**Lab 4 - Data visualization and clustering**

Systems modelling and data analysis
2016/2017

# 1 Preparing the data

1. Run RStudio

2. Set your Working Directory using the setwd() command.

3. Download, extract and then load the data from the file krakow-kurdwanow.zip. The data comes from the monthly reports from 2015 from the Kraków-Kurdwanów station (source: http://monitoring.krakow.pios.gov.pl/).

```
         download.file("http://home.agh.edu.pl/~mmd/_media/dydaktyka/
as-is/krakow-kurdwanow.zip", "krakow-kurdwanow.zip")
unzip("krakow-kurdwanow.zip")
data <- dget("./krakow-kurdwanow")
```

4. Check the list of available devices.

```
?Devices
```

5. Check the active device. By default, the data will be displayed on the screen.

```
dev.cur()
```

# 2 Data visualization

The most commonly used data visualization packages in R are: base, lattice and ggplot2. In the following exercises we will use only the base package. To learn about the rest of the packages you can type: ??lattice and ??ggplot2.

1. View a summary of the loaded NO2 measurement data, and then present them in a box-by-month graph.

```
summary(data$NO2)
boxplot(data$NO2 ~ format(data$date, "%m"), xlab = "months", ylab =
"NO2")
```

2. Display the NO2 histogram and draw a vertical line representing the median.

```
hist(data$NO2, col="blue")
abline(v = median(data$NO2), lwd = 5, col="red")
```

3. Display 4 histograms with quarterly data. In particular, notice the order in which the histograms are displayed - where histogram number 2 is displayed.

   ```
   par(mfrow = c(2,2))
   hist(data[quarters(data$date) == "Q1",]$NO2)
   hist(data[quarters(data$date) == "Q2",]$NO2)
   hist(data[quarters(data$date) == "Q3",]$NO2)
   hist(data[quarters(data$date) == "Q4",]$NO2)
   ```

4. Display a graph showing PM10 and PM2.5 pairs.

   ```
   par(mfrow=c(1,1))
   with(data, plot(PM10, PM25))
   ```

5. Add a title to the chart: PM10 PM2.5

   ```
   title("PM10 ~ PM2.5")
   ```

6. Mark in red all the points that represent the December measurements.

   ```
   with(data[format(data$date, "%m") == "12",], points(PM10, PM25,
   col="red"))
   ```

7. Add a legend in the upper left corner of the graph.

   ```
   legend("topleft", pch = 1, col = "red", legend = "December")
   ```

8. Add linear regression graph.

   ```
   model <- lm(data$PM25 ~ data$PM10)
   abline(model)
   ```

9. Create a graph showing the number of measurements in each month. Save the graph directly in a PDF file without displaying it on the screen.

   ```
   pdf("plot.pdf")
   barplot(table(format(data$date, "%m")))
   dev.off()
   ```

10. Create a graph showing the number of measurements in each month. Save the graph directly to a PNG file without displaying it on the screen.

    ```
    png("plot.png")
    barplot(table(format(data$date, "%m")))
    dev.off()
    ```

11. Create a graph showing the number of measurements in each month. Display it on the screen and save it as a PDF.

    ```
    barplot(table(format(data$date, "%m")))
    dev.copy(pdf, "plot2.pdf")
    dev.off()
    ```

# 3 Hierarchical clustering

1. Write a function to count the Euclidean distance between 2 points.

```
distance <- function(x1,y1,x2,y2) {
      sqrt((x2-x1)^2 + (y2-y1)^2)
}
```

2. Calculate the Euclidean distance of 2 points: p1 (24,13) and p2 (64,53).

```
distance(24,13,64,53)
```

3. Define data as a 10-row and 2-column data frame (in column 1, the x-coordinate, in column 2, y-coordinate), which will cluster 2 points around 5 points p1 (2.2), p2 (8.8), p3 (2,8), p4 (8,2), p5 (5,5).

```
x <- c(rnorm(2)+2,rnorm(2)+8,rnorm(2)+2,rnorm(2)+8,rnorm(2)+5)
y <- c(rnorm(2)+2,rnorm(2)+8,rnorm(2)+8,rnorm(2)+2,rnorm(2)+5)
points <- data.frame(cbind(x,y))
```

4. Present data in graphical form.

```
plot(y ~ x, points)
```

5. Create a dataframe that will contain pairs of points.

```
df <- data.frame(nrow=0,ncol=4)
for(i in 1:10) {
     for(j in 1:10) {
                if(i>j) {
                        df[10*(i-1)+j,1] <- points[i,1]
                        df[10*(i-1)+j,2] <- points[i,2]
                        df[10*(i-1)+j,3] <- points[j,1]
                        df[10*(i-1)+j,4] <- points[j,2]
                  }
       }
 }
 df <- df[complete.cases(df),]
```

6. Calculate the distance between points.

```
df$dist <- sqrt((df[,3]-df[,1])^2 + (df[,4]-df[,2])^2)
```

7. Sort the dataframe by distance.

```
df <- df[order(df$dist),]
```

8. Mark on the chart the first 3 pairs of points with the shortest distance.

```
for(i in 1:3) {
       points(df[i,1], df[i,2], col=i, pch=4)
       points(df[i,3], df[i,4], col=i, pch=4)
}
```

9. Using the above method, you can make full hierarchical clustering. Use the dist and hclust functions. Draw a dendogram.

   ```
   distance <- dist(points)
   cluster <- hclust(distance)
   plot(cluster)
   ```

10. Mark two groups of points on the dendogram.

    ```
    rect.hclust(cluster, k=2, border="red")
    ```

11. Divide the data into 2 groups and draw a chart where they will be marked with a different color.

    ```
    groups <- cutree(cluster, k=2)
    plot(y ~ x, points, col=groups)
    ```

## 4 Exercise

Draw 4 graphs on the screen. Arrange them as follows:

| | |
|---|---|
| Draw a box plot - PM2.5 divided into days of the week. | Draw a PM2.5 histogram. |
| Draw a graph showing the PM2.5 change over time (x-axis - time, y-axis - PM2.5). Mark red all points for which PM2.5 $>$ 100 | Make hierarchical clustering. Divide data into 4 groups. Draw a graph showing pairs of data PM10 (x-axis) and PM2.5 (y-axis) marked with different color depending on the group. |