

Regresja liniowa

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")
```

Regresja liniowa

1. Stwórz wektory `y` oraz `x`. Przypisz odpowiednio dane z kolumn: `data$PM25` oraz `data$PM10`.

```
y <- data$PM25
```

```
x <- data$PM10
```

2. Usuń brakujące dane z wektorów. Jeśli brakuje danych w wektorze `y` usuń także odpowiadające dane z wektora `x`. Podobnie jeśli w wektorze `x` brakuje danych usuń także odpowiadające dane z wektora `y`.

```
good <- complete.cases(x, y)
```

```
y <- y[good]
```

```
x <- x[good]
```

3. Oblicz korelację – sprawdź jak bardzo zmienne zależą od siebie liniowo.

```
n <- length(x)
```

```
l <- (n*sum(x*y) - sum(x) * sum(y))
```

```
m <- sqrt((n*sum(x^2) - sum(x)^2) * (n*sum(y^2) - sum(y)^2))
```

```
l/m
```

4. Wykorzystaj funkcję `cor()` do obliczenia korelacji

```
cor(x, y)
```

5. Dokonaj centralizacji zmiennej losowej.

```
ymean <- mean(y)
```

```
xmean <- mean(x)
```

```
y <- y - ymean
```

```
x <- x - xmean
```

6. Sprawdź, czy centralizacja została wykonana prawidłowo.

```
mean(y)
```

```
mean(x)
```

7. Stwórz funkcję, która liczyć będzie sumę kwadratów „pionowych” odległości punktów od prostej $y = ax$ dla podanego argumentu a . Funkcja powinna posiadać następujące argumenty:
- Współrzędne y -owe danych (wektor)
 - Współrzędne x -owe danych (wektor)
 - Parametr a prostej $y = ax$, dla której liczona jest suma kwadratów.

```
sum_of_the_squered <- function(y,x,a) {  
  sum <- 0  
  for(i in seq_along(y)) {  
    sum <- sum + (y[i] - (a*x[i]))^2  
  }  
  sum  
}
```

8. Stwórz funkcję, która wybierze parametr a z podanego wektora parametrów a (a_vector), dla którego suma kwadratów jest najmniejsza. Funkcja powinna posiadać następujące argumenty:
- Współrzędne y -owe danych (wektor)
 - Współrzędne x -owe danych (wektor)
 - Wektor parametrów a (a_vector).

```
find_a <- function(y,x,a_vector) {  
  min_sum <- Inf  
  min_a <- NA  
  for(a in a_vector) {  
    sum <- sum_of_the_squered(y,x,a)  
  
    if(sum < min_sum) {  
      min_sum <- sum  
      min_a <- a  
    }  
  }  
}
```

```
}  
min_a  
}
```

9. Wykorzystaj stworzoną funkcję, aby znaleźć parametr a . W tym celu zwiększaj dokładność szukanego parametru do 10 miejsc po przecinku. Wyświetl znaleziony parametr

```
a <- 0  
for(i in 0:10) {  
  a <- find_a(y,x,seq(a-10^(-i+1), a+10^(-i+1), 10^(-i)))  
}  
a
```

10. Użyj funkcji `lm()` do znalezienia parametru a , a następnie wyświetl znaleziony parametr.

```
model <- lm(y ~ x)  
model$coefficients["x"]
```

11. Policz i wyświetl różnice pomiędzy parametrem a znalezionym za pomocą stworzonej funkcji, a funkcji `lm()`.

```
model$coefficients["x"] - a
```

12. Narysuj wykres przedstawiający dane oraz regresję liniową wykonaną za pomocą stworzonych funkcji oraz funkcji `lm()`.

```
plot(y~x)  
abline(0,a, col="red", lwd=7)  
abline(model, col="blue", lwd=3)
```

13. Narysuj wykres przedstawiający dane oraz regresję liniową wykonaną za pomocą stworzonych funkcji oraz funkcji `lm()` dla niezcentralizowanych danych.

```
plot(data$PM25~data$PM10)  
abline(ymean-a*xmean,a, col="red", lwd=5)  
model <- lm(data$PM25 ~ data$PM10)  
abline(model, col="white")
```