Workshop 4
Overview of EM clustering and a Practical Cluster Analysis of the PIOMAS data

- Make sure you are up to date with Workshops 1 to 3.
- Exercise one is a quick look at Clustering using the EM algorithm
- The main focus of this workshop is applying the cluster analysis methods you have learnt to a real environmental data set.

Exercise 1 EM clustering

You will use a data set that you are familiar with the `USArrests` data, and compare the K-Means results with the EM-clustering results.

First of all start you R session in the usual way and install the R package `"EMCluster"`.
Run a K-means clustering with 4 clusters and plot the clusters for first 2 principal comonents

```
pr.out<-prcomp(USArrests,scale=TRUE)
plot(pr.out$x[,1:2],type="n")
text(pr.out$x[,1:2],labels=state.abb,col=km.out$cluster)
```

The approach in the `EMCluster` package is to set up an initial solution using *init.EM* and then run the EM algorithm to get the optimal solution using the `emcluster` function. In the third step we obtain the probabilities of the optimal solution by running one more `e.step`.

```
emobj<- init.EM(USArrests, nclass = 4)
emclobj <- emcluster(USArrests, emobj, assign.class = TRUE)
emprobs <- round(e.step(USArrests, emobj = emclobj)$Gamma,3)
```

When the data have more than two variables, the default plot for an EM-object (with class `emret`) is A *parallel coordinates plot*: This is rarely helpful.

```
plotem(emobj,USArrests,lwd=2)
```

Instead plot the 1st 2 principal components again and compare this with the K-Means plot.

```
plot(pr.out$x[,1:2],type="n")
text(pr.out$x[,1:2],labels=state.abb,col=emclobj$class)
```

We can look at each K-means cluster in more detail by returning the EM-algorithm probabilities. Here is the code for the 1st cluster. repeat this for each of the four K-means clusters.

```
round(emprobs[km.out$cluster==1,],3)
```

Exercise 2 PIOMAS data

The main part of this workshop is to analyse data published by the Polar Research Center, University of Washington, USA. The Arctic sea ice volume is calculated using the Pan-Arctic Ice Ocean Modeling and Assimilation System http://psc.apl.uw.edu/research/projects/arctic-sea-ice-volume-anomaly/. Monthly measurements are available since 1979 and are easy to download and load into R. You will use the methods learnt in weeks 1 to 3 to identify clusters in this dataset. Feel free to use R techniques learned in other courses such as data visualisation. These data were chosen to demonstrate the analysis of a practical dataset in an interesting application area, rather than a as dataset to teach clustering, which means there are no right or wrong answers to this analysis.

Add comments your code as you work, so you can refer back to the code later and save the script file regularly. Any specific remarks or findings can be added to the code as comments. If you are familiar with R-Markdown you can present your cluster analysis in a neat format.

Download the monthly PIOMAS data from Moodle (the original data was downloaded from the website http://psc.apl.uw.edu/research/projects/arctic-sea-ice-volume-anomaly/data/). The Arctic sea ice volume in is given 1000 km^3 for every month since January 1979 until March 2018. These data are in a very clean format: one line for each year, space delimited and no missing values, but as always with data analysis we need to shape the data into a format suitable for our analysis.

The following code reads in the data, uses the year as row names rather than the first variable and assigns meaningful variable names. Because 2018 has incomplete data (why?), we will remove it from out data frame. One line for each year is plotted using the function `matplot`.

```
##data
PIOMAS<-read.table("???/PIOMAS.txt",header=FALSE,row.names=1)
names(PIOMAS)<-month.abb
year<-as.numeric(row.names(PIOMAS))
dim(PIOMAS)
PIOMAS<-PIOMAS[-40,]
matplot(t(PIOMAS),type="l",xlab="Month",ylab="Sea Ice Volume")
```

The mean values for each year can be plotted.

```
yave<-apply(PIOMAS,1,mean)
plot(year,yave,ylab="Sea Ice Volume")
```

Start by looking at K-means clustering. Choose a suitable value for K and look at the results graphically using

```
plot(year,yave,ylab="Sea Ice Volume",col=???)
matplot(t(PIOMAS),type="l",xlab="Month",ylab="Sea Ice Volume",col=???)
```

Now investigate the principal components of the PIOMAS data. What do you notice about the variance in the first few PCs?

These data form a time series and are very highly *autocorrelated*, the value of one value is closely related to the value of the previous one. Most of this autocorrelation can be removed from the data by considering the month on month differences in the sea ice volume. Notice that the first variable in each row is the difference February minus January, hence the odd variable names.

```
#read in the original data and convert to a vector
tt<-c(t(read.table("Data/PIOMAS.txt",header=FALSE,row.names=1)))
#calculate the differences for the first 39 years of data
tt<-diff(tt[1:(39*12+1)])
#put back into data frame format
PIOdiff<-as.data.frame(matrix(tt,byrow=T,ncol=12))
#variable names
names(PIOdiff)<-month.abb[c(2:12,1)]
#row names
row.names(PIOdiff)<-row.names(PIOMAS)
matplot(t(PIOdiff),type="l")
```

Large negative values imply a large amount of sea ice melting in that month.

Continue with the cluster analysis including the following methods for as long as time allows.

- K-means clustering of the difference data.
- PCA of the difference data
- K-Medoids
- Hierarchical clustering
- EM clustering