Segmentation Test Analysis

Objective:

Use K-nearest neighbor to determine number of clusters in segmentation test in excel.

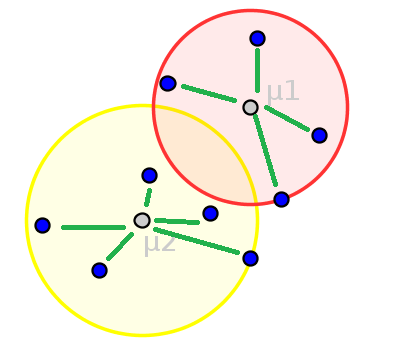
Assumptions:

* The data provided uses arithmetic averages, which does not assume distribution of data.
* Training data and test data set are not determined.

Analysis:

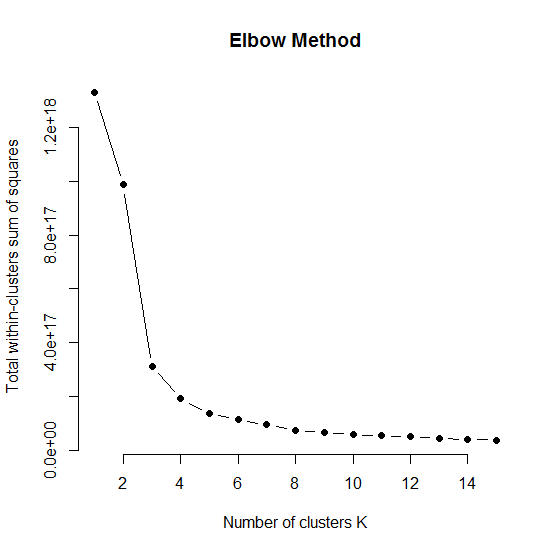
**Elbow Method**

In order to perform the k nearest neighbor cluster algorithm, I would need to take an elbow method to determine the best number of clusters. The elbow method takes a within-cluster sum of squares for each cluster, for lower numbers of clusters there are more within-cluster sum of squares and for higher numbers of clusters there are less within-cluster sum of squares. To simplify, please view graph below:



There are two clusters, one highlighted in yellow and another highlighted in red. The blue dots are the data points and the grey dots are the centroids ("centers of each cluster"). The distances from the centroids and the data points are in the green lines, in which the sum of this is the within-cluster sum of squares. As you can imagine a lower number of clusters will have more within-cluster sum of squares because each data point will inherently be further away from each centroid because there are less centroids and the opposite applies to a higher number of clusters. With more centroids, there are less distances between the centroid and its clusters data points. With less centroids, there are more distances between the centroid and its clusters data points.

If more centroids produce less distances between each data point and its centroid, why not use as many clusters as possible? The answer to this is to use the **optimal** number of clusters, not the **most** number of clusters. In the elbow method, the way to define an optimal number of clusters would be as k number of clusters increase, pick the cluster that has the least amount of decrease in within-cluster sum of squares. To provide example, I have generated the elbow method using R for our segmentation data.



As you can see from the elbow method chart, the recommended number of clusters is 4. After cluster 4, the decrease of within-clusters sum of squares starts to be steady indicating more clusters will not make the model stronger.

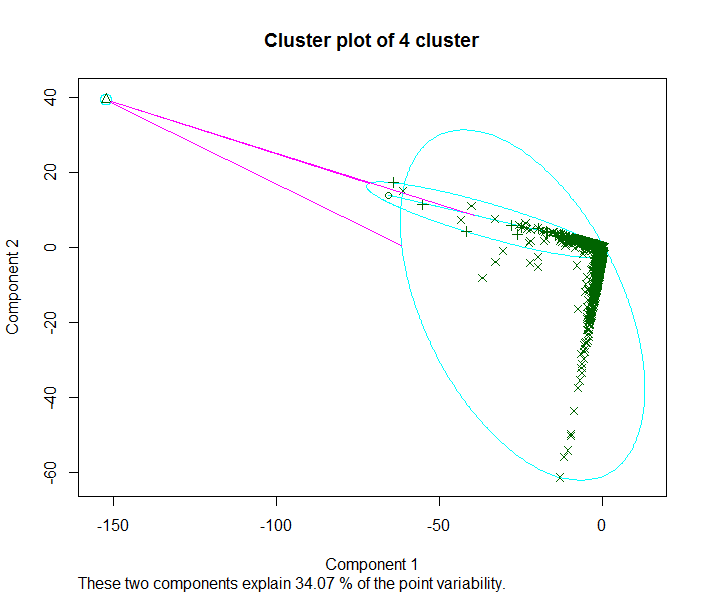
**K-means clustering:**

The K-means clustering will be performed with our conclusive 4 clusters from the elbow method mentioned before. After executing my K-means algorithm code, I was able to get cluster results:



Most of the data points fall in cluster 4 with (45,950) which is 99% of the data. This shows most of data is bucket in this category which is not surprising since the product value and volume are both low. The outliers appear to be clusters 1 and 2 which has 6 accounts in this entire sample of 46,025 accounts. Based on my analysis, it doesn't appear that the k-means algorithm is incorrect instead the data itself is very skewed.

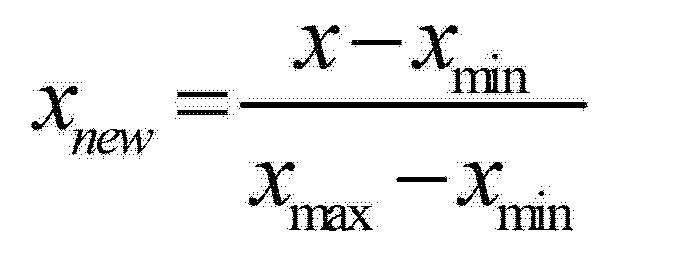
**Cluster plot:**

To get a better visual of the data in 4 clusters, the graph below shows the 4 clusters in the data set.

The graph displayed shows the 4 clusters in a 2 dimensional frame. The clusplot chart above uses principal components analysis to put our 10 dimensional segmentation data into 2 dimensions to help visualize our clusters. It appears 3 clusters hover in the same cluster but this could be because some data points are in other dimensions not displayed. This is where 34.07% resembles the amount of data being captured in our plot.

**Other solutions:**

In additional to using the elbow method, I have tried using Gap Statistic and Silhouette Method which both measure for optimal cluster size but unfortunately there is too much data, the quantity and volatility is huge. I used a formula to normalize the data (fit all data points from 0 to 1) in their respective variables but was still unable to calculate Euclidean distance to perform those analysis. The formula for normalizing data is below:



**Conclusion:**

Based on my analysis, I conclude that by using the elbow method I was able to make the determination that the most optimal cluster for k means is when k = 4. The reason the cluster sizes are very lopsided (most data in cluster 4) is because there are huge differences in PROD\_VAL ranging from minimum of 0 to millions, see chart below. The chart output was from using R summary and pasting the results into excel.



My solution and suggestions would be to experiment more on the actual data itself. Bucketing the outliers into one set and try to separate them from the k means clustering. Including those outliers causes problems to bucket the 4th cluster and once it is removed the fourth cluster will have more disparity.