Kunal Mehta
kunalkam
50352960

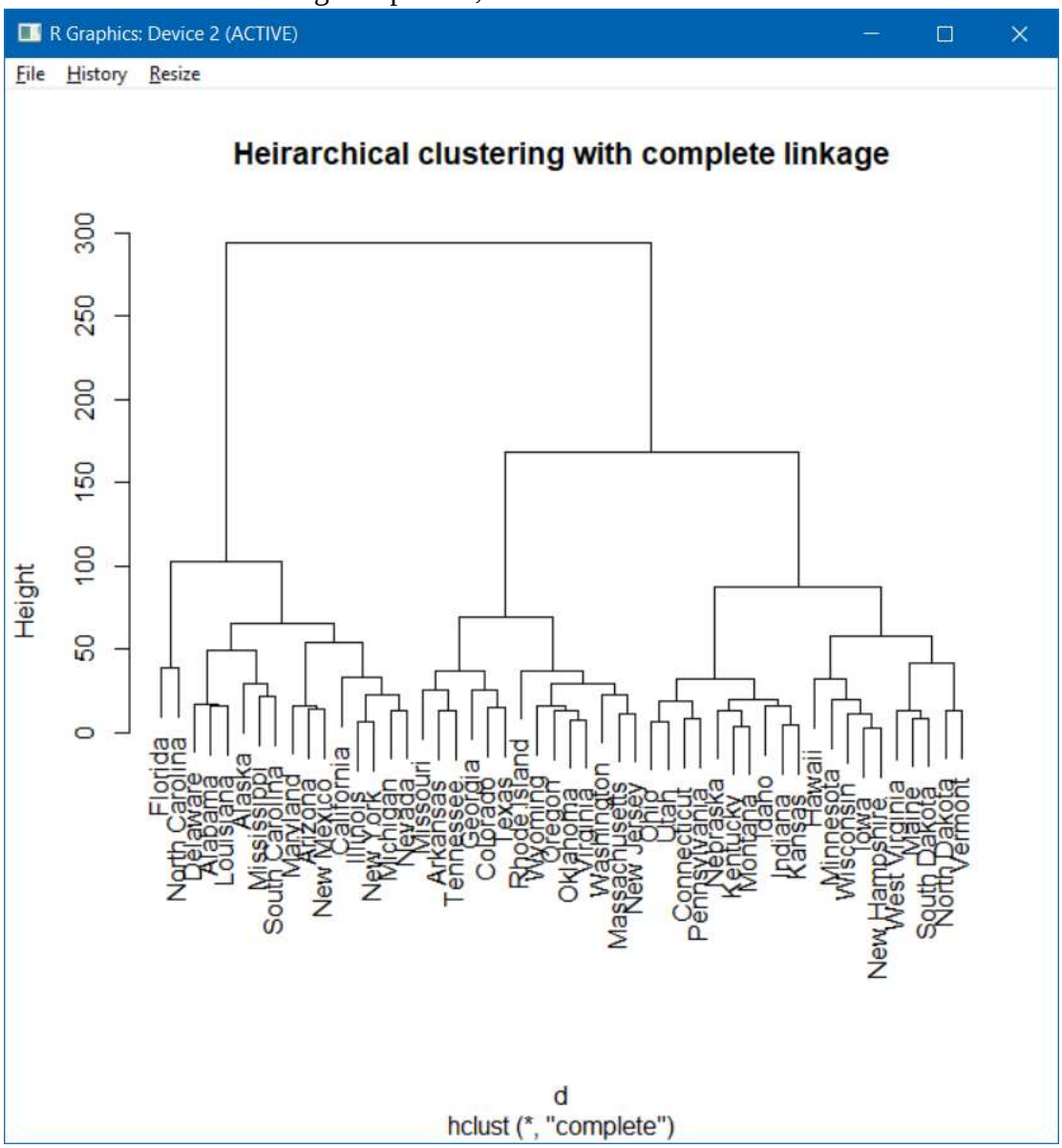
Homework 2

Textbook Problems:

Problem 9

a. First step, I imported the USArrests data, fairly straightforward. As per the question, I was to perform hierarchical clustering with complete linkage on this data using Euclidean distance. Therefore, I used the dist() function to calculate the Euclidean distances of the dataset and stored it in the variable d. Then, I used the hclust function, passing variable d as the distances argument, with method = complete to have the function implement complete linkage.

The result of the clustering was plotted, and the screenshot of the same is shown below.

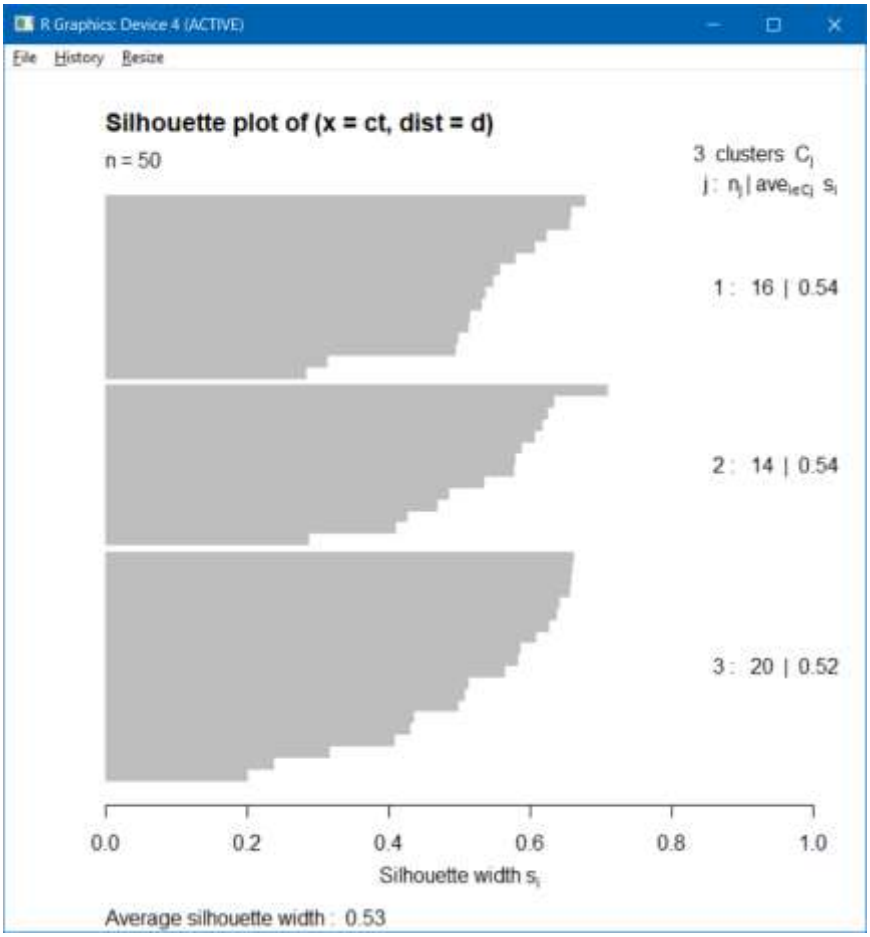


b. As you can see from the image of the dendrogram, the safe approximate range of heights where the dendrogram could be cut to yield 3 distinct clusters would be around 110 – 160, but since we have been told explicitly to cut it such that there are 3 clusters, it's

much safer to use the k = 3 argument of the cutree() function instead of choosing a height based on visual observation and passing it as the h argument. Therefore I cut the dendrogram and store the values in variable ct, which I print next. The output indicates what cluster each of the states is in.

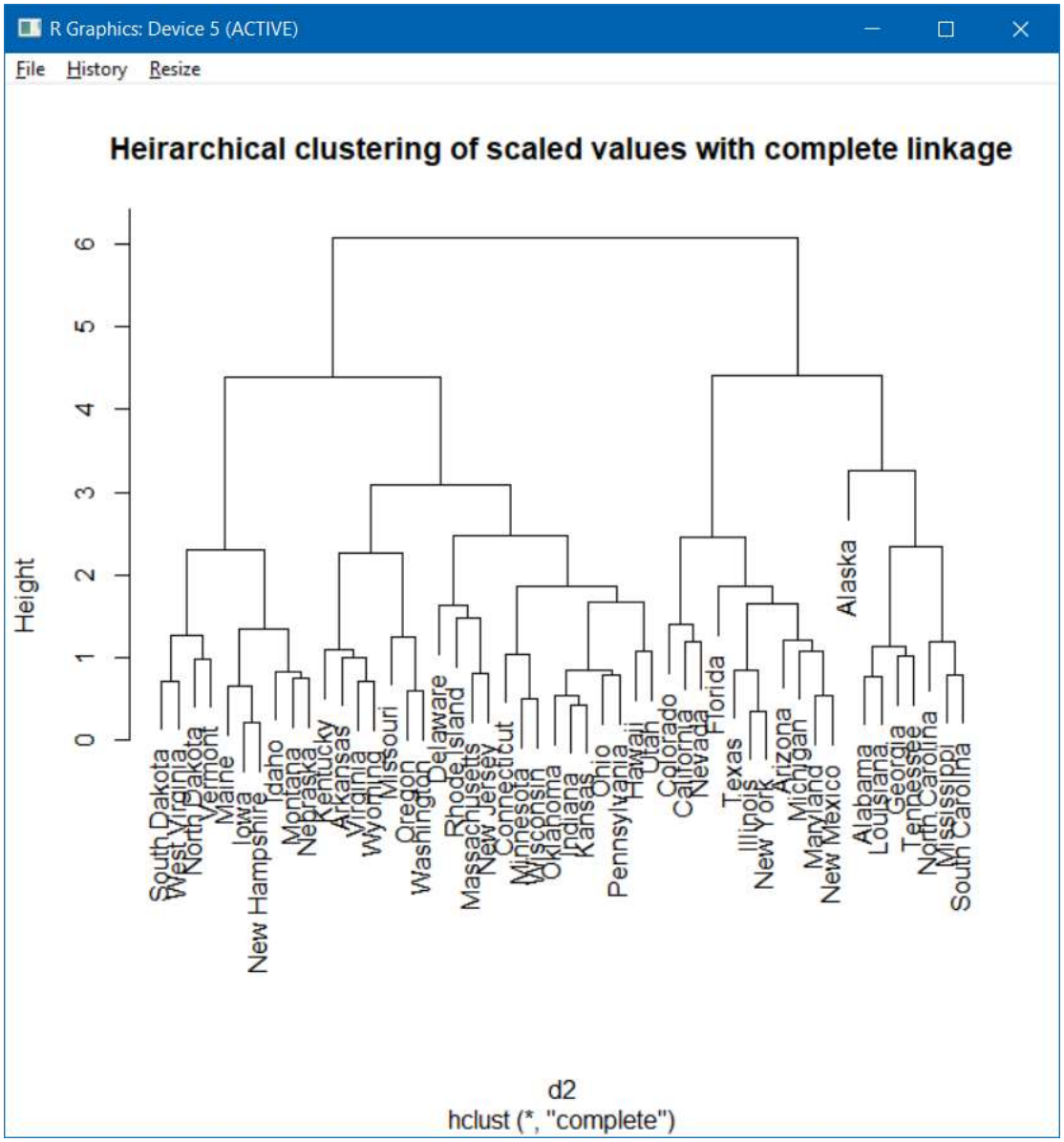
```
     Alabama      Alaska     Arizona    Arkansas
           1           1           1           2
  California    Colorado Connecticut    Delaware
           1           2           3           1
     Florida     Georgia      Hawaii       Idaho
           1           2           3           3
    Illinois     Indiana        Iowa      Kansas
           1           3           3           3
    Kentucky   Louisiana       Maine    Maryland
           3           1           3           1
Massachusetts    Michigan   Minnesota Mississippi
           2           1           3           1
    Missouri     Montana    Nebraska      Nevada
           2           3           3           1
New Hampshire  New Jersey  New Mexico    New York
           3           2           1           1
North Carolina North Dakota        Ohio    Oklahoma
           1           3           3           2
      Oregon Pennsylvania Rhode Island South Carolina
           2           3           2           1
  South Dakota   Tennessee       Texas        Utah
           3           2           2           3
     Vermont    Virginia  Washington West Virginia
           3           2           2           3
   Wisconsin     Wyoming
           3           2
> |
```

I also tried plotting the silhouette plot of this, just to see the average silhouette width. The value I observed was 0.53



Kunal Mehta
kunalkam
50352960

c. In this case, we had to scale the variables first to have standard deviation one, and then hierarchically cluster them with complete linkage like we did in (a). Therefore, using the scale() function, it was pretty easy to scale the data. Next, I calculated the Euclidean distances and stored it in d2, followed by hierarchical clustering with the hclust() function. The plot of this clustering is as shown below.



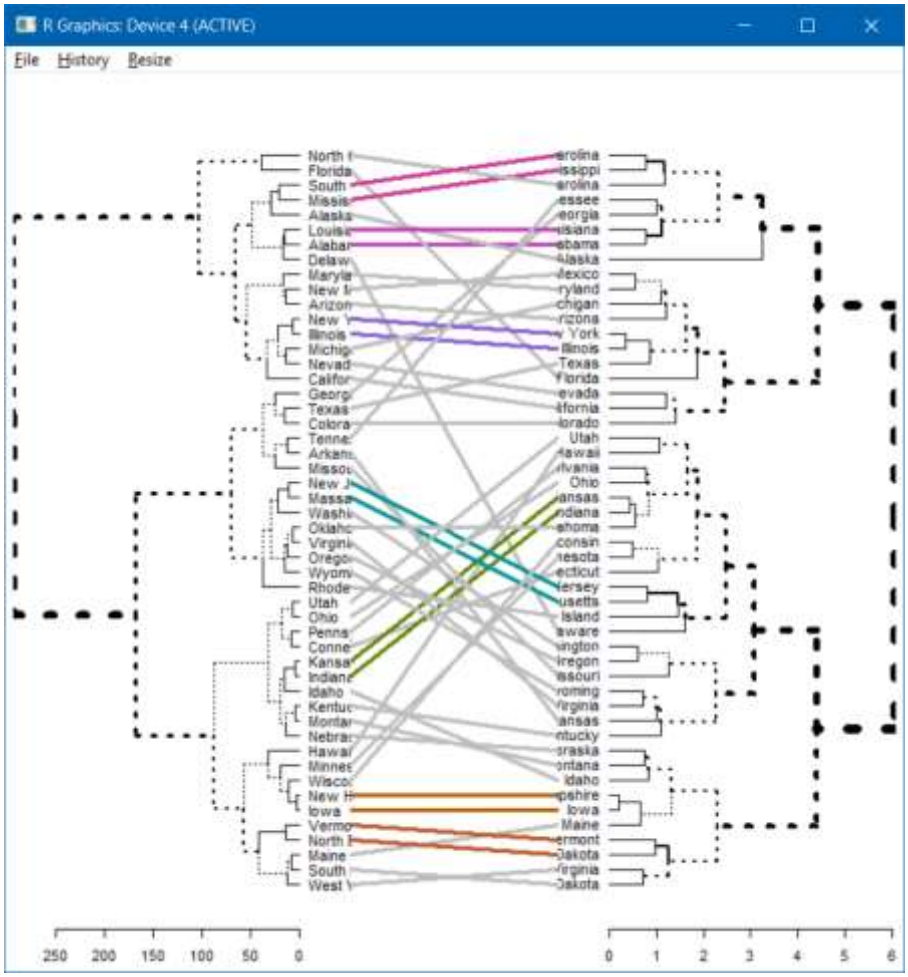
Kunal Mehta
kunalkam
50352960

d. To observe the effects of scaling, I cut this dendrogram such that it would yield 3 clusters, exactly like I did last time. The results are as shown below.

```
   Alabama     Alaska    Arizona   Arkansas
         1          1          2          3
California   Colorado Connecticut   Delaware
         2          2          3          3
   Florida    Georgia     Hawaii      Idaho
         2          1          3          3
  Illinois    Indiana       Iowa     Kansas
         2          3          3          3
  Kentucky  Louisiana      Maine   Maryland
         3          1          3          2
Massachusetts  Michigan  Minnesota Mississippi
         3          2          3          1
  Missouri    Montana   Nebraska     Nevada
         3          3          3          2
New Hampshire New Jersey New Mexico   New York
         3          3          2          2
North Carolina North Dakota       Ohio   Oklahoma
         1          3          3          3
    Oregon Pennsylvania Rhode Island South Carolina
         3          3          3          1
 South Dakota  Tennessee      Texas       Utah
         3          1          2          3
   Vermont   Virginia Washington West Virginia
         3          3          3          3
 Wisconsin    Wyoming
         3          3
> |
```

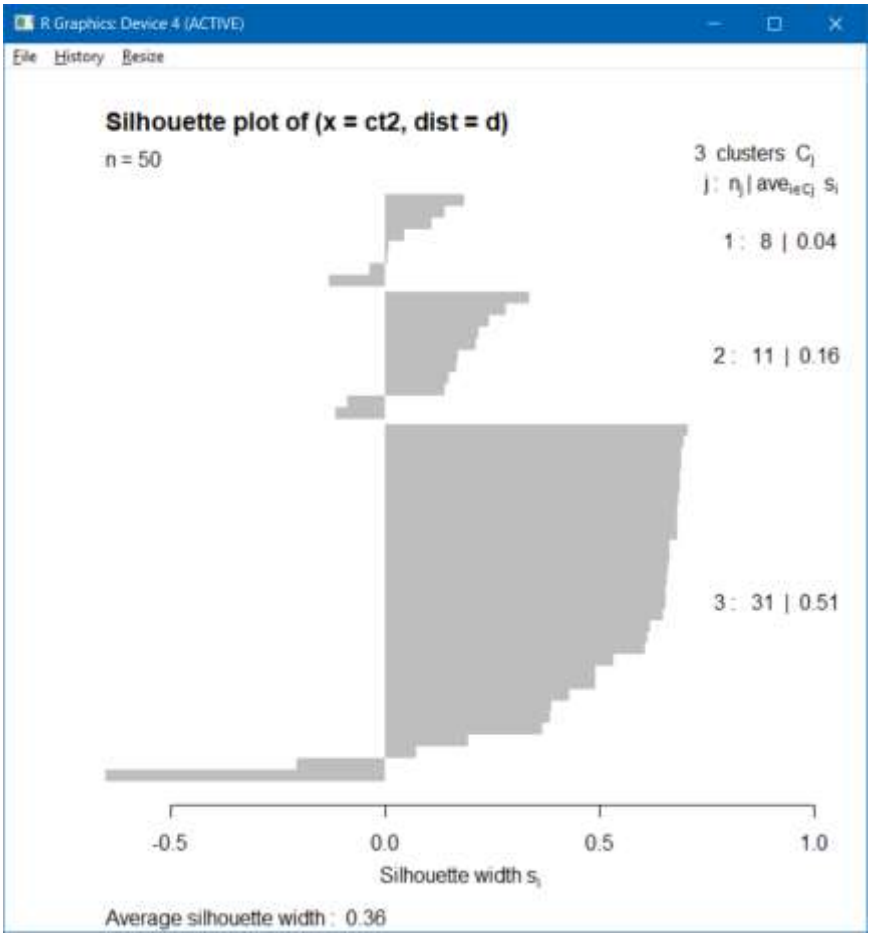
Even a cursory glance shows that the results differ markedly from those observed before. To clearly compare the two, I looked up a tutorial on Datanovia (cited below) and created a tanglegram for them.

Datanovia link: https://www.datanovia.com/en/lessons/comparing-cluster-dendrograms-in-r/



Kunal Mehta
kunalkam
50352960

Like for the previous case, I also did a silhouette plot, which gave me an average silhouette width of 0.36.



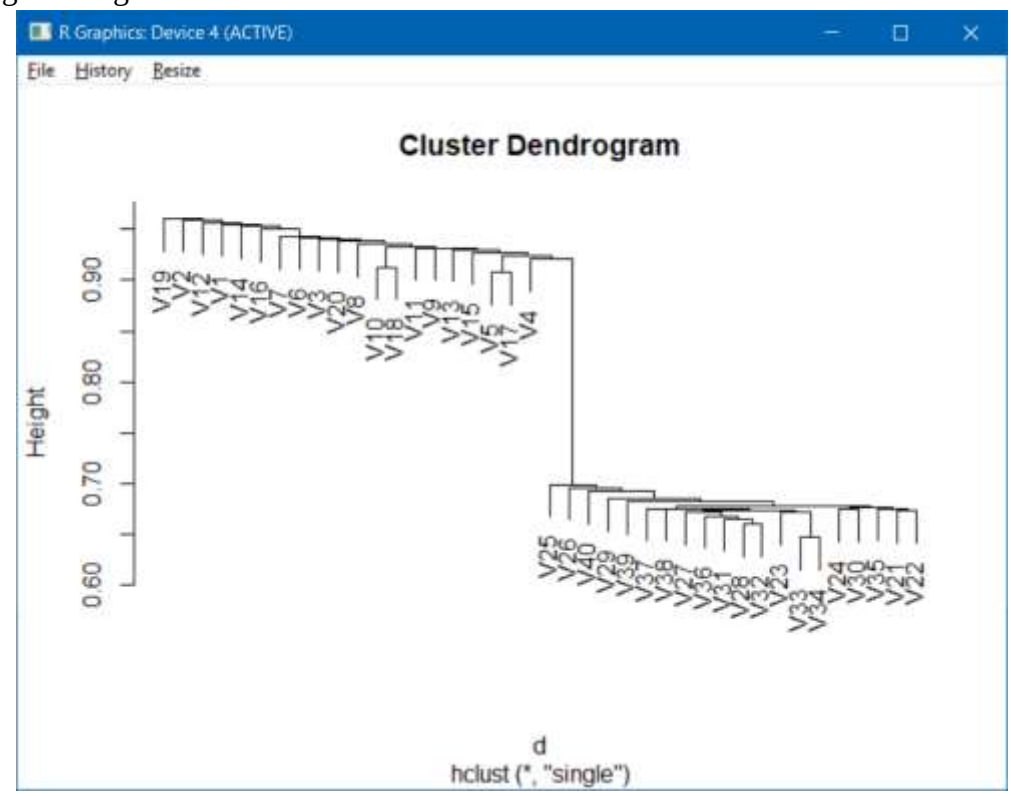
I think scaling of variables must be done, because Euclidean distance is calculated on the numerical value of the variables, without taking into account the fact that the variables may have different scales. For instance, there may be two variables like the length of the owner's swimming pool, measured in metres, and his physical height, measured in cm. The Euclidean distance variable simply takes the numerical values of these variables, so not scaling variables like these would be intuitively wrong.

Problem 11

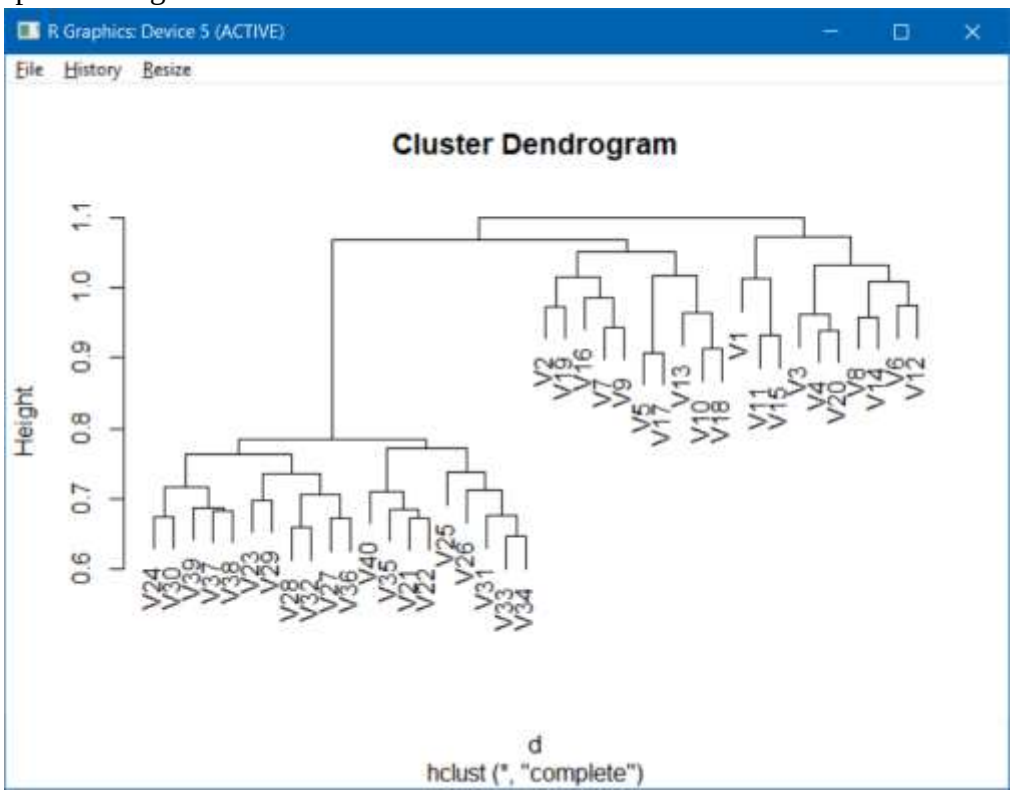
a. This problem required me to use a dataset from the textbook's website, so I downloaded it and imported it in R first, then using the read.csv() function on it with header = F as directed.
b. Next up, I had to apply hierarchical clustering using correlation based distance. Therefore I used the cor() function to calculate the correlation among the values of the dataset and stored the values after coercing them into distance form in variable d. Note that I stored 1 minus the correlation value, since cor() gives the values of correlation, a measure of how similar the values are. As per instruction, I performed the clustering

Kunal Mehta
kunalkam
50352960
for single, complete and average linkage methods. The results of those are as shown below.

Single linkage:

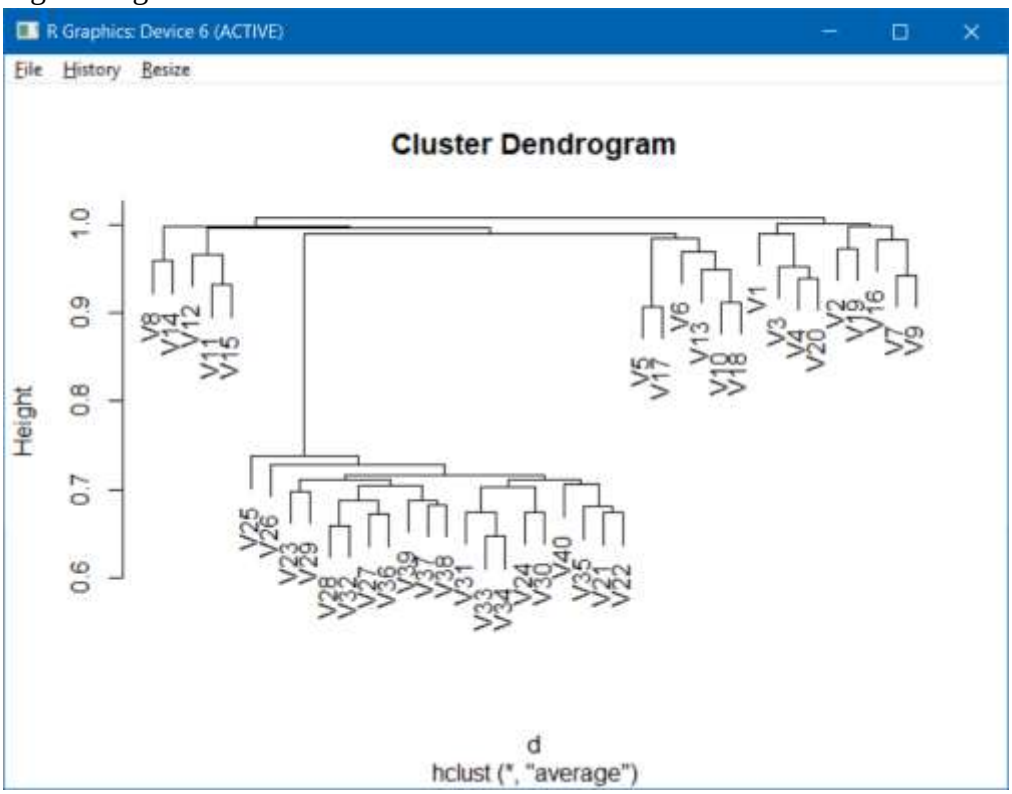


Complete linkage:



Kunal Mehta
kunalkam
50352960

Average linkage:



As observed, the method of linkage does affect the results. In fact, even the number of clusters changes, since we get 3 clusters for average linkage while only 2 for both complete and single linkage. Therefore the genes separate the samples into two groups, only in the case of single and complete linkage.

c. To figure out which genes differ the most, I did principal component analysis. Therefore, I used the prcomp() function, fairly straightforward, storing the results in p. I used the transpose of the dats frame for prcomp because each column of the dataset is one patient, when ideally it should be each row is one patient. Then, I used the apply function to use the sum() function on all the rows of the p$rotation to get the total loadings. Then, all that remained was to arrange the absolute values of the total loadings in descending order to figure out the ones the genes that differed the most since they'd have the largest values. Finally, after ordering them, I printed the head of this list, getting the genes that are the most different across the two groups. These are:

```
> head(ind)
[1] 865  68 911 428 624  11
>
```

Kunal Mehta
kunalkam
50352960

Problem 3

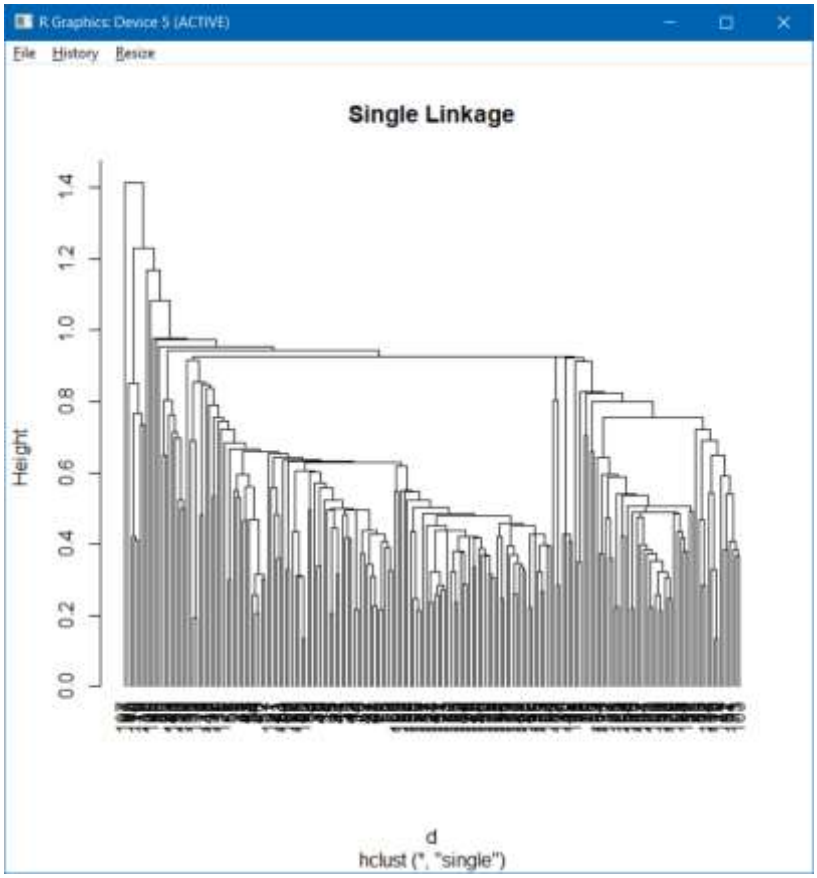
a. I have to perform hierarchical clustering on the dataset without including the seed group column. Therefore the first thing to do was to read the tab separated dataset, which I did using the read.delim() function. Now, the seeds group column is the last column, which I am not supposed to include when clustering, hence, I stored all the rows and all the columns except the seeds group column as a data frame named dats. This I did by dat[,1:7], which indicates all rows, and all columns in the range 1-7, including 7.

```
> head(dat)
   Area Perimeter Compactness Length.Kernel Width.Kernel
1 15.26     14.84      0.8710         5.763        3.312
2 14.88     14.57      0.8811         5.554        3.333
3 14.29     14.09      0.9050         5.291        3.337
4 13.84     13.94      0.8955         5.324        3.379
5 16.14     14.99      0.9034         5.658        3.562
6 14.38     14.21      0.8951         5.386        3.312
  Asymmetry Length.Kernel.Grove Seed.Group
1     2.221               5.220          A
2     1.018               4.956          A
3     2.699               4.825          A
4     2.259               4.805          A
5     1.355               5.175          A
6     2.462               4.956          A
> dats <- dat[ ,1:7]
> head(dats)
   Area Perimeter Compactness Length.Kernel Width.Kernel
1 15.26     14.84      0.8710         5.763        3.312
2 14.88     14.57      0.8811         5.554        3.333
3 14.29     14.09      0.9050         5.291        3.337
4 13.84     13.94      0.8955         5.324        3.379
5 16.14     14.99      0.9034         5.658        3.562
6 14.38     14.21      0.8951         5.386        3.312
  Asymmetry Length.Kernel.Grove
1     2.221               5.220
2     1.018               4.956
3     2.699               4.825
4     2.259               4.805
5     1.355               5.175
6     2.462               4.956
>
```

Next was the simple task of calculating distances and performing hierarchical clustering using all 3 methods, I did those and stored them as hc1, hc2 and hc3 for single, complete and average linkage respectively. The plots for those shown below.

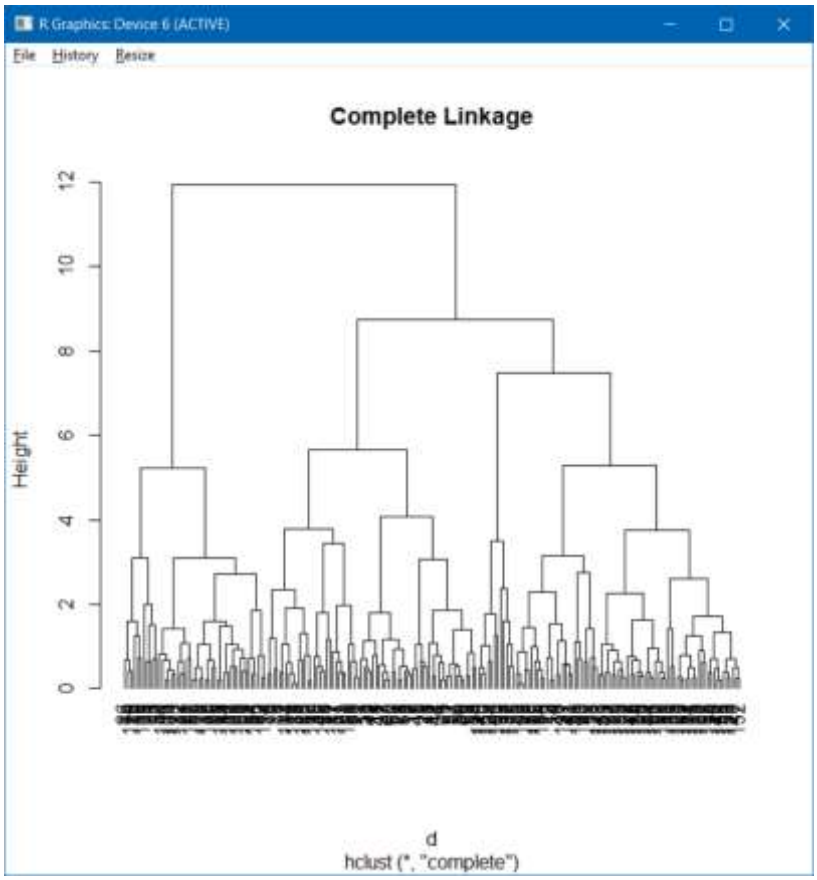
Kunal Mehta
kunalkam
50352960

Single linkage:

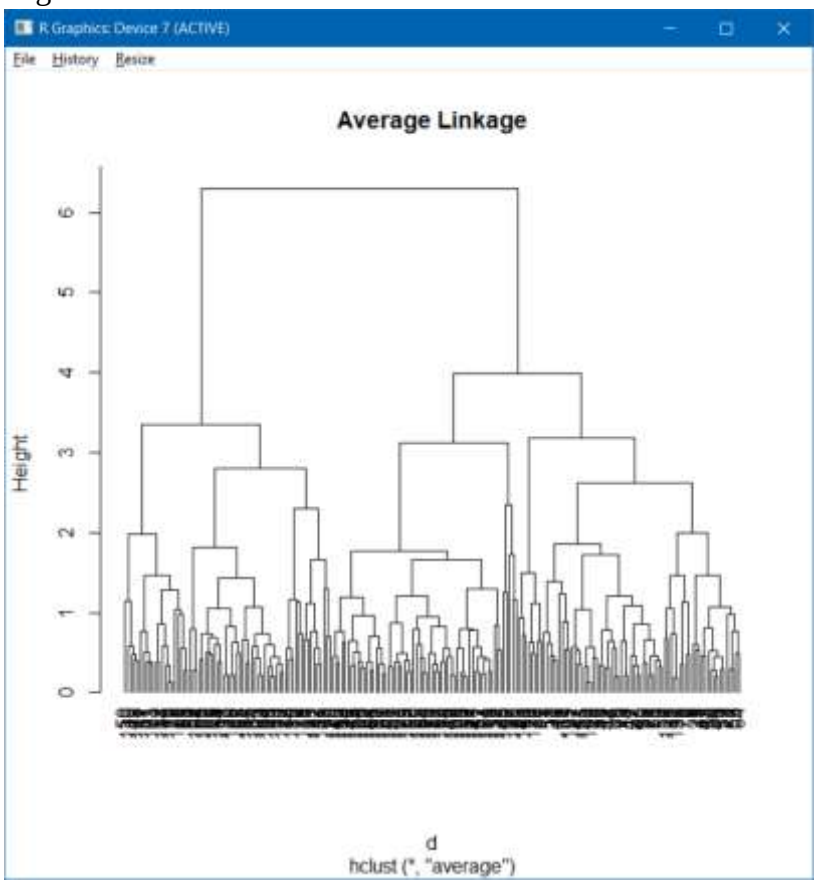


Complete linkage:

Kunal Mehta
kunalkam
50352960



Average linkage:



Kunal Mehta
kunalkam
50352960

Single linkage:

I used the table() function to evaluate the results I received when cutting the dendrogram for increasing values of k, but like I feared from looking at the dendrogram, there isn't a feasibly low value for the number of clusters for which there is even a somewhat distinguished distribution of the observations as per the labels. The observations below illustrate what I mean. Therefore I have gone for a grouping that has k = 3, simply because even on having 22 different clusters, the first one invariably hogs all the observations, and the other two linkage methods have 3 clusters as the best groupings.

k=2

ct1	A	B	C
1	66	68	64
2	0	0	1

k = 3

ct1	A	B	C
1	66	62	64
2	0	6	0
3	0	0	1

k=5

ct1	A	B	C
1	65	61	64
2	1	0	0
3	0	6	0
4	0	1	0
5	0	0	1

k = 10

ct1	A	B	C
1	55	3	60
2	3	58	0
3	1	0	0
4	5	0	3
5	1	0	0
6	1	0	0
7	0	6	0
8	0	1	0
9	0	0	1
10	0	0	1

h = 0.8

ct1	A	B	C
1	51	1	59
2	2	0	0
3	2	2	0
4	1	0	0
5	1	0	0
6	3	0	3
7	2	0	0
8	1	2	0
9	1	1	0
10	1	0	0
11	1	0	0
12	0	49	0
13	0	5	0
14	0	2	0
15	0	1	0
16	0	1	0
17	0	2	0
18	0	1	0
19	0	1	0
20	0	0	1
21	0	0	1
22	0	0	1

h = 0.9

ct1	A	B	C
1	55	3	59
2	2	2	0
3	1	0	0
4	5	0	3
5	1	52	0
6	1	0	0
7	1	0	0
8	0	6	0
9	0	3	0
10	0	1	0
11	0	1	0
12	0	0	1
13	0	0	1
14	0	0	1

h = 1.0	ct1 A B C 1 65 61 64 2 1 0 0 3 0 6 0 4 0 1 0 5 0 0 1	h = 0.85	ct1 A B C 1 54 1 59 2 2 2 0 3 1 0 0 4 5 0 3 5 1 2 0 6 1 52 0 7 1 0 0 8 1 0 0 9 0 5 0 10 0 3 0 11 0 1 0 12 0 1 0 13 0 1 0 14 0 0 1 15 0 0 1 16 0 0 1

Adjusted rand index for 3 clusters is 0.001509422

Complete linkage:
I did the same for complete linkage, comparing the results I received from different values of k and h. The best grouping that I could see from these is to have 3 clusters. The reason for this is that the labels are distributed relatively distinctly in each of the clusters, with the third cluster having exclusively those observations of that have seed group B. Cluster 1 has slightly more than 2/3rds of its observations having seed group A, while cluster 2 has over 3/4th of its observations having seed group C, which makes 3 a healthy number for the clusters.
Adjusted rand index for 3 clusters is 0.520021.
Average linkage:
I did the same for average linkage as well, comparing the results I received from different values of k and h. The best grouping I can see is to have 3 clusters, again. The reason for this is that the labels are distributed very distinctly in each of the 3 clusters with cluster 1 having almost all observations of seed group A, cluster 2 having B and cluster 3 having C. Therefore for average linkage as well, 3 is a good cluster number.
Adjusted rand index for 3 clusters is 0.7482942.

As per the adjusted rand index values, the best performing method of linkage is the average linkage, with single linkage producing the worst results. I expected complete linkage to produce the best result, but totally expected the single linkage results to the worst, and it didn't disappoint, ironically.

b. To determine the best value of k for kmeans, I used the method of bootstrapping cluster stability, doing it with both scheme2 as T and F. Both of these gave 3 as the best number of clusters, and therefore I clustered the data into 3 clusters using both kmeans and kmedoids, just to see what the adj rand index of kmedoids would be for 3 clusters out of curiosity.
The adjusted rand index of kmeans with 3 clusters was 0.7402708
The adjusted rand index of kmeans with 3 clusters was 0.7195739

Kunal Mehta
kunalkam
50352960
Therefore, based on these values, the hierarchical clustering with average linkage very narrowly edges kmeans as the best performing method.
However, despite the fact that hierarchical clustering proved to be the slightly better performing method, I still feel that this is merely a coincidence due to the nature of the values of this dataset. In a more general sense, in a dataset like this where the values are numeric, I believe kmeans or kmedoids will generally produce better performance than hierarchical clustering.