Q3 Coming up with a linear relation between the variables X and Y is equivalent to reducing the number of independent variables to 2. PCA can be employed in this. We define the matrices below (n is the sample size):

$$\text{The sample matrix: } S_{n\times 2} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

It holds the values from points2D_Set1.mat for now and from points2D_Set2.mat later.

$$\text{The standardized sample matrix: } \overline{S}_{n\times 2} = \begin{bmatrix} \overline{x_1} & \overline{y_1} \\ \vdots & \vdots \\ \overline{x_n} & y_n \end{bmatrix} = S_{n\times 2} - \begin{bmatrix} \mu_x & \mu_y \\ \vdots & \vdots \\ \mu_x & \mu_y \end{bmatrix}$$

where μ_x and μ_y are means of the first and second columns of S respectively.

$$\text{The covariance matrix: } C_{2\times 2}(x,y) = \begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix}$$

where $Cov(u,v)$ denotes the covariance of u and v. The function used for $Cov(u,v)$ in the program is getcovar(x, y). Its precondition is that $\mu_x = \mu_y = 0$. In that case, $Cov(u,v) = \frac{1}{n}\sum_{i=1}^{n} u_i v_i$.

We follow PCA and calculate unit eigenvectors E_1, E_2 and eigenvalues v_1, v_2 of the covariance matrix of $C(\overline{x}, \overline{y})$. We take the first principle component as E_1 with the higher eigenvalue v_1. We procure the projection matrix $P_{n\times 1}$:

$$P_{n\times 1} = S_{n\times 2} \times (E_1)_{2\times 1}$$

which projects the original sample's coordinates onto the span of E_1. The projection matrix holds only the magnitude of these projections. To give them direction, we multiply the projection matrix with E_1, and factor in the intercepts, we shift by the mean matrix $\mu_{N\times 2}$ which is obtained by replicating $\mu_{1\times 2} = \begin{bmatrix} \mu_x & \mu_y \end{bmatrix}$ across N rows.

$$V_{n\times 2} = P_{n\times 1} \times (E_1)_{2\times 1} + \mu_{N\times M}$$

$V_{n\times 2}$ holds the sample coordinates projected onto the First Principle Component, E_1, shifted by the mean.

For documented code, please see Q3.py and PCA.py

The scatter plots are on the next page.

The approximation for the first set seems accurate (considering the presence of noise in our

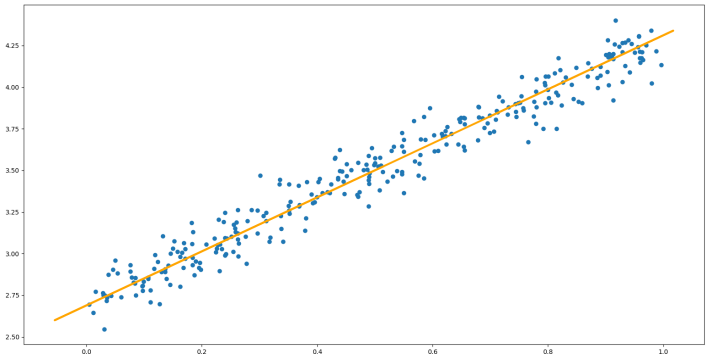

Figure 1: Dimensionality Reduction with PCA - Set 1

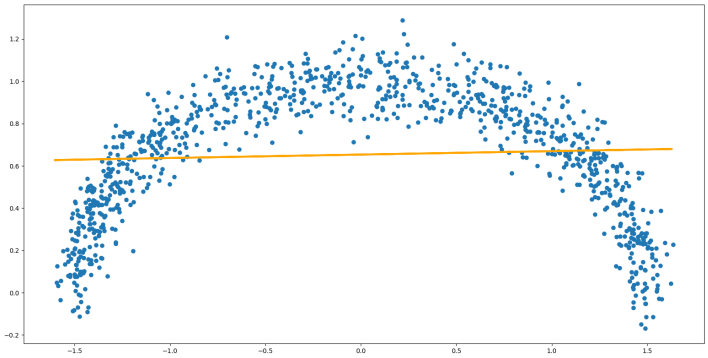

Figure 2: Dimensionality Reduction with PCA - Set 2

measurements) because the relation between the two variable is linear, and can be represented with projections on a single vector without much loss of information.
The approximation for the second set is not accurate because PCA looks to represent the data set with a finite number of fixed vectors. Therefore, it works best in data sets that show a linear relation. The second data set here follows a non-linear pattern that cannot be represented by finite vectors, least of all a single vector. A lot of information is lost from the set in trying to project it onto a single vector.