Wizualizacja danych i klastrowanie

Przygotowanie danych

1. Uruchom RStudio.
2. Ustaw swój Working Directory używając polecenia setwd().

```
setwd("F:/inazwisko")
```

3. Wykorzystaj dane, które zostały przygotowane w podobny sposób jak na poprzednim laboratorium. Dane pochodzą z raportów miesięcznych z 2015 roku ze stacji Kraków-Kurdwanów (źródło: http://monitoring.krakow.pios.gov.pl/). Pobierz, rozpakuj, a następnie wczytaj dane z pliku.

```
download.file("http://home.agh.edu.pl/~mmd/_media/dydaktyka/adp/dane-pomiarowe-dla-stacji-krakow-kurdwanow.zip", "dane-pomiarowe-dla-stacji-krakow-kurdwanow.zip")
```

```
unzip("dane-pomiarowe-dla-stacji-krakow-kurdwanow.zip")
```

```
data <- dget("./dane-pomiarowe-dla-stacji-krakow-kurdwanow")
```

4. Sprawdź listę dostępnych urządzeń.

```
?Devices
```

5. Sprawdź aktywne urządzenie. Domyślnie dane będą wyświetlane na ekranie.

```
dev.cur()
```

Wizualizacja danych

Do wizualizacja danych najczęściej wykorzystywane są pakiety: base, lattice oraz ggplot2. W poniższych ćwiczeniach wykorzystamy wyłącznie pakiet base. W celu zapoznania się z pozostałymi pakietami możesz wpisać: ??lattice oraz ??ggplot2.

1. Wyświetl podsumowanie wczytanych danych pomiarowych NO2, a następnie przedstaw je w formie wykresu pudełkowego z podziałem na miesiące.

```
summary(data$NO2)
```

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

2. Wyświetl histogram wartości NO2 oraz narysuj pionową linię oznaczającą medianę.

```
hist(data$NO2, col="blue")
```

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

3. Wyświetl 4 histogramy z podziałem na kwartały. W szczególności zwróć uwagę w jakiej kolejności wyświetlane są histogramy – w którym miejscu wyświetlany jest 2 histogram.

```
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. Wyświetl wykres, na którym zostaną przedstawione pary wartości PM10 oraz PM25.

```
par(mfrow=c(1,1))

with(data, plot(PM10, PM25))
```

5. Dodaj tytuł do wykresu: PM10 ~ PM25

```
title("PM10 ~ PM25")
```

6. Oznacz kolorem czerwonym wszystkie punkty, które przedstawiają pomiary z grudnia.

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

7. Dodaj legendę w lewym górnym rogu wykresu.

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

8. Dodaj wykres regresji liniowej.

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

abline(model)
```

9. Stwórz wykres przedstawiający liczbę pomiarów w poszczególnych miesiącach. Zapisz wykres bezpośrednio w pliku PDF bez wyświetlania go na ekranie.

```
pdf("plot.pdf")

barplot(table(format(data$date, "%m")))

dev.off()
```

10. Stwórz wykres przedstawiający liczbę pomiarów w poszczególnych miesiącach. Zapisz wykres bezpośrednio w pliku PNG bez wyświetlania go na ekranie.

```
png("plot.png")

barplot(table(format(data$date, "%m")))

dev.off()
```

11. Stwórz wykres przedstawiający liczbę pomiarów w poszczególnych miesiącach. Wyświetl go na ekranie, a następnie zapisz jako PDF.

```
barplot(table(format(data$date, "%m")))

dev.copy(pdf, "plot2.pdf")

dev.off()
```

Klastrowanie hierarchiczne

1. Napisz funkcję do liczenia odległości euklidesowej pomiędzy 2 punktami.

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

2. Policz odległość euklidesową 2 punktów: p1(24,13) oraz p2(64,53).

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

3. Zdefiniuj dane w formie data frame o 10 wierszach i 2 kolumnach (w 1 kolumnie współrzędna x, w 2 kolumnie współrzędna y), które będą skupiały po 2 punkty wokół 5 punktów 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. Przedstaw dane w formie graficznej.

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

5. Stwórz dataframe, w którym znajdą się pary punktów.

```
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. Policz odległość między punktami

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

7. Posortuj dataframe wg odległości

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

8. Zaznacz na wykresie 3 pierwsze pary punkty o najmniejszej odległości.

```
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. Wykorzystując powyższy sposób można dokonać pełnego klastrowania hierarchicznego. Wykorzystaj funkcje dist oraz hclust. Narysuj dendogram.

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

10. Zaznacz na dendogranie 2 grupy punktów

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

11. Podziel dane na 2 grupy i narysuj wykres, na którym zostaną one oznaczone innym kolorem.

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