PART - 1

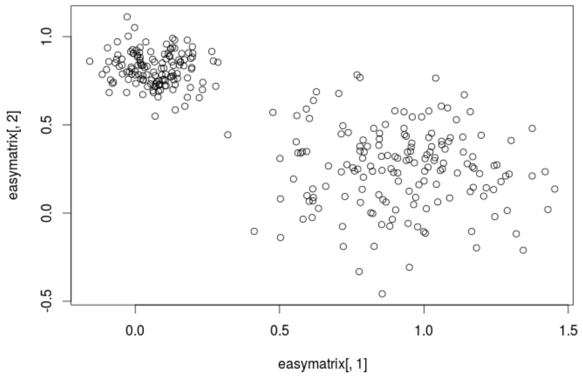
K-MEANS CLUSTERING:

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid using a proximity measure (e.g., Euclidean distance)
- Number of clusters, K, must be specified

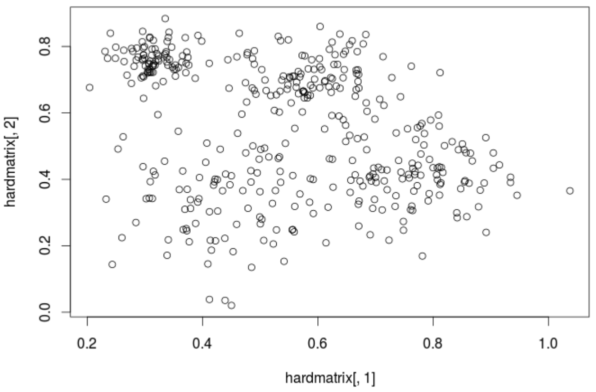
PREPROCESSING:

Upon careful analysis of the datasets, it was found that the datasets are free of outliers. Shown below is the scatter plot of A.Easy and A.Hard. The scatter plot of A.Easy reveals the existence of two visually distinct clusters. The given dataset (which has been pre-clustered already) also contains two unique cluster ids ‘1’ and ‘2’ in the third column. On the other hand, A.Hard can be thought of four clusters although not very distinct. The cluster column of A.Hard has four unique ids, ‘1’, ‘2’, ‘3’, ‘4’.

For wine dataset, from the previous analysis (HW4), it was evident that the dataset was free from outliers. It can be normalized, but upon normalization the SSE and SSB values reduce greatly and make it difficult to comprehend the Elbow graph (explained in next section). Normalization also brings down the field alcohol to 0 to 1 range which is undesirable (Result from previous assignment: Alcohol had a correlation of 0.5 with quality and so not normalizing alcohol gives more weightage to alcohol attribute in distance calculation because all other attributes except alcohol ranges from 0 to 1). Hence normalization is not preferred on wine dataset.



Scatterplot of A.easy dataset



Scatterplot of A.hard dataset

A) IMPLEMENTATION AND DESIGN:

The program consists of major parts:

1) Code to perform k-means clustering
2) Function to plot the clustered output
3) Function to calculate SSE and SSB
4) Function to calculate Average Silhouette width
5) Function to calculate Elbow graph

1. A random function is used to choose ‘k’ points in the dataset as initial means. Then each point in the dataset is assigned a cluster based on its distance with these initial means. Once all the points have been assigned some cluster, a counter decides whether another iteration is necessary or not. This counter increments when a datapoint is assigned a cluster id that is different from its previous iteration’s cluster

id. (All clusters ids are initially assigned as -1). The dataset is said to be converged when this counter reads zero, (i.e) this iteration has not changed the cluster id of any data point. If it is not zero, then new initial means for next iteration are calculated as the mean of current clusters.

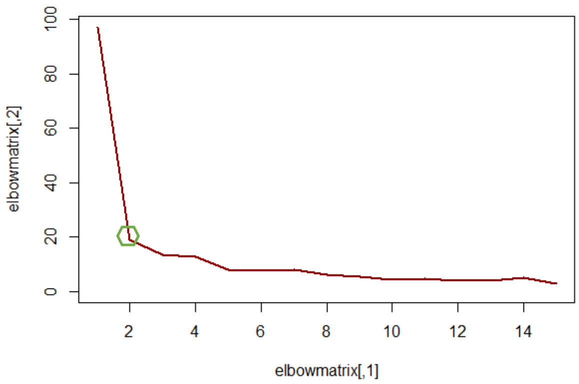
2. The function plotkmeans takes in datapoints and their clustered ids as input and assigns different colors and labels for each clusters and outputs a scatter plot.

3. The function to calculate SSE and SSB takes in same input as plotkmeans and returns total SSE of all clusters. SSE and SSB together is called Sum of Squared errors and it is constant for a given dataset. SSE is the sum of distance of all points in that cluster with its centroid (or mean). Overall SSE is the sum of SSE of all clusters. SSB is the distance of a centroid with overall centroid times the number of datapoints in that cluster. Overall SSB is the sum of SSB of all clusters. Depending on the number of clusters, SSE and SSB vary. A clustering with large number of good clusters will have low SSE and high SSB. A clustering with very few clusters will have high SSE and low SSB.

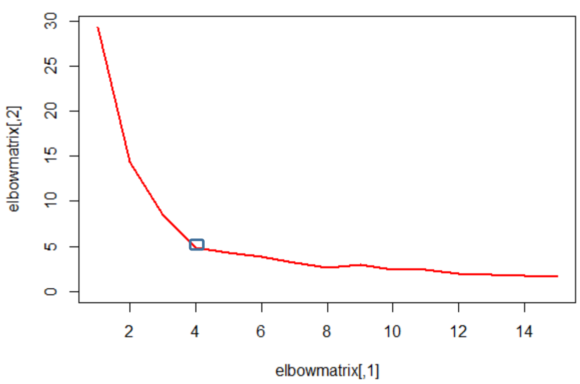
4. Silhouette width of a cluster is another is a measure that combines both internal cohesion (SSE) and external cohesion (SSB). It has values between -1 to 1. 1 being ideal and -1 being worst case. Silhouette width of a cluster is the average silhouette coefficient of all points in that cluster. For a data point i, S.C. = b -a / max(a,b) where a=avg. dist of i to all points in its cluster; b= min(avg. dist to other clusters)

5. Silhouette width and SSE and SSB are used to validate different type of clusterings given the number of clusters but none of these measure helps in deciding the number of clusters. An Elbow graph is a graph plotted against overall SSE and number of clusters. The final elbow-graph function calculates overall SSE values for different values of k and generates the elbow graph. The point where there is a steep decrease in elbow plot is the optimal number of clusters. Beyond that point the SSE stabilizes. Elbow plots of A.easy and A.hard show the optimal number of clusters is 2 for A.easy and 4 for A.hard.

1. ELBOW PLOT FOR A.EASY



2. ELBOW PLOT FOR A.HARD



B) SSE and SSB of true clusters of A.easy dataset:

No of points in cluster 1 : 138 Mean of cluster 1 : 0.06968869 , 0.81709841 SSE: 2.359592
No of points in cluster 2 : 162 Mean of cluster 2 : 0.9198995 , 0.2508134 SSE: 16.998597
SSE: 19.358189 SSB: 77.764384 SSE+SSB: 97.122573

Cluster id	Cluster size	Average silhouette width
1	138	0.9131621
2	162	0.7810652

Average silhouette coefficient of all clusters 0.847114
Low SSE, high SSB and average silhouette coefficient close to 1 indicate good clustering.

SSE and SSB of true clusters of A.hard dataset:

```
No of points in cluster 1 : 89          Mean of cluster 1 : 0.3187732, 0.7620575        SSE: 0.3128477
No of points in cluster 2 : 100         Mean of cluster 2 : 0.5915308, 0.7094224        SSE: 0.9025336
No of points in cluster 3 : 97          Mean of cluster 3 : 0.4410650, 0.3425525        SSE: 2.4301187
No of points in cluster 4 : 114         Mean of cluster 4 : 0.7635166, 0.4254296        SSE: 1.9107155
SSE: 5.556216   SSB: 23.748126 SSE+SSB: 29.304341
```

Cluster id	Cluster size	Average Silhouette width
1	89	0.9318807
2	100	0.8875274
3	97	0.8259718
4	114	0.8596403

Average silhouette coefficient of all clusters: 0.876255

Low SSE, high SSB and average silhouette coefficient close to 1 indicate good clustering. Also the size of all clusters are comparable indicating that there is no empty cluster. Empty clusters occur when an outliers is chosen as initial mean for first iteration of k-means clustering and it never converges.

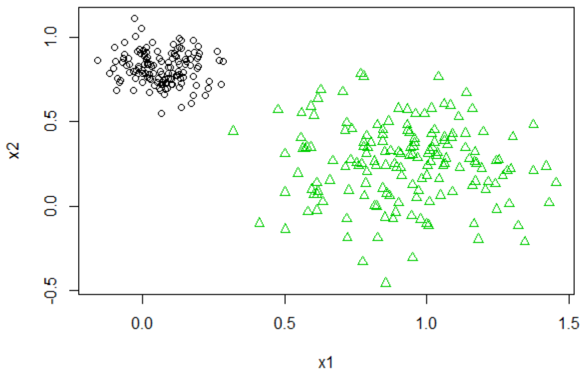
C) Run k-means 3 different times for easy dataset

1st run k =2

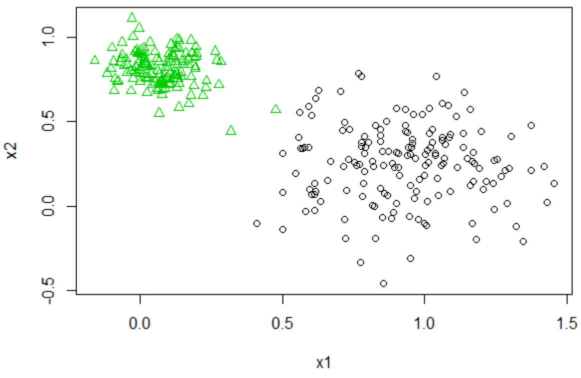
```
"Rows used as initial means"
225, 173
"Initial means"
[x1]            [x2]
1.1364716 0.3457385
0.8042914 0.3785595
Converged at 5
No of points in cluster 1 : 160         Mean of cluster 1 : 0.9264137, 0.2476079        SSE: 16.295196
No of points in cluster 2 : 140         Mean of cluster 2 : 0.07438974, 0.81267206      SSE: 2.782616
SSE: 19.077812  SSB: 78.044761 SSE+SSB: 97.122573
   Cluster id Cluster size Average silhouette width
[1,]       1        160           0.7851322
[2,]       2        140           0.9073197
Average silhouette coefficient of all clusters: 0.846226
```

	1	2
1	138	0
2	2	160

TRUE CLUSTER SCATTER PLOT



OUTPUT OF KMEANS CLUSTERING



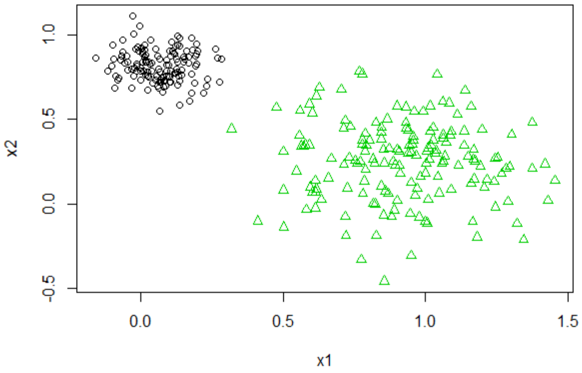
2nd Run k=2:

```
"Rows used as initial means"
259, 161
"Initial means"
[,1]                    [,2]
1.134466  0.4308556
1.319936 -0.1175686

Converged at 6
No of points in cluster 1 : 140             Mean of cluster 1 : 0.07438974, 0.81267206        SSE: 2.782616
No of points in cluster 2 : 160             Mean of cluster 2 : 0.9264137, 0.2476079          SSE: 16.295196
SSE: 19.077812  SSB: 78.044761 SSE+SSB: 97.122573
   Cluster id Cluster size Average silhouette width
         1          140              0.9073197
         2          160              0.7851322
Average silhouette coefficient of all clusters: 0.846226
```

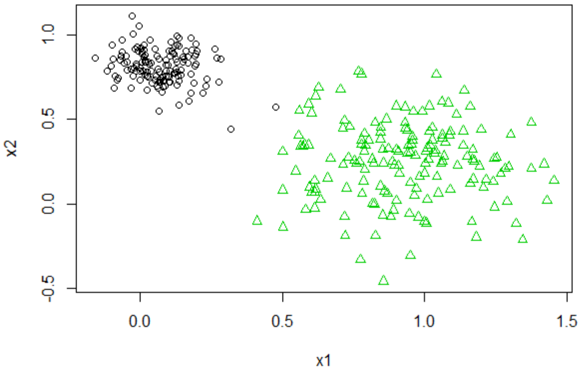
Confusion Matrix:

	1	2
1	138	0
2	2	160

TRUE CLUSTER SCATTER PLOT



OUTPUT OF KMEANS CLUSTERING



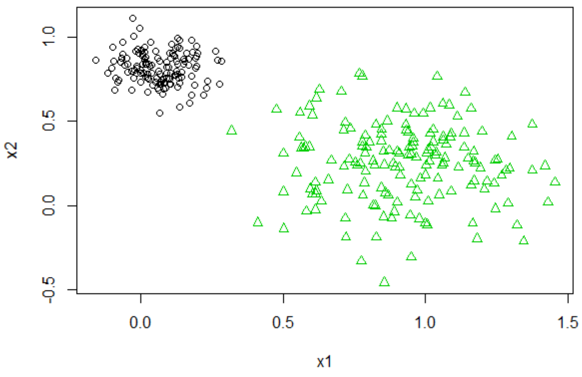
3rd run k=2:

```
"Rows used as initial means"
 164, 154
"Initial means"
        [,1]      [,2]
[1,] 0.6598657 0.1516759
[2,] 0.8549330 0.2418022

Converged at 5
No of points in cluster 1 : 140             Mean of cluster 1 : 0.07438974, 0.81267206        SSE: 2.782616
No of points in cluster 2 : 160             Mean of cluster 2 : 0.9264137, 0.2476079          SSE: 16.295196
SSE: 19.077812  SSB: 78.044761 SSE+SSB: 97.122573
   Cluster id Cluster size Average silhouette width
         1          140              0.9073197
         2          160              0.7851322
```

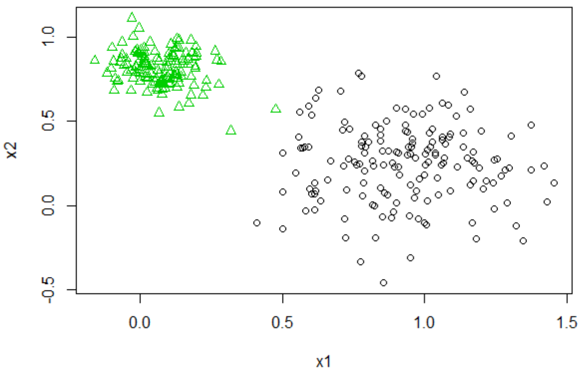
```
Average silhouette coefficient of all clusters: 0.846226
Confusion Matrix:
```

	1	2
1	138	0
2	2	160

TRUE CLUSTER SCATTER PLOT



OUTPUT OF KMEANS CLUSTERING



The initial choice of means affect the number of iterations needed to converge. When the initial centroids are chosen from two different clusters the algorithm converges easily. When initial centroids are chose from the same cluster it needs more iterations to converge. This affects the running time of program. When the dataset is really huge (A.easy has 300 rows only), this extra iterations can increase the running time of the algorithm greatly. When an outlier is chosen as initial centroid, an empty cluster (no datapoint except for the initial centroid) will be created. But since the A.easy dataset did not have any outlier and its relatively small, the choice of outliers did not have a significant effect on the clustering.

C) Run k-means 3 different times for hard dataset

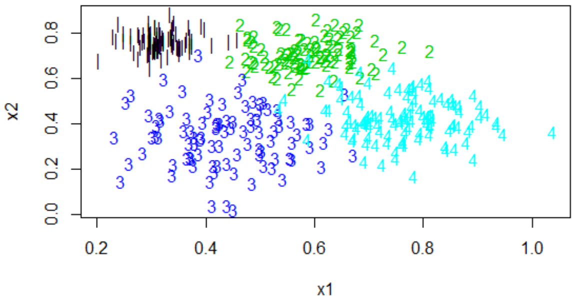
1st Run k=4:

```
"Rows used as initial means"
330, 71, 67, 305
"Initial means"
        [,1]      [,2]
[1,] 0.7724036 0.3810644
[2,] 0.2639380 0.7544361
[3,] 0.3064983 0.7496867
[4,] 0.8919679 0.2403554
Converged at 8
No of points in cluster 1 : 90        Mean of cluster 1 : 0.4410117, 0.3261311        SSE: 1.8446030
No of points in cluster 2 : 95        Mean of cluster 2 : 0.3222129, 0.7548252        SSE: 0.5004806
No of points in cluster 3 : 108       Mean of cluster 3 : 0.5958037, 0.6976751        SSE: 1.0764851
No of points in cluster 4 : 107       Mean of cluster 4 : 0.7728973, 0.4119890        SSE: 1.4705342
SSE: 4.892103            SSB: 24.412238 SSE+SSB: 29.304341
    Cluster id Cluster size Average silhouette width
[1,]       3          90            0.8535503
[2,]       1          95            0.9205114
[3,]       2         108             0.8799705
[4,]       4         107             0.8723423
Average silhouette coefficient of all clusters: 0.881594
Confusion Matrix:
```

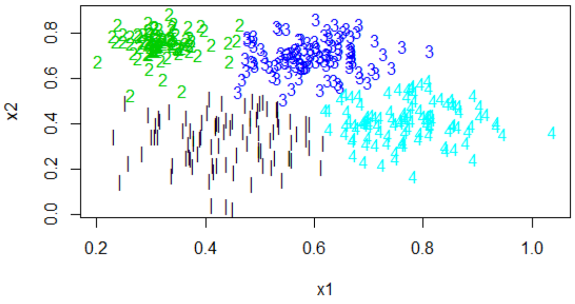
	1	2	3	4
1	89			
2		100		
3	6	1	90	
4		7		107

Note: The cluster ids in true cluster and k means output do not match exactly. So they are renamed based on their centers to match with true clusters for all plots.

TRUE CLUSTER SCATTER PLOT



OUTPUT OF KMEANS CLUSTERING



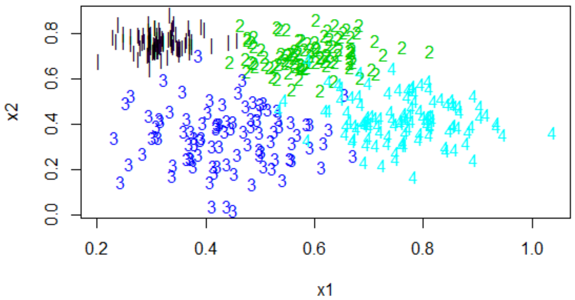
2nd run k=4:

```
"Rows used as initial means"
200, 144, 351, 29
"Initial means"
         [,1]      [,2]
[1,] 0.3872941 0.7041044
[2,] 0.5345032 0.6652981
[3,] 0.8886740 0.3888789
[4,] 0.3425450 0.7606140

Converged at 7
No of points in cluster 1 : 90        Mean of cluster 1 : 0.4410117, 0.3261311      SSE: 1.8446030
No of points in cluster 2 : 108       Mean of cluster 2 : 0.5958037, 0.6976751      SSE: 1.0764851
No of points in cluster 3 : 107       Mean of cluster 3 : 0.7728973, 0.4119890      SSE: 1.4705342
No of points in cluster 4 : 95        Mean of cluster 4 : 0.3222129, 0.7548252      SSE: 0.5004806
SSE: 4.892103          SSB: 24.412238 SSE+SSB: 29.304341
    Cluster id Cluster size Average silhouette width
[1,]         3          90            0.8509734
[2,]         2         108             0.8888619
[3,]         4         107             0.8723423
[4,]         1          95            0.9394694
Average silhouette coefficient of all clusters: 0.887912
```

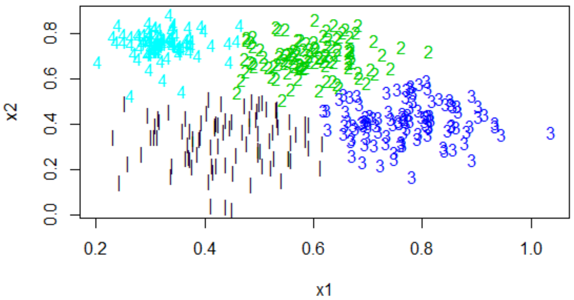
Confusion Matrix:

	1	2	3	4
1	89			
2		100		
3	6	1	90	
4		7		107

TRUE CLUSTER SCATTER PLOT



OUTPUT OF KMEANS CLUSTERING

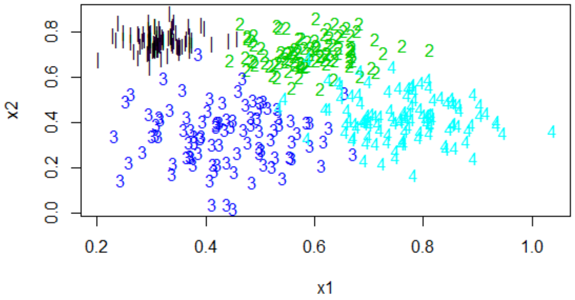


3rd run k=4:

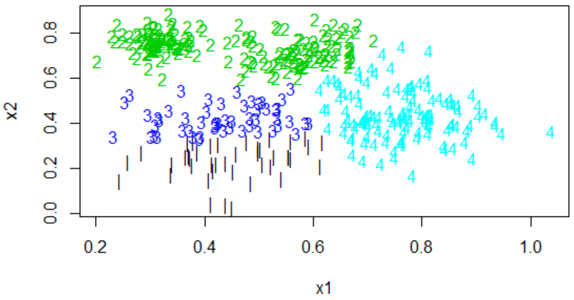
```
"Rows used as initial means"
275, 56, 254, 346
"Initial means"
        [,1]      [,2]
[1,] 0.4591563 0.2645070
[2,] 0.3081094 0.8253480
[3,] 0.4190315 0.3824918
[4,] 0.7998722 0.4349117
Converged at 6
No of points in cluster 1 : 42          Mean of cluster 1 : 0.4516136, 0.2297360        SSE: 0.5623145
No of points in cluster 2 : 184         Mean of cluster 2 : 0.4514733, 0.7370224        SSE: 4.4371226
No of points in cluster 3 : 53          Mean of cluster 3 : 0.4322255, 0.4221864        SSE: 0.6407472
No of points in cluster 4 : 121         Mean of cluster 4 : 0.7636470, 0.4368182        SSE: 2.0739982
SSE: 7.711999              SSB: 21.592342 SSE+SSB: 29.304341
    Cluster id Cluster size Average silhouette width
[1,]        2          42            0.8875240
[2,]        1         184             0.8453168
[3,]        3          53            0.8760006
[4,]        4         121             0.8518229
Average silhouette coefficient of all clusters: 0.865166
```

	1	2	3	4
1	89			
2	95			5
3		42	53	2
4				114

TRUE CLUSTER SCATTER PLOT



OUTPUT OF KMEANS CLUSTERING



Since A.Hard is has lesser density (datapoints are spread across and not concentrated in a particular region) of datapoints, the clusters are not very distinct and distinguishable. In such a cluster the choice of initial centroids can effect more. For example during the third run when two initial centroids were selected from same cluster (from true cluster 3), they converged in a different manner: the third cluster (in left plot above) was split into two; first and second clusters (in left plot) were condensed into a single cluster. When the clusters are very dense the initial choice of centroids doesn't have much significance but when the data points are not very dense, the initial centroids have a profound effect on convergence and output.

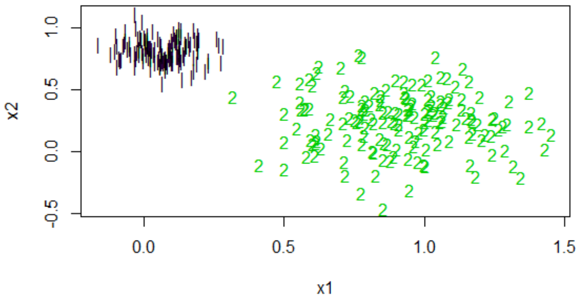
D) Run k-means with k=3 different times for easy and hard dataset

EASY DATASET:

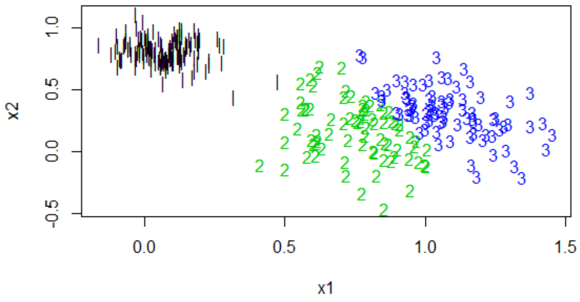
```
Enter value for k: 3
"Rows used as initial means"
255, 188, 263
"Initial means"
        [,1]        [,2]
[1,] 0.9308774  0.57360517
[2,] 0.5029749 -0.13854997
[3,] 1.2069475  0.09578408

Converged at 7
No of points in cluster 1 : 140        Mean of cluster 1 : 0.07438974, 0.81267206      SSE: 2.560486
No of points in cluster 2 : 77         Mean of cluster 2 : 0.7535905, 0.1701550       SSE: 4.605061
No of points in cluster 3 : 83         Mean of cluster 3 : 1.0867437, 0.3194619       SSE: 6.528024
SSE: 13.753963  SSB: 83.368610  SSE+SSB: 97.122573
    Cluster id Cluster size Average silhouette width
[1,]        1         140              0.9033617
[2,]        2          77             0.7378246
[3,]        3          83             0.7576133
Average silhouette coefficient of all clusters: 0.799600
Confusion Matrix:
```

	1	2	3
1	138		
2	2	77	83

TRUE CLUSTER SCATTER PLOT



OUTPUT OF KMEANS CLUSTERING



Changing the number of clusters to 3 has decreased the overall silhouette coefficient. This means that this is not a good cluster compared to 2-cluster design. It can also be seen from the graph that there is no visual distinction between cluster 2 and 3. Hence k=2 is preferred over k=3 for A.easy dataset.

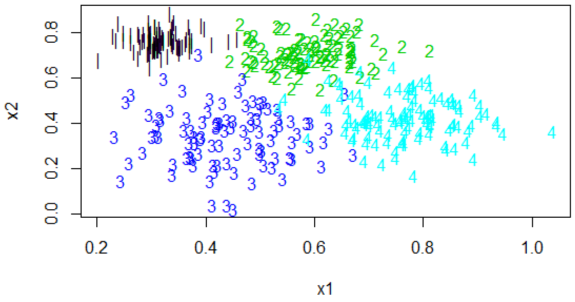
HARD DATASET:

```
"Rows used as initial means"
337, 269, 267
"Initial means"
         [,1]      [,2]
[1,] 0.6988988 0.5812629
[2,] 0.4996267 0.3627728
[3,] 0.4223245 0.2148383

Converged at 9
No of points in cluster 1 : 189          Mean of cluster 1 : 0.4547105, 0.7338377          SSE: 4.565794
No of points in cluster 2 : 119          Mean of cluster 2 : 0.7639561, 0.4320040          SSE: 2.039480
No of points in cluster 3 : 92           Mean of cluster 3 : 0.4401851, 0.3303058          SSE: 1.959401
SSE: 8.564675          SSB: 20.739666 SSE+SSB: 29.304341
    Cluster id Cluster size Average silhouette width
[1,]       1        189             0.8543567
[2,]       2        119             0.8560416
[3,]       3         92            0.8414629
Average silhouette coefficient of all clusters: 0.850620
```

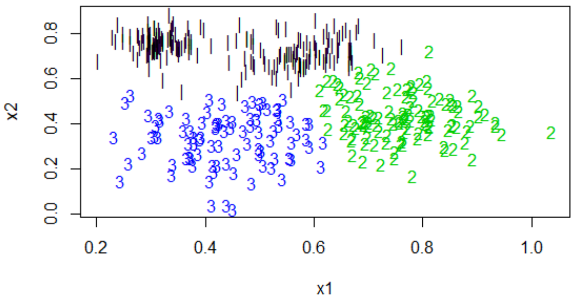
Confusion Matrix:

	1	2	3
1	89		
2	100		
3		5	92
4		114	

TRUE CLUSTER SCATTER PLOT



OUTPUT OF KMEANS CLUSTERING



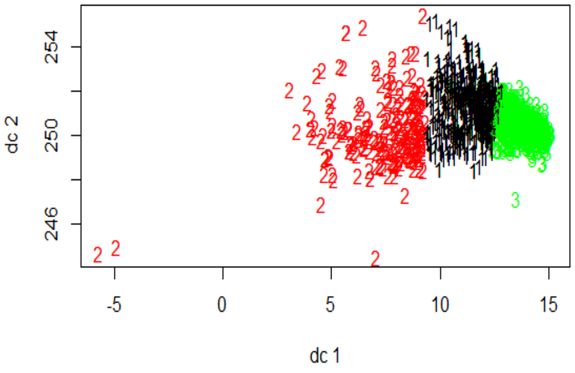
For the hard dataset, specifyng k=4, has merged two true clusters (1 and 2) into a single cluster. The SSE for this new cluster is higher and it has lowered the silhouette coefficient too. So it is not preferred. K=4 for the A.hard dataset is the optimum number of clusters. This is supported by the elbow graph too. (Fig.2)

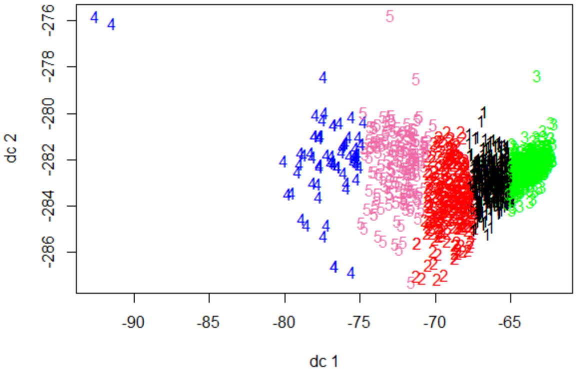
WINE DATASET:

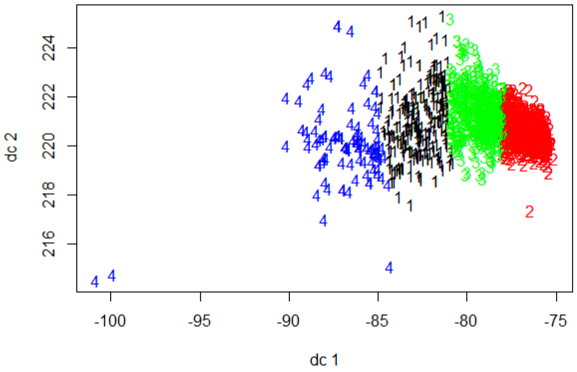
For wine dataset, k=2,3,4,5 and 6 were tested. With the knowledge of wine dataset from previous analysis these values of k were chosen. K=2 because the class attribute had two classes high and low. K=6 because the attribute quality had six distinct values (Quality – 3,4,5,6,7,8). K=3 because values of quality were paired ((3 and 4), (5 and 6), (7 and 8)). K=4 because cluster with quality = 5 and 6 had more than 1000 datapoints and others had very less and (5 and 6) cluster was split into 2 (Quality = (3 and 4), 5, 6, (7 and 8)).

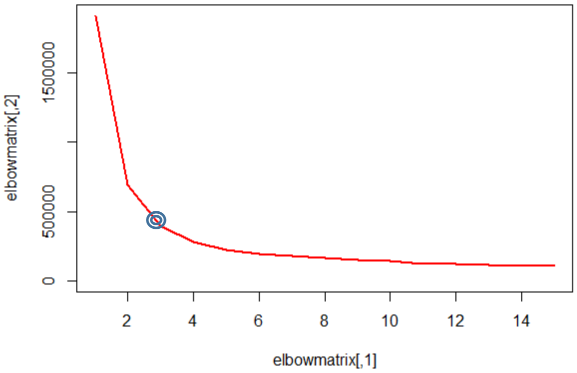
K=2 had Average silhouette coefficient of all clusters: 0.516629 and Overall SSE: 691909
K=3 had Average silhouette coefficient of all clusters: 0.453164 and Overall SSE: 395285
K=4 had Average silhouette coefficient of all clusters: 0.427078 and Overall SSE: 283160
K=6 had Average silhouette coefficient of all clusters: 0.400540 and Overall SSE: 193445

K=6 is not preferred because its silhouette coefficient is very low. K=4 also has smaller sil.coefficient compared to K=3 and the SSE value hasn't reduced much (relatively). So k=4 and k=6 are not preferred. Between k=2 and k=3, it can be observed that though k=2 has better sil.coeff value, its SSE is almost double of SSE of k=3. SSE should be less for good clusters. So k=3 is preferred. This decision is backed up by the elbow graph, a sharp decrease in the SSE value at k=3 and the curve slides down gradually from there. This is the breaking point and the value of k corresponding to this breaking point is 3. Thus the optimal number of clusters is chosen as 3 for wine dataset.
Note: Since the wine dataset is 11 dimensional, the clustering cannot be visualized and hence a principal component analysis is done on these 11 attributes and dc1 and dc2 are the two final components from PCA. This helps in visualizing the clusters.









	Q = 3	Q = 4	Q = 5	Q = 6	Q = 7	Q = 8	Size of each cluster
Cid = 1	0	5	175	45	14	2	214
Cid = 2	3	15	224	247	54	3	546
Cid = 3	7	33	282	346	131	13	812
Total	10	53	681	638	199	18	1600

The quality attribute is not of much use in validating the clusters mainly because it is very dense around Q=5 and Q=6, so any cluster will have a significant number of Q=5 and Q=6 datapoints.

PART – 2

DBSCAN and Hierarchical clustering were used on the same dataset. R has built-in functions to perform DBSCAN and Hierarchical clustering. It can also plot the clusters on a graph.

DBSCAN:

It is a density based approach. It classifies points into core, border and noise points depending on the number of neighbors each point has in within a given distance epsilon. The number of neighbors and the distance epsilon are the two parameters that can be tweaked. Number of neighbors also called as MinPts. A point is a core point if it has at least MinPts members within a circle of radius epsilon. A point is a border point if it is not core point but lies in the radius of some core point. It becomes a noise point if it not a core point of its own and doesn't lie within any core points radius (radius=Epsilon).

For both A.easy and A.hard, DBSCAN with MinPts = 5 and Epsilon = 0.2 gave the exact same results as true clusters. Increasing the number of neighbors (MinPts) generates more noise points. Increasing epsilon combines clusters (decreases number of clusters) and decreasing epsilon breaks clusters (increases number of clusters). The number of clusters were as high as 300 when epsilon was 0 and MinPts was 1. Essentially this setting treats every single data point as a new cluster. To the contrary when Epsilon was 2 and MinPts was 1, the whole dataset condensed into a single cluster. The tables below represents different runs of DBSCAN on A.easy (left) and A.hard (right) datasets and the plots below the tables represent the plots of DBSCAN for easy hard and wine datset respectively.

Each cell in table represents = (No. of cluster, No. of Noise points)

From Left to Right: Noise points increase; From Top to bottom: Number of clusters decrease

MinPts ---------- Epsilon	1	**5**	10	20	50
0	300,0	0,300	0,300	0,300	0,300
0.05	74,0	7,168	2,189	3,264	0,300
0.1	20,0	3,31	3,71	2,137	1,196
0.2	3,0	**2,1**	2,5	2,15	2,67
0.5	2,0	2,0	2,0	2,0	2,0
2	1,0	1,0	1,0	1,0	1,0

MinPts ---------- Epsilon	1	**5**	10	20	50
0	400,0	0,400	0,400	0,400	0,400
0.05	36,0	4,52	5,139	3,229	1,339
0.1	7,0	4,5	4,7	4,21	4,146
0.2	4,0	**4,0**	4,0	4,0	4,0
0.5	4,0	4,0	4,0	4,0	4,0
2	1,0	1,0	1,0	1,0	1,0

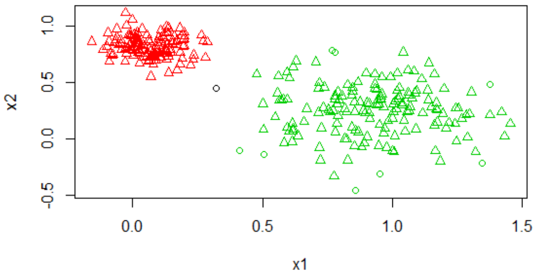
Confusion Matrices:

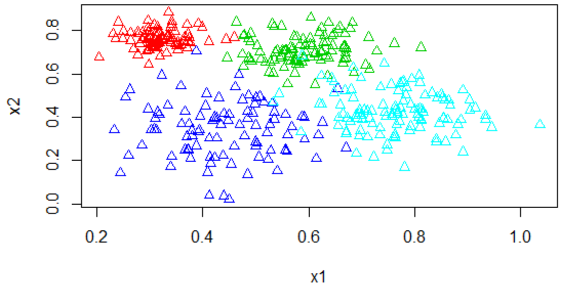
	1	2
1	138	0
2	0	161

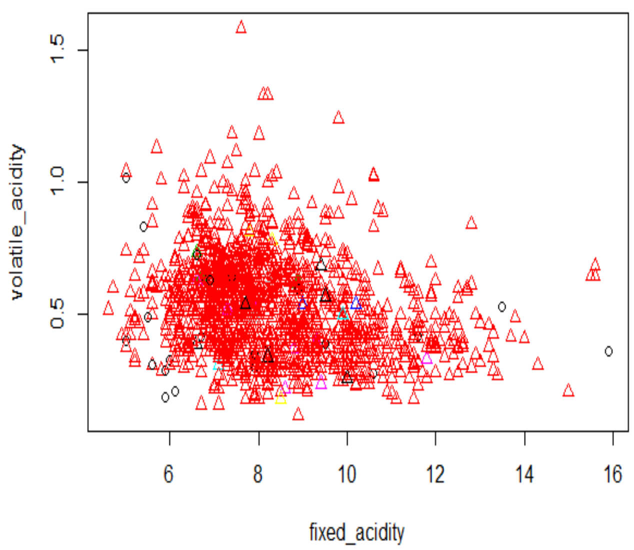
1 Noise point in A.easy

	1	2	3	4
1	89			
2		100		
3			90	
4				107

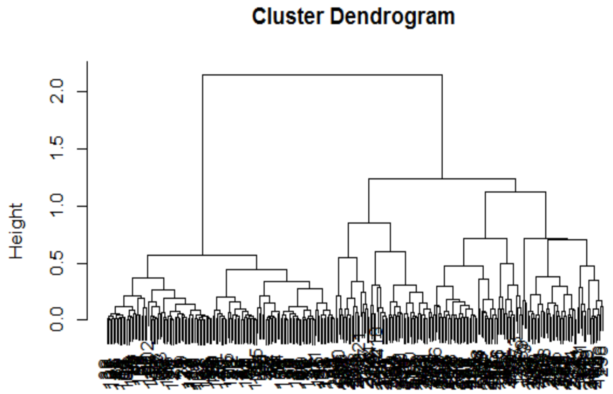
	A.Hard	A.Easy	Wine
No. of Clusters	4	2	2
Overall SSE	5.55	19	33000
Avg.Sil.	0.87	0.85	0.2
Noise	0	1	26







The most challenging part in DBSCAN is to find the optimum value of Epsilon. A small variation in epsilon can break clusters into many small ones or they become noise points if MinPts is very high. A large number of permutations of epsilon and minpts should be tried and the resulting clustering should be evaluated to find the perfect eps and minpts value. DBSCAN doesn't work well for high dimensional data. This problem is serious in multi-dimensional datasets like wine. The value of epsilon in a non-normalized dataset can be very high. So wine dataset was normalized for DBSCAN clustering. In most iterations there was one huge cluster which has more than 80% datapoints and the other clusters were very small (less than 2% of datapoints).



DBSCAN clustering is as good as k means with respect to A.easy and A.hard datastes. But so many iterations were performed to arrive at this optimum epsilon and minpts value. Though the final results are same due to this overhead in computation k means is better. With respect to wine dataset DBSCAN performs very worse since it is a multi-dimensional dataset. This is backed up by high SSE and very low silhouette coefficient values of DBSCAN cluster output.

Hierarchical clustering is not preferred as the output would be a dendogram (left figure), which becomes messy when the number of datapoints increases, making it hard to comprehend. Also in real life datasets there is no hierarchical relationships between datapoints. Rather they are distinct and tend to favor K-means or DBscan clustering. In k means clustering we have to specify the number of clusters initially. Number of clusters can be chosen by visualizing the data. For higher dimensions Principal Component Analysis can be done and the number of clusters can be chosen by plotting dc1 and dc2 (the two principal components). Another problem with K means clustering is choice of initial centroids. Poor choice of centroids can increase the number of iterations to converge or even produce empty clusters (when an outlier is chosen as initial centroid). These can be avoided by normalizing dataset, choosing one point from each distinct visual clusters while choosing the number of clusters. With the right initial centroids and correct number of clusters, K means converge very fast. A disadvantage with k means is that it always tends to produce globular clusters and is not very well suited for clusters of varying sizes and density. DBSCAN on the other hand is most suited for non-globular clusters and is resistant to noise unlike k means clustering. But it does not perform well with high dimensional data and determining optimal Epsilon and MinPts can also become very costly, sometimes impossible too.

Though kmeans has some issues, they can be avoided by techniques like choosing more clusters and discarding/combing them at the end and choosing optimal initial centroids (by methods discussed above). Thus K means is my preferred clustering method.