Assignment 5

Sandeep More

04/25/2021

Assignment 5 - Hierarchical Clustering

Use hierarchical clustering to analyze consumer ratings for 77 breakfast cereals.

Data preprocessing

Load given data and remove rows with any NA value in it

Dimension reduction

We do not care about the following columns

- 'mfr' (manufacturer)
- 'type' (hot or cold)
- 'shelf' - we don't care which shelf it is placed, unrelated to health

so we can eliminate these columns

Normalizing

Note the data used has units milligrams for some and grams for others. There are also different units used for other data. Let's normalize the data to make sure units don't affect our clusters.

```
# Normalize the data
data.df.norm <- sapply(data.df, scale)
row.names(data.df.norm) <- row.names(data.df)
#head(data.df.norm)
```

Calculate Distance

Calculate distance using euclidean method.

```
#1. euclidean
euclidean.dist <- dist(data.df.norm, method="euclidean")
#print(euclidean.dist)
```

The problem here is that we have way too many variables to get any meaningful clusters out. Let us try to reduce dimensions using PCA

PCA

For details on PCA : Data Mining for Business Analytics (R) Chapter 4, pg:101 Let us try to run PCA and see how the data are corelated and whether we can remove some of the attributes.

```r
# normalize the data and compute PCs on all the dimensions
pcs.cor <- prcomp(data.df, scale. = T)
summary(pcs.cor)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4    PC5     PC6     PC7
## Standard deviation     1.8904 1.7202 1.3818 0.99793 0.8541 0.84077 0.76711
## Proportion of Variance 0.2978 0.2466 0.1591 0.08299 0.0608 0.05891 0.04904
## Cumulative Proportion  0.2978 0.5444 0.7035 0.78647 0.8473 0.90617 0.95521
##                            PC8     PC9    PC10    PC11      PC12
## Standard deviation     0.59974 0.30491 0.25563 0.13953 1.502e-08
## Proportion of Variance 0.02997 0.00775 0.00545 0.00162 0.000e+00
## Cumulative Proportion  0.98518 0.99293 0.99838 1.00000 1.000e+00
```

```r
# see first 7 PCs
pcs.cor$rotation[,1:7]
```

```
##                 PC1        PC2         PC3         PC4         PC5         PC6
## calories  0.3670078  0.3370596 -0.11462186  0.27660732  0.24758737  0.08912183
## protein  -0.2772241  0.2449194 -0.27689496  0.42031136 -0.18304085  0.12153045
## fat       0.1049438  0.3343406  0.20508488  0.60037932 -0.36602704 -0.10945914
## sodium    0.2015610  0.1150253 -0.38922969 -0.23670780 -0.32044177 -0.71728694
## fiber    -0.4180964  0.2880965 -0.06933016 -0.20825935 -0.04343596 -0.01086524
## carbo     0.1636662 -0.1948813 -0.56267964  0.20718081  0.38287999 -0.03790173
## sugars    0.2874024  0.3141683  0.35567565 -0.28893761  0.10210150  0.18011067
## potass   -0.3443208  0.3970189 -0.06712531 -0.09011254 -0.02365719  0.03488628
## vitamins  0.1557868  0.1196450 -0.38810943 -0.34092431 -0.49287096  0.48109792
## weight    0.1281124  0.4578598 -0.24665137 -0.12160132  0.41597684  0.12103468
## cups      0.2510333 -0.2738283 -0.14025953  0.12920887 -0.27373395  0.38620107
## rating   -0.4799624 -0.1586012 -0.18184294  0.06876466  0.13483653  0.14261284
##                  PC7
## calories -0.009769147
## protein   0.147553524
## fat      -0.185133459
## sodium    0.222566270
## fiber     0.168689192
## carbo    -0.119675258
## sugars    0.204731752
## potass    0.202493463
## vitamins -0.463409721
## weight    0.100570037
## cups      0.737306160
## rating    0.011924746
```

PC Analysis

Looking at the PC analysis we can see that we need 7 principal components to account for more than 95% of variability (looking at the Cumulative Proportion for PC7 = 0.95521). The first 3 components account for more than 70% of variability.

Looking at the weights for PC1 we can see that it measures balance between two sets of attributes

- Calories, cups, sugars, sodium (high positive weights) and
- protein, fiber, potassium and ratings (high negative weights)

I couldn't reduce dimensions looking at this data, they all look like they introduce some variance. We will

look at the PC Analysis later in summary analysis.

Reduce dimensions

More details see: https://stats.stackexchange.com/questions/57467/how-to-perform-dimensionality-reduction-with-pca-in-r

```
# eigenvalues
pcs.cor$sdev
```

```
##  [1] 1.890356e+00 1.720171e+00 1.381791e+00 9.979273e-01 8.541336e-01
##  [6] 8.407658e-01 7.671100e-01 5.997359e-01 3.049136e-01 2.556267e-01
## [11] 1.395346e-01 1.502344e-08
```

```
length(pcs.cor$sdev)
```

```
## [1] 12
```

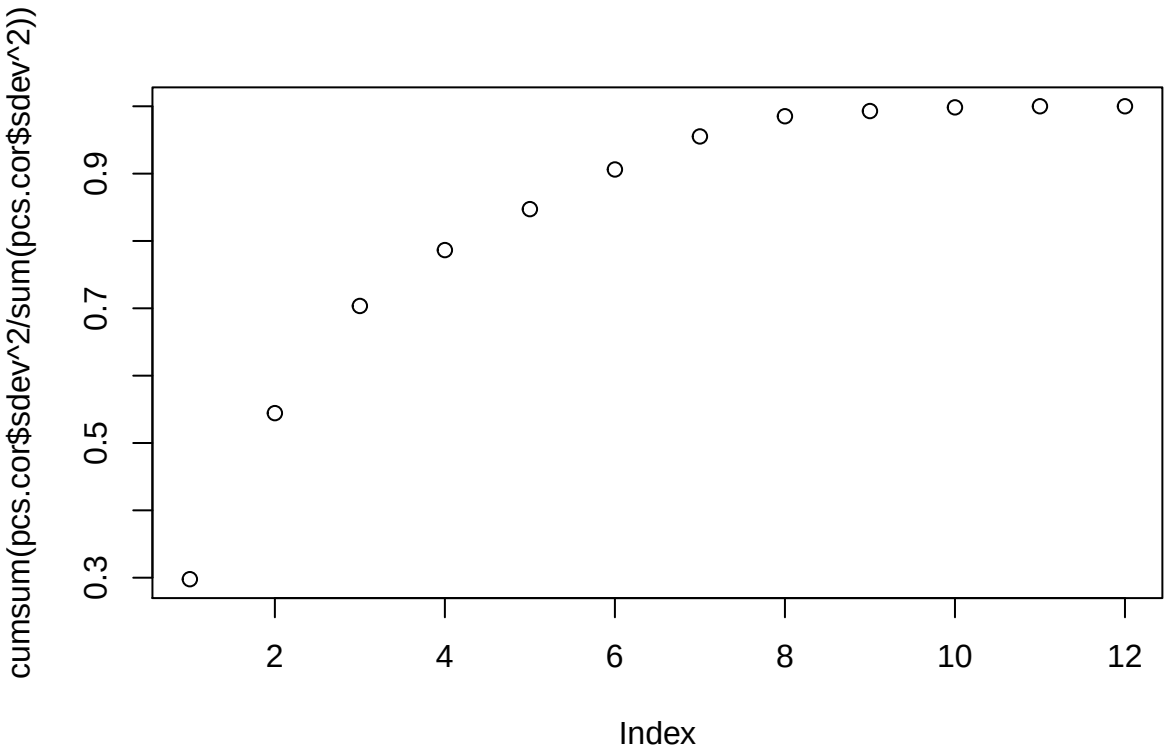
```
dim(pcs.cor$rotation)
```

```
## [1] 12 12
```

```
# see first 7 PCs
#pcs.cor$x[,1:7]
#dim(pcs.cor$x)
```

By squaring the eigenvalues, we get the variance explained by each PC:

```
# By squaring the eigenvalues, we get the variance explained by each PC:
plot(cumsum(pcs.cor$sdev^2/sum(pcs.cor$sdev^2))) #cumulative explained variance
```



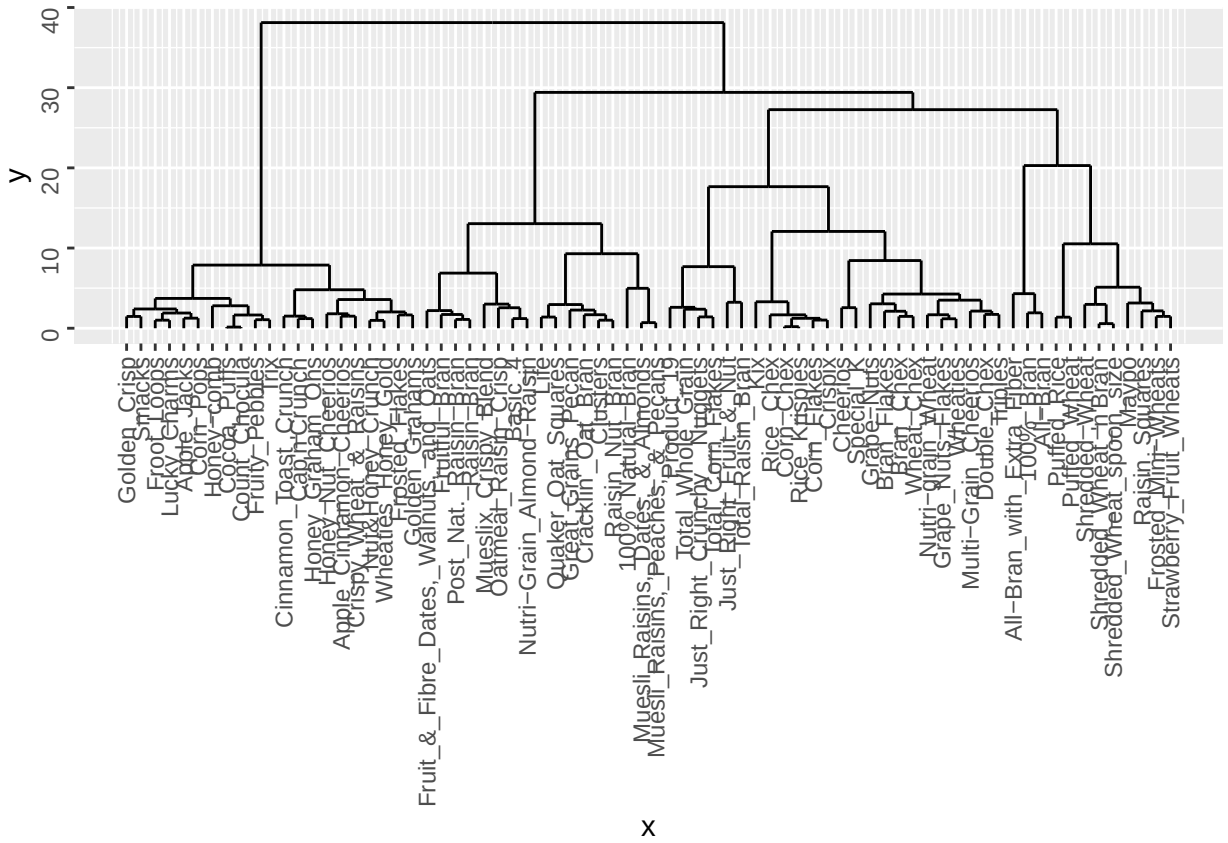
The above plot of eigenvalues confirms that PC7 gives most variance.

Clustering

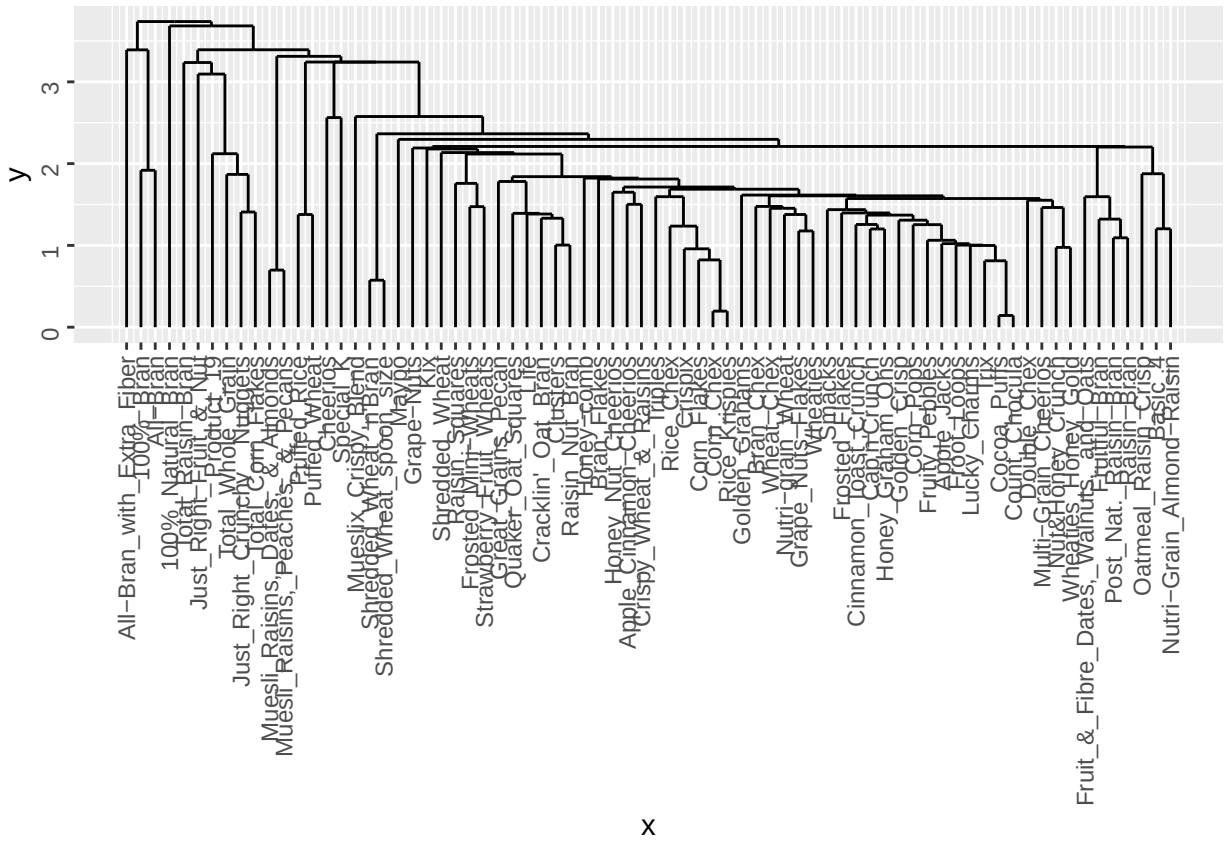
Agglomerative Cluster

Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward. Here we are using the package ggdendrogram

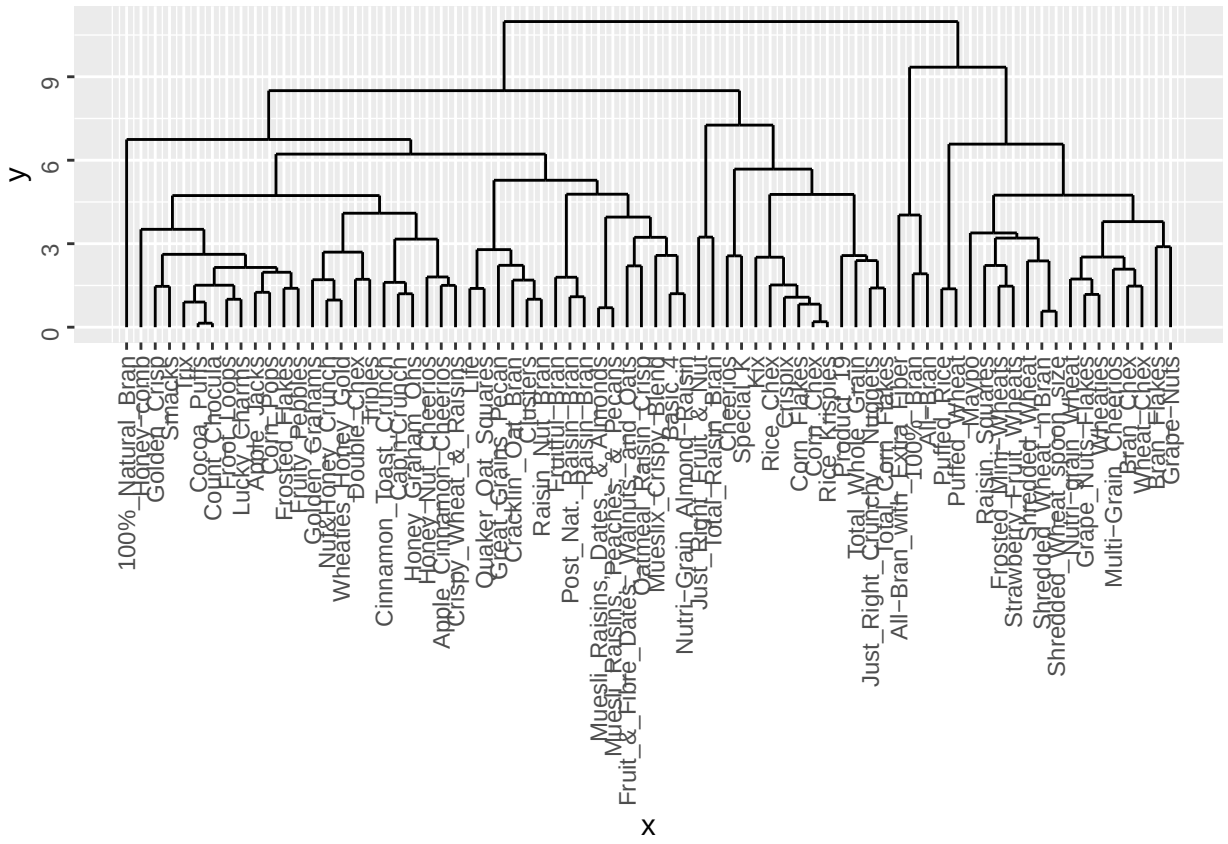
```
library(ape)
library(ggplot2)
library(ggdendro)
# More info on dendogram plotting
# https://www.gastonsanchez.com/visually-enforced/how-to/2012/10/03/Dendrograms/

#method could be "ward.D", "single", "complete", "average", "median", "centroid"
agglo.cluster.ward <- hclust(euclidean.dist, method = "ward.D")
# Put the labels at the same height: hang = -1
#plot(agglo.cluster.ward, hang = -1, ann=FALSE)
#plot(as.phylo(agglo.cluster.ward), cex = 0.9)
ggdendrogram(agglo.cluster.ward, theme_dendro = FALSE)
```



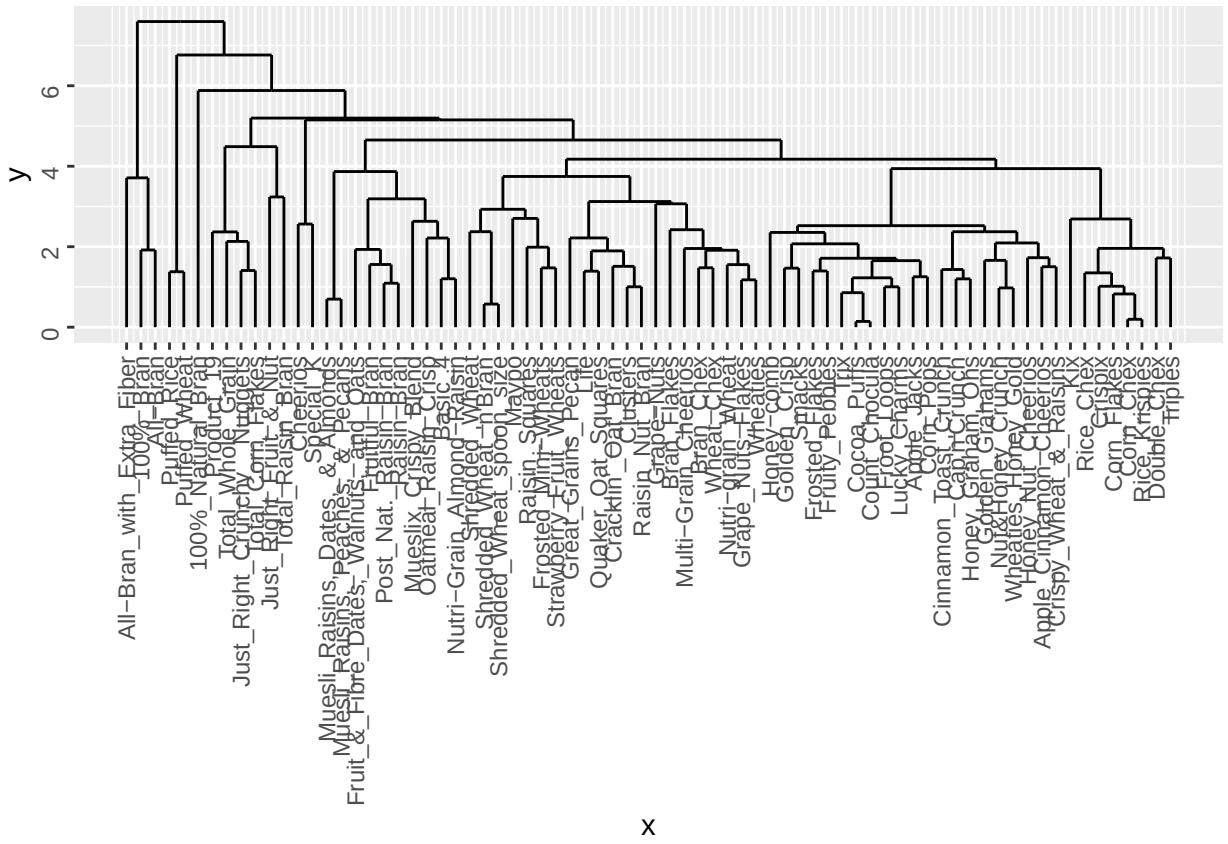

```
agglo.cluster.single <- hclust(euclidean.dist, method = "single")
#plot(agglo.cluster.single, hang = -1, ann=FALSE)
ggdendrogram(agglo.cluster.single, theme_dendro = FALSE)
```



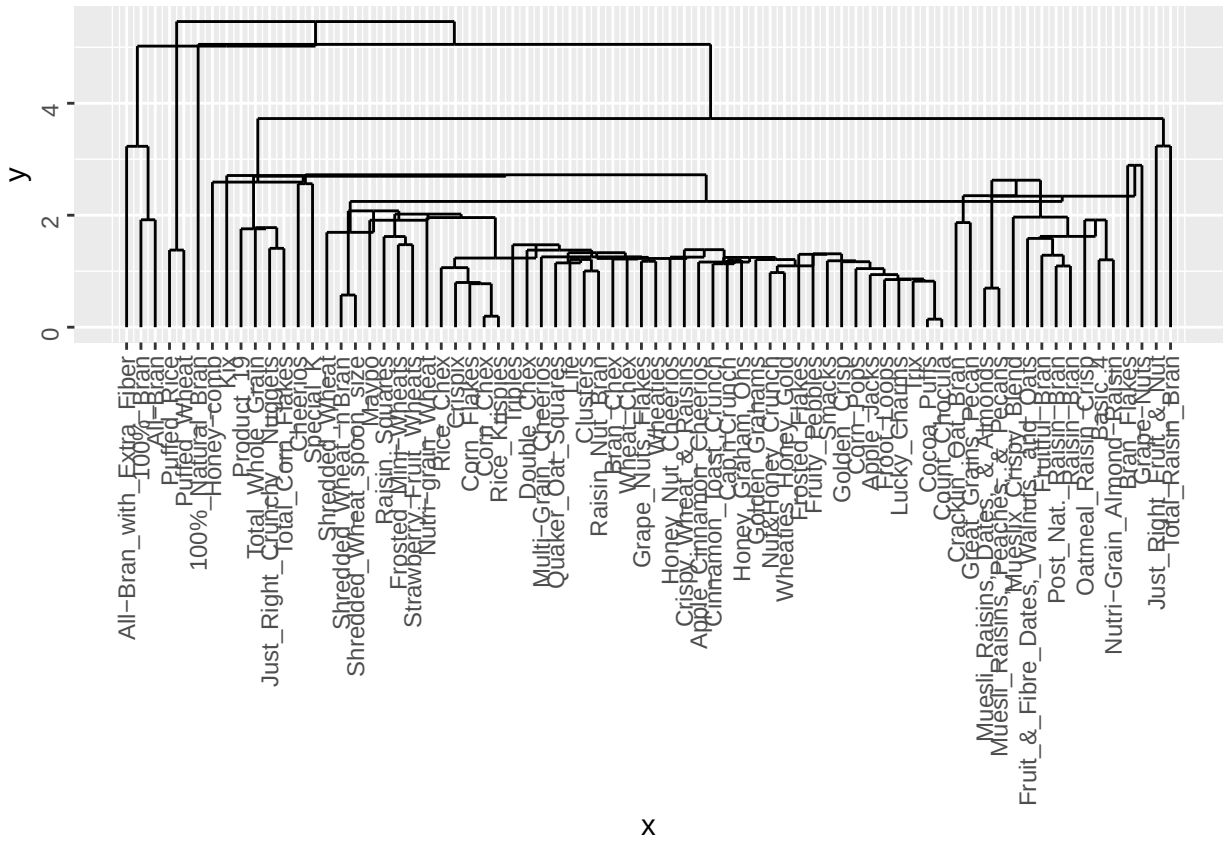

```
agglo.cluster.complete <- hclust(euclidean.dist, method = "complete")
#plot(agglo.cluster.complete, hang = -1, ann=FALSE)
ggdendrogram(agglo.cluster.complete, theme_dendro = FALSE)
```



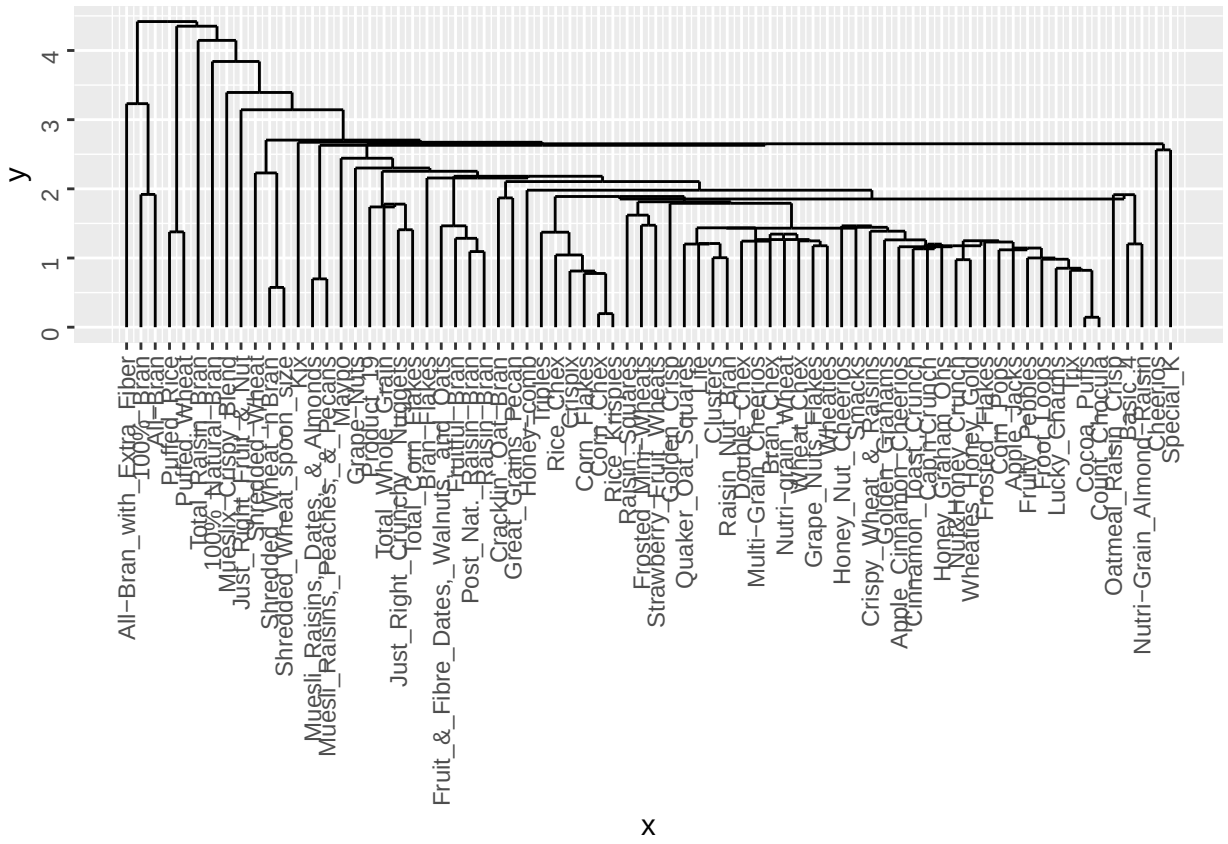

```
agglo.cluster.average <- hclust(euclidean.dist, method = "average")
#plot(agglo.cluster.average, hang = -1, ann=FALSE)
ggdendrogram(agglo.cluster.average, theme_dendro = FALSE)
```




```r
agglo.cluster.median <- hclust(euclidean.dist, method = "median")
#plot(agglo.cluster.median, hang = -1, ann=FALSE)
ggdendrogram(agglo.cluster.median, theme_dendro = FALSE)
```




```
agglo.cluster.centroid <- hclust(euclidean.dist, method = "centroid")
#plot(agglo.cluster.centroid, hang = -1, ann=FALSE)
ggdendrogram(agglo.cluster.centroid, theme_dendro = FALSE)
```



Best Approach Looking at the above dendograms it apepars that Ward's method appears to cluster better. Unlike the other methods, Ward's method measuring the distance directly, it analyzes the variance of clusters. Ward's is said to be the most suitable method for quantitative variables.

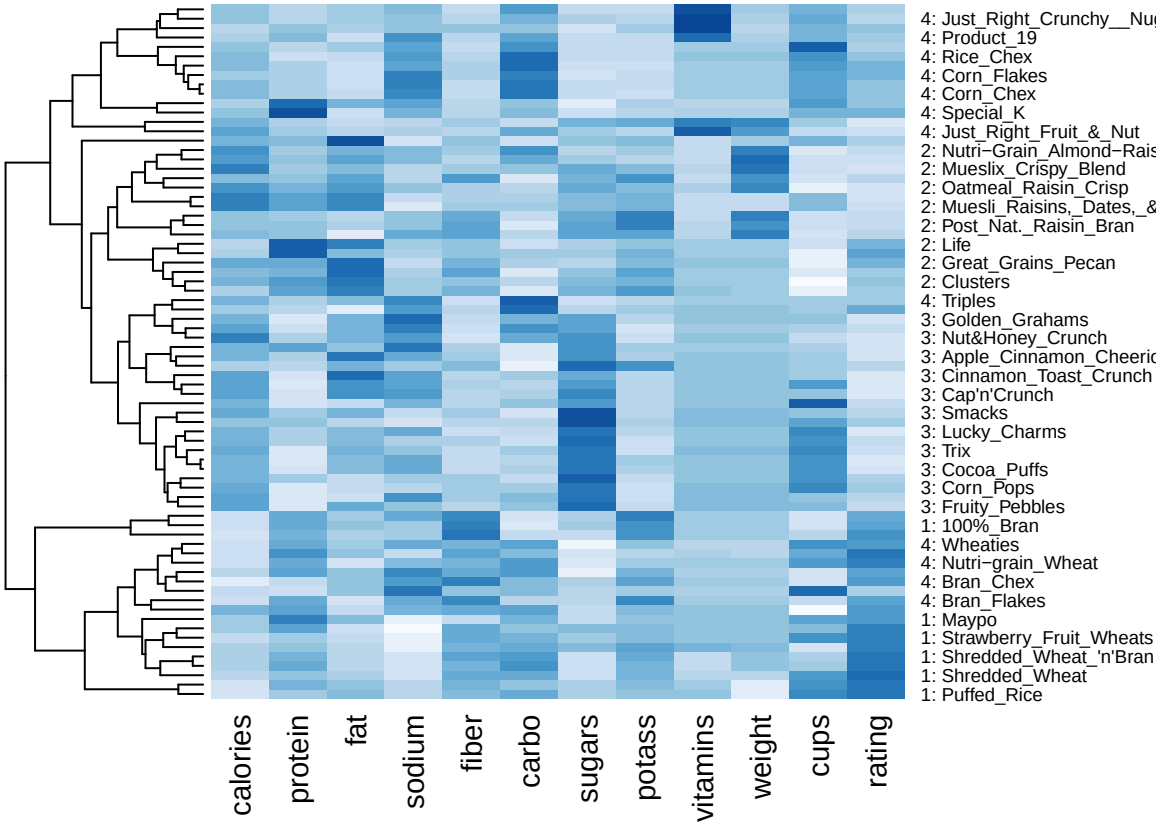
Also, we can see that 4 clusters stand out. We can use 4 as a as a good starting point and then experiment with different values of k.

Experimenting with number of clusters

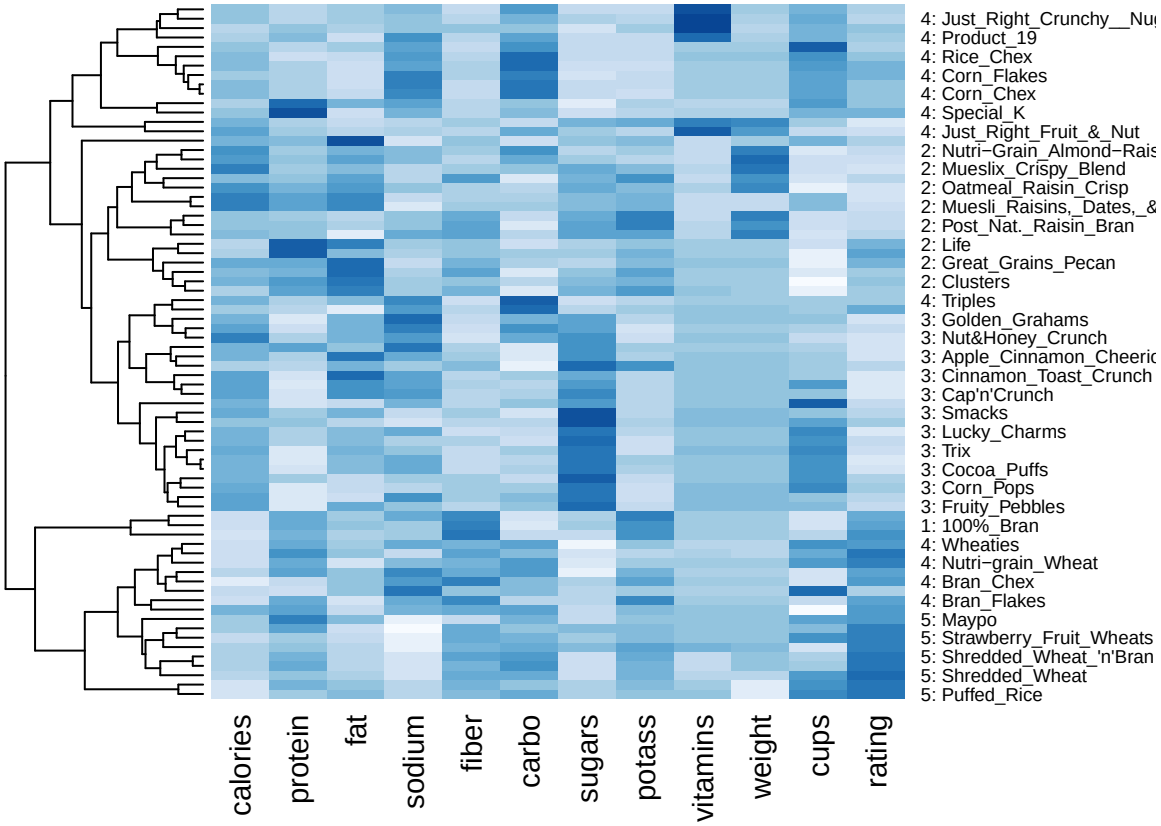
```
agglo.cluster.ward.cut.4 <- cutree(agglo.cluster.ward, k = 4)
agglo.cluster.ward.cut.5 <- cutree(agglo.cluster.ward, k = 5)
agglo.cluster.ward.cut.6 <- cutree(agglo.cluster.ward, k = 6)
agglo.cluster.ward.cut.8 <- cutree(agglo.cluster.ward, k = 8)
agglo.cluster.ward.cut.9 <- cutree(agglo.cluster.ward, k = 9)
#print(agglo.cluster.ward.cut)
```

Heatmap

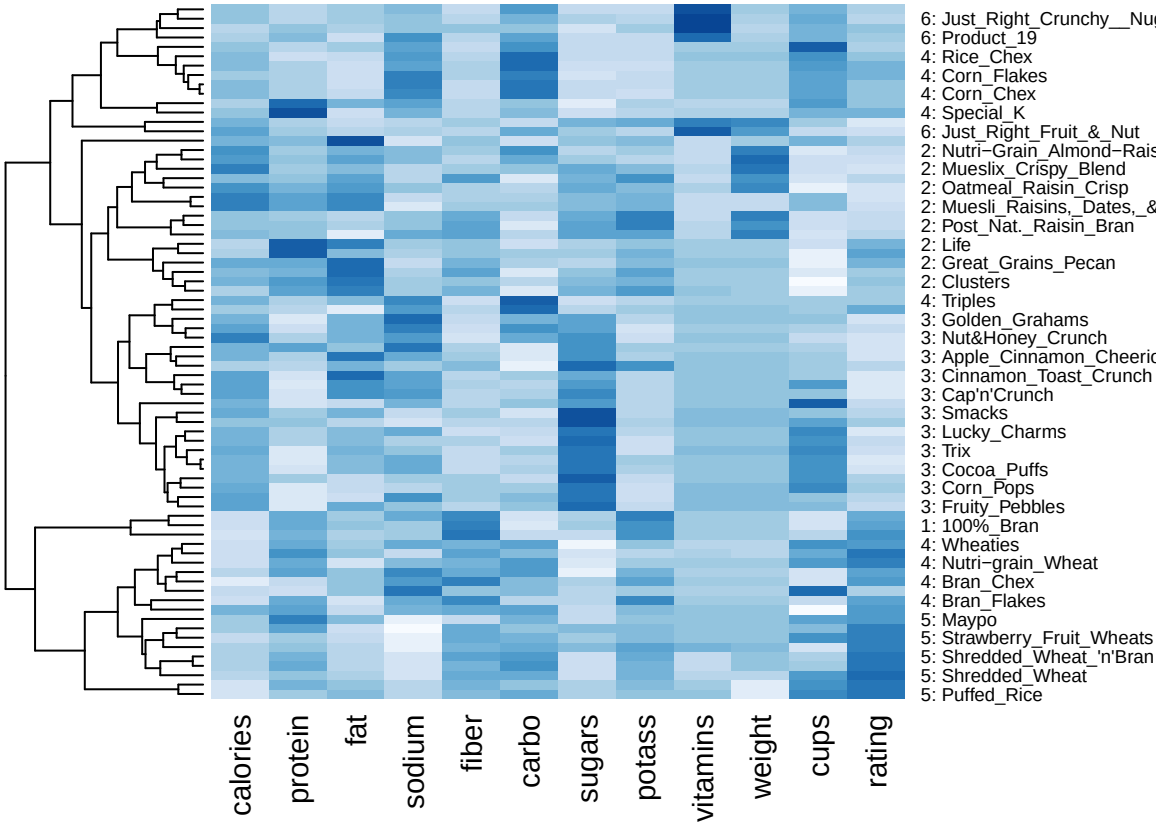
```
library(RColorBrewer)
# Make the labels as cluster membership (determined from cuttree) : row name
row.names(data.df.norm) <- paste(agglo.cluster.ward.cut.4, ": ", row.names(data.df), sep = "")

# plot
#color=rev(paste("gray", 1:99,sep = ""))
#color = terrain.colors(256)
color = colorRampPalette(brewer.pal(8, "Blues"))(25)
heatmap(as.matrix(data.df.norm), Colv = NA, hclustfun = hclust, col = color)
```



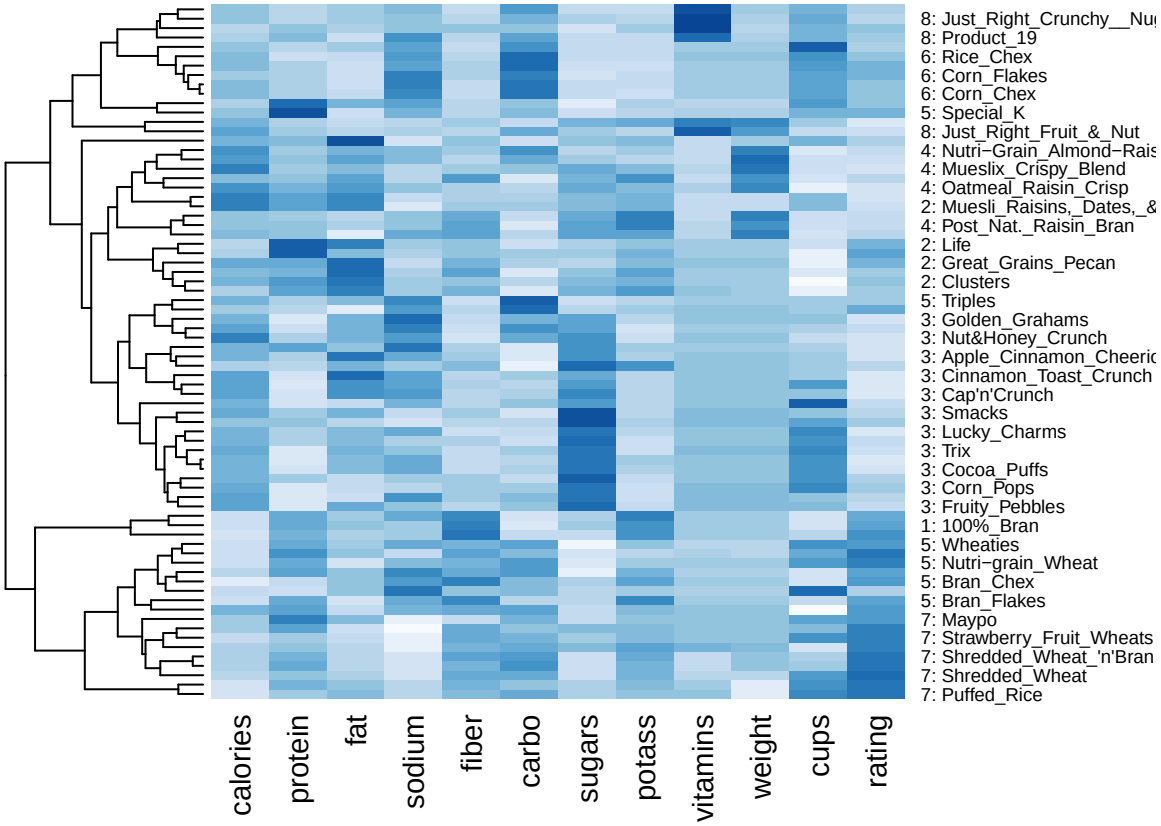

```
##### 5 cuts ####
row.names(data.df.norm) <- paste(agglo.cluster.ward.cut.5, ": ", row.names(data.df), sep = "")

color = colorRampPalette(brewer.pal(8, "Blues"))(25)
heatmap(as.matrix(data.df.norm), Colv = NA, hclustfun = hclust, col = color)
```



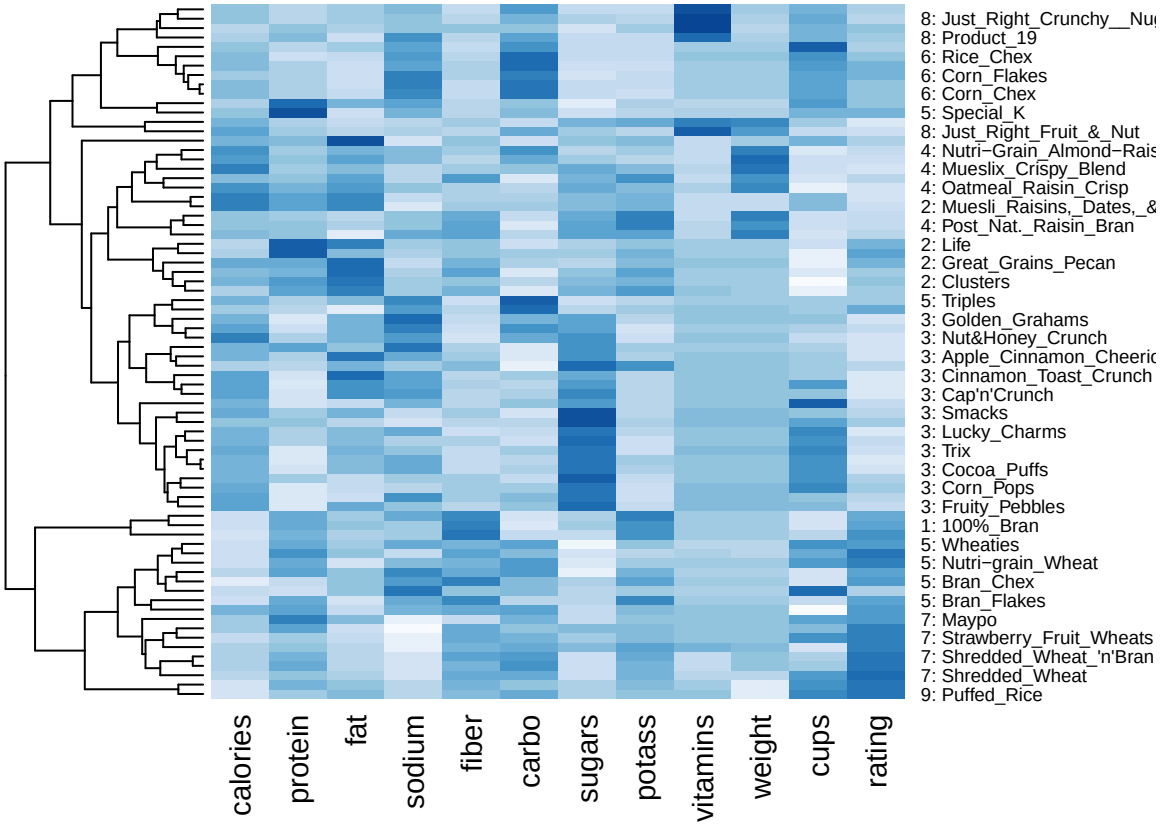

```
##### 6 cuts ####
row.names(data.df.norm) <- paste(agglo.cluster.ward.cut.6, ": ", row.names(data.df), sep = "")

color = colorRampPalette(brewer.pal(8, "Blues"))(25)
heatmap(as.matrix(data.df.norm), Colv = NA, hclustfun = hclust, col = color)
```




```
##### 8 cuts ####
row.names(data.df.norm) <- paste(agglo.cluster.ward.cut.8, ": ", row.names(data.df), sep = "")

color = colorRampPalette(brewer.pal(8, "Blues"))(25)
heatmap(as.matrix(data.df.norm), Colv = NA, hclustfun = hclust, col = color)
```




```
##### 9 cuts ####
row.names(data.df.norm) <- paste(agglo.cluster.ward.cut.9, ": ", row.names(data.df), sep = "")

color = colorRampPalette(brewer.pal(8, "Blues"))(25)
heatmap(as.matrix(data.df.norm), Colv = NA, hclustfun = hclust, col = color)
```



Value of K

Looking at the heat map generated for cluster of size 4,5,6,8 and 9 we can see that the best value of K=8 beyond 8 we see splintered clusters.

Cluster stability

Here we use clVlaid package to determine cluster validity. clValid performns cluster validity by partitioning the data so we do not have to do it manually, this is the reason why this was chosen. Here we are specifically interested in the following measurements:

- internal - Take only the data set and the clustering partition as input and use intrinsic information in the data to assess the quality of the clustering
- stability - Evaluate the consistency of a clustering result by comparing it with the clusters obtained after each column is removed, one at a time.

We are looking for the following values to find optimum value of k:

- Internal Measures
  - Connectivity - Describes the connectivity between NN and should be minimized,
  - Silhouette - Silhouette value measures the degree of confidence in the clustering assignment of a particular observation, should be maximized
  - Dunn Index - Ratio of the smallest distance between observations not in the same cluster to the largest intra-cluster distance, should be maximized
- Stability Measures - the following measures should all be minimized
  - APN - average proportion of non-overlap
  - AD - average distance
  - ADM - average distance between mean
  - FOM - figure of merit

```r
# "ward","single","complete","average","median","centroid"
library(clValid)
```

```
## Loading required package: cluster
```

```r
val.ward <- clValid(data.df.norm, nClust = c(4:10, 15), clMethods = "agnes", method = "ward", validatio
summary(val.ward)
```

```
##
## Clustering Methods:
##  agnes
##
## Cluster sizes:
##  4 5 6 7 8 9 10 15
##
## Validation Measures:
##                            4       5       6       7       8       9      10      15
##
## agnes APN             0.1602  0.1139  0.1285  0.1174  0.1376  0.0937  0.1023  0.0894
##       AD              3.5135  3.1186  2.9347  2.7445  2.6216  2.4201  2.3264  1.7785
##       ADM             0.9425  0.6036  0.6484  0.5577  0.6402  0.4765  0.5224  0.3290
##       FOM             0.8869  0.7878  0.7285  0.7137  0.7006  0.6797  0.6798  0.5957
##       Connectivity   24.5369 29.8044 35.1389 38.4520 42.0849 49.6587 52.2683 75.5802
##       Dunn            0.1831  0.1893  0.2464  0.2464  0.2464  0.2575  0.2696  0.2632
##       Silhouette      0.2326  0.2698  0.2855  0.3034  0.3166  0.3185  0.3194  0.3182
##
## Optimal Scores:
##
##              Score   Method Clusters
## APN           0.0894 agnes  15
## AD            1.7785 agnes  15
## ADM           0.3290 agnes  15
## FOM           0.5957 agnes  15
## Connectivity 24.5369 agnes  4
## Dunn          0.2696 agnes  10
## Silhouette    0.3194 agnes  10
```

```r
val.single <- clValid(data.df.norm, nClust = c(4:10, 15), clMethods = "agnes", method = "single", valida
summary(val.single)
```

```
##
## Clustering Methods:
##  agnes
##
## Cluster sizes:
##  4 5 6 7 8 9 10 15
##
## Validation Measures:
##                            4       5       6       7       8       9      10      15
##
## agnes APN             0.0119  0.0305  0.0536  0.0399  0.0233  0.0207  0.0121  0.0569
##       AD              3.8910  3.8207  3.7657  3.5441  3.4173  3.2975  3.1744  2.6576
##       ADM             0.1652  0.2132  0.4641  0.3407  0.3235  0.2494  0.1251  0.3773
##       FOM             0.9350  0.9246  0.9143  0.9036  0.8915  0.8704  0.8418  0.7993
##       Connectivity   13.7302 16.0635 20.2464 24.3472 28.1052 29.7052 32.2091 51.7877
```

```
##       Dunn           0.3272  0.3271  0.3194  0.4061  0.4061  0.4054  0.3877  0.3361
##       Silhouette     0.2089  0.1938  0.1601  0.1767  0.1818  0.1711  0.1598  0.1611
##
## Optimal Scores:
##
##              Score   Method Clusters
## APN           0.0119 agnes  4
## AD            2.6576 agnes  15
## ADM           0.1251 agnes  10
## FOM           0.7993 agnes  15
## Connectivity 13.7302 agnes  4
## Dunn          0.4061 agnes  7
## Silhouette    0.2089 agnes  4
```

```
val.complete <- clValid(data.df.norm, nClust = c(4:10, 15), clMethods = "agnes", method = "complete", va
summary(val.complete)
```

```
##
## Clustering Methods:
##  agnes
##
## Cluster sizes:
##  4 5 6 7 8 9 10 15
##
## Validation Measures:
##                            4       5       6       7       8       9      10      15
##
## agnes APN             0.2183  0.2510  0.3130  0.3378  0.1938  0.1916  0.1557  0.1512
##       AD              3.7837  3.6134  3.5220  3.2834  2.8429  2.6851  2.5025  1.9256
##       ADM             1.2130  1.1817  1.3128  1.2882  0.8388  0.7546  0.6682  0.5989
##       FOM             0.8744  0.8624  0.8527  0.7655  0.7073  0.6978  0.6849  0.6112
##       Connectivity   32.6833 35.2873 38.3274 42.4282 47.2476 47.4698 50.8206 76.6175
##       Dunn            0.2015  0.2169  0.2224  0.2352  0.2576  0.2771  0.3062  0.3400
##       Silhouette      0.2208  0.2146  0.2043  0.2056  0.2709  0.2894  0.2797  0.2831
##
## Optimal Scores:
##
##              Score   Method Clusters
## APN           0.1512 agnes  15
## AD            1.9256 agnes  15
## ADM           0.5989 agnes  15
## FOM           0.6112 agnes  15
## Connectivity 32.6833 agnes  4
## Dunn          0.3400 agnes  15
## Silhouette    0.2894 agnes  9
```

```
val.average <- clValid(data.df.norm, nClust = c(4:10, 15), clMethods = "agnes", method = "average", vali
summary(val.average)
```

```
##
## Clustering Methods:
##  agnes
##
## Cluster sizes:
##  4 5 6 7 8 9 10 15
```

```
##
## Validation Measures:
##                             4       5       6       7       8       9      10      15
##
## agnes APN            0.0465  0.0863  0.1127  0.1363  0.1890  0.1878  0.1711  0.0940
##       AD             3.9397  3.7262  3.5581  3.3340  3.2098  2.9117  2.7217  1.9332
##       ADM            0.2595  0.4283  0.4854  0.7404  0.8598  1.0546  1.1164  0.4332
##       FOM            0.9139  0.9086  0.8912  0.8771  0.8551  0.8091  0.7987  0.6207
##       Connectivity  11.5944 17.8310 21.6889 27.7933 30.3972 42.0020 44.2714 61.0313
##       Dunn           0.3680  0.4061  0.4061  0.3124  0.3124  0.2548  0.2596  0.3024
##       Silhouette     0.2649  0.2228  0.2153  0.2352  0.2294  0.2821  0.3266  0.3501
##
## Optimal Scores:
##
##              Score   Method Clusters
## APN           0.0465 agnes  4
## AD            1.9332 agnes  15
## ADM           0.2595 agnes  4
## FOM           0.6207 agnes  15
## Connectivity 11.5944 agnes  4
## Dunn          0.4061 agnes  5
## Silhouette    0.3501 agnes  15
```

```
val.kmeans <- clValid(data.df.norm, nClust = c(4:10, 15), clMethods = c("kmeans","pam"), validation = c
summary(val.kmeans)
```

```
##
## Clustering Methods:
##  kmeans pam
##
## Cluster sizes:
##  4 5 6 7 8 9 10 15
##
## Validation Measures:
##                             4       5       6       7       8       9      10      15
##
## kmeans APN            0.2510  0.2175  0.0882  0.1575  0.1769  0.2032  0.1568  0.1552
##        AD             3.6952  3.3497  2.9079  2.8435  2.8021  2.7195  2.5184  1.9393
##        ADM            1.3394  0.9345  0.5081  0.8309  0.9657  0.8588  0.8850  0.5273
##        FOM            0.8204  0.7975  0.7476  0.7745  0.7578  0.7436  0.7287  0.6018
##        Connectivity  27.8440 31.8548 33.0452 38.0508 40.9837 49.6913 52.3647 65.8175
##        Dunn           0.1731  0.1958  0.2169  0.2464  0.2464  0.2424  0.3171  0.2733
##        Silhouette     0.2416  0.2635  0.2897  0.3071  0.3135  0.2896  0.3316  0.3481
## pam    APN            0.0796  0.0769  0.0947  0.1292  0.0868  0.0857  0.1451  0.1492
##        AD             3.3738  3.0925  2.8871  2.7380  2.5598  2.4354  2.3835  1.8666
##        ADM            0.3737  0.3715  0.4054  0.5388  0.3825  0.3962  0.5363  0.5051
##        FOM            0.8413  0.7560  0.7333  0.7080  0.6900  0.6806  0.6651  0.6150
##        Connectivity  36.7226 37.2639 35.5313 45.5996 53.3020 57.0250 59.6294 74.4607
##        Dunn           0.1021  0.1256  0.1256  0.1299  0.1394  0.1490  0.1490  0.3256
##        Silhouette     0.2124  0.2570  0.2802  0.2865  0.2619  0.2708  0.2763  0.3002
##
## Optimal Scores:
##
##              Score   Method Clusters
## APN           0.0769 pam    5
```

```
## AD            1.8666 pam    15
## ADM           0.3715 pam    5
## FOM           0.6018 kmeans 15
## Connectivity 27.8440 kmeans 4
## Dunn          0.3256 pam    15
## Silhouette    0.3481 kmeans 15
```

Looking at the summary ouput for different methods, we see that k=10 is recommended. Comparing it with the previous results (PCA, Heatmaps and dendograms) we think value of k=8 would work the best. The stability numbers do not look good and are concerning. We tried to use other methods such as kmeans and pam to see if the stability numbers improve (results above) but they do not. At this point I belive further works needs to be done to explore why cluster stability is low and how it can be fixed.

Healthy Cereals - cluster

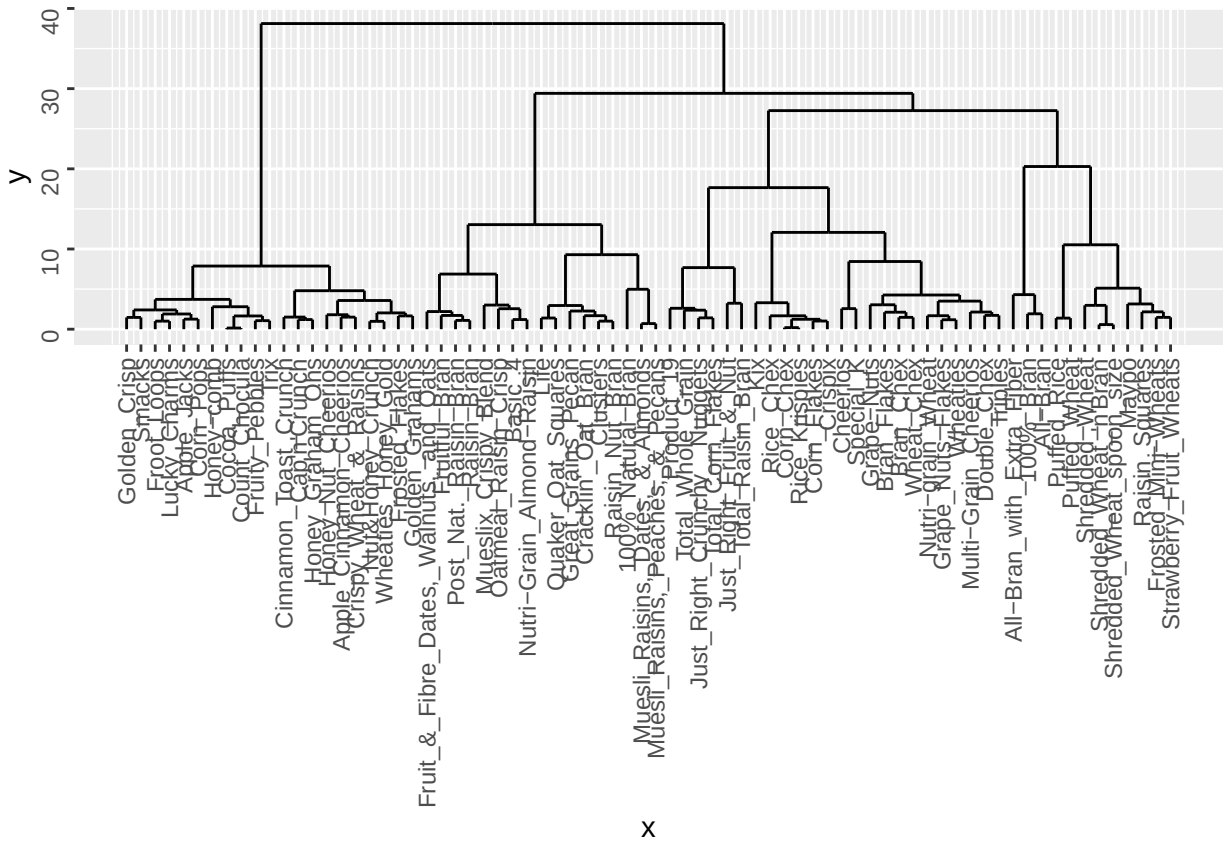
Data normalization

Of course the data needs to be normalized. There units for the data are different, e.g. sodium is measured in milligrams and potassium in grams so the clustering without normalization would be skewed.

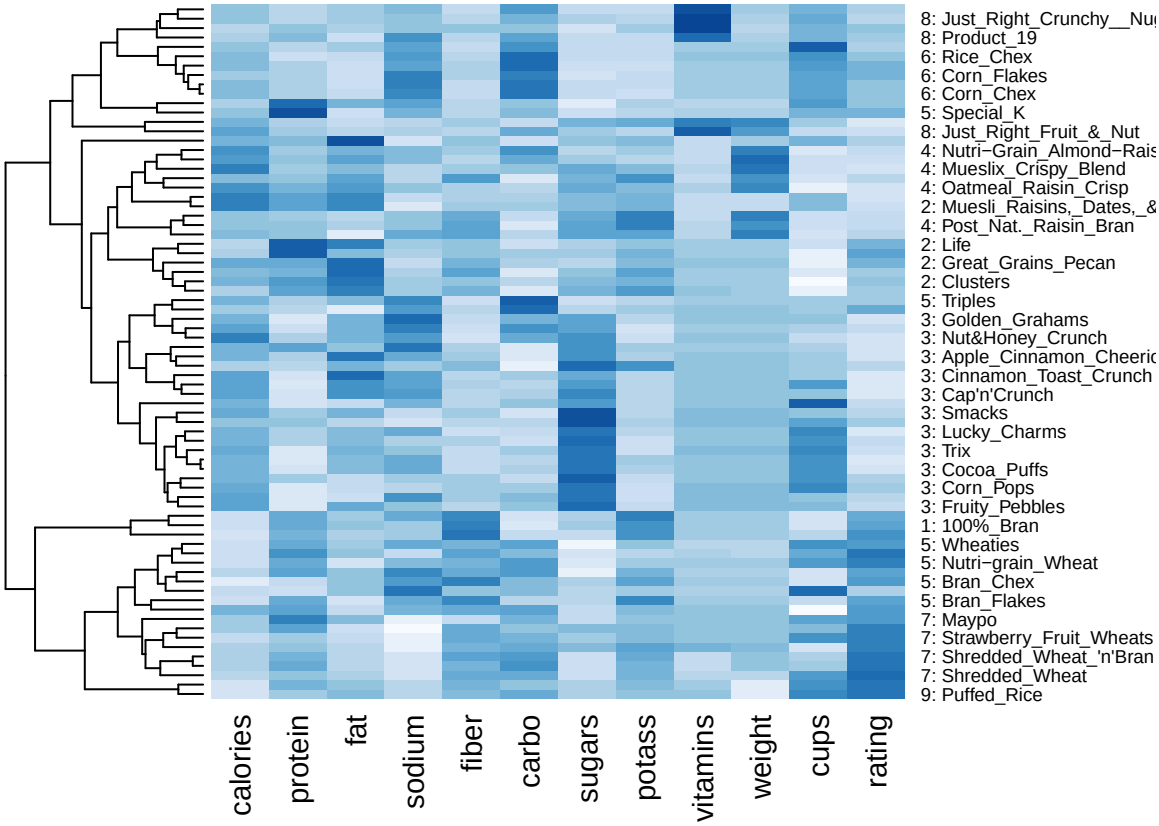
Summary

Healthy is a relative term and depends on various factors such as every child has different nutrition needs. We could debate, should high calories be classified as unhealthy since calories are essential to growing children. We need more information on what healthy means to be able to recommend a Healthy Cereals cluster. For the sake of this assignment we assume some requirements, such as cerials with low sugar and calories and high protein and fiber are healthy.

```
# summary results

# Chosen dendogram
ggdendrogram(agglo.cluster.ward, theme_dendro = FALSE)
```




```r
# Chosen heatmap
color = colorRampPalette(brewer.pal(8, "Blues"))(25)
heatmap(as.matrix(data.df.norm), Colv = NA, hclustfun = hclust, col = color)
```




```
# PCAnalysis
print(pcs.cor$rotation[,1:7])
```

```
##                PC1        PC2         PC3         PC4         PC5         PC6
## calories  0.3670078  0.3370596 -0.11462186  0.27660732  0.24758737  0.08912183
## protein  -0.2772241  0.2449194 -0.27689496  0.42031136 -0.18304085  0.12153045
## fat       0.1049438  0.3343406  0.20508488  0.60037932 -0.36602704 -0.10945914
## sodium    0.2015610  0.1150253 -0.38922969 -0.23670780 -0.32044177 -0.71728694
## fiber    -0.4180964  0.2880965 -0.06933016 -0.20825935 -0.04343596 -0.01086524
## carbo     0.1636662 -0.1948813 -0.56267964  0.20718081  0.38287999 -0.03790173
## sugars    0.2874024  0.3141683  0.35567565 -0.28893761  0.10210150  0.18011067
## potass   -0.3443208  0.3970189 -0.06712531 -0.09011254 -0.02365719  0.03488628
## vitamins  0.1557868  0.1196450 -0.38810943 -0.34092431 -0.49287096  0.48109792
## weight    0.1281124  0.4578598 -0.24665137 -0.12160132  0.41597684  0.12103468
## cups      0.2510333 -0.2738283 -0.14025953  0.12920887 -0.27373395  0.38620107
## rating   -0.4799624 -0.1586012 -0.18184294  0.06876466  0.13483653  0.14261284
##                PC7
## calories -0.009769147
## protein   0.147553524
## fat      -0.185133459
## sodium    0.222566270
## fiber     0.168689192
## carbo    -0.119675258
## sugars    0.204731752
## potass    0.202493463
## vitamins -0.463409721
## weight    0.100570037
## cups      0.737306160
```

```
## rating    0.011924746
```

For a healthy cluster we would recomment cluster #7 which has following properties

- High in protein
- High in carbs
- High in fiber
- High in potassium
- Highest ratings
- Low on calories
- Low on sugar

E.g. of the brands that fall into this cluster are (extrememe right of dendogram)

- Strawberry Fruit Wheats
- Maypo
- Shredded Wheat n'Bran
- Shredded Wheat