Tutorial 10: Non-metric Multidimensional Scaling

Exercise 1: Andrews Curves

We'll illustrate Andrews curves using the swiss bank note data. Firstly, install the package `andrews`

```
> install.packages("andrews")
> library("andrews")
```

After installing the package `andrews`, The following illustrates the Andrews curve for the Swiss Bank data set (found in *swisslab.dat*) in the course home page. Import the data set.

```
> www =
"https://www.mimuw.edu.pl/~noble/courses/MultivariateStatistics/data/
swisslab.dat"
> swisslab = read.csv(www,sep=";",header= T)
```

Let us consider banknotes 91 - 110. That gives 10 real and 10 counterfeit banknotes.

```
> fram <- swisslab[91:110,]
> fram

> reversedata <- fram[,c("DIAGONAL","UPPERMARGIN","LOWERMARGIN",
    "RIGHTHEIGHT","LEFTHEIGHT","LENGTH")]
> reversedata

> andrews(reversedata,clr=5,ymax=3)
```

Exercise 2: Mahalanobis Distance

The computation may be implemented in R as follows: the data is found on the course home page under `skulls.dat` and is loaded as follows:

```
> www2 =
"https://www.mimuw.edu.pl/~noble/courses/MultivariateStatistics/data/
skulls.dat"
> skulls = read.table(www2,header=T)
```

To get the means for each variable by category, try

```
> x2 <- by(skulls[,-5],skulls[,5],colMeans)
> y<-simplify2array(x2)
```

and for the covariance matrices by category, try

```
> S <- by(skulls[,-5],skulls[,5],cov)
> Sarray<-simplify2array(S)
```

To obtain the pooled covariance,

```
> Spooled <-
29*(Sarray[,,1]+Sarray[,,2]+Sarray[,,3]+Sarray[,,4]+Sarray[,,5])/145
> Spooled
            MB          BH          BL        NH
MB 21.11080460  0.03678161  0.07908046  2.008966
BH  0.03678161 23.48459770  5.20000000  2.845057
BL  0.07908046  5.20000000 24.17908046  1.133333
NH  2.00896552  2.84505747  1.13333333 10.152644
```

To obtain the Mahalanobis distance,

```
> mahalanobis(y[,1],y[,2],Spooled,inverted=FALSE)
[1] 0.09103424
```

Exercise: find (or write) an R-script that gives the whole array of Mahalanobis distances presented in a single matrix.

Exercise 3: Mantel Randomisation Test

To perform a Mantel test, the `ade4` package may be used:

```
> install.packages("ade4")
> library("ade4")
```

We'll try it on the `ozone.csv` data set.

```
> www3 =
"https://www.mimuw.edu.pl/~noble/courses/MultivariateStatistics/data/
ozone.csv"
> ozone = read.csv(www3,header= T)
> head(ozone)
  Station   Av8top      Lat       Lon
1      60 7.225806 34.13583 -117.9236
2      69 5.899194 34.17611 -118.3153
3      72 4.052885 33.82361 -118.1875
4      74 7.181452 34.19944 -118.5347
5      75 6.076613 34.06694 -117.7514
6      84 3.157258 33.92917 -118.2097
```

This contains ozone measurements from thirty-two locations in the Los Angeles area aggregated over one month. The dataset includes the station number (Station), the latitude and longitude of the station (Lat and Lon), and the average of the highest eight hour daily averages (Av8top). We want to test whether the differences in ozone measurements are smaller for stations that are closer together than for stations that are far apart.

To run a Mantel test, generate two distance matrices, one containing spatial distances and one containing distances between measured outcomes at the given points. In the spatial distance matrix, entries for pairs of points that are close together are lower than for pairs of points that are far apart. In the measured outcome matrix, entries for pairs of locations with similar outcomes are lower than for pairs of points with dissimilar outcomes. This may be done using the `dist` function. The Mantel test function requires objects of this 'distance' class.

```
> station.dists <- dist(cbind(ozone$Lon, ozone$Lat))
> ozone.dists <- dist(ozone$Av8top)
> as.matrix(station.dists)[1:5, 1:5]
> as.matrix(ozone.dists)[1:5, 1:5]
```

These are the two matrices which the function will test for a correlation. The test consists of calculating the correlation of the entries in the matrices, then permuting the matrices and calculating the same test statistic under each permutation and comparing the original test statistic to the distribution of test statistics from the permutations to generate a p-value. The number of permutations defines the precision with which the p-value can be calculated. The function to perform the Mantel test is `mantel.rtest` and the required arguments are the two distance matrices. The number of permutations can also be specified by the user; the default value is 99.

```
> mantel.rtest(station.dists, ozone.dists, nrepet = 9999)
```

Based on these results, can we reject the null hypothesis that these two matrices, spatial distance and ozone distance, are unrelated with $\alpha = .05$?

Exercise 4: Classical MDS

We'll use the data set `swiss` which comes with R. The data set is 'Swiss Fertility and Socioeconomic Indicators (1888) Data'. This gives 47 observations on 6 variables. Type `?swiss` to get a description. A classical MDS can be carried out as follows:

```
data("swiss")
head(swiss)

# Load required packages
library(magrittr)
```

```
library(dplyr)
library(ggpubr)
# Cmpute MDS
mds <- swiss %>%
  dist() %>%
  cmdscale() %>%
  as_tibble()
colnames(mds) <- c("Dim.1", "Dim.2")
# Plot MDS
ggscatter(mds, x = "Dim.1", y = "Dim.2",
          label = rownames(swiss),
          size = 1,
          repel = TRUE)
```

and we can create 3 groups using K-means and colour them as follows:

```
# K-means clustering
clust <- kmeans(mds, 3)$cluster %>%
  as.factor()
mds <- mds %>%
  mutate(groups = clust)
# Plot and color by groups
ggscatter(mds, x = "Dim.1", y = "Dim.2",
          label = rownames(swiss),
          color = "groups",
          palette = "jco",
          size = 1,
          ellipse = TRUE,
          ellipse.type = "convex",
          repel = TRUE)
```

Exercise 5: Non-Metric MDS

We can perform a non-metric MDS as follows:

```
#Non-metric MDS
library(magrittr)
library(dplyr)
library(ggpubr)
```

```
# Cmpute MDS using Kruskal
library(MASS)
mds <- swiss %>%
  dist() %>%
  isoMDS() %>%
  .$points %>%
  as_tibble()
colnames(mds) <- c("Dim.1", "Dim.2")
# Plot MDS
ggscatter(mds, x = "Dim.1", y = "Dim.2",
          label = rownames(swiss),
          size = 1,
          repel = TRUE)

#Sammon's Non-linear Mapping
# Cmpute MDS
library(MASS)
mds <- swiss %>%
  dist() %>%
  sammon() %>%
  .$points %>%
  as_tibble()
colnames(mds) <- c("Dim.1", "Dim.2")
# Plot MDS
ggscatter(mds, x = "Dim.1", y = "Dim.2",
          label = rownames(swiss),
          size = 1,
          repel = TRUE)
```

Exercise 5

The file `morse.rda` in the course data directory contains morse code data, giving the percentages of times that a signal corresponding to the row label was identified as being the same as the signal corresponding to the column label. A row of this table shows the confusion rate for that particular morse-code signal when presented *before* each of the column signals, whereas a column of the table shows the confusion rate for that particular signal when presented *after* each of the row signals. This table of confusion rates is not symmetric and the diagonal elements are not each 100%. Now, every square matrix M can be decomposed uniquely as the sum $M = A + B$ where $A = \frac{1}{2}(M + M')$ and $B = \frac{1}{2}(M - M')$.

Ignore the part of the morse-code data provided by B and carry out a non-metric scaling only of the symmetric part A. Decide how many dimensions you think are appropriate for representing this data.

Exercise 6

The file `BritishTowns.rda` in the course directory contains a proximity matrix for the distances between 48 towns in Great Britain. Carry out a classical scaling of these pairwise distances and construct a map of Great Britain.