Hierarchical Clusrering Model and K-means Algorithms

Jyoti Phogat

1/12/2022

Introduction

The purpose of this assignment is to use Hierarchical Clustering and compared it with K-means algorithm (non-hierarchical).

The dataset we are going to utilize is the Cereals.csv which includes nutritional information, store display, and consumer ratings for 77 breakfast cereals. The variables in this dataset are the following:

- name: Name of cereal
- mfr: Manufacturer of cereal, which is classified as
  - A = American Home Food Products;
  - G = General Mills
  - K = Kelloggs
  - N = Nabisco
  - P = Post
  - Q = Quaker Oats
  - R = Ralston Purina
- type:
  - cold
  - hot
- calories: calories per serving
- protein: grams of protein
- fat: grams of fat
- sodium: milligrams of sodium
- fiber: grams of dietary fiber
- carbo: grams of complex carbohydrates
- sugars: grams of sugars
- potass: milligrams of potassium
- vitamins: vitamins and minerals - 0, 25, or 100, indicating the typical percentage of FDA recommended
- shelf: display shelf (1, 2, or 3, counting from the floor)
- weight: weight in ounces of one serving
- cups: number of cups in one serving
- rating: a rating of the cereals (Possibly from Consumer Reports?)

Contents

Data Exploration

```r
# Load the libraries needed
library(readr)
library(factoextra)
library(cluster)
library(caret)

# Import the dataset
cereals <- read_csv("cereals.csv")

# To see the first 6 rows
head(cereals)
```

```
# A tibble: 6 x 16
  name  mfr   type  calories protein   fat sodium fiber carbo sugars potass
  <chr> <chr> <chr>    <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 100%~ N     C           70       4     1    130  10      5       6    280
2 100%~ Q     C          120       3     5     15   2      8       8    135
3 All-~ K     C           70       4     1    260   9      7       5    320
4 All-~ K     C           50       4     0    140  14      8       0    330
5 Almo~ R     C          110       2     2    200   1     14       8     NA
6 Appl~ G     C          110       2     2    180   1.5  10.5     10     70
# ... with 5 more variables: vitamins <dbl>, shelf <dbl>, weight <dbl>,
#   cups <dbl>, rating <dbl>
```

```r
# To see the last 6 rows
tail(cereals)
```

```
# A tibble: 6 x 16
  name  mfr   type  calories protein   fat sodium fiber carbo sugars potass
  <chr> <chr> <chr>    <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 Tota~ G     C          100       3     1    200     3    16      3    110
2 Trip~ G     C          110       2     1    250     0    21      3     60
3 Trix  G     C          110       1     1    140     0    13     12     25
4 Whea~ R     C          100       3     1    230     3    17      3    115
5 Whea~ G     C          100       3     1    200     3    17      3    110
6 Whea~ G     C          110       2     1    200     1    16      8     60
# ... with 5 more variables: vitamins <dbl>, shelf <dbl>, weight <dbl>,
#   cups <dbl>, rating <dbl>
```

Here we can see that the first and last 6 rows of the data frame are similar among its data points.

```r
# Show the dimentions of the dataset
dim(cereals)
```

```
[1] 77 16
```

The Cereals dataset has 77 observations and 16 variables.

```r
# Descriptive statistics
summary(cereals)
```

```
     name               mfr                type             calories
 Length:77          Length:77          Length:77          Min.   : 50.0
 Class :character   Class :character   Class :character   1st Qu.:100.0
 Mode  :character   Mode  :character   Mode  :character   Median :110.0
                                                          Mean   :106.9
                                                          3rd Qu.:110.0
                                                          Max.   :160.0

    protein           fat             sodium          fiber
 Min.   :1.000   Min.   :0.000   Min.   :  0.0   Min.   : 0.000
 1st Qu.:2.000   1st Qu.:0.000   1st Qu.:130.0   1st Qu.: 1.000
 Median :3.000   Median :1.000   Median :180.0   Median : 2.000
 Mean   :2.545   Mean   :1.013   Mean   :159.7   Mean   : 2.152
 3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:210.0   3rd Qu.: 3.000
 Max.   :6.000   Max.   :5.000   Max.   :320.0   Max.   :14.000

     carbo          sugars           potass          vitamins
 Min.   : 5.0   Min.   : 0.000   Min.   : 15.00   Min.   :  0.00
 1st Qu.:12.0   1st Qu.: 3.000   1st Qu.: 42.50   1st Qu.: 25.00
 Median :14.5   Median : 7.000   Median : 90.00   Median : 25.00
 Mean   :14.8   Mean   : 7.026   Mean   : 98.67   Mean   : 28.25
 3rd Qu.:17.0   3rd Qu.:11.000   3rd Qu.:120.00   3rd Qu.: 25.00
 Max.   :23.0   Max.   :15.000   Max.   :330.00   Max.   :100.00
 NA’s   :1      NA’s   :1        NA’s   :2
     shelf          weight          cups           rating
 Min.   :1.000   Min.   :0.50   Min.   :0.250   Min.   :18.04
 1st Qu.:1.000   1st Qu.:1.00   1st Qu.:0.670   1st Qu.:33.17
 Median :2.000   Median :1.00   Median :0.750   Median :40.40
 Mean   :2.208   Mean   :1.03   Mean   :0.821   Mean   :42.67
 3rd Qu.:3.000   3rd Qu.:1.00   3rd Qu.:1.000   3rd Qu.:50.83
 Max.   :3.000   Max.   :1.50   Max.   :1.500   Max.   :93.70
```

It determines that variables like carbo, sugars, and potassium have a few missing data. We can also confirm that the cereal’s name, cereal’s manufacturer, and cereal’s type are qualitative variables. The other 13 variables are quantitative.

Data Preparation

First, transform the cereal name into a row name.

```
# Transform the cereal name into a row name
cereals1 <- data.frame(cereals, row.names = 'name')

# Show the number of rows and columns of the dataset
dim(cereals1)
```

```
[1] 77 15
```

It shows us the column name of Cereal's dataset changed from column to row name, for this reason, we have 15 variables now, instead of 16.

```
# Select only the numerical variables. Remove also Shelf variable because it's categorical.
num_cereals <- cereals1[, c(3:11,13:15)]

# Normalize the data using scale function (z-score)
num_cereals <- scale(num_cereals)

#To see the first 6 rows and the four first variables
head(num_cereals)[1:6, 1:4]
```

```
                            calories    protein         fat     sodium
100%_Bran                 -1.8929836  1.3286071 -0.01290349 -0.3539844
100%_Natural_Bran          0.6732089  0.4151897  3.96137277 -1.7257708
All-Bran                  -1.8929836  1.3286071 -0.01290349  1.1967306
All-Bran_with_Extra_Fiber -2.9194605  1.3286071 -1.00647256 -0.2346986
Almond_Delight             0.1599704 -0.4982277  0.98066557  0.4810160
Apple_Cinnamon_Cheerios    0.1599704 -0.4982277  0.98066557  0.2424445
```

```
# Remove missing values
num_cereals <- na.omit(num_cereals)

# New dimention of the dataset
dim(num_cereals)
```

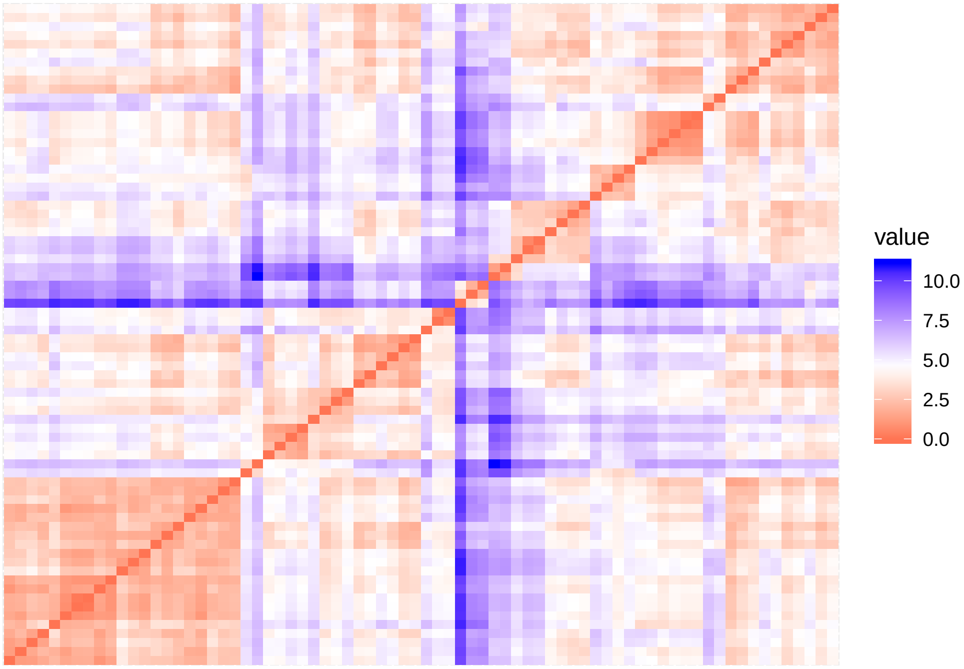
```
[1] 74 12
```

It removed the missing values from 77 variables to 74, and we have only 12 variables which are the quantitative variables.

```
# Compute the distances. Remember Euclidean distance is used by default.
distance <- get_dist(num_cereals)

#To see the first 6 rows
head(distance)
```

```
[1] 7.546309 1.904228 3.408981 6.651946 7.462129 7.372052
```

```r
# Let's visualize our distances. The fviz_dist() function visualizes a distance matrix
fviz_dist(distance, show_labels = FALSE)
```



This graph is a distance matrix. As we can see, the diagonal values are zeros (dark orange) because it is showing the distance between any point against itself. The purple and blue represent the furthest distance between any pair of observations.

Question 1

1. Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements. Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward. Choose the best method.

Hierarchical Clustering

Clustering is an unsupervised data mining technique to group data points based on their similarities. It identifies a pattern on the data, which could be used for decision making. The following figure shows the two categories of clustering:

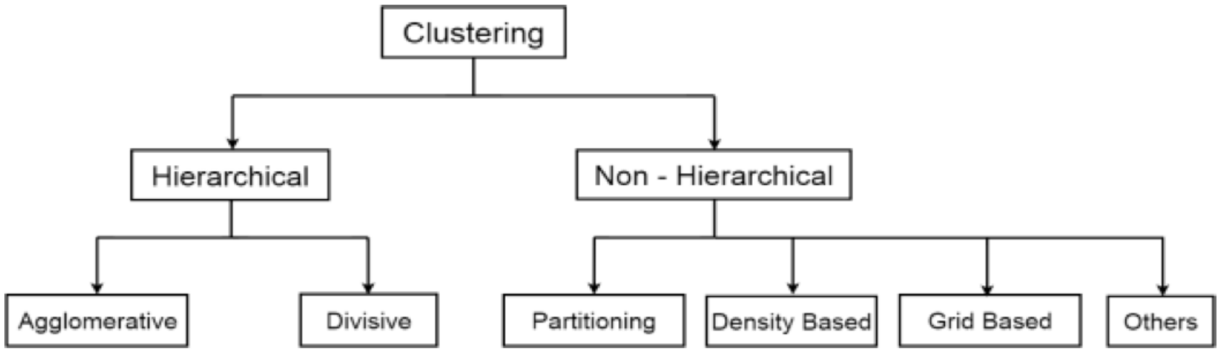


Figure 1: Categorization of clustering algorithms. Retrieved from Kaya & Sahin (2017)

Non-hierarchical models make the clusters directly, it needs a hyperparameter (k) to be able to run, and all its data points are grouped based on the centers. Based on the image above, the K-means model is classified under the partitioning approach. Hierarchical models, on the other hand, are identified for forming clusters gradually.

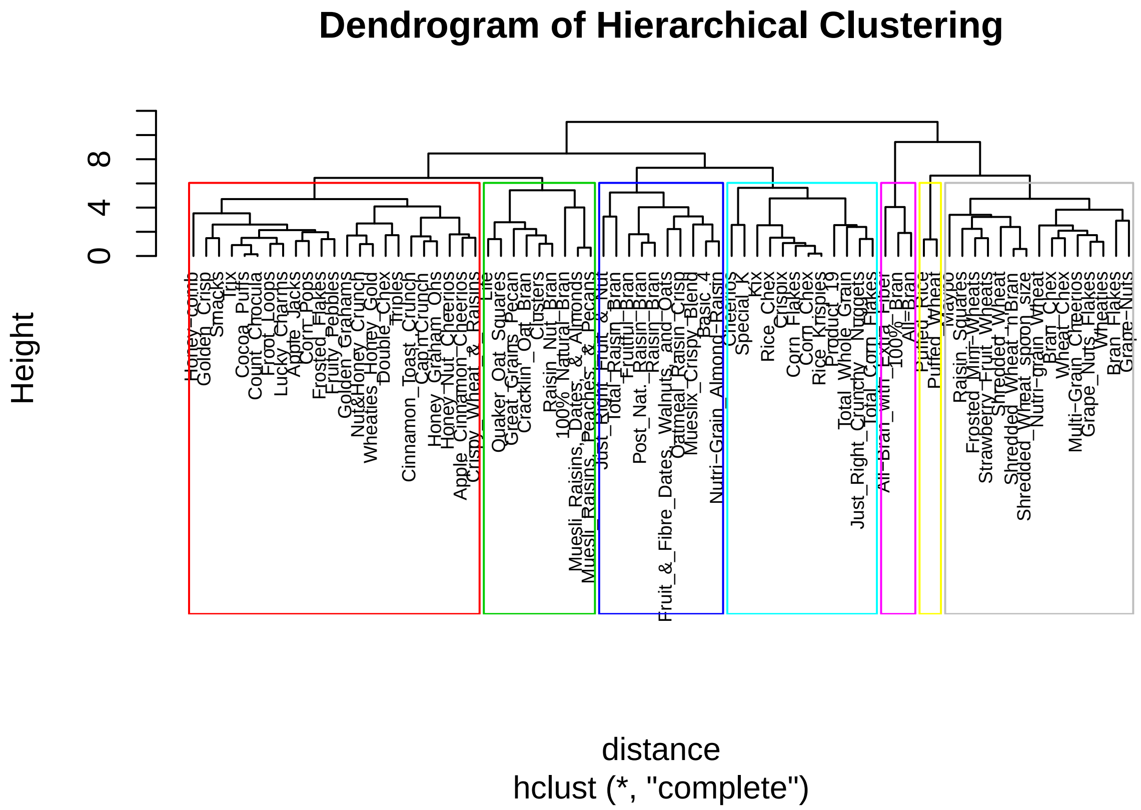
As we learned during module 8, hierarchical models use two main approaches to group the data:

+ **Divisive** (called Top-bottom or DIANA)

+ **Agglomerative** approach (called Bottom-up or AGNES)

Now, we are going to run the hierarchical model using the hclust function.

```
# To maintain same values
set.seed(123)

# Hierarchical clustering using Complete Linkage
hierarchical_cluster <- hclust(distance, method = "complete" )

# Plot the obtained dendrogram
plot(hierarchical_cluster, cex = 0.6, hang = -1,
     main = "Dendrogram of Hierarchical Clustering")
rect.hclust(hierarchical_cluster, k = 7, border = 2:10)
```



The dendrogram helps us to define the number of clusters needed to classify this dataset. Here we can see that 7 clusters are a good number to group similar data.

AGNES Method

AGNES is the bottom-up approach, and now we are going to use this method to compare different linkages.

```
# To maintain same values
set.seed(123)

# Compute with AGNES and with different linkage methods
hc_single <- agnes(distance, method = "single")
hc_complete <- agnes(distance, method = "complete")
hc_average <- agnes(distance, method = "average")
hc_ward <- agnes(distance, method = "ward")

# Compare AGNES (agglomerative) coefficients
print(hc_single$ac)
```

```
[1] 0.6091225
```

```
print(hc_complete$ac)
```

```
[1] 0.8508357
```

```
print(hc_average$ac)
```

```
[1] 0.7888569
```

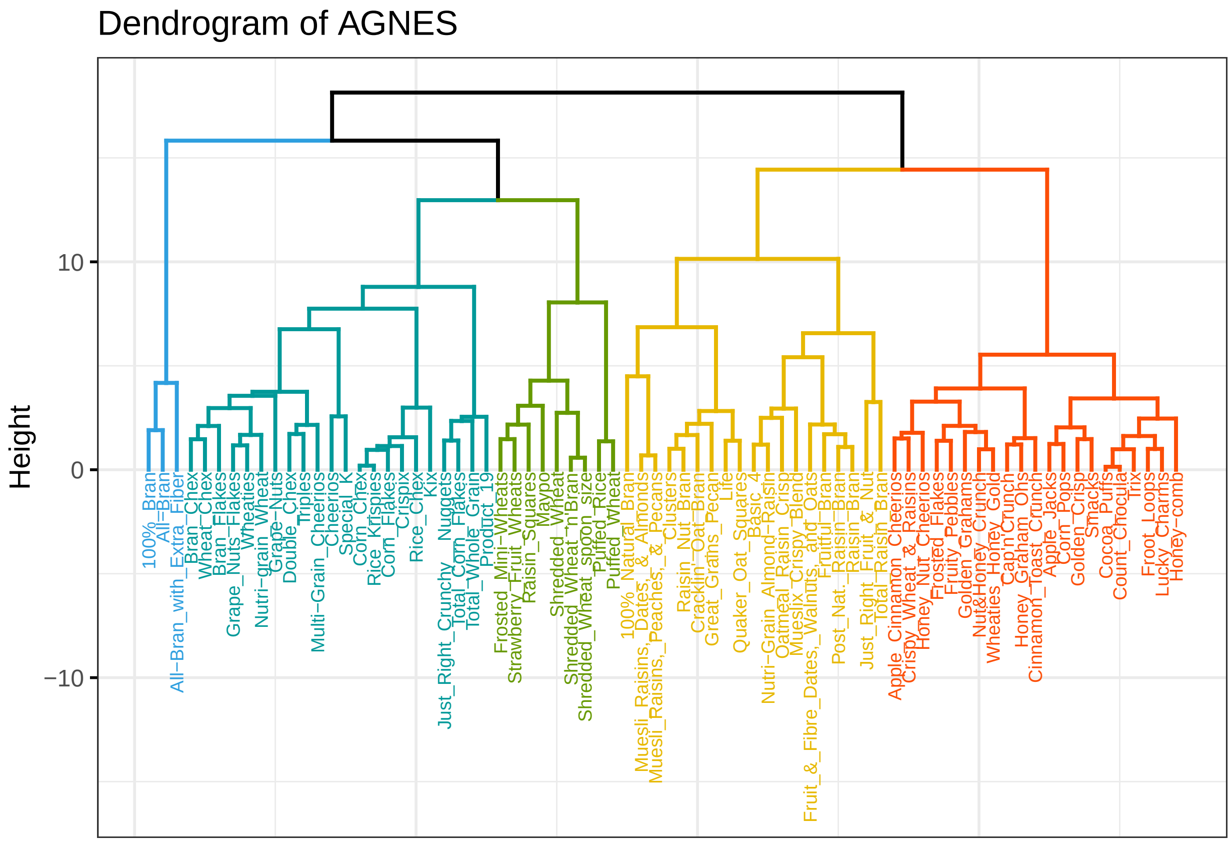
```
print(hc_ward$ac)
```

```
[1] 0.9088247
```

These outputs affirm that the best agglomerative (AGNES) linkage to use is the Ward linkage, which gives 90.50% accuracy.

Let's now plot the dendrogram.

```
set.seed(123)
# Plot the Dendrogram of AGNES
fviz_dend(hc_ward, k = 5,
          main = "Dendrogram of AGNES",
          cex = 0.5,
          k_colors = c("#2E9FDF", "#009999", "#669900", "#E7B800", "#FC4E07"),
          color_labels_by_k = TRUE,
          labels_track_height = 16,
          ggtheme = theme_bw())
```



Utilizing the Ward linkage, 5 clusters seem to be a good number to group the data.

Question 2

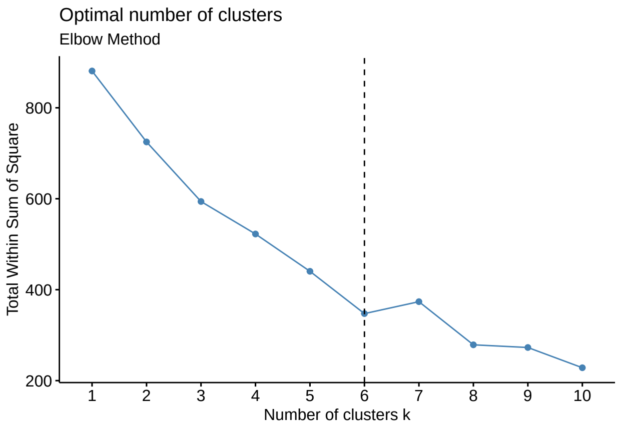
2. Comment on differences between hierarchical Clustering and K-means.

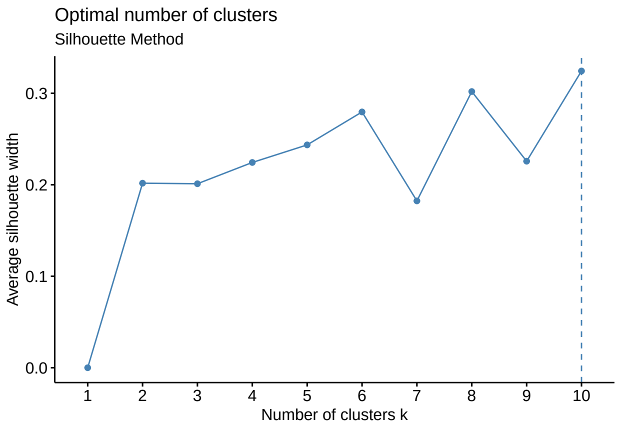
In order to be able to see the differences, let's run the K-means model and compare it with the AGNES technique.

K-means

Now, we are going to run the K-means model. But first, we will go to use methods like the Elbow and Silhouette to determine the optimal K to use.

The following two plots are the Elbow Method and Silhouette Method determining the optimal number of clusters:





The Elbow Method determines the optimal k = 6, and the Silhouette Method defines it as k= 10. For our purposes, we will use k = 6.

Now, let's run the K-means model.

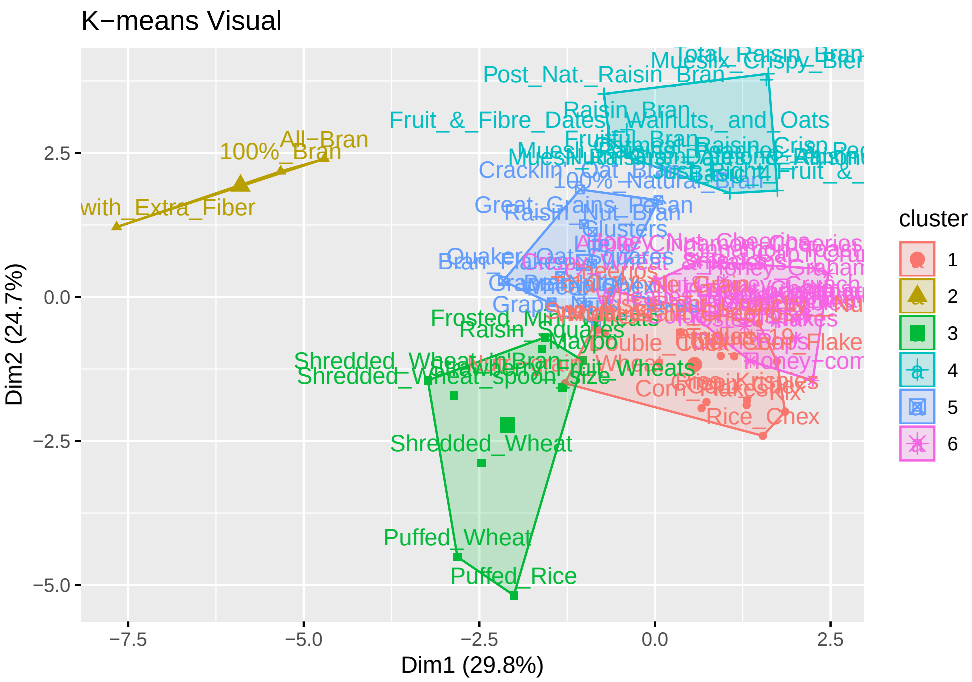
```
# To maintain same values
set.seed(123)

# To run the kmeans model
my_kmeans <- kmeans(num_cereals, centers = 6, nstart = 30)

# Let's run table function to identify to which cluster those cluster's size belong to.
table(my_kmeans$cluster)
```

```

 1  2  3  4  5  6
17  3  9 12 12 21
```

Here we can identify the number of data points that belong to each cluster.

The following visual shows the 6 clusters developed for the K-means model.

```
set.seed(123)
# To visualize the output
fviz_cluster(my_kmeans, data = num_cereals, main = "K-means Visual")
```

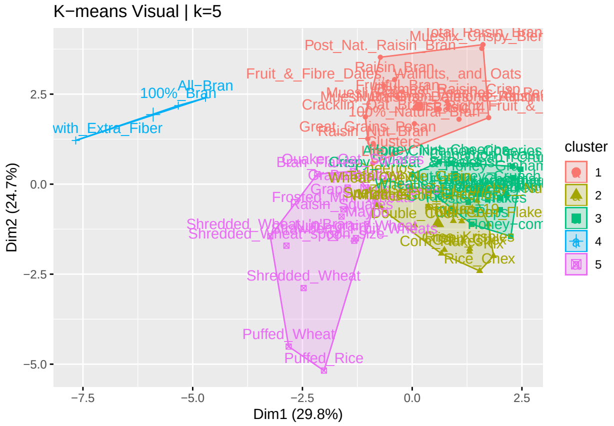


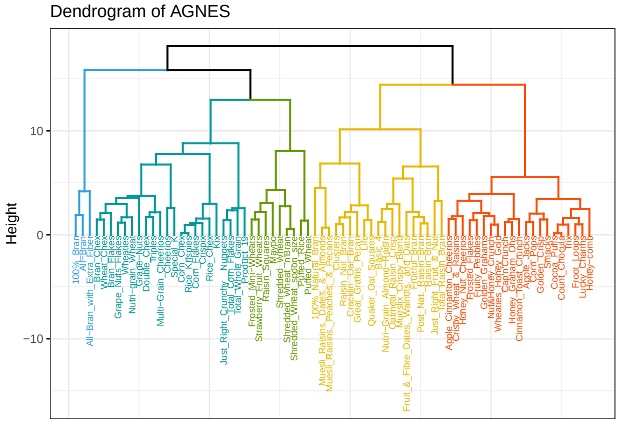
It is the K-means plot utilizing K=6. I, however, think that many clusters could be reduced to get a more defined cluster. For example, the K-means visual shows crowded data points on cluster 3 and cluster 6 that are hard to differentiate.

So, going back to the main question, what are the differences between hierarchical Clustering and K-means?

As we learned, K-means is a non-hierarchical model, and it needs to predetermine the optimal k first to be able to run. The hierarchical model, on the other hand, does not need any k. We can decide at the end the number of clusters by analyzing the dendrogram.

The following two visuals show K-means and AGNES together to compare them easier. Here we changed the number of clusters in K-means from 6 to 5 to have an equal comparison.





Following are two summary tables which show the number of observations per cluster in K-means and AGNES methods:

```
K_means_table
 1  2  3  4  5
18 17 21  3 15
```

```
Agmes_clusters_5
 1  2  3  4  5
 3 19 21 22  9
```

As we can see, after running the hierarchical method using the AGNES technique, we determined 5 as the optimal number of clusters. Regarding K-means, even though the elbow method said the optimal number of clusters is 6, we changed it to 5 to compare both methods easier.

By looking at both plots, the clusters were classified similarly. There are some data points in different clusters, but most of the time they coincide with each other. For example:

- Both blue clusters match exactly.
- Cluster 5 (Pink) in K-means most of the time match with a yellow cluster from AGNES method, except by "Quater_Out_Square" which was classified on the turquoise cluster in AGNES.
- Cluster 1 (red) in K-means match very well with the turquoise cluster in AGNES. We can see the types of cereals are almost the same in both clusters. Nevertheless, the turquoise cluster in AGNES has two extra data points, the red cluster in K-means has 18 observations, and the turquoise cluster in AGNES has 20.
- Cluster 2 (yellow) in K-means has 17 data points, and it coincides with cluster red. However, cluster red has only 11 data points, and for that reason, some observations are also in the yellow and green clusters from AGNES. For example, Crispix is part of the yellow cluster, and Multi-Grain Cheerios on the green cluster.

Question 3

3. How many clusters would you choose?

Regarding the K-means, as mentioned in previous sections, I would choose fewer clusters than given by the Elbow Method and Silhouette Method. I would say that 5 clusters would be enough to classify the data and make predictions.

Regarding the AGNES method and based on its dendrogram, 5 clusters are a good number to classify the cereal's dataset.

Question 4

4. Comment on the structure of the clusters and their stability. Hint: To check stability, partition the data, and see how well clusters formed based on one part apply to the other part.

Data Split

We will partition the dataset equally into two groups: partition A and partition B. To do the split, we are going to use the protein column.

```
# To get the same random variables
set.seed(123)

#Divide data into training and validation using protein
```

```
datasplit <-createDataPartition(num_cereals[,’protein’], p=0.5, list=FALSE)

#Now, let’s to create a dataframe with 50% for training sets and 50% validation sets.
partitionA <-num_cereals [datasplit, ]
partitionB  <-num_cereals [-datasplit, ]

#Now, let’s do summary function to get some descritive statistics and see how
# well it has been distributed.
summary(partitionA[,’protein’])
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-1.41165 -0.49823 -0.04152  0.03059  0.41519  3.15544
```

```
summary(partitionB[,’protein’])
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-1.41165 -0.49823 -0.04152 -0.09226  0.41519  1.32861
```

These outputs allow us to see that data was almost equally distributed, we can see that the range of Partition A is larger than the second subset.

```
# Compute the distances. Remember Euclidean distance is used by default.
distance_pA <- get_dist(partitionA)
```

In the following section, we are going to run the AGNES method for Partition A subset and compare it with the complete dataset.

AGNES method Partition A

Here we are going to run AGNES to do the clustering on partition A dataset.

```
# To maintain same values
set.seed(123)

# Compute with AGNES and with different linkage methods
hc_single_pA <- agnes(distance_pA, method = "single")
hc_complete_pA <- agnes(distance_pA, method = "complete")
hc_average_pA <- agnes(distance_pA, method = "average")
hc_ward_pA <- agnes(distance_pA, method = "ward")

# Compare AGNES (agglomerative) coefficients
print(hc_single_pA$ac)
```

```
[1] 0.7171847
```

```
print(hc_complete_pA$ac)
```

```
[1] 0.7980254
```

```r
print(hc_average_pA$ac)
```

```
[1] 0.7738333
```

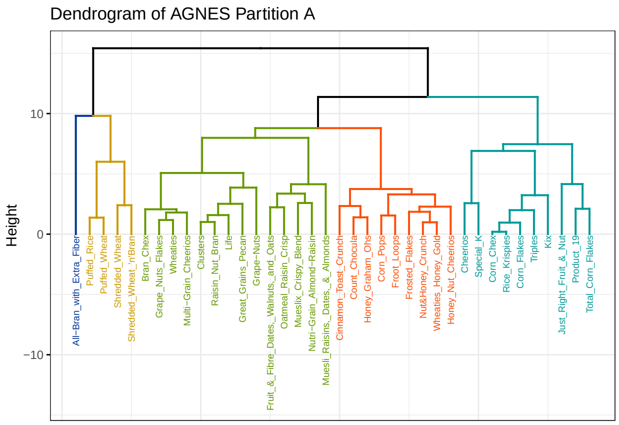
```r
print(hc_ward_pA$ac)
```

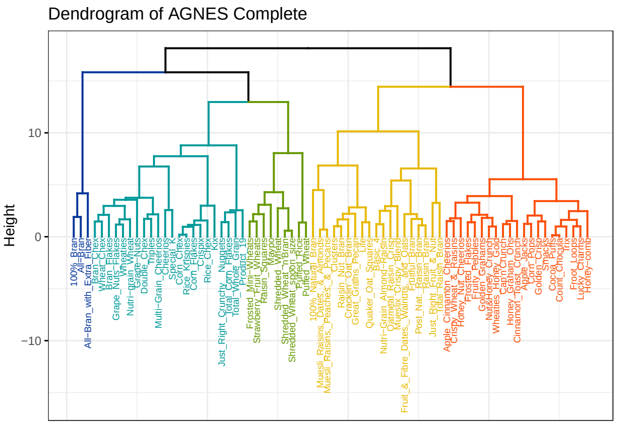
```
[1] 0.8605825
```

It allows us to determine that the best linkage is Ward with 86.05% accuracy.

Comparison of different dataset's sizes

Here we are going to compare the subset (partition A) with the whole dataset utilizing the dendrograms through the color cluster.





These plots show the following:

- Blue and red clusters were grouped the same groups on both datasets.
- The yellow cluster from Partition A was almost classified the same except for two data points, "Brain_chex" AND "Wheaties" that used to belong to the yellow cluster from the whole dataset and changed to cluster green on Partition A sub-dataset.
- Same happens with "Triplex" and "Grapes_Nuts" that used to belong to the yellow cluster from the whole dataset and changed to turquoise in the Partition subset. But in general, the model did a good job by grouping the cereal's type in the same cluster most of the time.

Question 5

5. The elementary public schools would like to choose a set of cereals to include in their daily cafeterias. Every day a different cereal is offered, but all cereals should support a healthy diet. For this goal, you are requested to find a cluster of "healthy cereals." Should the data be normalized? If not, how should they be used in the cluster analysis?

To analyze which group of cereals are healthier to distribute daily in cafeterias in elementary public schools, we will use the non-standardized dataset. In my opinion, it is more meaningful and easier to compare if we look at the variables in their original scale.

Here is a table summarizing the number of cereals per cluster:

```
clusters_5
 1  2  3  4  5 
 3 19 21 22  9 
```

Here we can see there are 3 bowls of cereal in cluster 1 (blue cluster in the AGNES Dendrogram), 20 in the second one, and so on.

The following output shows the name of cereals in each cluster, which we already identified on the AGNES Dendrogram in previous sections.

```
# See all the observations that belong to each cluster
sapply(unique(clusters_5),function(g)cereal$name[clusters_5 == g])
```

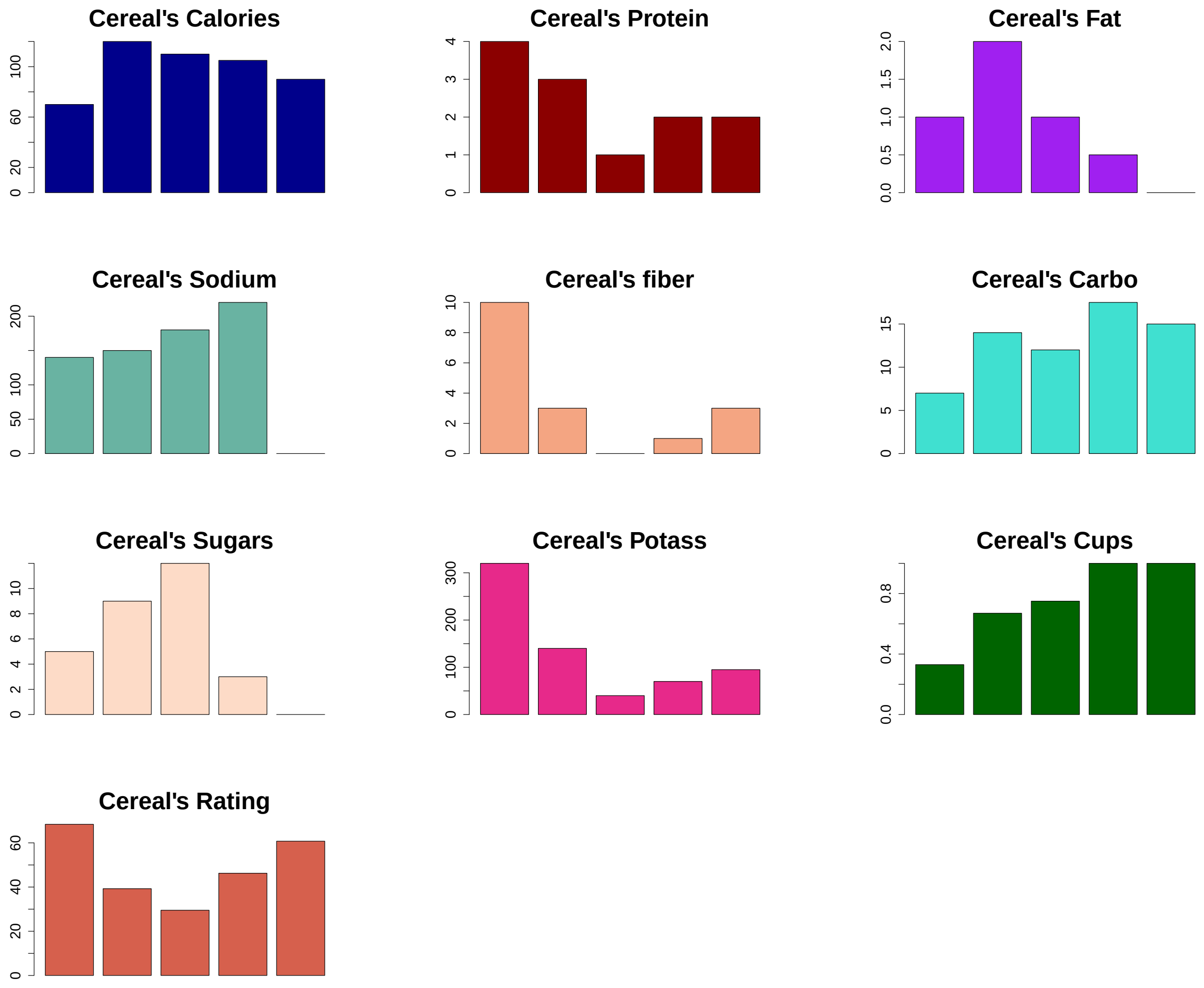
```
[[1]]
[1] "100%_Bran"                 "All-Bran"                  "All-Bran_with_Extra_Fiber"

[[2]]
 [1] "100%_Natural_Bran"                      "Basic_4"                              
 [3] "Clusters"                               "Cracklin'_Oat_Bran"                   
 [5] "Fruit_&_Fibre_Dates,_Walnuts,_and_Oats" "Fruitful_Bran"                        
 [7] "Great_Grains_Pecan"                     "Just_Right_Fruit_&_Nut"               
 [9] "Life"                                   "Muesli_Raisins,_Dates,_&_Almonds"     
[11] "Muesli_Raisins,_Peaches,_&_Pecans"      "Mueslix_Crispy_Blend"                 
[13] "Nutri-Grain_Almond-Raisin"              "Oatmeal_Raisin_Crisp"                 
[15] "Post_Nat._Raisin_Bran"                  "Quaker_Oat_Squares"                   
[17] "Raisin_Bran"                            "Raisin_Nut_Bran"                      
[19] "Total_Raisin_Bran"                     

[[3]]
 [1] "Apple_Cinnamon_Cheerios" "Apple_Jacks"             "Cap'n'Crunch"           
 [4] "Cinnamon_Toast_Crunch"   "Cocoa_Puffs"             "Corn_Pops"              
 [7] "Count_Chocula"           "Crispy_Wheat_&_Raisins"  "Froot_Loops"            
[10] "Frosted_Flakes"          "Fruity_Pebbles"          "Golden_Crisp"           
[13] "Golden_Grahams"          "Honey_Graham_Ohs"        "Honey_Nut_Cheerios"     
[16] "Honey-comb"              "Lucky_Charms"            "Nut&Honey_Crunch"       
[19] "Smacks"                  "Trix"                    "Wheaties_Honey_Gold"    

[[4]]
 [1] "Bran_Chex"                   "Bran_Flakes"                 "Cheerios"                   
 [4] "Corn_Chex"                   "Corn_Flakes"                 "Crispix"                    
 [7] "Double_Chex"                 "Grape_Nuts_Flakes"           "Grape-Nuts"                 
[10] "Just_Right_Crunchy__Nuggets" "Kix"                         "Multi-Grain_Cheerios"       
[13] "Nutri-grain_Wheat"           "Product_19"                  "Rice_Chex"                  
[16] "Rice_Krispies"               "Special_K"                   "Total_Corn_Flakes"          
[19] "Total_Whole_Grain"           "Triples"                     "Wheat_Chex"                 
[22] "Wheaties"                   

[[5]]
[1] "Frosted_Mini-Wheats"       "Maypo"                     "Puffed_Rice"              
[4] "Puffed_Wheat"              "Raisin_Squares"            "Shredded_Wheat"           
[7] "Shredded_Wheat_'n'Bran"    "Shredded_Wheat_spoon_size" "Strawberry_Fruit_Wheats"  
```

The next table is essential because it will allow us to identify the healthier cluster for cafeterias in public schools. To do the following, we are going to use the median of each cluster, then compare it with each other.

```
# Summary table showing the median of each variable
cluster_table <- aggregate(cereal[,-c(1:3)],list(clusters_5),median)
cluster_table
```

```
  Group.1 calories protein fat sodium fiber carbo sugars potass vitamins shelf
1       1       70       4 1.0    140    10   7.0      5    320       25   3.0
2       2      120       3 2.0    150     3  14.0      9    140       25   3.0
3       3      110       1 1.0    180     0  12.0     12     40       25   2.0
4       4      105       2 0.5    220     1  17.5      3     70       25   2.5
5       5       90       2 0.0      0     3  15.0      0     95        0   2.0
  weight cups   rating
1   1.00 0.33 68.40297
2   1.25 0.67 39.25920
3   1.00 0.75 29.50954
4   1.00 1.00 46.26108
5   1.00 1.00 60.75611
```

This summary table is a great way to conclude. Nevertheless, let's plot some data to get more insights.



The previous outputs shows that:

- Cluster 1 fits the needs of our client! It has the highest levels of protein, fiber, potassium. Low levels of sodium, fat, sugars, and the lowest in carbs and calories. Additionally, its ratings are the highest of all five groups. Nevertheless, part of our client's petition is to have a different cereal per day, which this cluster does not satisfy this need. For this reason, we will also recommend cluster 4 to satisfy this request. Cluster 4 has zero fats, a low number of sugars, and it has the second-lowest number of calories after cluster 1. It also has a good number of proteins and fiber.

Question 6

6. How do you compare hierarchical clustering and k-means? What are the main advantages of hierarchical clustering compared to k-means?

I compared both methods by solving the same problem. It allowed me to compare both methods, see their strength but also their drawbacks.

I think the hierarchical clustering model is easy to apply. The option of having a beautiful dendrogram to see the division of the clusters is easy to interpret and simple to apply. Regarding its drawbacks, the idea of not having an algorithm choosing or calculating the optimal number of clusters is something I do not like. I think it is heuristic, and the number of clusters from my perspective could be different from another person.

Based on K-means, it is widely used in large dataset applications. It is easy to apply, and a fast algorithm. In my opinion, some disadvantages are that it needs the hyperparameter (number of k) to run the model. Even though some methods guide us to set the optimal k, sometimes we can get different outputs between those methods.

Sources

Kaya, F., & Sahin, S. (2017). Comparison of Hierarchical and Non-Hierarchical Clustering Algorithms.

Stats Berkeley. (2011) Cluster Analysis. Retrieved from: https://www.stat.berkeley.edu/~s133/Cluster2a.html