Series 3

1. **Covariance matrix and principal component analysis:**
We work again with the data `woodRous.dat` from Series 2. In this exercise we consider only the explanatory variables (i.e. all variables except `SpGew`). We will compute the covariance matrix and run a principal component analysis on the data. Load the data via

```
> d.wood <- read.table("http://stat.ethz.ch/Teaching/Datasets/cas-das/woodRous.dat",
                       header = TRUE)
> d.wood <- d.wood[, 1:5]
```

a) Load the data into R and transform it according to Tukey's first aid transformations as we did in Series 2.

b) Estimate the covariance matrix classically and robustly. What differences do you see between the two covariance matrices?
R-Hint: For the robust estimation, use the MCD method implemented in the R functions `covMcd()` or `CovRobust(..., control = "mcd")` from the package `rrcov`.

c) Calculate the classical and the robust Mahalanobis distances for the observations and identify the outliers. Take a look at the Chi-squared QQ plot of the Mahalanobis distances against the quantiles of the Chi-squared distribution as in Figure 4.1.d. in the script. Also compare the classical and the robust Mahalanobis distances by plotting the distances against the indices of the observations as in Figure 4.1.k. in the script.
R-Hint:

- Use the function `mahalanobis(..., center = ..., cov = ...)` to calculate the squared Mahalanobis distance in R.
- In the robust case, the function `CovRobust()` returns an `S4`-class object (see `?CovRobust` for the explanation of the `S4`-class object), so it is not possible to access a part of the returned object of the `CovRobust`-function with the `$`-sign. Use `@` instead (i.e. for example `fit@cov` instead of `fit$cov`).

Remark: You can already see the outliers in the pairs plot. Label the observations in the plot in order to identify the outliers.

d) Next, we want to run a principal component analysis on the data. As the variables have different units, we first standardize all the variables. Recall that running a PCA on standardized variables (using the covariance matrix) yields the same results as running a PCA using the correlation matrix. Perform a PCA on the data:

- using the classical approach on the standardized data,
R-Hint: Use `princomp(..., cor = ...)`.
- using the classical approach on the robustly standardized data, i.e. the data is first standardized using robust location and scale estimates (e.g. median and MAD),
R-Hint: Use `scale(x = ..., center = ..., scale = apply(..., ..., mad))`. Note that the function `scale()` requires a data matrix in the argument `x`.
- using the robust approach which is based on a robustly estimated correlation matrix.
R-Hint: Use the function `PcaCov(..., scale = ...)` from the package `rrcov`.

Compare the three solutions with respect to the number of principle components needed to explain at least 80% of the variance in the data. Take a look at the PCA output and at the scree-plot.

e) Examine the loadings of the three approaches. What can you observe?
R-Hint: Use `loadings()` or `$loadings` to access the loadings from the `princomp()`-object. Use `@loadings` combined with the command `stats:::print.loadings()` for the robust method to get a more standard output.

f) For all three approaches, make a pairs-plot of the principal component scores of the data. Where can you see the outliers?
R-Hint: Use `pairs(predict(...))`.

2. Linear Discriminant analysis:
In this exercise we apply both a classical and robust linear discriminant analysis on an artificial dataset `http://stat.ethz.ch/Teaching/Datasets/cas-das/rob-disk.dat`. In this dataset you find the variable `Klasse` which is the class of an observation. The variables `x1` and `x2` are continuous variables measuring different properties of the objects.

a) Read in the data and make a plot of `x2` vs. `x1`. Label and color the points according to the class identifier.

b) Perform a linear discriminant analysis with ...

- ... the classical method: `lda(...)`,
- ... the robustly estimated covariance matrix W: `lda(..., method = "mve")`,
- ... where both the matrix W and the location of the centers are estimated robustly by the MCD-estimator: `rlda(...)`.

Make a plot of the data with respect to the first and second discriminant variables for each method as in Figure 4.3.c. in the script. Can you observe any differences in the plots?
R-Hint: For the plots, source the file `rg2-fkt.R` which can be found at `http://stat.ethz.ch/Teaching/Datasets/WBL` (see R Code below). It contains the function `p.ldv()` which plots the data with respect to the first and second discriminant variables such that the scales for both axes are equal:

```
> source("http://stat.ethz.ch/Teaching/Datasets/cas-das/rg2-fkt-v2.R")
> library(MASS)
> t.momlda <- lda(... ~ ..., ...) # classical
> t.mvelda <- lda(... ~ ..., ..., method = ...) # robust W
> t.rlda <- rlda(..., grouping =...) # robust W and B
> p.ldv(..., group = ...) # plots
> title(...)
```

c) Draw the “classification-boundaries” of the three different `lda` approaches into (separate) scatterplots of the original variables (as in Figure 4.3.g. in the script). What can you observe?
R-Hint: Use the function `p.predplot()`, which is available in the file `rg2-fkt-v2.R`, to make the plots. The function is an improved version from the book of Venables und Ripley (*Modern Applied Statistics with S-Plus*). If you want to know how the function works just write “`p.predplot`” into the console.

Exercise hour: Monday, June 17, morning.