Series 5

1. The dataset *fossilien.dat* contains measurements (length, width of the corpus, ...) of “cocoliths” of the species *Gephyrocapsa* and other variables as salt content, temperature and Chlorophyll. There is a short explanation of some variables:

- `lLength`: log10(Length)
- `sAngle`: sqrt(Angle)
- `rWidth`: Width/Length
- `rClength`: CLength/Length
- `Cratio`: Cwidth/Clength / (Width/Length)
- `SST.mean`: Annual Mean Temperature
- `Salinity`: salinity
- `lChlorophyll`: log10(Chlorophyll)

In this exercise we would like to learn the method of multivariate regression.

a) Load the data and have a first look at it.

b) We would like to test, whether the body size is associated with the environmental conditions during that period. Make a multivariate regression. Target variables are `sAngle`, `lLength` and `rWidth`; predictors are `SST.Mean`, `Salinity` and `lChlorophyll`.

c) Make a Wilks Test to check if any predictor has an influence on any target variable.

d) Have a look at the individual regression summaries.

Does `lChlorophyll` have an equally significant effect on all three responses?

2. We test the differences between the unforged and forged banknotes: `CODE`: 0 unforged banknotes, 1 foraged banknotes

- `LENGTH`, `LEFT`, `RIGHT`, `BOTTOM`, `TOP`, `DIAGONAL`: different measures of the banknotes.

You may find the data in http://stat.ethz.ch/Teaching/Datasets/WBL/banknot.dat.

a) Load the data and have a first look at it.

b) We would like to know whether the variable `LENGTH` differs significantly for the unforged and the foraged banknotes. Make a *t*-Test for the variable `LENGTH`.

Make the same test for all other variables - we would like to know whether the other variables can seperate the unforged from the forged banknotes significantly. Write a small program that returns the *p*-value of the test. Use `apply()`.

R-Hint:

```r
t.test(LENGTH ~ CODE, ...)
f.ttest <- function(y) {
  r.ttest <- t.test(y ~ bn[,"CODE"])
  r.ttest$...
} apply(..., f.ttest)
```

c) Install the package `ICSNP`. Look at the help file of the function `HotellingsT2`.

d) Use Hotellings's T-test for unpaired groups in order to decide, whether the unforged banknotes differ from the forged ones.

3. Model-Based Clustering: In this excercise we would like to apply the Model-Based-Clustering to the dataset `banknot.dat`.

a) Make a clustering with `Mclust()` from the package `mclust` using the maximum likelihood method.

What number of clusters and what model do you propose?

R-hint: Look in the helpfile for more information.
library(mclust)
d.banknot <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/banknot.dat")
ml.banknot<-Mclust(d.banknot[,,-1])
plot(... , what="BIC")
ml.cluster<-Mclust(... , modelNames="..." , G= ...)
where modelNames is the choice of method for the structure of the covariance matrix (EEE to VVV) and G the number of clusters.

b) Make a table with the misclassification of the model based method with respect to CODE. Keep in mind: CODE=0 are the genuine banknotes an CODE=1 the forged ones.
Make a pairs plot of the variables by chosing the color of the dots according to CODE (col=) and the shape according to their model based method (pch=). Comment?

c) Carry out the PAM-algorithm for the same number of clusters as above and the euclidean metric.
Make a table with the “misclassification” of the model based method compared to the PAM algorithm.
Make a pairs plot of the variables by chosing the color of the dots according to the PAM cluster (col=) and the shape according to their model based method (pch=). Comment?

4. Agglomerative Clustering: In the dataframe empl2.dat (of employment.dat), the rounded rate (in percent) of employment in 9 different sectors in 10 chosen european states are given. The data are from 1979. The observation (states) are:

a) First look at the data using the scatterplot. Can you find clusters by eye?
R-hint: With the following R-Code, you can label the points in the scatterplotmatrix with the acronym of their countries (you can find further examples in the helpfile of pairs()).
t.url <- "http://stat.ethz.ch/Teaching/Datasets/WBL/empl2.dat"
empl <- read.table(t.url, header=T)
labempl <- rownames(empl)
pairs(empl, panel=function(x,y) text(x,y, labels=labempl, xpd=T))

b) Calculate the euclidean distances between the states. Which two states are first combined into a cluster?
R-hint: With as.matrix() you can convert the result of daisy() (Package cluster) into a distance matrix. Useful R-commands include sort() and unique().

c) Carry out a hierarchical cluster analysis by hand using the “Single Linkage”-method.

d) Carry out the previous cluster analysis using the function agnes(). Verify your result of c) by comparing the first five steps.
R-hint: sing.empl <- agnes(empl, method="single")
# 2 Plots (Bannerplot und Dendrogramm)
par(mfrow=c(1,2))
plot(sing.empl)

e) Carry out the cluster analysis with the same distances but with the methods average and complete. Compare the dendrograms of all three methods (including the Single Linkage Method).
R-hint: If you set the arguments which.plot in plot to 2, only the dendrogram is plotted.

f) Group the states into k clusters. Choose for instance k = 3 and k = 4. Compare the different methods. Also plot an MDS-plot and mark the observed groups of states with colors (for one k and one method).
R-hint:
# Classification in four groups using average Linkage
r.4cl <- cutree(aver.empl, k=4)
split(labempl, r.4cl)
# MDS-Plot:
r.mds <- cmdscale(daisy(empl))
plot(r.mds, type = "n", main = "'Average clustering, MDS coordinates")
text(r.mds, labempl, col = 1 + r.4cl)
5. Load the data \texttt{banknot.dat} - we will need this data again in this exercise. We use the partitions algorithm PAM to separate the forged from the unforged banknotes and compare it with the K-means method. In order to be able to compare your results with the sample solution use \texttt{set.seed(10)}. Take the whole dataset \texttt{banknot.dat} and choose the variables \texttt{CODE}, \texttt{BOTTOM} and \texttt{DIAGONAL}. You will need the package \texttt{cluster} and \texttt{MASS}. Reading in the data:

\begin{verbatim}
> d.bank.org <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/banknot.dat")
> d.bank <- d.bank.org[,c("CODE","BOTTOM","DIAGONAL")]
\end{verbatim}

\textbf{a) K-means-algorithm:}
- Apply the K-means algorithm (without \texttt{CODE}), to obtain 2 optimal clusters.
- Make a table of the misclassifications with respect to the "true" classifications (\texttt{CODE}).
- Make a silhouette-plot. Comments?
  \textbf{R-Hint:} For the silhouette-plot we need the distance matrix and proceed as follows:
  \begin{verbatim}
  ?kmeans ... <- dist( ... ,method="euclidean")
  ... <- silhouette(...$cluster,t.bank)
  plot(...)
  \end{verbatim}

\textbf{b) PAM-algorithm:}
- Find the optimal partition in 2 clusters for this dataset (without \texttt{CODE}) using the PAM method for the euclidean metric.
- Represent the two variables by producing different point shapes for the different clusters and different colors for the \texttt{CODE}. Which observations were classified wrong?
- Make a table of the misclassifications, as in a).
- Make a silhouette-plot. Comments?
  \textbf{R-Hint:} The silhouette-plot is easier for \texttt{pam}: \texttt{plot(\ldots , which=2)}. What does \texttt{plot(\ldots , which=1)} yield?

\textbf{c) Compare the two results. Make a table with the differences with respect to the two clusterings.}
\textbf{Note:} The clustervalues of a point do not have to be the same - this means a point can have value 1 with the K-means-method and value 2 with the PAM-method (\texttt{...$cluster} and \texttt{...$clustering})

\textbf{d) We would like to show by means of simple simulation, that the K-means algorithm finds local minima. Use the K-means algorithm for 3 clusters 100 times and change the size of the 3 clusters (number of points in the cluster) in each run. You can use the following code:}

\begin{verbatim}
set.seed(10)
v.einer1 <- rep(1,dim(d.bank)[1])
v.einer2 <- rep(1,100)
t.kmeans <- NULL
for (i in 1:100){
kmean.bank.cluster <- kmeans(d.bank[,-1],centers=3)$cluster
t.kmeans <- cbind(t.kmeans,sort(aggregate(v.einer1,by=list(kmean.bank.cluster),
FUN=sum)[,2]))
}
trans.kmeans <- t(t.kmeans)
m.kmeans <- aggregate(v.einer2,by=list(trans.kmeans[,1],trans.kmeans[,2],
  trans.kmeans[,3]),FUN=sum)

\end{verbatim}

Adapt the code for the PAM-algorithm. Comment? Do the same with 2,4,5 etc. clusters.

\textbf{Preliminary discussion: 27.05.13.}

\textbf{Deadline:} No hand-in.