Predictive Modeling Notes 8/3/2015 Session #2

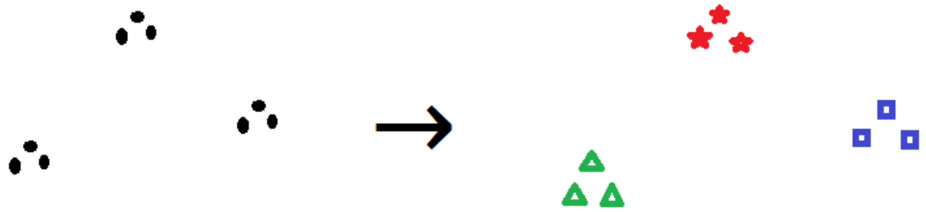
Matt Bonfante, Alec Grubbs, Erik Kahnke

(4) Latent Classes

Basics of Clustering
Introduction to K-Means Clustering

Clustering

1. Definition: dividing data points into categories which are not defined in advance
    a. (Latent-Class Model)

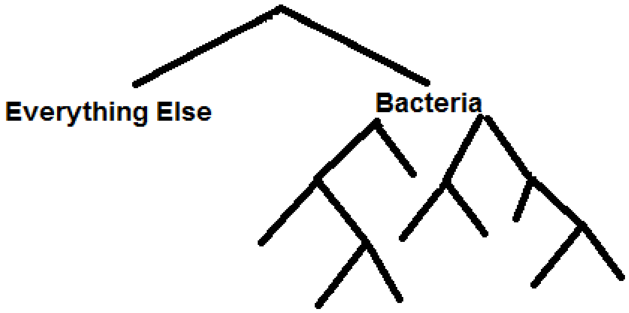


Three Clusters of Data Points

Associated Cluster Labels

Criteria for "Clusters":

1. Clusters should partition the data
    a. Mutually Exclusive, Collectively Exhaustive (MECE)
2. Data points within a cluster are similar/homogenous/close
3. Different clusters should be different
4. Clusters should be balanced
    a. Example of unbalanced clustering:



5. Clusters should be few in number

Clustering Algorithms need to define the notion of distance.

Consider the following:

Two data points: x_i, x_j

Distance between x_i, x_j: $d(x_i, x_j)$

What Makes a Distance?

1. $d(x_i, x_j) \geq 0$
    a. Distances are never negative
2. $d(x_i, x_j) = 0$ if and only if $x_i = x_j$
    a. Distance is equal to zero if x_i and x_j correspond to the same data point
3. $d(x_i, x_j) = d(x_j, x_i)$
    a. Distances are symmetric
4. $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k)$
    a. Triangle Inequality

Distance Formulas

Euclidean distance (ℓ^1) formula where $x_i, x_j \in \mathbb{R}^D$

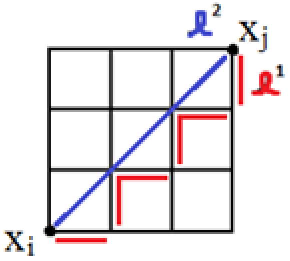
$$d(x_i, x_j) = \sqrt{\sum_{d=1}^{D} (x_{id} - x_{jd})^2}$$

Just the Pythagorean Theorem.

ℓ^2 distance formula (also known as Manhattan or Taxi-cab distance)

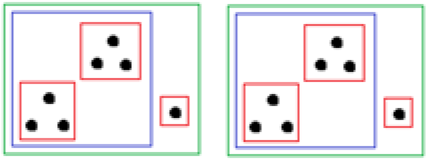
$$d(x_i, x_j) = \sum_{d=1}^{D} |x_{id} - x_{jd}|$$

Comparing the distance formulas:



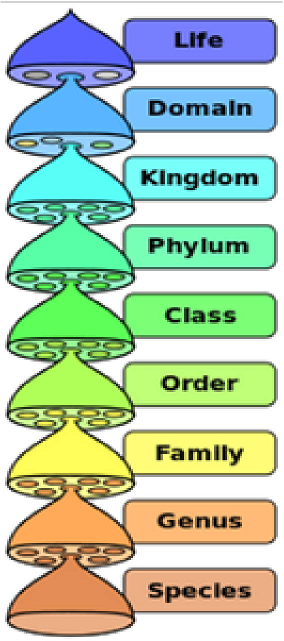
Difficulties with Clustering

1) Clusters are Ambiguous

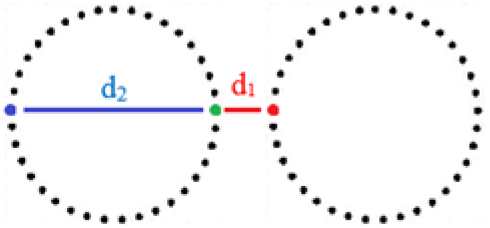


How many clusters are there? 2, 4, or 6?
There may be disagreement on the right number of clusters. This gets harder with higher dimensions.

2) Partitional vs. Hierarchical Clustering
Partitional means there is no association between adjacent clusters.
Hierarchical means there is an association between clusters.
An example would be the Linnaus rank-based system for biological classification.



3) Distance Based Clustering Isn’t Magic

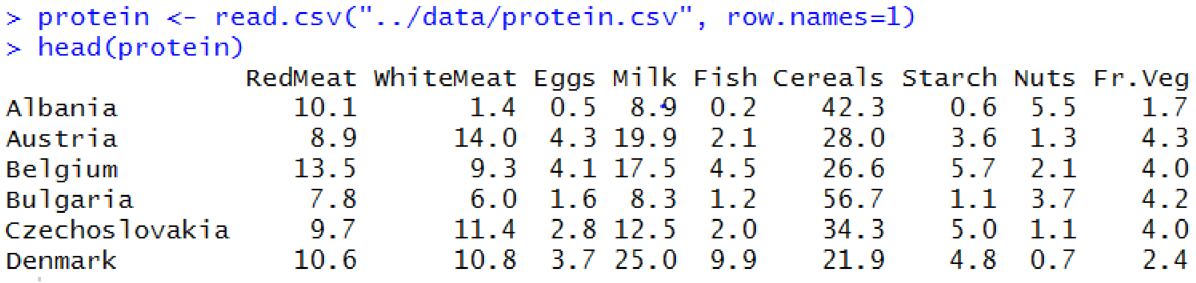


Should the green dot be clustered with the circle of dots on the left or the right? Visually, it makes sense to include it in the circle of dots on the left. However, in this case, $d_1 > d_2$ so going just be our distance formula, the green dot should be in the same cluster as the red dot. The definition of the distance may need to be changed to accurately capture visual groupings.

Protein.R Example

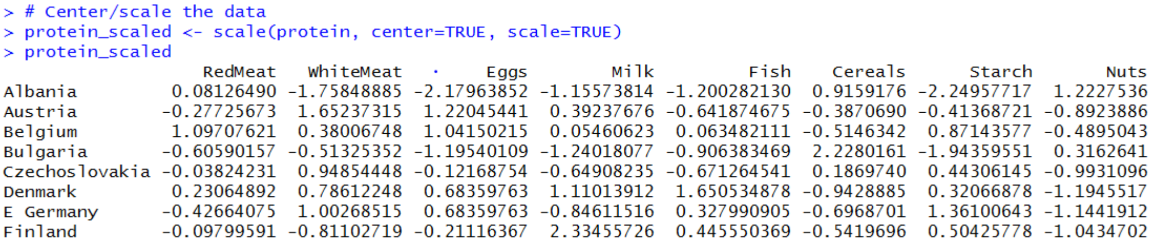
The data used is a CSV file with 25 rows of individual European countries and 9 attributes/columns detailing the amount of protein in grams received from various food sources on a daily basis.

First we read in the data and look at the data for the first 5 countries



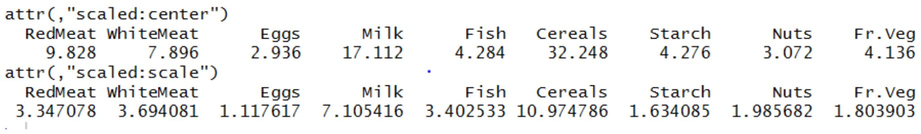
```
> protein <- read.csv("../data/protein.csv", row.names=1)
> head(protein)
               RedMeat WhiteMeat Eggs Milk Fish Cereals Starch Nuts Fr.Veg
Albania           10.1       1.4  0.5  8.9  0.2    42.3    0.6  5.5    1.7
Austria            8.9      14.0  4.3 19.9  2.1    28.0    3.6  1.3    4.3
Belgium           13.5       9.3  4.1 17.5  4.5    26.6    5.7  2.1    4.0
Bulgaria           7.8       6.0  1.6  8.3  1.2    56.7    1.1  3.7    4.2
Czechoslovakia     9.7      11.4  2.8 12.5  2.0    34.3    5.0  1.1    4.0
Denmark           10.6      10.8  3.7 25.0  9.9    21.9    4.8  0.7    2.4
```

Next we scale the data. This function determines the mean and SD of each attribute, and then converts each row into z-scores for each attribute. They can now be compared easily. Below, Center and Scale are set to TRUE by default, but it was shown coded for learning purposes.



```
> # Center/scale the data
> protein_scaled <- scale(protein, center=TRUE, scale=TRUE)
> protein_scaled
                   RedMeat   WhiteMeat        Eggs        Milk         Fish    Cereals      Starch       Nuts
Albania         0.08126490 -1.75848885 -2.17963852 -1.15573814 -1.200282130  0.9159176 -2.24957717  1.2227536
Austria        -0.27725673  1.65237315  1.22045441  0.39237676 -0.641874675 -0.3870690 -0.41368721 -0.8923886
Belgium         1.09707621  0.38006748  1.04150215  0.05460623  0.063482111 -0.5146342  0.87143577 -0.4895043
Bulgaria       -0.60590157 -0.51325352 -1.19540109 -1.24018077 -0.906383469  2.2280161 -1.94359551  0.3162641
Czechoslovakia -0.03824231  0.94854448 -0.12168754 -0.64908235 -0.671264541  0.1869740  0.44306145 -0.9931096
Denmark         0.23064892  0.78612248  0.68359763  1.11013912  1.650534878 -0.9428885  0.32066878 -1.1945517
E Germany      -0.42664075  1.00268515  0.68359763 -0.84611516  0.327990905 -0.6968701  1.36100643 -1.1441912
Finland        -0.09799591 -0.81102719 -0.21116367  2.33455726  0.445550369 -0.5419696  0.50425778 -1.0434702
```

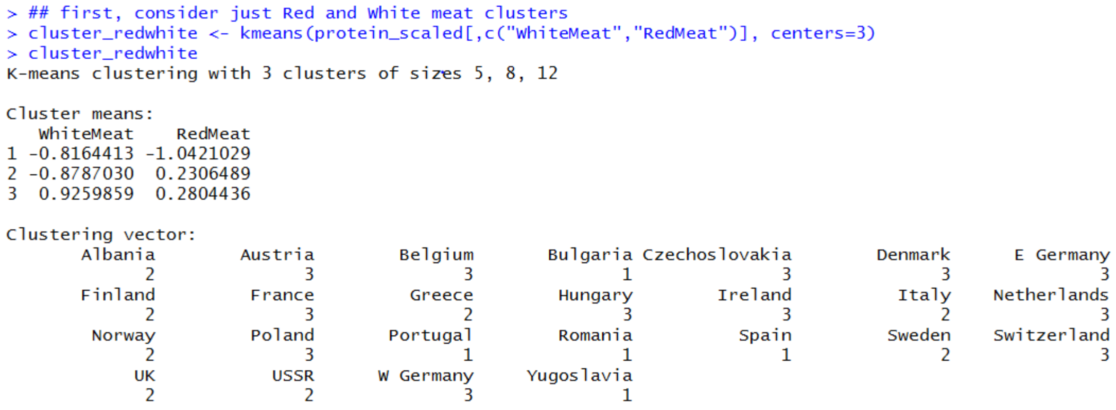
This also shows the means and SD for each attribute, which were used in determining the z-scores above.



```
attr(,"scaled:center")
  RedMeat WhiteMeat      Eggs      Milk      Fish   Cereals    Starch      Nuts    Fr.Veg
    9.828     7.896     2.936    17.112     4.284    32.248     4.276     3.072     4.136
attr(,"scaled:scale")
  RedMeat WhiteMeat      Eggs      Milk      Fish   Cereals    Starch      Nuts    Fr.Veg
 3.347078  3.694081  1.117617  7.105416  3.402533 10.974786  1.634085  1.985682  1.803903
```

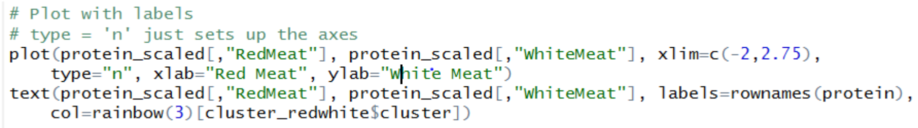
Next we cluster the scaled data. This instance only clusters on “WhiteMeat” and “RedMeat”. Center=3 means that the data will be separated into 3 clusters.

By printing cluster_redwhite we can see the means for each attribute within the clusters, as well as which cluster each country was placed into.



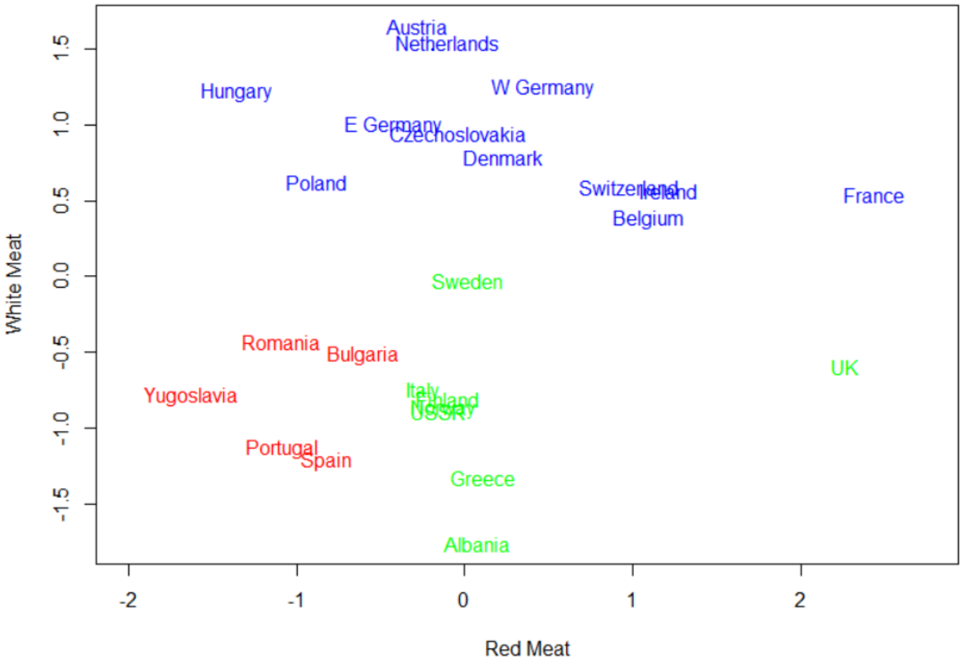
```
> ## first, consider just Red and White meat clusters
> cluster_redwhite <- kmeans(protein_scaled[,c("WhiteMeat","RedMeat")], centers=3)
> cluster_redwhite
K-means clustering with 3 clusters of sizes 5, 8, 12

Cluster means:
   WhiteMeat    RedMeat
1 -0.8164413 -1.0421029
2 -0.8787030  0.2306489
3  0.9259859  0.2804436

Clustering vector:
       Albania        Austria        Belgium       Bulgaria Czechoslovakia        Denmark      E Germany 
             2              3              3              1              3              3              3 
       Finland         France         Greece        Hungary        Ireland          Italy    Netherlands 
             2              3              2              3              3              2              3 
        Norway         Poland       Portugal        Romania          Spain         Sweden    Switzerland 
             2              3              1              1              1              2              3 
            UK           USSR      W Germany     Yugoslavia 
             2              2              3              1 
```

Next we plot cluster_redwhite by setting attributes redmeat/whitemeat to x/y , setting the range of x values and creating labels.

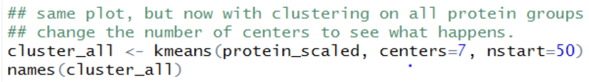


```
# Plot with labels
# type = 'n' just sets up the axes
plot(protein_scaled[,"RedMeat"], protein_scaled[,"WhiteMeat"], xlim=c(-2,2.75),
    type="n", xlab="Red Meat", ylab="White Meat")
text(protein_scaled[,"RedMeat"], protein_scaled[,"WhiteMeat"], labels=rownames(protein),
    col=rainbow(3)[cluster_redwhite$cluster])
```

The graphical output can be seen below split into 3 color coded clusters.

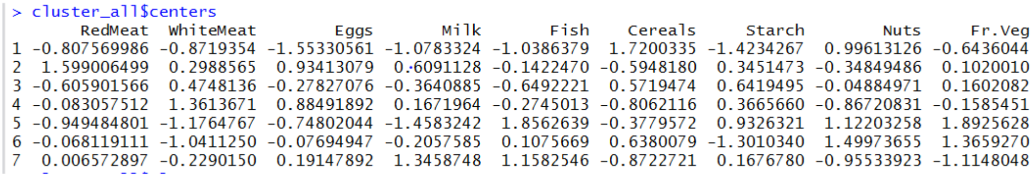


Now we will create a new variable called cluster_all that will create clusters based off all 9 attributes, and create 7 different cluster bins denoted by centers=7. nstart=50 means clusters will be created starting at 50 different points and then choosing the best cluster output.



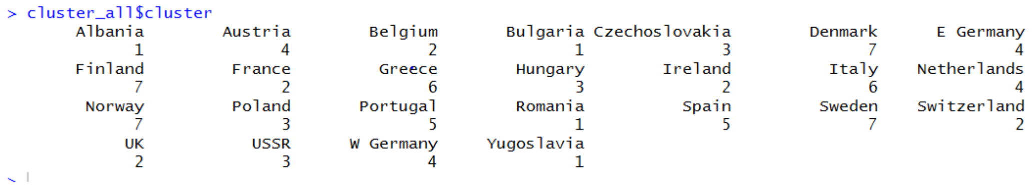
```
## same plot, but now with clustering on all protein groups
## change the number of centers to see what happens.
cluster_all <- kmeans(protein_scaled, centers=7, nstart=50)
names(cluster_all)
```

By printing cluster_all$centers, we can see the means of each attribute (protein) within each cluster group.



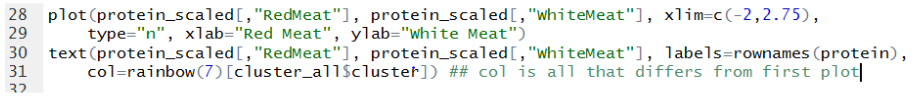
```
> cluster_all$centers
      RedMeat  WhiteMeat        Eggs       Milk       Fish    Cereals     Starch        Nuts     Fr.Veg
1 -0.807569986 -0.8719354 -1.55330561 -1.0783324 -1.0386379  1.7200335 -1.4234267  0.99613126 -0.6436044
2  1.599006499  0.2988565  0.93413079  0.6091128 -0.1422470 -0.5948180  0.3451473 -0.34849486  0.1020010
3 -0.605901566  0.4748136 -0.27827076 -0.3640885 -0.6492221  0.5719474  0.6419495 -0.04884971  0.1602082
4 -0.083057512  1.3613671  0.88491892  0.1671964 -0.2745013 -0.8062116  0.3665660 -0.86720831 -0.1585451
5 -0.949484801 -1.1764767 -0.74802044 -1.4583242  1.8562639 -0.3779572  0.9326321  1.12203258  1.8925628
6 -0.068119111 -1.0411250 -0.07694947 -0.2057585  0.1075669  0.6380079 -1.3010340  1.49973655  1.3659270
7  0.006572897 -0.2290150  0.19147892  1.3458748  1.1582546 -0.8722721  0.1676780 -0.95533923 -1.1148048
```

By printing cluster_all$cluster, we can view which cluster each country falls into. Just by looking at this we can see it makes sense. Two mediterranean countries, Italy & Greece, were placed in the same clusters as they probably have similar diets as one would expect. The same can be easily seen in cluster 4.



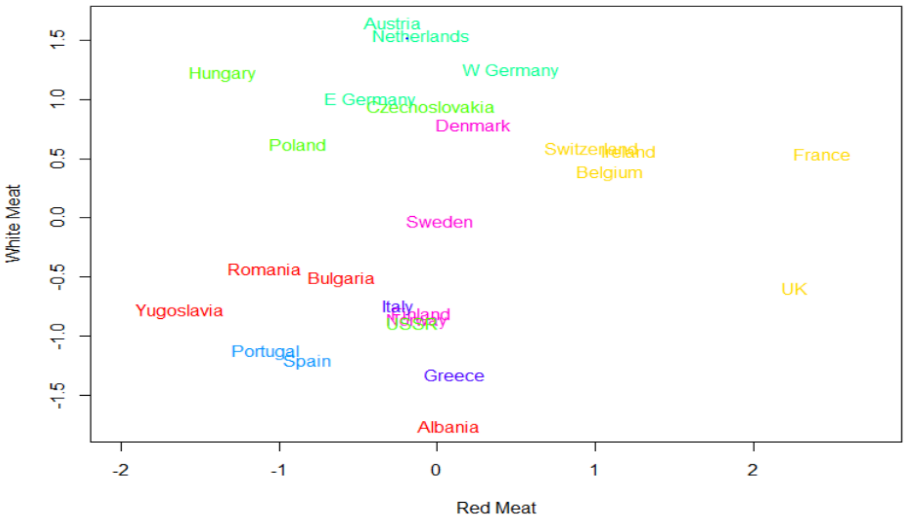
```
> cluster_all$cluster
       Albania        Austria        Belgium       Bulgaria Czechoslovakia        Denmark      E Germany 
             1              4              2              1              3              7              4 
       Finland         France         Greece        Hungary        Ireland          Italy    Netherlands 
             7              2              6              3              2              6              4 
        Norway         Poland       Portugal        Romania          Spain         Sweden    Switzerland 
             7              3              5              1              5              7              2 
            UK           USSR      W Germany     Yugoslavia 
             2              3              4              1 
```

Finally we plot the data again observing RedMeat & WhiteMeat for easy viewing in 2-dimensions. Very little changes except now we denote 7 colors for each respective cluster.



```
plot(protein_scaled[,"RedMeat"], protein_scaled[,"WhiteMeat"], xlim=c(-2,2.75),
     type="n", xlab="Red Meat", ylab="White Meat")
text(protein_scaled[,"RedMeat"], protein_scaled[,"WhiteMeat"], labels=rownames(protein),
     col=rainbow(7)[cluster_all$cluster]) ## col is all that differs from first plot
```

The graphical output of this is shown below. Although the graph may at first seem counterintuitive as the clusters do not look well separated, this is due to the fact that the clusters were built using 9 dimensions, but are only being displayed in a 2-dimensional graph.


Austria
Netherlands
Hungary
W Germany
E Germany
Czechoslovakia
Denmark
Poland
Switzerland
Ireland
France
Belgium
Sweden
Romania
Bulgaria
UK
Yugoslavia
Italy
Finland
Norway
USSR
Portugal
Spain
Greece
Albania
White Meat
Red Meat