Assignment 5

Gabriel Arsego

2025-11-19

Preprocessing

```
Cereals <- na.omit(Cereals)
#After trying to create the dendograms many times, and failing to get a nice
result, I realized that if I shortened the names of the cereals, it would be
easier to vizualise the graphs:
short_c.names <- substr(Cereals$name, 1, 20)
#The short_c.names I will use later to name the rows for the normalized data.
I'm not sure if this is the best option, but I believe that it at least
improves visualization. The maximum number of characters at 20 avoids having
non-unique names.
head(Cereals)
```

```
##                         name mfr type calories protein fat sodium fiber
carbo
## 1                 100%_Bran   N    C       70       4   1    130  10.0
5.0
## 2         100%_Natural_Bran   Q    C      120       3   5     15   2.0
8.0
## 3                  All-Bran   K    C       70       4   1    260   9.0
7.0
## 4 All-Bran_with_Extra_Fiber   K    C       50       4   0    140  14.0
8.0
## 6   Apple_Cinnamon_Cheerios   G    C      110       2   2    180   1.5
10.5
## 7               Apple_Jacks   K    C      110       2   0    125   1.0
11.0
##   sugars potass vitamins shelf weight cups   rating
## 1      6    280       25     3      1 0.33 68.40297
## 2      8    135        0     3      1 1.00 33.98368
## 3      5    320       25     3      1 0.33 59.42551
## 4      0    330       25     3      1 0.50 93.70491
## 6     10     70       25     1      1 0.75 29.50954
## 7     14     30       25     2      1 1.00 33.17409
```

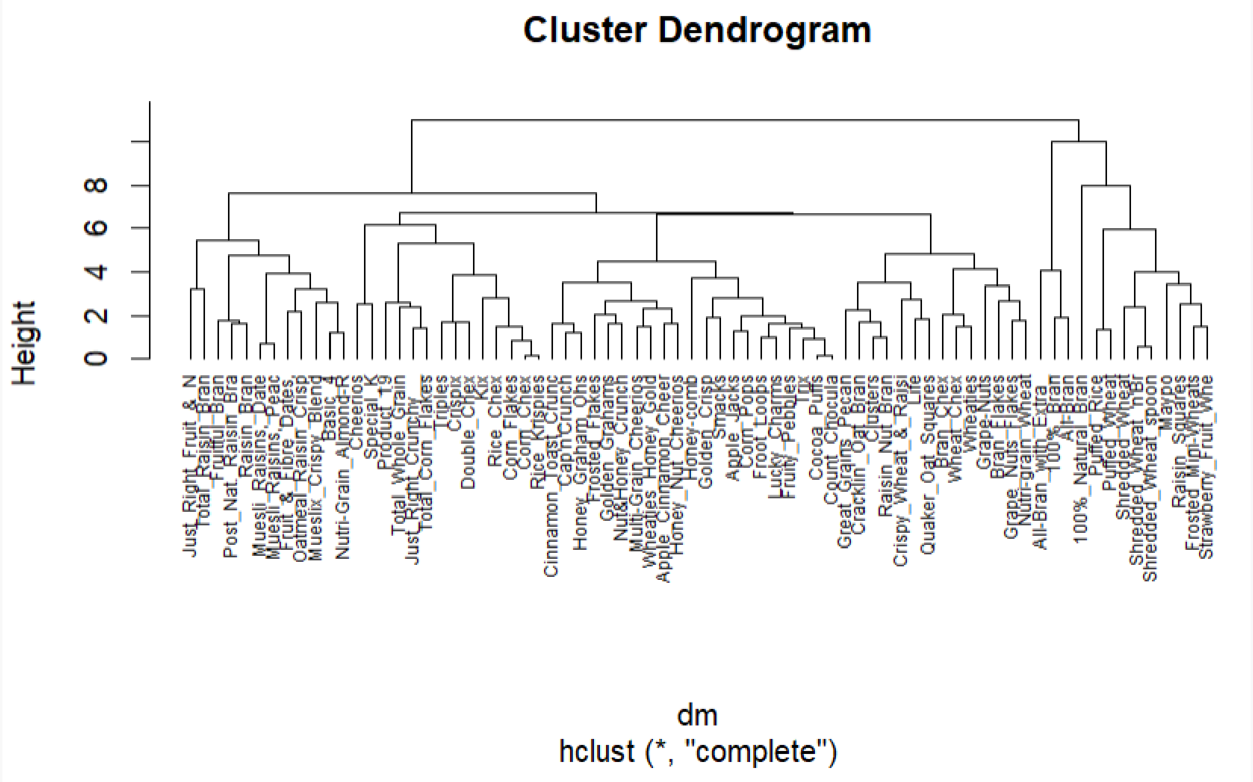
Standardizing here is necessary for hierarchical clustering, so I removed the first 3 columns that are not numeric to apply scale() to the data:

```
Numbers <- Cereals[ ,c(4:16)]
rownames(Numbers) <- short_c.names
Cereals.norm <- scale(Numbers)
head(Cereals.norm)
```

```
##                        calories    protein        fat     sodium
fiber
## 100%_Bran            -1.8659155  1.3817478  0.0000000 -0.3910227
3.22866747
## 100%_Natural_Bran     0.6537514  0.4522084  3.9728810 -1.7804186 -
0.07249167
## All-Bran             -1.8659155  1.3817478  0.0000000  1.1795987
2.81602258
## All-Bran_with_Extra_ -2.8737823  1.3817478 -0.9932203 -0.2702057
4.87924705
## Apple_Cinnamon_Cheer  0.1498180 -0.4773310  0.9932203  0.2130625 -
0.27881412
## Apple_Jacks           0.1498180 -0.4773310 -0.9932203 -0.4514312 -
0.48513656
##                           carbo     sugars     potass   vitamins
shelf
## 100%_Bran            -2.5001396 -0.2542051  2.5605229 -0.1818422
0.9419715
## 100%_Natural_Bran    -1.7292632  0.2046041  0.5147738 -1.3032024
0.9419715
## All-Bran             -1.9862220 -0.4836096  3.1248675 -0.1818422
0.9419715
## All-Bran_with_Extra_ -1.7292632 -1.6306324  3.2659536 -0.1818422
0.9419715
## Apple_Cinnamon_Cheer -1.0868662  0.6634132 -0.4022862 -0.1818422 -
1.4616799
## Apple_Jacks          -0.9583868  1.5810314 -0.9666308 -0.1818422 -
0.2598542
##                          weight       cups     rating
## 100%_Bran            -0.2008324 -2.0856582  1.8549038
## 100%_Natural_Bran    -0.2008324  0.7567534 -0.5977113
## All-Bran             -0.2008324 -2.0856582  1.2151965
## All-Bran_with_Extra_ -0.2008324 -1.3644493  3.6578436
## Apple_Cinnamon_Cheer -0.2008324 -0.3038480 -0.9165248
## Apple_Jacks          -0.2008324  0.7567534 -0.6553998
```

Next I created the dissimilarity matrix using euclidean distance, so that I could apply hclust():

```
dm <- dist(Cereals.norm, method = "euclidean")
hc.cereals <- hclust(dm, method = "complete")
plot(hc.cereals, cex = 0.6, hang = -1)
```



Then I use Agnes with single, complete, and average linkage, and Ward, to find out which is the best method:

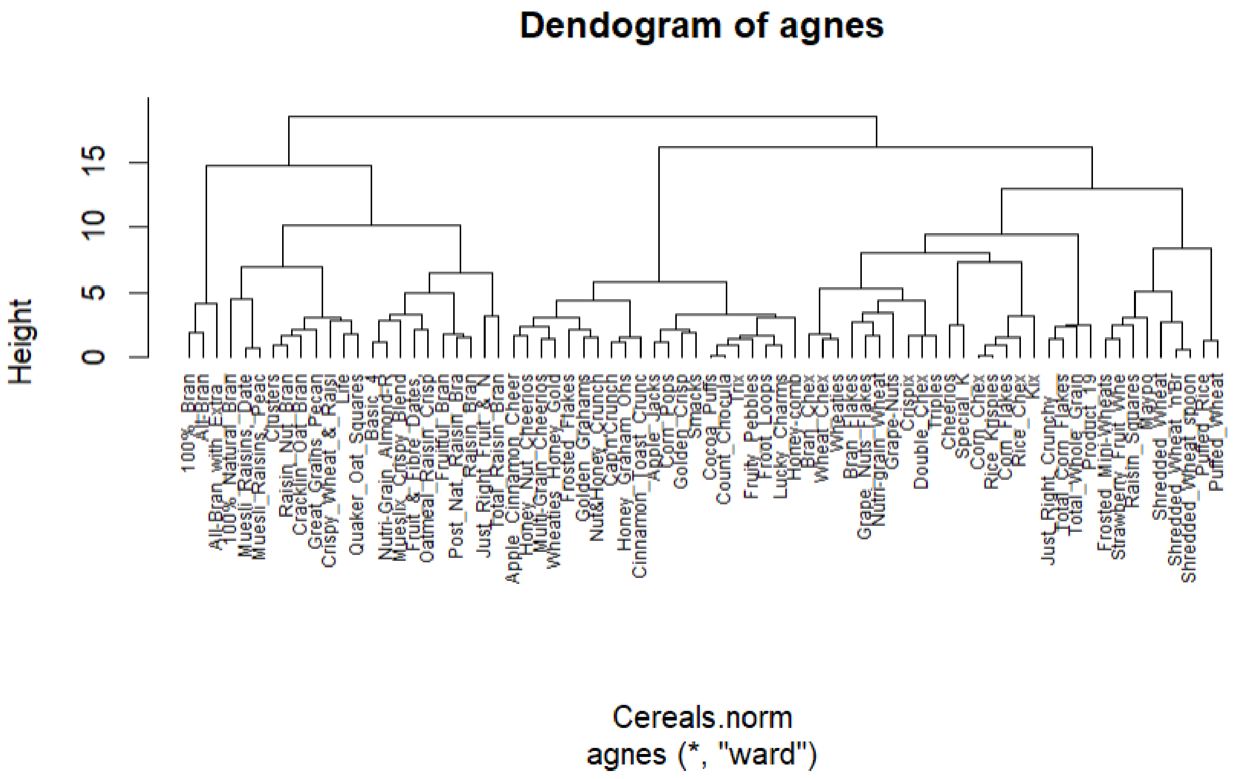
```
sg.cereals <- agnes(Cereals.norm, method = "single")
ct.cereals <- agnes(Cereals.norm, method = "complete")
avg.cereals <- agnes(Cereals.norm, method = "average")
wrd.cereals <- agnes(Cereals.norm, method = "ward")
```

Printing the Agglomerative coefficients to compare the methods:

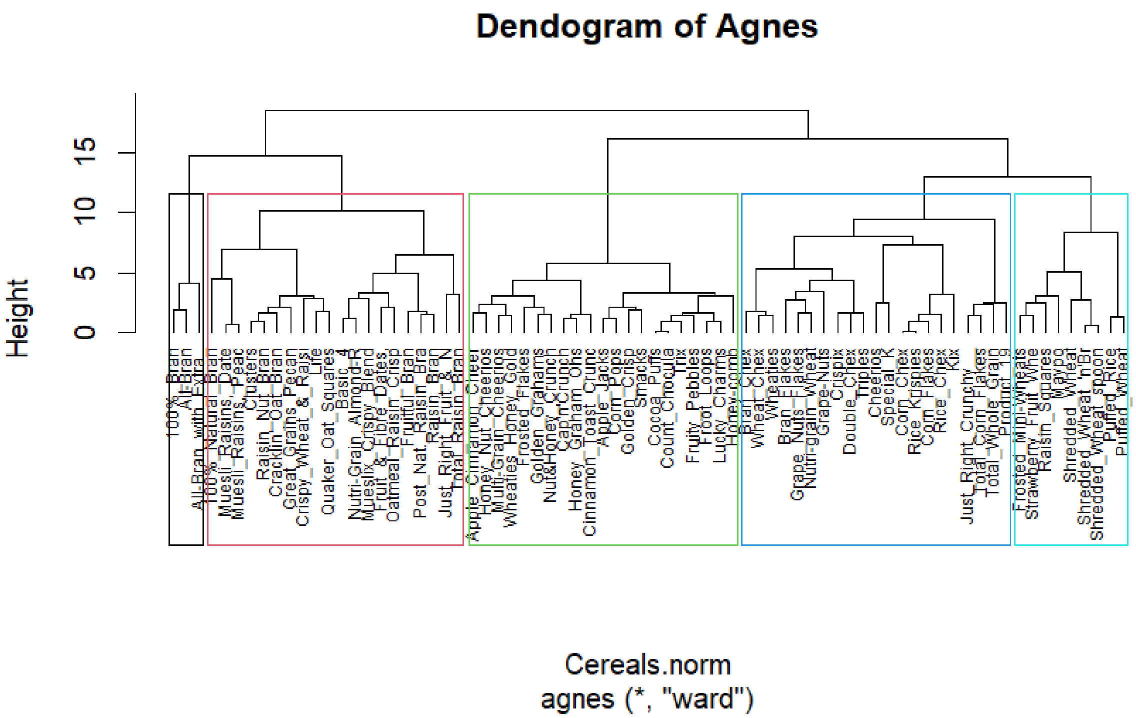
```
print(sg.cereals$ac)

## [1] 0.6067859

print(ct.cereals$ac)

## [1] 0.8353712

print(avg.cereals$ac)

## [1] 0.7766075

print(wrd.cereals$ac)

## [1] 0.9046042
```

The Ward method gives us the highest value for the agglomerative coefficient meaning that it has the strongest clustering structure. For that reason Ward is the best method. Next I applied pltree() to wrd.cereals to create a dendogram:

```r
pltree(wrd.cereals, cex = 0.6, hang = -1, main = "Dendogram of agnes")
```



```r
wrd.cereals.boxes <- as.hclust(wrd.cereals)
Cluster.wcb <- cutree(as.hclust(wrd.cereals), k = 5)
plot(wrd.cereals.boxes, main = "Dendogram of Agnes", cex = 0.6, hang = -1)
rect.hclust(wrd.cereals.boxes, k = 5, border = 1:5)
```



```r
#Here using cutree() I created Cluster.wcb which I inserted into the Cereals
dataset, this way we can view the clusters more clearly.
Cereals$cluster <- Cluster.wcb
```

My choice here was for 5 clusters. More than that would be too much, since it would separate similar observations unnecessarily, and would make it harder to understand each cluster. I thought of doing four, but then one of the clusters (the one to the right on the dendogram) would be disproportionately larger than the other ones.

Now, to check stability I will follow the steps provided:

```r
Partition <- createDataPartition(Numbers$calories, p = 0.5, list = FALSE)
A <- Numbers[Partition, ]
B <- Numbers[-Partition, ]
A.scale <- scale(A)
B.scale <- scale(B)
```

Now clustering Partition A using Agnes clustering and k = 5.

```r
wrd.A <- agnes(A.scale, method = "ward")
Cluster.wrd.A <- cutree(as.hclust(wrd.A), k = 5)
A$cluster <- Cluster.wrd.A
```

NOTE: For the question asking to check stability, I did as much as I could. From this part on, I am not sure how to proceed: "- 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."

To find a cluster of "healthy cereals", I will use only the variables related to health indicators (removing shelf and rating for example):

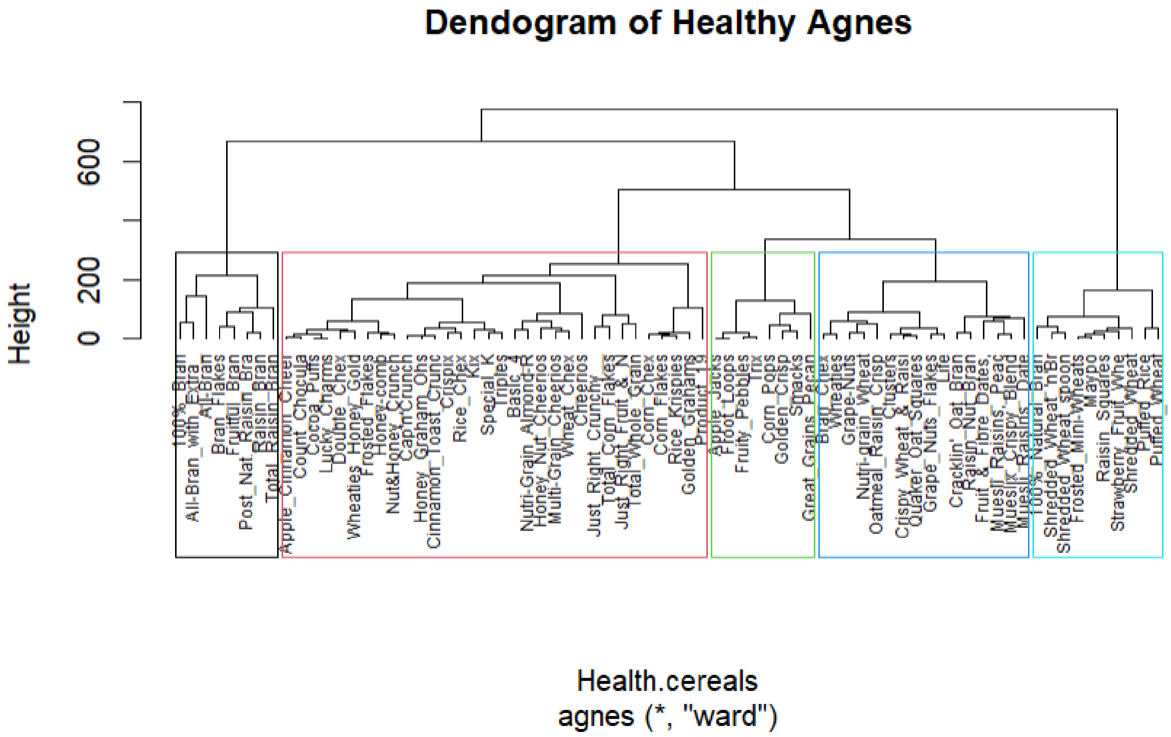
```r
Health.cereals <- Numbers[ ,c(1:9)]
rownames(Health.cereals) <- short_c.names
```

Now, doing the same process as earlier, I will try to find the best method to cluster this dataset:

```r
sg.cereals2 <- agnes(Health.cereals, method = "single")
ct.cereals2 <- agnes(Health.cereals, method = "complete")
avg.cereals2 <- agnes(Health.cereals, method = "average")
wrd.cereals2 <- agnes(Health.cereals, method = "ward")
print(sg.cereals2$ac)

## [1] 0.7406263

print(ct.cereals2$ac)

## [1] 0.926701

print(avg.cereals2$ac)

## [1] 0.8653696

print(wrd.cereals2$ac)

## [1] 0.9618369
```

“Ward” is the best method again. So, I’ll use that method to create the Agnes Dendogram:

```
Healthy.wrd <- as.hclust(wrd.cereals2)
plot(Healthy.wrd, main = "Dendogram of Healthy Agnes", cex = 0.6, hang = -1)
rect.hclust(Healthy.wrd, k = 5, border = 1:5)
```



```
Cluster.Healthy.wrd <- cutree(as.hclust(wrd.cereals2), k = 5)
Health.cereals$cluster <- Cluster.Healthy.wrd
```

Cluster 2 seems to contain the healthiest options, having less sodium, fat and sugar. For this analysis, I haven’t normalized the data. I believe a physician or a nutritionist would have more knowledge of the impacts of each variable in our bodies and would know how to weight each variable in a more appropriate manner. So, for the question of “Should the data be normalized?”, my answer would be “It depends.”