|  |
| --- |
| [Type the company name] |
| STAT 445 Assignment 7 |
| Kun Yang 301178299 |
|  |
|  |
|  |

|  |
| --- |
|  |

**Problems 12.14 and 12.15**

* 1. ***Calculate the Euclidean distances between the cereal brands.***

(Print out the upper left-hand corner of this table corresponding to the first five cereal brands.)

**> setwd("C:\\Users\\Kun\\Desktop\\homework")**

**> data<-read.csv("Assignment 7 Cereal Data.csv")**

**> (mat<- dist(data[4:11],method="euclidian"))**

1 2 3 4 5 6

2 102.861557

3 15.636496 110.918889

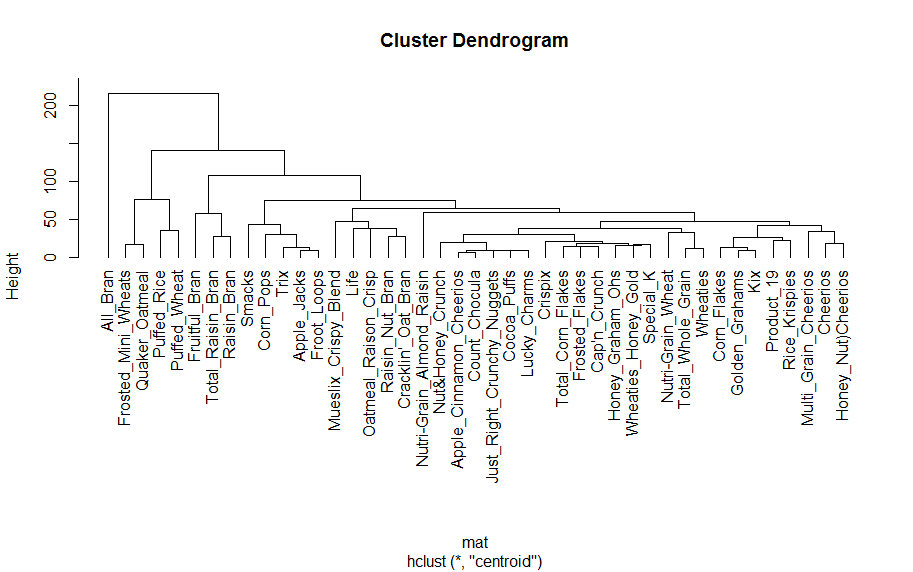
4 6.442049 104.038454 10.440307

5 99.330257 60.811183 98.635693 97.211110

* 1. ***Generate a dendrogram for each of the centroid and Ward methods for hierarchical clustering.***

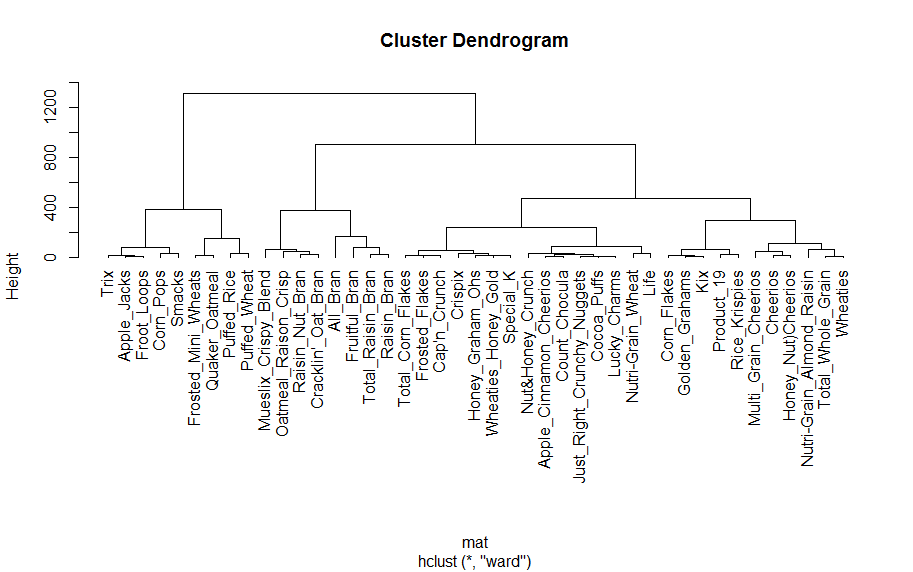
**> hc\_c=hclust(mat, method='centroid')**

**> plot(hc\_c,hang=-1,labels=data[,2])**



**> hc\_c=hclust(mat, method="ward")**

**> plot(hc\_c,hang=-1,labels=data[,2])**



* 1. ***Comment on any notable differences between the above two dendrograms.***

In ward’s method, the larger clusters will tend to be judged as further apart and the smaller clusters will tend to be amalgamated. Therefore, we obtain more evenly sized clusters with ward’s method than with the centroid. We also find an inversion in the centroid.

* 1. ***Use the k-means approach to generate K = 3, 4, and 5 clusters.***

**> # 3 cluster solution**

**> (fit3<-kmeans(data[4:11], 3))**

K-means clustering with 3 clusters of sizes 7, 9, 27

Cluster means:

Calories Protein Fat Sodium Fibre Carbohydrates Sugar Potassium

1 117.14286 3.142857 1.4285714 198.71429 4.642857 12.50000 10.142857 205.71429

2 94.44444 2.111111 0.5555556 57.88889 1.188889 10.55556 8.222222 46.66667

3 110.00000 2.407407 1.0000000 219.51852 1.129630 15.94444 6.740741 65.55556

Clustering vector:

[1] 3 3 3 3 3 3 3 3 3 3 1 3 1 3 2 3 3 1 2 3 2 1 3 2 3 2 1 3 1 3 3 3 3 1 3 2 3 3 3 3 2 2 2

Within cluster sum of squares by cluster:

[1] 49514.00 42404.57 83043.07

(between\_SS / total\_SS = 63.7 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" "size" "iter"

[9] "ifault"

**> # 4 cluster solution**

**> (fit4<-kmeans(data[4:11], 4))**

K-means clustering with 4 clusters of sizes 17, 16, 4, 6

Cluster means:

Calories Protein Fat Sodium Fibre Carbohydrates Sugar Potassium

1 111.17647 2.352941 1.294118 161.11765 1.558824 13.38235 9.058824 83.23529

2 111.25000 2.437500 0.875000 248.75000 0.968750 17.21875 5.937500 56.25000

3 112.50000 3.250000 0.750000 241.50000 5.750000 12.50000 10.750000 245.00000

4 86.66667 2.333333 0.500000 25.66667 1.450000 10.00000 5.833333 55.83333

Clustering vector:

[1] 1 2 1 1 2 2 2 1 2 1 1 2 3 1 1 1 2 3 1 2 4 1 2 1 2 4 3 1 1 1 2 1 2 3 2 4 2 2 2 1 4 4 4

Within cluster sum of squares by cluster:

[1] 47513.35 34191.34 15272.00 20799.84

(between\_SS / total\_SS = 75.5 %)

**> # 5 cluster solution**

**> (fit5<-kmeans(data[4:11], 5))**

K-means clustering with 5 clusters of sizes 6, 9, 14, 10, 4

Cluster means:

Calories Protein Fat Sodium Fibre Carbohydrates Sugar Potassium

1 86.66667 2.333333 0.500000 25.66667 1.4500000 10.00000 5.833333 55.83333

2 114.44444 3.111111 1.666667 172.11111 2.7777778 15.00000 6.555556 123.88889

3 112.14286 1.571429 1.071429 177.07143 0.5357143 13.46429 10.500000 45.00000

4 107.00000 2.900000 0.600000 269.10000 0.9500000 17.95000 4.300000 57.00000

5 112.50000 3.250000 0.750000 241.50000 5.7500000 12.50000 10.750000 245.00000

Clustering vector:

[1] 3 4 3 3 4 4 4 3 4 2 2 3 5 2 3 2 3 5 3 4 1 2 4 3 3 1 5 3 2 3 2 2 4 5 4 1 4 3 3 2 1 1 1

Within cluster sum of squares by cluster:

[1] 20799.84 19557.67 19064.96 13643.75 15272.00

(between\_SS / total\_SS = 81.7 %)

* 1. ***Compare the results of this non-hierarchical approach to each of the two hierarchical ones. Which hierarchical method seems to give more similar results to the k-means method? Is this generally to be expected or not?***

Based on above analysis, I think the result from the ward method is more similar to the k-means method.

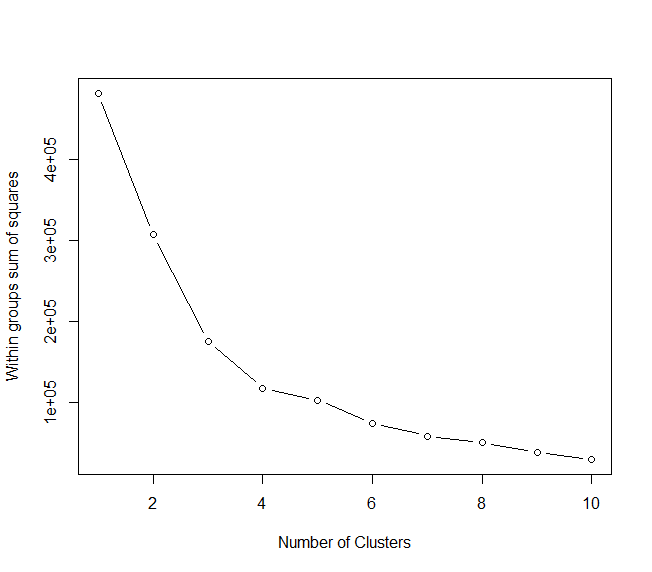
We can see that the 43 observations are roughly distributed into evenly sized clusters. And I think this trend is pretty general, but not always correct.

* 1. ***Sketch the portion of the scree plot for the within-cluster sum of squares for K = 1 to 10.***

**> wss <- (nrow(data[4:11])-1)\*sum(apply(data[4:11],2,var))**

**> for (i in 2:10) wss[i] <- sum(kmeans(data[4:11],centers=i)$withinss)**

**> plot(1:10, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")**



* 1. ***Does this scree plot give any clear indication of the appropriate number of clusters to use?***

From the scree plot, we can see that the elbow is on k=4 or k=5. Therefore, I would like to choose 4 clusters to use.

* 1. ***The function, “kmeans” will not let you construct a set of K = 43 clusters, but you can easily figure out what it would have to be. What would the value of this within-cluster sum of squares be for K = 43?***

In this question, the within-cluster sum of squares decrease as the number of clusters increase. Then, when K=43, the means of each cluster will equal to the individual data itself. Thus, the within-cluster sum of squares will equal to 0.

* 1. ***You were not asked to standardize the variables before proceeding. Might there be some advantage to doing so?***

There are some advantages. For example, in this problem, the variables (Calories, Protein, Fat, etc.) are measured by different units. Standardize the variables before proceeding can gets rid of the differences in scale.