Assignment -5

Vamshikrishna Sunnam

2023-04-16

R Markdown

```
getwd()

## [1] "C:/Users/vamsh/Downloads"

setwd("C:/Users/vamsh/Downloads")
read.csv("C:/Users/vamsh/Downloads/Cereals.csv")

##                                      name mfr type calories protein fat
sodium
## 1                               100%_Bran   N    C       70       4   1
130
## 2                       100%_Natural_Bran   Q    C      120       3   5
15
## 3                                 All-Bran   K    C       70       4   1
260
## 4                 All-Bran_with_Extra_Fiber   K    C       50       4   0
140
## 5                           Almond_Delight   R    C      110       2   2
200
## 6                   Apple_Cinnamon_Cheerios   G    C      110       2   2
180
## 7                               Apple_Jacks   K    C      110       2   0
125
## 8                                   Basic_4   G    C      130       3   2
210
## 9                                 Bran_Chex   R    C       90       2   1
200
## 10                              Bran_Flakes   P    C       90       3   0
210
## 11                            Cap'n'Crunch   Q    C      120       1   2
220
## 12                                Cheerios   G    C      110       6   2
290
## 13                  Cinnamon_Toast_Crunch   G    C      120       1   3
210
## 14                                Clusters   G    C      110       3   2
140
## 15                              Cocoa_Puffs   G    C      110       1   1
180
## 16                                Corn_Chex   R    C      110       2   0
280
```

```
## 17                            Corn_Flakes   K    C      100        2   0
290
## 18                              Corn_Pops   K    C      110        1   0
90
## 19                          Count_Chocula   G    C      110        1   1
180
## 20                      Cracklin'_Oat_Bran   K    C      110        3   3
140
## 21                  Cream_of_Wheat_(Quick)   N    H      100        3   0
80
## 22                                Crispix   K    C      110        2   0
220
## 23                  Crispy_Wheat_&_Raisins   G    C      100        2   1
140
## 24                              Double_Chex   R    C      100        2   0
190
## 25                              Froot_Loops   K    C      110        2   1
125
## 26                           Frosted_Flakes   K    C      110        1   0
200
## 27                      Frosted_Mini-Wheats   K    C      100        3   0
0
## 28 Fruit_&_Fibre_Dates,_Walnuts,_and_Oats   P    C      120        3   2
160
## 29                            Fruitful_Bran   K    C      120        3   0
240
## 30                           Fruity_Pebbles   P    C      110        1   1
135
## 31                             Golden_Crisp   P    C      100        2   0
45
## 32                           Golden_Grahams   G    C      110        1   1
280
## 33                       Grape_Nuts_Flakes   P    C      100        3   1
140
## 34                              Grape-Nuts   P    C      110        3   0
170
## 35                      Great_Grains_Pecan   P    C      120        3   3
75
## 36                        Honey_Graham_Ohs   Q    C      120        1   2
220
## 37                      Honey_Nut_Cheerios   G    C      110        3   1
250
## 38                              Honey-comb   P    C      110        1   0
180
## 39            Just_Right_Crunchy__Nuggets   K    C      110        2   1
170
## 40                 Just_Right_Fruit_&_Nut   K    C      140        3   1
170
## 41                                    Kix   G    C      110        2   1
260
```

```
## 42                                  Life   Q    C      100        4   2
150
## 43                            Lucky_Charms   G    C      110        2   1
180
## 44                                 Maypo   A    H      100        4   1
0
## 45       Muesli_Raisins,_Dates,_&_Almonds   R    C      150        4   3
95
## 46      Muesli_Raisins,_Peaches,_&_Pecans   R    C      150        4   3
150
## 47                   Mueslix_Crispy_Blend   K    C      160        3   2
150
## 48                   Multi-Grain_Cheerios   G    C      100        2   1
220
## 49                       Nut&Honey_Crunch   K    C      120        2   1
190
## 50              Nutri-Grain_Almond-Raisin   K    C      140        3   2
220
## 51                     Nutri-grain_Wheat   K    C       90        3   0
170
## 52                   Oatmeal_Raisin_Crisp   G    C      130        3   2
170
## 53                  Post_Nat._Raisin_Bran   P    C      120        3   1
200
## 54                             Product_19   K    C      100        3   0
320
## 55                            Puffed_Rice   Q    C       50        1   0
0
## 56                           Puffed_Wheat   Q    C       50        2   0
0
## 57                      Quaker_Oat_Squares   Q    C      100        4   1
135
## 58                          Quaker_Oatmeal   Q    H      100        5   2
0
## 59                             Raisin_Bran   K    C      120        3   1
210
## 60                        Raisin_Nut_Bran   G    C      100        3   2
140
## 61                         Raisin_Squares   K    C       90        2   0
0
## 62                               Rice_Chex   R    C      110        1   0
240
## 63                          Rice_Krispies   K    C      110        2   0
290
## 64                         Shredded_Wheat   N    C       80        2   0
0
## 65                  Shredded_Wheat_'n'Bran   N    C       90        3   0
0
## 66              Shredded_Wheat_spoon_size   N    C       90        3   0
0
```

```
## 67                                Smacks   K    C      110       2   1
70
## 68                             Special_K   K    C      110       6   0
230
## 69                 Strawberry_Fruit_Wheats   N    C       90       2   0
15
## 70                      Total_Corn_Flakes   G    C      110       2   1
200
## 71                      Total_Raisin_Bran   G    C      140       3   1
190
## 72                      Total_Whole_Grain   G    C      100       3   1
200
## 73                               Triples   G    C      110       2   1
250
## 74                                  Trix   G    C      110       1   1
140
## 75                             Wheat_Chex   R    C      100       3   1
230
## 76                               Wheaties   G    C      100       3   1
200
## 77                     Wheaties_Honey_Gold   G    C      110       2   1
200
##    fiber carbo sugars potass vitamins shelf weight cups   rating
## 1   10.0   5.0      6    280       25     3   1.00 0.33 68.40297
## 2    2.0   8.0      8    135        0     3   1.00 1.00 33.98368
## 3    9.0   7.0      5    320       25     3   1.00 0.33 59.42551
## 4   14.0   8.0      0    330       25     3   1.00 0.50 93.70491
## 5    1.0  14.0      8     NA       25     3   1.00 0.75 34.38484
## 6    1.5  10.5     10     70       25     1   1.00 0.75 29.50954
## 7    1.0  11.0     14     30       25     2   1.00 1.00 33.17409
## 8    2.0  18.0      8    100       25     3   1.33 0.75 37.03856
## 9    4.0  15.0      6    125       25     1   1.00 0.67 49.12025
## 10   5.0  13.0      5    190       25     3   1.00 0.67 53.31381
## 11   0.0  12.0     12     35       25     2   1.00 0.75 18.04285
## 12   2.0  17.0      1    105       25     1   1.00 1.25 50.76500
## 13   0.0  13.0      9     45       25     2   1.00 0.75 19.82357
## 14   2.0  13.0      7    105       25     3   1.00 0.50 40.40021
## 15   0.0  12.0     13     55       25     2   1.00 1.00 22.73645
## 16   0.0  22.0      3     25       25     1   1.00 1.00 41.44502
## 17   1.0  21.0      2     35       25     1   1.00 1.00 45.86332
## 18   1.0  13.0     12     20       25     2   1.00 1.00 35.78279
## 19   0.0  12.0     13     65       25     2   1.00 1.00 22.39651
## 20   4.0  10.0      7    160       25     3   1.00 0.50 40.44877
## 21   1.0  21.0      0     NA        0     2   1.00 1.00 64.53382
## 22   1.0  21.0      3     30       25     3   1.00 1.00 46.89564
## 23   2.0  11.0     10    120       25     3   1.00 0.75 36.17620
## 24   1.0  18.0      5     80       25     3   1.00 0.75 44.33086
## 25   1.0  11.0     13     30       25     2   1.00 1.00 32.20758
## 26   1.0  14.0     11     25       25     1   1.00 0.75 31.43597
## 27   3.0  14.0      7    100       25     2   1.00 0.80 58.34514
```

```
## 28   5.0  12.0     10    200       25     3   1.25 0.67 40.91705
## 29   5.0  14.0     12    190       25     3   1.33 0.67 41.01549
## 30   0.0  13.0     12     25       25     2   1.00 0.75 28.02576
## 31   0.0  11.0     15     40       25     1   1.00 0.88 35.25244
## 32   0.0  15.0      9     45       25     2   1.00 0.75 23.80404
## 33   3.0  15.0      5     85       25     3   1.00 0.88 52.07690
## 34   3.0  17.0      3     90       25     3   1.00 0.25 53.37101
## 35   3.0  13.0      4    100       25     3   1.00 0.33 45.81172
## 36   1.0  12.0     11     45       25     2   1.00 1.00 21.87129
## 37   1.5  11.5     10     90       25     1   1.00 0.75 31.07222
## 38   0.0  14.0     11     35       25     1   1.00 1.33 28.74241
## 39   1.0  17.0      6     60      100     3   1.00 1.00 36.52368
## 40   2.0  20.0      9     95      100     3   1.30 0.75 36.47151
## 41   0.0  21.0      3     40       25     2   1.00 1.50 39.24111
## 42   2.0  12.0      6     95       25     2   1.00 0.67 45.32807
## 43   0.0  12.0     12     55       25     2   1.00 1.00 26.73451
## 44   0.0  16.0      3     95       25     2   1.00 1.00 54.85092
## 45   3.0  16.0     11    170       25     3   1.00 1.00 37.13686
## 46   3.0  16.0     11    170       25     3   1.00 1.00 34.13976
## 47   3.0  17.0     13    160       25     3   1.50 0.67 30.31335
## 48   2.0  15.0      6     90       25     1   1.00 1.00 40.10596
## 49   0.0  15.0      9     40       25     2   1.00 0.67 29.92429
## 50   3.0  21.0      7    130       25     3   1.33 0.67 40.69232
## 51   3.0  18.0      2     90       25     3   1.00 1.00 59.64284
## 52   1.5  13.5     10    120       25     3   1.25 0.50 30.45084
## 53   6.0  11.0     14    260       25     3   1.33 0.67 37.84059
## 54   1.0  20.0      3     45      100     3   1.00 1.00 41.50354
## 55   0.0  13.0      0     15        0     3   0.50 1.00 60.75611
## 56   1.0  10.0      0     50        0     3   0.50 1.00 63.00565
## 57   2.0  14.0      6    110       25     3   1.00 0.50 49.51187
## 58   2.7    NA     NA    110        0     1   1.00 0.67 50.82839
## 59   5.0  14.0     12    240       25     2   1.33 0.75 39.25920
## 60   2.5  10.5      8    140       25     3   1.00 0.50 39.70340
## 61   2.0  15.0      6    110       25     3   1.00 0.50 55.33314
## 62   0.0  23.0      2     30       25     1   1.00 1.13 41.99893
## 63   0.0  22.0      3     35       25     1   1.00 1.00 40.56016
## 64   3.0  16.0      0     95        0     1   0.83 1.00 68.23588
## 65   4.0  19.0      0    140        0     1   1.00 0.67 74.47295
## 66   3.0  20.0      0    120        0     1   1.00 0.67 72.80179
## 67   1.0   9.0     15     40       25     2   1.00 0.75 31.23005
## 68   1.0  16.0      3     55       25     1   1.00 1.00 53.13132
## 69   3.0  15.0      5     90       25     2   1.00 1.00 59.36399
## 70   0.0  21.0      3     35      100     3   1.00 1.00 38.83975
## 71   4.0  15.0     14    230      100     3   1.50 1.00 28.59278
## 72   3.0  16.0      3    110      100     3   1.00 1.00 46.65884
## 73   0.0  21.0      3     60       25     3   1.00 0.75 39.10617
## 74   0.0  13.0     12     25       25     2   1.00 1.00 27.75330
## 75   3.0  17.0      3    115       25     1   1.00 0.67 49.78744
## 76   3.0  17.0      3    110       25     1   1.00 1.00 51.59219
## 77   1.0  16.0      8     60       25     1   1.00 0.75 36.18756
```

```r
# installing required packages
library(ISLR)
library(caret)

## Loading required package: ggplot2

## 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

library(cluster)
library(factoextra)

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

library(NbClust)
library(ppclust)

## Warning: package 'ppclust' was built under R version 4.2.3

library(dendextend)

##
## ---------------------
## Welcome to dendextend version 1.16.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at:
https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:
suppressPackageStartupMessages(library(dendextend))
## ---------------------

##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ lubridate 1.9.2     ✓ tibble    3.1.8
## ✓ purrr     1.0.1     ✓ tidyr     1.3.0
## ✓ readr     2.1.4
```

```
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ purrr::lift()   masks caret::lift()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force
all conflicts to become errors
```

```
library(ggplot2)
library(proxy)
```

```
##
## Attaching package: 'proxy'
##
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
##
## The following object is masked from 'package:base':
##
##     as.matrix
```

```
# To import the data set "cereal"
Cereals <- read.csv("Cereals.csv")
# Using head getting the first few rows of the data collection
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
```

```
## 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

# Using str to examine the data set's organization
str(Cereals)

## 'data.frame':    77 obs. of  16 variables:
##  $ name    : chr  "100%_Bran" "100%_Natural_Bran" "All-Bran" "All-
Bran_with_Extra_Fiber" ...
##  $ mfr     : chr  "N" "Q" "K" "K" ...
##  $ type    : chr  "C" "C" "C" "C" ...
##  $ calories: int  70 120 70 50 110 110 110 130 90 90 ...
##  $ protein : int  4 3 4 4 2 2 2 3 2 3 ...
##  $ fat     : int  1 5 1 0 2 2 0 2 1 0 ...
##  $ sodium  : int  130 15 260 140 200 180 125 210 200 210 ...
##  $ fiber   : num  10 2 9 14 1 1.5 1 2 4 5 ...
##  $ carbo   : num  5 8 7 8 14 10.5 11 18 15 13 ...
##  $ sugars  : int  6 8 5 0 8 10 14 8 6 5 ...
##  $ potass  : int  280 135 320 330 NA 70 30 100 125 190 ...
##  $ vitamins: int  25 0 25 25 25 25 25 25 25 25 ...
##  $ shelf   : int  3 3 3 3 3 1 2 3 1 3 ...
##  $ weight  : num  1 1 1 1 1 1 1 1.33 1 1 ...
##  $ cups    : num  0.33 1 0.33 0.5 0.75 0.75 1 0.75 0.67 0.67 ...
##  $ rating  : num  68.4 34 59.4 93.7 34.4 ...

# utilizing the summary to analyze the data set
summary(Cereals)

##      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
##
```

Now I'm scaling the data to remove any NA values from the set.

```
# For planning purposes I'm creating a duplicate of this data collection
here.
Scaled_Cereals <- Cereals
# I'm scaling the data set right now to fit it into a clustering method.
Scaled_Cereals[ , c(4:16)] <- scale(Cereals[ , c(4:16)])
# Here, I'm removing the NA values from the data collection using the omit
function.
Preprocessed_Cereal <- na.omit(Scaled_Cereals)
# using head to display the first few rows after removing NA
head(Preprocessed_Cereal)

##                         name mfr type   calories    protein         fat
## 1                  100%_Bran   N    C -1.8929836  1.3286071 -0.01290349
## 2          100%_Natural_Bran   Q    C  0.6732089  0.4151897  3.96137277
## 3                   All-Bran   K    C -1.8929836  1.3286071 -0.01290349
## 4 All-Bran_with_Extra_Fiber   K    C -2.9194605  1.3286071 -1.00647256
## 6   Apple_Cinnamon_Cheerios   G    C  0.1599704 -0.4982277  0.98066557
## 7                Apple_Jacks   K    C  0.1599704 -0.4982277 -1.00647256
##       sodium       fiber      carbo     sugars     potass   vitamins
shelf
## 1 -0.3539844  3.29284661 -2.5087829 -0.2343906  2.5753685 -0.1453172
0.9515734
## 2 -1.7257708 -0.06375361 -1.7409943  0.2223705  0.5160205 -1.2642598
0.9515734
## 3  1.1967306  2.87327158 -1.9969238 -0.4627711  3.1434645 -0.1453172
0.9515734
## 4 -0.2346986  4.97114672 -1.7409943 -1.6046739  3.2854885 -0.1453172
0.9515734
## 6  0.2424445 -0.27354112 -1.1011705  0.6791317 -0.4071355 -0.1453172 -
```

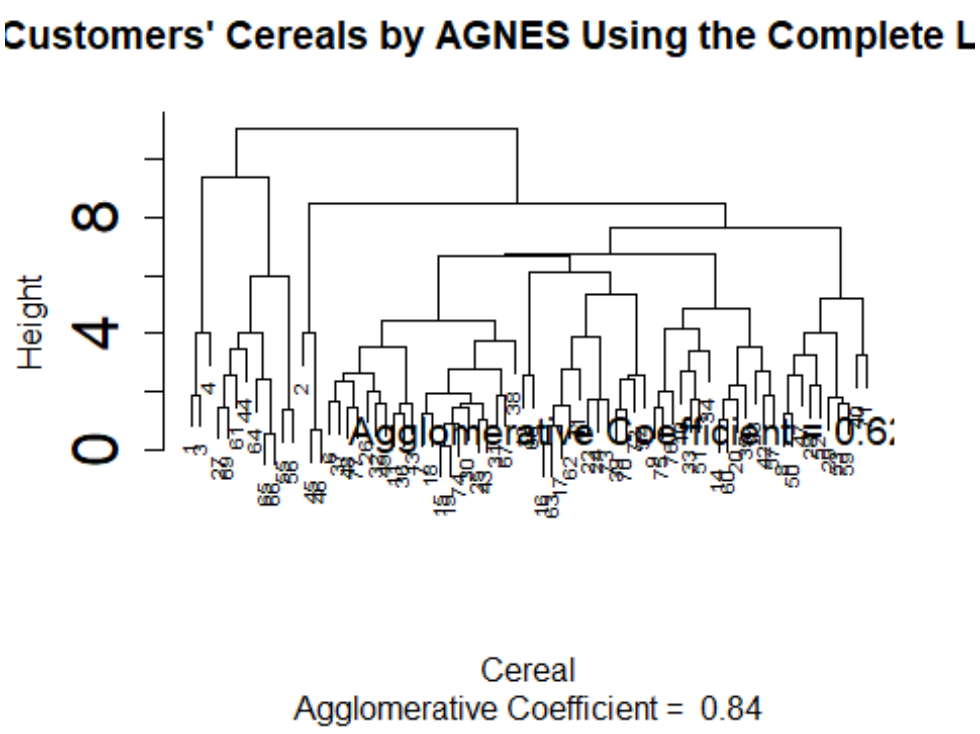
```
1.4507595
## 7 -0.4136273 -0.48332864 -0.9732057  1.5926539 -0.9752315 -0.1453172 -
0.2495930
##       weight       cups     rating
## 1 -0.1967771 -2.1100340  1.8321876
## 2 -0.1967771  0.7690100 -0.6180571
## 3 -0.1967771 -2.1100340  1.1930986
## 4 -0.1967771 -1.3795303  3.6333849
## 6 -0.1967771 -0.3052601 -0.9365625
## 7 -0.1967771  0.7690100 -0.6756899
```

The total number of observations dropped from 77 to 74 after pre-processing and scaling the data. Only 3 records had the value "NA".

Q) 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.

Single Linkage: The single linkage method produces a dendrogram that has long chains of connected data points, which are also known as "chaining." These long chains are caused by the fact that the single linkage method merges clusters based on the two closest data points. However, this method can also create spurious clusters because of its sensitivity to noise in the data.

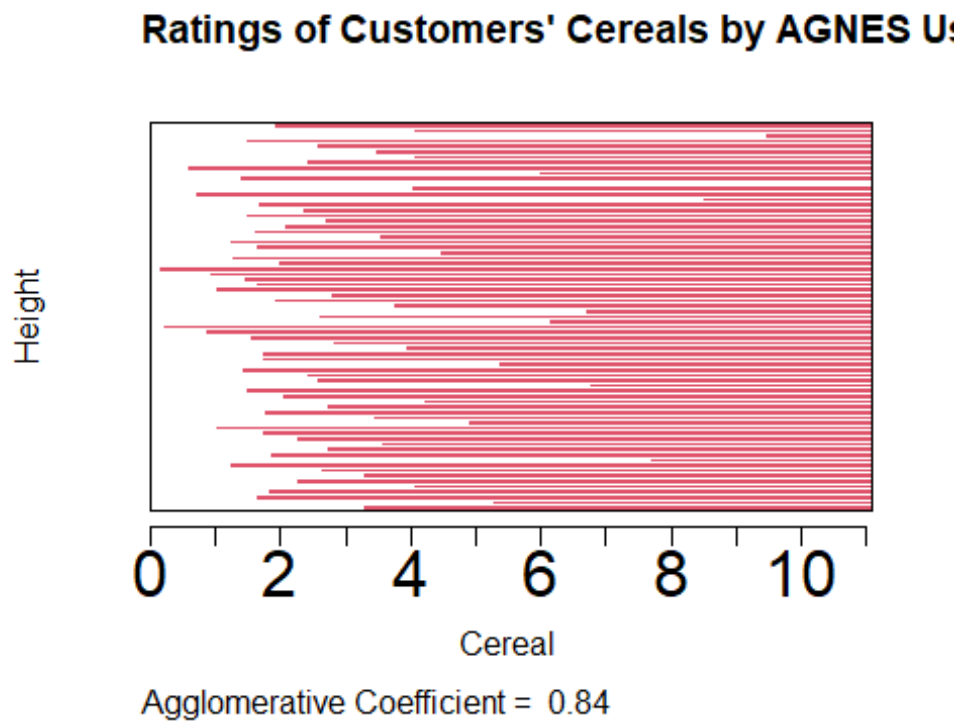
```
# The dissimilarity matrix is produced using Euclidean distance calculations
for each numerical value in the data set.
Cereal_Euclidean <- dist(Preprocessed_Cereal[ , c(4:16)], method =
"euclidean")
# Using the complete linkage method, a hierarchical clustering is carried
out.
HC_Complete <- agnes(Cereal_Euclidean, method = "complete")
# Here, I'm displaying the results of different strategies.
plot(HC_Complete,
     main = "Ratings of Customers' Cereals by AGNES Using the Complete
Linkage Method",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 2,
     cex = 0.60,
     which.plots = 2)  # 2 means to plot the dendrogram and the agglomerative
coefficient
text(x = 16, y = 0.62, labels = "Agglomerative Coefficient = 0.62", col =
"Black", pos = 4, cex = 1.2)
```

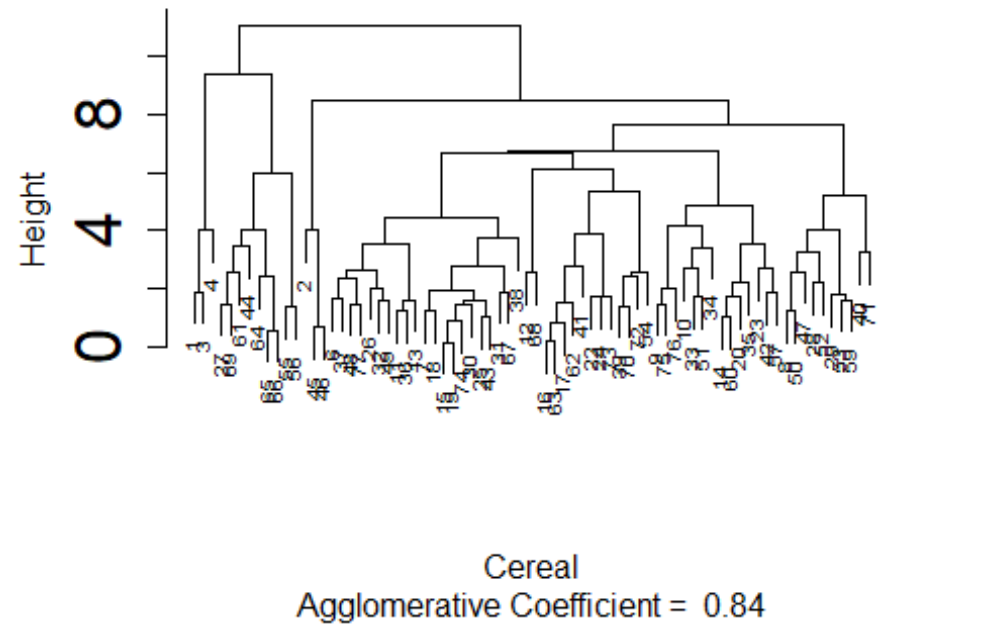


Complete Linkage:The dendrogram obtained using the complete linkage method shows that there are three main clusters. The first cluster contains mostly hot cereals, while the second cluster contains a mix of hot and cold cereals. The third cluster contains mostly cold cereals.

It's worth noting that the dendrogram appears to be less refined than that produced using the single linkage method, and the clusters appear to be less well-defined.

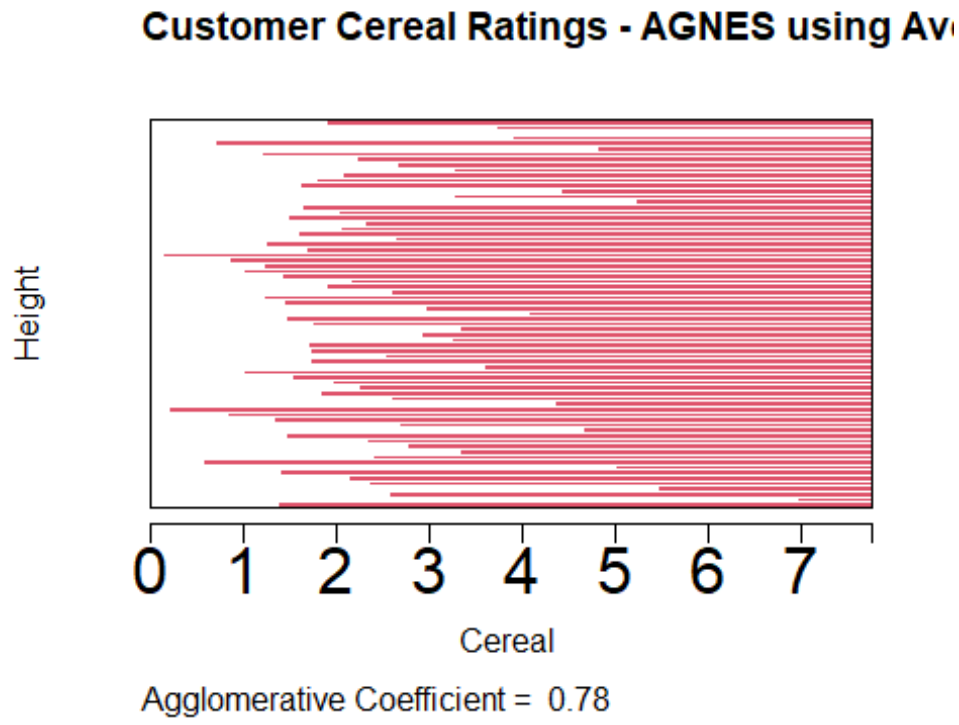
```
# utilizing all linking techniques to produce hierarchical clustering
HC_Complete <- agnes(Cereal_Euclidean, method = "complete")
# Here, I'm displaying the results of different strategies.
plot(HC_Complete,
     main = "Ratings of Customers' Cereals by AGNES Using the Complete
Linkage Method",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 2,
     cex = 0.60)
```

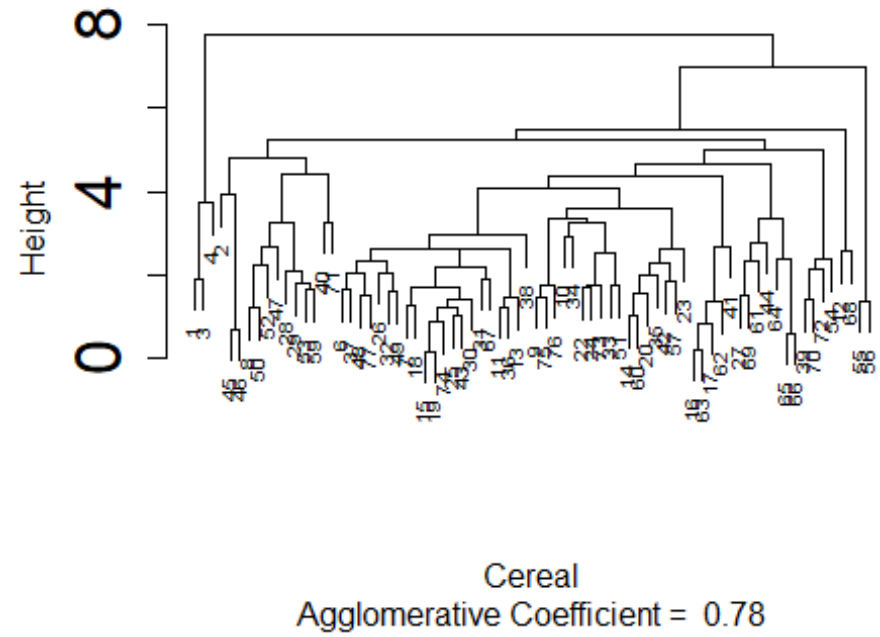

Ratings of Customers' Cereals by AGNES U
Height
0
2
4
6
8
10
Cereal
Agglomerative Coefficient = 0.84


Customers' Cereals by AGNES Using the Complete L
Height
0
4
8
Cereal
Agglomerative Coefficient = 0.84

Average Linkage: Average linkage is a hierarchical clustering method that computes the average dissimilarity between all pairs of observations in different clusters. It is based on the average distance between each point in one cluster and every point in the other cluster. This linkage method is often preferred when the clusters in the data have different sizes and densities, as it tends to create more balanced clusters.

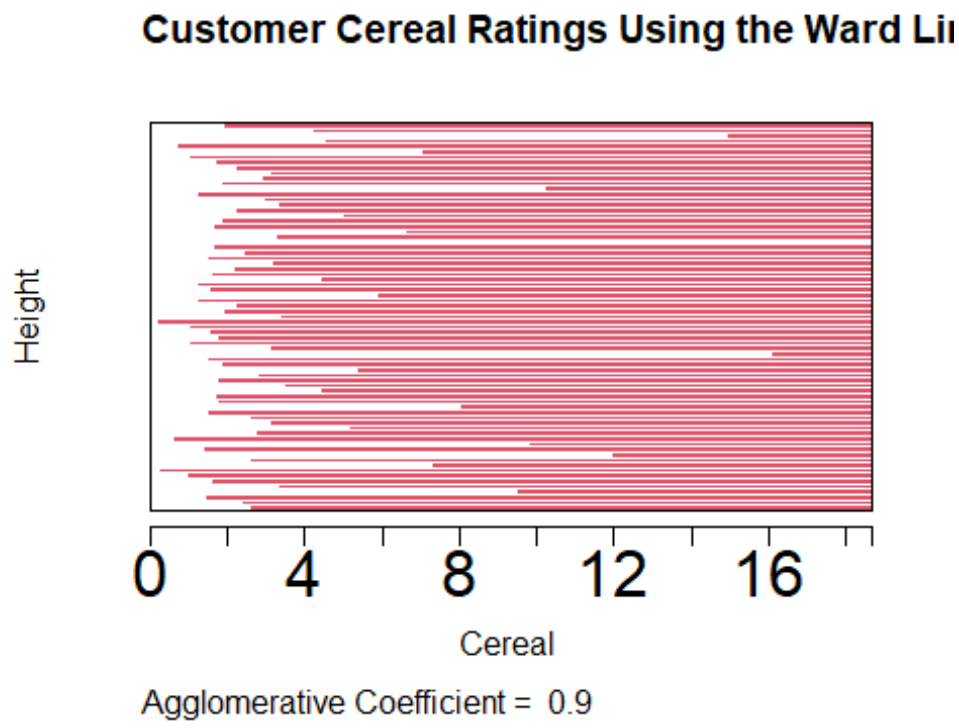
```
# Performing the average linkage method for hierarchical clustering
HC_Average <- agnes(Cereal_Euclidean, method = "average")
# Here I am Plotting the results of the different methods
plot(HC_Average,
     main = "Customer Cereal Ratings - AGNES using Average Linkage Method",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 2,
     cex = 0.60)
```

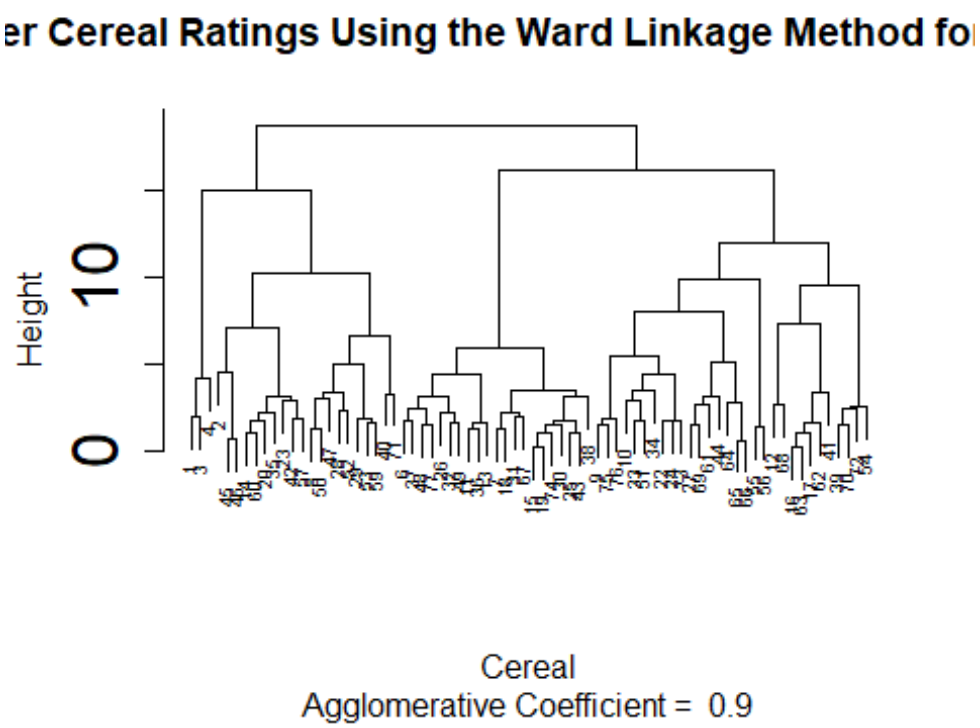

Customer Cereal Ratings - AGNES using Av
Height
0 1 2 3 4 5 6 7
Cereal
Agglomerative Coefficient = 0.78


omer Cereal Ratings - AGNES using Average Linkag
Height
0 4 8
Cereal
Agglomerative Coefficient = 0.78

Ward Method: Ward linkage is a hierarchical clustering method that minimizes the variance within each cluster. The method is based on a measure of the increase in variance when two clusters are merged, and it aims to minimize this increase at each step of the clustering process. Ward linkage is often used when the goal is to identify compact, homogeneous clusters with small variances.

```
# Performing the ward linkage method for hierarchical clustering
HC_Ward <- agnes(Cereal_Euclidean, method = "ward")
# I am  Plotting the outcomes of the different methods
plot(HC_Ward,
     main = "Customer Cereal Ratings Using the Ward Linkage Method for the
AGNES",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 2,
     cex = 0.56)
```





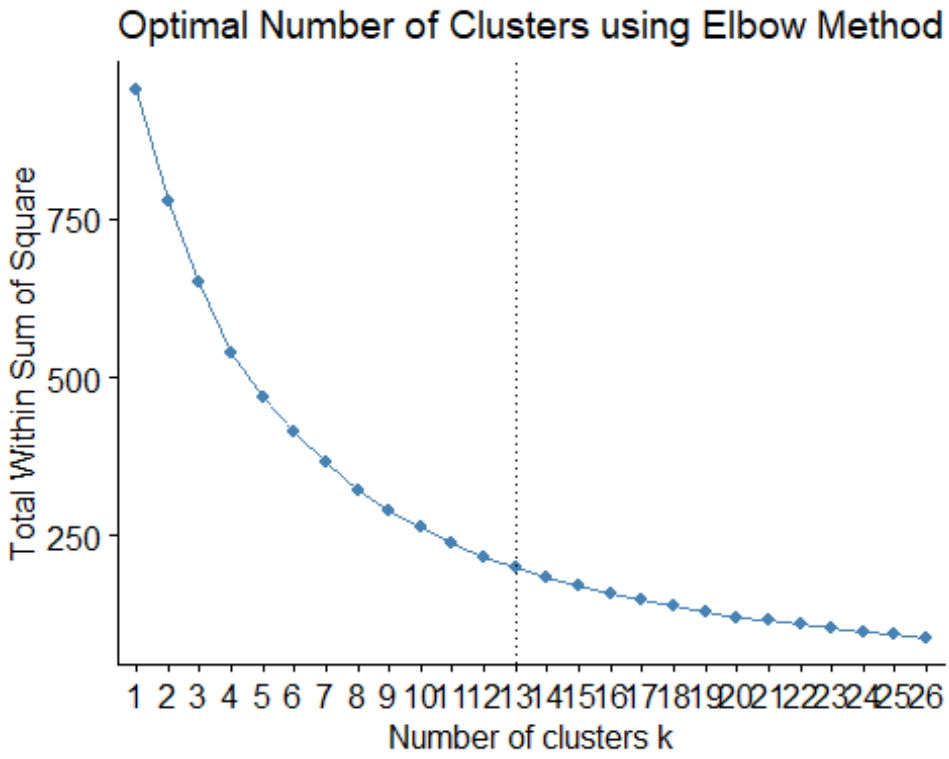
The clustering structure is closer if the value is close to 1.0. Therefore, the method with the value closest to 1.0 will be chosen. Single Linkage: 0.62 Complete Linkage: 0.85 Average Linkage: 0.78 Ward Method: 0.91 The Ward method is the best clustering model based on the results in this case.

Q) How many clusters would you choose?

Here, I'm calculating the right number of clusters using the elbow and silhouette methods.

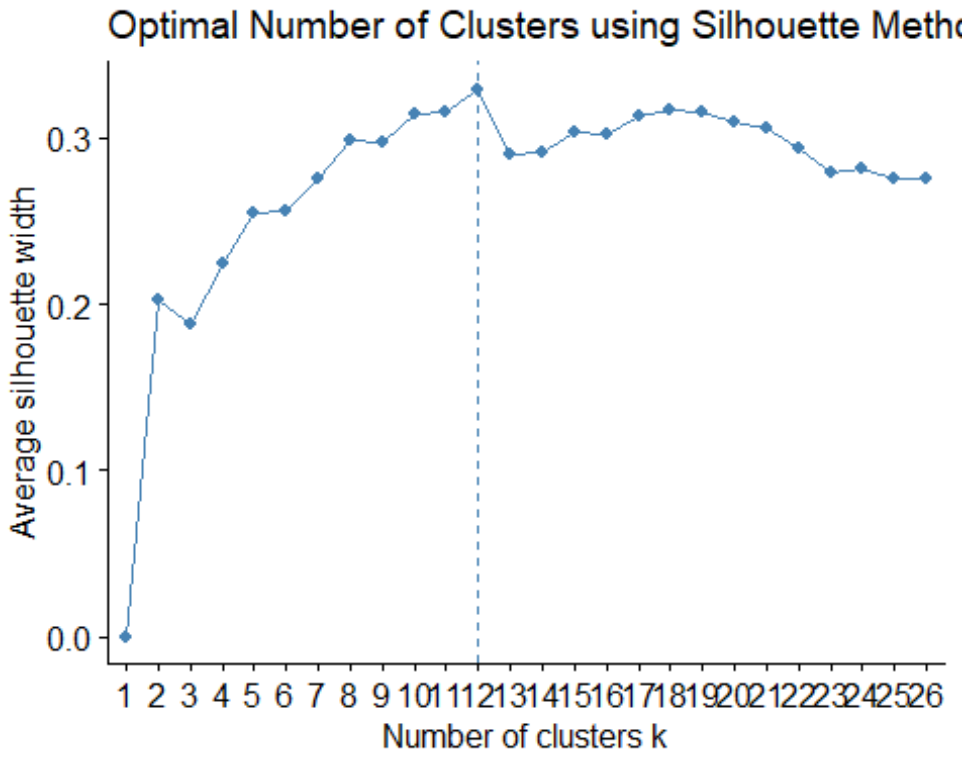
Elbow Method:

```
fviz_nbclust(Preprocessed_Cereal[ , c(4:16)], hcut, method = "wss", k.max =
26) +
  labs(title = "Optimal Number of Clusters using Elbow Method") +
  geom_vline(xintercept = 13, linetype = 3)
```



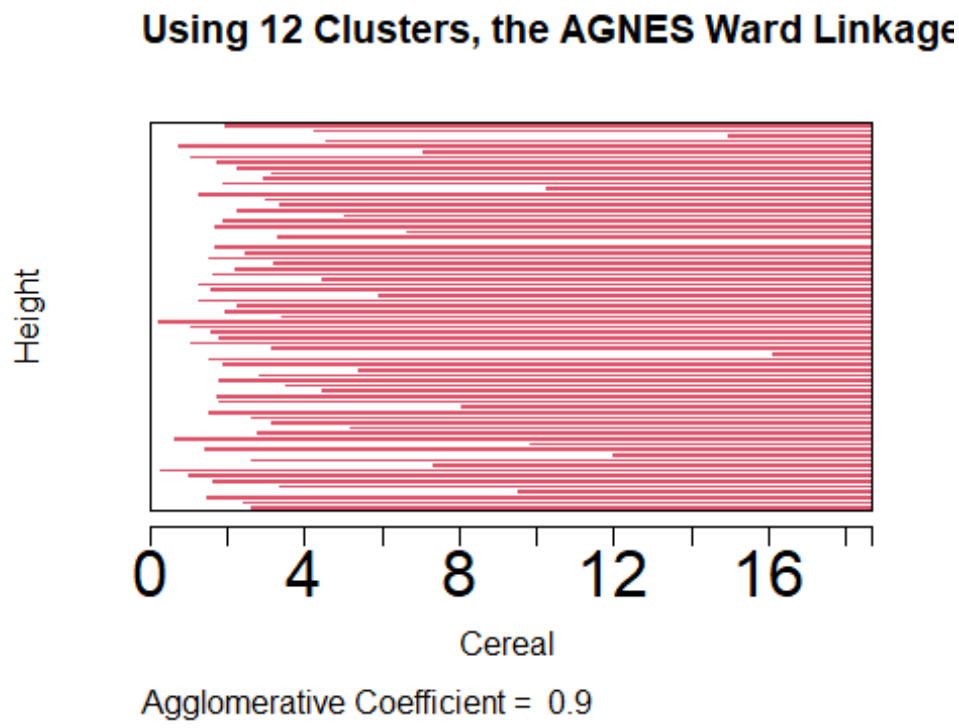
##Silhouette Method:

```
fviz_nbclust(Preprocessed_Cereal[ , c(4:16)],
                                  hcut,
                                  method = "silhouette",
                                  k.max = 26) +
labs(title = "Optimal Number of Clusters using Silhouette Method")
```

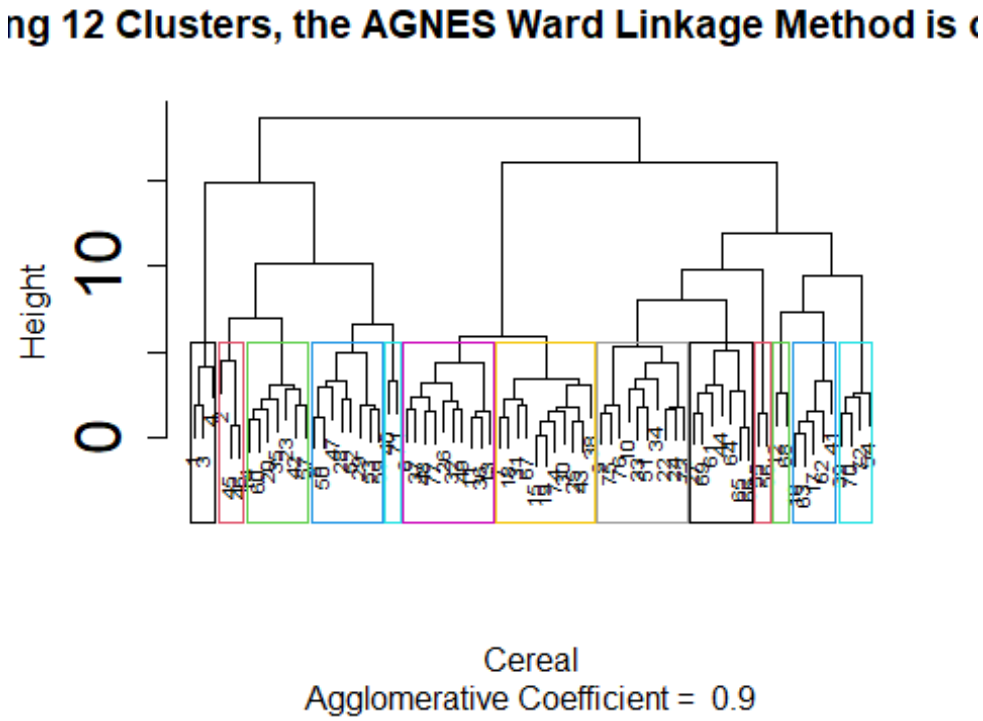


The results of the elbow and silhouette approaches suggest that the ideal number of clusters would be twelve.

```
# I've highlighted the 12 groups in this Ward hierarchical tree plot for easy reference.
plot(HC_Ward,
     main = "Using 12 Clusters, the AGNES Ward Linkage Method is outlined",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 2,
     cex = 0.60,)
```




```
rect.hclust(HC_Ward, k = 13, border = 1:13)
```



Q) 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?

Because the nutritional information for cereal is standardized based on the sample of cereal being evaluated, normalizing the data would not be appropriate in this situation. Standardizing the nutritional information for cereal based on the sample being evaluated makes normalization inappropriate in this case. Normalizing the data could exclude cereals that are extremely high in sugar and low in fiber, iron, or other nutrients. Normalizing also makes it challenging to predict the nutritional value of a cereal for a child. For instance, a cereal with an iron content of 0.999 may be the best of the worst in the sample set and provide no nutritional value. Therefore, using the ratio of the daily recommended amounts of nutrients for a child would be a better way to preprocess the data. This would prevent a few significant variables from overriding the distance estimates and enable analysts to make more informed cluster decisions. An analyst may determine what portion of a student's daily nutritional needs would be met by a cereal cluster's average when examining the clusters. This approach would help employees choose "healthy" cereal clusters more thoughtfully.