**RESEARCH AND SURVEY STATISTICS – STA3022F**

**COMPUTER PRACTICAL 5**

**CLUSTER ANALYSIS and MULTIDIMENSIONAL SCALING**

1. Open the script file **prac5.R**.

From the top menu, select File, Open File

Select the file prac5.R which you saved from Vula

Click Open

The script file appears in the Window in the top left.

1. **Example: Cluster Analysis**

We will use a data set on protein consumption in twenty-five European countries for nine food groups. The data set is available at <http://lib.stat.cmu.edu/DASL/Stories/ProteinConsumptioninEurope.html>.

Load the file protein.csv into R.

Although some variables have larger values than others, all variables indicate the proportion of that food group eaten on average in that country. If less of that food is eaten, it should contribute less to the analysis. We will therefore not standardise the data here.

Run lines 2 and 3 to create the data matrix .

Ensure you activate the package cluster by selecting it under Packages in the bottom right or by running line 4: require (cluster).

# Single linkage clustering

## Perform Single linkage clustering with Euclidean distance.

Run lines 7 and 8. Decide subjectively to cut the tree at height 9. Run lines 9 to 13.

## Perform Single linkage clustering with City block distance.

Run lines 15 and 16. Decide subjectively to cut the tree into 6 clusters. Run lines 17 to 21.

## Perform Single linkage clustering with Correlation.

Run lines 23 and 24. (*This behaviour to add one item to the existing cluster at each merging stage is the biggest problem with Single linkage and known as chain forming*.)

# Complete linkage clustering

## Perform Complete linkage clustering with Euclidean distance.

Run lines 27 and 28. Decide subjectively to cut the tree into 5 clusters. Run lines 29 to 33.

## Perform Complete linkage clustering with City block distance.

Run lines 35 and 36. Decide subjectively to cut the tree at height 40. Run lines 37 to 41.

## Perform Complete linkage clustering with Correlation.

Run lines 43 and 44. Decide subjectively to cut the tree into 5 clusters. Run lines 45 to 49.

# Centroid clustering

## Perform Centroid clustering with Euclidean distance.

Run lines 52 and 53.

## Perform Centroid clustering with City block distance.

Run lines 54 and 55.

## Perform Centroid clustering with Correlation.

Run lines 56 and 57.

# Perform K-means clustering.

You need to decide on the number of clusters.

Set the number of clusters selected by running line 60.

Perform K-means clustering for 6 clusters with line 62-65.

Notice that the clustering of the K-means algorithm into six clusters agrees with that of the Complete linkage algorithm based on Euclidean distance. The only different being that Belgium, France, UK, Austria, Ireland, Netherlands, Switzerland and West Germany is split into two clusters by the K-means. These two algorithms provides sensible results in terms of the spatial location of the countries. We will use the clustering from the Complete linkage with Euclidean distance into 5 clusters for the remainder of the analysis.

1. **Example: Cluster profiling**

Select the clustering output to use in the profiling analysis with line 68.

Compute the means for each cluster for each variable. Run lines 70 to 72. Create a plot of the cluster means to illustrate differences between clusters. Run lines 74 to 80.

The blue (no 3) cluster (Bulgaria, Romania and Yugoslavia) tend to eat more cereal and nuts than the other countries. The green (no 1) cluster (Czechoslovakia, Hungary, USSR, Poland, Albania, Greece and Italy) also eat more cereal and nuts than the remaining countries, but less than Bulgaria, Romania and Yugoslavia and more of the other food groups. The black (no 5) cluster (East Germany, Portugal and Spain) also eat fair amounts of cereal and nuts, but use less milk than all the other countries and eat the most fish and fruit and veg. The yellow (no 2) cluster (West Germany, Austria, Switzerland, France, Belgium, Netherlands, UK and Ireland) eat the most red and white meat and use fair amounts of milk. The red (no 4) cluster (Scandinavia) eats lots of fish, use the most milk at the yellow cluster, eat fair amounts of red and white meat and eat the smallest quantities of cereal, nuts and fruit and veg.

1. **Example: Multidimensional scaling**

We will colour the different MDS outputs according to the clustering in 5 clusters from the Complete Linkage with Euclidean distance. If one wants to represent the clustering output in MDS the distances used for MDS must be the same as that used for the clustering algorithm.

Perform Classical scaling. Run lines 87 to 94.

Perform Metric SMACOF. Run lines 96 to 104.

Perform Non-metric SMACOF. Run lines 106 to 114.

Perform a Sammon MDS. Run lines 116 to 124.

Perform Kruskal’s non-metric scaling. Run lines 126 to 134.

1. **Donor data analysis for Quiz:**

The detailed information on frequency and amount donated for major donors from the data frame derived will be used. The final column with coded information on Recency, Frequency and Amount of donations will be included in the analysis.

If the data frame derived is not available in your workspace, import it from the file derived.csv.

Euclidean distances will be used for the numeric data and the distances for categorical data will be calculated separately before combining the two sets of distances with scaling to give each equal weight in the analysis. Run lines 138 to 153.

Perform Single linkage, Complete linkage, Centroid Clustering with code similar to that of lines 7 and 8, 27-28, 52-53, replacing the calls to daisy() with the distance object Dmat:

NB: Instead of the following in line 7:

out.single.euc<-hclust(daisy(X,metric="euclidean"),method="single")

You must change it to the following:

out.single.euc <- hclust(Dmat,method="single")

Motivate why the Complete linkage output is preferred above the other methods for this data set. Cut your Complete linkage clustering tree at a height of 20 with code similar to line 38. You can use the command

max(clusvec) to find the number of clusters that was formed.

Use Sammon mapping to obtain a 2D representation of the data with code similar to lines 116 − 124. Again, replace the call to daisy() with the distance object Dmat. To make an extended list of colours, you can get a list of all the colour-terms R understands by running the command

colors()

For cluster profiling, use code similar to lines 71-72, replacing the data matrix ‘X’ with the R object ‘numeric.dat’. For the three categorical variables you can obtain profiles with the following code:

table(clusvec,cat.dat$Recency)

table(clusvec,cat.dat$Freq)

table(clusvec,cat.dat$Amount)

Complete the Quiz Prac5.

Variable frequencies ‘freq94’, ‘freq95’, ‘freq96’: describe how many times donations are done for each year 1994, 1995, and 1996 respectively.

Variables ‘freqT1 … freqT8’: For each type of donations, these variables describe how many times donations are done.

Variables ‘amount94 … amount96’: Amount of donations given for each year

Variable ‘amountT1 … amountT8’: Amount of donations given for each type of donations

MDMAUD The Major Donor Matrix code

The codes describe frequency and amount of

giving for donors who have given a $100+

gift at any time in their giving history.

An RFA (recency/frequency/monetary) field.

The (current) concatenated version is a nominal

or symbolic field. The individual bytes could separately be

used as fields and refer to the following:

First byte: Recency of Giving

C=Current Donor

L=Lapsed Donor

I=Inactive Donor

D=Dormant Donor

2nd byte: Frequency of Giving

1=One gift in the period of recency

2=Two-Four gifts in the period of recency

5=Five+ gifts in the period of recency

3rd byte: Amount of Giving

L=Less than $100(Low Dollar)

C=$100-499(Core)

M=$500-999(Major)

T=$1,000+(Top)