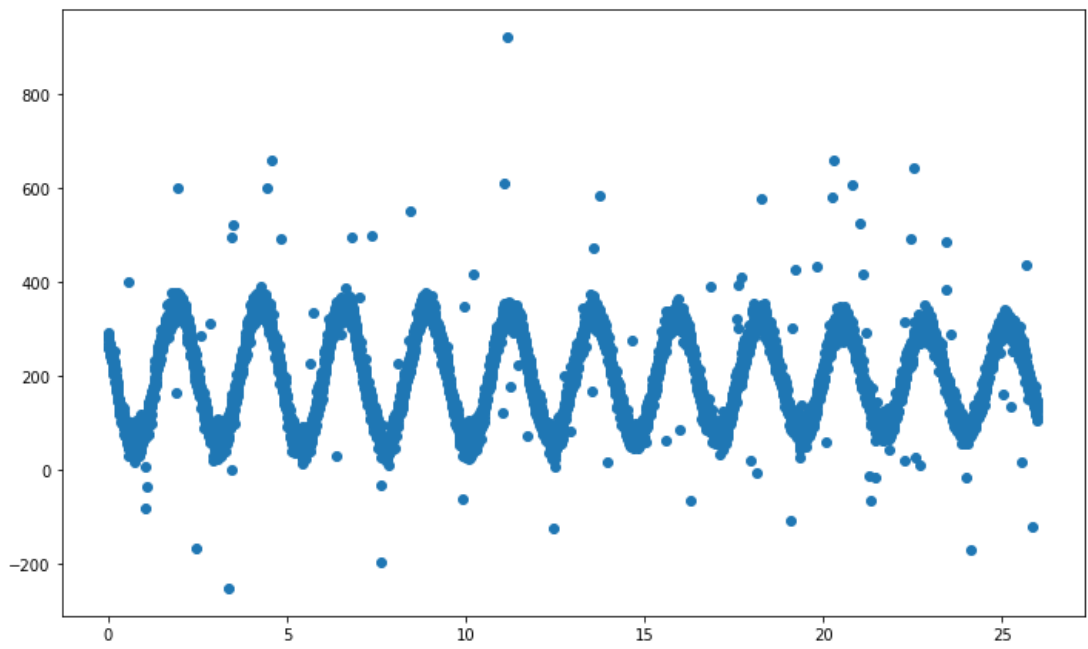# HA 1

prepared by Danila Danko

Please, refer to [Project setup](#) to run the code.

## Task 1

In Task 1, I took the dataset for my case (case 1) and visualized it.



I noticed that the points resemble a $sin$ wave. That's why, I decided not to use Linear Regression.
Instead, I learned about Gauss-Newton's method here ([link](#)) and implemented it for my problem.

I switched to the following formulation of my problem:

Gauss-Newton algorithm directly deals with this type of problems. Given m data points $(x_i, y_i)$ for regression with a function of n parameters $\vec{\beta} = (\beta_1, \ldots, \beta_n)$

$$min_{\vec{\beta}}\ S(\vec{\beta})\ where\ S(\vec{\beta}) = \sum_{i=1}^{m} r_i(\vec{\beta})^2 = (y_i - f(\vec{\beta}, x_i))^2$$

And implemented the suggested algorithm. I tried it with different initial values and [damping factor](#), and it was unable to find the correct amplitude. Unsurprisingly, because the algorithm [may not converge](#). Here is the final result.
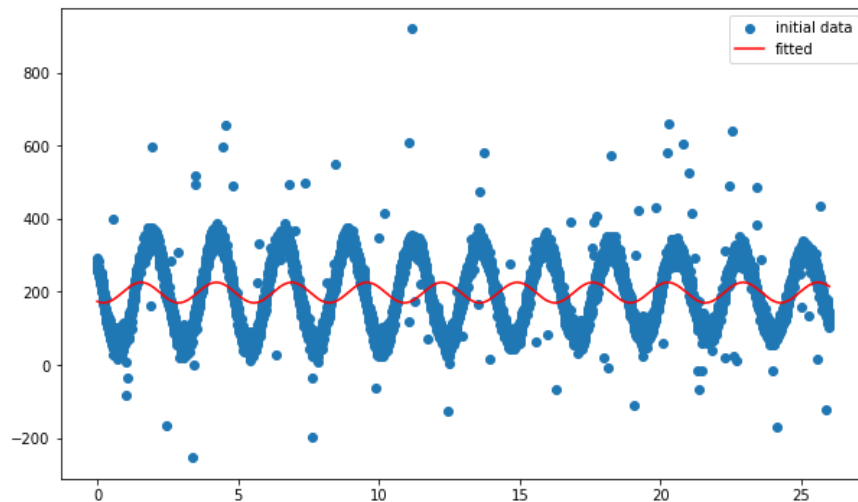
## Task 2

In task 2, I used **dataset 8** and `sklearn.linear_model.RANSACRegressor` to fit the data provided there. To make an experiment, I nade some points the outliers (blue). The inliers are red.
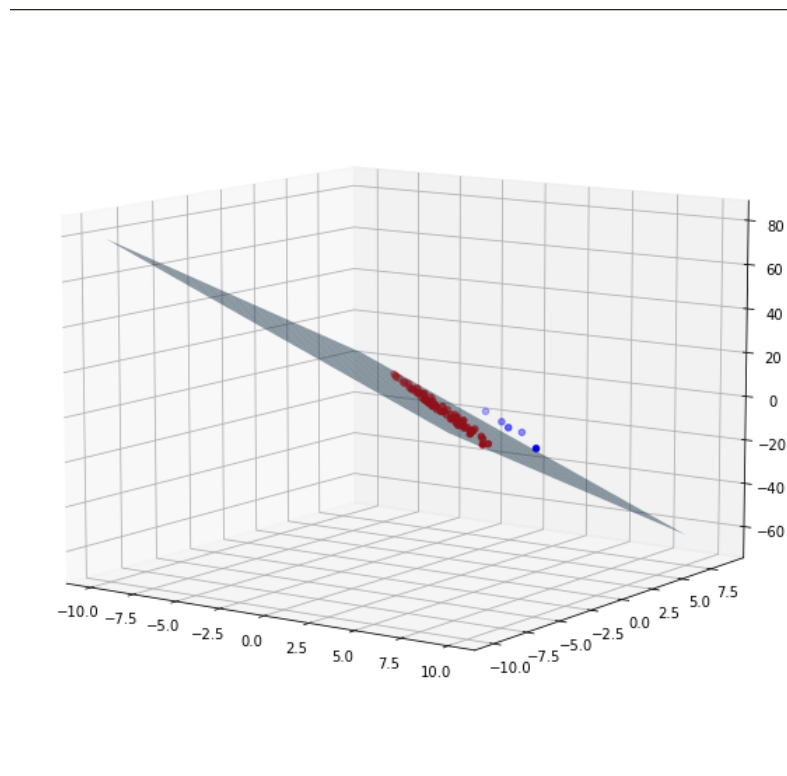


For `residual_threshold`, I applied [median absolute deviation](median absolute deviation) since it's a "robust estimator of dispersion".

I set `min_samples` = 3, since it's the minimal number of points required to construct a plane from a non-degenerate triangle. That the data represents a plane is obvious from the scatter plots.

Finally, I observed that 100 iterations are enough to exclude all my outliers.