COLLEGE OF INFORMATION TECHNOLOGY EDUCATION
ITE 404 – Introduction to Data Science in Python

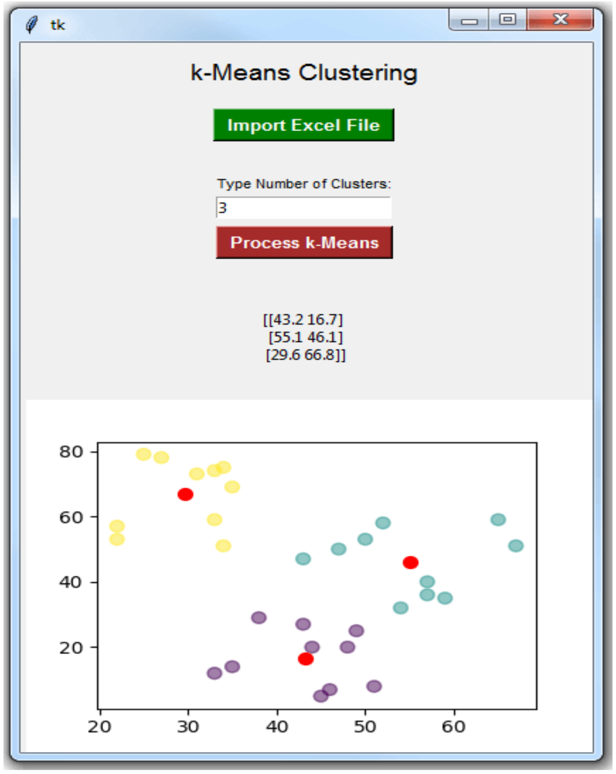
Name: Araneta, Emeric Joseph Z., Macatantan, Bryan D.	Date: 3/25/2023
Section: CS32S6 Program: ITE 404	Instructor: Ms. Nila D. Santiago
Assessment Task: Assignment 3.1 K-Means Clustering in Python PART 1	

Example of K-Means Clustering in Python
PART 1:
K-Means Clustering is a concept that falls under Unsupervised Learning. This algorithm can be used to find groups within unlabeled data. To demonstrate this concept, I'll review a simple example of K-Means Clustering in Python.

Topics to be covered:

- Creating the DataFrame for two-dimensional dataset
- Finding the centroids for 3 clusters, and then for 4 clusters
- Adding a graphical user interface (GUI) to display the results

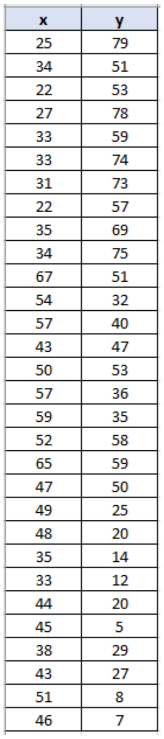
By the end of this activity, you'll be able to create the following GUI in Python:



```
[[43.2 16.7]
 [55.1 46.1]
 [29.6 66.8]]
```

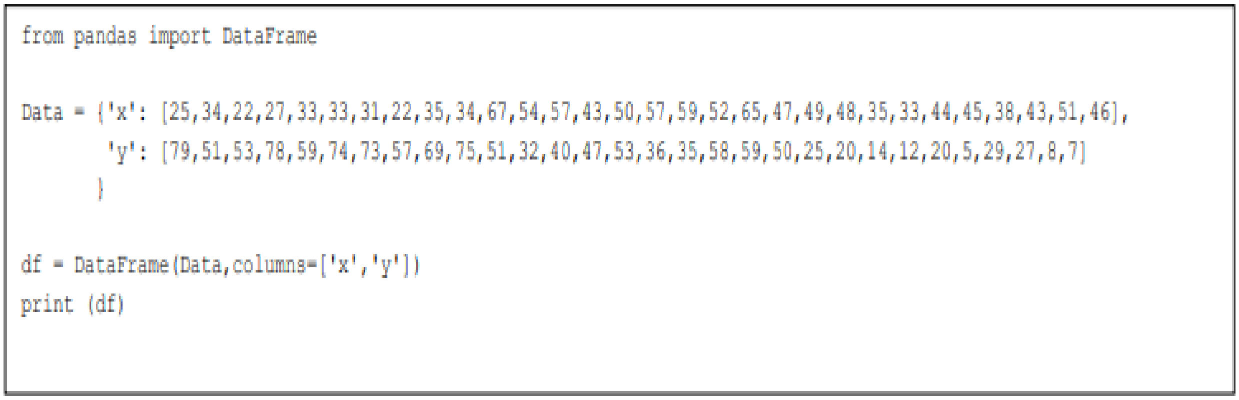
Example of K-Means Clustering in Python

To start, let’s review a simple example with the following two-dimensional dataset:



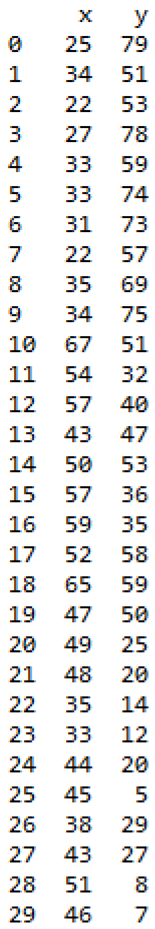
x	y
25	79
34	51
22	53
27	78
33	59
33	74
31	73
22	57
35	69
34	75
67	51
54	32
57	40
43	47
50	53
57	36
59	35
52	58
65	59
47	50
49	25
48	20
35	14
33	12
44	20
45	5
38	29
43	27
51	8
46	7

You can then capture this data in Python using pandas DataFrame:



```
from pandas import DataFrame

Data = {'x': [25,34,22,27,33,33,31,22,35,34,67,54,57,43,50,57,59,52,65,47,49,48,35,33,44,45,38,43,51,46],
        'y': [79,51,53,78,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,25,20,14,12,20,5,29,27,8,7]
       }

df = DataFrame(Data,columns=['x','y'])
print (df)
```

If you run the code in Python, you’ll get this output, which matches with our dataset:



```
     x   y
0   25  79
1   34  51
2   22  53
3   27  78
4   33  59
5   33  74
6   31  73
7   22  57
8   35  69
9   34  75
10  67  51
11  54  32
12  57  40
13  43  47
14  50  53
15  57  36
16  59  35
17  52  58
18  65  59
19  47  50
20  49  25
21  48  20
22  35  14
23  33  12
24  44  20
25  45   5
26  38  29
27  43  27
28  51   8
29  46   7
```

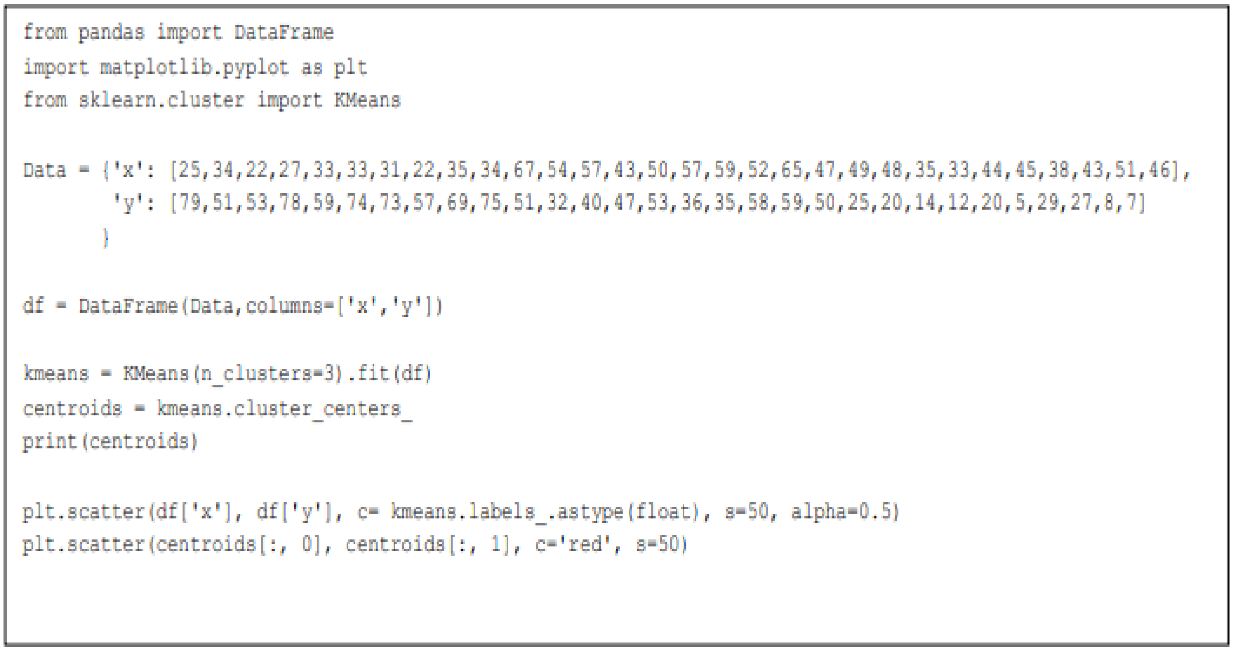
Next you'll see how to use sklearn to find the centroids for 3 clusters, and then for 4 clusters.
K-Means Clustering in Python – 3 clusters

Once you created the DataFrame based on the above data, you'll need to import 2 additional Python modules:

- matplotlib – for creating charts in Python
- sklearn – for applying the K-Means Clustering in Python

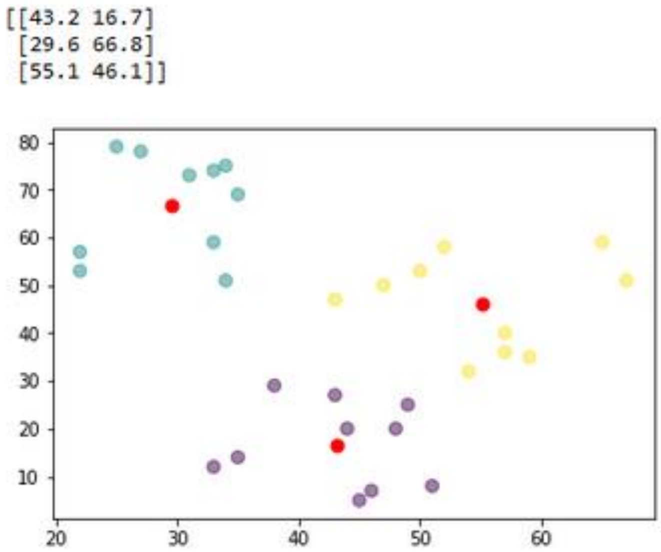
In the code below, you can specify the number of clusters. For this example, assign 3 clusters as follows:

KMeans(n_clusters=3).fit(df)



```
from pandas import DataFrame
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

Data = {'x': [25,34,22,27,33,33,31,22,35,34,67,54,57,43,50,57,59,52,65,47,49,48,35,33,44,45,38,43,51,46],
        'y': [79,51,53,78,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,25,20,14,12,20,5,29,27,8,7]
       }

df = DataFrame(Data,columns=['x','y'])

kmeans = KMeans(n_clusters=3).fit(df)
centroids = kmeans.cluster_centers_
print(centroids)

plt.scatter(df['x'], df['y'], c= kmeans.labels_.astype(float), s=50, alpha=0.5)
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)
```

Run the code in Python, and you’ll see 3 clusters with 3 distinct centroids:



```
[[43.2 16.7]
 [29.6 66.8]
 [55.1 46.1]]
```

Note that the center of each cluster (in red) represents the mean of all the observations that belong to that cluster.

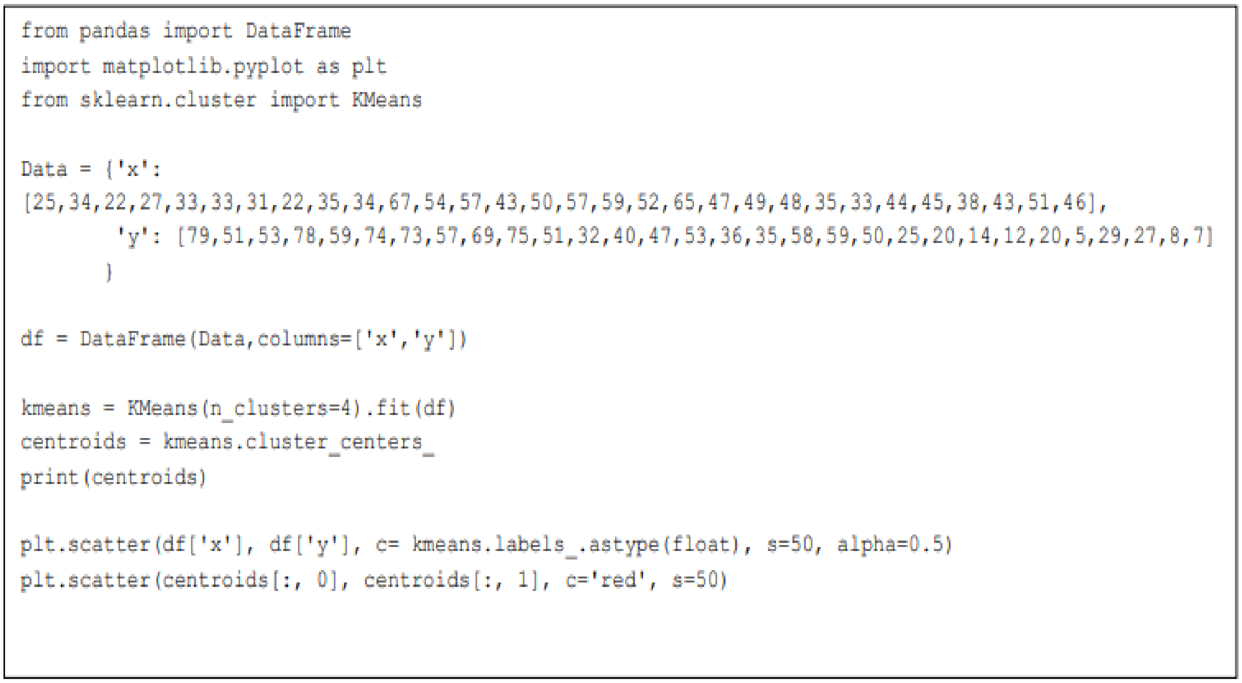
As you may also see, the observations that belong to a given cluster are closer to the center of that cluster, in comparison to the centers of other clusters.

K-Means Clustering in Python – 4 clusters

Let’s now see what would happen if you use 4 clusters instead. In that case, the only thing that you’ll need to do is to change the n_clusters from 3 to 4:

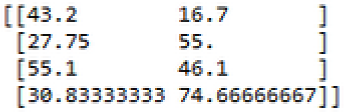
KMeans(n_clusters=4).fit(df)

And so, your full Python code for 4 clusters would look like this:

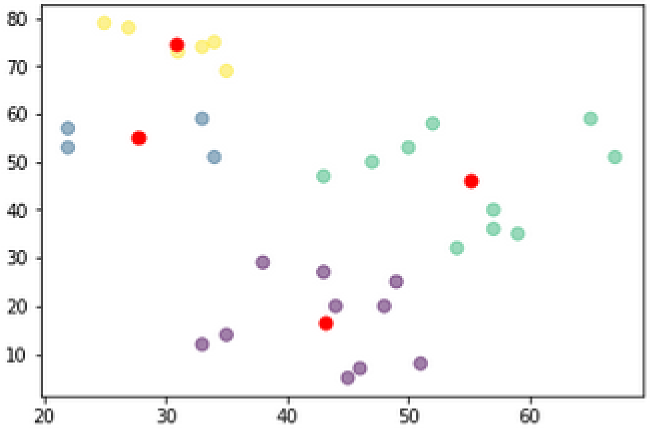


```
from pandas import DataFrame
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

Data = {'x':
[25,34,22,27,33,33,31,22,35,34,67,54,57,43,50,57,59,52,65,47,49,48,35,33,44,45,38,43,51,46],
        'y': [79,51,53,78,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,25,20,14,12,20,5,29,27,8,7]
       }

df = DataFrame(Data,columns=['x','y'])

kmeans = KMeans(n_clusters=4).fit(df)
centroids = kmeans.cluster_centers_
print(centroids)

plt.scatter(df['x'], df['y'], c= kmeans.labels_.astype(float), s=50, alpha=0.5)
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)
```

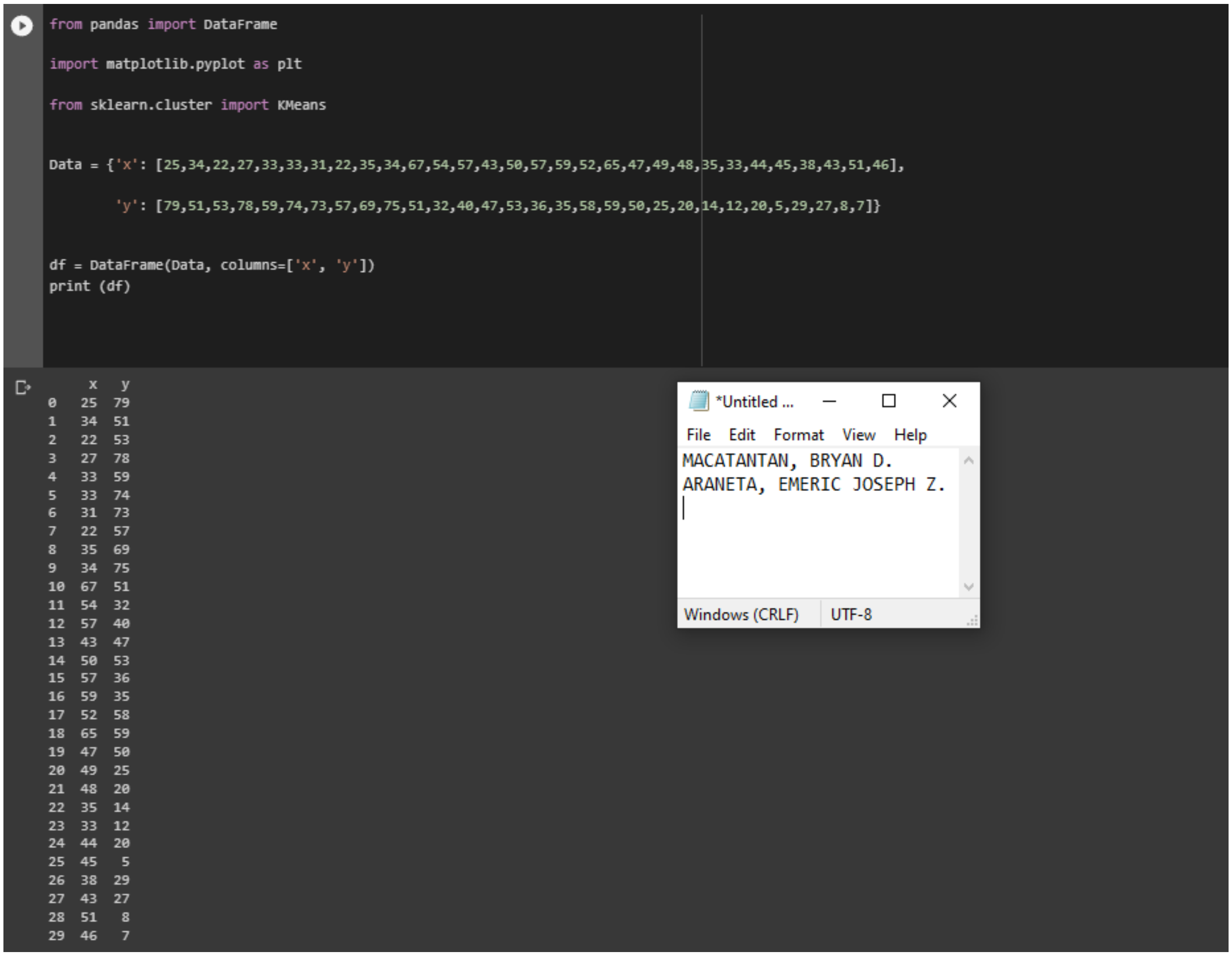
Run the code, and you’ll now see 4 clusters with 4 distinct centroids:



```
[[43.2        16.7       ]
 [27.75       55.        ]
 [55.1        46.1       ]
 [30.83333333 74.66666667]]
```



Answers:

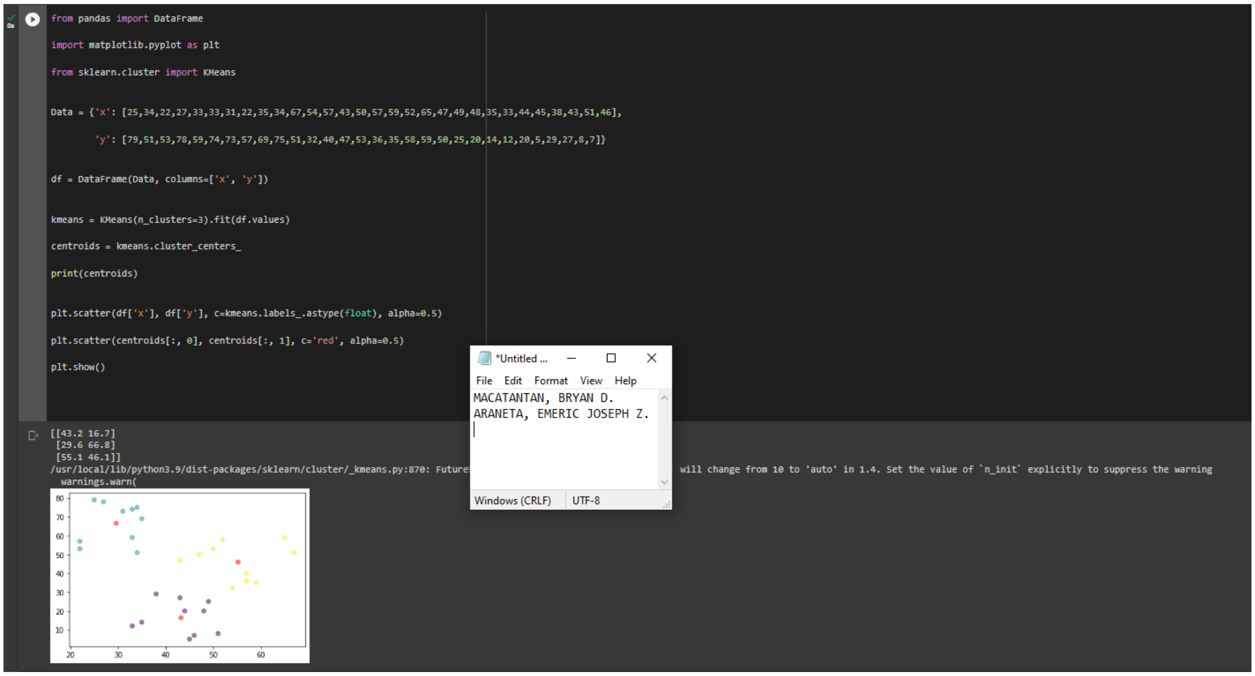


```
from pandas import DataFrame

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans


Data = {'x': [25,34,22,27,33,33,31,22,35,34,67,54,57,43,50,57,59,52,65,47,49,48,35,33,44,45,38,43,51,46],

        'y': [79,51,53,78,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,25,20,14,12,20,5,29,27,8,7]}


df = DataFrame(Data, columns=['x', 'y'])
print (df)
```

```
     x   y
0   25  79
1   34  51
2   22  53
3   27  78
4   33  59
5   33  74
6   31  73
7   22  57
8   35  69
9   34  75
10  67  51
11  54  32
12  57  40
13  43  47
14  50  53
15  57  36
16  59  35
17  52  58
18  65  59
19  47  50
20  49  25
21  48  20
22  35  14
23  33  12
24  44  20
25  45   5
26  38  29
27  43  27
28  51   8
29  46   7
```

MACATANTAN, BRYAN D.
ARANETA, EMERIC JOSEPH Z.

First, we input the information that will be utilized for the k-means clustering in the appropriate columns for the first set of code (x and y). The values are correctly printed in order in their respective columns, as seen in the image up top.

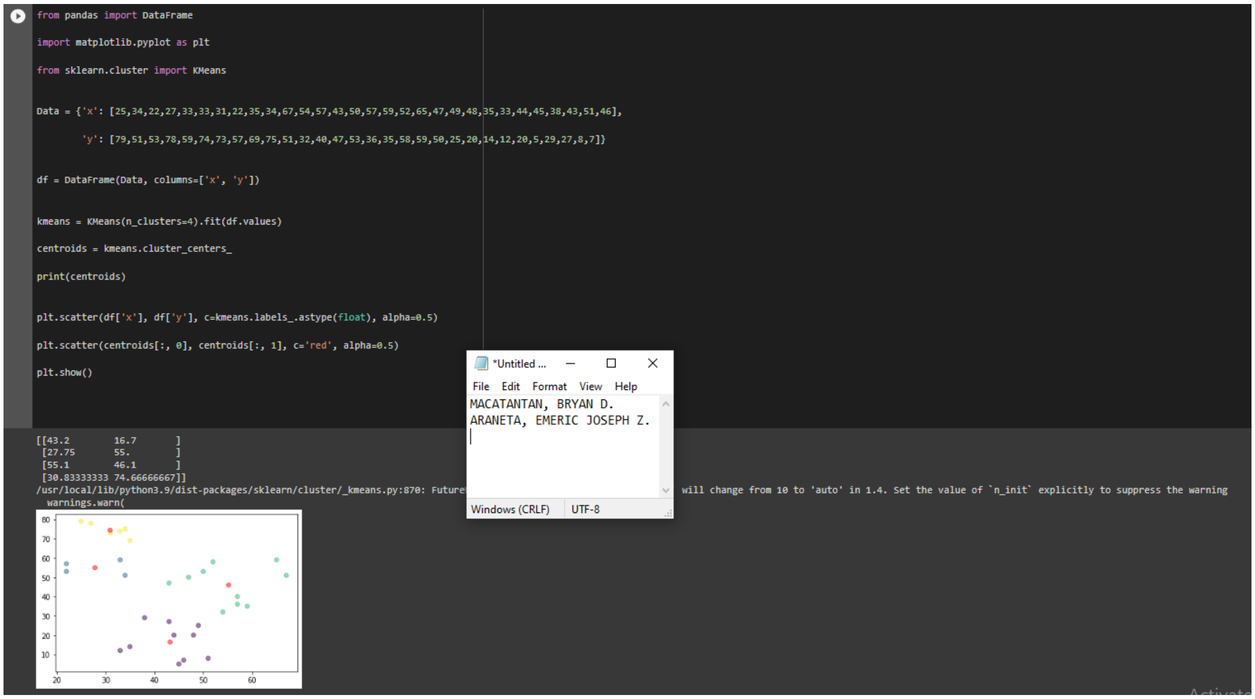


```
from pandas import DataFrame

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans


Data = {'x': [25,34,22,27,33,33,31,22,35,34,67,54,57,43,50,57,59,52,65,47,49,48,35,33,44,45,38,43,51,46],

        'y': [79,51,53,78,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,25,20,14,12,20,5,29,27,8,7]}


df = DataFrame(Data, columns=['x', 'y'])


kmeans = KMeans(n_clusters=3).fit(df.values)

centroids = kmeans.cluster_centers_

print(centroids)


plt.scatter(df['x'], df['y'], c=kmeans.labels_.astype(float), alpha=0.5)

plt.scatter(centroids[:, 0], centroids[:, 1], c='red', alpha=0.5)

plt.show()
```

```
[[43.2 16.7]
 [29.6 66.8]
 [55.1 46.1]]
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: Future will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

```
MACATANTAN, BRYAN D.
ARANETA, EMERIC JOSEPH Z.
```

Second, the matplotlib and sklearn libraries are imported to create charts and run k-means clustering on the data. As you can see in the figure above, we've also included code that enables us to plot the clusters. The red dots symbolizes the center of the clusters, 3 clusters are shown in this example that is why there are 3 red dots.



```
from pandas import DataFrame

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans


Data = {'x': [25,34,22,27,33,33,31,22,35,34,67,54,57,43,50,57,59,52,65,47,49,48,35,33,44,45,38,43,51,46],

        'y': [79,51,53,78,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,25,20,14,12,20,5,29,27,8,7]}


df = DataFrame(Data, columns=['x', 'y'])


kmeans = KMeans(n_clusters=4).fit(df.values)

centroids = kmeans.cluster_centers_

print(centroids)


plt.scatter(df['x'], df['y'], c=kmeans.labels_.astype(float), alpha=0.5)

plt.scatter(centroids[:, 0], centroids[:, 1], c='red', alpha=0.5)

plt.show()
```

```
[[43.2        16.7       ]
 [27.75       55.        ]
 [55.1        46.1       ]
 [30.83333333 74.66666667]]
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: Future will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

```
MACATANTAN, BRYAN D.
ARANETA, EMERIC JOSEPH Z.
```

Third, we’ve changed it into 4 clusters. As shown in the example the upper left clusters are now divided into two clusters while the other lower and upper right clusters remain the same.

“We affirm that we have not given or received any unauthorized help on this assignment, and that this work is our own.”