PPPD - Lab. 06

Napisz funkcję `regresja()`, która na wejściu przyjmuje dwie listy liczbowe `x` i `y` tej samej długości n, wyznacza wartości współczynników α, β w tzw. modelu prostej regresji liniowej i zwraca je w postaci 2-elementowej listy $[\alpha, \beta]$, gdzie:

$$\beta = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

$$\alpha = \bar{y} - \beta\,\bar{x},$$

oraz $\bar{x}, \bar{y}$ oznaczają, odpowiednio, średnie arytmetyczne wartości z `x`, `y`.

Można pokazać (Analiza matematyczna II...), że współczynniki te określają prostą $y = \alpha + \beta x$, która minimalizuje sumę kwadratów błędów:

$$E(\alpha, \beta; \mathtt{x}, \mathtt{y}) = \sum_{i=1}^{n} (\alpha + \beta x_i - y_i)^2 .$$

Zaimplementuj powyższy wzór w postaci funkcji `E(alpha, beta, x, y)`.

Oczywiście nie do każdych danych ma sens dopasowywanie prostej. Z tego powodu w praktyce analizy danych często wyznacza się współczynnik korelacji liniowej r Pearsona, dany wzorem:

$$r(\mathtt{x}, \mathtt{y}) = \frac{1}{n-1} \sum_{i=1}^{n} \frac{x_i - \bar{x}}{s_x}\,\frac{y_i - \bar{y}}{s_y},$$

gdzie s_x, s_y to odchylenia standardowe wartości z list wejściowych. Zauważ, że wartości współczynnika bliskie co do modułu 1 oznaczają silną liniową zależność między zmiennymi. Zaimplementuj powyższy wzór w postaci funkcji `r(x, y)`.

---

1. Dla różnych `x` (np. wygenerowanych losowo) i `y` (np. różnych funkcji x z dodanym losowym błędem wyznacz wartości współczynników prostej regresji i narysuj ją.
2. Zweryfikuj empirycznie, że prosta regresji w istocie minimalizuje miarę E, porównując wartości błędów dla różnych (np. losowo wybranych) prostych.
3. Wyznacz współczynnik korelacji liniowej r Pearsona dla tak wygenerowanych danych. Co oznacza współczynnik korelacji równy 1 i -1? Czy wartości r bliskie 0 oznaczają brak jakiegokolwiek związku między zmiennymi?

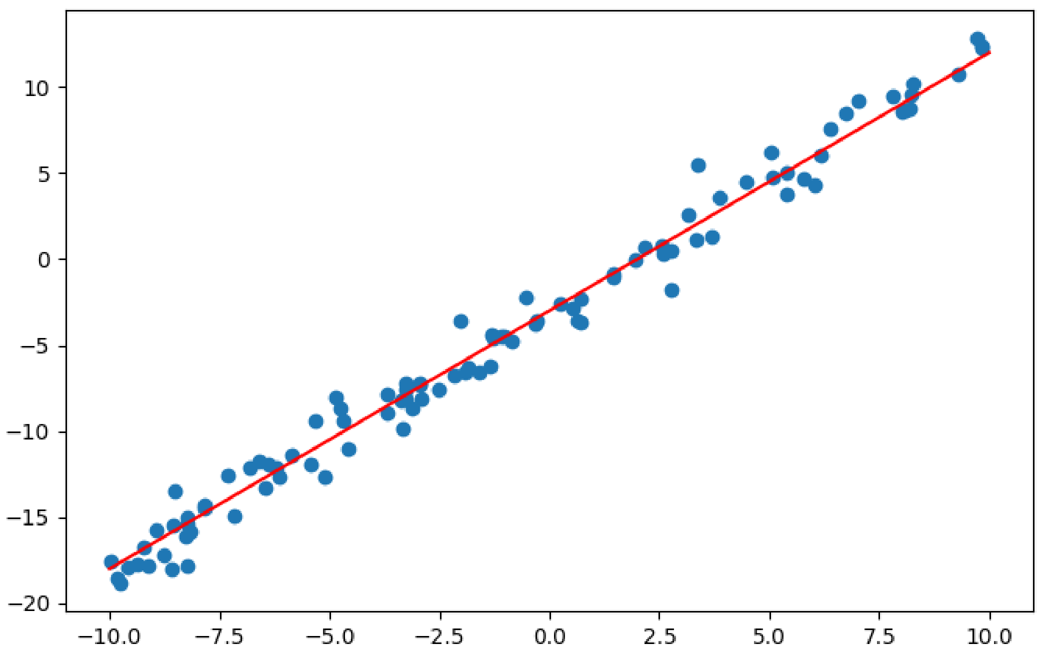
Kody pomocnicze (generowanie przykładowych zbiorów danych, rysowanie):

```
import random
random.seed(123)

alpha0 = -3
beta0  = 1.5
n      = 100

# poniżej używamy tzw. wyrażenia listotwórczego (ang. list comprehension)
x = [ random.uniform(-10, 10) for i in range(n) ]
y = [ alpha0+beta0*x[i]+random.normalvariate(0, 1) for i in range(n) ]
# czyli y=alpha0+beta0*x+szum z rozkładu normalnego N(0,1)


import matplotlib.pyplot as plt
# wykres rozproszenia:
plt.scatter(x, y)
# rysowanie odcinka [xmin, xmax], [ymin, ymax]:
plt.plot([-10, 10], [alpha0+beta0*(-10), alpha0+beta0*10], color="red")
# zapis do pliku PNG:
plt.savefig("zadanie_6_01.png")
```



Rysunek 1: Ilustracja do zadania