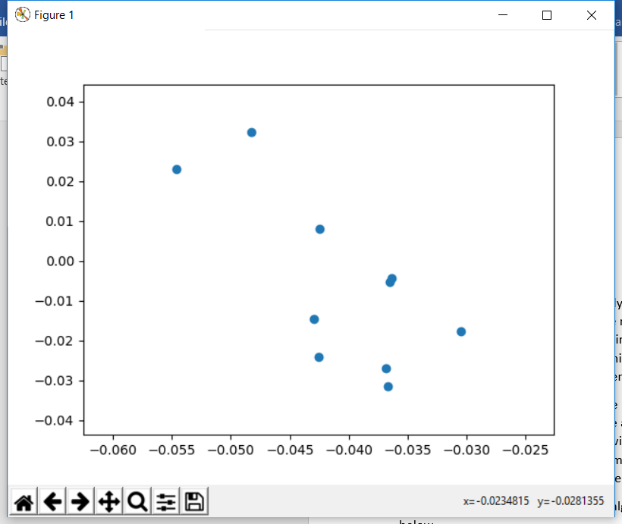
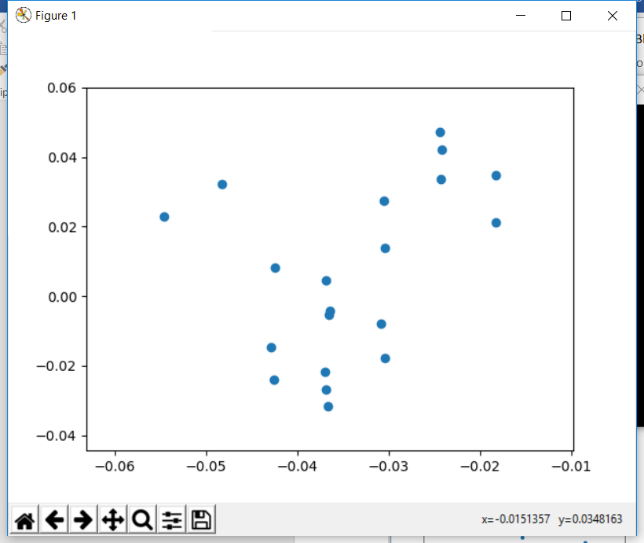
# Lab 3

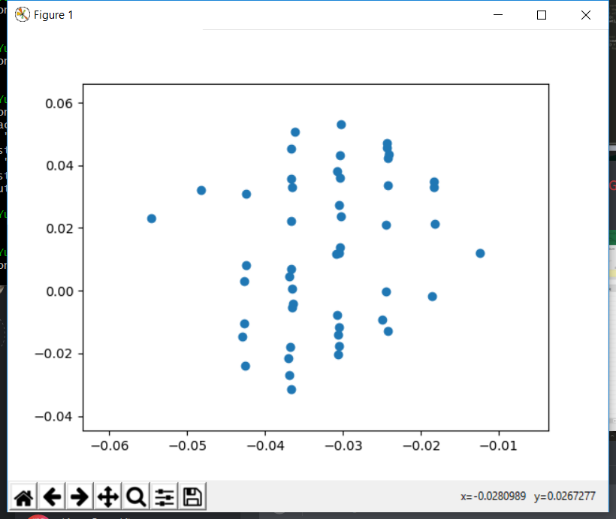
## Explanation of algorithm:

My algorithm appends the theoretically most interesting image of [1,1,1,1,1,1,1,1,1,1] to the be the first element of the list. It will then find the next image in the list with hamming distance = 1 and appends that to the list. If no element of hamming distance 1 is found, the algorithm will search through the list again looking for a number with hamming distance = 2 and so on. This repeats until k numbers have been appended to the list. The first element [1,1,1,1,1,1,1,1,1,1] will then be popped from the list.

I believe my algorithm is good because it ensures that all the images are different from each other. Also, consecutive images in the list will have a hamming distance of at least 1. This means that the hamming distance between every other image will be at least 2. I also believe that using this algorithm, the images on each end will likely have a large hamming distance. For example, when k = 20, the hamming distance between the first and last elements is equal to 5.

The plot of the coordinates when my algorithm is run for k = 10,20 and 50 are shown below.



As seen in the displayed plots, many of the points are evenly distributed on the graphs.

## Improvements:

My algorithm searches for images with a certain hamming distance. If the hamming distance is 0, the number will not be selected. This means that the image [1,1, 1,1,1,1,1,1,1,1] will never be appended to the list using my algorithm.

## Results:

Overall, my algorithm is good for finding the best algorithms. The algorithm finds the closest image to the image that is theoretically the most interesting. It then finds a list of images that are spaced a good distance from each other.