1. **Implement the hierarchical agglomerative clustering with the following linkage: single, complete, average and centroid. You can use built-in R functions to visualise your results.**

The hierarchical agglomerative clustering was implemented in R. A dataset needs to passed to in distance() first, which calculates the Euclidean distance for nrows() of datapoints and this output needs to passed to hierarchical.agglomerative() which returns a hclust object.

2. **Apply your program to the NCI microarray data set. This dataset has 64 columns and 6830 rows, where each column is an observation (a cell line) and each row represents a feature (a gene). (Therefore, the data set is represented via its transposed data matrix.) Pre-process the data set as appropriate.**

Applying the hierarchical.agglomerative() to NCI microarray dataset with labels. This implementation algorithm is applied for different linkage methods as “single”, “complete” and “average” in the function call. These results are visualised with the help of a dendrogram.

3. **Discuss the performance of hierarchical agglomerative clustering when using different linkage functions.**

As stated in task 2, hierarchical.agglomerative() was applied on NCI microarray dataset using single, complete and average clustering linkage functions. And the following are the dendrograms for the single, complete and average clustering linkage respectively.

Single Linkage : The ordering of leaf nodes from left to right:

```
 [1] 41 10 37 36 35 40 39 55 18  5 20 19 56 58 57 63 59 64 60 62 61  8  7  6 25

[26] 54 26  4 23 28 47 22 21 52 49 51 50 48 53 43 27 29 24 33 32 31 30 45 46 44

[51] 42  9 11 17 14 13 12 16 15  3  2  1 38 34
```

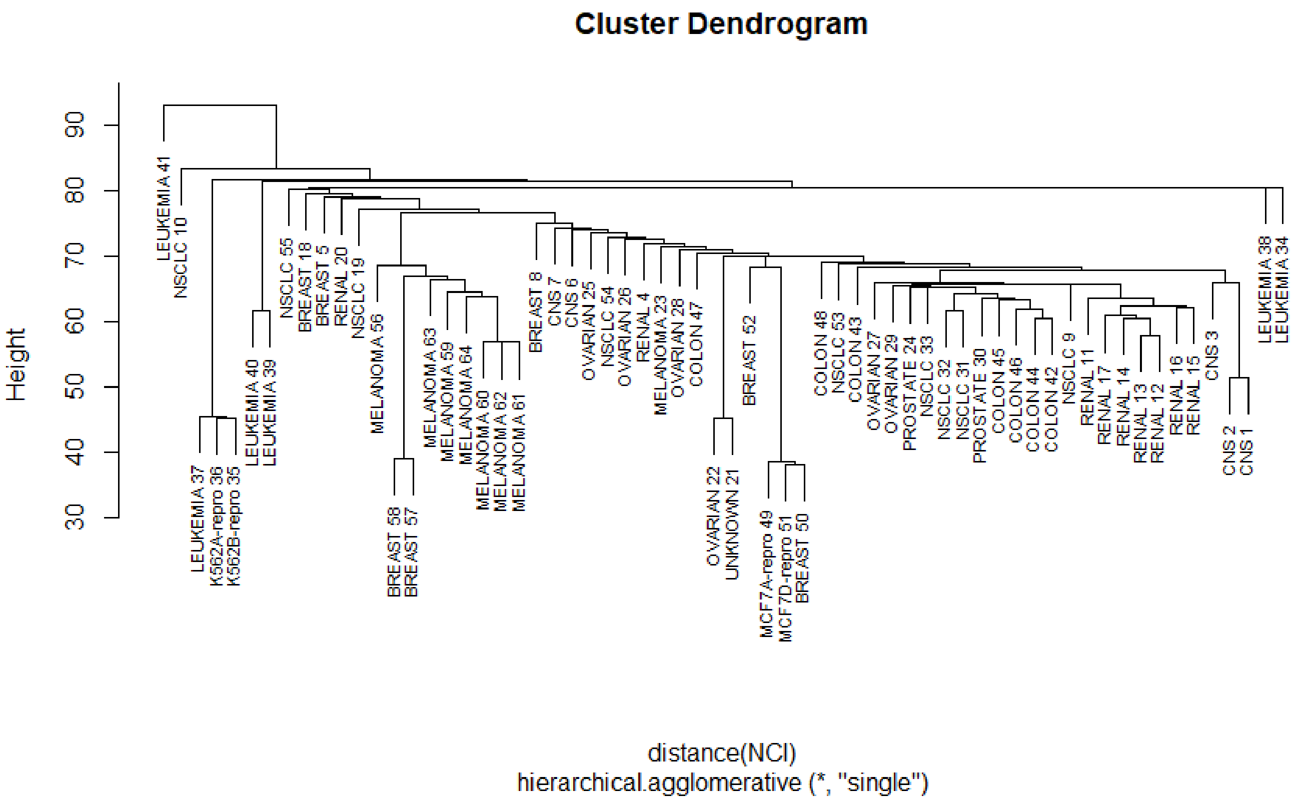
Complete Linkage : The ordering of leaf nodes from left to right:

```
[1] 52 49 51 50 43 44 42 46 45 48 47 40 39 41 37 36 35 38 34 10  5  4 55 54 56

[26] 59 60 62 61 58 57 64 63 20 22 21 19 18  2  1  7  8  6 26 25 11 13 12 17 14

[51] 16 15 28 27 53 32 31 33 23  9  3 24 30 29
```

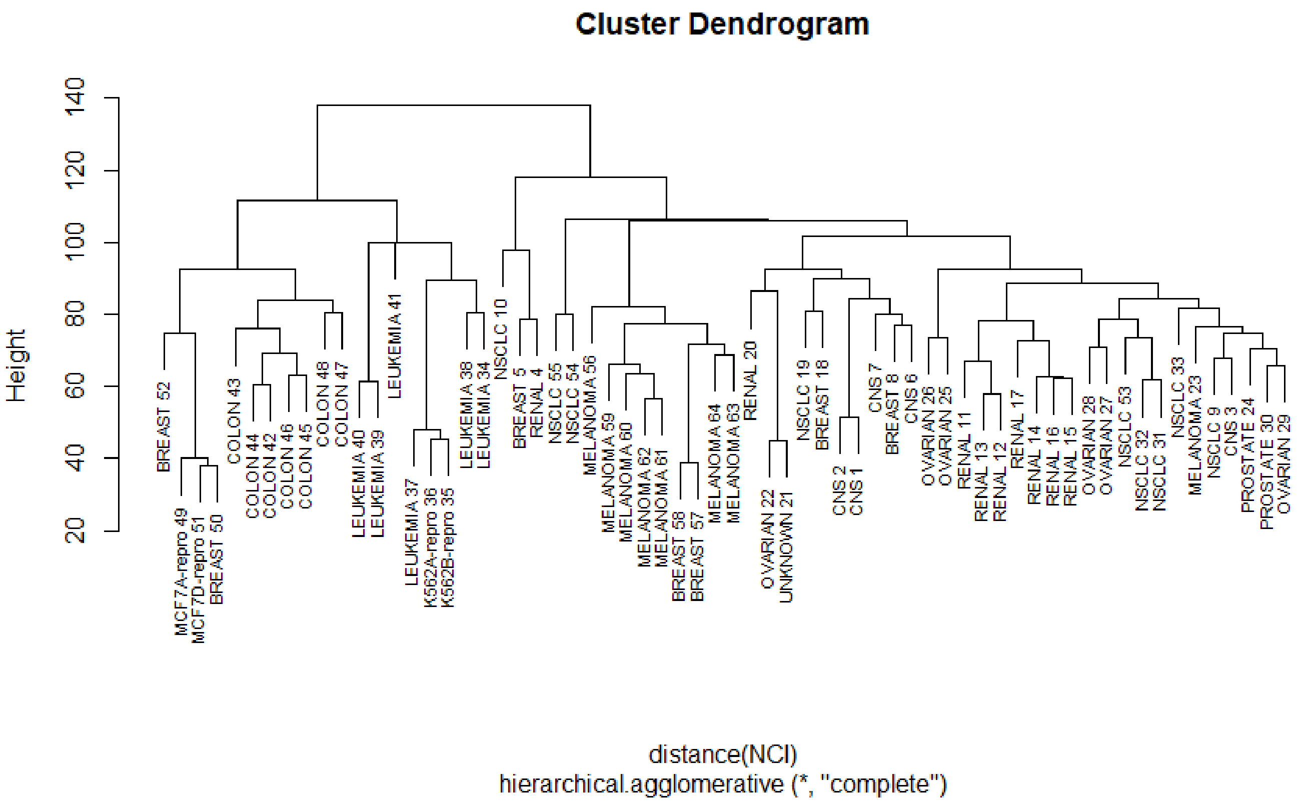
Average Linkage : The ordering of leaf nodes from left to right:

```
[1] 41 40 39 37 36 35 38 34 10 20 19 18 26 25 17 11 13 12 14 16 15 33 32 31 53

[26] 24 30 29 28 27 22 21 23  9  7  8  6  5  4  3  2  1 56 63 58 57 59 64 60 62

[51] 61 55 54 52 49 51 50 47 48 43 44 42 46 45
```

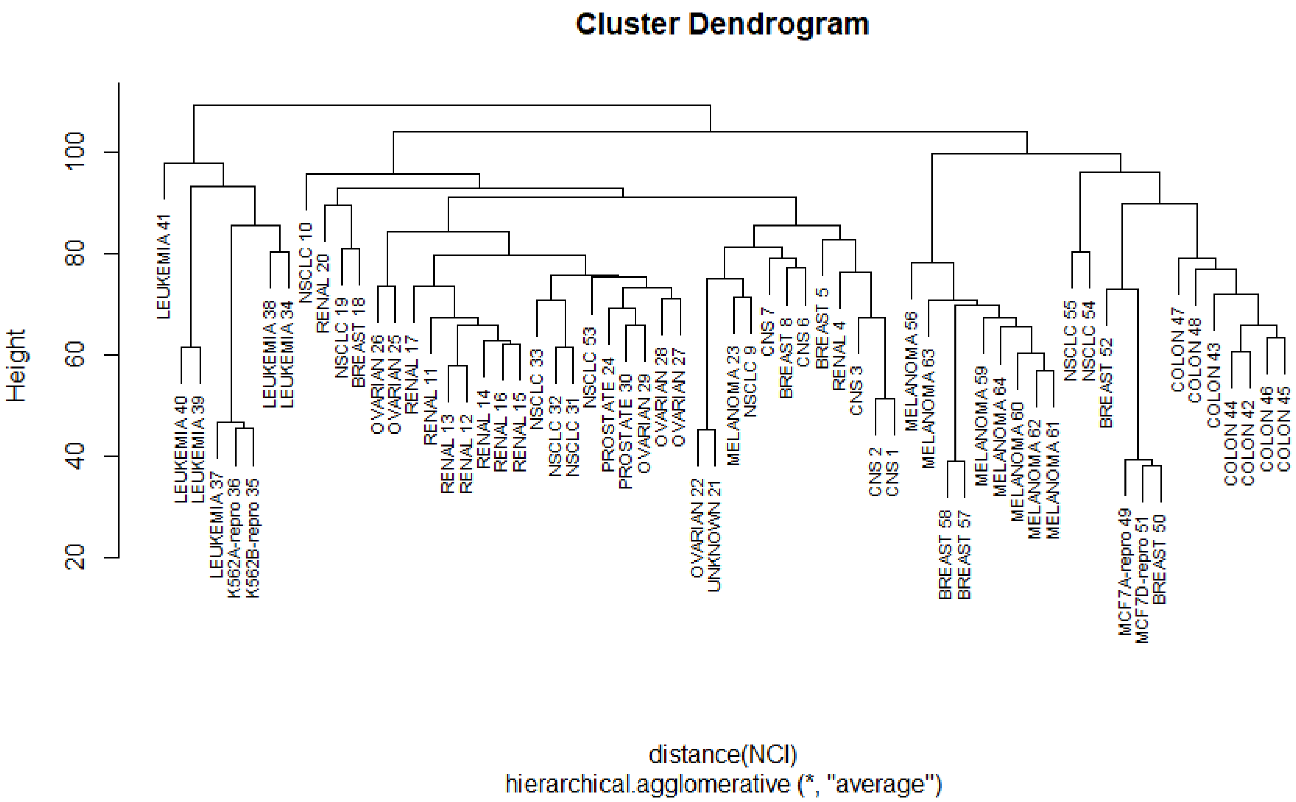
1 Single linkage function Clustering



2 Complete linkage function Clustering



3 Average linkage function Clustering



The difference between the three-linkage function is that single linkage is used to merge the two clusters who have member with smallest distance (min), whereas complete linkage is used to merge the two cluster who have the largest distance (max), when compared to average linkage, it is used to merge the two clusters.

The single linkage function makes more nested branches, a complete linkage function makes branches closer and tighter to each other, whereas the average linkage function has an even spread of the branches.

4. Apply the R function kmeans() to the above NCI microarray data set with different K and discuss its performance.

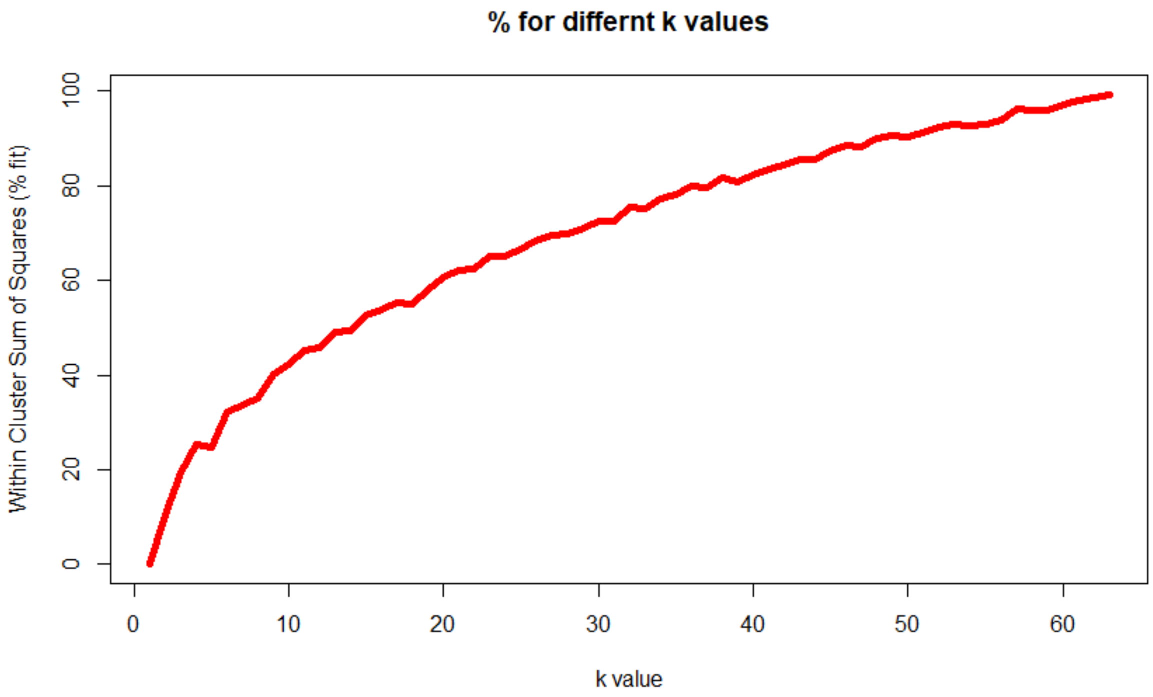
Applying the kmeans() to NCI microarray data set with k = 3, k = 7 and k = 14 (the original number of clusters in the dataset). The below are the cluster assignments for the datapoints.

Output:

```
[1] "For K ="

[1] 3

V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19
```

```
3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
V20 V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38
  3   3   3   3   2   2   2   2   3   2   3   3   3   3   1   1   1   1   1
V39 V40 V41 V42 V43 V44 V45 V46 V47 V48 V49 V50 V51 V52 V53 V54 V55 V56 V57
  1   1   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   3   3
V58 V59 V60 V61 V62 V63 V64
  3   3   3   3   3   3   3

[1] "For K ="
[1] 7
 V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19
  3   3   3   3   3   3   1   3   3   3   3   3   3   3   3   3   3   1   1
V20 V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38
  1   3   3   3   6   6   6   6   6   6   6   6   6   6   4   4   4   4   4
V39 V40 V41 V42 V43 V44 V45 V46 V47 V48 V49 V50 V51 V52 V53 V54 V55 V56 V57
  4   4   4   6   5   5   5   5   5   5   5   5   5   5   6   6   6   2   2
V58 V59 V60 V61 V62 V63 V64
  2   7   7   7   7   7   7

[1] "For K ="
[1] 14
 V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19
  6   6   6  13  13   2   2   2   6   2   9   9   9   9   9   9   9   7   7
V20 V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38
  7   3   3   6   1   1   1   1   1   1   1   1   1   1   4   5   5   5   4
V39 V40 V41 V42 V43 V44 V45 V46 V47 V48 V49 V50 V51 V52 V53 V54 V55 V56 V57
 12  12   4   1   8   8   8   8   8   8  10  10  10  10   1  11  11  14  14
V58 V59 V60 V61 V62 V63 V64
 14  14  14  14  14  14  14
```

With respect to the performance of the k-means with different values of k, below is a plot showing how the within cluster sum of square’s percentage fit i.e., % fit which is (between_SS/total_SS)*100 showing how the does the datapoints inside the cluster fit with each value of k.



We can observe that ratio between_SS/total_SS for k = 14 (from the original dataset) is around 0.49, but as k-means is an unsupervised clustering algorithm we did not pass the labels in order to compare the clusters. The above plot indicates that as the k value is near to the number of datapoints the ratio is near 1, for k = number of data points this ratio will be one as each datapoint is its own cluster. Our aim is to select an optimal value for k based on the application.

5. Compare and contrast the performance of K-means and hierarchical agglomerative clustering.

Comparing K-means and Hierarchical agglomerative clustering based on the NCI microarray dataset:

1. For k-means clustering we always need to specify a value k which will be the deciding factor for the number of clusters.
2. For hierarchical clustering we do not require any k values for number of clusters, as initially each of the datapoint is its own cluster and we merge the datapoints based on the linkage methods and form one big cluster with different branches and each datapoints as the leaves.
3. The output of a k-means clustering shows the more quantitative statistics about the dataset, for example the percentage fit as within cluster sum of squares, to show the quality of fitting.

4. The output of a hierarchical clustering shows the visualisation of a dendrogram, which shows the tree structure or hierarchy of clustering which is very appealing and better to understand.
5. The most basic of all, hierarchical clustering is not good for datasets with large number of datapoints as the hierarchy or tree becomes very difficult to understand and sometimes meaningless to plot them.
6. Whereas, k-means is most suitable for large number of datapoints and if the dataset is in two or three dimensions, then it even better as we can visualise it.
7. With respect to the program execution, k-means is faster than hierarchical agglomerative clustering.

6. Discuss how to choose the number of clusters in the K-means and hierarchical agglomerative clustering.

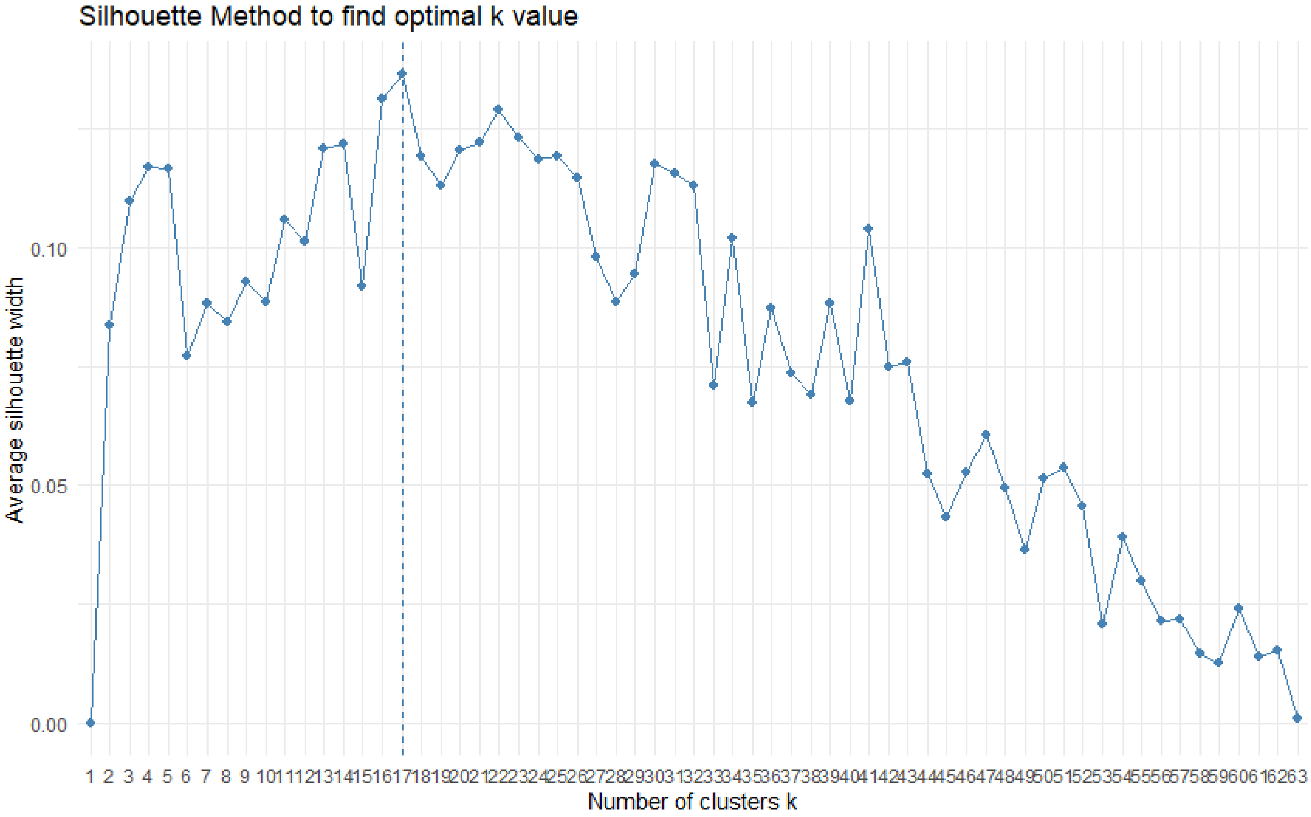
However, there are 14 clusters or 14 unique labels in the NCI dataset. There are different methods to choose the number of clusters in the K-means and Hierarchical agglomerative clustering.

- How to choose the number of clusters in K-means clustering:

There are two methods using which we can choose the optimal number of clusters ‘K’:

1. Elbow method: in this method the within cluster sum of squared errors is calculated for different values of K, and the optimal K is choosing for which the within cluster sum of squared errors first starts to decrease. Here, within cluster sum of squared errors means $betweenss/$totss ratio for kmeans() in R.
2. Silhouette method: in this method the optimal value of K is choosing the maximum silhouette value, which is the measure of how similar a point is to its own cluster (inter cluster distance) compared to other clusters (distance of clusters or separation).

From the above the plot, we can observe that for NCI microarray dataset, the optimal value for k is 17 clusters using Silhouette method.



- How to choose the number of clusters in Hierarchical agglomerative clustering:

Hierarchical clustering is better understood using a dendrogram tree diagram and any decision to choose the number of clusters will depend of this output. One of the methods is to look for the clusters with the longest (height) branches, the shorter the height the more similar they are to their following leaves and sometimes it depends on the application needs.