Solution to Series 3

1. Read in data:

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

a) J. W. Tukey suggests the root transformation for counting data and the arc-sine squareroot transformation for portions.
R-Code:

```
> d.wood[, 1:2] <- sqrt(d.wood[, 1:2])         # Squareroot
> d.wood[, 3:5] <- asin(sqrt(d.wood[, 3:5]))   # Arc-sine square root
```

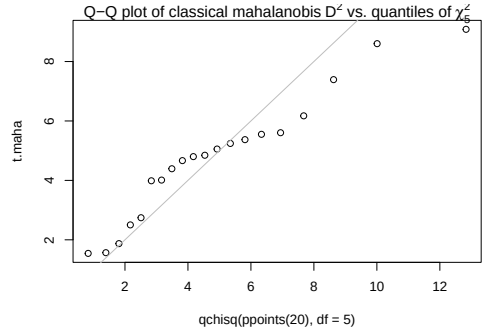
b)

```
> # classical
> t.cov <- cov(d.wood)
> t.cov
          nFrFas  nSoFas    AntFr  FrLicht SoLicht
nFrFas   3.8023 -0.9044  0.07846  0.06043 -0.1319
nSoFas  -0.9044  9.0633 -0.04414 -0.11292  0.1801
AntFr    0.0785 -0.0441  0.00424  0.00160 -0.0037
FrLicht  0.0604 -0.1129  0.00160  0.00398 -0.0024
SoLicht -0.1319  0.1801 -0.00370 -0.00240  0.0106
> # robust
> t.rcov <- CovRobust(d.wood, control = "mcd")
> t.rcov
Call:
CovMcd(x = x, control = obj)
-> Method:  Fast MCD(alpha=0.5 ==> h=13); nsamp = 500; (n,k)mini = (300,5)

Robust Estimate of Location:
 nFrFas   nSoFas    AntFr  FrLicht  SoLicht
 24.184   34.896    0.817    0.824    1.240

Robust Estimate of Covariance:
          nFrFas     nSoFas     AntFr      FrLicht    SoLicht
nFrFas    4.323214   5.492122   0.064930  -0.011235  -0.050052
nSoFas    5.492122   9.606410   0.179994  -0.008575   0.006519
AntFr     0.064930   0.179994   0.006745  -0.000876   0.000542
FrLicht  -0.011235  -0.008575  -0.000876   0.002873   0.002988
SoLicht  -0.050052   0.006519   0.000542   0.002988   0.007097
```

The entries in the two different covariance-matrices often have different signs.
Note: If the method for the robust estimation of the covariance matrix is not specified with the argument `control=...` in the function `CovRobust`, the function will select the robust estimator itself according to the size of the dataset.
If there are less than 1000 observations and less than 10 variables or less than 5000 observations and less than 5 variables, the "Stahel-Donoho" estimator will be used. Otherwise, if there are less than 50000 observations either the bisquare S-estimator (for less than 10 variables) or the Rocke type S-estimator (for 10 to 20 variables) will be used. In both cases the R iteration starts at the initial MVE estimate. Finally, if there are more than 50000 observations and/or more than 20 variables the Orthogonalized Quadrant Correlation estimator (CovOgk with the corresponding parameters) is used. If we hadn't specified the method, the "Stahel-Donoho" estimator would be used in this case. However for this particular example, it turns out to give very unstable results.

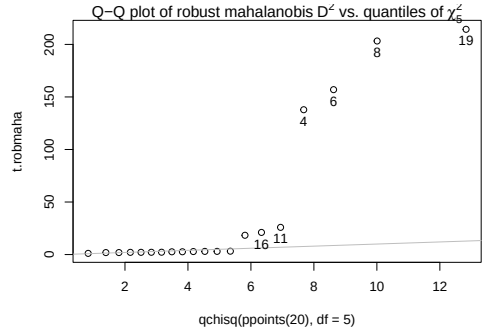
c) We can try to find the outliers by looking at the QQ-plot or at the pairs-plot.

```
> ## classical
> t.maha <- mahalanobis(d.wood, center = colMeans(d.wood), cov = t.cov)
> qqplot(qchisq(ppoints(20), df = 5), t.maha,
         main = expression("Q-Q plot of classical mahalanobis" * ~D^2 *
               " vs. quantiles of" * ~ chi[5]^2 ))
> abline(0, 1, col = "gray")
```



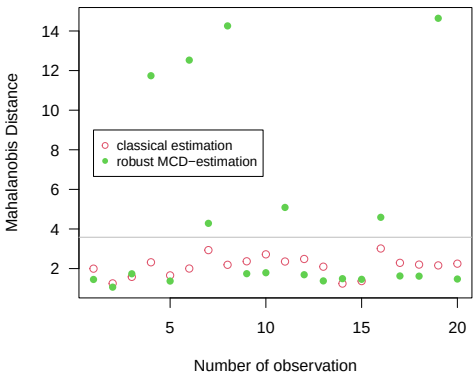
In the classical non-robust case, it is hard to say which of the points are outliers.

```
> ## robust
> t.robmaha <- mahalanobis(d.wood, center = t.rcov@center, cov = t.rcov@cov)
> qqplot(qchisq(ppoints(20), df = 5), t.robmaha,
        main = expression("Q-Q plot of robust mahalanobis" * ~D^2 *
               " vs. quantiles of" * ~ chi[5]^2))
> abline(0, 1, col = "gray")
> t.tmp <- qqplot(qchisq(ppoints(20), df = 5), t.robmaha, plot = FALSE)
> t.lab <- order(t.robmaha)
> identify(t.tmp$x, t.tmp$y, labels = t.lab)
```



For the robust case, the outliers are easy to detect. We identify the observations 19, 8, 6 and 4 (maybe also 11 and 16) as outliers.

```
> t.maha <- mahalanobis(d.wood, center = colMeans(d.wood), cov = t.cov)
> plot(c(1, nrow(d.wood)), range(sqrt(c(t.maha, t.robmaha))), type = "n", las = 1,
       xlab = "Number of observation", ylab = "Mahalanobis Distance")
> points(1:nrow(d.wood), sqrt(t.maha),  col = 2)
> points(1:nrow(d.wood), sqrt(t.robmaha), col = 3, pch = 16)
> abline(h = sqrt(qchisq(0.975, df = 5)), col = "gray")
> legend(x = 1, y = 9, col = c(2, 3), pch = c(1, 16), cex = 0.8,
        legend=c("classical estimation", "robust MCD-estimation"))
```



The observations lying above the grey line (97.5%-χ^2_5 quantile) can be considered as being outliers. In the classical case, all the points are lying below the line, we thus find no outliers.

Note: Rousseeuw and Leroy have modified the observations 4, 6, 8 and 19. We find these outliers with the robust method (see last two plots). (Observations 11 and 16 were not modified but maybe can still be said to be outliers.)

Remark: The outliers can easily be spotted in the pairs plot:

```
> pairs(d.wood, panel = function(x, y) text(x, y, labels = 1:20))
```

d)

```
> Wood.pca <- princomp(d.wood, cor = TRUE)
> summary(Wood.pca)
```

```
Importance of components:
                       Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
Standard deviation      1.693  1.013  0.777 0.6444 0.2989
Proportion of Variance  0.573  0.205  0.121 0.0831 0.0179
Cumulative Proportion   0.573  0.778  0.899 0.9821 1.0000
```

```
> WoodT1 <- scale(data.matrix(d.wood), center = apply(d.wood, 2, median),
                  scale = apply(d.wood, 2, mad))
> Wood.pca2 <- princomp(WoodT1, cor = FALSE)
> summary(Wood.pca2)
```

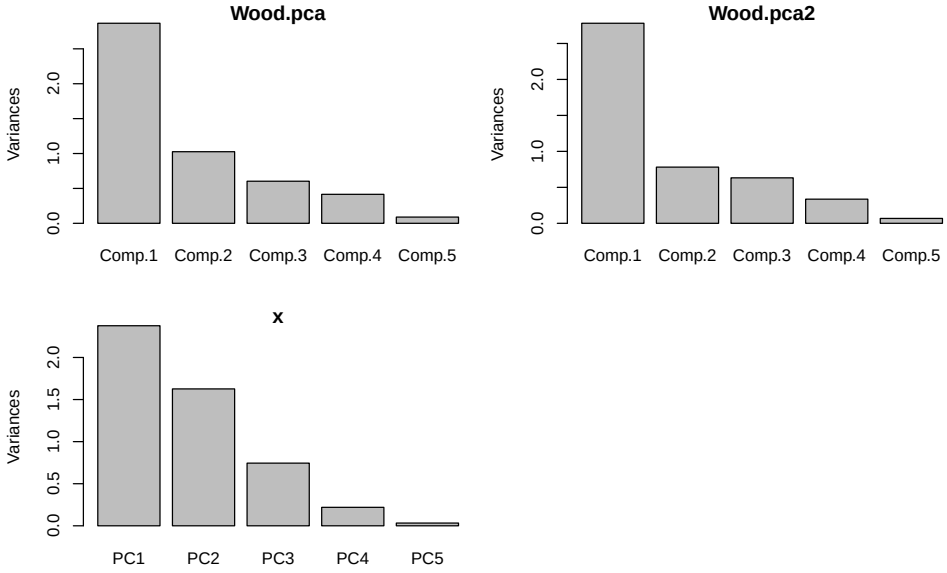
```
Importance of components:
                       Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
Standard deviation      1.668  0.884  0.795  0.579 0.2609
Proportion of Variance  0.605  0.170  0.137  0.073 0.0148
Cumulative Proportion   0.605  0.775  0.912  0.985 1.0000
```

```
> Wood.pca3 <- PcaCov(d.wood, scale = TRUE)  # from package rrcov
> summary(Wood.pca3)
```

```
Call:
PcaCov(x = d.wood, scale = TRUE)
Importance of components:
                         PC1   PC2   PC3    PC4    PC5
Standard deviation     1.542 1.275 0.863 0.4684 0.1803
Proportion of Variance 0.475 0.325 0.149 0.0439 0.0065
Cumulative Proportion  0.475 0.801 0.950 0.9935 1.0000
```

For the first and second method, the first three principle components are needed in order to explain at least 80% of the variance in the data. Note that we are close to the limit of 80% (between 77 and 79%) of the variance explained with the first two principle components. For the third method, the first two principle components are needed.

```
> ## Scree-Plot
> par(mfrow = c(2, 2))
> screeplot(Wood.pca)
> screeplot(Wood.pca2)
> screeplot(Wood.pca3)
```



We find a first elbow after one component for method ond and two.
Note: Since a shift of the data doesn't change the results of PCA, we would have obtained the same results by subtracting the mean instead of the median when standardizing the data in the second approach (similar to taking the standard deviation instead of the MAD).

e) *> loadings(Wood.pca)*

```
Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
nFrFas   0.468  0.455  0.168  0.540  0.504
nSoFas  -0.386  0.696  0.264  0.153 -0.523
AntFr    0.439  0.428         -0.788
FrLicht  0.436 -0.351  0.720         -0.404
SoLicht -0.500         0.618 -0.243  0.555

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
SS loadings       1.0    1.0    1.0    1.0    1.0
Proportion Var    0.2    0.2    0.2    0.2    0.2
Cumulative Var    0.2    0.4    0.6    0.8    1.0
```

> loadings(Wood.pca2)

```
Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
nFrFas   0.344  0.241         0.658  0.617
nSoFas  -0.301  0.614         0.453 -0.567
AntFr    0.490  0.588  0.366 -0.530
FrLicht  0.353 -0.467  0.659  0.259 -0.394
SoLicht -0.653         0.645 -0.119  0.377

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
SS loadings       1.0    1.0    1.0    1.0    1.0
Proportion Var    0.2    0.2    0.2    0.2    0.2
Cumulative Var    0.2    0.4    0.6    0.8    1.0
```

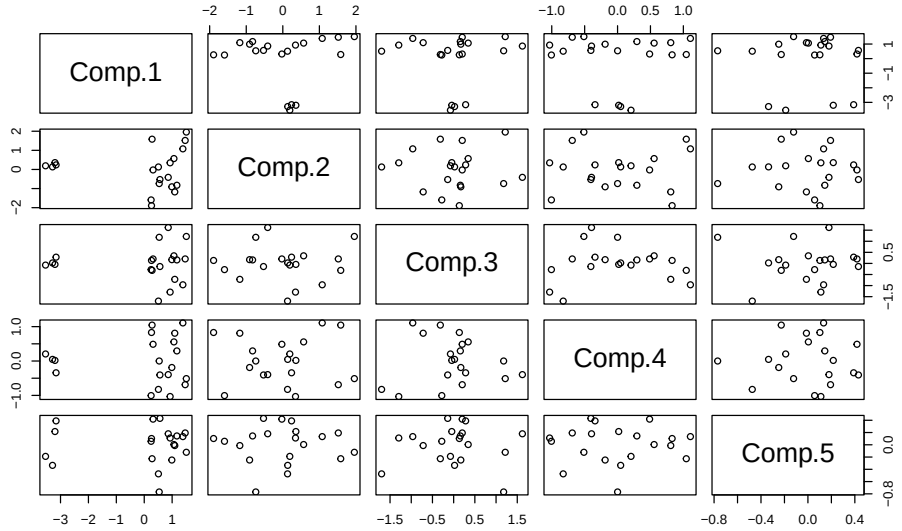
> stats:::print.loadings(Wood.pca3@loadings)

```
Loadings:
        PC1    PC2    PC3    PC4    PC5
nFrFas   0.562         -0.555  0.149 -0.593
nSoFas   0.603  0.254          0.203  0.723
AntFr    0.486  0.195  0.661 -0.469 -0.262
FrLicht -0.223  0.643 -0.419 -0.596
SoLicht -0.185  0.694  0.269  0.601 -0.225
```

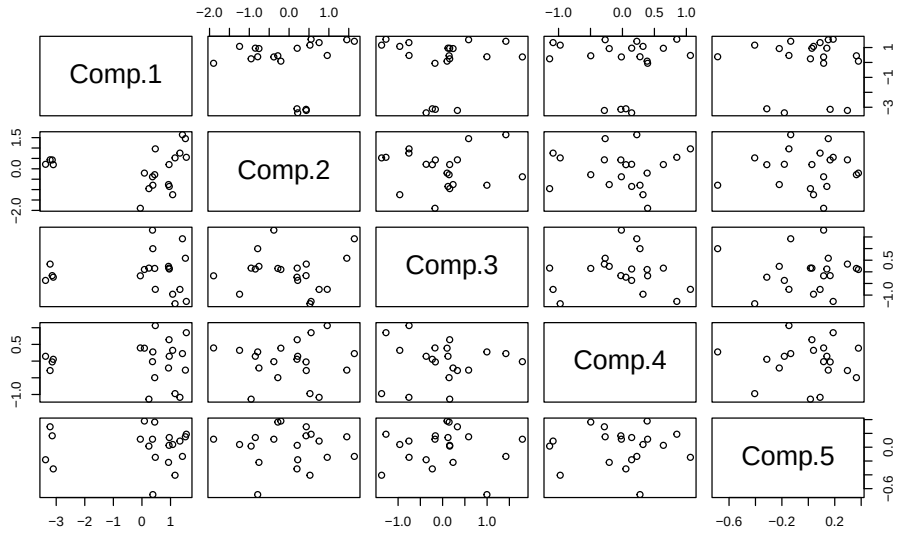
```
                PC1 PC2 PC3 PC4 PC5
SS loadings     1.0 1.0 1.0 1.0 1.0
Proportion Var  0.2 0.2 0.2 0.2 0.2
Cumulative Var  0.2 0.4 0.6 0.8 1.0
```

The three methods yield very different loadings, which will result in very different interpretations.

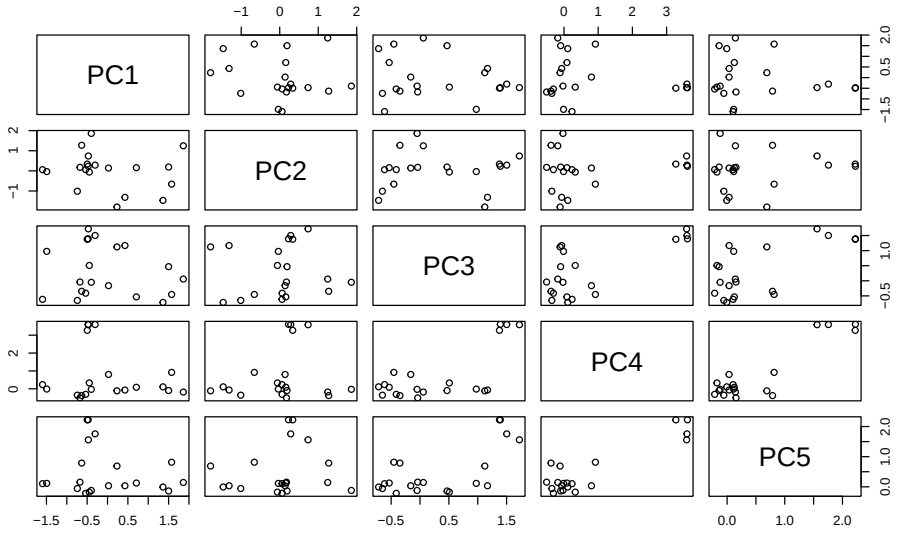
f) *> pairs(predict(Wood.pca))*



> pairs(predict(Wood.pca2))

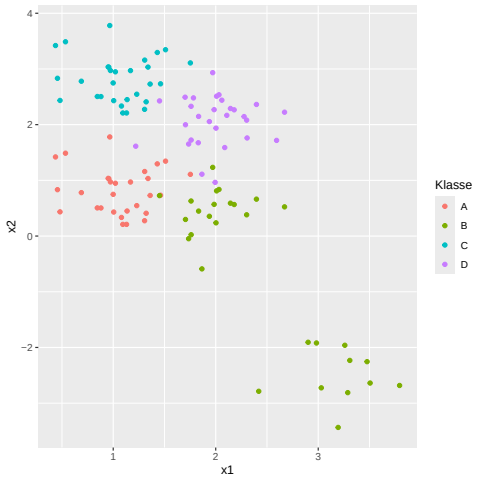


> pairs(predict(Wood.pca3))



When we use the classical method (PCA on the classically or robustly standardized data), we find the outliers in the first component because the outliers account for a large part of the variability.
When we use the robust method (PCA with robustly estimated correlation matrix), we can see the outliers in the fourth component. In general, the outliers can occur in any principal component when we use a robust estimation (but they do not have to), since they have little influence on the estimation of the covariance matrix. To find the outliers, we have to take a look at the Mahalanobis distances.

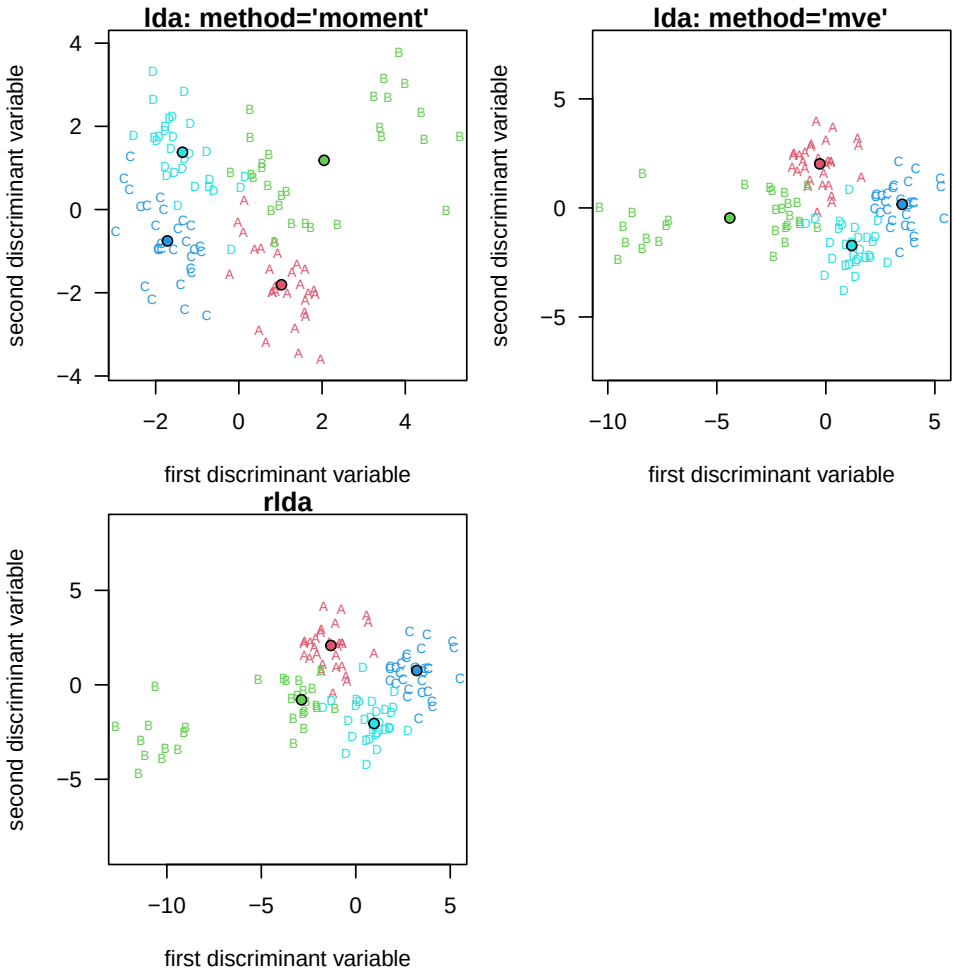
2. **a)**
```
> d.disk <- read.table("http://stat.ethz.ch/Teaching/Datasets/cas-das/rob-disk.dat",
                       header = TRUE)
> str(d.disk)
'data.frame':        116 obs. of  3 variables:
 $ Klasse: chr  "A" "A" "A" "A" ...
 $ x1    : num  1.169 0.456 1.005 1.23 1.13 ...
 $ x2    : num  0.97 0.832 0.431 0.546 0.21 ...
> d.disk$Klasse <- as.factor(d.disk$Klass)
> # make plot
> library(ggplot2)
> ggplot(data=d.disk, mapping=aes(x=x1, y=x2, col=Klasse))+
      geom_point()
```



The data consists of four classes that lie "close" together. The classes A, C and D are pretty homogenous, whereas B separates into two subgroups.

b)
```
> library(MASS)
> source("http://stat.ethz.ch/Teaching/Datasets/cas-das/rg2-fkt-v2.R")
```

```
> par(mfrow = c(2, 2))
> ## classical
> t.momlda <- lda(Klasse ~ ., data = d.disk)
> p.ldv(t.momlda, data = d.disk[, 2:3], group = d.disk$Klasse)
> title("lda: method='moment'")
> ## robust (MVE estimator)
> t.mvelda <- lda(Klasse ~ ., data = d.disk, method = "mve")
> p.ldv(t.mvelda, data = d.disk[, 2:3], group = d.disk$Klasse)
> title("lda: method='mve'")
> ## robust (MCD-estimator and robust location estimation)
> t.rlda <- rlda(x = d.disk[, 2:3], grouping = d.disk$Klasse)
> p.ldv(t.rlda, data = d.disk[, 2:3], group = d.disk$Klasse)
> title("rlda")
```

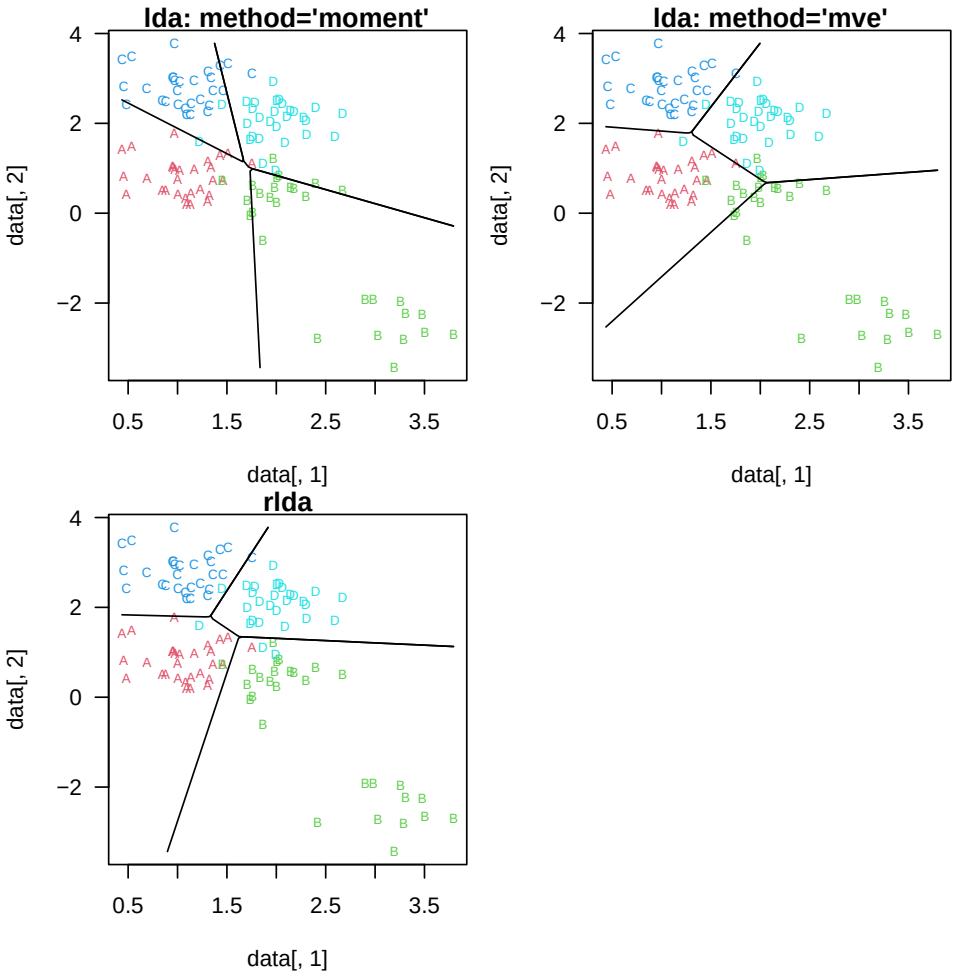


The plot-function `p.ldv()` internally calls the function `eqscplot()` (from the package `MASS`), that produces a scatterplot where the scales for both axis are equal, which means that the differences in cm between 0 and 2 are the same for the x-axis as well as the y-axis. The four point clouds should thus appear as circles if the assumptions are met.
The point clouds "A", "C" and "D" look like an ellipsoid in the classical case, while they look more like circles when using the robust methods. When using the MCD-estimation of the covariance matrix together with the robust location estimator, the centres of the classes seem most reasonable.

c)

```
> ### Plot
> par(mfrow = c(2, 2))
> ##  classical
> p.predplot(t.momlda, data = d.disk[, 2:3], group = d.disk$Klasse)
> title("lda: method='moment'")
> ##  robust using MVE-estimator
> p.predplot(t.mvelda, data = d.disk[, 2:3], group = d.disk$Klasse)
> title("lda: method='mve'")
```

```
> ##  robust using MCD-estimator and robust location estimation
> p.predplot(t.rlda, data = d.disk[, 2:3], group = d.disk$Klasse)
> title("rlda")
```



The "`rlda`-method" seems to best separate the four groups judging by the misclassification rates, i.e. the number of observations which are misclassified (the observations which are not lying in the correct area) compared to the total number of observations.
The first two lda approaches - the classical one and the one with a robustly estimated W - have difficulties separating the group B from the groups A and D. The robust method `rlda` (with robustly estimated locations and W) yields a more satisfactory solution for separating the group B from the groups A and D.