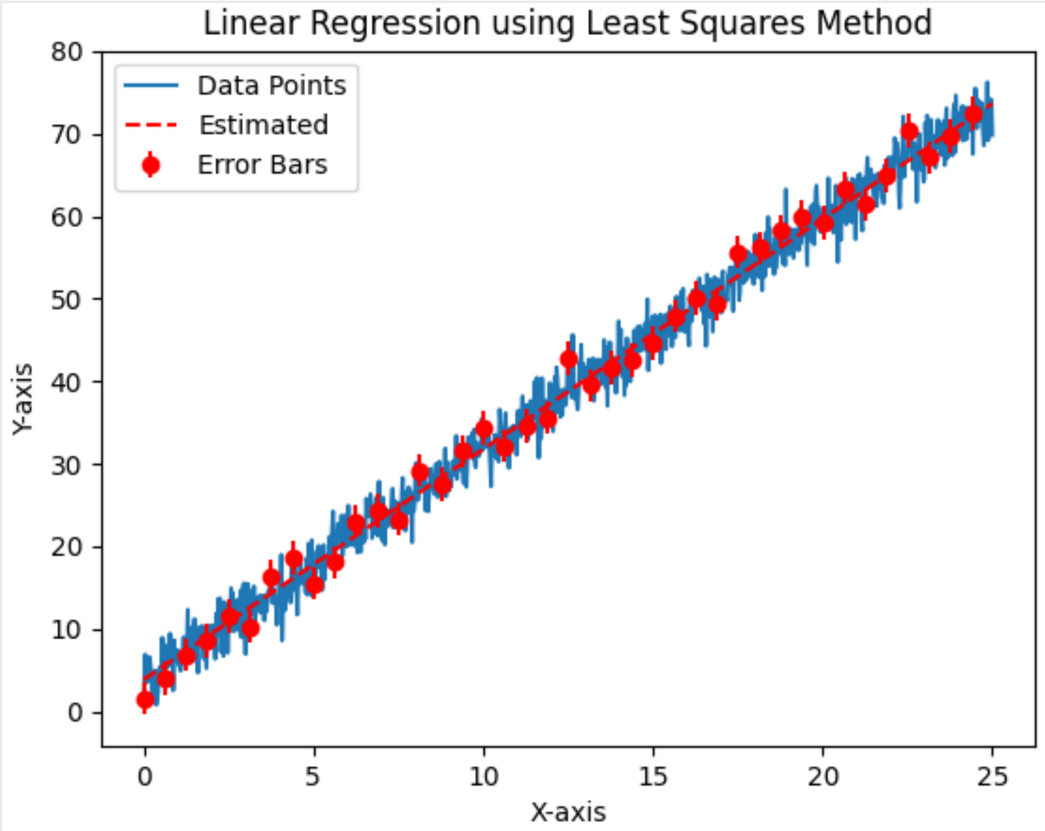
$$M = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}$$

Plot of Original Noisy Data with Error Bars

I visualised the original noisy data through a scatter plot. To account for the uncertainties in the data, I added error bars. Notably, to enhance clarity, I chose to plot error bars for every 25 data points.

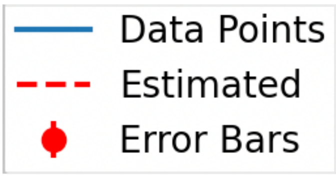
Overlaying the Estimated Line on the Plot

To evaluate how well the estimated line fits the data, I overlaid it on the same plot as the original data. The estimated line was derived through the least squares regression, and its slope and intercept were determined based on the characteristics of the dataset.



Adding a Legend to the Plot

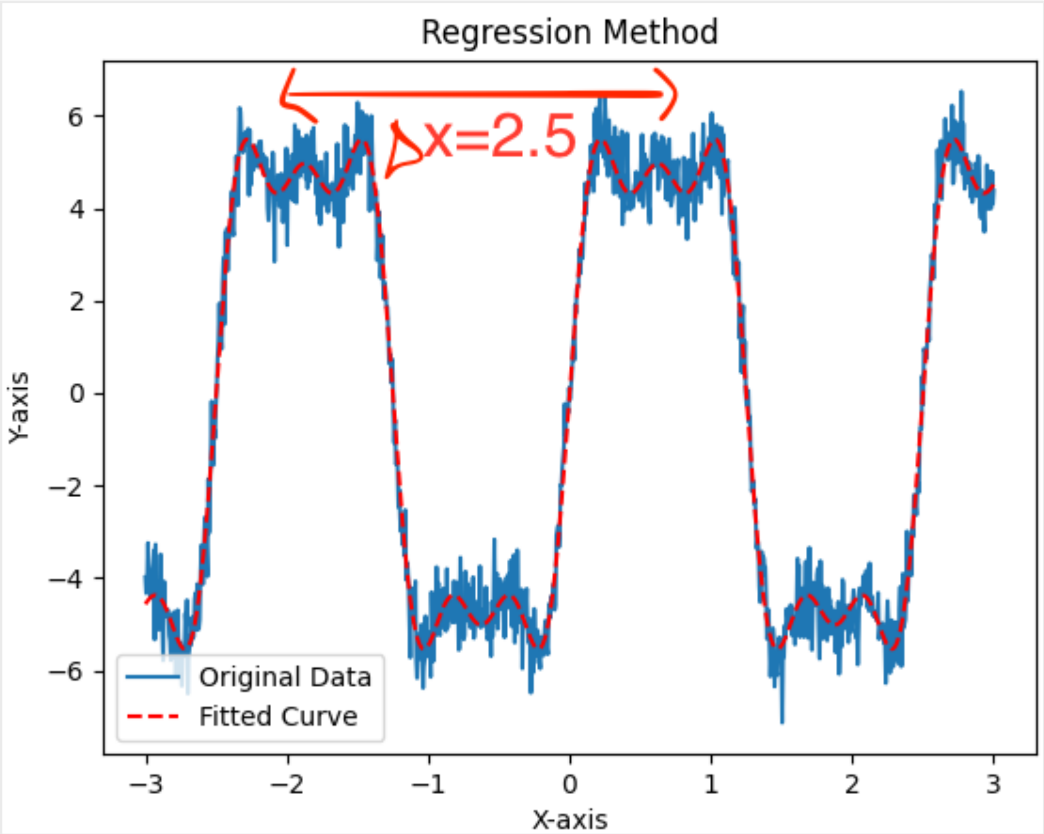
For better interpretation, I added a legend to the plot. The legend included entries for both the original data points and the estimated line. This addition aimed to make it easier to distinguish between the two and provide a clear visual reference for understanding the relationship between the data and the fitted model.



Documentation for Dataset2

Estimation of Sine Wave Periodicity

For estimating the periodicity of the sine waves, I relied on a visual inspection approach. By identifying prominent peaks or troughs in the sinusoidal patterns and measuring the distance between consecutive peaks or troughs, I intuitively approximated the periodicity as 2.5.



Construction of the M Matrix for Least Squares

In the context of least squares regression, the M matrix played a pivotal role in capturing the sinusoidal patterns within the dataset. The M matrix was constructed by forming columns corresponding to different sine waves with distinct frequencies.

Mathematically, the equation setup for least squares was:

$$f(t) = k_1 \sin(2\pi t/T) + k_2 \sin(3 \cdot 2\pi t/T) + k_3 \sin(5 \cdot 2\pi t/T) + k_4$$

And the Matrix looks like:

$$M = \begin{bmatrix} \sin(2\pi t_1/T) & \sin(3 \cdot 2\pi t_1/T) & \sin(5 \cdot 2\pi t_1/T) & 1 \\ \sin(2\pi t_2/T) & \sin(3 \cdot 2\pi t_2/T) & \sin(5 \cdot 2\pi t_2/T) & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \sin(2\pi t_n/T) & \sin(3 \cdot 2\pi t_n/T) & \sin(5 \cdot 2\pi t_n/T) & 1 \end{bmatrix}$$

Comparative Analysis

After obtaining parameter estimates from both the least squares regression and the curve_fit method, a comparative analysis was conducted. This comparison aimed to assess differences or similarities between the results, offering insights into the accuracy and reliability of the employed methods.

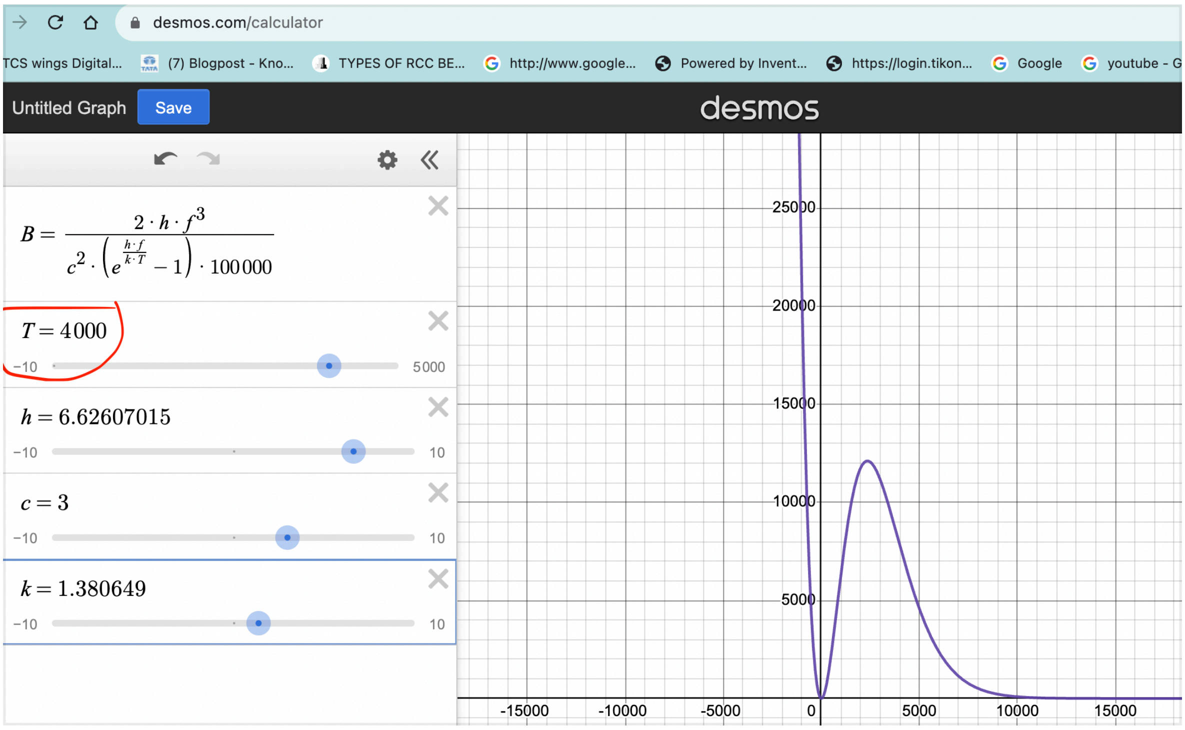
```
Amplitudes of sin(t), sin(3t), and sin(5t) from regression method: 6.011152934913333, 2.001543820042487, 0.9802390885802197, -0.025875188673358293 with time period 2.5
Amplitudes of sin(t), sin(3t), and sin(5t) from curve_fit method: 6.011120069207352, 2.0014582413321165, 0.9809078856869915, -0.025875183549991392 with time period 2.500536
```

Documentation for Dataset3

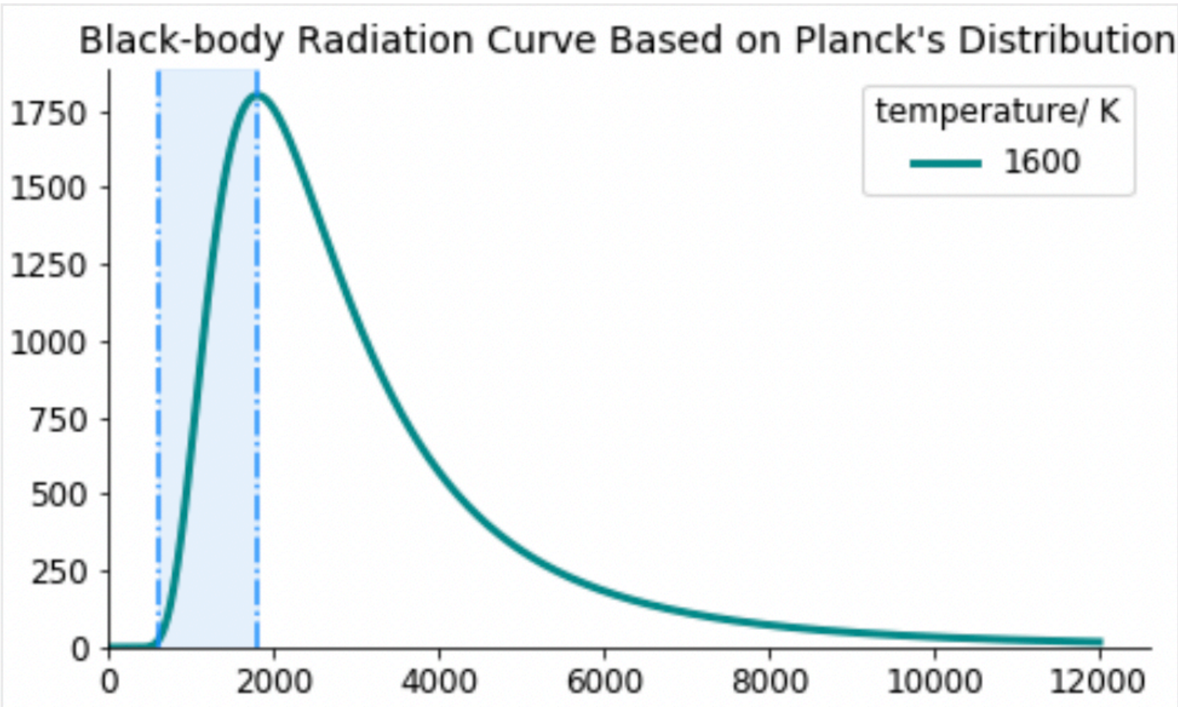
Trail and Error Approach I've used for estimating initial parameter of Temperature: (p0=4000)

I've done this by plotting the graph of the equation on desmos (https://www.desmos.com/calculator) :

$$B(f,T) = \frac{2hf^3}{c^2} \cdot \frac{1}{\exp\left(\frac{hf}{kT}\right) - 1}$$



By sliding the values of T on desmos I made sure that the graph fits the graph of black-body radiation that I found on Google:



Source: https://chemistry.stackexchange.com/questions/138806/why-is-black-body-radiation-curve-smooth-without-a-sharp-cutoff

Q) Comment on the quality of the results you obtain. How close are they to the actual values, and can you think of ways to improve your estimates.

Answer) To improve the quality of the estimates we should revise the values of the initial parameters we are fitting inside curve_fit function every time we run the code.

We start off with the actual values:

```
, p0=[6.63e-34, 3e8, 1.38e-23, 4000])
```

From the output, we take these and replace them into the values of p0 each time we get new output. We do these until we get closer to the actual values of the constants.

```
(base) santhoshgandham@Santhoshs-Air a3-data % python3 dataset3_2.py
Optimal value of h: 5.2101597168013285e-33
Optimal value of c: 842197566.5024782
Optimal value of k: 4.318231322715927e-23
Optimal value of T: 12575.243573171785
```

Bookmarks Version Control TODO Problems Terminal Python Packages Python

PEP 8: E305 expected 2 blank lines after class or function definition, found 1