ETH Zurich

Applied Multivariate Statistics **Exam ID AMS01**

1. 2.000
2. 02.392
3. 05.390
4. 2.000

1. **Problem**
   In a hospital, a group of 30 patients was examined using a chemical analysis of their blood. Each analysis contains 8 variables, each measuring a different chemical compound.

   The goal is to produce a visualization of the chemical contents of each patient and compare the patients in terms of their chemical analysis results.

   Load the data: The file *readViz*.338801.*csv* contains the data in comma-separated form. Each row corresponds to a patient, each column to a chemical compound. Patient names are in the first column and should be used in the visualization. The first row contains the names of the columns.

   Scaling: What is the scale of each variable? Would you scale, if we want to compare the patients in relative terms?

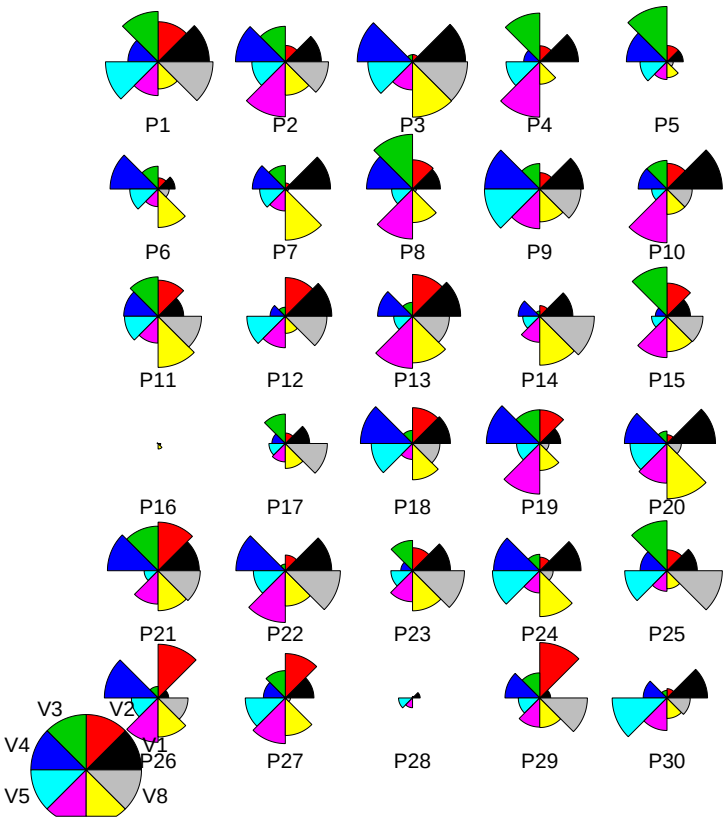
   Visualization and Interpretation: Make a suitable visualization of the data set. There is a certain type of disease where all blood chemicals are reduced. Some of these patients might be in this sample. Can you detect them? Make a legend.

   Solution

   Sick people: 2

   Code:

```
> fn <- "/sfs/u/staff/kalisch/teaching/pruefungAMS/exams/data/readViz.csv"
> dd <- read.csv(file = paste(pth, fname, sep = ""), row.names = 1, header = TRUE)
> apply(dd, 2, sd)

          V1           V2           V3           V4           V5           V6
3.143100e-06 3.069961e-05 3.233904e-06 3.648974e-01 3.145859e-03 3.574516e+00
          V7           V8
3.322664e-06 3.353106e-03

> stars(dd, draw.segments = TRUE, key.loc = c(1,1))
```



2. **Problem**
   Use LDA for supervised learning on simulated cancer data set. y=0 - no cancer, y = 1 - cancer. There are 8 explanatory variables.

   The data is saved in the data frame `dat` in the R-data file (*supLearn*2.156264.*rda*). Load the data.

   Fit LDA to the data. Then, create a new data frame containing the old observations 1,150 and 450. What tumor class is predicted by your LDA model? How does this compare to the true labels?

   Use leave-one-out cross-validation to estimate the accuracy of LDA on new observations. What error rate would you expect?

   Use LDA to visualize the separation of the four groups. Plot the value of the FIRST discriminant vs. a random ordering of the observation number (sample(1 :nrow(dat)). Use a different color for each symbol. Is there a horizontal line, that separates the groups well?

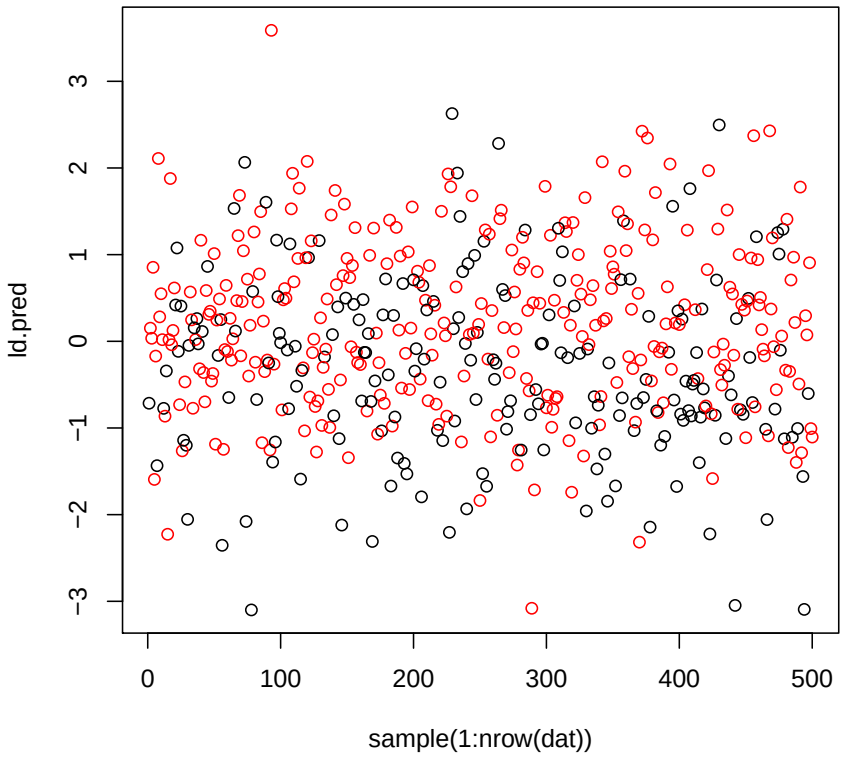
   What is the maximum number of linear discriminants we can compute in this problem with two groups and ten variables?

   Solution
   Code:

```
> library(MASS)
> lda.fit <- lda(y ~ ., data = dat)
> lda.fit

Call:
lda(y ~ ., data = dat)
```

```
Prior probabilities of groups:
  0   1
0.4 0.6

Group means:
           X1          X2          X3         X4          X5         X6
0 -0.06371504 -0.16841011 -0.16843495 0.34530709  0.01122434 0.06820072
1  0.02206664  0.01179378  0.01403858 0.02090264 -0.05322646 0.02345557
            X7          X8
0  0.376914649  0.56922381
1 -0.007036322 -0.01241636

Coefficients of linear discriminants:
           LD1
X1  0.25709292
X2  0.33894624
X3  0.57967064
X4 -0.59481228
X5 -0.03958142
X6  0.21109163
X7 -0.08580920
X8 -0.19760791

> nd <- dat[c(1,150,450),]
> predict(lda.fit, newdata = nd)

$class
[1] 1 1 1
Levels: 0 1

$posterior
            0         1
1   0.4442108 0.5557892
150 0.3167381 0.6832619
450 0.3090418 0.6909582

$x
           LD1
1   -0.3991784
150  0.6426236
450  0.7110929

> dat$y[c(1,150,450)]

[1] 0 1 0
Levels: 0 1

> lda.CV <- lda(y ~ ., data = dat, CV = TRUE)
> tab <- table(lda.CV$class, dat$y)
> errRate <- 1 - (sum(diag(tab)) / nrow(dat))
> errRate

[1] 0.392
```

```
> ld.pred <- predict(lda.fit)$x
> myCol <- as.numeric(dat$y)
> plot(sample(1:nrow(dat)), ld.pred, col = myCol)
```



3. **Problem**
   We need to re-organize the seating for 16 persons in a very busy office. Some people need to visit each other in person quite a lot ("low contact distance"), others less so ("high contact distance"). To be efficient, everybody should sit closer to people with high contact activity than with low contact activity.

   The management made a poll and asked everybody to rate the "contact distance" with every other person in the office (0: contact all the time; 10: no contact at all). The result was made symmetric afterwards.

   Suggest a seating plan that fulfills these requirements as good as possible.

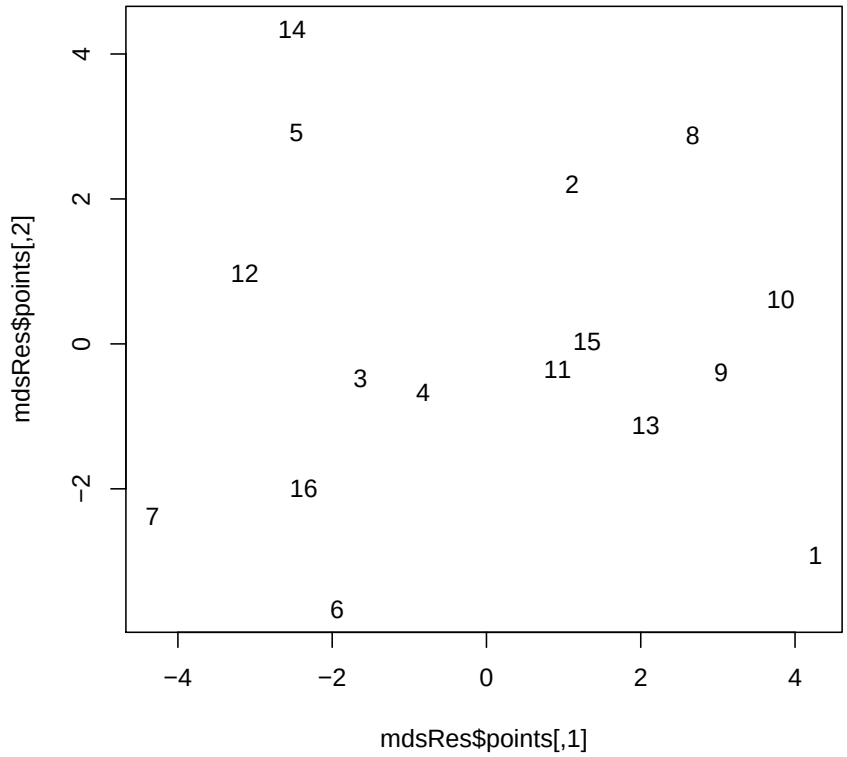
   The data is saved in matrix `dat` in the R-data file (*msc*1.922675.*rda*) and contains the result of the poll.

   Solution
   Code:

```
> library(MASS)
> mdsRes <- isoMDS(dat)

initial  value 6.530452
iter   5 value 5.396519
iter   5 value 5.391199
```

```
iter   5 value 5.389166
final  value 5.389166
converged
```

```
> plot(mdsRes$points, type = "n")
> text(mdsRes$points, labels = rownames(mdsRes$points))
> stress <- round(mdsRes$stress, 2)
```



STRESS: 5.39.

4. **Problem**
We have 482 soil samples. A chemical analysis involving 25 chemicals was done for each sample. The chemicals are all measured in the same units and have similar scales.

The data is saved in the data frame `dat` in the R-data file (*dis*1.660625.*rda*).

For further processing, we would like to reduce the dimensionality as much as possible. But we still want to explain most (e.g. 80%) of the variability in the data.

Find a suitable transformation for this goal.

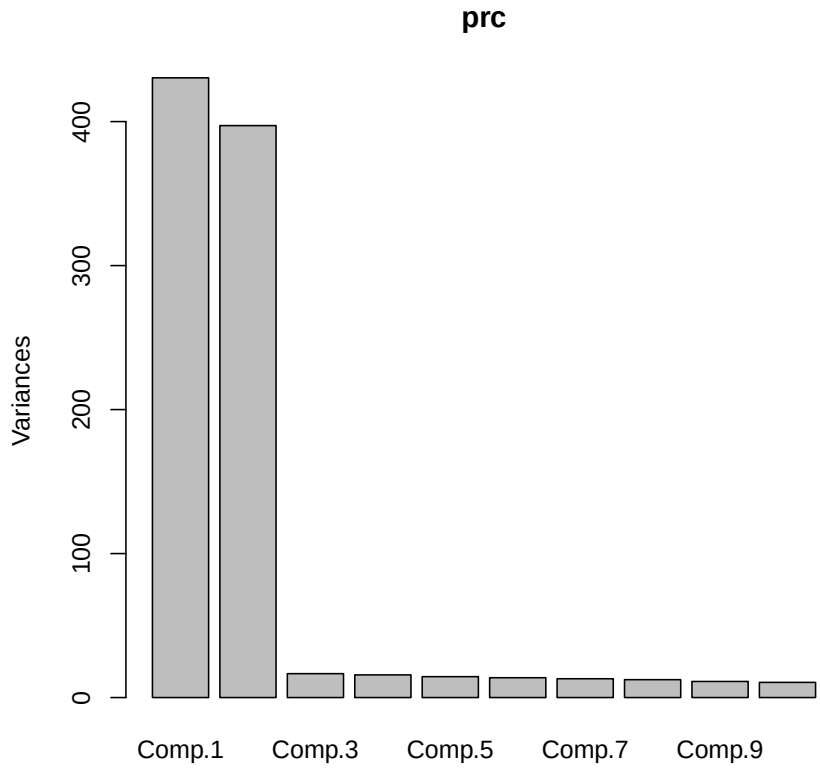
What are the coordinates of the first sample in the new coordinate system?

How can the value of the first new coordinate be computed from the old data set?

Solution
Number of PCs should be around: 2

Code:

```
> prc <- princomp(dat)
> summary(prc, loadings = FALSE)

Importance of components:
                          Comp.1     Comp.2     Comp.3     Comp.4     Comp.5
Standard deviation     20.7463437 19.9300872 4.07442633 3.96901198 3.80950944
Proportion of Variance  0.4203562  0.3879295 0.01621315 0.01538506 0.01417335
Cumulative Proportion   0.4203562  0.8082857 0.82449884 0.83988390 0.85405725
                          Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
Standard deviation     3.71180913 3.61510319 3.52480871 3.34350724 3.24606241
Proportion of Variance 0.01345568 0.01276368 0.01213404 0.01091789 0.01029078
Cumulative Proportion  0.86751293 0.88027660 0.89241064 0.90332854 0.91361931
                         Comp.11     Comp.12     Comp.13     Comp.14
Standard deviation     3.20770731 3.079963409 3.016037345 2.897377003
Proportion of Variance 0.01004902 0.009264574 0.008883984 0.008198688
Cumulative Proportion  0.92366834 0.932932911 0.941816895 0.950015584
                          Comp.15     Comp.16     Comp.17     Comp.18
Standard deviation     2.801135172 2.725232307 2.657329904 2.368004906
Proportion of Variance 0.007663065 0.007253396 0.006896446 0.005476455
Cumulative Proportion  0.957678648 0.964932045 0.971828490 0.977304946
                          Comp.19     Comp.20     Comp.21     Comp.22
Standard deviation     2.256553520 2.190080083 1.997120517 1.695174359
Proportion of Variance 0.004973082 0.004684404 0.003895318 0.002806487
Cumulative Proportion  0.982278028 0.986962432 0.990857750 0.993664237
                          Comp.23    Comp.24     Comp.25
Standard deviation     1.564431878 1.53340076 1.299439577
Proportion of Variance 0.002390274 0.00229639 0.001649098
Cumulative Proportion  0.996054511 0.99835090 1.000000000

> prc$scores[1,]

     Comp.1      Comp.2      Comp.3      Comp.4      Comp.5      Comp.6
25.00710592 -3.26839182 -2.81077379  1.50696594  5.19551195 -2.37002119
     Comp.7      Comp.8      Comp.9     Comp.10     Comp.11     Comp.12
 4.83366845  0.78130213  4.13893285  1.60865959  4.56357983 -1.10206245
    Comp.13     Comp.14     Comp.15     Comp.16     Comp.17     Comp.18
 0.85134284 -2.02886322 -0.80349138 -0.31788344  0.46000289 -2.11951630
    Comp.19     Comp.20     Comp.21     Comp.22     Comp.23     Comp.24
 0.40810308  1.11137993 -1.01233378 -0.61593040  1.53179855 -0.01436187
    Comp.25
-1.76644238

> prc$loadings[,1]

 [1]  0.09480420 -0.14324471 -0.26529907  0.03439226  0.34266897 -0.40964691
 [7] -0.23534504 -0.15762490  0.13616095 -0.23552333  0.16041545  0.12973014
[13]  0.16887762 -0.21207324 -0.04652328 -0.09699898 -0.06348763  0.30367828
[19] -0.19687992  0.04350867  0.19234664  0.16153247  0.35114738 -0.08936989
[25] -0.02306239

> plot(prc)
```


prc
Variances
0
100
200
300
400
Comp.1
Comp.3
Comp.5
Comp.7
Comp.9