Assignment_5

Andrew Gutierrez

2022-12-02

Installing requisite libraries:

```
library(cluster)
```

First, I'll read the Cereals.csv file into a DataFrame in R:

```
cereals = read.csv("C:\\Users\\gutiera9\\Documents\\MSBA KSU\\Cereals.csv",header=T,sep=",")

# set row names to the "utilities"name" column
row.names(cereals) <- cereals[,1]

head(cereals)
```

```
##                                          name mfr type calories protein
## 100%_Bran                           100%_Bran   N    C       70       4
## 100%_Natural_Bran           100%_Natural_Bran   Q    C      120       3
## All-Bran                             All-Bran   K    C       70       4
## All-Bran_with_Extra_Fiber All-Bran_with_Extra_Fiber   K    C       50       4
## Almond_Delight                 Almond_Delight   R    C      110       2
## Apple_Cinnamon_Cheerios   Apple_Cinnamon_Cheerios   G    C      110       2
##                           fat sodium fiber carbo sugars potass vitamins shelf
## 100%_Bran                   1    130  10.0   5.0      6    280       25     3
## 100%_Natural_Bran           5     15   2.0   8.0      8    135        0     3
## All-Bran                    1    260   9.0   7.0      5    320       25     3
## All-Bran_with_Extra_Fiber   0    140  14.0   8.0      0    330       25     3
## Almond_Delight              2    200   1.0  14.0      8     NA       25     3
## Apple_Cinnamon_Cheerios     2    180   1.5  10.5     10     70       25     1
##                           weight cups   rating
## 100%_Bran                      1 0.33 68.40297
## 100%_Natural_Bran              1 1.00 33.98368
## All-Bran                       1 0.33 59.42551
## All-Bran_with_Extra_Fiber      1 0.50 93.70491
## Almond_Delight                 1 0.75 34.38484
## Apple_Cinnamon_Cheerios        1 0.75 29.50954
```

I'll then pre-process the data by removing all blank values from the dataframe.

```
cereals <- na.omit(cereals)
head(cereals)
```

```
##                                             name mfr type calories protein
## 100%_Bran                                100%_Bran   N    C       70       4
## 100%_Natural_Bran                100%_Natural_Bran   Q    C      120       3
## All-Bran                                  All-Bran   K    C       70       4
## All-Bran_with_Extra_Fiber All-Bran_with_Extra_Fiber   K    C       50       4
## Apple_Cinnamon_Cheerios     Apple_Cinnamon_Cheerios   G    C      110       2
## Apple_Jacks                             Apple_Jacks   K    C      110       2
##                           fat sodium fiber carbo sugars potass vitamins shelf
## 100%_Bran                   1    130  10.0   5.0      6    280       25     3
## 100%_Natural_Bran           5     15   2.0   8.0      8    135        0     3
## All-Bran                    1    260   9.0   7.0      5    320       25     3
## All-Bran_with_Extra_Fiber   0    140  14.0   8.0      0    330       25     3
## Apple_Cinnamon_Cheerios     2    180   1.5  10.5     10     70       25     1
## Apple_Jacks                 0    125   1.0  11.0     14     30       25     2
##                           weight cups   rating
## 100%_Bran                      1 0.33 68.40297
## 100%_Natural_Bran              1 1.00 33.98368
## All-Bran                       1 0.33 59.42551
## All-Bran_with_Extra_Fiber      1 0.50 93.70491
## Apple_Cinnamon_Cheerios        1 0.75 29.50954
## Apple_Jacks                    1 1.00 33.17409
```

- 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.

My first step here will be to normalize the numerical variable values, and then to compute the Euclidean distance

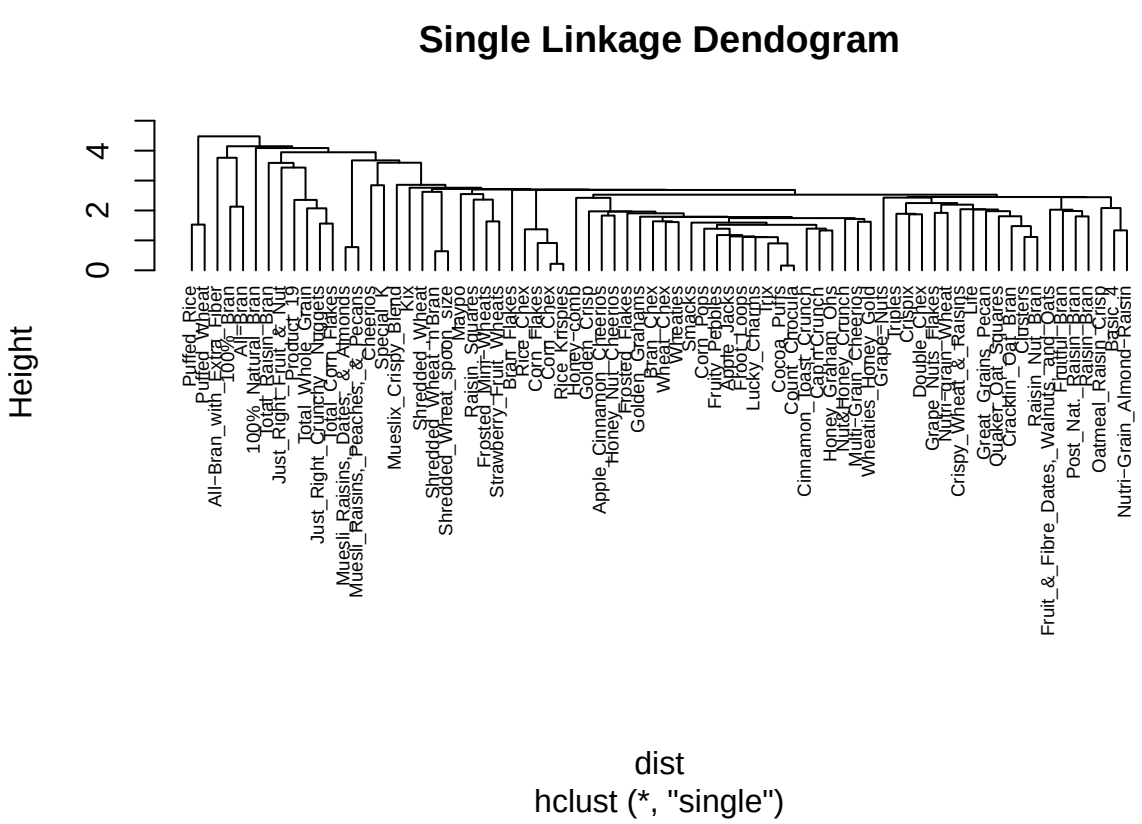
```
cereals[,c(4:16)] <- scale(cereals[,c(4:16)]) # normalize

dist <- dist(cereals, method = "euclidean") # compute euclidean
```

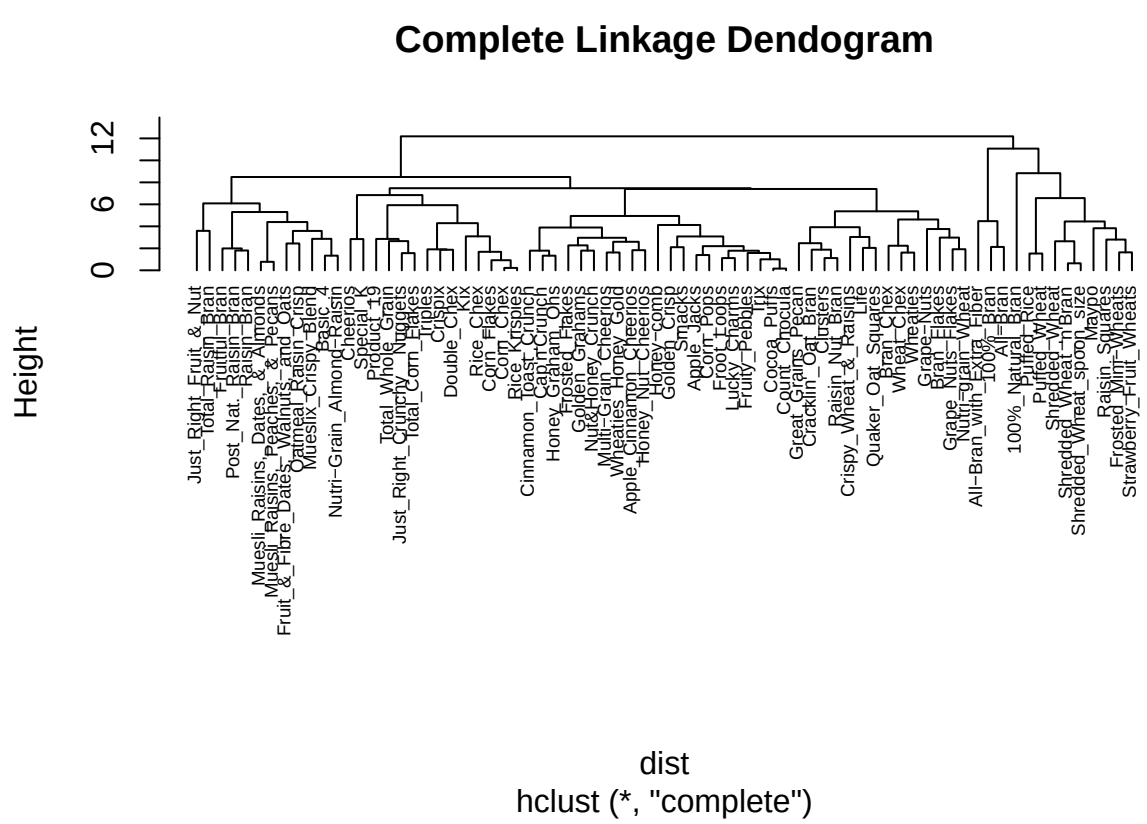
```
## Warning in dist(cereals, method = "euclidean"): NAs introduced by coercion
```

Now, using the hclust function, I'll compare the single linkage, complete linkage, average linkage, and Ward method plots.

```
single <- hclust(dist, method = "single") # single linkage
plot(single, cex = 0.6, hang = -1, main="Single Linkage Dendogram")
```

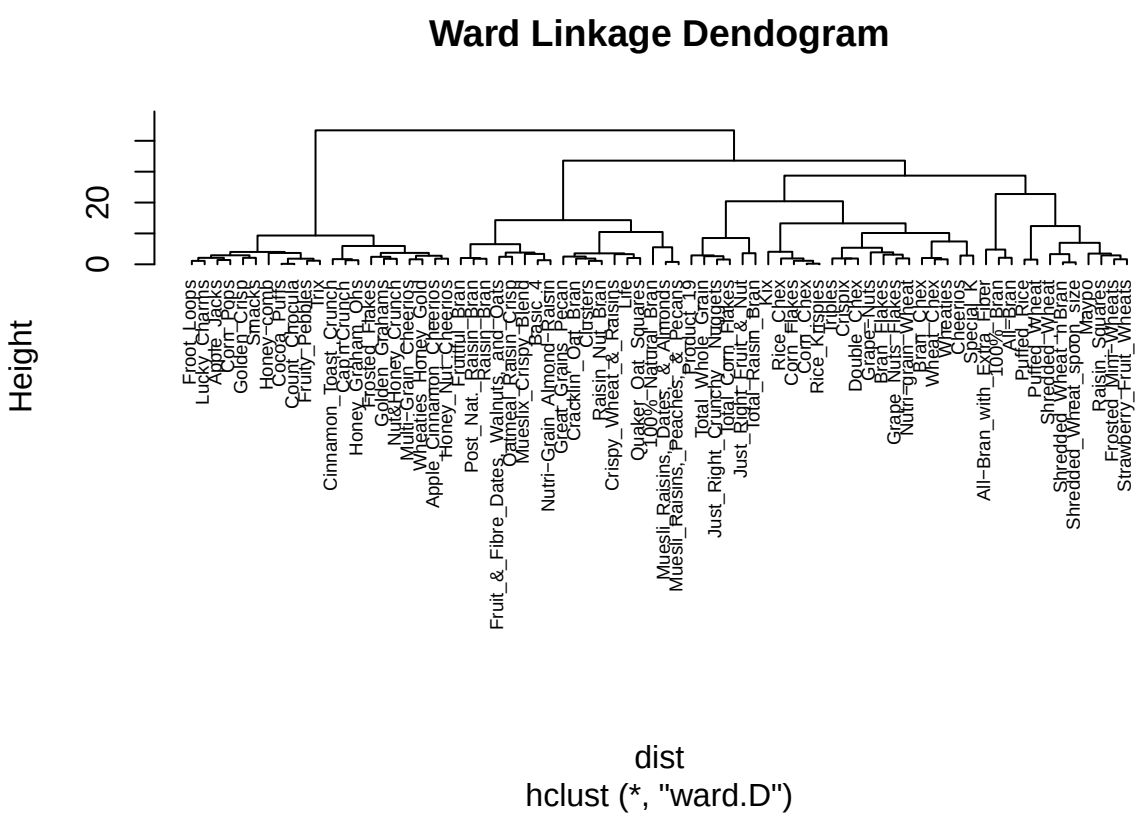



```
complete <- hclust(dist, method = "complete") # complete linkage
plot(complete, cex = 0.6, hang = -1, main="Complete Linkage Dendogram")
```




```
average <- hclust(dist, method = "average") # average linkage
plot(average, cex = 0.6, hang = -1, main="Average Linkage Dendogram")
```


```
ward <- hclust(dist, method = "ward.D") # ward method
plot(ward, cex = 0.6, hang = -1, main="Ward Linkage Dendogram")
```



Based on the four plots, I believe the Ward method is the best method, as it clearly features the strongest clustering structure from among the four methods. This is borne out by a look at the agglomerative coefficients for each of the methods when applied to the dataset.

```
single_coef <- agnes(dist, method = "single")
complete_coef <- agnes(dist, method = "complete")
average_coef <- agnes(dist, method = "average")
ward_coef <- agnes(dist, method = "ward")

print("Single Linkage Coefficient: ")
```

```
## [1] "Single Linkage Coefficient: "
```

```
print(single_coef$ac)
```

```
## [1] 0.6067859
```

```
print("Complete Linkage Coefficient: ")
```

```
## [1] "Complete Linkage Coefficient: "
```

```
print(complete_coef$ac)
```

```
## [1] 0.8353712
```

```
print("Average Linkage Coefficient: ")
```

```
## [1] "Average Linkage Coefficient: "
```

```
print(average_coef$ac)
```

```
## [1] 0.7766075
```

```
print("Ward Method Coefficient: ")
```

```
## [1] "Ward Method Coefficient: "
```

```
print(ward_coef$ac)
```

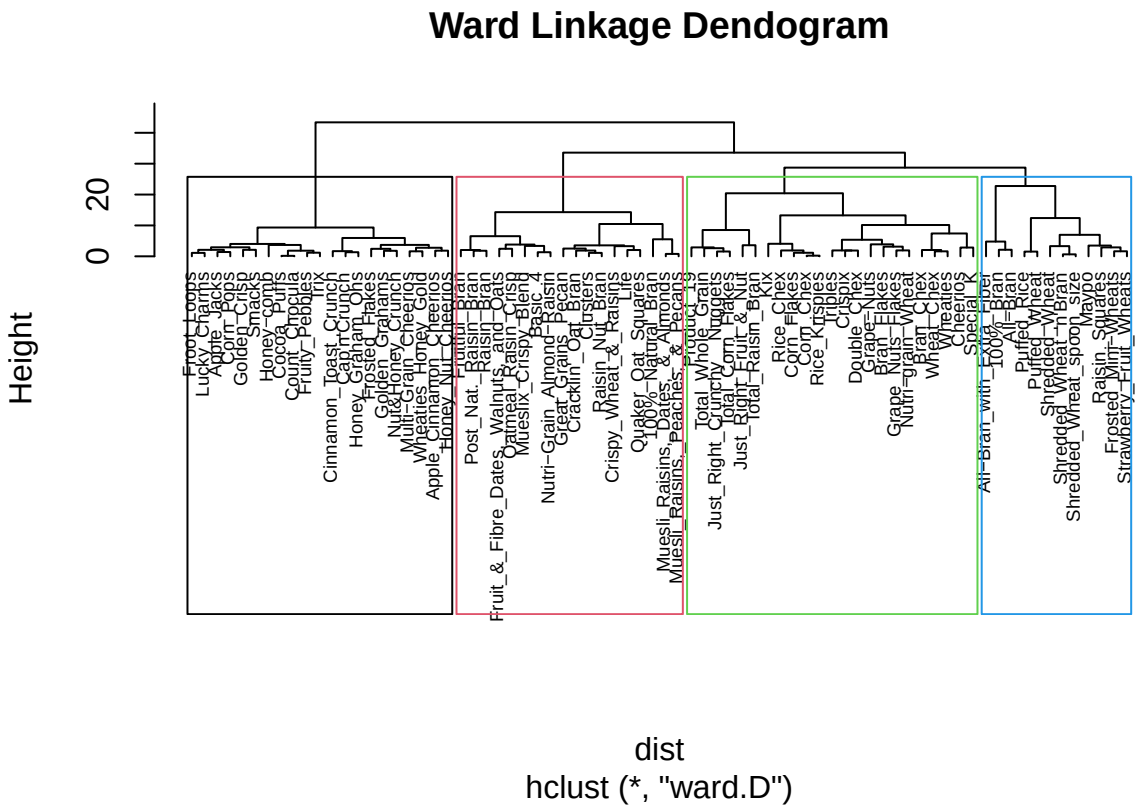
```
## [1] 0.9046042
```

Of the four linkages the Ward method has the highest agglomerative coefficient of 0.9, backing up my choice that it is the best form of method for this particular dataset.

- How many clusters would you choose?

To my eye, I see four clear clusters on the ward dendogram. Drawing the dendogram with borders around the clusters makes this apparent (see below).

```
plot(ward, cex = 0.6, hang = -1, main="Ward Linkage Dendogram")
rect.hclust(ward, k=4, border = 1:4)
```



Comment on the structure of the clusters and on their stability. Hint: To check stability, partition the data and see how well clusters formed based on one part apply to the other part. To do this: - Cluster partition A - Use the cluster centroids from A to assign each record in partition B (each record is assigned to the cluster with the closest centroid). - Assess how consistent the cluster assignments are compared to the assignments based on all the data.

First, I'll partition the dataset into two partitions (A and B). Then, I'll cluster the first partition using the Ward method like before.

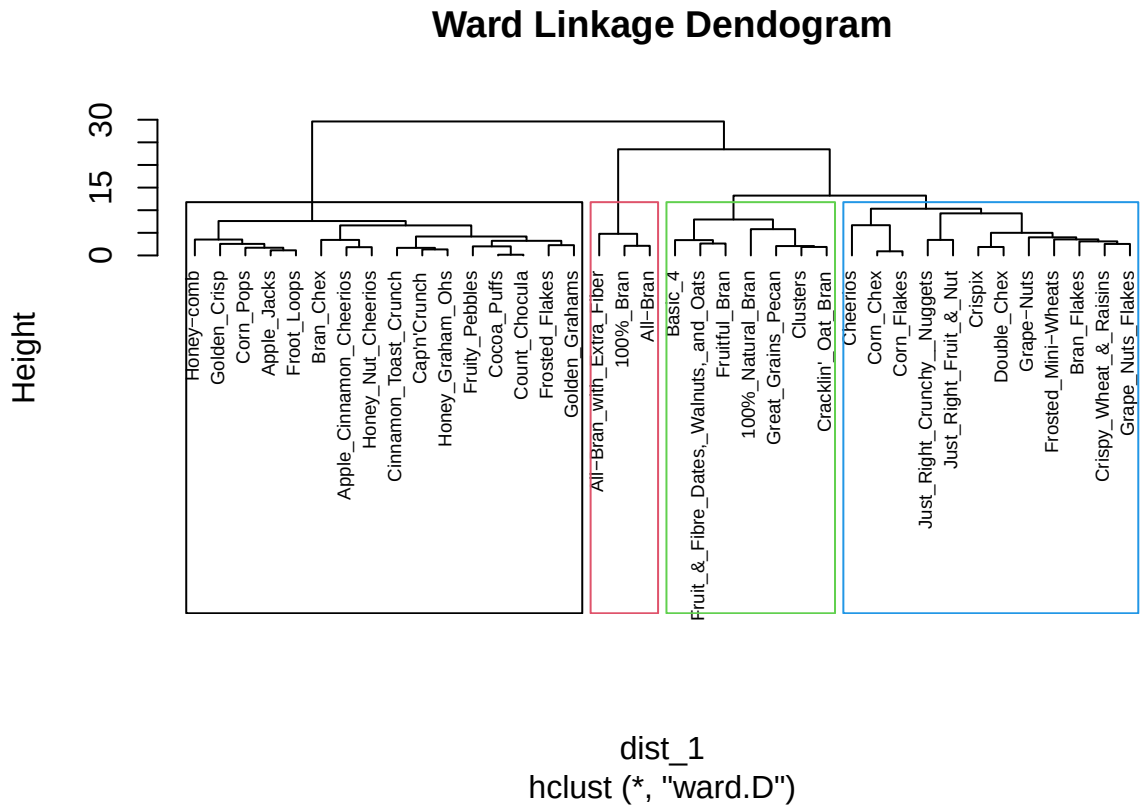
```
# partition data and compute euclidean distances
dist_1 <- dist(cereals[0:38,], method = "euclidean")
```

```
## Warning in dist(cereals[0:38, ], method = "euclidean"): NAs introduced by
## coercion
```

```
dist_2 <- dist(cereals[39:76,], method = "euclidean")
```

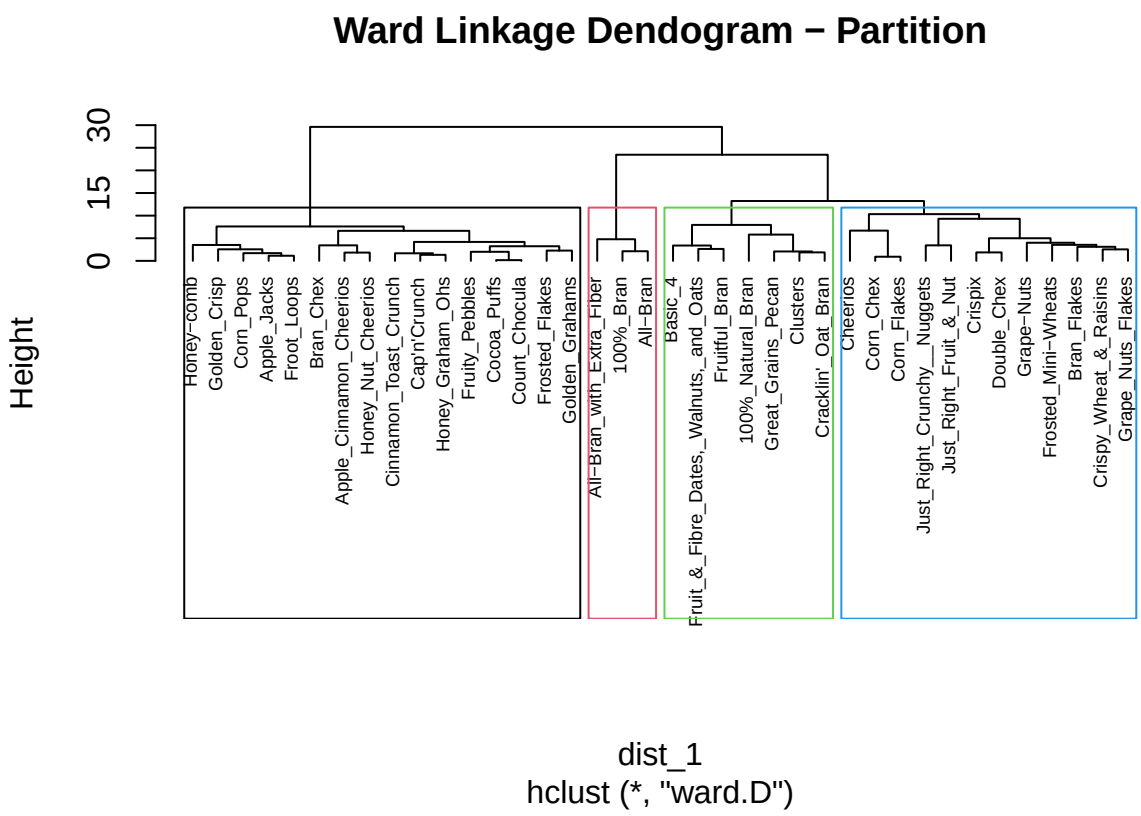
```
## Warning in dist(cereals[39:76, ], method = "euclidean"): NAs introduced by
## coercion
```

```
#cluster first partition
ward_1 <- hclust(dist_1, method = "ward.D")
plot(ward_1, cex = 0.6, hang = -1, main="Ward Linkage Dendogram")
rect.hclust(ward_1, k=4, border = 1:4)
```

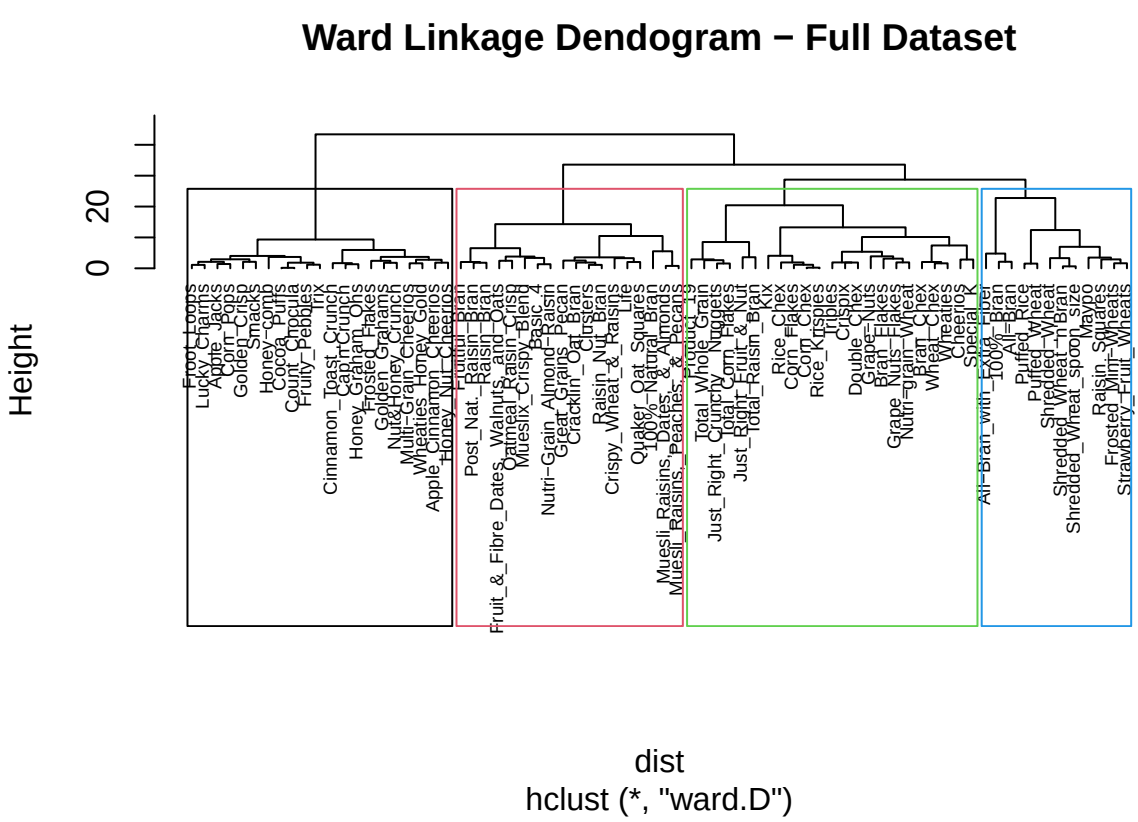


Let's see how the plot from partition 1 compares with the plot of the overall dataset:

```
### Plot of partition 1
plot(ward_1, cex = 0.6, hang = -1, main="Ward Linkage Dendogram - Partition")
rect.hclust(ward_1, k=4, border = 1:4)
```




```
plot(ward, cex = 0.6, hang = -1, main="Ward Linkage Dendogram - Full Dataset")
rect.hclust(ward, k=4, border = 1:4)
```



In comparing the clusters from the partition vs. the clusters from the overall dataset, one finds that the allocation is fairly similar, indicating a general stableness to the hierarchy. For example, the partition has All-Bran with Extra Fiber, 100% Bran, and All-Bran clustered together, as does the full dataset. The partition has Basic 4, Fruitful Bran, and Cracklin' Oat Bran all clustered together, as does the full dataset. And the partition and full dataset both have Honeycombs, Corn-Pops, and Apple Jacks all clustered together. In fact, in reviewing the clusters from the partition, I can't find a single inconsistency where two cereals are clustered together in the partition but NOT in the full dataset, or vice-versa.

- 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?

First, I'll calculate the Euclidean distances for a subset of the variables in the dataset that specifically have to do with "healthiness" (specifically calories, protein, fat, sodium, fiber, carbo, sugars, potass, and vitamins). Note that these variables should in fact be normalized, as the scale of variables can affect our hierarchical clustering model.

Once I have those distances, I'll calculate the agglomerative coefficients for the data using the Single, Complete, Average, and Ward Method linkages, in order to determine which method works best with this subset of variables.

```
disthealth <- dist(cereals[,c(4:12)], method = "euclidean")

single_coef_health <- agnes(disthealth, method = "single")
complete_coef_health <- agnes(disthealth, method = "complete")
```

```r
average_coef_health <- agnes(disthealth, method = "average")
ward_coef_health <- agnes(disthealth, method = "ward")

print("Single Linkage Coefficient: ")
```

```
## [1] "Single Linkage Coefficient: "
```

```r
print(single_coef_health$ac)
```

```
## [1] 0.6695003
```

```r
print("Complete Linkage Coefficient: ")
```

```
## [1] "Complete Linkage Coefficient: "
```

```r
print(complete_coef_health$ac)
```

```
## [1] 0.8614259
```

```r
print("Average Linkage Coefficient: ")
```

```
## [1] "Average Linkage Coefficient: "
```

```r
print(average_coef_health$ac)
```

```
## [1] 0.8123086
```

```r
print("Ward Method Coefficient: ")
```

```
## [1] "Ward Method Coefficient: "
```

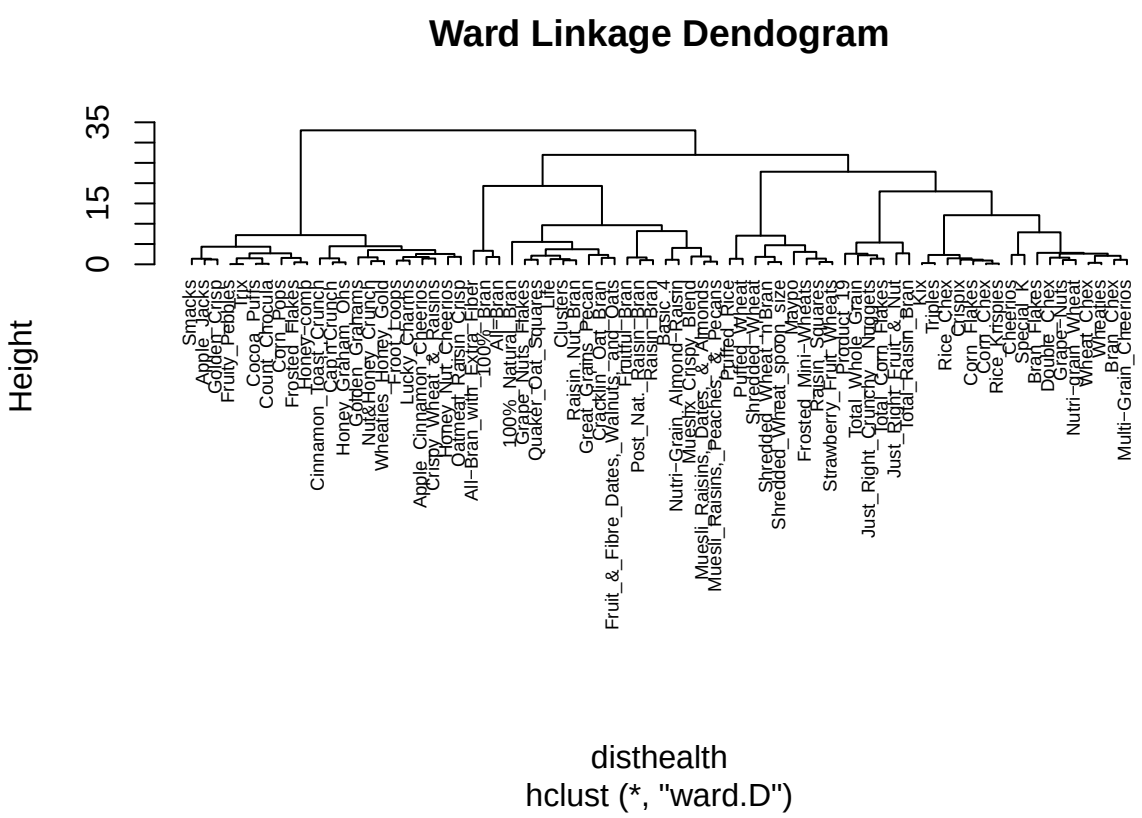
```r
print(ward_coef_health$ac)
```

```
## [1] 0.9114504
```

This shows that the Ward method still has the strongest clustering structure, at 0.91.

Now, I'll plot the clusters using the Ward method.

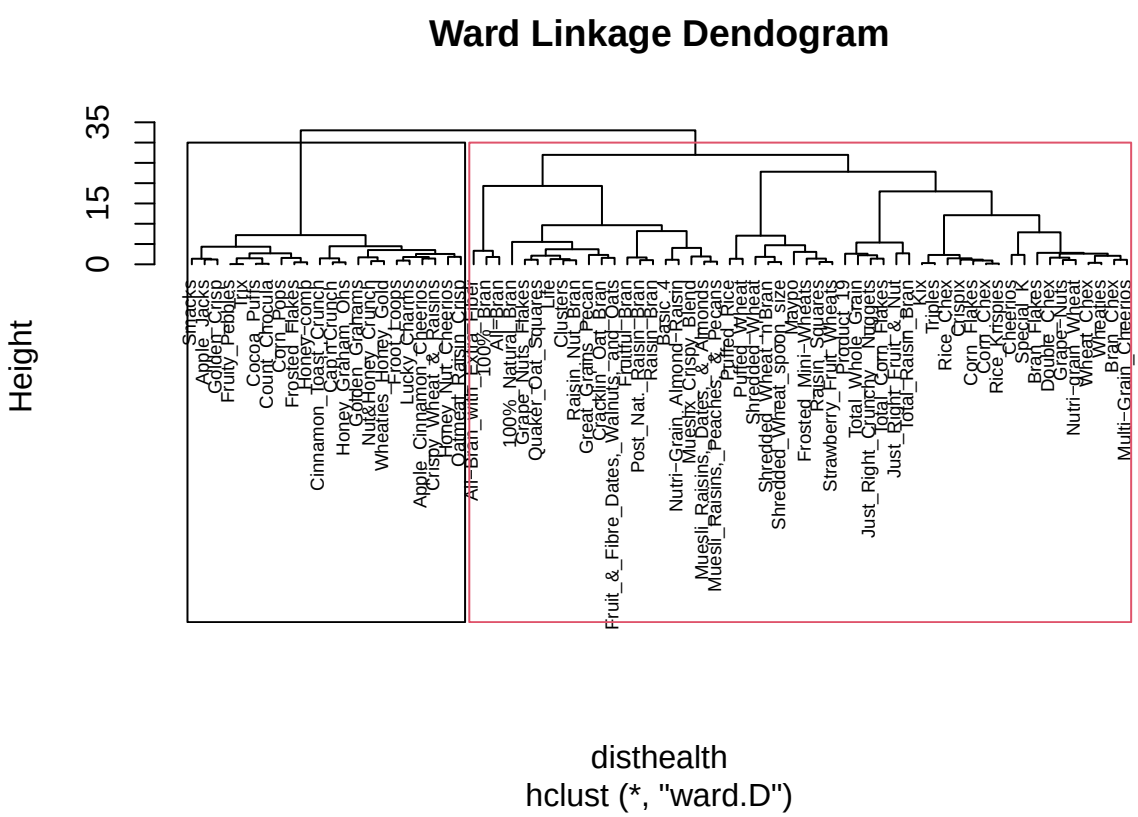
```r
ward_health <- hclust(disthealth, method = "ward.D") # ward method
plot(ward_health, cex = 0.6, hang = -1, main="Ward Linkage Dendogram")
```



disthealth
hclust (*, "ward.D")

Let's try drawing some borders on this dendogram in order to see two distinct clusters - healthy, and unhealthy.

```
plot(ward_health, cex = 0.6, hang = -1, main="Ward Linkage Dendogram")
rect.hclust(ward_health, k=2, border = 1:3)
```



The “healthy” cluster on this dendogram is clearly the second cluster, aka “red”. This cluster features cereals that generally have a lower sugar content - including Rice Chex, Corn Flakes, Raisin Nut Bran, Puffed Wheat, and Shredded Wheat. The “unhealthy” cluster, by contrast, features cereals such as Cocoa Puffs, Count Chocula, Fruity Pebbles, and Apple Jacks - all cereals that are notorious for having high sugar content. The elementary public schools would be advised to select cereals only from the second “red” cluster.