Introduction:

I utilized the Jupyter notebook (lab) provided within Coursera platform inside week 5 for my coding. I then downloaded the notebook as a Python script, which I have included alongside this report.

Strategy 1:

I entered the four last digits of my student ID and with that generated my set of initial centroids. I first viewed the initial utilized the initial centroids locations on the data through scatter plot. From time to time I visualized my centroid and assigned clusters through colored scattered points.

Then I created a function "dist_toCentroid" that takes a set of centroids and a data frame with two columns (x and y). The function then generate a new set of centroids and a data frame. The generated data frame contains distances of each point from the centroids, a column known as closest which indicate the cluster that each point is assigned, and a column named color which contains letters respective to the assigned cluster (the letter can be used to assign colors to data points in scatter plots).

Then I produced while loop that ensures that the function "dist_toCentroid" is used to iteratively generate new centroids until the centroids don't change (when no single data points change clusters anymore). This is the final centroids that I submitted in the Quiz section.

Finally I calculated the objective function, by calculating the squared errors for each cluster data point to the cluster's centroid, and then summed up the squared errors. This sum of squared errors is the objective function which is submitted in the quiz section.

This process was executed for both k1 (K=3) and k2 (K =5). To be sure that my result was correct I in addition to carrying out KMean from scratch in Python, cross checked with Sklearn.

My results were the same for both from scratch and with Sklearn and are shown below:

K1 (K=3) Result

```
Final centroid: {1: [[7.2397511895844495, 2.4820826910731952]], 2: [[4.
                    833753175392286, 7.3160582360435 74]], 3: [[3.248964
                    2305948876, 2.5802769113756954]]}
Objective function: 1338.1059838
```

K2 (K=5) Result

```
Final centroid: {1: [[2.8749081274874264, 7.010822811156844]], 2: [[2.6
                    81986334188929, 2.094615867800809]], 3: [[7.5561678
                    22397726, 2.2351679598575 34]], 4: [[5.2572241053885
                    98, 4.2551862565934 18]], 5: [[6.775311757464789, 8.
                    1154019364069 47]]}

Objective function: 598.678798534
```

Strategy 2:

For Strategy two I started with only the first centroid and then computed the remaining initial centroids by assigning data point that is furthest from the previous centroid(s).

After obtaining the remaining initial centroids, I computed the final centroids and the objective function as in Strategy1. My results are shown below:

K1 (K=4) Result

```
Final centroid: {1: [[7.252626831256577, 2.4001582635520533]], 2: [[3.2
                    285300905383707, 2.5240486292057867]], 3: [[6.62592
                    53846324615, 7.5761491676226 78]], 4: [[2.9054774114
                    449513, 6.9051227633399 48]]}

Objective function: 789.237972218
```

K2 (K=6) Result

```
Final centroid: {1: [[4.022935100912096, 3.870824728369343]], 2: [[7.75
                    6483249146484, 8.556689279063415]], 3: [[2.82544756
                    0760127, 1.6546753620598549]], 4: [[2.5122573578867
                    69, 7.057170033892222]], 5: [[7.477682316761515, 2.
                    2990031525317582]], 6: [[5.464277356727894, 6.83771
                    3536435891]]}

Objective function: 484.04926087
```

References:

1. Lecture videos
2. Slack conversation with peers
3. https://www.youtube.com/watch?v=1XqG0kaJVHY