Assignment_5

Brenden Hale

```
library(analogue)
```

```
## Loading required package: vegan
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.6-4
```

```
## analogue version 0.17-6
```

```
library(cluster)
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(caret)
```

```
## Registered S3 methods overwritten by 'pROC':
##   method    from
##   print.roc analogue
##   plot.roc  analogue
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:vegan':
##
##     tolerance
```

```
library(Rfast)
```

```
## Loading required package: Rcpp
```

```
## Loading required package: RcppZiggurat
```

```
library(ISLR)
```

```
# Load cereal.csv
cereal <- read.csv("Cereals.csv")
head(cereal)
```

```
##                         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
## 5             Almond_Delight   R    C      110       2   2    200   1.0  14.0
```

```
## 6   Apple_Cinnamon_Cheerios   G    C      110       2   2    180   1.5  10.5
##   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
## 5      8     NA       25     3      1 0.75 34.38484
## 6     10     70       25     1      1 0.75 29.50954
```

```
summary(cereal)
```

```
##      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
##
```

```
set.seed(123)
```

Data preprocessing

```
# Normalizing data
rownames(cereal) <- cereal$name
cereal <- cereal[,c(-1:-3)]
cereal_scaled <- scale(cereal[,1:13])

# Remove all cereals with missing values
```

```r
cereal.norm <- na.omit(cereal_scaled)
```

Part 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.

```r
# Dissimilarity matrix
d <- dist(cereal.norm, method = "euclidean")
```

```r
# Hierarchical clustering
# Single linkage
hc_single <- agnes(d, method = "single")
hc_single$ac
```

```
## [1] 0.6094447
```

```r
# Complete linkage
hc_complete <- agnes(d, method = "complete")
hc_complete$ac
```

```
## [1] 0.8413498
```

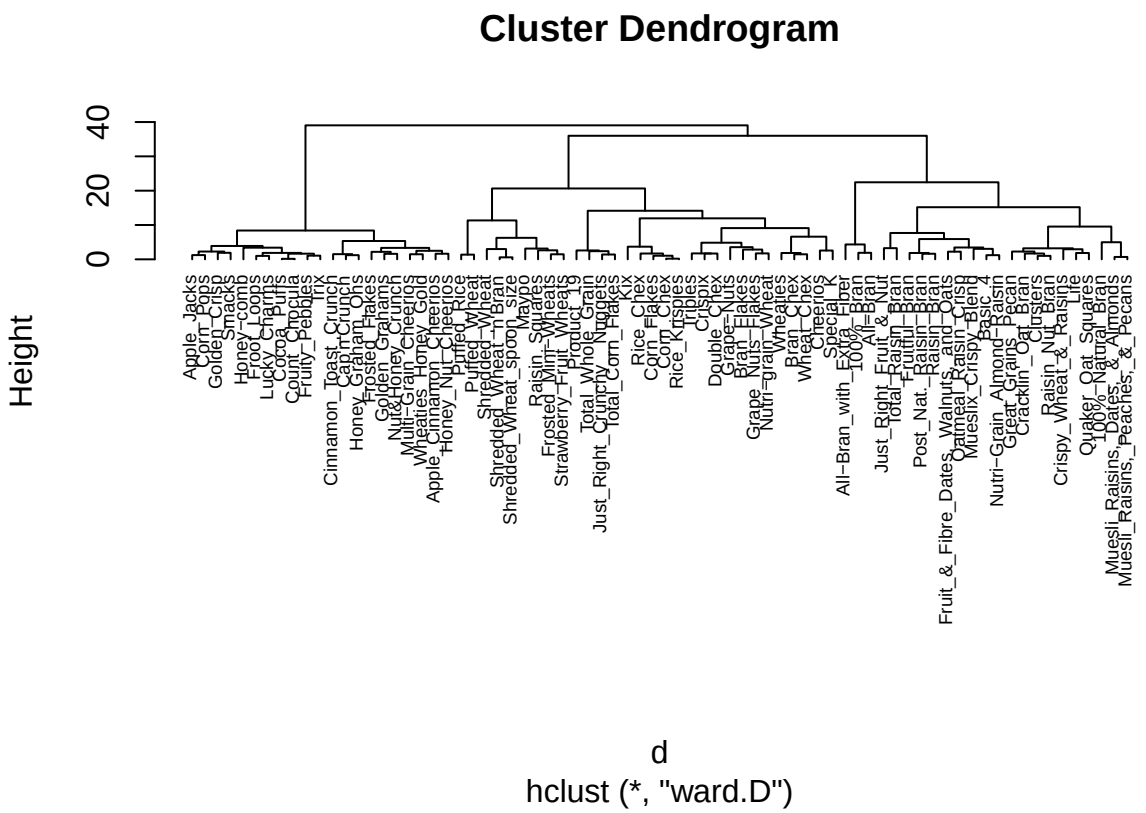
```r
# Average linkage
hc_average <- agnes(d, method = "average")
hc_average$ac
```

```
## [1] 0.7814484
```

```r
# Ward
hc_ward <- agnes(d, method = "ward")
hc_ward$ac
```

```
## [1] 0.9049881
```

```r
d_ward <- hclust(d, method = "ward.D")
plot(d_ward, cex = 0.6, hang = -1)
```

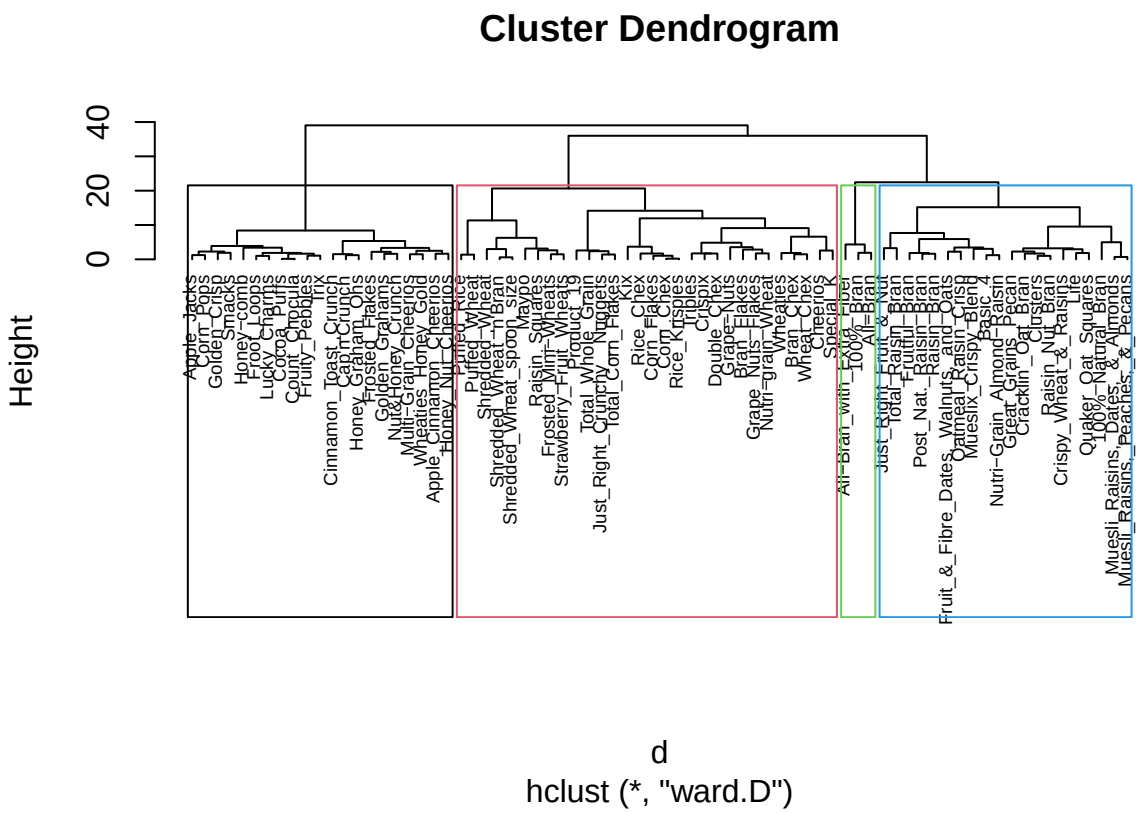



```
# The best method is the Ward method because it is the closest value to 1
```

Part 2

How many clusters would you choose?

```
plot(d_ward, cex = 0.6, hang = -1)
rect.hclust(d_ward, k = 4, border = 1:4)
```

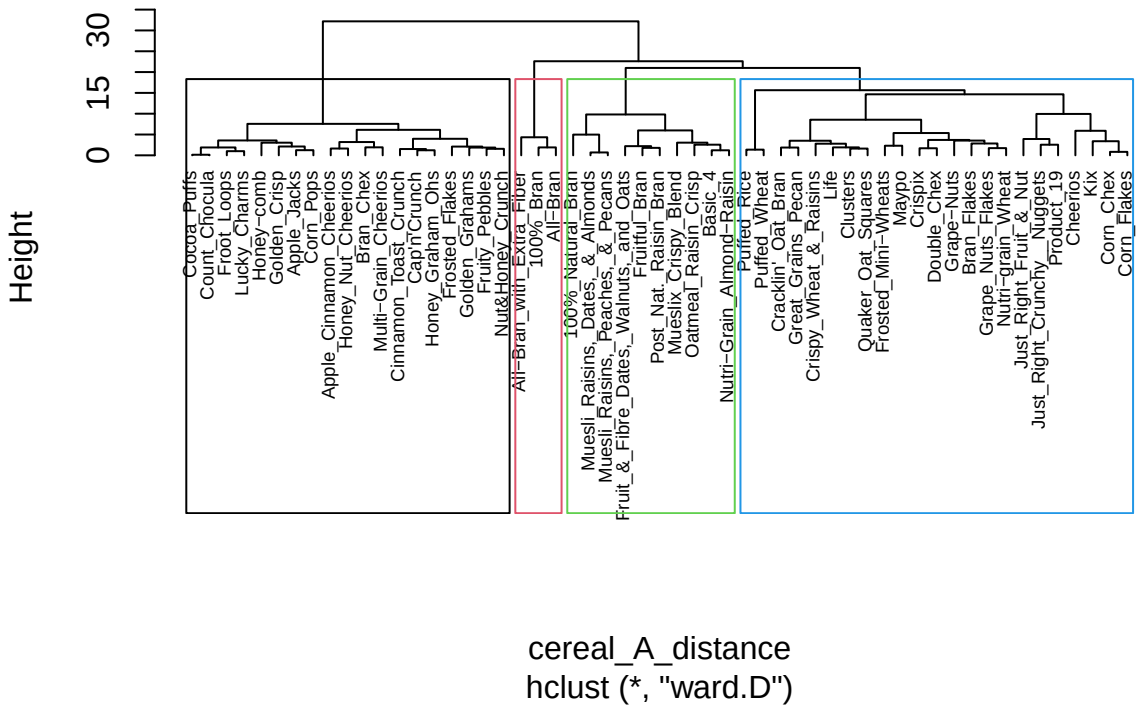



```
clusters.4 <- cutree(d_ward, k = 4)
clustered_cereal <- as.data.frame(cbind(cereal.norm, clusters.4))

# The optimal number of clusters appears to be 4 clusters
```

Part 3

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.

```
# First partition the data. Using 75% for cereal_A and 25% for cereal B
cereal_A <- cereal.norm[1:55,]
cereal_B <- cereal.norm[56:74,]

# Use cluster centroids and plot
cereal_A_distance <- dist(cereal_A, method = "euclidean")
cereal_A_hclust = hclust(cereal_A_distance, method = "ward.D")
plot(cereal_A_hclust, cex = 0.6, hang = -1)
rect.hclust(cereal_A_hclust, k = 4, border = 1:4)
```




```
clustered_cereal_A <- cutree(cereal_A_hclust, k = 4)
clusters_A <- as.data.frame(cbind(cereal_A, clustered_cereal_A))

# Identify 4 clusters
clust.1 <- colMeans(clusters_A[clusters_A$clustered_cereal_A == "1",])
clust.2 <- colMeans(clusters_A[clusters_A$clustered_cereal_A == "2",])
clust.3 <- colMeans(clusters_A[clusters_A$clustered_cereal_A == "3",])
clust.4 <- colMeans(clusters_A[clusters_A$clustered_cereal_A == "4",])

centroid <- rbind(clust.1,clust.2,clust.3,clust.4)
cluster_distance <- rowMins(distance(cereal_B, centroid[,-14]))
partition <- c(clusters_A$clustered_cereal_A, cluster_distance)
clustered_cereals_AB <- cbind(clustered_cereal, partition)

# Full data set comparison vs the test (cereals B)
table(clustered_cereals_AB$clusters.4==clustered_cereals_AB$partition)
```

```
##
## FALSE  TRUE
##     9    65
```

```
table(clustered_cereals_AB$clusters.4[56:74]==clustered_cereals_AB$partition[56:74])
```

```
##
## FALSE  TRUE
##     1    18
```

```
# As witnessed by the two comparison tables above, the stability of the clusters is 86.15% consistent w
```

Part 4

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?

```
# The data should not be normalized because it's important we analyze the actual values and not the sca

schools_cluster <- as.data.frame(cbind(na.omit(cereal),clusters.4))

colMeans(schools_cluster[schools_cluster$clusters.4==1,])
```

```
##    calories     protein         fat      sodium       fiber       carbo
##  63.3333333   4.0000000   0.6666667 176.6666667  11.0000000   6.6666667
##      sugars      potass    vitamins       shelf      weight        cups
##   3.6666667 310.0000000  25.0000000   3.0000000   1.0000000   0.3866667
##      rating  clusters.4
##  73.8444633   1.0000000
```

```
colMeans(schools_cluster[schools_cluster$clusters.4==2,])
```

```
##   calories    protein        fat     sodium      fiber      carbo     sugars
##  124.00000    3.15000    1.95000  155.00000    3.10000   13.95000    9.35000
##     potass   vitamins      shelf     weight       cups     rating clusters.4
##  151.50000   31.25000    2.90000    1.17250    0.69250   38.26161    2.00000
```

```
colMeans(schools_cluster[schools_cluster$clusters.4==3,])
```

```
##    calories     protein         fat      sodium       fiber       carbo
## 110.9523810   1.5238095   1.0000000 172.3809524   0.5714286  12.6190476
##      sugars      potass    vitamins       shelf      weight        cups
##  11.2857143  45.9523810  25.0000000   1.6666667   1.0000000   0.8871429
##      rating  clusters.4
##  28.8482485   3.0000000
```

```
colMeans(schools_cluster[schools_cluster$clusters.4==4,])
```

```
##    calories     protein         fat      sodium       fiber       carbo
##  97.3333333   2.6333333   0.4000000 158.8333333   1.8000000  17.5333333
##      sugars      potass    vitamins       shelf      weight        cups
##   3.0333333  78.8333333  30.8333333   2.0666667   0.9610000   0.9053333
##      rating  clusters.4
##  51.4311125   4.0000000
```

```
# Cluster one resembles the best cluster for schools to select. It has the lowest calories, highest pro
```