R Notebook

```
library(readr)
Cereals <- read_csv("Downloads/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.
```

```
View(Cereals)
library(stats)
library(cluster)
library(class)
library(caret)
```

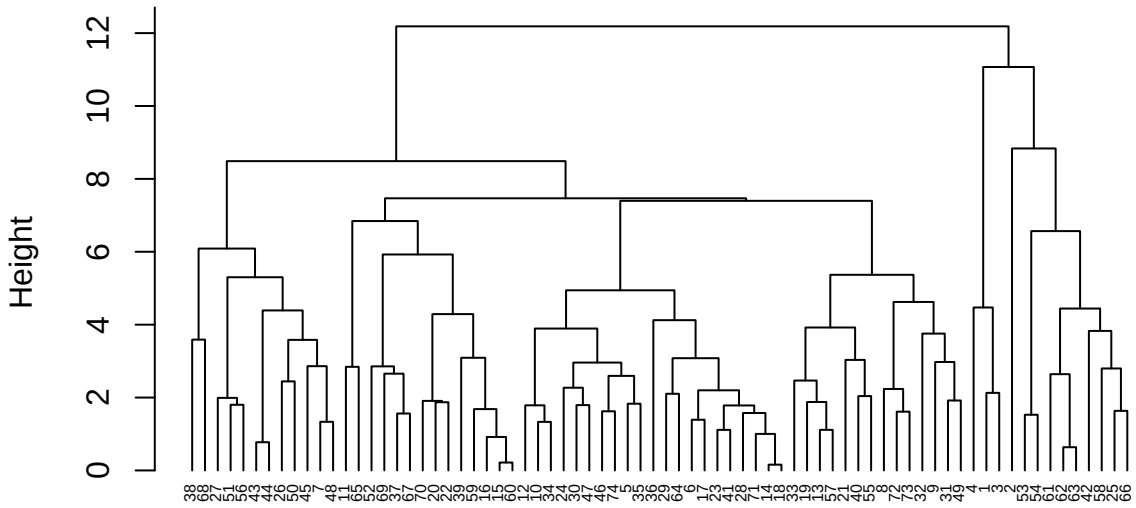
```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
Cereals <- na.omit(Cereals)
numeric_cereal<- sapply(Cereals, is.numeric)
scale_numeric <- scale(Cereals[,numeric_cereal])
scale_df <- as.data.frame(scale_numeric)
Cereals_scaled <- cbind(Cereals[!numeric_cereal], scale_df)
```

```
euclidean <- dist(Cereals_scaled, method = "euclidean")
```

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

```
euclid_cluster <- hclust(euclidean, method = "complete")
plot(euclid_cluster, cex =.5, hang =-5)
```




```
agnes_single <- agnes(Cereals_scaled, method = "single")
print(agnes_single)
```

```
## Call:     agnes(x = Cereals_scaled, method = "single")
## Agglomerative coefficient:  0.4984398
## Order of objects:
##  [1]  1  3  4  2  5  6  7  8  9 10 12 14 13 17 18 19 21 23 24 25 20 15 16 22 26
## [26] 27 28 29 30 31 32 33 34 36 35 37 38 39 41 42 40 43 44 45 48 50 46 47 51 49
## [51] 52 55 56 57 58 61 62 63 66 64 65 59 60 67 69 70 71 73 74 68 72 11 53 54
## Height (summary):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.153   3.569   4.315   4.210   5.018   7.217
##
## Available components:
## [1] "order"  "height" "ac"     "merge"  "diss"   "call"   "method" "data"
```

```
agnes_complete <- agnes(Cereals_scaled, method = "complete")
print(agnes_complete)
```

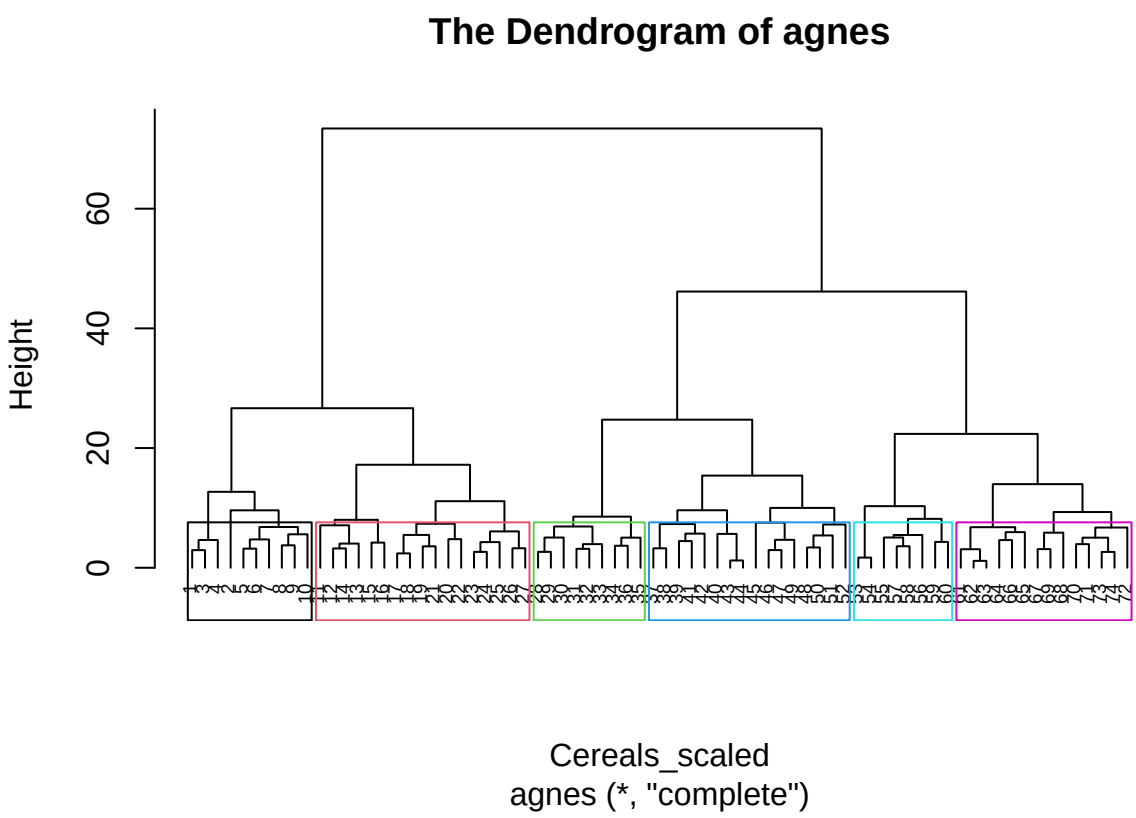
```
## Call:     agnes(x = Cereals_scaled, method = "complete")
## Agglomerative coefficient:  0.946568
## Order of objects:
##  [1]  1  3  4  2  5  6  7  8  9 10 11 12 14 13 15 16 17 18 19 21 20 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 36 35 37 38 39 41 42 40 43 44 45 46 47 49 48 50
## [51] 51 52 53 54 55 57 58 56 59 60 61 62 63 64 66 65 67 69 68 70 71 73 74 72
## Height (summary):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.153   3.596   5.059   8.023   7.537  73.397
##
```

```
## Available components:
## [1] "order"  "height" "ac"     "merge"  "diss"   "call"   "method" "data"
```

```
agnes_average <- agnes(Cereals_scaled, method = "average")
print(agnes_average)
```

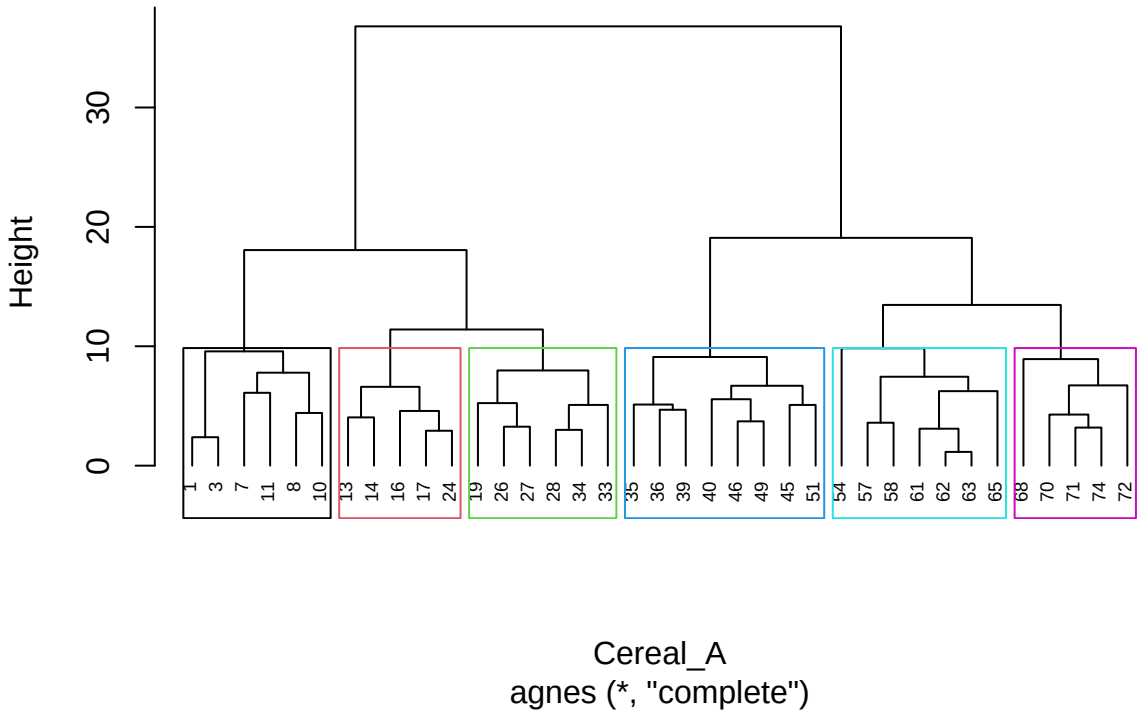
```
## Call:     agnes(x = Cereals_scaled, method = "average")
## Agglomerative coefficient:  0.8982144
## Order of objects:
##  [1]  1  3  4  2  5  6  7  8  9 10 11 12 14 13 15 16 17 18 20 19 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 36 35 37 38 39 41 42 40 43 44 45 48 50 46 47 49
## [51] 51 52 53 54 55 56 57 58 59 60 61 62 63 64 66 65 67 69 68 70 71 73 74 72
## Height (summary):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.153   3.596   4.817   6.092   6.466  37.617
##
## Available components:
## [1] "order"  "height" "ac"     "merge"  "diss"   "call"   "method" "data"
```

```
# The complete agnes was the best result, since it was the closest to one. I would choose 6 clusters si
pltree(agnes_complete, cex =.6, hang = -5, main = "The Dendrogram of agnes")
rect.hclust(agnes_complete, k = 6, border = 1:6)
```

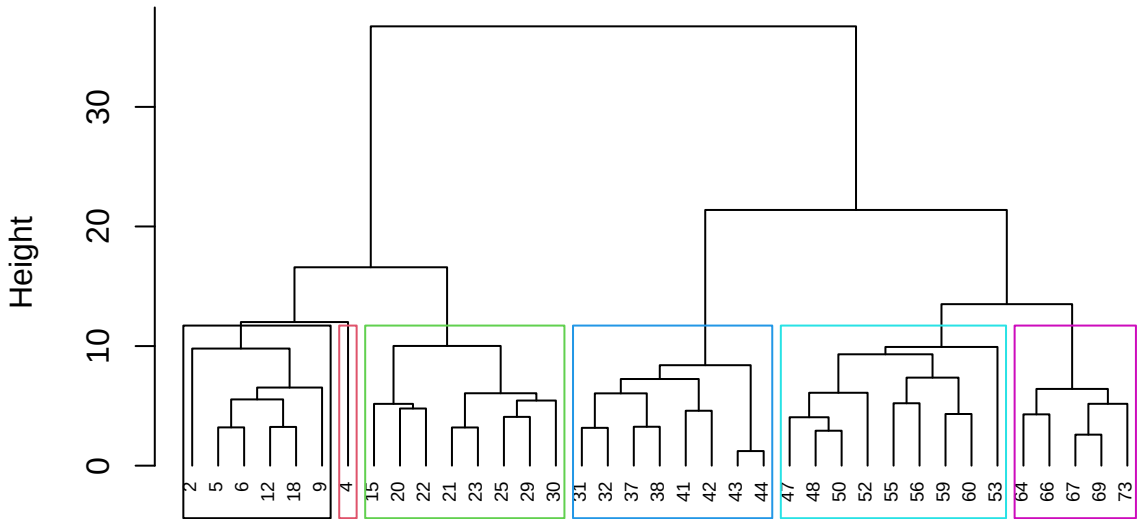



```
Cereals_index <- createDataPartition(Cereals_scaled$calories, p =.5, list = FALSE)
Cereal_A <- Cereals_scaled[Cereals_index,]
Cereal_B <- Cereals_scaled[-Cereals_index,]
```

```
Cereal_A_Agnes <- agnes(Cereal_A, method = "complete")
pltree(Cereal_A_Agnes, cex =.6, hang = -5, main = "The Dendrogram of Cereal A")
rect.hclust(Cereal_A_Agnes, k = 6, border = 1:6)
```




```
Cereal_B_Agnes <- agnes(Cereal_B, method = "complete")
pltree(Cereal_B_Agnes, cex =.6, hang = -5, main = "The Dendrogram of Cereal B")
rect.hclust(Cereal_B_Agnes, k = 6, border = 1:6)
```




```
print(Cereal_A_Agnes)
```

```
## Call:     agnes(x = Cereal_A, method = "complete")
## Agglomerative coefficient:  0.8826354
## Order of objects:
##  [1] 1  3  7  11 8  10 13 14 16 17 24 19 26 27 28 34 33 35 36 39 40 46 49 45 51
## [26] 54 57 58 61 62 63 65 68 70 71 74 72
## Height (summary):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.153   3.960   5.402   7.396   8.213  36.798
##
## Available components:
## [1] "order"     "height"    "ac"        "merge"     "diss"      "call"
## [7] "method"    "order.lab" "data"
```

```
print(Cereal_B_Agnes)
```

```
## Call:     agnes(x = Cereal_B, method = "complete")
## Agglomerative coefficient:  0.8791114
## Order of objects:
##  [1] 2  5  6  12 18 9  4  15 20 22 21 23 25 29 30 31 32 37 38 41 42 43 44 47 48
## [26] 50 52 55 56 59 60 53 64 66 67 69 73
## Height (summary):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.219   4.071   5.493   7.468   8.629  36.734
##
## Available components:
## [1] "order"     "height"    "ac"        "merge"     "diss"      "call"
## [7] "method"    "order.lab" "data"
```

```
# By looking at this, the cluster assignments look like they are very very similar between each of the
# The data should be normalized because the clusters will be very skewed if they are not. You wouldn't
```