Homework 1 – REPORT

Submitted in partial fulfillment of the requirements for the course of

ENPM673 – PERCEPTION FOR AUTONOMOUS SYSTEMS

By

Adheesh Chatterjee

UID – 116236935

M.Eng Robotics

Course Faculty
Prof Cornelia Fermuller

A. James Clark
SCHOOL OF ENGINEERING

Co-variance Matrix – Geometric Interpretation

The co-variance matrix is composed of the co-variance and variance. In the data provided, we have only two dimensions – x and y. So we calculate the variance of x and y separately and then the covariance of x and y to determine the whole matrix. The diagonal entries of the covariance matrix are the individual variances and the other entries are the covariance. The calculation for the covariance matrix can be also expressed as –

$$C = \begin{pmatrix} \sigma(x,x) & \sigma(x,y) \\ \sigma(y,x) & \sigma(y,y) \end{pmatrix}$$

$$\sigma_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \qquad \sigma(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Here σ_x^2 is the variance for x or $\sigma(x,x)$ as represented in C . Similarily the variance for y can be found as $\sigma(y,y)$. The covariances are represented as $\sigma(x,y)$ and $\sigma(y,x)$

The co-variance matrix is used to fully analyze and characterize the two-dimensional variation of the data. This matrix determines the stretch of the data in the x and y direction, while also taking into account the diagonal stretch

Eigen Values and Eigen Vectors

An eigenvector is a vector whose direction remains unchanged when a linear transformation is applied to it. It can be expressed as

$$Av = \lambda v$$

where v is an eigenvector of A and λ is the corresponding eigenvalue. If we put all eigenvectors into the columns of a Matrix V and all eigenvalues as the entries of a diagonal matrix L we can write for our covariance matrix C the following equation

$$CV = VL$$

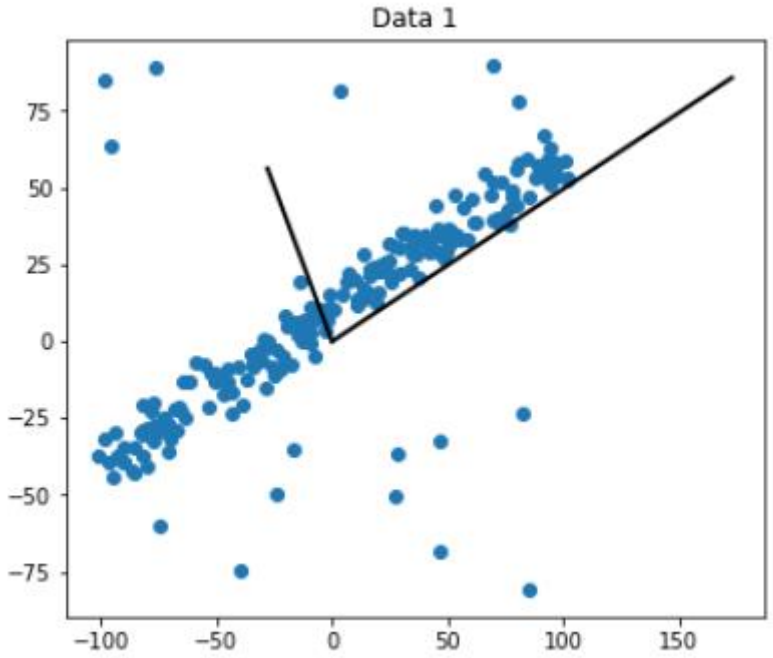
From the co-variance matrix we can determine its respective eigenvalues and eigenvectors. The eigenvalues represent the co-variances of the data, while the eigenvectors represent linearly independent directions of variations in the data. It is worth mentioning that the direction of an eigenvector does not change. Additionally, if the co-variance is equal to zero, then the values of the eigenvalues and eigenvectors are equal.

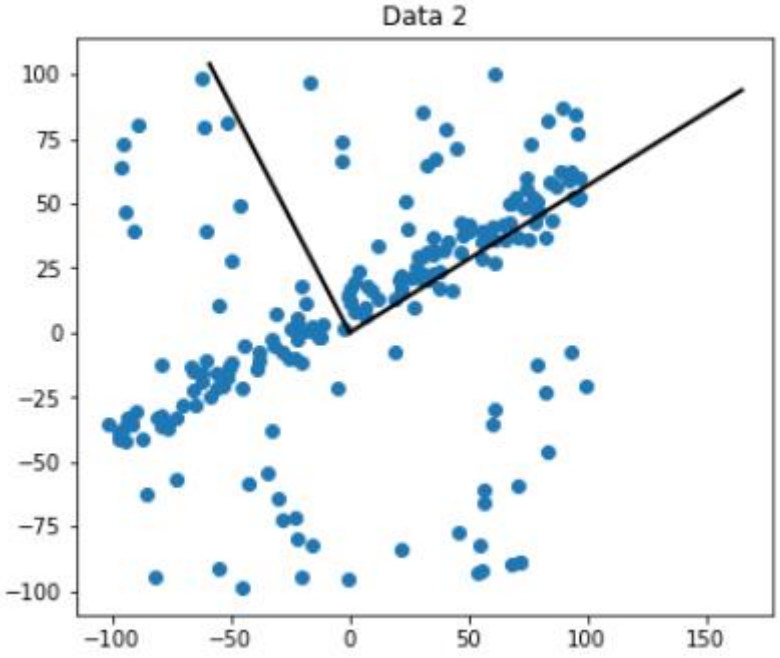
$$T = RS$$

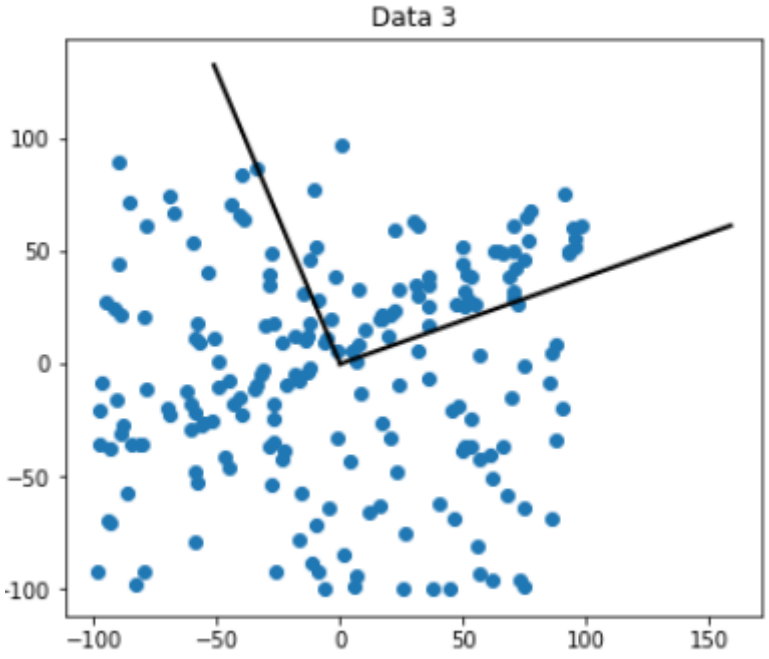
The geometric interpretation is achieved by using both the eigenvalues and eigenvectors to display direction and the highest spread of the data. The T variable is

represented by the scaling factor S, and rotation matrix R. The scaling factor S is determine by placing the eigenvalues diagonally in order of its eigenvector, and taking the square root of the matrix.

Utilizing the method described above we are able to solve and show the geometric representation of the 3 data-sets that were provided.






Data 3
100
50
0
−50
-100
−100
−50
0
50
100
150

Least Squares – Vertical and Orthogonal

There are several methods that allow for line fitting, such as vertical least squares and total least squares. Line fitting is the process of fitting a series of data points by a straight line.

Least Squares Vertical

Vertical Least Squares can be defined as the sum of the vertical distances of the dependent variable between each data point and the corresponding regression line. As a result, the smaller the sum of the distances the better the model fits the data given.

$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

We use the equation above to minimize the error or vertical distances to the regression line.

We then represent the equation by matrices as

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad X = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \qquad B = \begin{bmatrix} m \\ b \end{bmatrix}$$

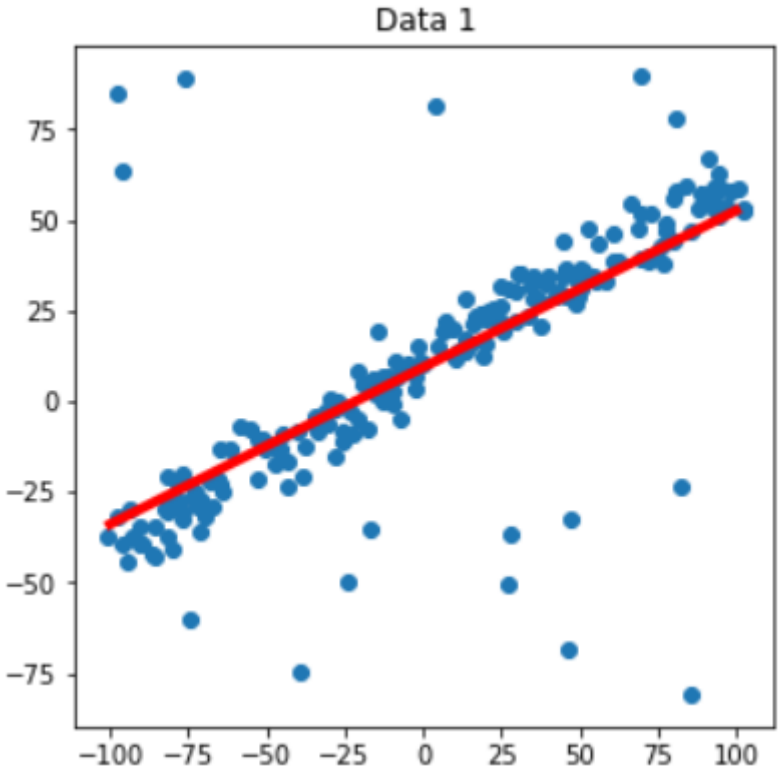
$$E = \|Y - XB\|^2 = (Y - XB)^T (Y - XB) = Y^T Y - 2(XB)^T Y + (XB)^T (XB)$$

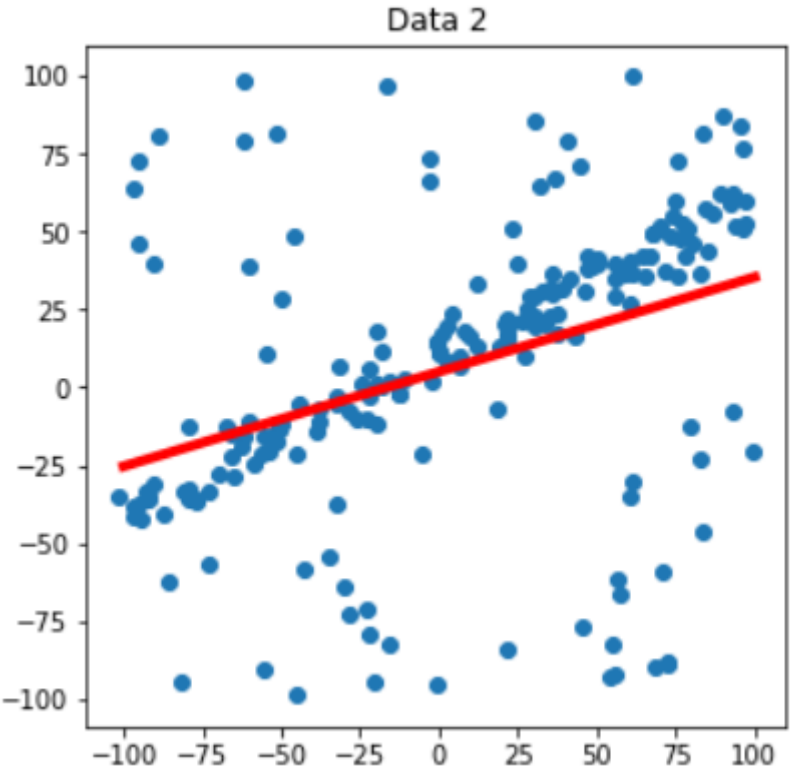
By applying the pseudo-inverse to the XB = Y equation, we are able to simplify and solve for the slope and the intercept of the fitted line.

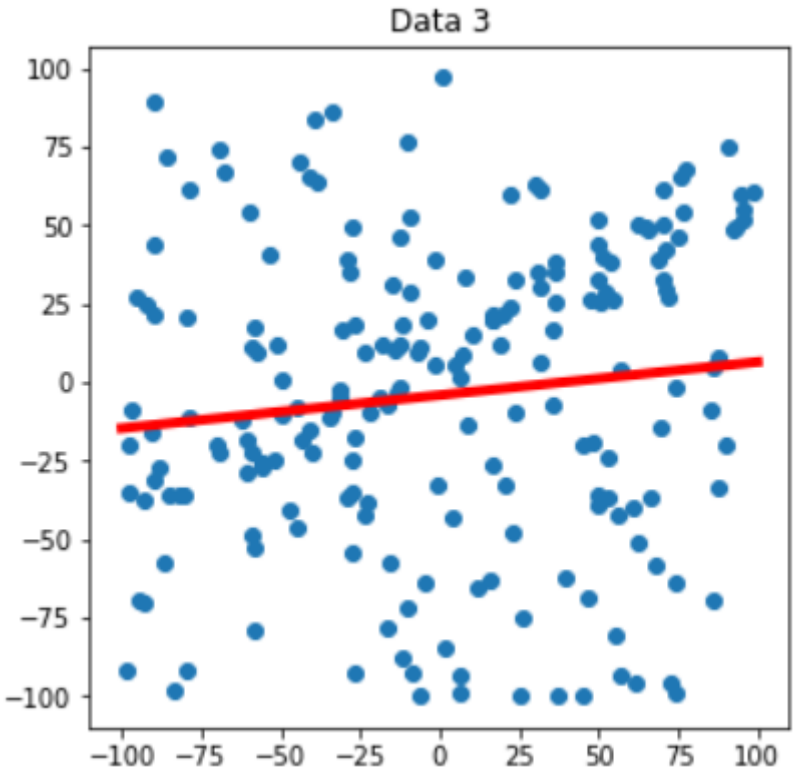
$$\frac{dE}{dB} = 2X^T XB - 2X^T Y = 0$$

$$X^T XB = X^T Y$$

$$B = (X^T X)^{-1} (X^T Y)$$


Data 1
75
50
25
0
−25
−50
−75
−100 −75 −50 −25 0 25 50 75 100


Data 2
100
75
50
25
0
−25
−50
−75
−100
−100 −75 −50 −25 0 25 50 75 100



By analyzing the results of the three data-sets it is clear that vertical least squares doesn't always best fit the data. This is mainly because it is very sensitive to outliers. There are a large number of outliers is data 2 and data 3 and hence they carry a lot of weight which throws off our model.

Least Squares – Orthogonal

Total least squares can be defined as the sum of the orthogonal distances of both x and y from the points of the fitted line. Additionally, this method takes into account the uncertainties of both x and y

We use the equation below to minimize the error between the point (x_i, y_i) to the line $ax + by = d$

$$E = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

$$E = \sum_{i=1}^{n} (a(x_i - \bar{x}) + b(y_i - y))^2$$

$$\left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = (UN)^T (UN)$$

$$U = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \quad U^T U = \begin{bmatrix} \sum_{i=1}^{n} (x_i - \bar{x})^2 & \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^{n} (y_i - \bar{y})^2 \end{bmatrix}$$

The solution of the total least squares regression is obtained by using Singular Value Decomposition.

We apply SVD to the matrix shown in the previous equation will solve for the eigenvalues and eigenvectors. Then, we find the slope and intercept of the linearized regression by considering the smallest eigenvector. This vector will give the best solution to the linear regression.

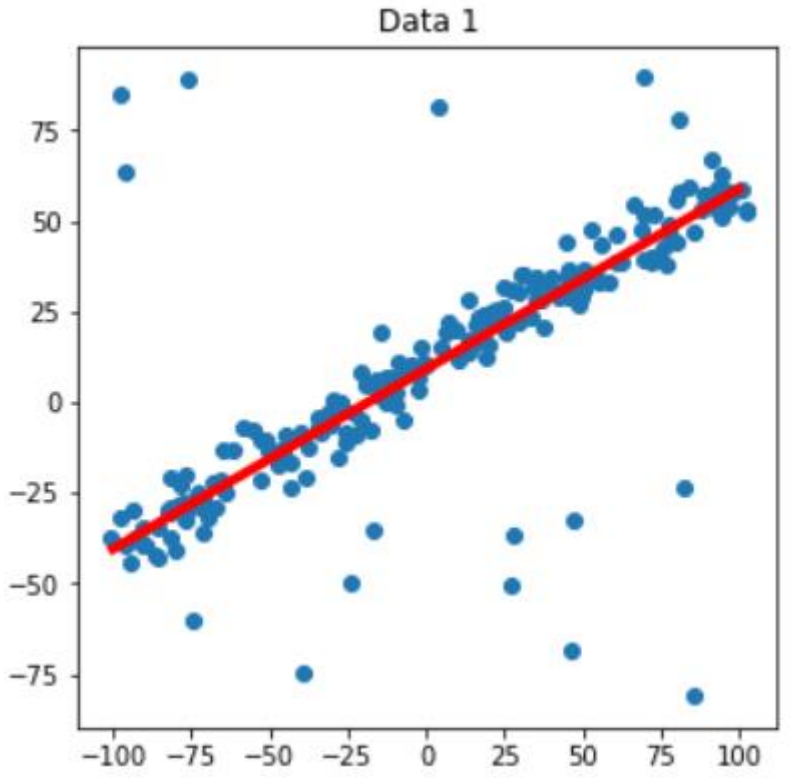
However, this whole process can be simplified by calculating the eigenvector directly with simplified equations. This was done due to the fact that hardcoding SVD on python is an arduous task. Thus the following equations were used for simplicity. They serve the same purpose as an SVD but are easier to code.

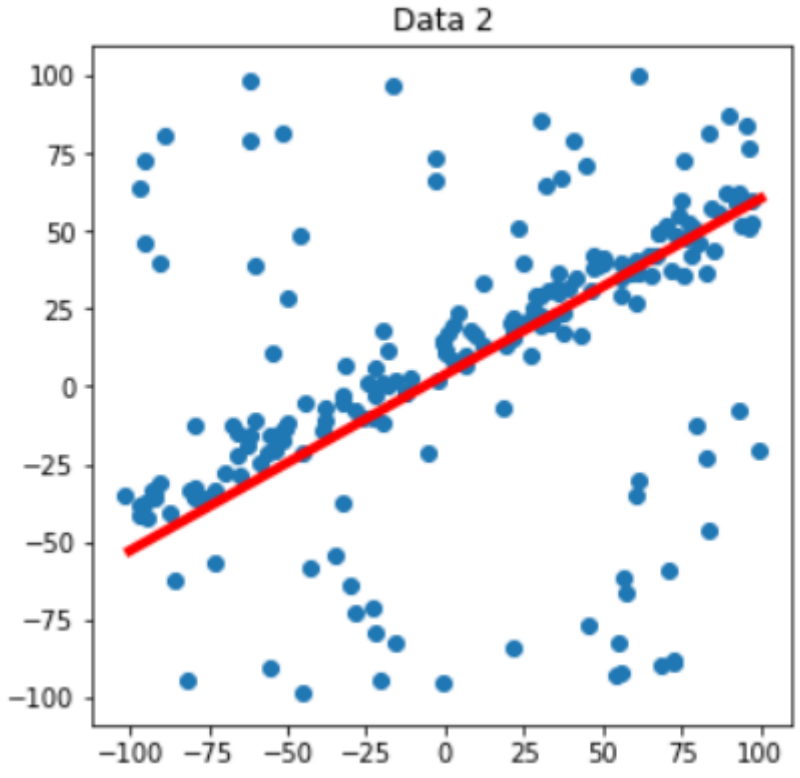
The equation shown below is the method used to determine the required eigenvector

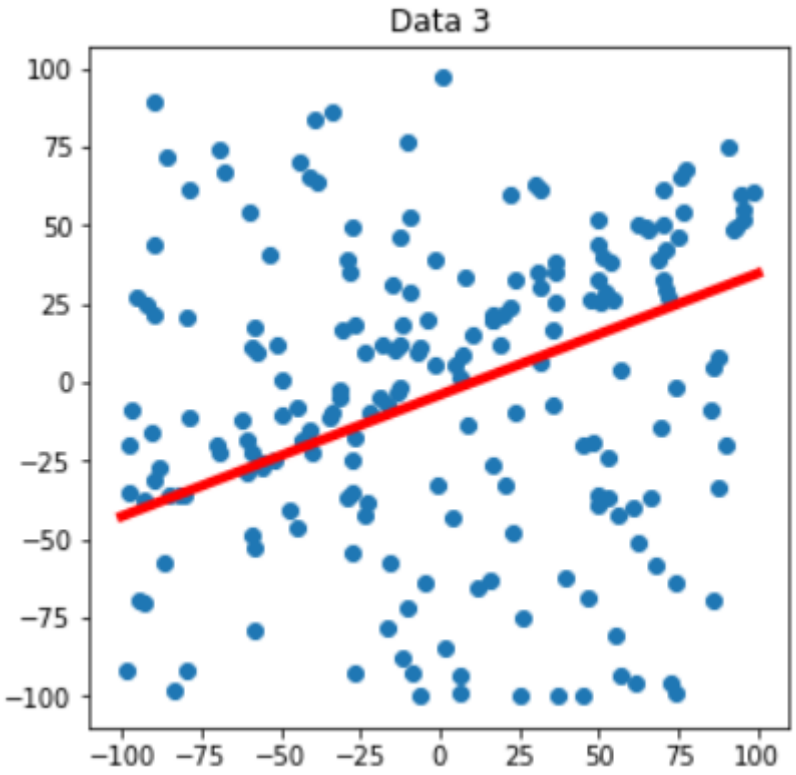
$$b = \frac{w + \sqrt{w^2 + r^2}}{r}$$

The w and r are derived from the U and $U^T U$ equation shown previously. These equations simplify the calculation of the eigenvector that gives the slope and intercept of the fitted line.

$$w = \sum_{i=1}^{n} (y_i - \bar{y})^2 - \sum_{i=1}^{n} (x_i - \bar{x})^2 \qquad r = 2 \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$


Data 1
75
50
25
0
−25
−50
−75
−100 −75 −50 −25 0 25 50 75 100


Data 2
100
75
50
25
0
−25
−50
−75
−100
−100 −75 −50 −25 0 25 50 75 100



By analyzing the results of the three data-sets it is clear that orthogonal least squares is a better fit than vertical least squares. This is mainly because orthogonal distances take into account the uncertainties of both variables unlike vertical least squares which is very sensitive to outliers. This happens mainly because the large residuals that are squared carry a lot of weight. Since all 3 data-sets contain outliers, these will skew the fitted line of the vertical least squares, and therefore Total or Orthogonal Least Squares best fits the line. This is especially visible in the third dataset as that contains the maximum outliers and yet, Total Least Squares fits the data well.

OUTLIER REJECTION

RANSAC

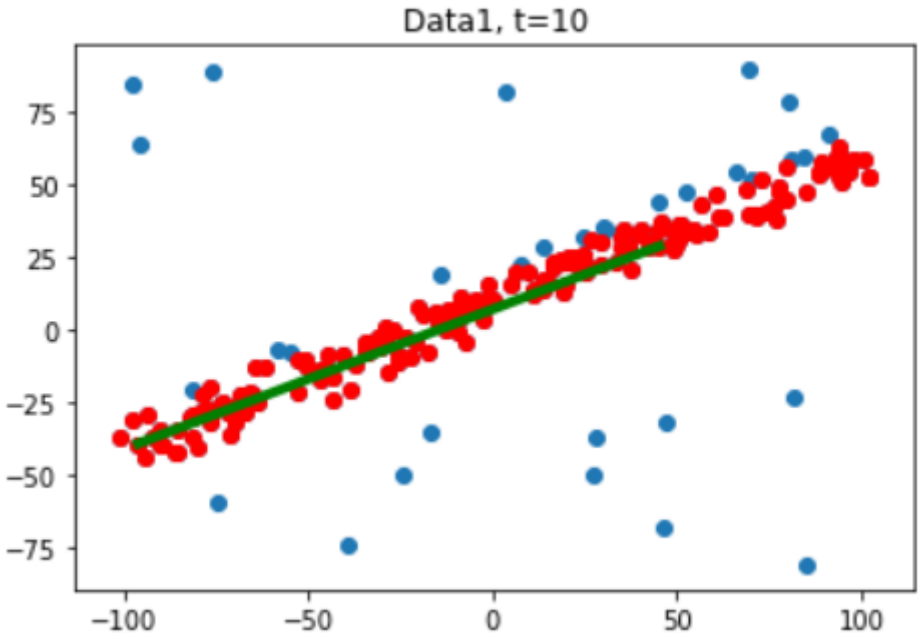
Linear regression models can be heavily impacted by the presence of outliers. As an alternative to throwing out outliers, we use a robust method of regression using the RANdom SAmple Consensus (RANSAC) algorithm, which is a regression model to a subset of the data, the inliers.

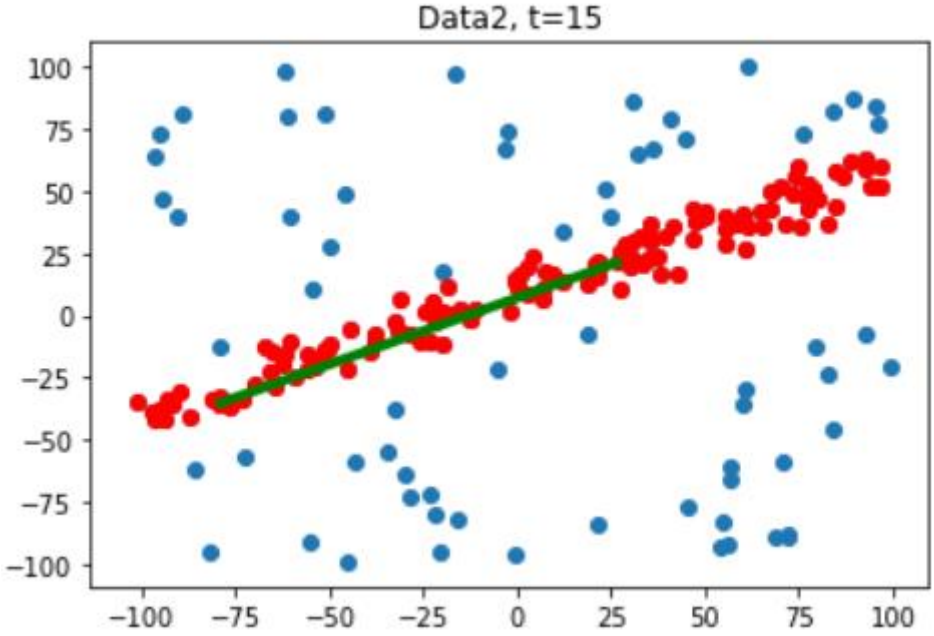
RANSAC iteratively estimates the parameters from the data set. At each iteration the following steps are performed:

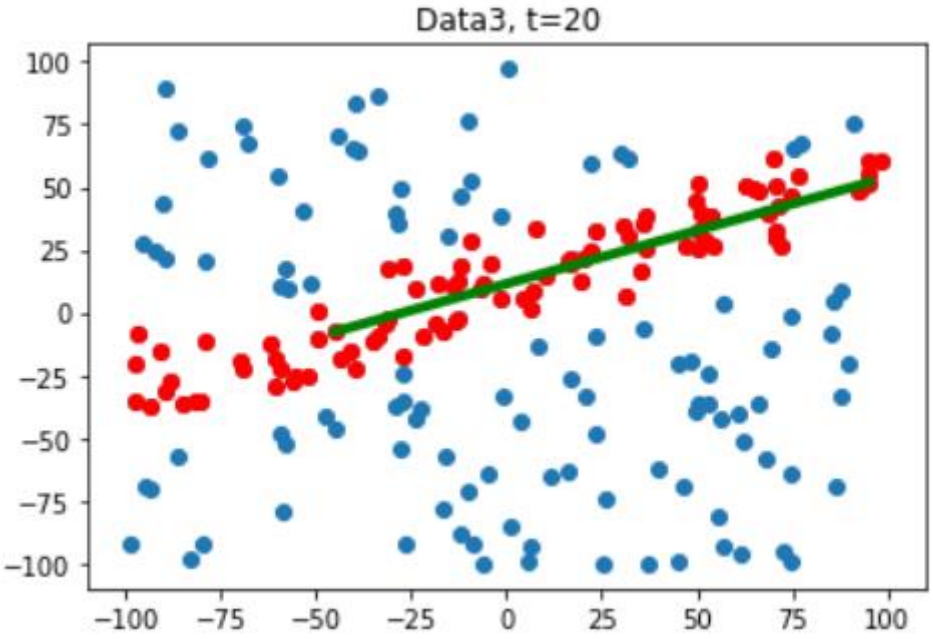
- 2 random points are selected
- The distance between a line drawn between these 2 randomly selected points and the remaining data points are calculated
- A threshold is chosen which is the maximum distance between the line and the inliers.
- The inliers are saved and the entire process is repeated N times to find the best 2 points which give the largest number of inliers.

$$N = \log(1-p) / \log\left(1-(1-e)^s\right)$$

Where e = 1 – (number of inliers)/(total number of points)
p is the probability of inliers for a threshold t
s is the minimum number of points needed to fit the model. Here we used 2







RANSAC is our preferred outlier rejection technique because it is simple and can be applied to many different problems. The parameters are easy to choose and even with the presence of a high level of outliers, it is able to efficiently only take into account the necessary inliers.

It has its drawbacks though, as there are a lot of parameters we have to tune and it doesn't work for low inlier ratio as we will have too many iterations and it can fail completely. Also, we can't always get a good initialization of the model based on the minimum number of samples.