Problem 1:

$\because \quad y = k x^n$

$\therefore \quad \ln(y) = \ln k + n \ln(x)$

Since we have the values of x, and the values of y, we can using np.log(x) and np.log(y) in python to easily get the values of $\ln x$ and values of $\ln y$.

Than using

np.vstack([x, np.ones(len(x))]).T

to build the matrix A.

Finally, using

np.linalg.lstsq(A, y, round=None)[0]

to get the value of $\ln k$ and n.

The math function and matrix for this question should be:

$$\begin{array}{c} \ln(x) \\ \begin{bmatrix} \ln 400 & 1 \\ \ln 70 & 1 \\ \ln 45 & 1 \\ \ln 2 & 1 \\ \ln 0.3 & 1 \\ \ln 0.16 & 1 \end{bmatrix} \end{array} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{array}{c} \ln(y) \\ \begin{bmatrix} \ln 270 \\ \ln 82 \\ \ln 50 \\ \ln 4.8 \\ \ln 1.45 \\ \ln 0.97 \end{bmatrix} \end{array}$$

$$A \quad x = b$$

```
A
array([[ 5.99146455,  1.        ],
       [ 4.24849524,  1.        ],
       [ 3.80666249,  1.        ],
       [ 0.69314718,  1.        ],
       [-1.2039728 ,  1.        ],
       [-1.83258146,  1.        ]])
```

by calculating in python programming

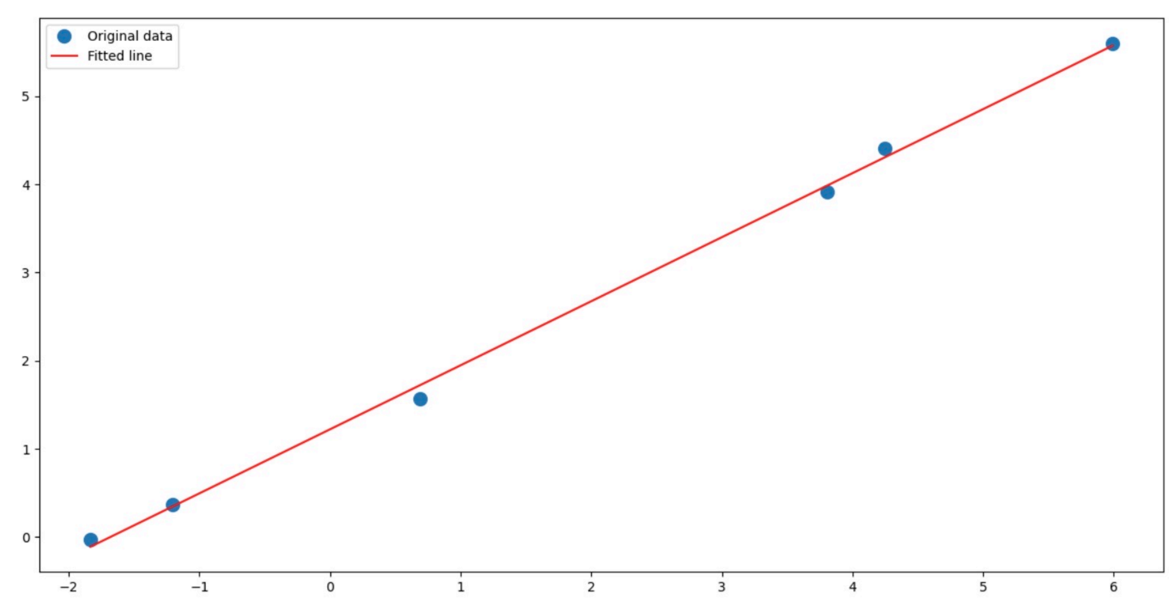
```
k_
1.2206253314907227
n
0.7265637989675519
```

$\ln k \approx 1.2206$

$n \approx 0.7266$

$\therefore k = e^{\ln k} \approx 3.3893$

Here is the log-log plot:



Problem 2:

a) Since we have the list of x and the list of y, we can find the list of y^2, xy, x^2 by using np.pow(y,2), np.multiply(x,y) and np.pow(x,2)

We can using the function

np.vstack([y², xy, x, y, np.ones(len(x))]).T

to build the matrix A.

The matrix A should like this:

```
A
array([[0.1521, 0.3978, 1.02  , 0.39  , 1.    ],
       [0.1024, 0.304 , 0.95  , 0.32  , 1.    ],
       [0.0729, 0.2349, 0.87  , 0.27  , 1.    ],
       [0.0484, 0.1694, 0.77  , 0.22  , 1.    ],
       [0.0324, 0.1206, 0.67  , 0.18  , 1.    ],
       [0.0225, 0.084 , 0.56  , 0.15  , 1.    ],
       [0.0169, 0.0572, 0.44  , 0.13  , 1.    ],
       [0.0144, 0.036 , 0.3   , 0.12  , 1.    ],
       [0.0169, 0.0208, 0.16  , 0.13  , 1.    ],
       [0.0225, 0.0015, 0.01  , 0.15  , 1.    ]])
```

$\therefore$ $Ax = b$:

$$A\begin{bmatrix}x_1\\x_2\\x_3\\\vdots\\x_{10}\end{bmatrix} = \begin{array}{c}\text{array of } x^2\\ \begin{bmatrix}\vdots\\\vdots\end{bmatrix}\end{array}$$

by using python programming to solve the leaste square problem above, we get the parameters a, b, c, d, e

```
a
-2.635625483712129
b
0.1436461825988984
c
0.551446963140356
d
3.2229403381059063
e
-0.43289427026445115
```

$\therefore\ a \approx -2.6356$

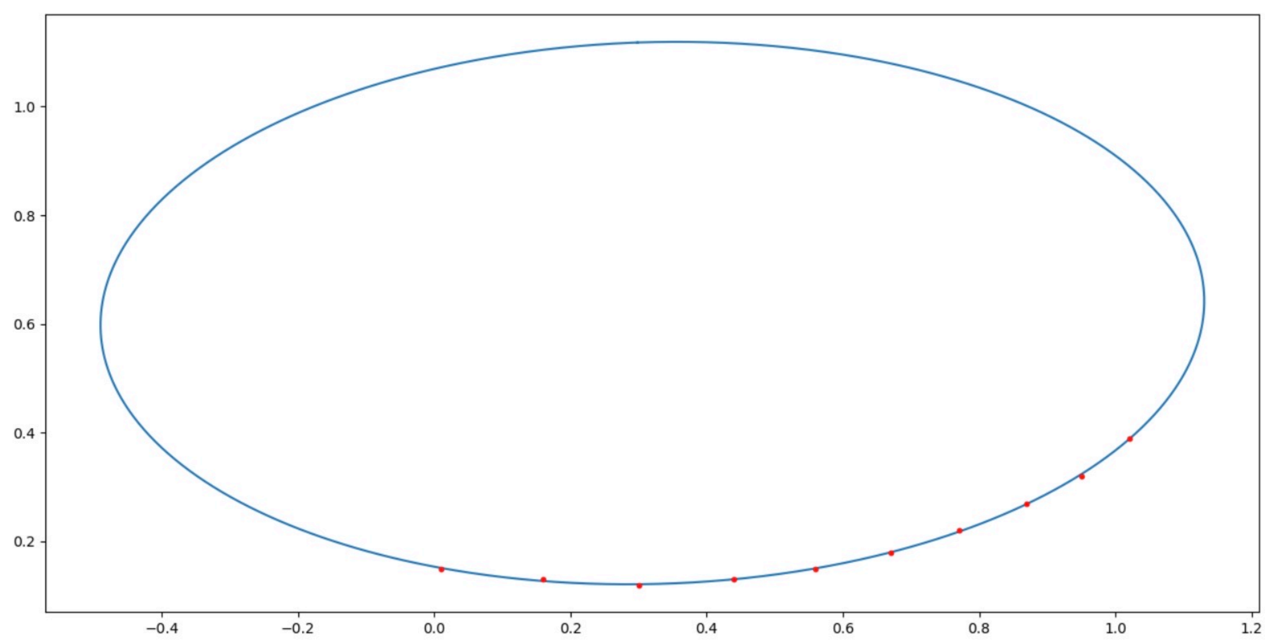
$b \approx 0.1436$

$c \approx 0.5514$

$d \approx 3.2229$

$e \approx -0.4329$

Here is the plot resulting orbit:



b). new parameters:

```
a
-3.9262502525330616
b
0.4968575214727604
c
0.47652648070457104
d
3.5910417330576903
e
-0.4491068572121828
```

$a \approx -3.9263$

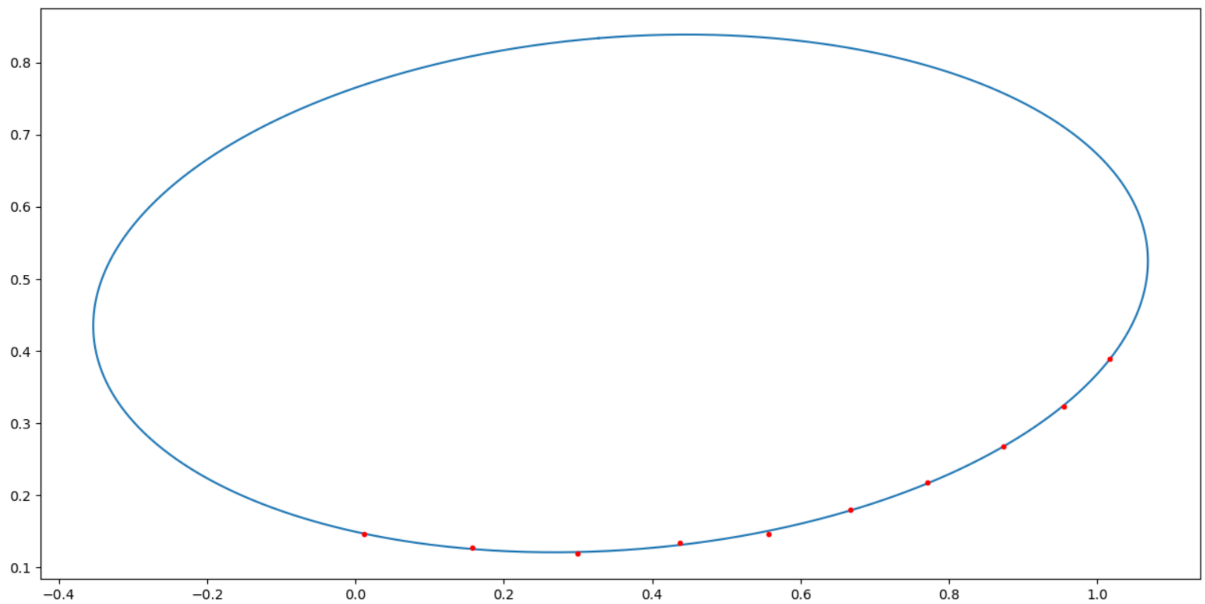
$b \approx 0.4969$

$c \approx 0.4765$

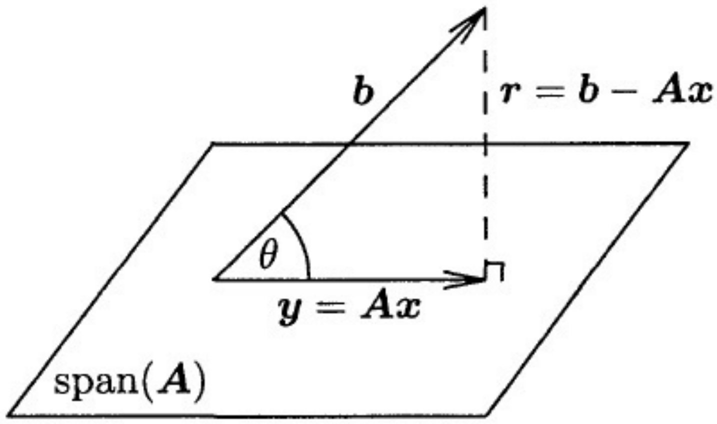
$d \approx 3.5910$

$e \approx -0.4491$

new plot:



by runing the system to solve the least square problem, we can find that the plot of the orbit seems be enlarged.



Reasons:

Because the least square problem is sensitivity and conditioning.

As the span(A) shows above, once we preturbed the data, We change the matrix A and b. Since we changed A and b, the angle θ will be changed as well.

Also for conditioning, a small changes to b can cause a relatively large change in y, and hence in the least squares solution x.

Then for θ, when residual is small, the condition number is approximately cond(A), when a residual is moderate, the condition number is square of cond(A), when residual is large, the condition number can be really large.

As the explain above, the plot of the orbit should be enlarged.