CS5483 Data Warehousing & Data Mining – Assignment 2
Name: Poon Darren Yuk Ho
Student ID: 52833413

Glass Identification

Data set description: number of attributes: 9 (RI, Na, Mg, Al, Si, K, Ca, Ba, Fe)
number of instances:214
classes: “1”, “2”, “3”, “4”, “5”, “6”, “7”

1) Apply normalization to the attributes of the data sets.

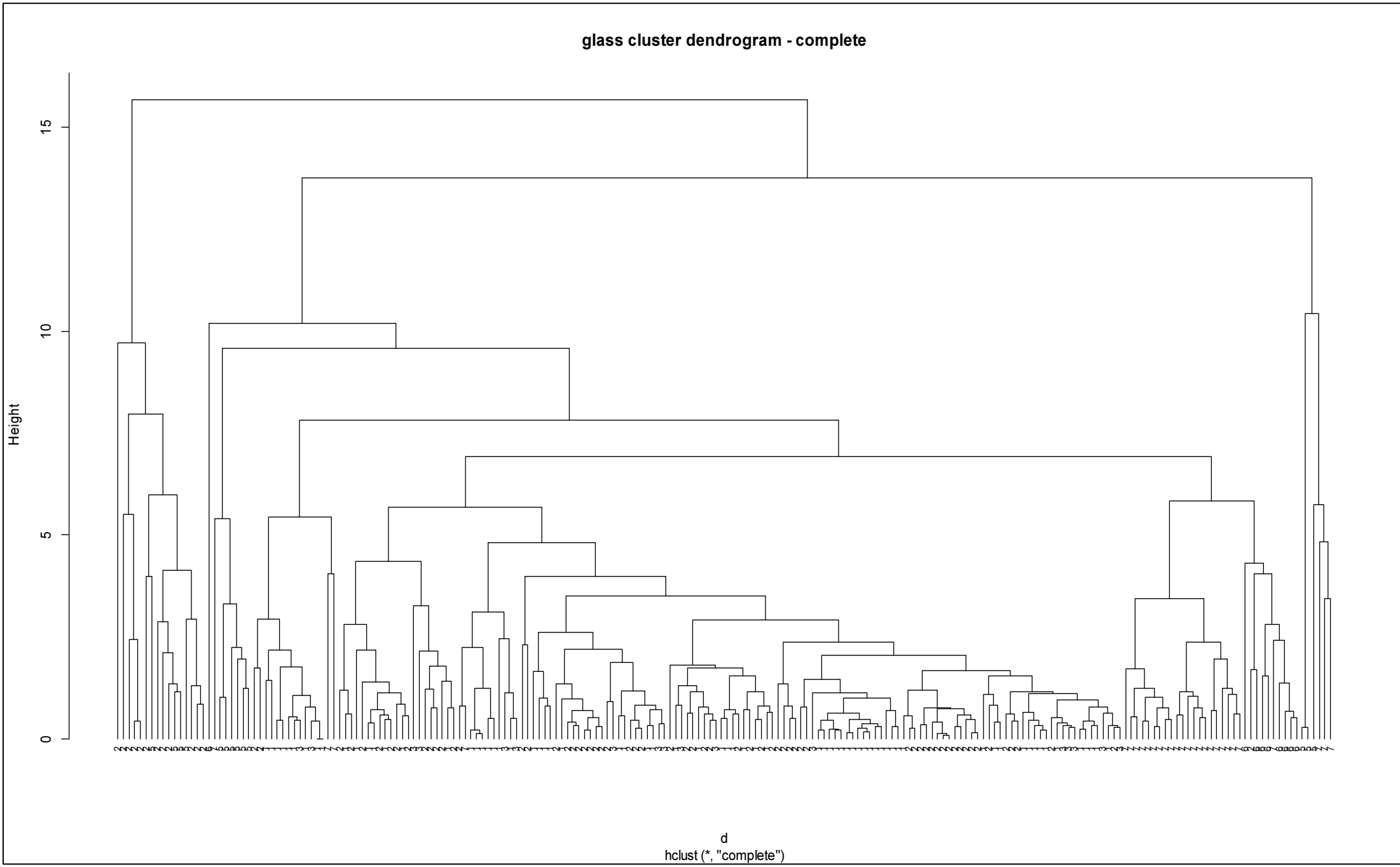
- For the result, please refer to the excel file (under the section “Normalized Data Sets”).
- I used the following equation to normalize the data sets.
  - $x' = (x - m) / sd$
  - Where x is the data, m is the mean of all data under the attribute, and sd is the standard deviation of all data under the attribute. Therefore, the new variable has a mean of 0, and a standard deviation of 1.

2) Compare the hierarchical structures generated using single link, complete link, and group average for the data sets.

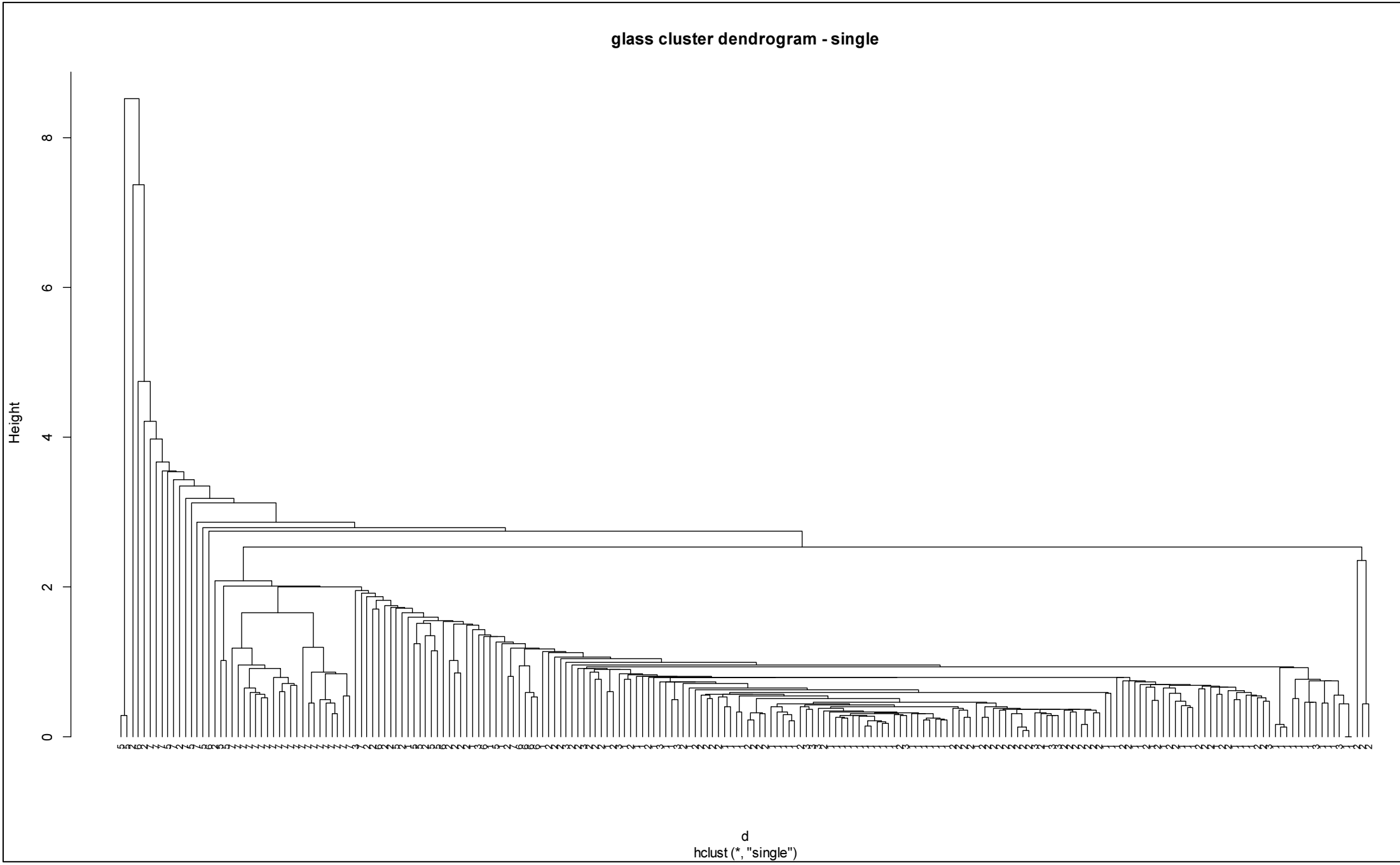
- The hierarchical structure of complete link is more balanced in compared to single link, and group average. We could see the data are grouped more evenly in complete link. However, most data in single link and group average are grouped within the same clusters if the height is 5.

- You may refer to the diagrams as follows.

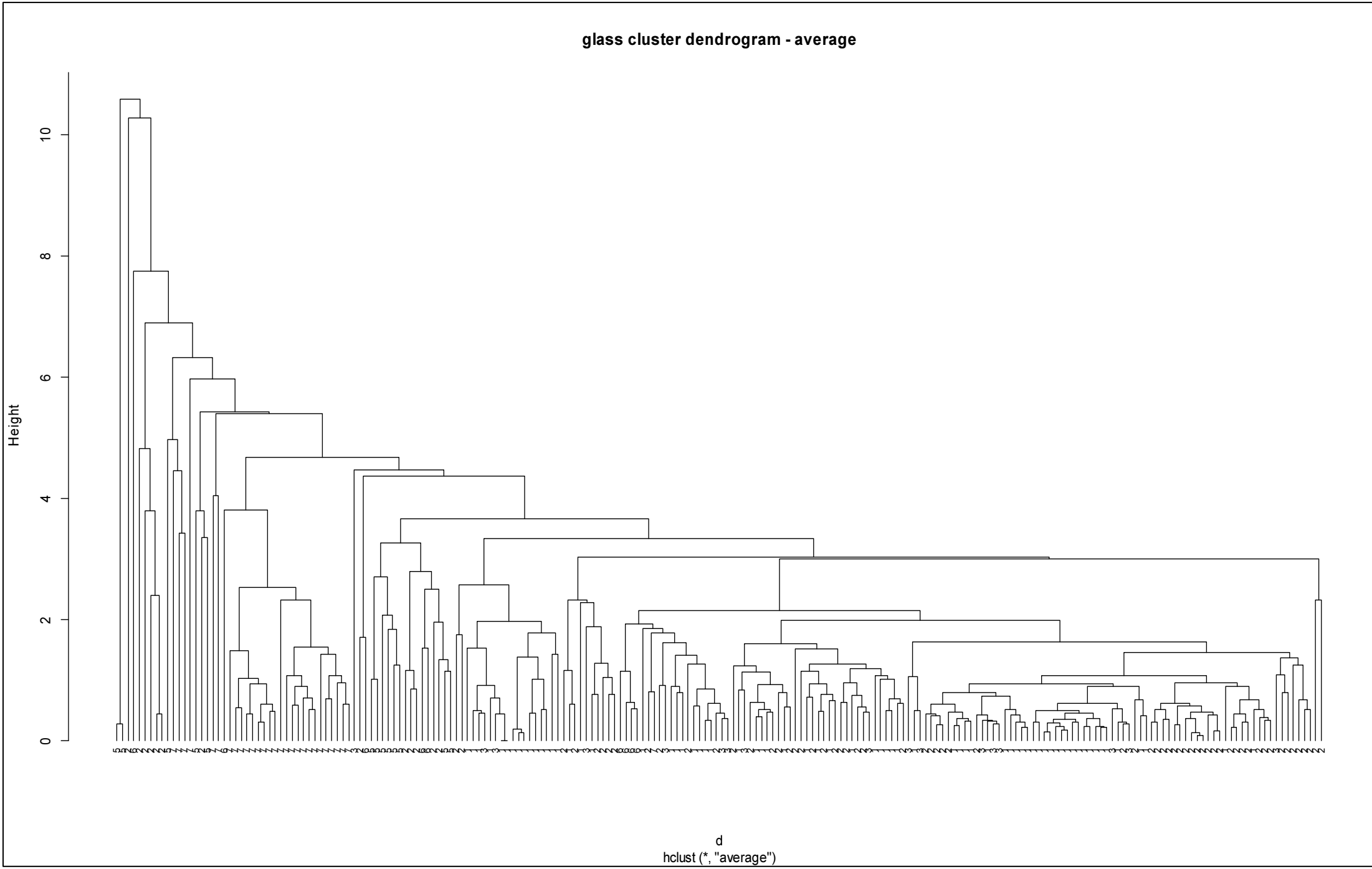
Complete link



Single link


glass cluster dendrogram - single
Height
0
2
4
6
8
d
hclust (*, "single")

Group average

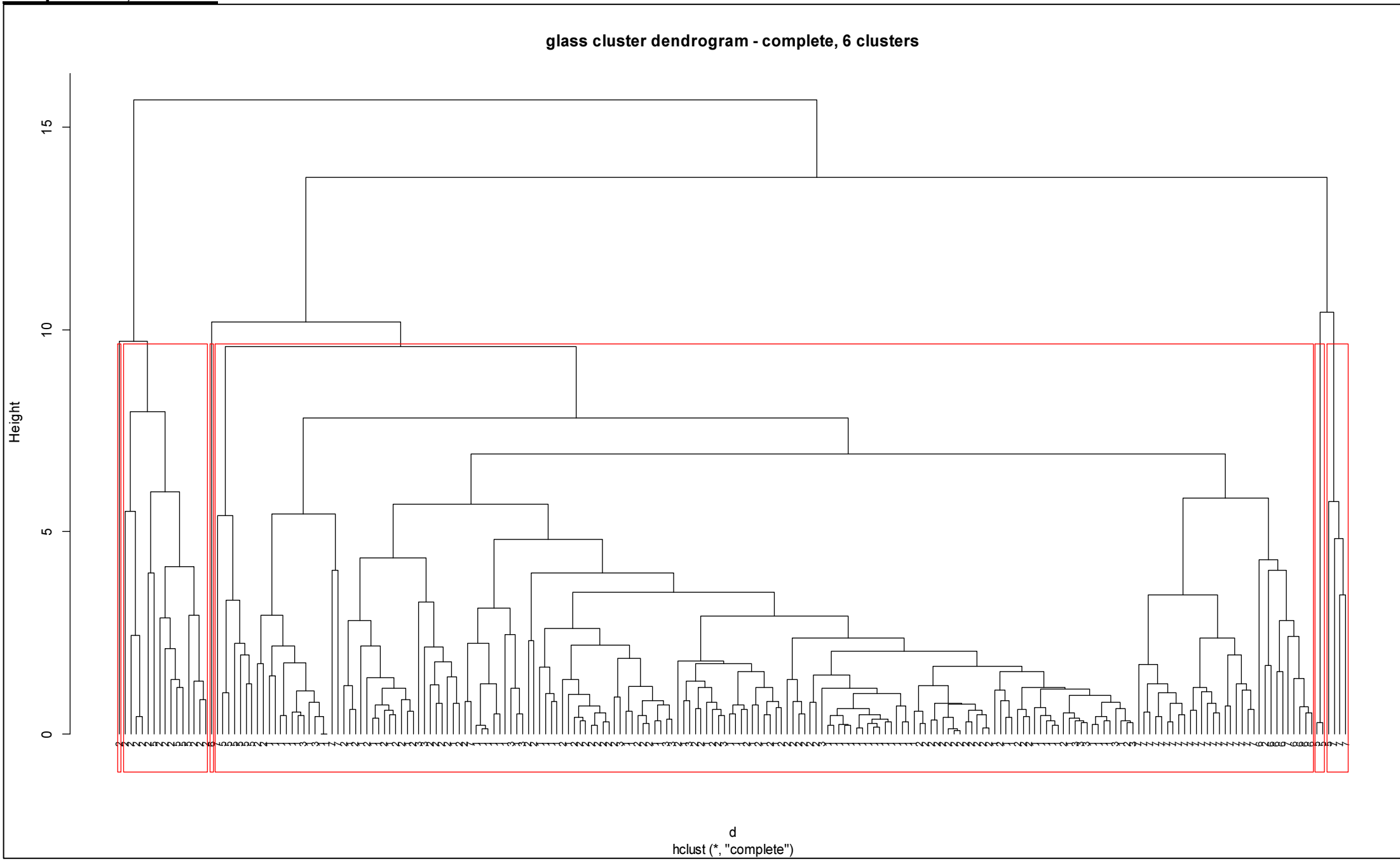


3) Select some of the clustering solutions from the hierarchical structures, and compare the cluster groupings with the actual class structure of the data points.

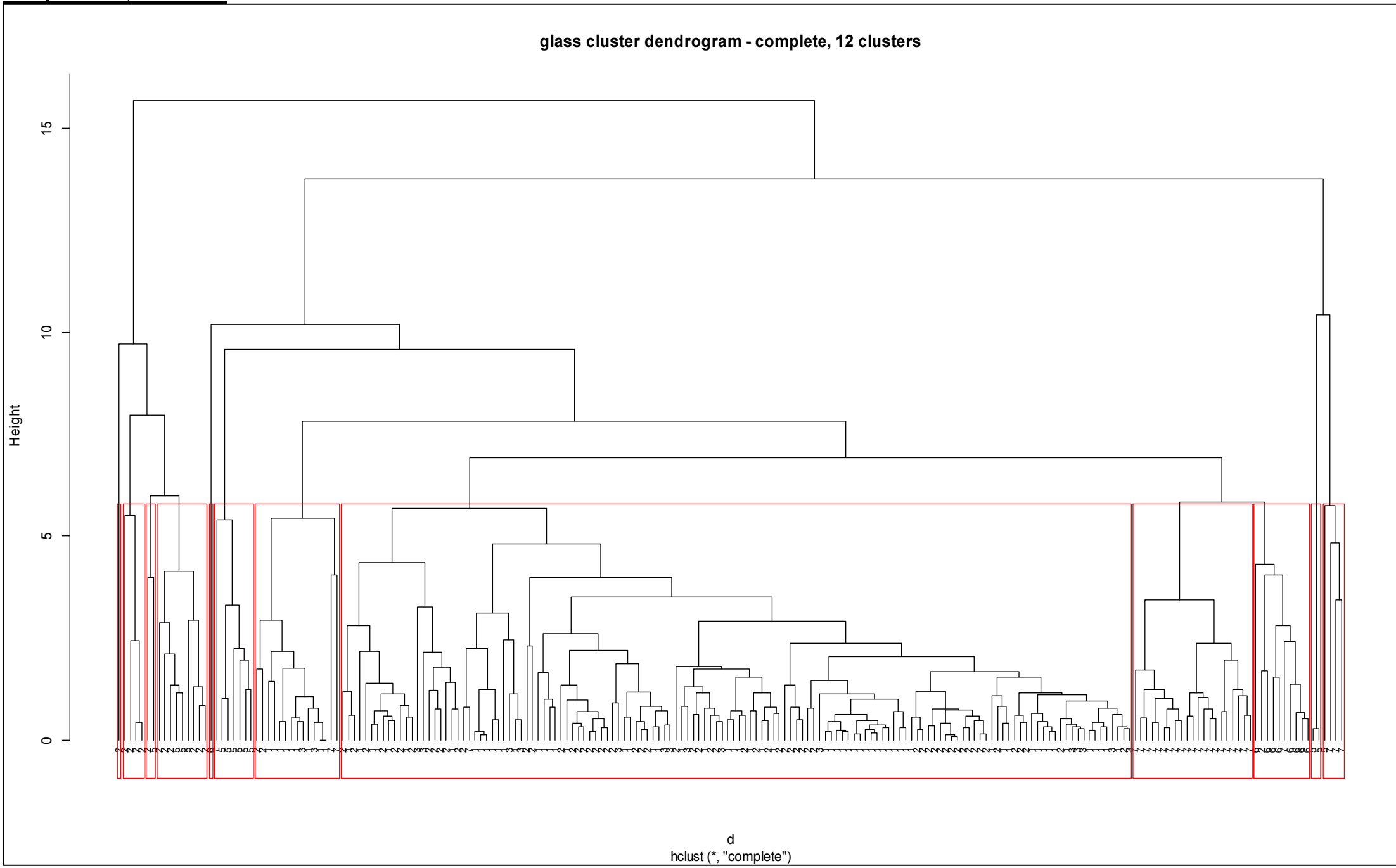
- In this section, I tried different cluster groupings (6, 12, 48 clusters) on the dendrograms that we have constructed in last sections.

- You may refer to the diagrams and results statistics in the following pages.

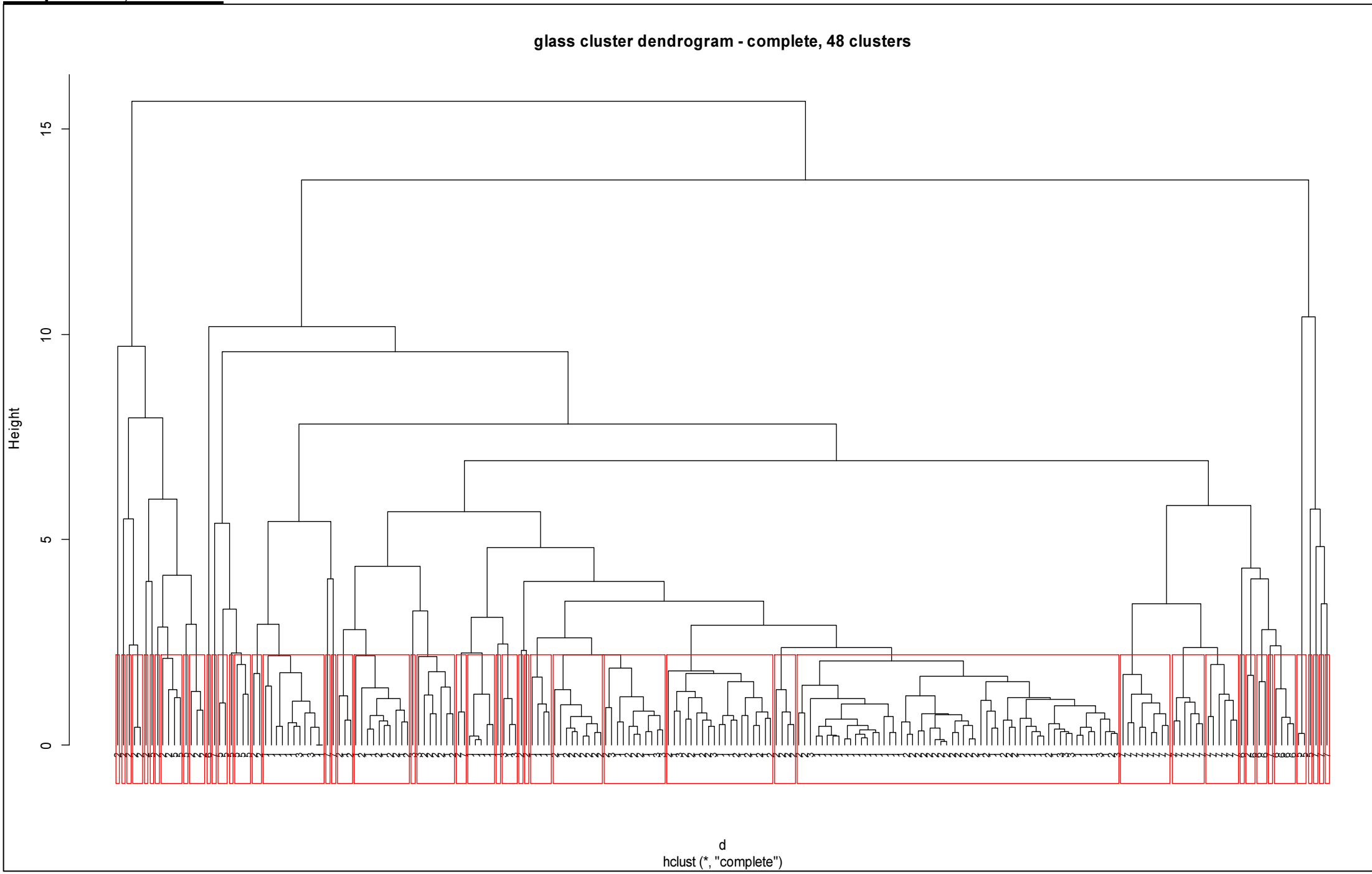
Complete link, 6 clusters



Complete link, 12 clusters



Complete link, 48 clusters



Grouping results of complete link

6 clusters:

```
> table(gComplete.6, nglassdata$glassType)

gComplete.6  1  2  3  5  6  7
          1 70 64 17  6  8 26
          2  0 11  0  4  0  0
          3  0  1  0  0  0  0
          4  0  0  0  1  0  3
          5  0  0  0  2  0  0
          6  0  0  0  0  1  0
```

12 clusters:

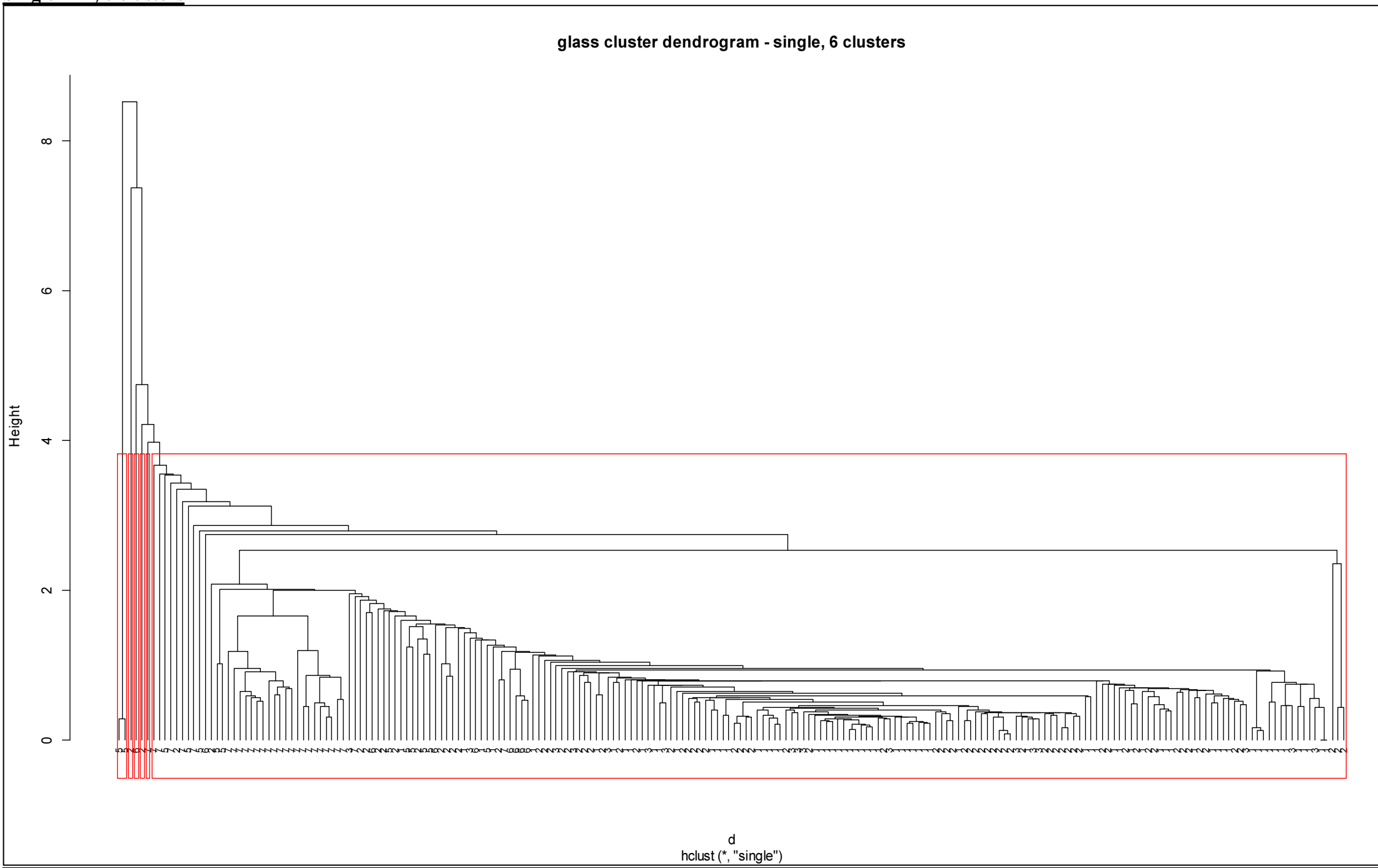
```
> table(gComplete.12, nglassdata$glassType)

gComplete.12  1  2  3  5  6  7
          1  61 61 15  0  0  1
          2   9  2  2  0  0  2
          3   0  1  0  1  0  0
          4   0  1  0  0  0  0
          5   0  4  0  0  0  0
          6   0  6  0  3  0  0
          7   0  1  0  0  8  1
          8   0  0  0  1  0  3
          9   0  0  0  6  0  1
          10  0  0  0  2  0  0
          11  0  0  0  0  1  0
          12  0  0  0  0  0 21
```

48 clusters:

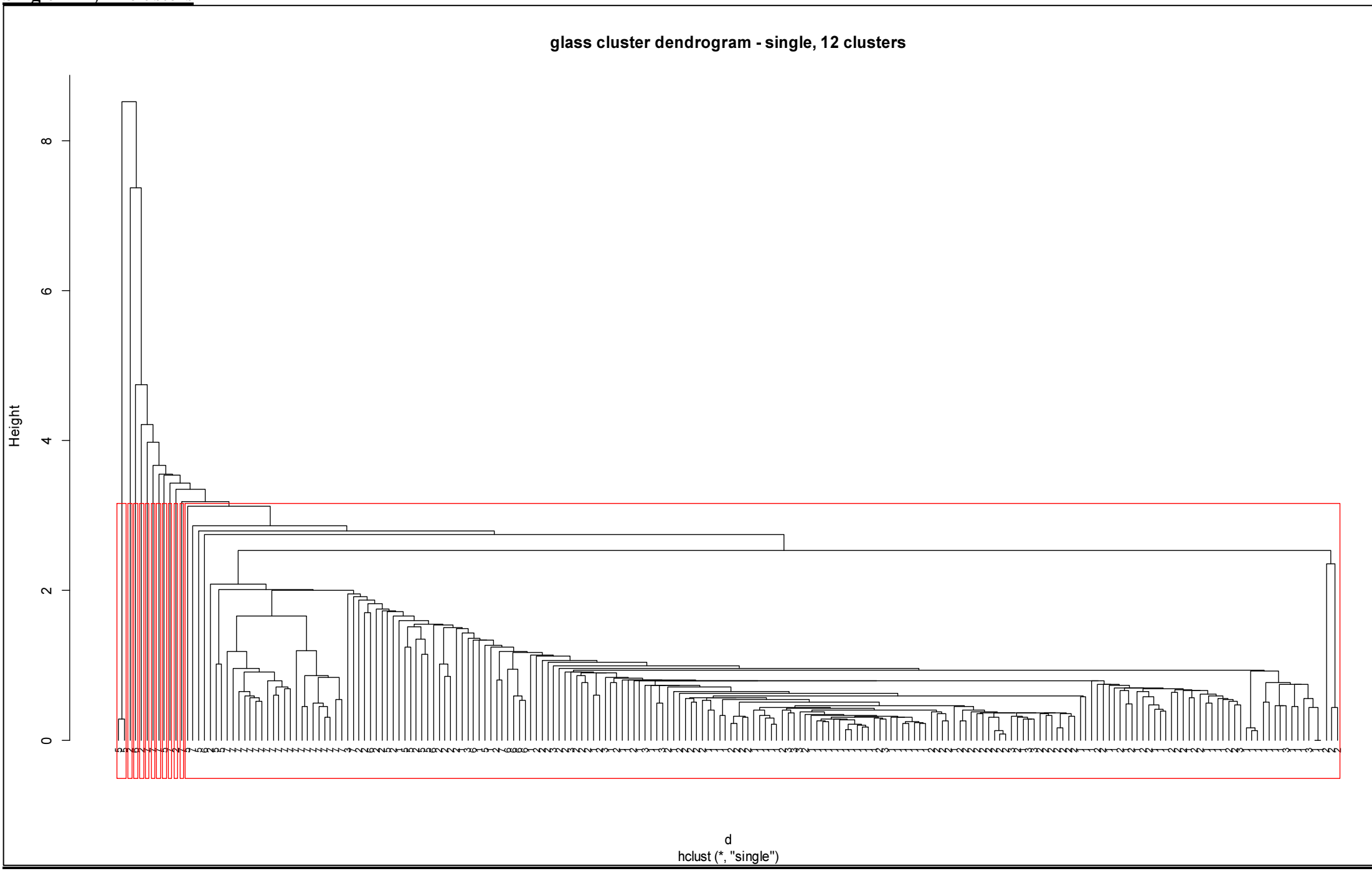
```
> table(gComplete.48, nglassdata$glassType)

gComplete.48  1  2  3  5  6  7
          1   4  0  0  0  0  0
          2   1  8  0  0  0  0
          3  30 21  6  0  0  0
          4   5  5  0  0  0  0
          5   4  4  3  0  0  0
          6   8  9  2  0  0  0
          7   9  0  2  0  0  0
          8   1  0  0  0  0  0
          9   1  5  1  0  0  0
          10  1  2  0  0  0  0
          11  1  0  2  0  0  0
          12  5  0  0  0  0  0
          13  0  1  0  0  0  0
          14  0  4  0  0  0  0
          15  0  1  0  0  0  0
          16  0  2  0  0  0  0
          17  0  1  0  0  0  0
          18  0  1  0  0  0  0
          19  0  1  0  0  0  0
          20  0  2  0  2  0  0
          21  0  1  0  0  1  0
          22  0  2  0  0  0  0
          23  0  1  0  0  0  0
          24  0  1  0  0  0  1
          25  0  3  0  0  0  0
          26  0  1  0  0  0  0
          27  0  0  1  0  0  0
          28  0  0  0  1  0  0
          29  0  0  0  1  0  0
          30  0  0  0  2  0  0
          31  0  0  0  3  0  0
          32  0  0  0  2  0  0
          33  0  0  0  1  0  0
          34  0  0  0  1  0  0
          35  0  0  0  0  4  0
          36  0  0  0  0  1  0
          37  0  0  0  0  2  0
          38  0  0  0  0  1  0
          39  0  0  0  0  0  1
          40  0  0  0  0  0  1
          41  0  0  0  0  0  1
          42  0  0  0  0  0  1
          43  0  0  0  0  0  1
          44  0  0  0  0  0  6
          45  0  0  0  0  0  6
          46  0  0  0  0  0  9
          47  0  0  0  0  0  1
          48  0  0  0  0  0  1
```

- Here, we could see how the data points are grouped in different cluster groupings in complete link.
- In 6 clusters grouping, we could see many data points are misgrouped. Statistically, 64 + 17 + 6 + 8 + 26 + 4 + 1 = 126 data points are grouped incorrectly.
- In 12 clusters grouping, the performance is better than 6 clusters grouping. Statistically, 61 + 15 + 1 + 2 + 2 + 2 + 1 + 3 + 1 + 1 + 1 = 90 data points are grouped incorrectly.
- In 48 cluster groupings, the performance is the best. Only, 1 + 21 + 6 + 5 + 4 + 8 + 2 + 1 + 1 + 1 + 1+ 2 + 1 + 1 = 55 data points are grouped incorrectly.
- Therefore, we could see the performance is getting better with a suitable number of clusters grouping in complete link.

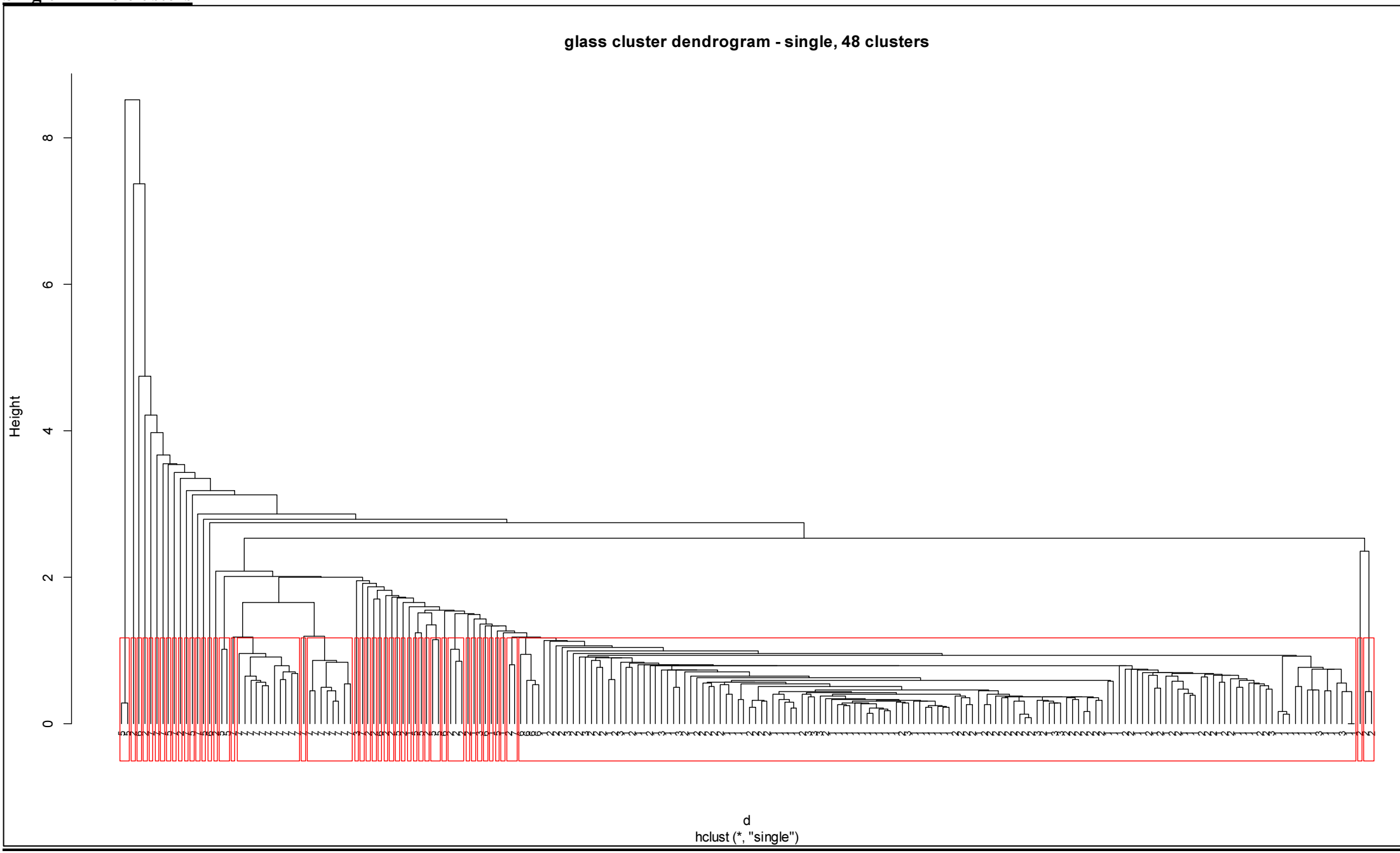
Single link, 6 clusters



Single link, 12 clusters



Single link 48 clusters



Grouping results of single link

6 clusters:

```
> table(gSingle.6, nglassdata$glassType)

gSingle.6  1  2  3  5  6  7
        1 70 74 17 11  8 28
        2  0  1  0  0  0  0
        3  0  1  0  0  0  0
        4  0  0  0  2  0  0
        5  0  0  0  0  1  0
        6  0  0  0  0  0  1
```

12 clusters:

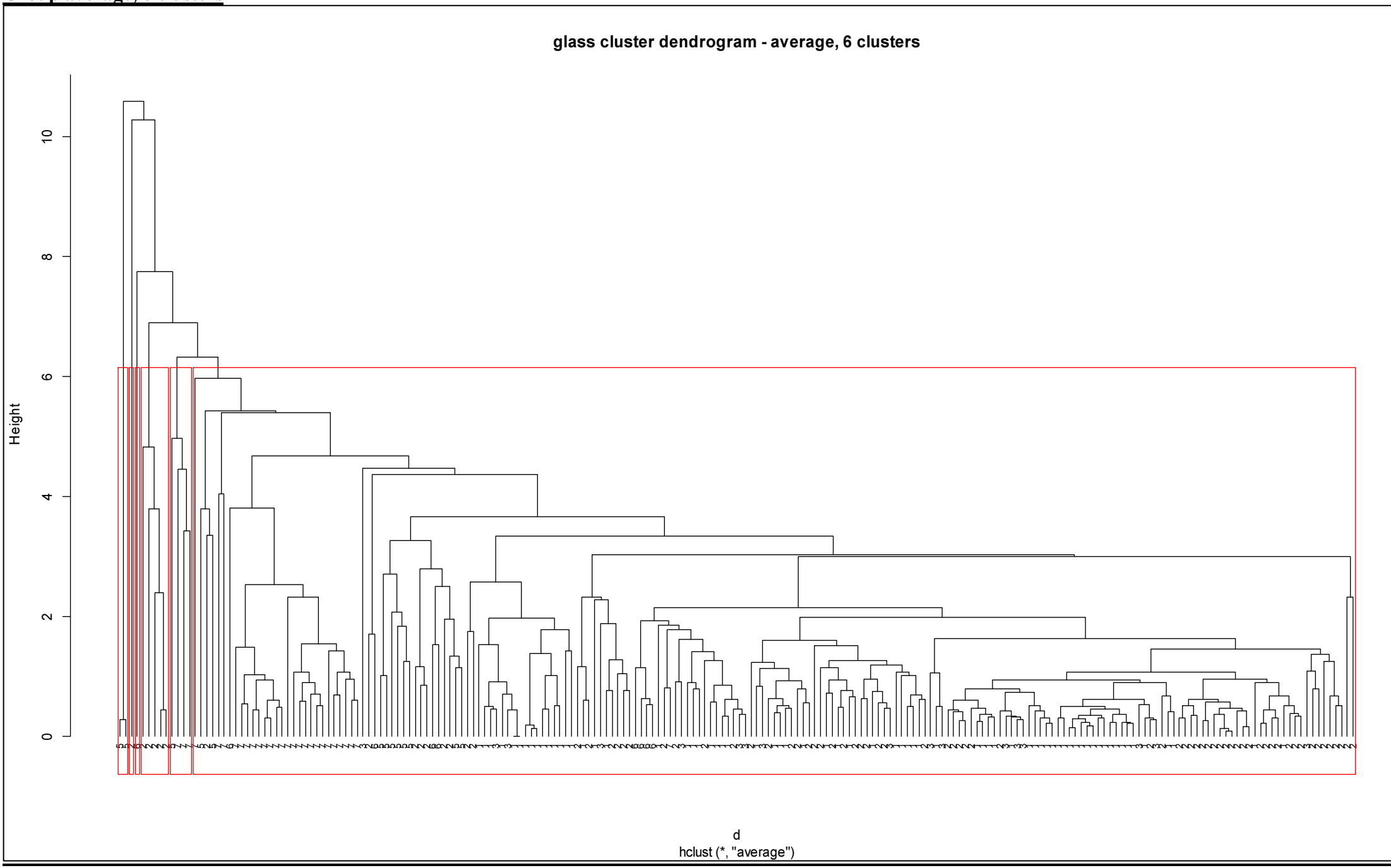
```
> table(gSingle.12, nglassdata$glassType)

gSingle.12  1  2  3  5  6  7
        1  70 73 17 10  8 24
        2   0  1  0  0  0  0
        3   0  1  0  0  0  0
        4   0  1  0  0  0  0
        5   0  0  0  1  0  0
        6   0  0  0  2  0  0
        7   0  0  0  0  1  0
        8   0  0  0  0  0  1
        9   0  0  0  0  0  1
        10  0  0  0  0  0  1
        11  0  0  0  0  0  1
        12  0  0  0  0  0  1
```

48 clusters:

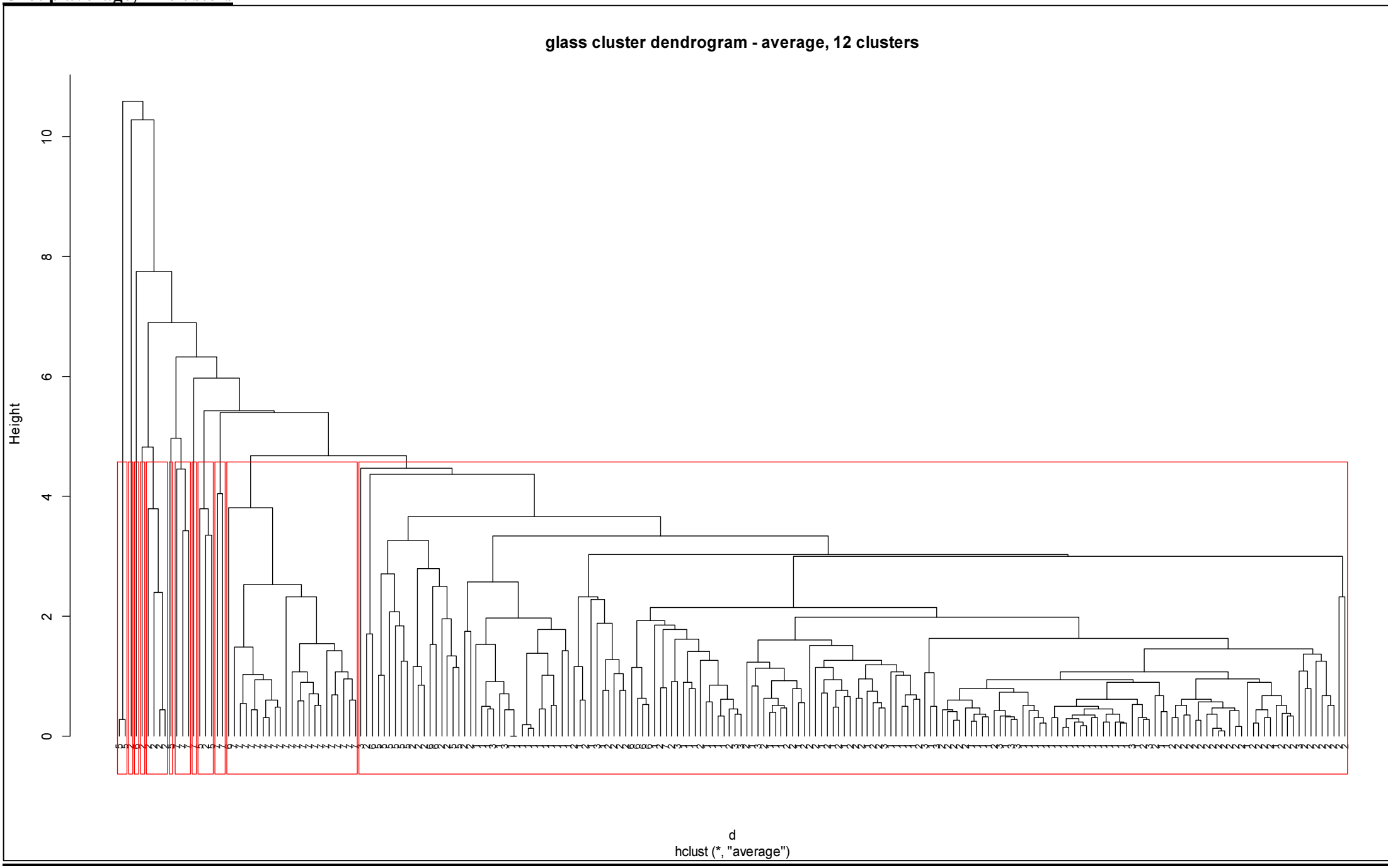
```
> table(gSingle.48, nglassdata$glassType)

gSingle.48  1  2  3  5  6  7
        1  66 58 15  0  4  0
        2   1  0  0  0  0  0
        3   1  0  0  0  0  0
        4   1  0  0  0  0  0
        5   1  0  0  0  0  0
        6   0  1  0  0  0  0
        7   0  1  0  0  0  0
        8   0  1  0  0  0  0
        9   0  1  0  0  0  0
        10  0  1  0  0  0  0
        11  0  1  0  0  0  0
        12  0  1  0  0  0  0
        13  0  1  0  0  0  0
        14  0  1  0  0  0  0
        15  0  2  0  0  0  0
        16  0  1  0  0  0  0
        17  0  1  0  0  0  1
        18  0  3  0  0  0  0
        19  0  1  0  0  0  0
        20  0  1  0  0  0  0
        21  0  0  1  0  0  0
        22  0  0  1  0  0  0
        23  0  0  0  1  0  0
        24  0  0  0  1  0  0
        25  0  0  0  2  0  0
        26  0  0  0  1  0  0
        27  0  0  0  1  0  0
        28  0  0  0  1  0  0
        29  0  0  0  2  0  0
        30  0  0  0  2  0  0
        31  0  0  0  1  0  0
        32  0  0  0  1  0  0
        33  0  0  0  0  1  0
        34  0  0  0  0  1  0
        35  0  0  0  0  1  0
        36  0  0  0  0  1  0
        37  0  0  0  0  1  0
        38  0  0  0  0  0  1
        39  0  0  0  0  0  1
        40  0  0  0  0  0  1
        41  0  0  0  0  0  1
        42  0  0  0  0  0  1
        43  0  0  0  0  0 11
        44  0  0  0  0  0  8
        45  0  0  0  0  0  1
        46  0  0  0  0  0  1
        47  0  0  0  0  0  1
        48  0  0  0  0  0  1
```

- Here, we could see how the data points are grouped in different cluster groupings in single link.
- In 6 clusters grouping, we could see many data points are misgrouped. Statistically, 70 + 17 + 11 + 8 + 28 = 134 data points are grouped incorrectly.
- In 12 clusters grouping, the performance is still bad. Statistically, 70 + 17 + 10 + 8 + 24 = 129 data points are grouped incorrectly.
- In 48 cluster groupings, the performance is improved. Only 58 + 15 + 4 + 1= 78 data points are grouped incorrectly.
- Therefore, we could see 6 cluster and 12 cluster groupings do not work too well, as many of the clusters contains only 1-2 data points, hence, many data points are grouped incorrectly. Even though, the performance in 48 cluster grouping is greatly improved, there are still many clusters contain very few data points.

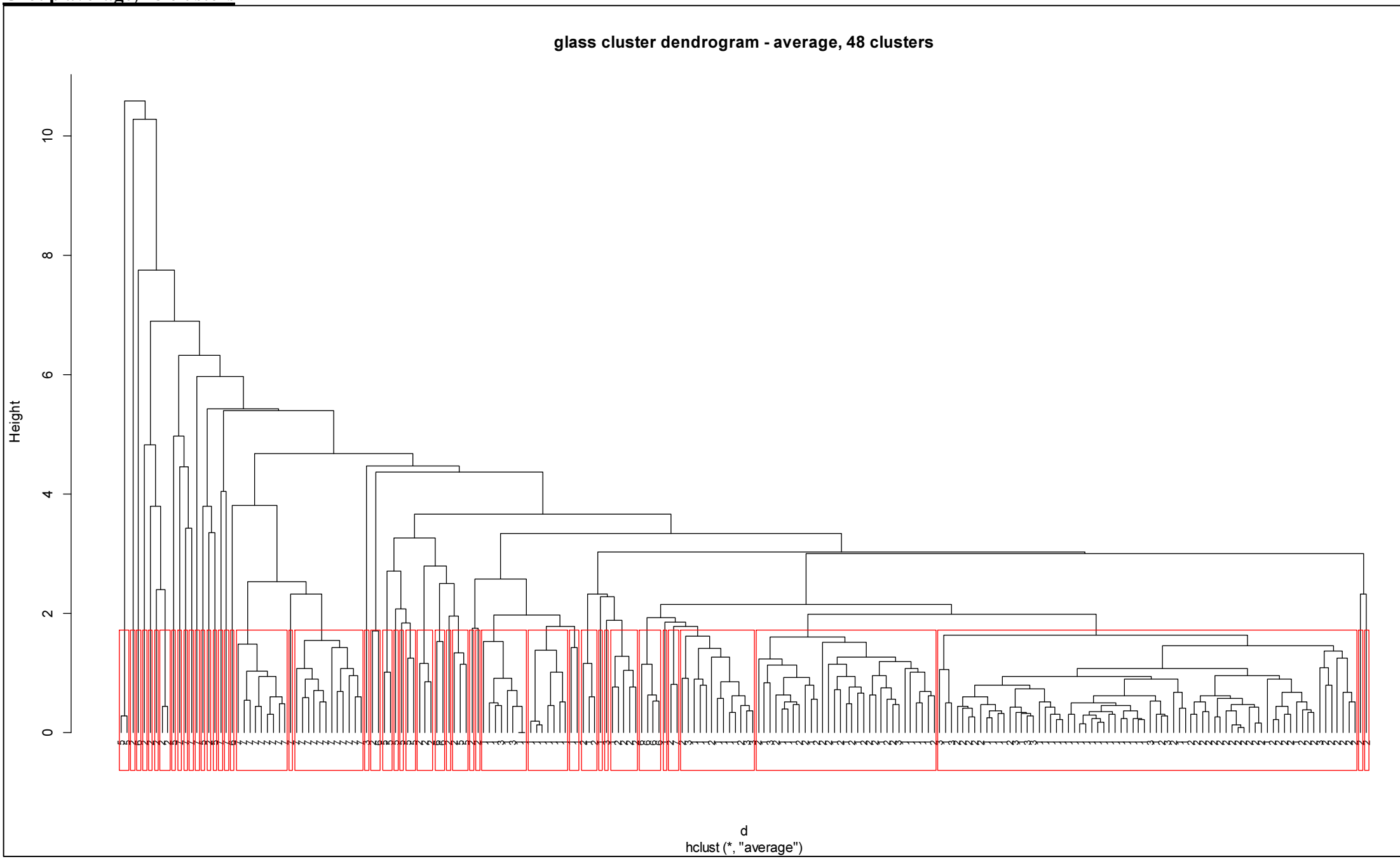
Group average, 6 clusters



Group average, 12 clusters



Group average, 48 clusters



Grouping results of group average

6 clusters:

```
> table(gAverage.6, nglassdata$glassType)

gAverage.6  1  2  3  5  6  7
         1 70 70 17 10  8 26
         2  0  1  0  0  0  0
         3  0  5  0  0  0  0
         4  0  0  0  1  0  3
         5  0  0  0  2  0  0
         6  0  0  0  0  1  0
```

12 clusters:

```
> table(gAverage.12, nglassdata$glassType)

gAverage.12  1  2  3  5  6  7
         1  70 69 17  8  7  1
         2   0  1  0  2  0  0
         3   0  1  0  0  0  0
         4   0  1  0  0  0  0
         5   0  4  0  0  0  0
         6   0  0  0  1  0  0
         7   0  0  0  2  0  0
         8   0  0  0  0  1 22
         9   0  0  0  0  1  0
         10  0  0  0  0  0  3
         11  0  0  0  0  0  2
         12  0  0  0  0  0  1
```

- Here, we could see how the data points are grouped in different cluster groupings in group average.
- In 6 clusters grouping, we could see many data points are misgrouped. Statistically, 70 + 17 + 10 + 8 + 26 + 1 = 132 data points are grouped incorrectly.
- In 12 clusters grouping, the performance is just slightly improved. Statistically, 69 + 17 + 8 + 7 + 1 + 1 = 103 data points are grouped incorrectly.
- In 48 cluster groupings, the performance is the best. Only, 3 + 3 + 30 + 8 + 14 + 2 + 2 + 1 + 1 + 1 + 1 + 1 + 1 = 68 data points are grouped incorrectly.
- Therefore, we could see 6 cluster and 12 cluster groupings do not work to well, as many of the clusters contains only 1-2 data points, hence, many data points are grouped incorrectly. Even though, the performance in 48 cluster grouping is greatly improved, there are still many clusters contain very few data points.

48 clusters:

```
> table(gAverage.48, nglassdata$glassType)

gAverage.48  1  2  3  5  6  7
         1   7  3  3  0  0  0
         2  30 34  8  0  0  0
         3  14 15  2  0  0  0
         4   6  0  2  0  0  0
         5   1  4  0  0  0  0
         6   2  0  0  0  0  0
         7   7  0  0  0  0  0
         8   1  2  0  0  0  0
         9   1  0  0  0  0  0
         10  1  0  0  0  0  0
         11  0  1  0  0  0  0
         12  0  1  0  0  0  0
         13  0  1  0  0  0  0
         14  0  1  0  0  0  0
         15  0  1  0  0  0  0
         16  0  1  0  0  0  0
         17  0  1  0  0  0  0
         18  0  1  0  0  0  0
         19  0  1  0  0  1  0
         20  0  2  0  0  0  0
         21  0  1  0  0  0  0
         22  0  1  0  0  0  1
         23  0  3  0  0  0  0
         24  0  1  0  2  0  0
         25  0  1  0  0  0  0
         26  0  0  1  0  0  0
         27  0  0  1  0  0  0
         28  0  0  0  1  0  0
         29  0  0  0  1  0  0
         30  0  0  0  2  0  0
         31  0  0  0  2  0  0
         32  0  0  0  1  0  0
         33  0  0  0  2  0  0
         34  0  0  0  1  0  0
         35  0  0  0  1  0  0
         36  0  0  0  0  4  0
         37  0  0  0  0  1  0
         38  0  0  0  0  2  0
         39  0  0  0  0  1  0
         40  0  0  0  0  0  1
         41  0  0  0  0  0  1
         42  0  0  0  0  0  1
         43  0  0  0  0  0  1
         44  0  0  0  0  0  1
         45  0  0  0  0  0 12
         46  0  0  0  0  0  9
         47  0  0  0  0  0  1
         48  0  0  0  0  0  1
```

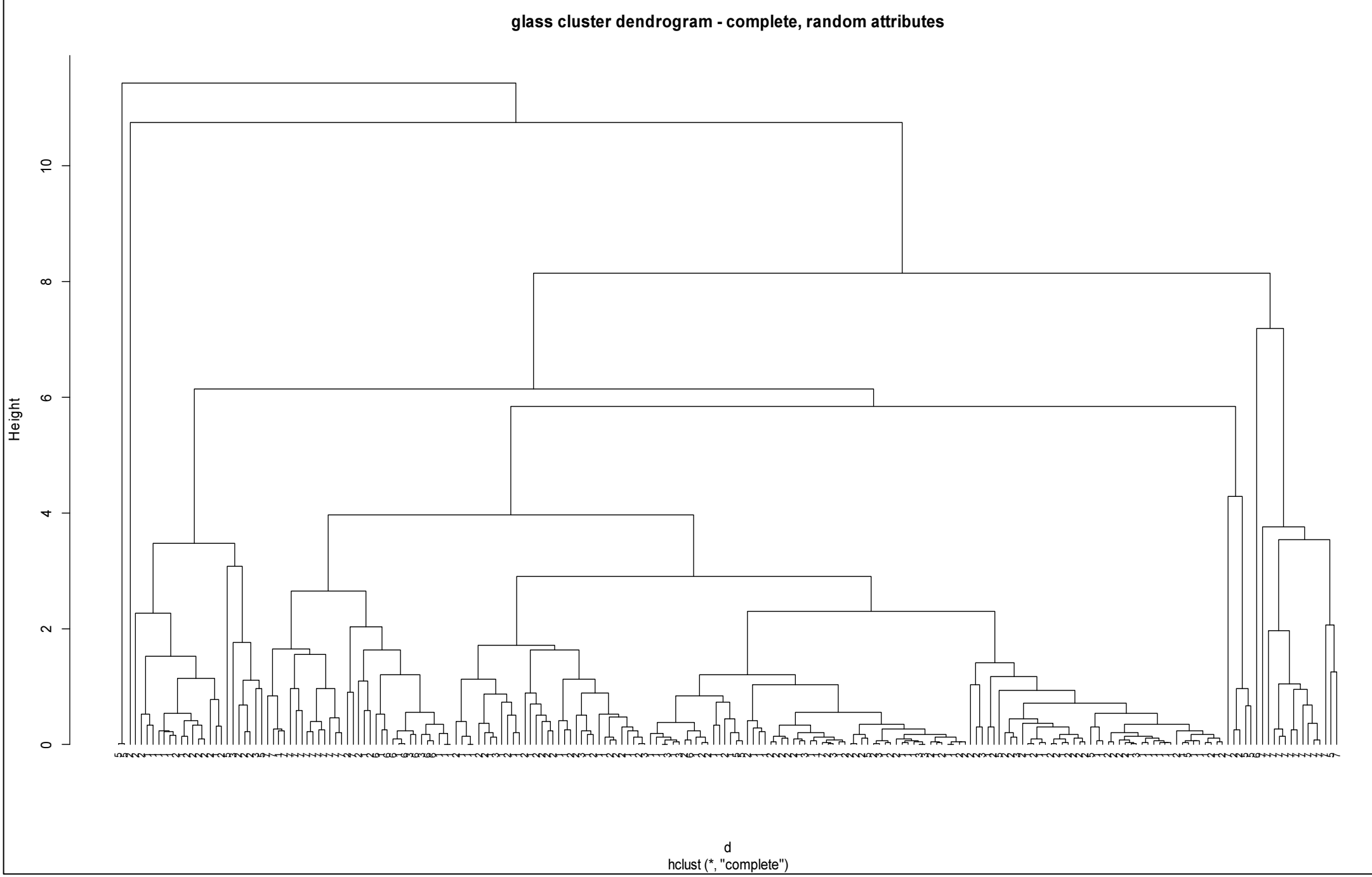
4) Select different random subsets of attributes from the data sets and re-perform hierarchical clustering. Compare the resulting hierarchical structures based on the selected attributes subsets with the original hierarchical structures.

- In this section, I randomly pick {Na, K, Ba, Fe} as the only attributes to re-perform hierarchical clustering.

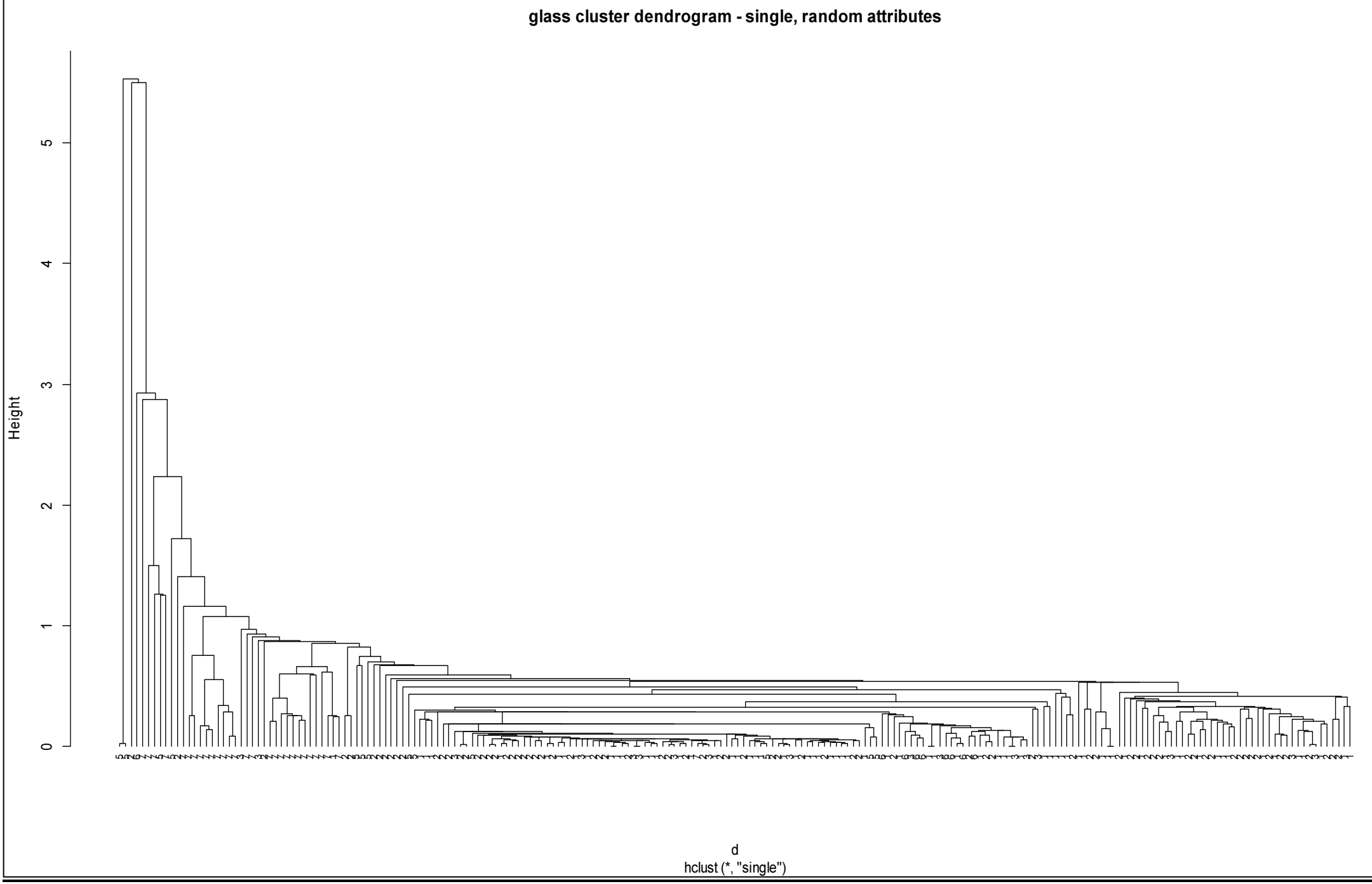
- Compared to the original hierarchical structures, the resulting hierarchical structures that are generated with the selected attributes have a very similar structure. Specifically, if we cut the dendrograms with a small number of clusters (for example, k = 6, cutree(tree, k=6)), most data points would still be grouped into the same clusters. But when we cut the dendrograms with a large number of clusters (for example, k = 48, cutree(tree, k=48)), the results of which clusters that data points will be grouped in a slightly different way.

- For dendrograms of re-performed hierarchical clustering, please refer to the following pages.

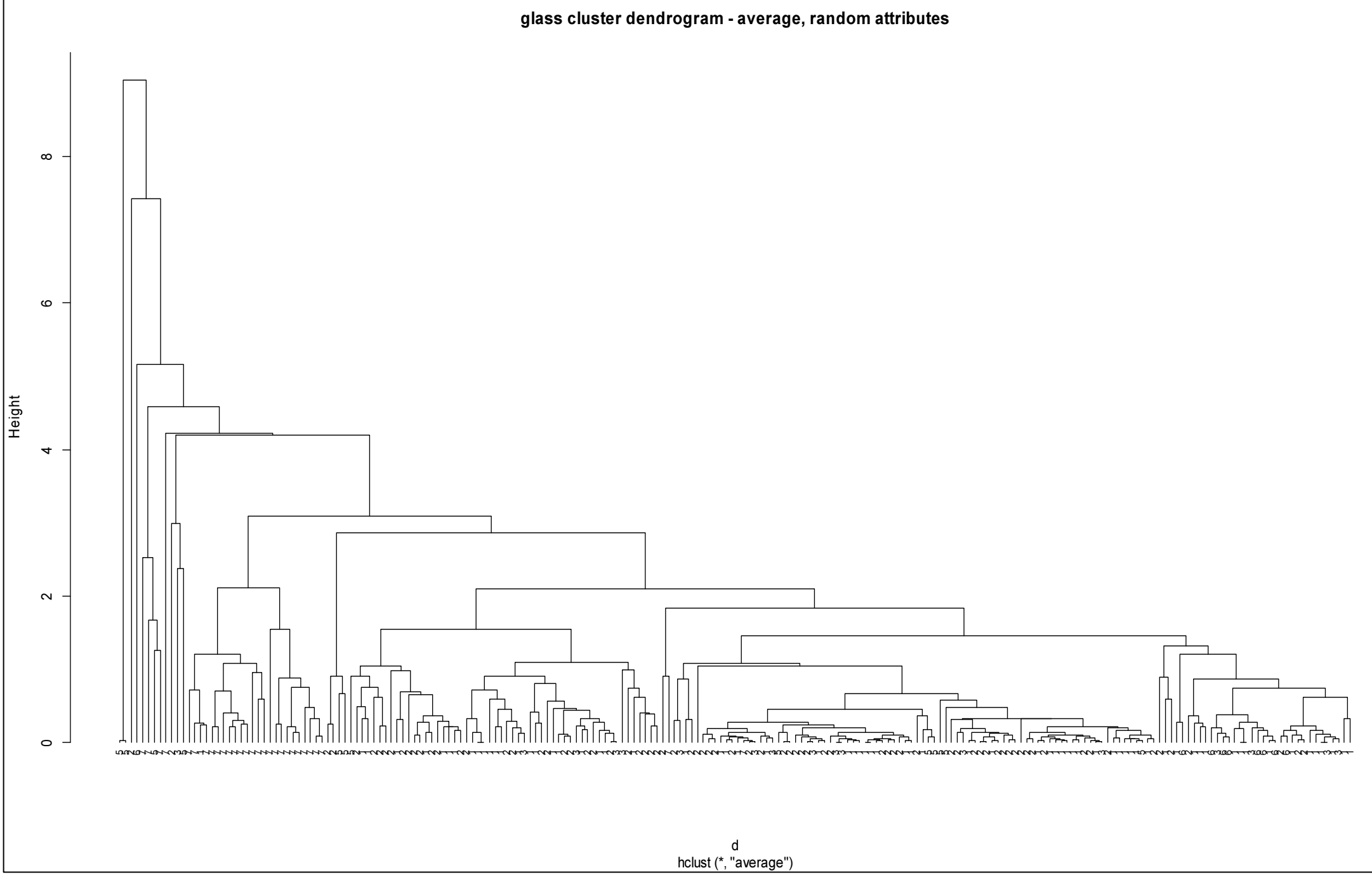
Complete link, random attributes



Single link, random attributes



Group average, random attributes



Wine

Data set description: number of attributes: 13 (Al, Ma, Ash, Aoa, Mag, TP, F, NP, P, CI, H, OD, Pro)
number of instances: 178
classes: “1”, “2”, “3”

1) Apply normalization to the attributes of the data sets.

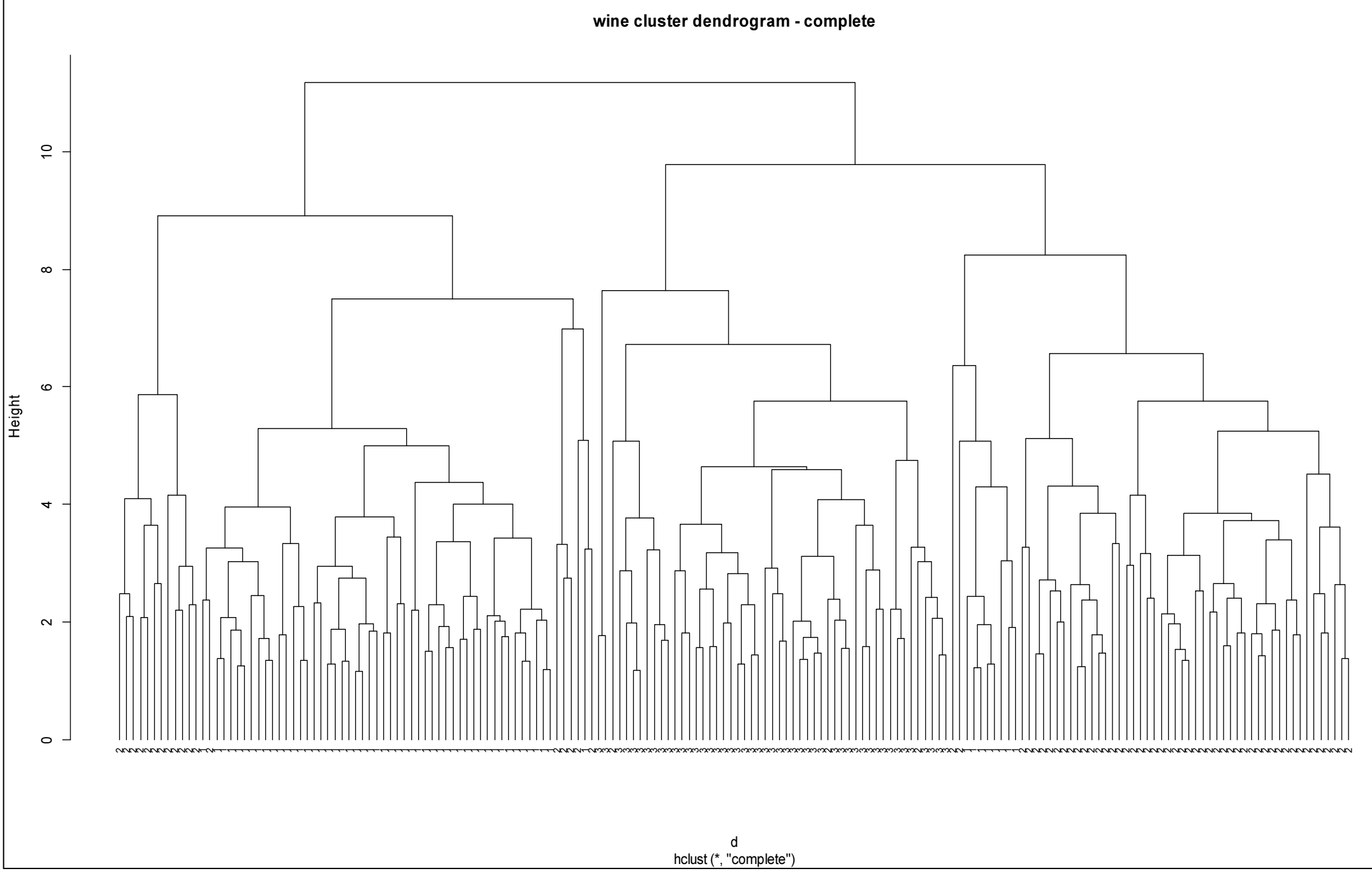
- For the result, please refer to the excel file (under the section “Normalized Data Sets”).
- I used the following equation to normalize the data sets.
  - $x' = (x - m) / sd$
  - Where x is the data, m is the mean of all data under the attribute, and sd is the standard deviation of all data under the attribute. Therefore, the new variable has a mean of 0, and a standard deviation of 1.

2) Compare the hierarchical structures generated using single link, complete link, and group average for the data sets.

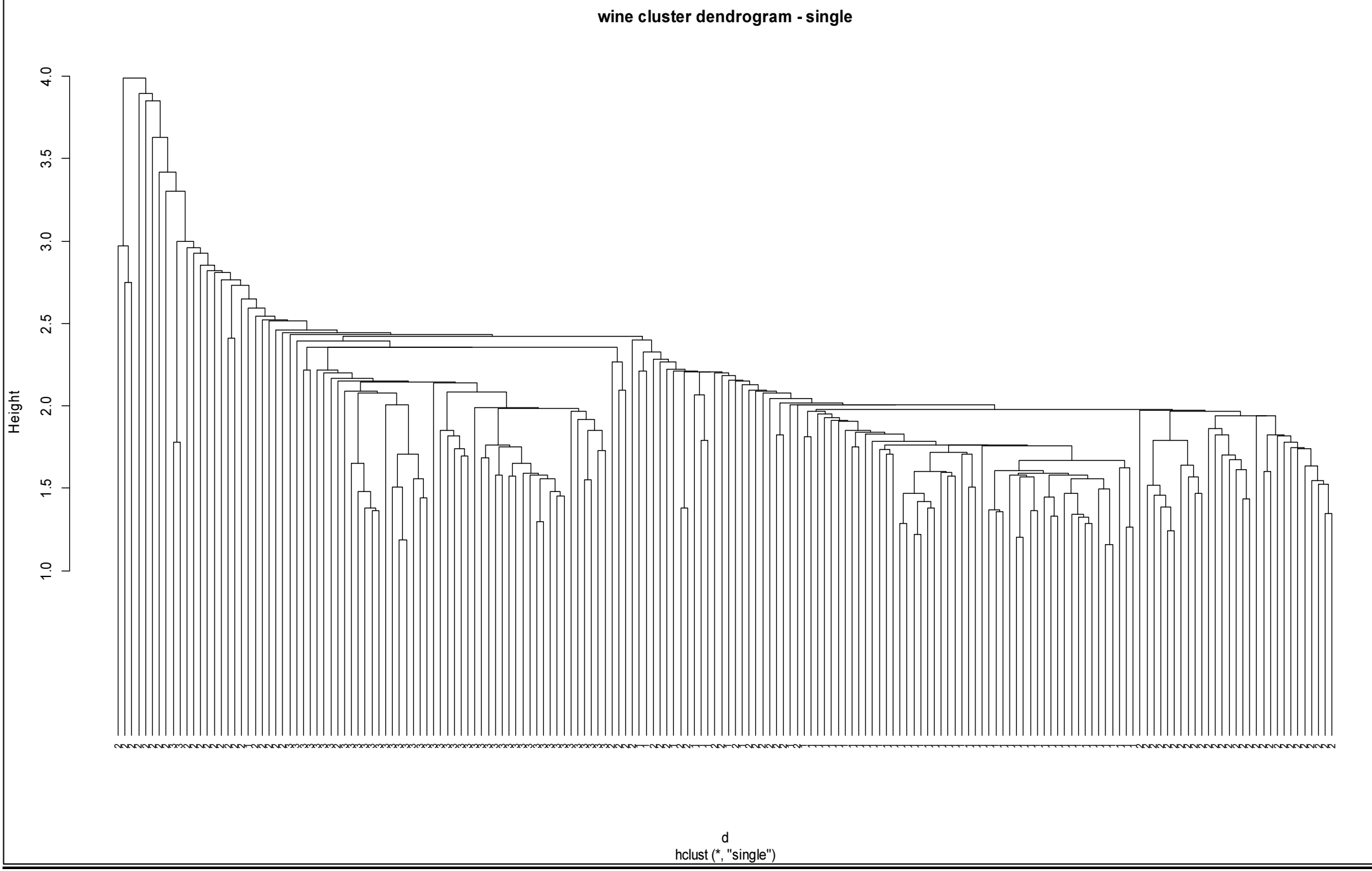
- The hierarchical structures of complete link and group average are more balanced in compared to single link. We could see the data are grouped more evenly in complete link and group average. However, most data in single link are grouped within the same clusters if the height is 4.

- You may refer to the diagrams as follows.

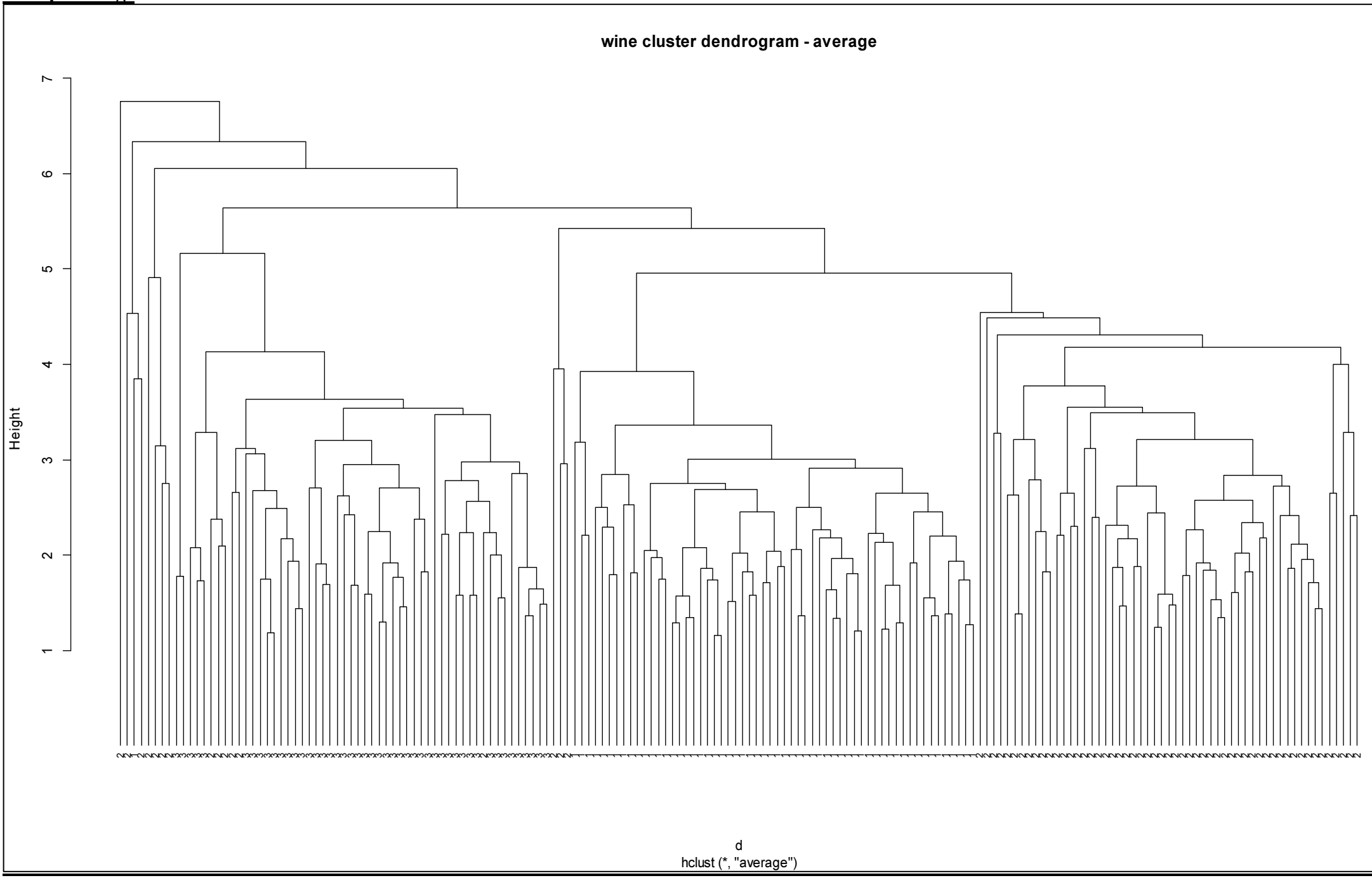
Complete link



Single link


wine cluster dendrogram - single
Height
4.0
3.5
3.0
2.5
2.0
1.5
1.0
d
hclust (*, "single")

Group Average

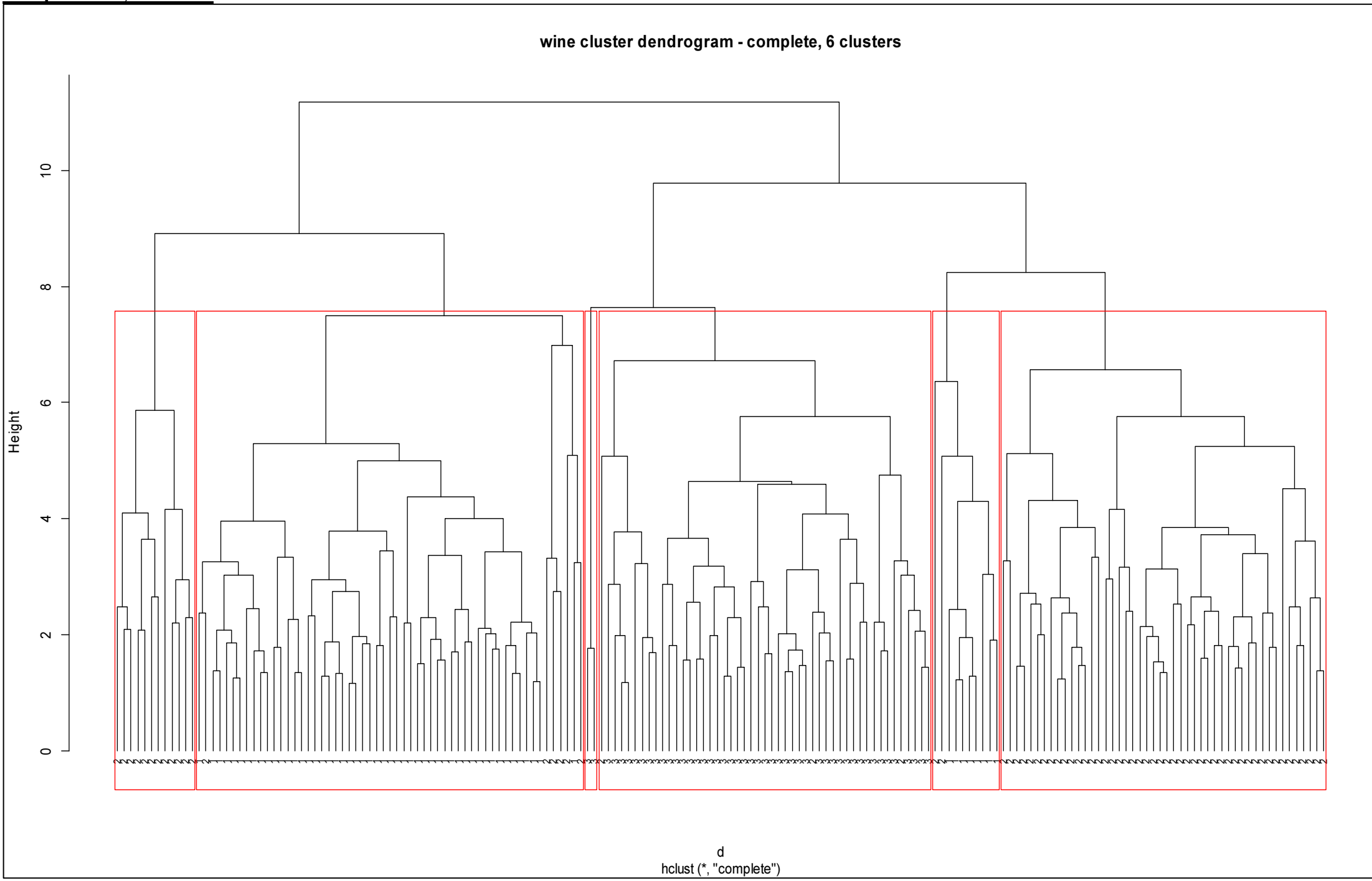


3) Select some of the clustering solutions from the hierarchical structures, and compare the cluster groupings with the actual class structure of the data points.

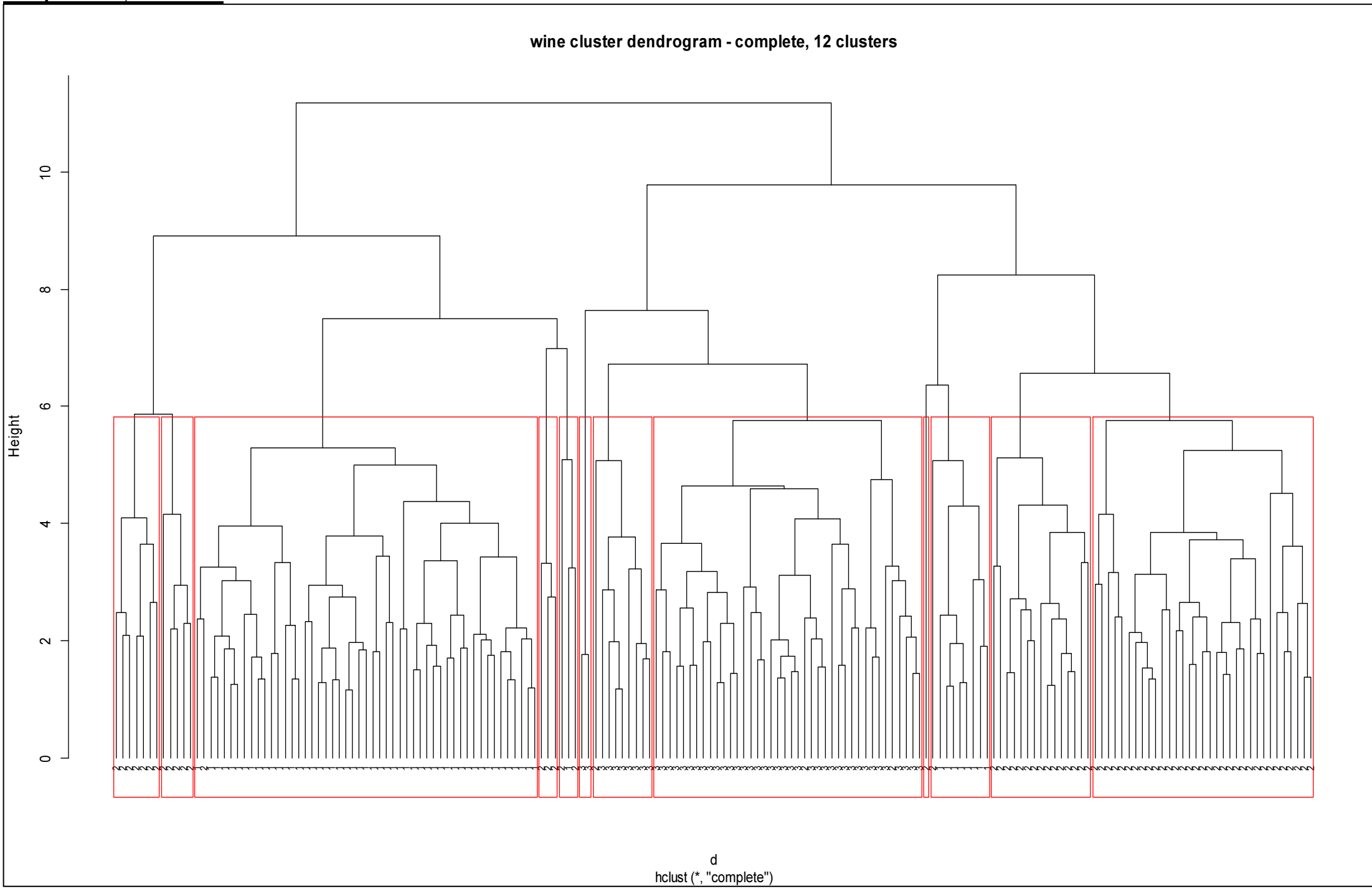
- In this section, I tried different cluster groupings (6, 12, 48 clusters) on the dendrograms that we have constructed in last sections.

- You may refer to the diagrams and results statistics in the following pages or the excel file.

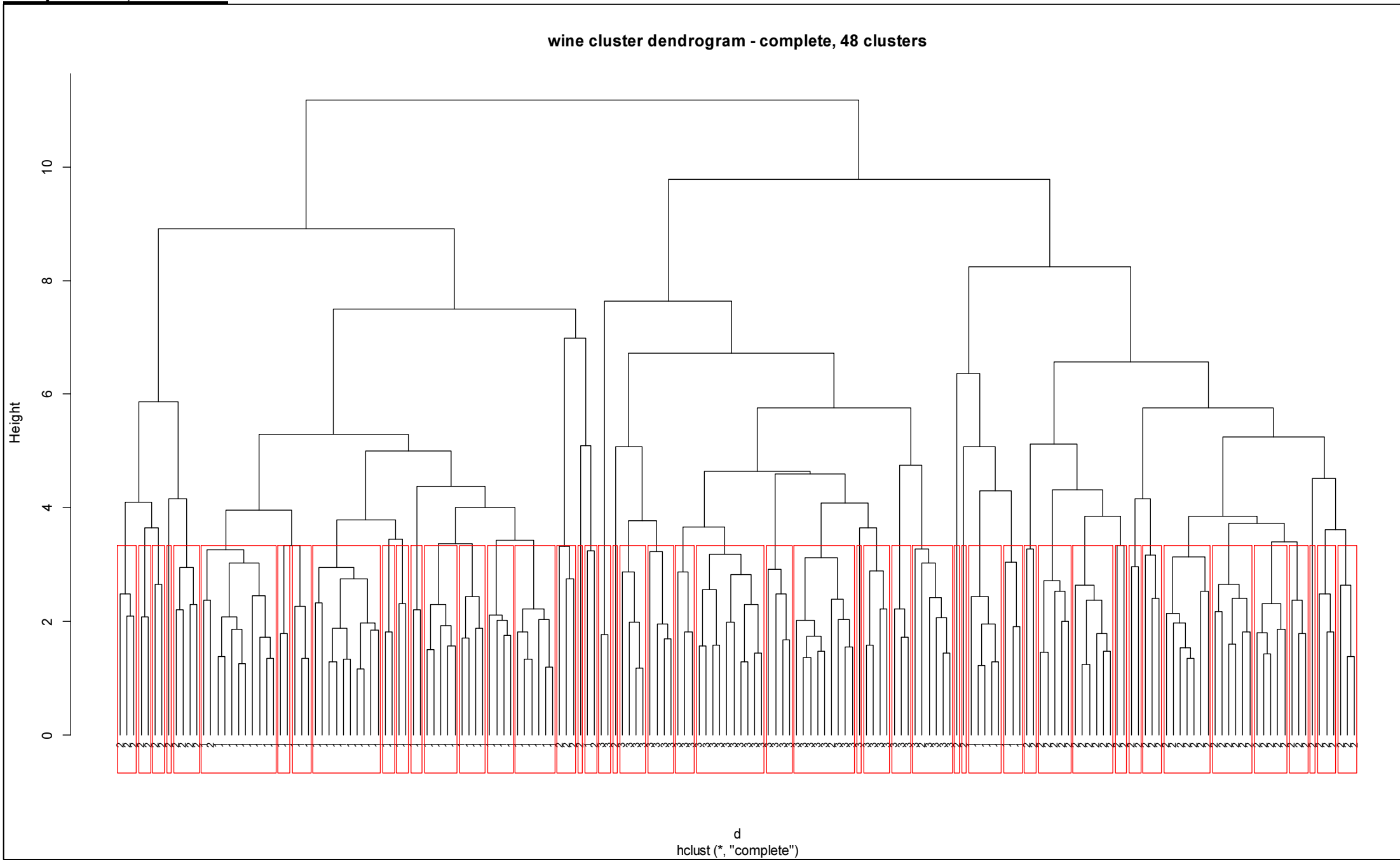
Complete link, 6 clusters



Complete link, 12 clusters



Complete link, 48 clusters



Grouping results of complete link

6 clusters:

```
> table(gComplete.6, nwinedata$class)

gComplete.6  1  2  3
           1 51  6  0
           2  8  2  0
           3  0 12  0
           4  0 48  0
           5  0  3 46
           6  0  0  2
```

12 clusters:

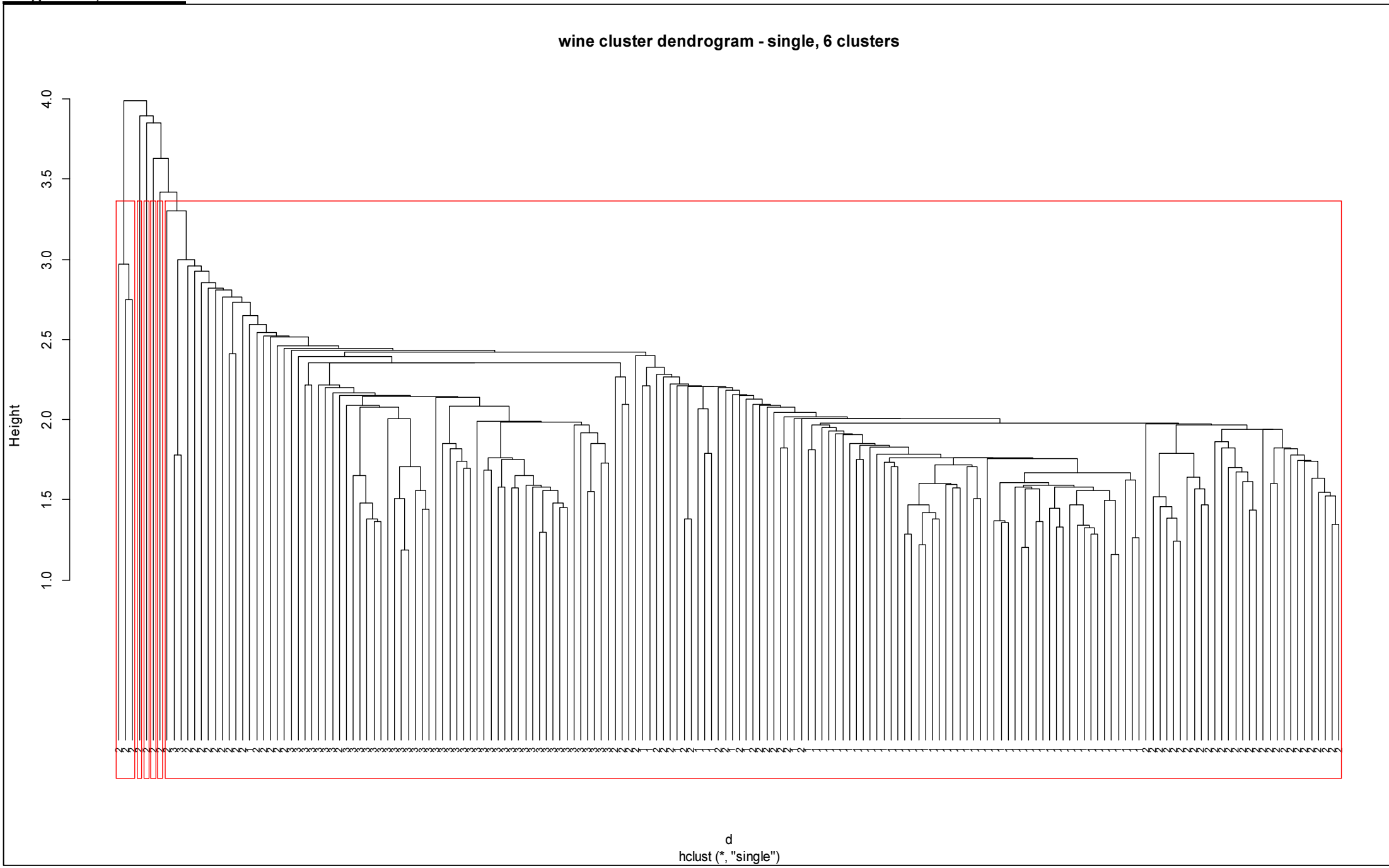
```
> table(gComplete.12, nwinedata$class)

gComplete.12  1  2  3
           1 50  1  0
           2  8  1  0
           3  1  2  0
           4  0  5  0
           5  0  7  0
           6  0 33  0
           7  0  3  0
           8  0 15  0
           9  0  2 38
           10 0  1  8
           11 0  1  0
           12 0  0  2
```

- Here, we could see how the data points are grouped in different cluster groupings in complete link.
- In 6 clusters grouping, we could see very few data points are misgrouped. Statistically, 6 + 2 + 3 = 11 data points are grouped incorrectly.
- In 12 clusters grouping, the performance is better. Statistically, 1 + 1 + 1 + 2 + 1 = 6 data points are grouped incorrectly.
- In 48 cluster groupings, the performance is improved. Only 1 + 1 + 1 = 3 data points are grouped incorrectly.
- Therefore, we could see 6 cluster, 12 cluster, and 48 Cluster groupings work very well, as all of the clusters contain correct data points, and only few data points are misgrouped. Hence, I would consider complete link would be one of the ideal approaches on this data set.

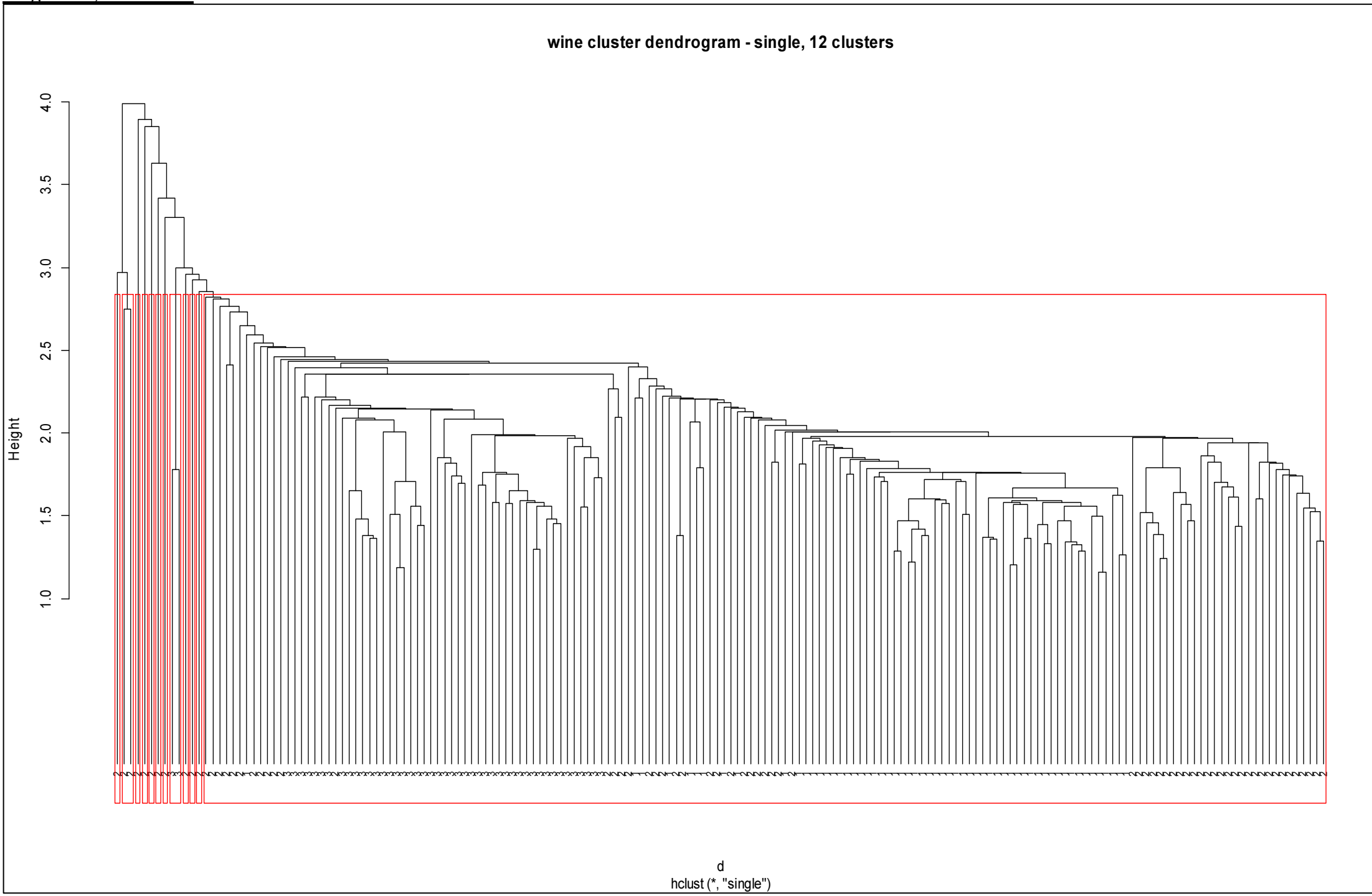
48 clusters:

```
> table(gComplete.48, nwinedata$class)

gComplete.48  1  2  3
           1 10  0  0
           2  3  0  0
           3  4  0  0
           4  4  0  0
           5  3  0  0
           6  5  0  0
           7  6  0  0
           8  2  0  0
           9  2  0  0
           10 5  0  0
           11 2  0  0
           12 10 1  0
           13 2  0  0
           14 1  1  0
           15 0  1  0
           16 0  3  0
           17 0  4  0
           18 0  3  0
           19 0  2  0
           20 0  7  0
           21 0  3  0
           22 0  2  0
           23 0  1  0
           24 0  1  0
           25 0  3  0
           26 0  6  0
           27 0  5  0
           28 0  1  8
           29 0  3  0
           30 0  6  0
           31 0  1  0
           32 0  2  0
           33 0  5  0
           34 0  3  0
           35 0  1  0
           36 0  2  0
           37 0  1  5
           38 0  1  0
           39 0  2  0
           40 0  0  4
           41 0  0  3
           42 0  0  4
           43 0  0  3
           44 0  0 10
           45 0  0  4
           46 0  0  1
           47 0  0  2
           48 0  0  4
```

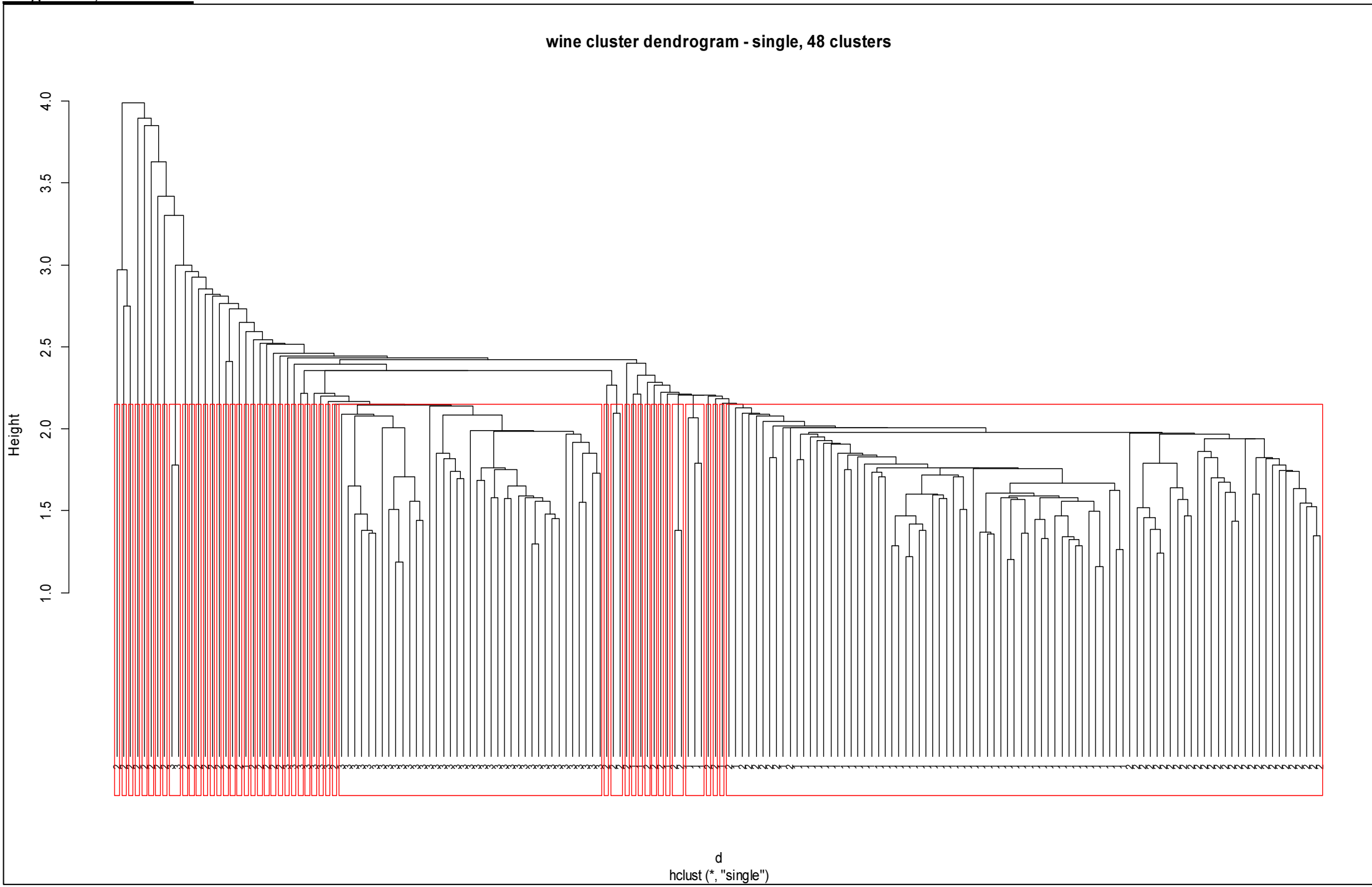
Single link, 6 clusters



Single link, 12 clusters



Single link, 48 clusters



Grouping results of single link

6 clusters:

```
> table(gSingle.6, nwinedata$class)

gSingle.6  1  2  3
        1 59 64 48
        2  0  1  0
        3  0  3  0
        4  0  1  0
        5  0  1  0
        6  0  1  0
```

12 clusters:

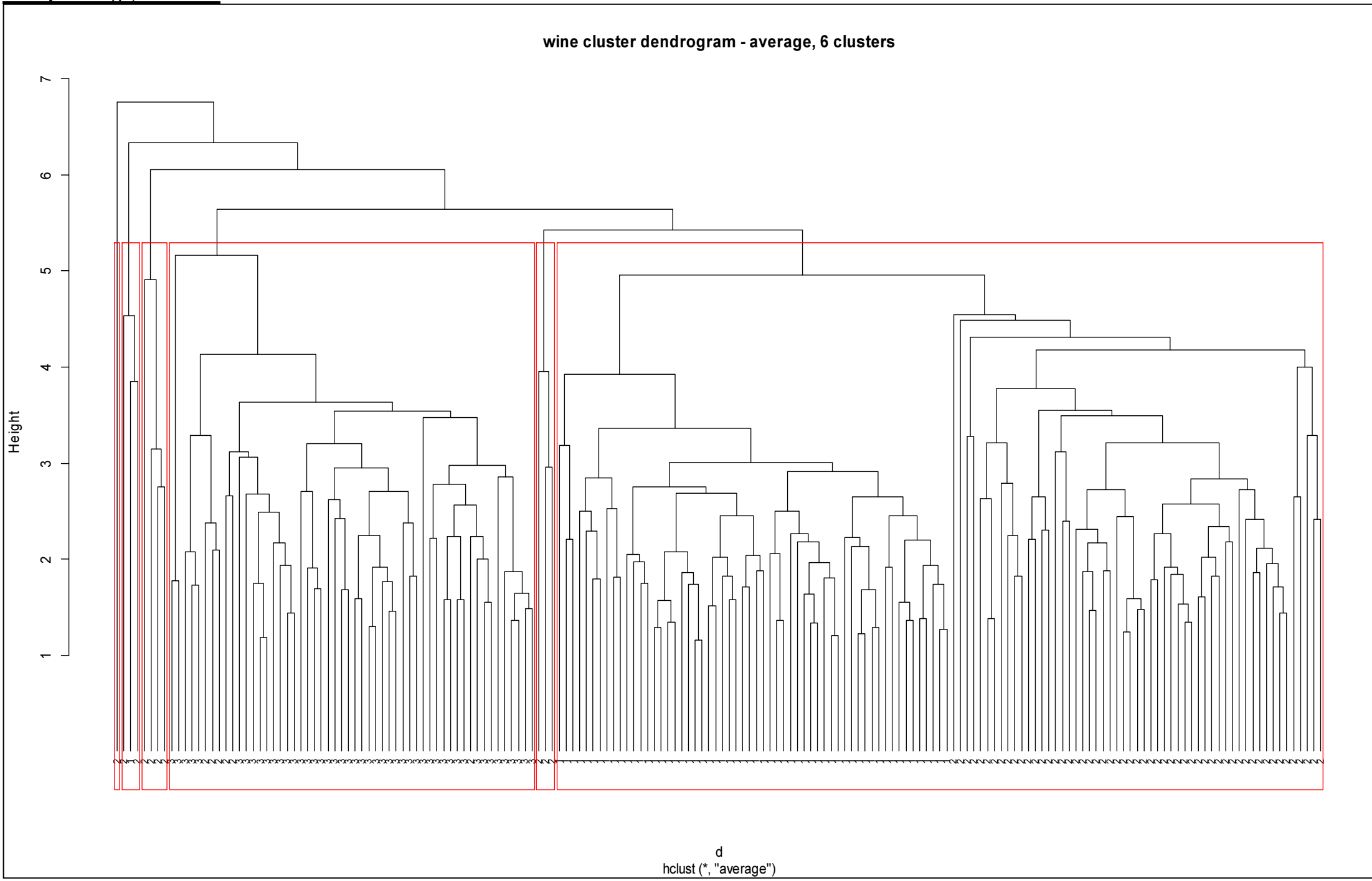
```
> table(gSingle.12, nwinedata$class)

gSingle.12  1  2  3
        1  59 60 46
        2   0  1  0
        3   0  2  0
        4   0  1  0
        5   0  1  0
        6   0  1  0
        7   0  1  0
        8   0  1  0
        9   0  1  0
        10  0  1  0
        11  0  1  0
        12  0  0  2
```

48 clusters:

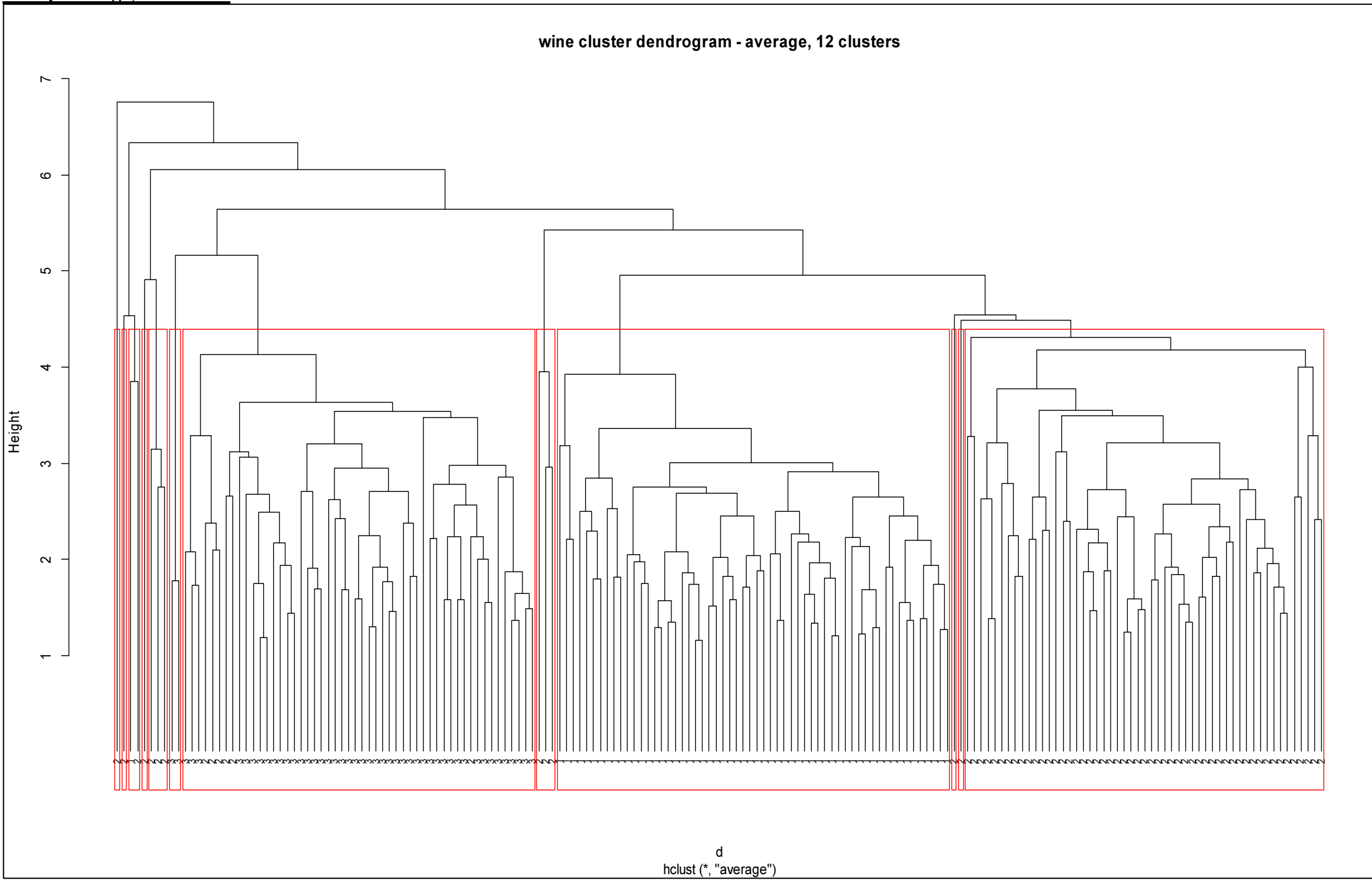
```
> table(gSingle.48, nwinedata$class)

gSingle.48  1  2  3
        1  51 37  0
        2   1  0  0
        3   1  0  0
        4   3  0  0
        5   1  0  0
        6   1  0  0
        7   1  0  0
        8   0  1  0
        9   0  2  0
        10  0  1  0
        11  0  1  0
        12  0  2  0
        13  0  1  0
        14  0  1  0
        15  0  1  0
        16  0  1  0
        17  0  1  0
        18  0  1  0
        19  0  1  0
        20  0  1  0
        21  0  1  0
        22  0  1  0
        23  0  1  0
        24  0  1  0
        25  0  1  0
        26  0  1  0
        27  0  1  0
        28  0  1  0
        29  0  1  0
        30  0  1  0
        31  0  1  0
        32  0  1  0
        33  0  1  0
        34  0  1  0
        35  0  1  0
        36  0  1  0
        37  0  1  0
        38  0  1  0
        39  0  1  0
        40  0  0  1
        41  0  0 39
        42  0  0  1
        43  0  0  1
        44  0  0  1
        45  0  0  1
        46  0  0  1
        47  0  0  2
        48  0  0  1
```

- Here, we could see how the data points are grouped in different cluster groupings in single link.
- In 6 clusters grouping, we could see many data points are misgrouped. Statistically, 59 + 48 = 107 data points are grouped incorrectly.
- In 12 clusters grouping, the performance is still bad. Statistically, 59 + 46 = 105 data points are grouped incorrectly.
- In 48 cluster groupings, the performance is slightly improved. Only 37 data points are grouped incorrectly, and most of the clusters only have 1 data point.
- Therefore, we could see 6 cluster, 12 cluster, and 48 Cluster groupings work really bad, as all of the clusters is poorly trained. Hence, I would not consider single link is a good approach in this data set.

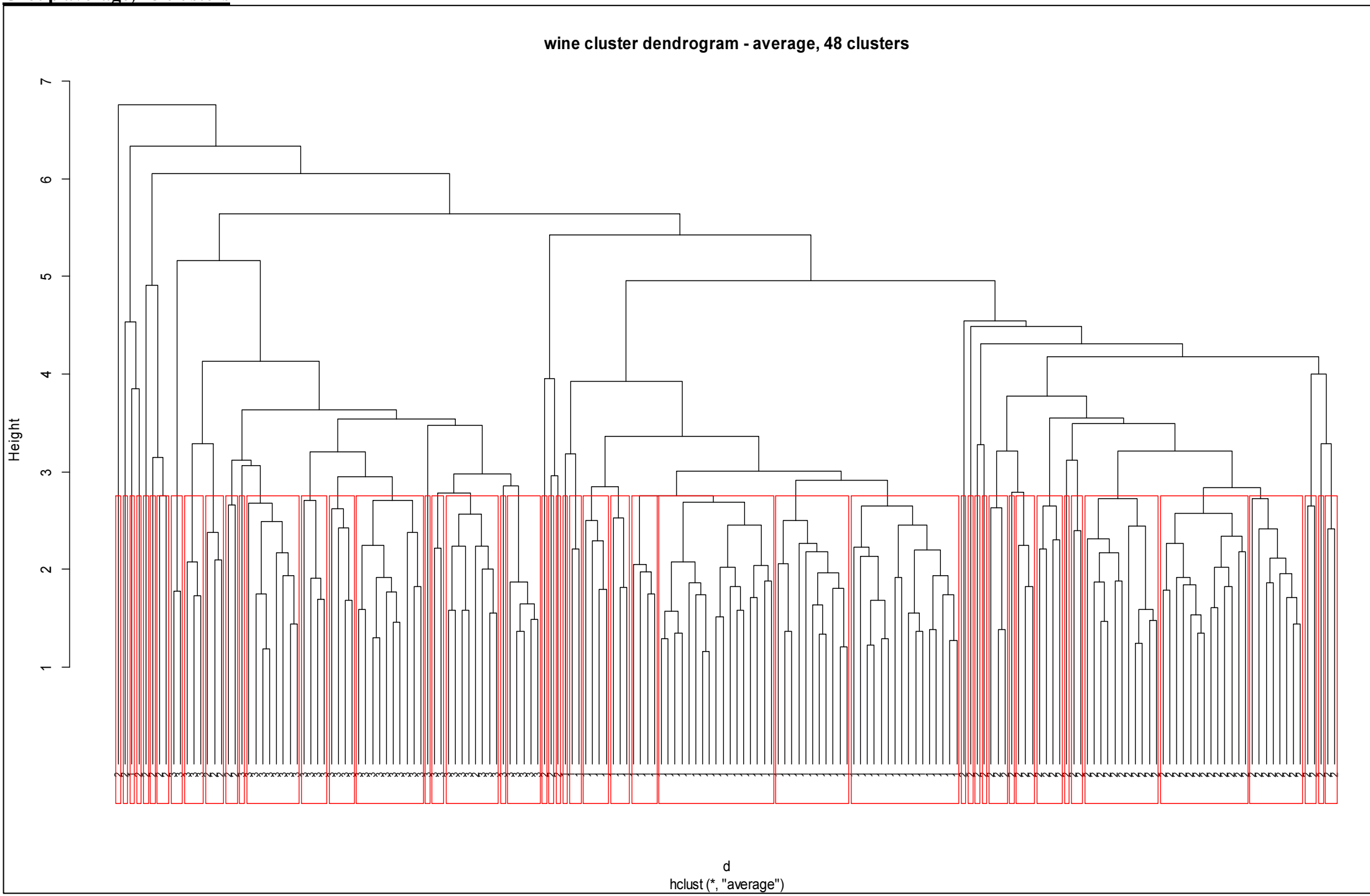
Group average, 6 clusters



Group average, 12 clusters



Group average, 48 clusters



Grouping results of group average

6 clusters:

```
> table(gAverage.6, nwinedata$class)

gAverage.6  1  2  3
         1 58 55  0
         2  1  2  0
         3  0  1  0
         4  0  6 48
         5  0  4  0
         6  0  3  0
```

12 clusters:

```
> table(gAverage.12, nwinedata$class)

gAverage.12  1  2  3
          1 58  0  0
          2  1  1  0
          3  0  1  0
          4  0  6 46
          5  0 53  0
          6  0  1  0
          7  0  3  0
          8  0  1  0
          9  0  1  0
          10 0  1  0
          11 0  3  0
          12 0  0  2
```

48 clusters:

```
> table(gAverage.48, nwinedata$class)

gAverage.48  1  2  3
          1 17  0  0
          2 11  0  0
          3  4  0  0
          4 16  0  0
          5  2  0  0
          6  3  0  0
          7  4  0  0
          8  1  0  0
          9  1  0  0
          10 0  1  0
          11 0  3  0
          12 0  4  0
          13 0  3  0
          14 0  2  0
          15 0  1  0
          16 0  1  0
          17 0 13  0
          18 0  2  0
          19 0  2  0
          20 0  1  0
          21 0  8  0
          22 0  1  0
          23 0  2  0
          24 0 11  0
          25 0  1  7
          26 0  3  0
          27 0  1  0
          28 0  1  0
          29 0  1  0
          30 0  1  0
          31 0  1  0
          32 0  1  0
          33 0  2  0
          34 0  1  0
          35 0  1  0
          36 0  1  0
          37 0  1  0
          38 0  0  1
          39 0  0  8
          40 0  0  3
          41 0  0  2
          42 0  0  5
          43 0  0  1
          44 0  0 10
          45 0  0  4
          46 0  0  1
          47 0  0  2
          48 0  0  4
```

- Here, we could see how the data points are grouped in different cluster groupings in group average.
- In 6 clusters grouping, we could see many data points are misgrouped. Statistically, 55 + 1 + 6 = 62 data points are grouped incorrectly.
- In 12 clusters grouping, the performance is greatly improved. Statistically, 1 + 6 = 7 data points are grouped incorrectly.
- In 48 cluster groupings, the performance is slightly improved. Only 1 data point is grouped incorrectly.
- Therefore, we could see 6 cluster do not perform too well. However, 12 cluster, and 48 cluster groupings work a lot better. Moreover, I would consider 12 cluster grouping is a good approach to classify this data set, as the error rates are low. For 48 cluster grouping, there are too many clusters contain only 1-3 data points, which might overfit the problem.

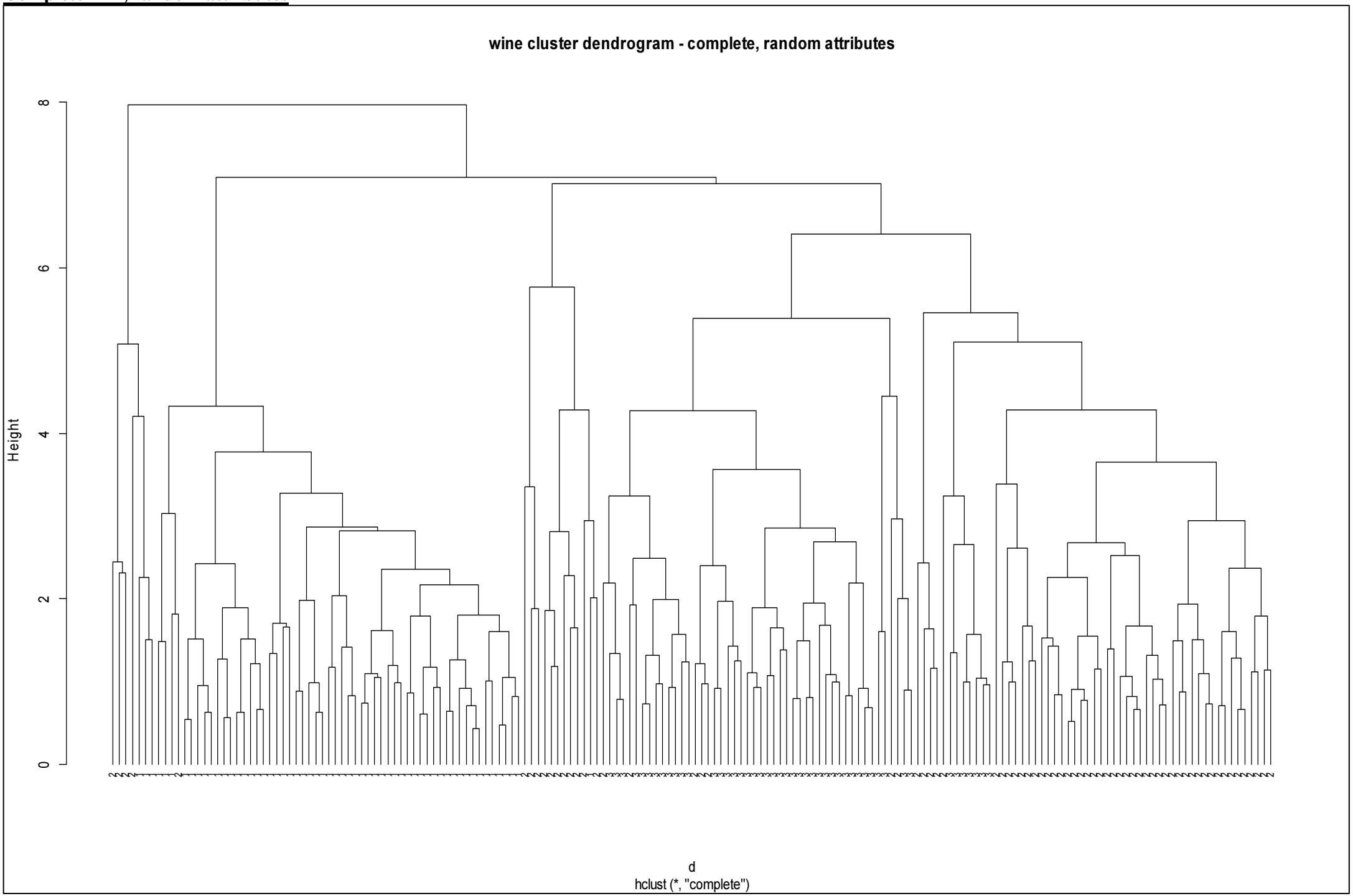
4) Select different random subsets of attributes from the data sets and re-perform hierarchical clustering. Compare the resulting hierarchical structures based on the selected attributes subsets with the original hierarchical structures.

- In this section, I randomly pick {Al, Aoa, Mag, NP, P, H} as the only attributes to re-perform hierarchical clustering.

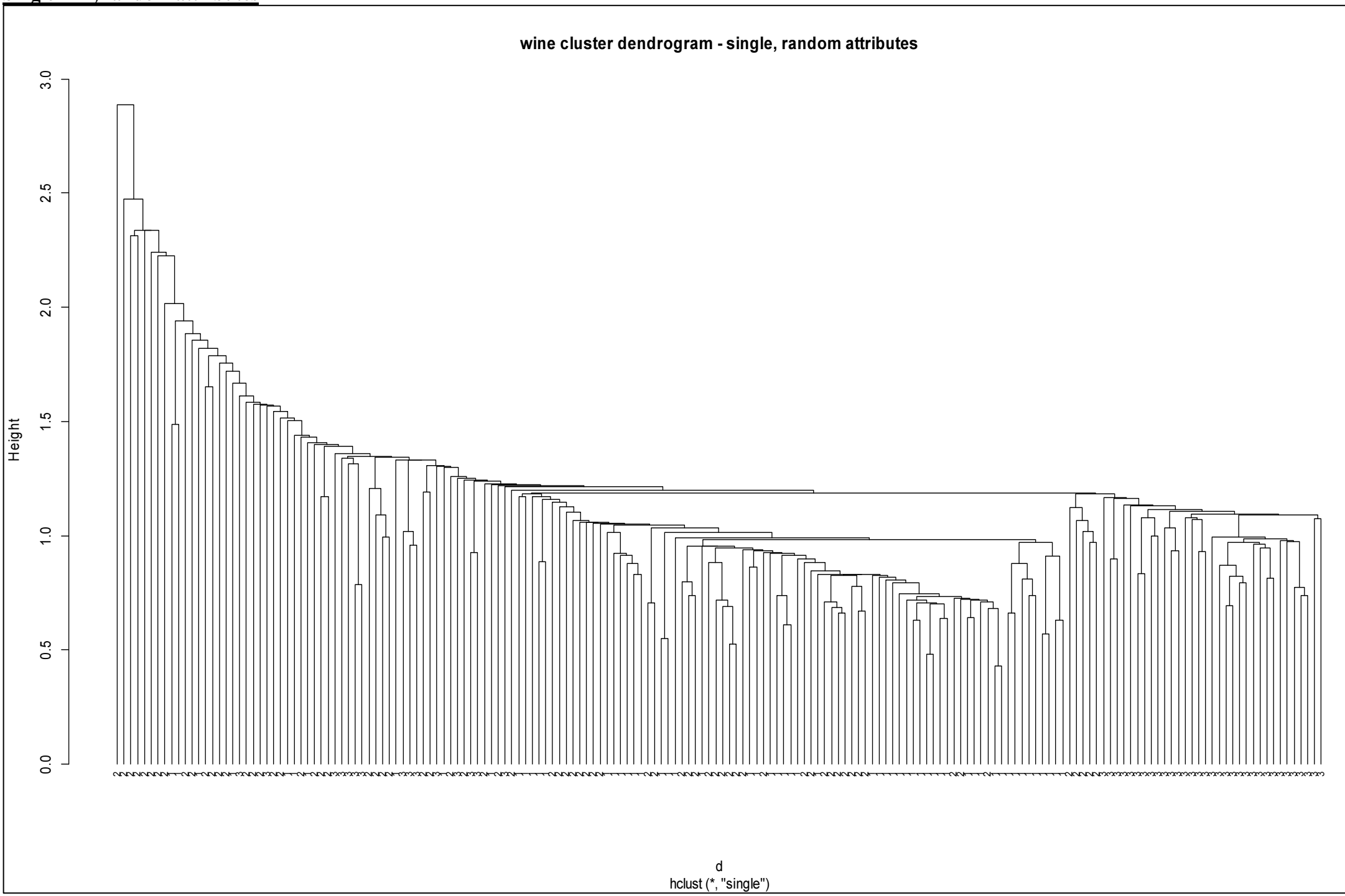
- Compared to the original hierarchical structures, the resulting hierarchical structures that are generated with the selected attributes have a very similar structure. Specifically, if we cut the dendrograms with a small number of clusters (for example, k = 6, cutree(tree, k=6)), most data points would still be grouped into the same clusters. But when we cut the dendrograms with a large number of clusters (for example, k = 48, cutree(tree, k=48)), the results of which clusters that data points will be grouped in a slightly different way.

- For dendrograms of re-performed hierarchical clustering, please refer to the following pages or the excel file.

Complete link, random attributes



Single link, random attributes



Group average, random attributes