Assignment_5

Matt Kostoff

2022-11-27

Data Preprocessing

```
Cereal<-read.csv("cereals.csv")
Cereal<-na.omit(Cereal)
row.names(Cereal) <- Cereal[,1]
Cereal <- Cereal[,c(-1,-2,-3)]
head(Cereal)
```

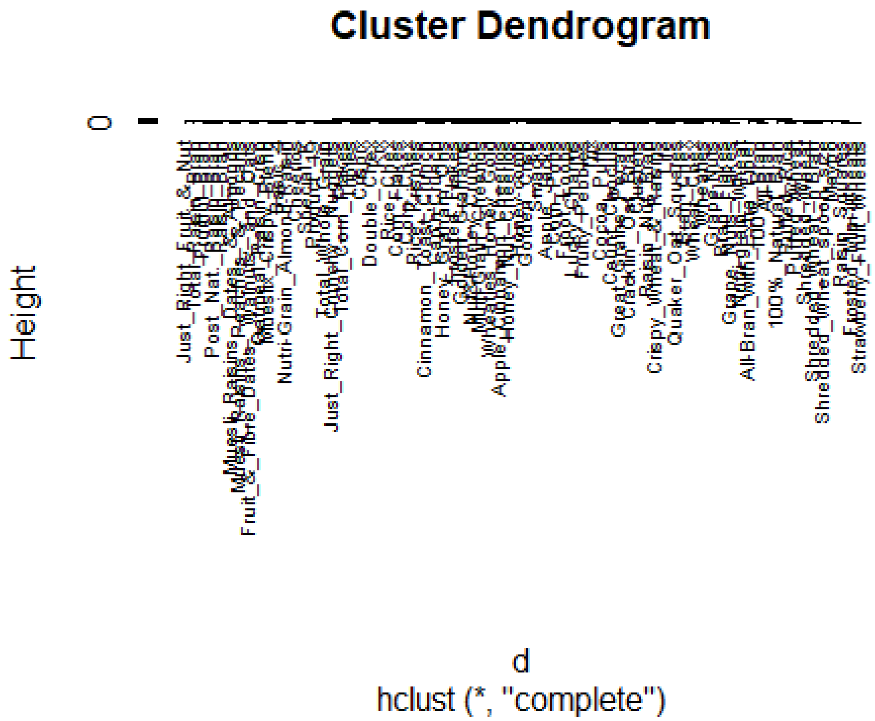
```
##                           calories protein fat sodium fiber carbo sugars
potass
## 100%_Bran                       70       4   1    130  10.0   5.0      6
280
## 100%_Natural_Bran              120       3   5     15   2.0   8.0      8
135
## All-Bran                        70       4   1    260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

```
Cereal<-scale(Cereal)
head(Cereal)
```

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

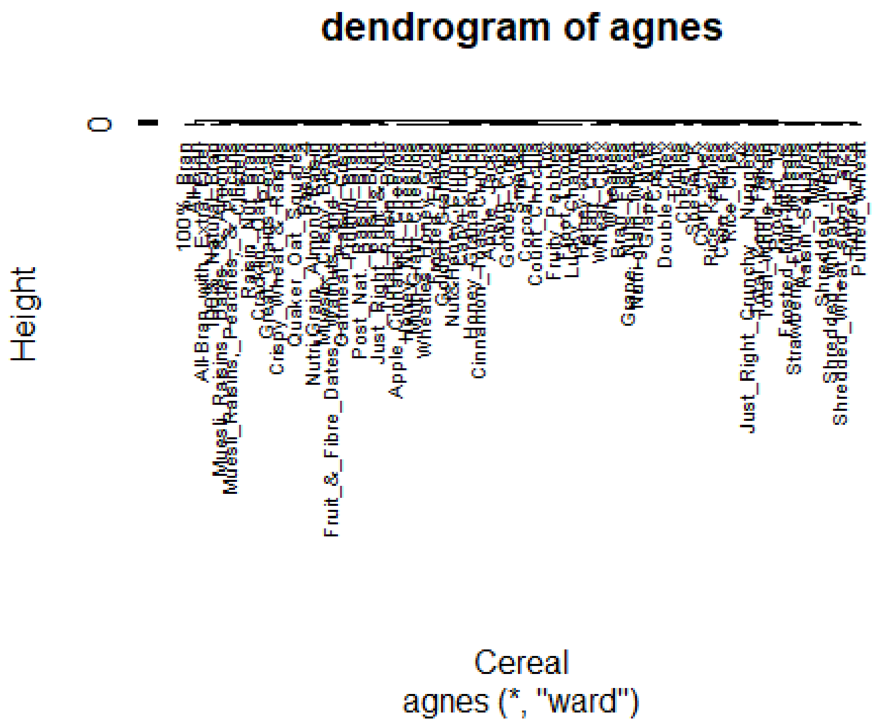
Apply hiearchical clustering

```
# Euclidean distance
d <- dist(Cereal, method = "euclidean")
hc1<-hclust(d, method = "complete")
plot(hc1, cex =0.6, hang = -1)
```

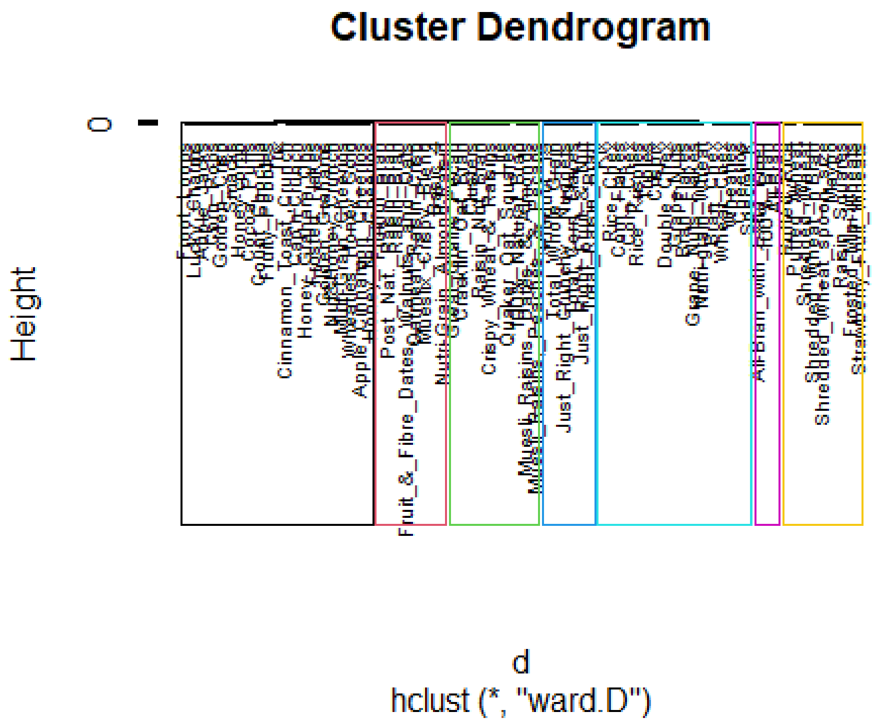


```
#Agnes
library(cluster)
```

```
hc_single<-agnes(Cereal, method = "single")
hc_complete<-agnes(Cereal, method = "complete")
hc_average<-agnes(Cereal, method = "average")
hc_ward<-agnes(Cereal, method = "ward")
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

#Ward is the best method
pltree(hc_ward, cex = 0.6, hang = -1, main = "dendrogram of agnes")
```



```
# cutting dendrograms
d <- dist(Cereal, method = "euclidean")
hc_ward<-hclust(d, method = "ward.D")
plot(hc_ward, cex=0.6)
rect.hclust(hc_ward, k=7, border = 1:7)
```



How many clusters to choose? - # based on running various K values, would choose k=7 as best fit

“Healthy Cereals” - should data be normalized? - # Generally data should be normalized as the distance measures can be sensitive to scale and highly influenced by larger scales. But, it could depend on how “healthy cereal” is defined. For example, in this definition the value for a variable such as “sugars” may need to have a larger influence in how they are clustered.