assignment_5

Hannah Cronin

2022-11-29

```
library(readr)
Cereals <- read_csv("/Users/hannahcronin/Desktop/GITHUB/64060_-HCRONIN-FML/Assignment_5/
Cereals.csv")
```

```
## Rows: 77 Columns: 16
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): name, mfr, type
## dbl (13): calories, protein, fat, sodium, fiber, carbo, sugars, potass, vita...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Cereal <- na.omit(Cereals)
head(Cereal)
```

```
## # A tibble: 6 × 16
##   name        mfr   type  calor…¹ protein   fat sodium fiber carbo sugars potass
##   <chr>       <chr> <chr>   <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>
## 1 100%_Bran   N     C          70       4     1    130  10     5        6    280
## 2 100%_Natur… Q     C         120       3     5     15   2     8        8    135
## 3 All-Bran    K     C          70       4     1    260   9     7        5    320
## 4 All-Bran_w… K     C          50       4     0    140  14     8        0    330
## 5 Apple_Cinn… G     C         110       2     2    180   1.5  10.5     10     70
## 6 Apple_Jacks K     C         110       2     0    125   1    11       14     30
## # … with 5 more variables: vitamins <dbl>, shelf <dbl>, weight <dbl>,
## #   cups <dbl>, rating <dbl>, and abbreviated variable name ¹calories
```

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

```
## Loading required package: ggplot2
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WB
a
```

```
library(caret)
```

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

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

To get rid of any rows with Nulls/NAs

```
df = scale(Cereal[4:16])
df = as.data.frame(df)
```

```
d = dist(df, method = 'euclidean')
hc_single = agnes(df, method = 'single', metric = 'euclidean')
hc_complete = agnes(df, method = 'complete', metric = 'euclidean')
hc_average = agnes(df, method = 'average', metric = 'euclidean')
hc_ward = agnes(df, method = 'ward', metric = 'euclidean')
```

I tried to use C for the above work as well, but the Dendogram had a height approaching 600 so I ommitted # the categorical variables.

Also the focus of this is on nutritional value, so the categorical variables did not hold much value.

```
print(hc_single$ac)
```

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

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

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

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

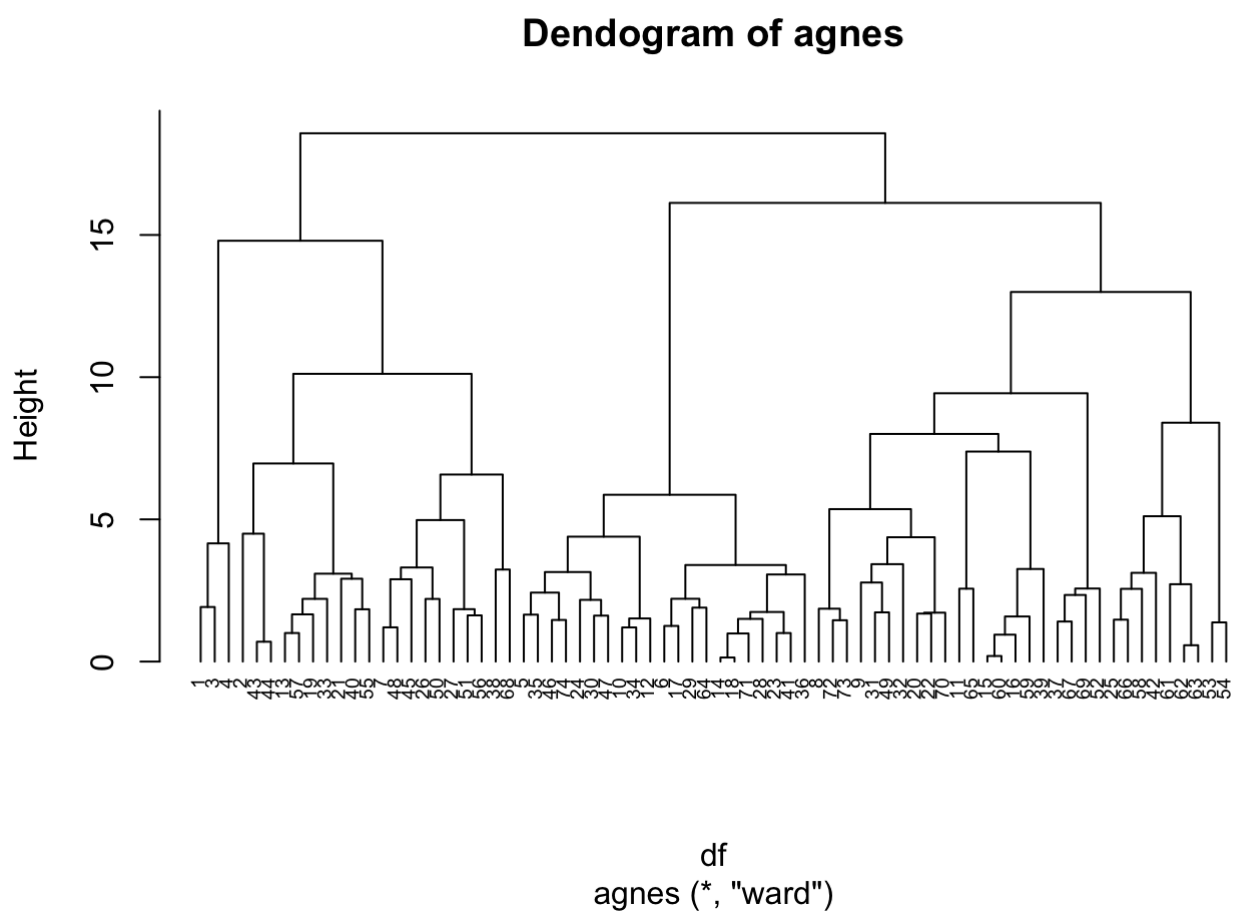
```
## [1] 0.7766075
```

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

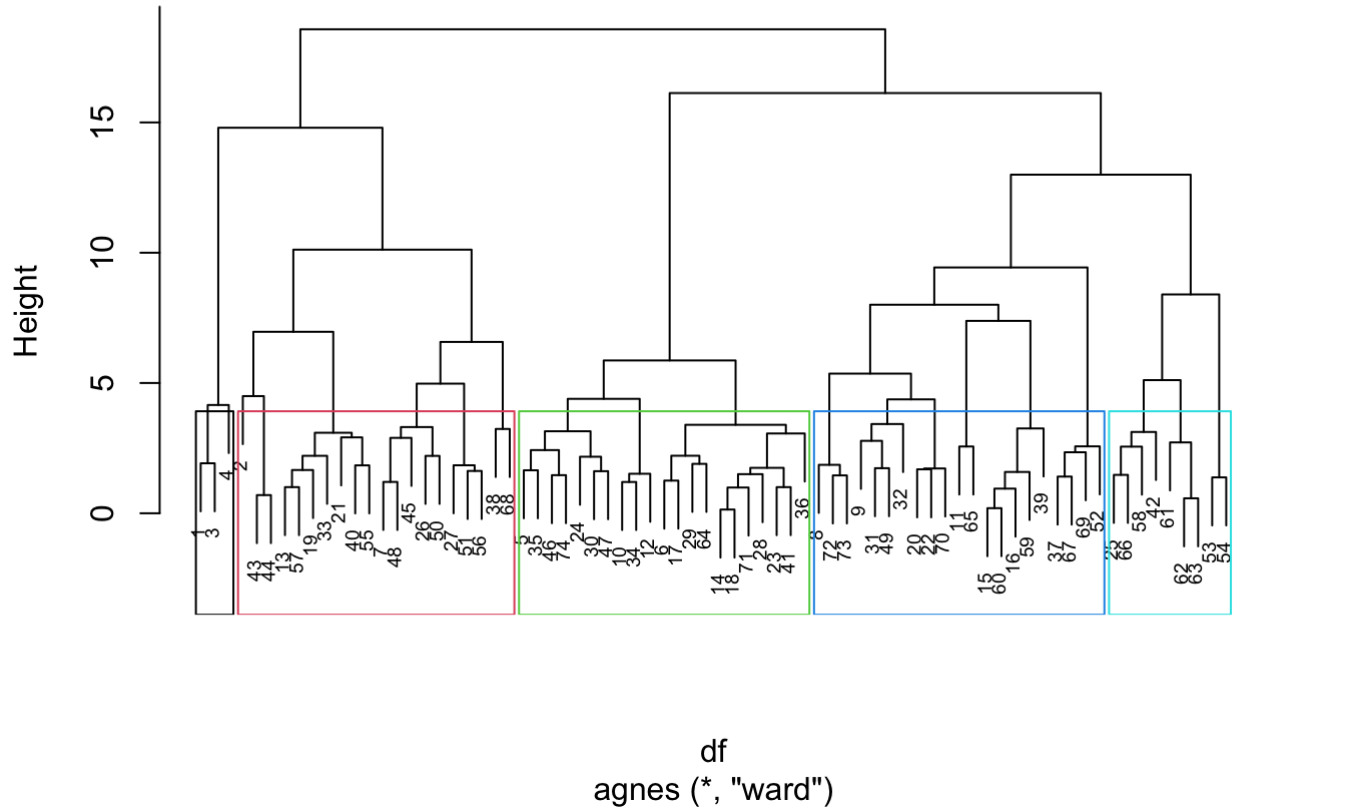
```
## [1] 0.9046042
```

The Ward method is preferred as it provides the strongest clustering structure.

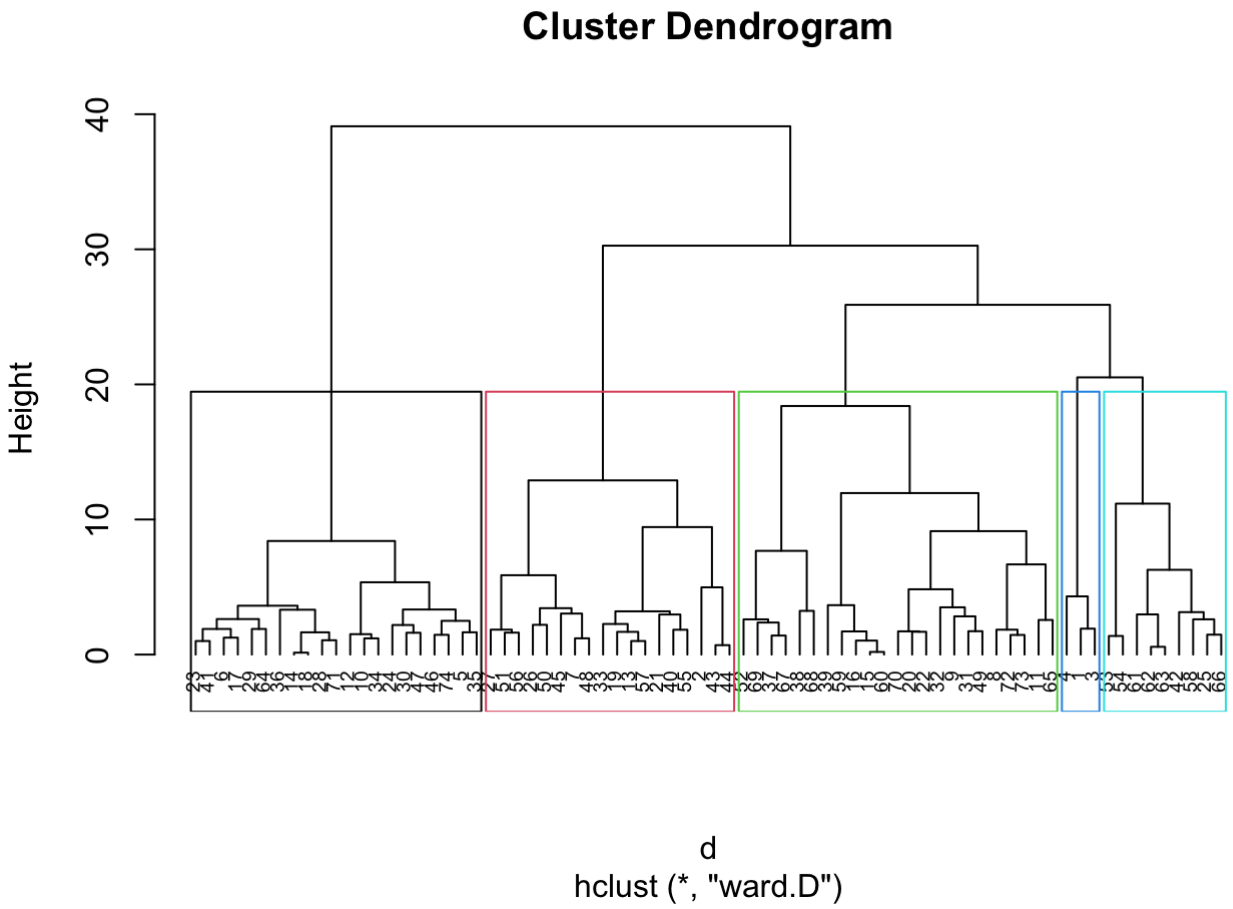
```
tree = pltree(hc_ward, cex = .6, hang = -1, main = 'Dendogram of agnes')
```




```
pltree(hc_ward, cex = 0.6)
rect.hclust(hc_ward, k = 5, border = 1:5)
```




```
hc_ward_d <- hclust(d,method = "ward.D")
plot(hc_ward_d, cex = 0.6, hang=-1)
rect.hclust(hc_ward_d, k = 5, border = 1:5)
```

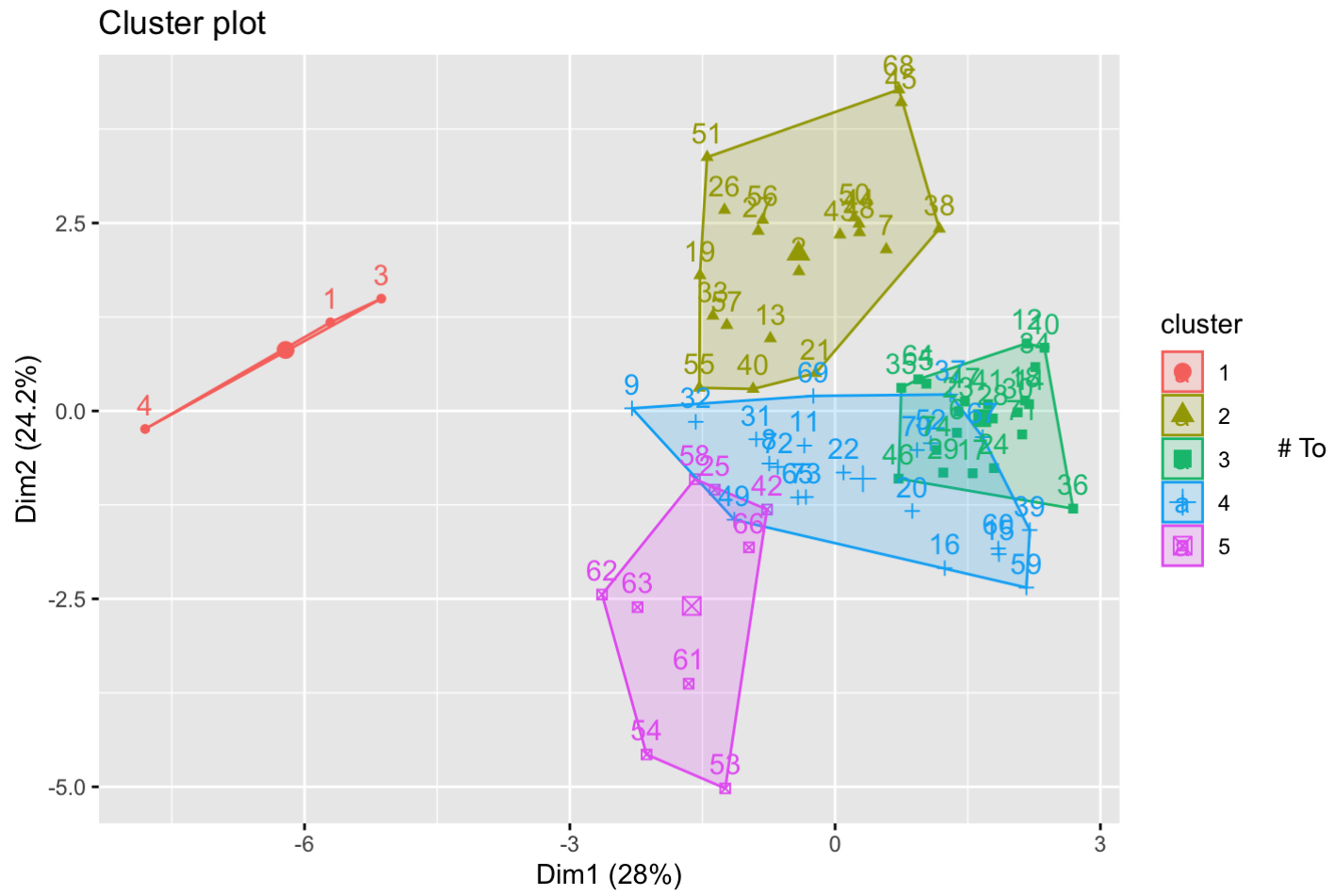


The height I've chosen is about 20- which results in 5 clusters based on Euclidean distance.I tried to build this same graph using the Agnes function however the height was always 5 or sub-5 which I didn't not believe to be an accurate distance.

```
cut = cutree(hc_ward, k = 5)
cut
```

```
##  [1] 1 2 1 1 3 3 2 4 4 3 4 3 2 3 4 4 3 3 2 4 2 4 3 3 5 2 2 3 3 3 4 4 2 3 3 3 4 2
## [39] 4 2 3 5 2 2 2 3 3 2 4 2 2 4 5 5 2 2 2 5 4 4 5 5 5 3 4 5 4 2 4 4 3 4 4 3
```

```
clusters <- fviz_cluster(list(data = df, cluster = cut))
clusters
```



partition data to check stability

```
mod = kmeans(df, centers = 5, nstart = 50)
mod$betweenss / mod$totss
```

```
## [1] 0.5252257
```

About 52.5% of data stays within initial cluster

Partitioning data

```
Test = createDataPartition(df$calories, p = .5, list = FALSE)
part1 = df[Test,]
part2 = df[-Test,]
```

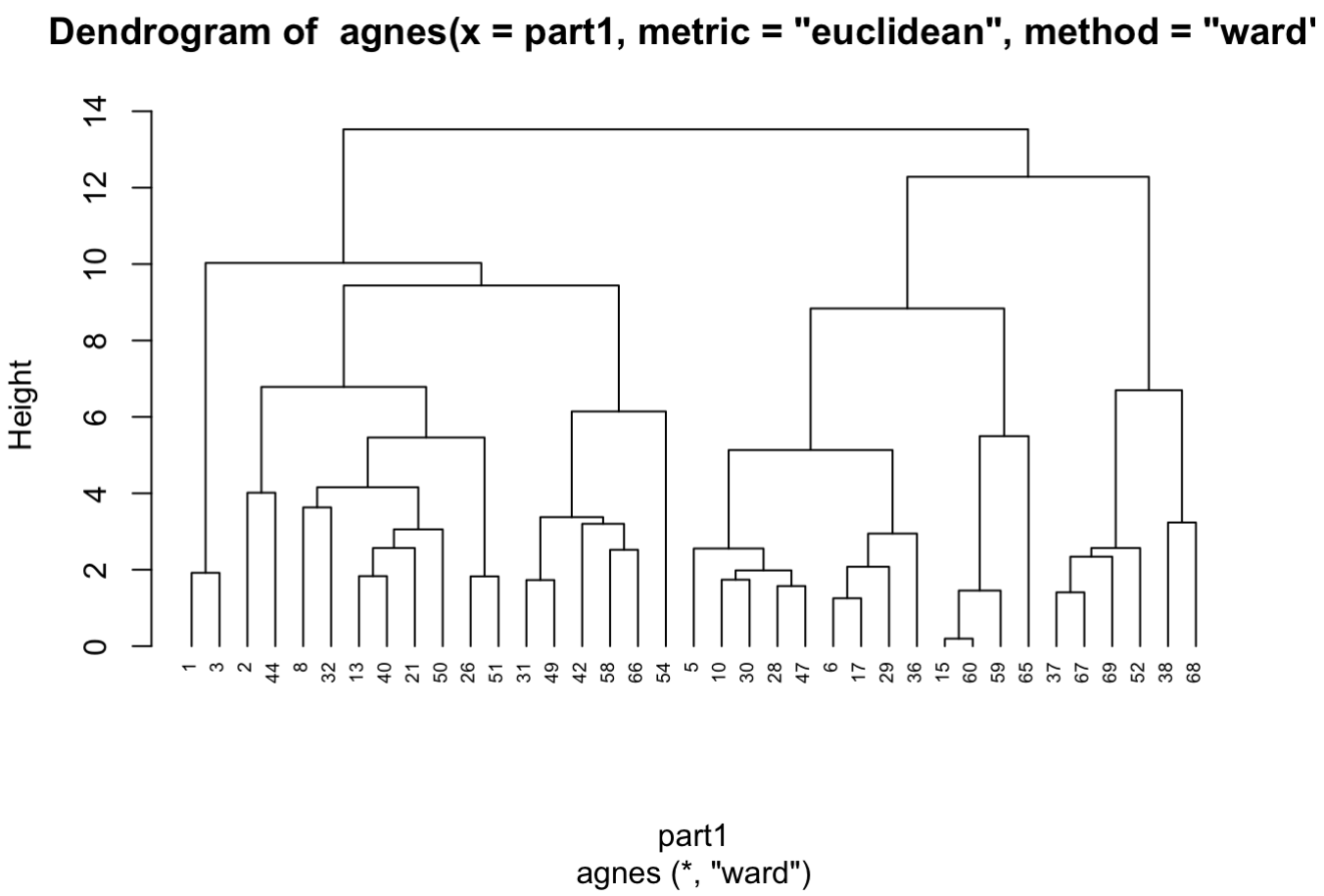
```
clust_1<- agnes(part1, method="ward", metric = "euclidean")
clust_2 <- agnes(part2, method="ward", metric = "euclidean")
print(clust_1$ac) # 0.837
```

```
## [1] 0.8238138
```

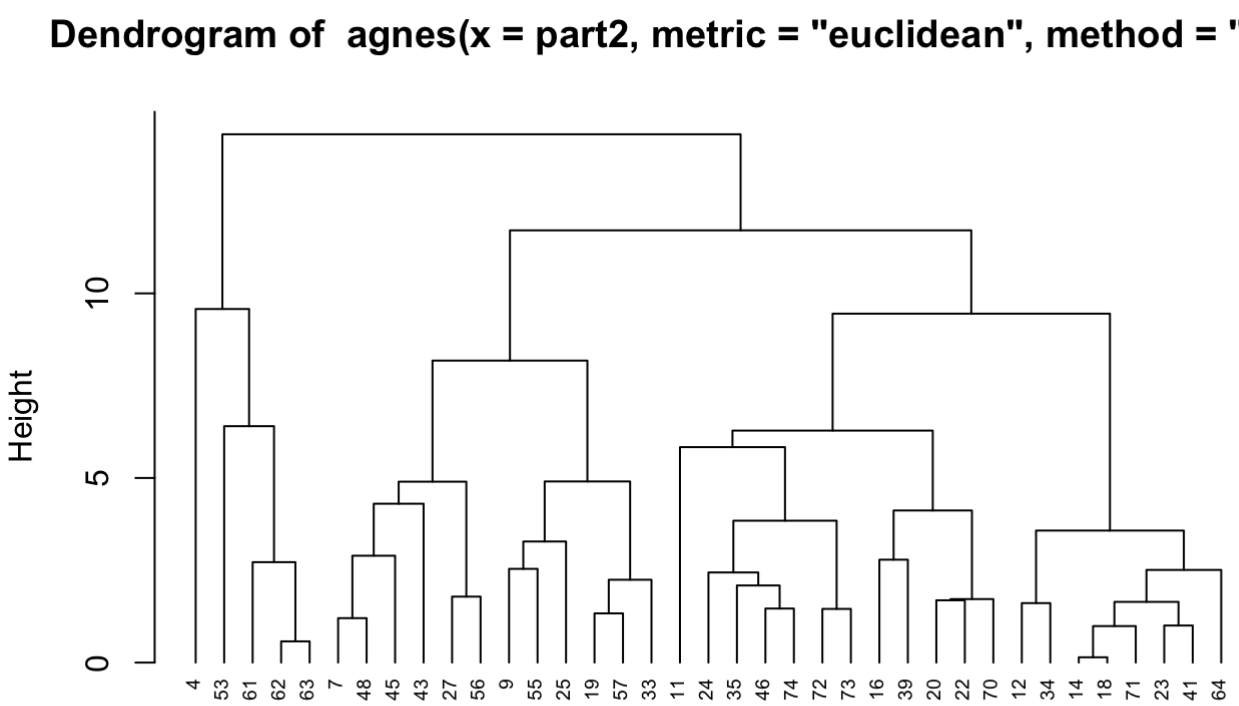
```
print(clust_2$ac) # 0.839
```

```
## [1] 0.8448
```

```
plot_1 <- pltree(clust_1, cex = 0.6, hang = -1)
```




```
plot_2 <- pltree(clust_2, cex = 0.6, hang = -1)
```



The partitioned clusters look very similar and their agglomerative coefficients values are almost identical.

To look at unscaled clusters

```
c = kmeans(Cereal[4:12], centers = 5, nstart = 50)
Cereal = data.frame(Cereal, c$cluster)
c$center
```

```
##    calories  protein       fat    sodium    fiber    carbo    sugars    potass
## 1 100.00000 3.333333 0.7777778 193.33333 7.000000 11.00000  8.666667 248.88889
## 2  91.53846 2.461538 0.7692308  16.92308 1.923077 13.76923  4.846154  86.92308
## 3 108.75000 2.437500 0.7500000 255.62500 0.843750 17.90625  4.812500  55.00000
## 4 116.31579 3.105263 1.5789474 162.63158 2.684211 15.26316  7.105263 120.26316
## 5 110.58824 1.529412 0.8823529 169.11765 0.500000 13.85294 10.176471  44.41176
##   vitamins
## 1 33.33333
## 2 13.46154
## 3 29.68750
## 4 32.89474
## 5 33.82353
```

Here we do not want to use standardized data, because the nutritional value is important to see on its own. For example, the ratio between carbohydrates and fiber (simple sugar vs complex/dietary sugars). Each factor becomes important to know rather than the scaled version. Based on these results, I'd recommend Cluster#2,

because it has the fewest calories, a moderate amount of protein, a lower amount of fat, low sodium, higher fiber, moderate amount of carbs, low sugar, some potassium, and vitamins.