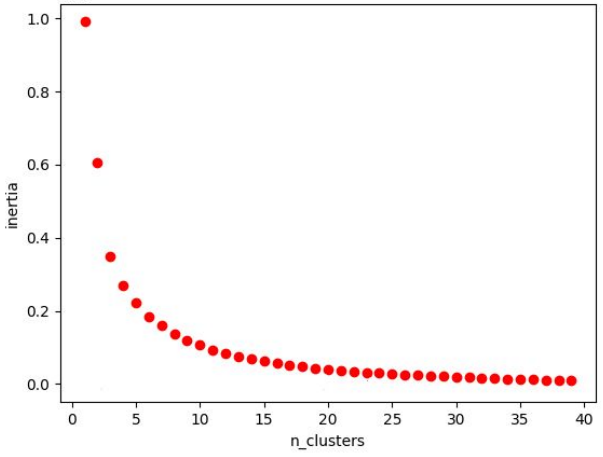
Data mining - COMP40370
Thomas Igoe - 17372013

Q1
The data is mostly spread out, with a cluster in the bottom left (around 2.5, 7.5). Data near this tightly clustered group is included with this cluster, with data further away is included in a 2nd cluster. The datapoint at 23.4 == x is not close enough to either cluster to be included and makes up its own “cluster”.

Q2

4
Config1 and config2 are almost entirely different from one another. Because the only difference is the greatly increased number of maximum runs I would expect that config2 (with increased number of maximum runs) is more accurate, as it is taking a wider variety of starting centroids and hence the local minimums found should more often resemble the global minimums.

6
As mentioned in this website (https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/) evaluating the optimal number of clusters can be done in a variety of ways. I chose to emulate the elbow method, and tried a number of k-means configurations with increasing number of clusters, and graphing that against the inertia value (Sum of squared distances of samples to their closest cluster center.) Upon visual inspection, the sweet spot seems to be around 5-7, indicating that n_clusters = 5 would be more optimal than 3.



Q3

1
Upon visual inspection of the plot generated in part 2, the clustering algorithm has divided some of the densely clustered data points into 2 clusters when they should most likely be part of the same cluster

6
The k-means algorithm operating on this data has made some mistakes, most notably on the pair of u-shaped clusters on the left of the graph. Because the centroids were calculated to be in the center of the 2 u-shaped clusters, the ends of the clusters that come close to one another are marked as part of the other cluster which would be incorrect for this dataset.

However the dbscan algorithm has correctly identified these clusters and can successfully separate them when eps == 0.4, as the minimum distance between nodes of the 2 clusters is > 0.4. This does however incorrectly categorize the cluster at the top. As the cluster becomes more sparse as the distance becomes too great for dbscan to categorize it correctly.

When eps is changed to be 0.8, the distance between the 2 clusters is not enough to distinguish them as separate clusters so the algorithm marks them as the same. (However this may be correct as it is very unusual to see data create this shape, almost looping around one another) However this increased eps does mean that the cluster at the top is correctly assigned the same cluster for its entirety.

It is hard to say which is the “best” algorithm, however for this dataset I would say that dbscan with esp=0.4 was most successful. It correctly split the 2 pairs of clusters, however it incorrectly classified the cluster at the top due to its sparseness towards the right of the cluster giving merit to esp=0.8.