Stats 519: HW 6 - Cluster Analysis

Due on April 13, 2009

Dr. Stephen Lee 1:30

Monte J. Shaffer

Questions 1 to 4 will examine the INTL_FOODS dataset found on the course homepage, listed below, and described in detail on page 263 of your book. This dataset consists of the percentage of households in each of 16 European countries which have each of 20 different types of food in the house.

Problem 1

Use `dist` and `hclust` to cluster the countries according to food usage. Some factors to consider in your analysis include choice of clustering method, choice of distance measure, and whether or not to standardize the variables. Select your favorite solution, and identify the number of clusters as well as which countries are in each cluster. Examine the first few principal components, identifying cluster with `col` and/or `pch`. Does this suggest your solution does a good job separating out "natural" clusters?

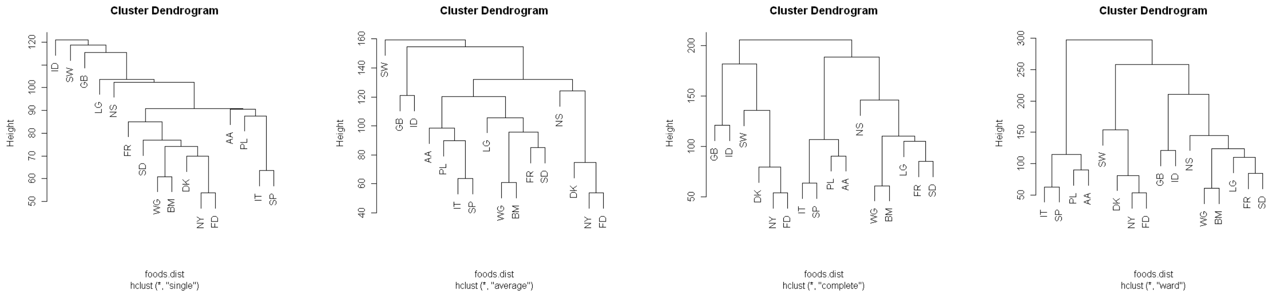


Figure 1: `hclust` from `dist`: Single, Average, Complete, Ward Methods

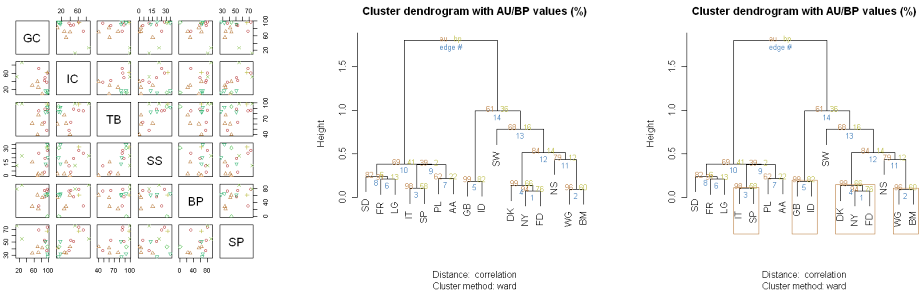


Figure 2: Pairs of first few foods and `pvclust`: Results and Best-Fits

Based on resampling techniques of `pvclust`, the results identify only 4 significant clusters ($AU = 1 - p - value$): Italy/Spain, Great Britain/Ireland, Denmark/Norway/Finland, West Germany/Belgium.

Program 1 Clustering using `dist` and `hclust`

```
setwd("C:/latex/statsMultiVariate/datasets");
     food = read.table("INTL_FOODS.txt");
myColumns = c("Ground coffee","Instant Coffee,"Tea","Sugarless Sweets","Biscuits","Soup packages","Soup

myRows = c("West Germany","Italy","France","Netherlands","Belgium","Luxembourg","Great Britain","Portuga

## scale across countries or food types?  #foods = scale(food);
foods = food;  ## data are percentages...

foods.dist = round(dist(foods),1); # default: euclidean, knn might be an interesting approach

                    DISTANCES ACROSS COUNTRIES [lower distance = more similar]

      WG    IT    FR    NS    BM    LG    GB    PL    AA    SD    SW    DK    NY    FD    SP
IT 123.0
FR 105.7 106.9
NS 102.5 178.9 136.0
BM  60.9 100.7  98.1 124.2
LG 104.6 142.9 104.8 145.8 110.1
GB 115.5 197.6 168.6 130.8 135.4 154.3
PL 136.1  92.0 138.8 188.4 125.0 146.3 205.3
AA 115.7 106.5 119.3 140.2 109.8 157.2 174.0  90.4
SD  76.9 114.1  85.1 118.3 102.9 103.7 141.4 128.2 108.3
SW 131.2 168.3 183.4 164.1 160.2 189.5 181.5 182.3 157.6 141.6
DK  74.2 152.2 133.4 114.2  99.3 135.1 137.0 161.7 122.8 123.6 121.4
NY  86.6 136.7 135.8 115.1 103.5 142.9 153.3 141.6 101.8 122.7 118.7  69.8
FD 105.6 143.7 142.1 143.5 114.6 156.3 167.6 141.7 106.5 139.5 135.4  79.5  53.7
SP 116.9  63.6  98.9 162.2  90.8 120.6 181.2  87.6  98.0 104.2 176.1 151.0 133.6 151.0
ID 128.9 156.4 149.5 153.3 140.8 182.1 120.9 181.4 136.2 129.0 178.2 145.7 143.5 153.3 162.5

foods.hclust.single = hclust(foods.dist , method="single");
     plot(foods.hclust.single);
foods.hclust.complete = hclust(foods.dist , method="complete");
     plot(foods.hclust.complete);
foods.hclust.ave = hclust(foods.dist , method="ave");
     plot(foods.hclust.ave);
foods.hclust.ward = hclust(foods.dist , method="ward");
     plot(foods.hclust.ward);
pairs(foods[,1:6], pch=cutree(foods.hclust.ward,6), col=cutree(foods.hclust.ward,6));

library(pvclust);
foods.pvClust.ward = pvclust(t(foods),method.hclust="ward");
     plot(foods.pvClust.ward);     # dendogram
     pvrect(foods.pvClust.ward);
```

Problem 2

Now use kmeans clustering to construct a solution with the same number of clusters as your solution to problem 1. Do the resulting clusters match those found in problem 1? Examine the data (broken down by clusters; you may want to look at pairs plots, or any subset of them) and the centers of the clusters. What are the key identifying features of each cluster; that is, what foods appear to distinguish each cluster, and in what way?

Program 2 kmeans

```
palette(rainbow(20, s = 0.6, v = 0.75));
# Determine number of clusters
wss = (nrow(foods)-1)*sum(apply(foods,2,var));
for (i in 2:15)
     {
     wss[i] = sum(kmeans(foods,centers=i)$withinss);
     }
plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares");

## I choose 4 based on SCREE and HW problem 1.

foods.kmeans.4 = kmeans(foods,4);
     stars(foods.kmeans.4$centers, len = 0.8, key.loc = c(7, 1.5), main = "Stars of KMEANS=4", draw.segm

foods.kmeans.5 = kmeans(foods,5);
     stars(foods.kmeans.5$centers, len = 0.8, key.loc = c(7, 1.5), main = "Stars of KMEANS=5", draw.segm
```

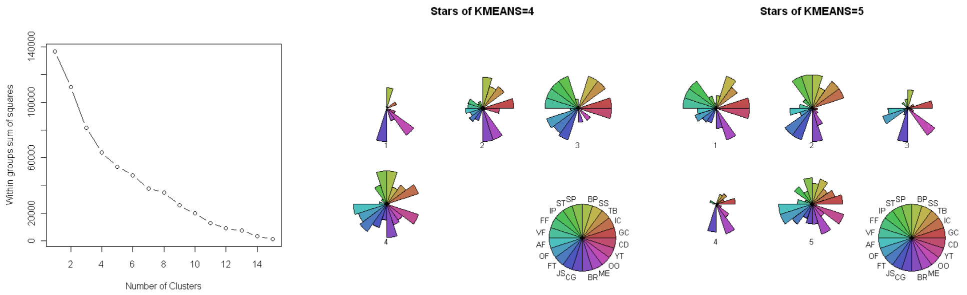


Figure 3: Within Sums of Squares (SCREE) plot, analysis of 4 and 5 clusters

Since I decided to use 4 clusters, I will describe the clusters based on the percentages of different foods consumed. Cluster 1 is high CG, YT and moderate BP: Garlic, Yogurt, and Package Biscuits. Cluster 2 is high BR, ME, GC, BP, and moderate TB, SS: Butter, Margarine, Ground Coffee, Packaged Biscuits, Tea, Sugarless sweets. Cluster 3 is high VF, FF, IP, ST, SS, TB, GC, CD, CG: Frozen vegetables, Frozen fish, Instant potatoes, Canned soup, Packaged soup, Tea, Ground cofee, Crispbread, and Garlic. Cluster 4 is high SP, BP, IC, YT, BR, FT, AF: Packaged soup, Packaged biscuits, Instant Coffee, Yogurt, Butter, Canned Fruit, Fresh apples.

Problem 3

Now transpose the original matrix with the `t` command, so that the observations are foods and the variables are countries. Repeat problem 1 for the foods, using `hclust` to identify clusters of foods.

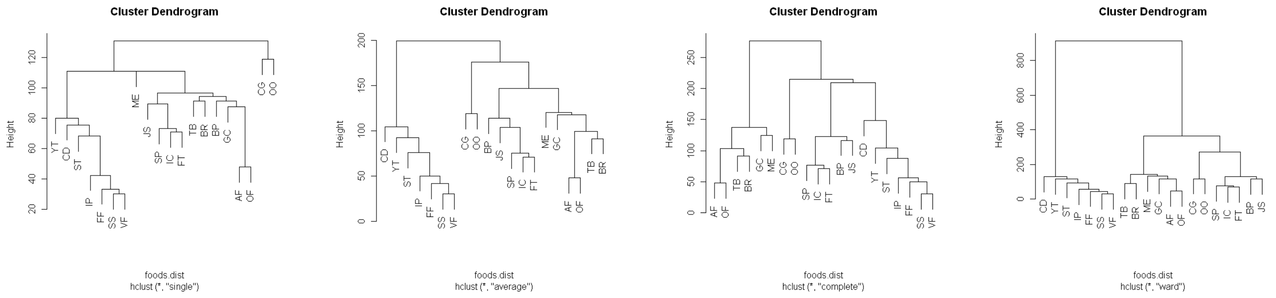


Figure 4: `hclust` from `dist`: Single, Average, Complete, Ward Methods

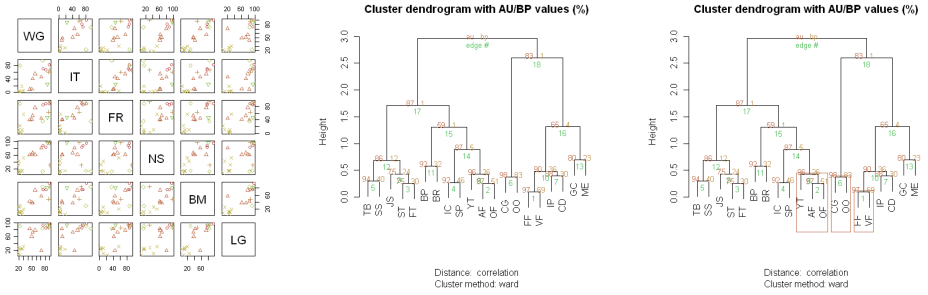


Figure 5: Pairs of first few foods and `pvclust`: Results and Best-Fits

Based on resampling techniques of `pvclust`, the results identify only 3 significant clusters ($AU = 1 - p - value$): Yogurt/Fresh Apples/Fresh Oranges, Garlic/Oil, Frozen fish/vegetables.

Program 3 Clustering using `dist` and `hclust`

```
foods = t(food);  ## transpose

foods.dist = round(dist(foods),1); # default: euclidean, knn might be an interesting approach

                    DISTANCES ACROSS FOODS [lower distance = more similar]

      GC    IC    TB    SS    BP    SP    ST    IP    FF    VF    AF    OF    FT    JS    CG    BR    ME
IC 222.4
TB 131.7 184.7
SS 261.5 127.6 251.8
BP 172.6 123.1 145.7 192.8
SP 173.6  73.1 137.3 144.3  99.9
ST 275.0 118.2 251.2  68.3 188.4 149.5
IP 276.5 144.2 273.6  52.1 209.7 161.2  76.9
FF 240.8 136.5 238.5  49.6 188.1 140.9  87.4  56.0
VF 264.7 140.7 258.8  30.0 200.8 155.2  70.8  42.4  33.3
AF 114.7 137.7 103.3 209.7  91.4  96.8 211.3 227.5 199.1 216.0
OF  87.4 155.5  94.5 222.3 109.3 106.2 224.1 237.1 205.8 226.9  47.9
FT 200.3  71.0 166.1 122.6 114.3  76.6 112.4 145.5 121.9 129.2 121.3 135.9
JS 179.8 117.2 121.4 172.8 116.6  89.5 159.4 189.6 165.9 178.6  99.6 117.1 104.1
CG 211.7 159.4 235.8 191.7 182.1 148.6 211.6 188.2 179.8 194.9 183.7 178.0 182.7 214.6
BR 137.2 183.7  91.2 251.3 121.6 143.3 252.3 266.5 235.9 254.9  97.1 100.9 168.0 139.4 213.9
ME 124.5 173.1 117.1 229.7 126.8 150.3 233.9 253.5 217.1 237.0 116.2 110.8 157.8 154.5 218.0 132.8
OO 175.0 150.3 186.5 203.0 151.9 130.9 205.6 204.8 182.9 201.5 144.6 132.9 158.3 167.0 118.8 162.1 178.5
YT 257.2 111.1 258.8  80.0 186.7 136.9 104.0  82.4 101.5  94.4 199.0 214.2 132.4 181.7 160.1 252.9 234.2
CD 228.4 163.1 223.6  95.5 198.1 149.6 117.5 100.2  75.6  89.3 201.0 202.2 137.1 163.2 208.6 227.7 215.5


foods.hclust.single = hclust(foods.dist , method="single");
     plot(foods.hclust.single);
foods.hclust.complete = hclust(foods.dist , method="complete");
     plot(foods.hclust.complete);
foods.hclust.ave = hclust(foods.dist , method="ave");
     plot(foods.hclust.ave);
foods.hclust.ward = hclust(foods.dist , method="ward");
     plot(foods.hclust.ward);
pairs(foods[,1:6], pch=cutree(foods.hclust.ward,6), col=cutree(foods.hclust.ward,6));

library(pvclust);
foods.pvClust.ward = pvclust(t(foods),method.hclust="ward");
     plot(foods.pvClust.ward);     # dendogram
     pvrect(foods.pvClust.ward);
```

Problem 4

Repeat problem 2 for the foods, using `kmeans` to identify clusters of foods and their key identifying features.

Program 4 `kmeans`

```
palette(rainbow(20, s = 0.6, v = 0.75));
# Determine number of clusters
wss = (nrow(foods)-1)*sum(apply(foods,2,var));
for (i in 2:15)
     {
     wss[i] = sum(kmeans(foods,centers=i)$withinss);
     }
plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares");

## I choose 3 based on SCREE and HW problem 1.

foods.kmeans.3 = kmeans(foods,3);
     stars(foods.kmeans.3$centers, len = 0.8, key.loc = c(4.5, 2.4), main = "Stars of KMEANS=3", draw.se

foods.kmeans.4 = kmeans(foods,4);
     stars(foods.kmeans.4$centers, len = 0.8, key.loc = c(7, 1.5), main = "Stars of KMEANS=4", draw.segm
```

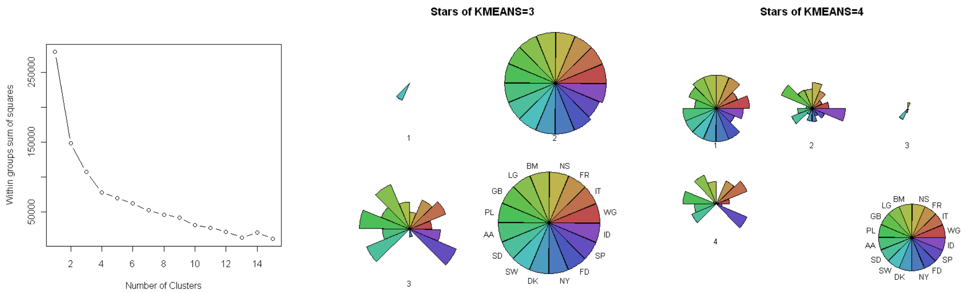
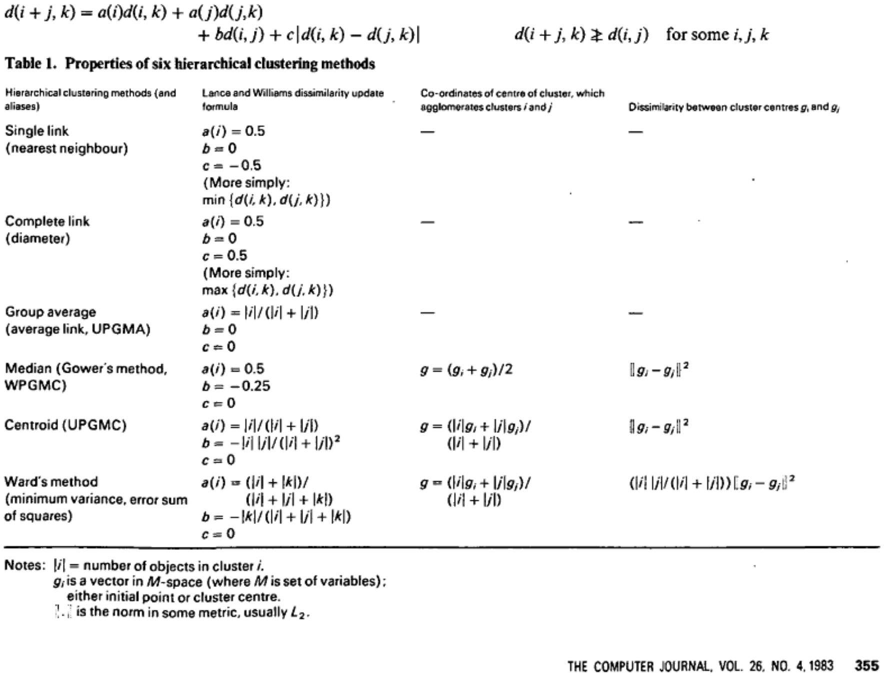


Figure 6: Within Sums of Squares (SCREE) plot, analysis of 3 and 4 clusters

Since I decided to use 3 clusters, I will describe the clusters based on the percentages of different foods consumed. Cluster 1 is a small Sweden (SW) group. Cluster 2 is a large group including all countries (slightly lower for Spain [SP]). Cluster 3 is high SP, SW, PL, LG: Spain, Sweden, Portugal, Luxembourg.

Problem 5

Brian Everitt Ex 6.1: Show that the intercluster distances used by single linkage, complete linkage, and group average clustering satisfy the following formula $d_{k(ij)} = \alpha_i d_{ki} + \alpha_j d_{kj} [+\beta d_{ij}] + \gamma |d_{ki} - d_{kj}|$ where $d_{k(ij)}$ is the distance between a group k and a group (ij) formed by the fusion of groups i and j; d_{ij} is the distance between groups i and j; n_i and n_j are the number of observations in groups i and j.



$d(i+j,k) = a(i)d(i,k) + a(j)d(j,k)$
$+ bd(i,j) + c|d(i,k) - d(j,k)|$ $\quad d(i+j,k) \ngeq d(i,j)$ for some i, j, k

Table 1. Properties of six hierarchical clustering methods

Hierarchical clustering methods (and aliases)	Lance and Williams dissimilarity update formula	Co-ordinates of centre of cluster, which agglomerates clusters i and j	Dissimilarity between cluster centres g_i and g_j
Single link (nearest neighbour)	$a(i) = 0.5$ $b = 0$ $c = -0.5$ (More simply: min $\{d(i,k), d(j,k)\}$)	—	—
Complete link (diameter)	$a(i) = 0.5$ $b = 0$ $c = 0.5$ (More simply: max $\{d(i,k), d(j,k)\}$)	—	—
Group average (average link, UPGMA)	$a(i) = \|i\|/(\|i\| + \|j\|)$ $b = 0$ $c = 0$	—	—
Median (Gower's method, WPGMC)	$a(i) = 0.5$ $b = -0.25$ $c = 0$	$g = (g_i + g_j)/2$	$\\|g_i - g_j\\|^2$
Centroid (UPGMC)	$a(i) = \|i\|/(\|i\| + \|j\|)$ $b = -\|i\|\,\|j\|/(\|i\| + \|j\|)^2$ $c = 0$	$g = (\|i\|g_i + \|j\|g_j)/$ $(\|i\| + \|j\|)$	$\\|g_i - g_j\\|^2$
Ward's method (minimum variance, error sum of squares)	$a(i) = (\|i\| + \|k\|)/$ $(\|i\| + \|j\| + \|k\|)$ $b = -\|k\|/(\|i\| + \|j\| + \|k\|)$ $c = 0$	$g = (\|i\|g_i + \|j\|g_j)/$ $(\|i\| + \|j\|)$	$(\|i\|\,\|j\|/(\|i\| + \|j\|))\\|g_i - g_j\\|^2$

Notes: $|i|$ = number of objects in cluster i.
g_i is a vector in M-space (where M is set of variables);
either initial point or cluster centre.
$\|\cdot\|$ is the norm in some metric, usually L_2.

THE COMPUTER JOURNAL, VOL. 26, NO. 4, 1983 355

Figure 7: Reference on distance formula, and unique values: [210] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," Comput. J., vol. 26, no. 4, pp. 354359, 1983.

- **a)** Single linkage: $\alpha_i = \alpha_j = \alpha = 0.5$ and $\gamma = -\frac{1}{2} = -0.5$ [and $\beta = 0$]

  Proof.

$$\begin{aligned} d_{k(ij)} &= \alpha_i d_{ki} + \alpha_j d_{kj} + \gamma |d_{ki} - d_{kj}| \\ &= \alpha d_{ki} + \alpha d_{kj} + -\frac{1}{2}|d_{ki} - d_{kj}| \\ &= min(d_{ik}, d_{jk}) \end{aligned} \tag{1}$$

□

- **b)** Complete linkage: $\alpha_i = \alpha_j = \alpha = 0.5$ and $\gamma = \frac{1}{2} = 0.5$ [and $\beta = 0$]

  Proof.

$$\begin{aligned} d_{k(ij)} &= \alpha_i d_{ki} + \alpha_j d_{kj} + \gamma |d_{ki} - d_{kj}| \\ &= \alpha d_{ki} + \alpha d_{kj} + \frac{1}{2}|d_{ki} - d_{kj}| \\ &= max(d_{ik}, d_{jk}) \end{aligned} \tag{2}$$

Problem 5 continued on next page. . .

□

- **c)** Average linkage: $\alpha_i = \frac{n_i}{n_i+n_j}$ and $\alpha_j = \frac{n_j}{n_j+n_i}$ and $\gamma = 0$ [and $\beta = 0$]

  Proof.

$$\begin{aligned} d_{k(ij)} &= \alpha_i d_{ki} + \alpha_j d_{kj} + \gamma |d_{ki} - d_{kj}| \\ &= \frac{n_i}{n_i + n_j} d_{ki} + \frac{n_j}{n_j + n_i} d_{kj} \end{aligned} \tag{3}$$

□

- **d)** Ward's method: $\alpha_i = \frac{n_i+n_k}{n_i+n_j+n_k}$ and $\alpha_j = \frac{n_j+n_k}{n_j+n_i+n_k}$ and $\gamma = 0$ [and $\beta = -\frac{n_k}{n_i+n_j+n_k}$]

Problem 6

Brian Everitt Ex 6.2: Ward (1963) proposed an agglomerative hierachical clustering procedure in which, at each step, the union of every possible pair of clusters is considered and the two clusters whose fustion results in the minimum increase in an error sum-of-squares criterion, ESS, are combined. For a single variable, ESS for a group with n individuals is simply $ESS = \sum_{i=1}^{n}(x_i - \bar{x})^2$.

- **a)** If ten individuals with variable values $\{2, 6, 5, 6, 2, 2, 2, 0, 0, 0\}$ are considered as a single group, calculate ESS.

  By reducing the within sum of squares, you create "tigher partitions" (WSS = ESS):

```
calculateESS = function(data)
     {
     myMean = mean(data);     myN = length(data); ESS = 0;
     for(i in 1:myN)
          {
          ESS = ESS + (data[i] - myMean)^2;
          }
     ESS;
     }

     > myTen = c(2,6,5,6,2,2,2,0,0,0);
     > calculateESS(myTen);
     [1] 50.5
     > Cluster1 = c(2,2,2,2,0,0,0);
     > calculateESS(Cluster1);
     [1] 6.857143
     > Cluster2 = c(6,5,6);
     > calculateESS(Cluster2);
     [1] 0.6666667
```

- **b)** Can you fit Ward's method into the general equation given in Everitt Ex 6.1 above?

  Yes, Figure 7 demonstrates that the Ward's method is defined by the general equation when the β term is included.

  Ward's method: $\alpha_i = \frac{n_i+n_k}{n_i+n_j+n_k}$ and $\alpha_j = \frac{n_j+n_k}{n_j+n_i+n_k}$ and $\gamma = 0$ [and $\beta = -\frac{n_k}{n_i+n_j+n_k}$]

Problem 7

Brian Everitt Ex 6.3: Reanalyze the pottery data using `Mclust`. To what model in `Mclust` does the $k-$mean approach approximate?

```
     summary(pottery.data);   # loaded from data command, standardized version.

# Determine number of clusters using k-means = 3
wss = (nrow(pottery.data)-1)*sum(apply(pottery.data,2,var));
for (i in 2:15)
     {
     wss[i] = sum(kmeans(pottery.data,centers=i)$withinss);
     }
plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares");
pottery.data.kmeans.3 = kmeans(pottery.data,3);
     stars(pottery.data.kmeans.3$centers, len = 0.8, key.loc = c(4.5, 2.5), main = "Stars of KMEANS=3",
pottery.data.kmeans.4 = kmeans(pottery.data,4);
     stars(pottery.data.kmeans.4$centers, len = 0.8, key.loc = c(6, 1.5), main = "Stars of KMEANS=4", d


# Determine number of clusters using Mclust = 4
library(mclust02);

pottery.data.Mclust=Mclust(pottery.data,1,20);
     max(pottery.data.Mclust$classification);
     table(pottery.data.Mclust$classification);
     pottery.data.Mclust$classification;
plot(pottery.data.Mclust,data=pottery.data);
          rownames(pottery.data.Mclust$mu)=colnames(pottery.data.kmeans.4$centers);
     stars(t(pottery.data.Mclust$mu), len = 0.8, key.loc = c(6, 1.5), main = "Stars of Mclust=4", draw.
```

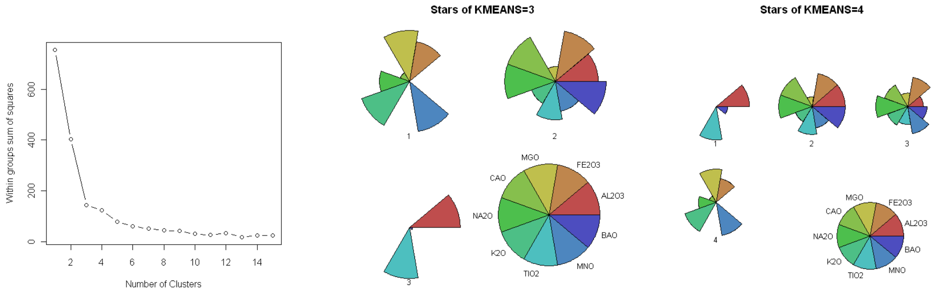


Figure 8: **K-means: Within Sums of Squares (SCREE) plot, analysis of 3 and 4 clusters**

If you compare 8 and 9, specifically K-means and Mclust at four clusters, you will see they are reporting the same values. K-means[1] is very similar to Mclust[4]; K-means[2] is very similar to Mclust[1]; K-means[3] is very similar to Mclust[2]; K-means[4] is very similar to Mclust[3]. Based on the WSS optimization of K-means, we could choose to keep 3 clusters; however, the elbow is decreasing at a decreasing rate between cluster 4 and 5, suggesting maybe cluster 4 can be kept. The optimization algorithm of Mclust (E-M) based

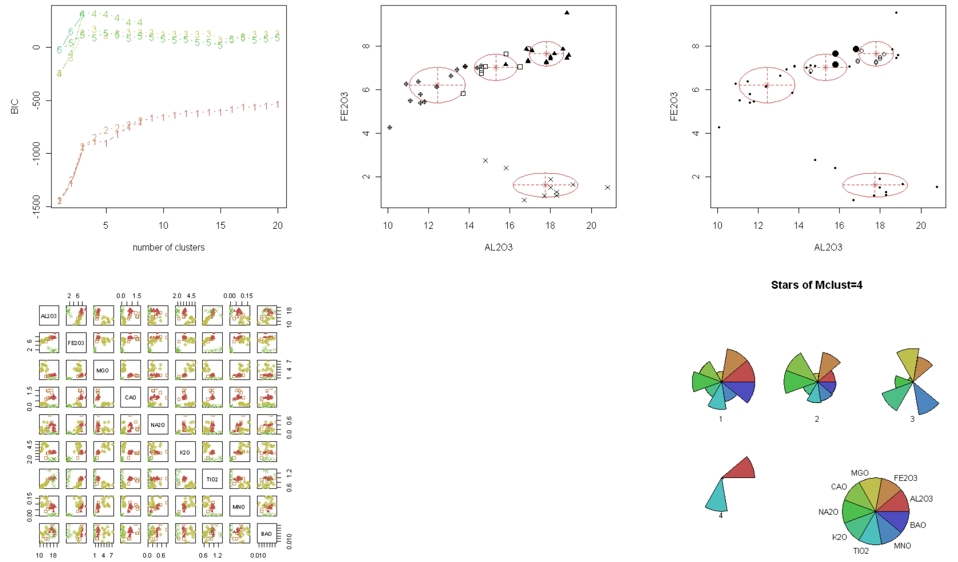


Figure 9: **Mclust: Analysis of mixing choice to get 4 clusters, pairs, stars of 4 clusters**

on a Gaussian mix suggests that 4 clusters are better. In the end, there are strong similarities, because the data is the same, and the techniques, albeit different, are both attempts at optimization.

Problem 8

Brian Everitt Ex 6.4: Construct a three-dimensional drop-line scatterplot of the plaents data in which the points are labelled with a suitable cluster label.

```
     # loaded from data command file.
planet=planet.dat;       # relative values to Jupiter's mass, Earth's revolution

palette(rainbow(12, s = 0.6, v = 0.75));
     library(mclust02);
planet.Mclust = Mclust(planet,1,9);
     planet.Mclust$classification;
     max(planet.Mclust$classification);
     table(planet.Mclust$classification);

     rownames(planet.Mclust$mu)=c("Mass","Period","Eccentricity");

     plot(planet.Mclust,data=planet);
stars(t(planet.Mclust$mu), len = 0.8, key.loc = c(4.5, 2.5), main = "Stars of Mclust=3", draw.segments =

> planet.Mclust$mu
                      1            2           3
Mass         1.15558719    5.8070403   1.5391731
Period       6.44901039 1263.0126523 303.8176595
Eccentricity 0.03536079    0.3637751   0.3075079
```

Problem 8 continued on next page...

```
# 3D Scatterplot with Coloring and Vertical Drop Lines
library(scatterplot3d);
     attach(planet);
     myPCH = 6+9*(planet.Mclust$classification-1);
scatterplot3d(Period,Mass,Eccentricity, highlight.3d=TRUE,  pch=myPCH, main="3D Scatterplot");
scatterplot3d(Period,Mass,Eccentricity, highlight.3d=TRUE, type="h", pch=myPCH, main="3D Scatterplot");
```

Cluster 1 is so small compared to the other clusters, it doesn't even appear in the Mclust stars graph. This is a reality of exo-planets. [Eccentricity may be interpreted as a measure of how much this orbit deviates from a circle.]

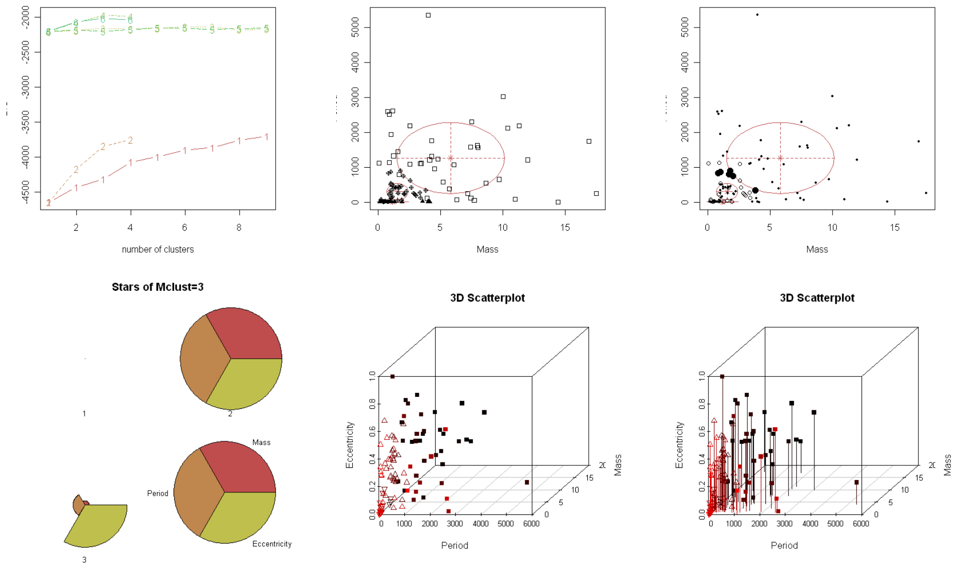


Figure 10: **Planets: Analysis of mixing choice to get 3 clusters, Stars of Mclust, 3-D scatterplots without and with droplines.**