Periandri\_Anthony\_Assignment 5

2025-06-23

library(cluster)  
library(factoextra)

## Loading required package: ggplot2

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

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

library(mclust)

## Package 'mclust' version 6.1.1  
## Type 'citation("mclust")' for citing this R package in publications.

library(dplyr)

##   
## Attaching package: 'dplyr'

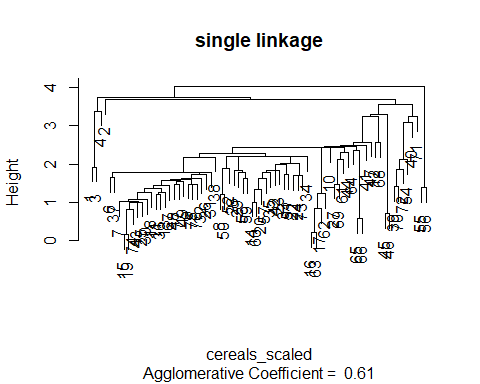
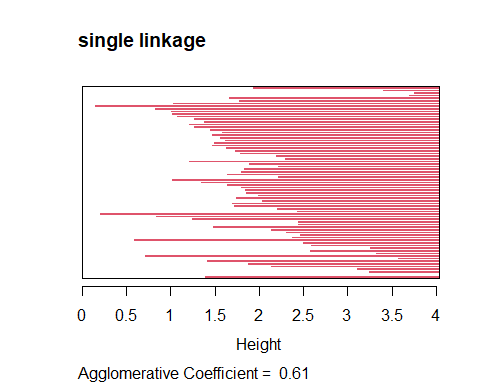
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

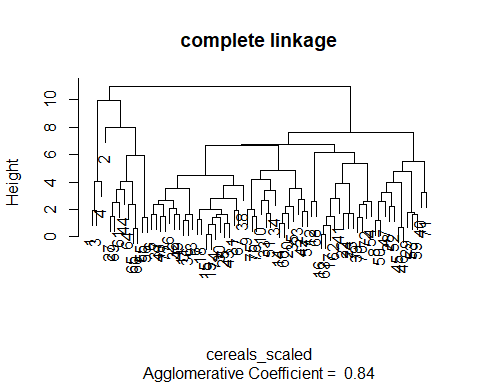
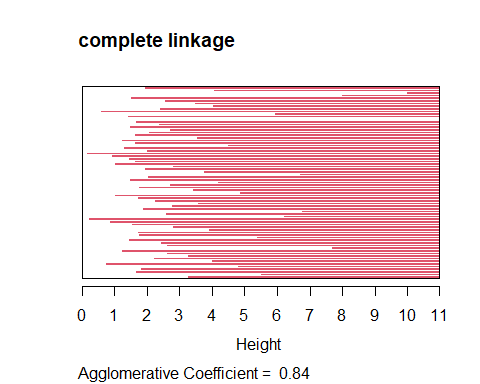
cereals <- read.csv("Cereals.csv", stringsAsFactors = TRUE)  
cereals\_clean <- na.omit(cereals)  
##show dataa  
print(cereals\_clean)

## 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  
## 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  
## 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  
## 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  
## 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  
## 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  
## 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

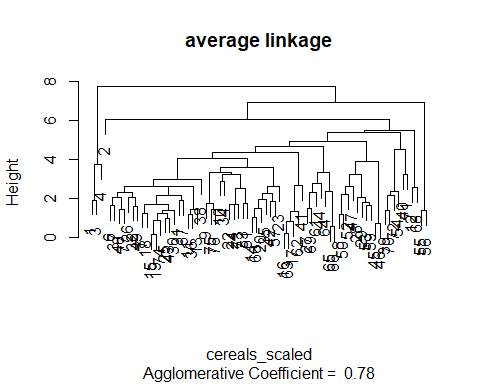
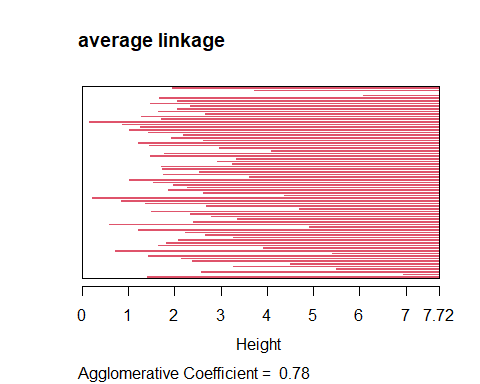
cereals\_num <- cereals\_clean[sapply(cereals\_clean, is.numeric)]  
##normalize dataset  
cereals\_scaled <- scale(cereals\_num)  
##agnes comparison  
agnes\_single <- agnes(cereals\_scaled, method = "single")  
agnes\_complete <- agnes(cereals\_scaled, method = "complete")  
agnes\_average <- agnes(cereals\_scaled, method = "average")  
agnes\_ward <- agnes(cereals\_scaled, method = "ward")  
##show agnes comparison graphically  
plot(agnes\_single, main = "single linkage")



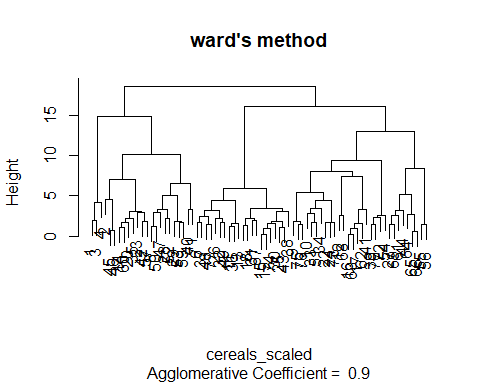
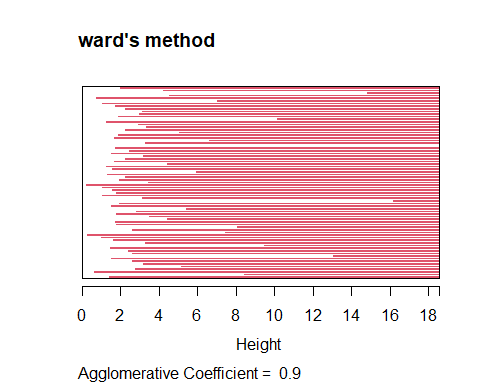
plot(agnes\_complete, main = "complete linkage")



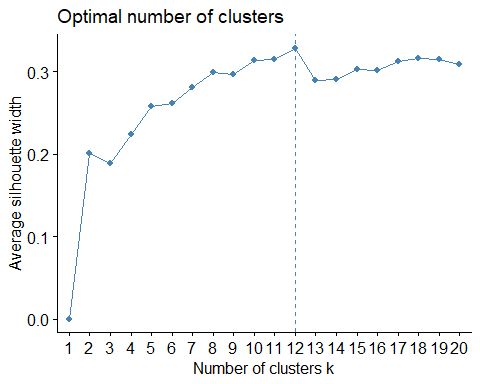
plot(agnes\_average, main = "average linkage")



plot(agnes\_ward, main = "ward's method")



fviz\_nbclust(cereals\_scaled, FUN = hcut, method = "silhouette", k.max = 20)



## choose k at elbow whcih is 2  
k <- 2  
clusters <- cutree(agnes\_ward, k = k)  
table(clusters)

## clusters  
## 1 2   
## 23 51

set.seed(1)  
##create data partitions  
idx <- sample(1:nrow(cereals\_scaled), size = 0.5 \* nrow(cereals\_scaled))  
partitionA <- cereals\_scaled[idx, ]  
partitionB <- cereals\_scaled[-idx, ]  
agnes\_A <- agnes(partitionA, method = "ward")  
clusters\_A <- cutree(agnes\_A, k = k)  
centroids\_A <- aggregate(partitionA, by = list(cluster = clusters\_A), FUN = mean)  
centroids\_matrix <- as.matrix(centroids\_A[, -1])  
dist\_Bcentroids <- proxy::dist(partitionB, centroids\_matrix)  
clusters\_B\_1 <- apply(as.matrix(dist\_Bcentroids), 1, which.min)  
agnes\_B <- agnes(partitionB, method = "ward")  
clusters\_B\_2 <- cutree(agnes\_B, k = k)  
ari <- adjustedRandIndex(clusters\_B\_1, clusters\_B\_2)  
##clsuters are showing to be unstable, increased numbers of clusters will be  
## able to have more accurate data, but assingment asks for one "healthy"  
## and one "unhealthy" group  
print(paste("adjusted rand Index for cluster stability:", round(ari, 3)))

## [1] "adjusted rand Index for cluster stability: 0.244"

if (ari > 0.7) {  
 cat("Clusters are stable.\n")  
} else {  
 cat("Clusters may not be stable.\n")  
}

## Clusters may not be stable.

##show the avg data between both clusters  
totaldata <- aggregate(cereals\_num, by = list(cluster = clusters), mean)  
print(totaldata)

## cluster calories protein fat sodium fiber carbo sugars  
## 1 1 116.0870 3.260870 1.7826087 157.8261 4.130435 13.0000 8.608696  
## 2 2 102.9412 2.176471 0.6470588 164.4118 1.294118 15.5098 6.431373  
## potass vitamins shelf weight cups rating  
## 1 172.17391 30.43478 2.913043 1.1500000 0.6526087 42.90285  
## 2 65.29412 28.43137 1.901961 0.9770588 0.8978431 42.13229

##show both cluster groupings  
cereals\_clean$Cluster <- clusters  
for (i in 1:k) {  
 cat(paste0("Cluster ", i, ":\n"))  
 print(cereals\_clean[cereals\_clean$Cluster == i, "name"])  
 cat("\n")  
}

## Cluster 1:  
## [1] 100%\_Bran   
## [2] 100%\_Natural\_Bran   
## [3] All-Bran   
## [4] All-Bran\_with\_Extra\_Fiber   
## [5] Basic\_4   
## [6] Clusters   
## [7] Cracklin'\_Oat\_Bran   
## [8] Crispy\_Wheat\_&\_Raisins   
## [9] Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats  
## [10] Fruitful\_Bran   
## [11] Great\_Grains\_Pecan   
## [12] Just\_Right\_Fruit\_&\_Nut   
## [13] Life   
## [14] Muesli\_Raisins,\_Dates,\_&\_Almonds   
## [15] Muesli\_Raisins,\_Peaches,\_&\_Pecans   
## [16] Mueslix\_Crispy\_Blend   
## [17] Nutri-Grain\_Almond-Raisin   
## [18] Oatmeal\_Raisin\_Crisp   
## [19] Post\_Nat.\_Raisin\_Bran   
## [20] Quaker\_Oat\_Squares   
## [21] Raisin\_Bran   
## [22] Raisin\_Nut\_Bran   
## [23] Total\_Raisin\_Bran   
## 77 Levels: 100%\_Bran 100%\_Natural\_Bran All-Bran ... Wheaties\_Honey\_Gold  
##   
## Cluster 2:  
## [1] Apple\_Cinnamon\_Cheerios Apple\_Jacks   
## [3] Bran\_Chex Bran\_Flakes   
## [5] Cap'n'Crunch Cheerios   
## [7] Cinnamon\_Toast\_Crunch Cocoa\_Puffs   
## [9] Corn\_Chex Corn\_Flakes   
## [11] Corn\_Pops Count\_Chocula   
## [13] Crispix Double\_Chex   
## [15] Froot\_Loops Frosted\_Flakes   
## [17] Frosted\_Mini-Wheats Fruity\_Pebbles   
## [19] Golden\_Crisp Golden\_Grahams   
## [21] Grape\_Nuts\_Flakes Grape-Nuts   
## [23] Honey\_Graham\_Ohs Honey\_Nut\_Cheerios   
## [25] Honey-comb Just\_Right\_Crunchy\_\_Nuggets  
## [27] Kix Lucky\_Charms   
## [29] Maypo Multi-Grain\_Cheerios   
## [31] Nut&Honey\_Crunch Nutri-grain\_Wheat   
## [33] Product\_19 Puffed\_Rice   
## [35] Puffed\_Wheat Raisin\_Squares   
## [37] Rice\_Chex Rice\_Krispies   
## [39] Shredded\_Wheat Shredded\_Wheat\_'n'Bran   
## [41] Shredded\_Wheat\_spoon\_size Smacks   
## [43] Special\_K Strawberry\_Fruit\_Wheats   
## [45] Total\_Corn\_Flakes Total\_Whole\_Grain   
## [47] Triples Trix   
## [49] Wheat\_Chex Wheaties   
## [51] Wheaties\_Honey\_Gold   
## 77 Levels: 100%\_Bran 100%\_Natural\_Bran All-Bran ... Wheaties\_Honey\_Gold

##healthy cereal cluster 1(subjective data, i believe more vitamins and fiber outweighs   
##calorie and sugar quantities)  
print(cereals\_clean[cereals\_clean$Cluster == 1, "name"])

## [1] 100%\_Bran   
## [2] 100%\_Natural\_Bran   
## [3] All-Bran   
## [4] All-Bran\_with\_Extra\_Fiber   
## [5] Basic\_4   
## [6] Clusters   
## [7] Cracklin'\_Oat\_Bran   
## [8] Crispy\_Wheat\_&\_Raisins   
## [9] Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats  
## [10] Fruitful\_Bran   
## [11] Great\_Grains\_Pecan   
## [12] Just\_Right\_Fruit\_&\_Nut   
## [13] Life   
## [14] Muesli\_Raisins,\_Dates,\_&\_Almonds   
## [15] Muesli\_Raisins,\_Peaches,\_&\_Pecans   
## [16] Mueslix\_Crispy\_Blend   
## [17] Nutri-Grain\_Almond-Raisin   
## [18] Oatmeal\_Raisin\_Crisp   
## [19] Post\_Nat.\_Raisin\_Bran   
## [20] Quaker\_Oat\_Squares   
## [21] Raisin\_Bran   
## [22] Raisin\_Nut\_Bran   
## [23] Total\_Raisin\_Bran   
## 77 Levels: 100%\_Bran 100%\_Natural\_Bran All-Bran ... Wheaties\_Honey\_Gold