Національний технічний університет України «КПІ»

Кафедра автоматизованих систем обробки інформації і управління

Лабораторна робота №1

з дисципліни «[Аналіз даних в інформаційних управляючих системах](https://do.ipo.kpi.ua/course/view.php?id=5470)»

на тему: «[МЕТОДИ ОБРОБКИ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ.](https://do.ipo.kpi.ua/mod/resource/view.php?id=149430)»

Виконав:

студент групи ІС-23

Шимків М.В.

Викладач:

Гавриленко О.В

Київ 2024

**Мета роботи**: сформувати у студентів уявлення про підхід до

вирішення завдання про середньоквадратичне наближення

функції, заданої таблично; прищепити знання про методи

апроксимації елементарними функціями; виробити навички

роботи в середовищі МАТLAB.

**Завдання до роботи**

1. Ознайомитися з конспектом лекцій та рекомендованою

літературою за темою роботи, а також з додатком до роботи, що

містить опис програмного забезпечення для побудови різних

видів регресій.

2. Використовуючи дані таблиці і застосовуючи стандартні

заміни змінних, знайти рівняння наступних видів регресій:

лінійної, гіперболічної, степеневої, показникової, логарифмічної.

3. Порівняти якість отриманих наближень шляхом порівняння їх

відхилень.

4. Побудувати графіки одержаних залежностей і табличних

значень аргументів і функції.

Варіант 25

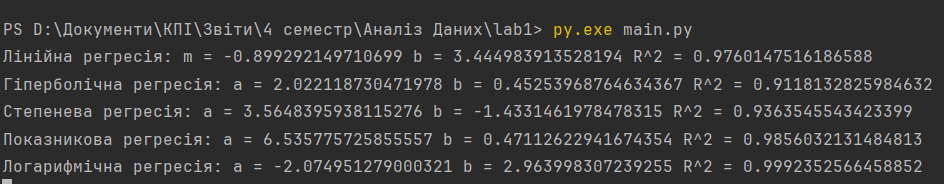
**Набір даних:**

****

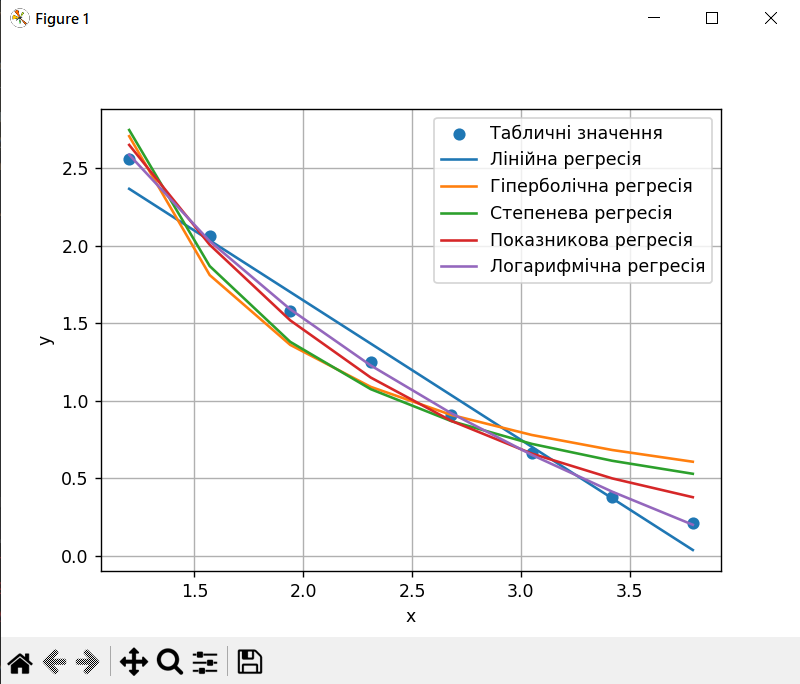
Запуск програми відбувається так:



Рівняння наступних видів регресій(лінійної, гіперболічної, степеневої, показникової, логарифмічної), а також порівняння якості отриманих наближень шляхом порівняння їх відхилень:



Графіки одержаних залежностей і табличних значень аргументів і функції:



**Код програми:**

import numpy as np  
import matplotlib.pyplot as plt  
from scipy.optimize import curve\_fit  
  
# Дані таблиці  
x\_data = np.array([1.20, 1.57, 1.94, 2.31, 2.68, 3.05, 3.42, 3.79])  
y\_data = np.array([2.56, 2.06, 1.58, 1.25, 0.91, 0.66, 0.38, 0.21])  
  
# Лінійна регресія (y = mx + b)  
def linear\_regression(x, m, b):  
 return m \* x + b  
  
# Гіперболічна регресія (y = a / (x - b))  
def hyperbolic\_regression(x, a, b):  
 return a / (x - b)  
  
# Степенева регресія (y = ax^b)  
def power\_regression(x, a, b):  
 return a \* np.power(x, b)  
  
# Показникова регресія (y = ab^x)  
def exponential\_regression(x, a, b):  
 return a \* np.power(b, x)  
  
# Логарифмічна регресія (y = a \* ln(x) + b)  
def logarithmic\_regression(x, a, b):  
 return a \* np.log(x) + b  
  
# Функція для порівняння регресій  
def compare\_regression(x\_data, y\_data, func):  
 popt, pcov = curve\_fit(func, x\_data, y\_data)  
 y\_fit = func(x\_data, \*popt)  
 residuals = y\_data - y\_fit  
 ss\_res = np.sum(residuals\*\*2)  
 ss\_tot = np.sum((y\_data - np.mean(y\_data))\*\*2)  
 r\_squared = 1 - (ss\_res / ss\_tot)  
 return popt, r\_squared  
  
# Визначення коефіцієнтів та оцінка якості регресії для кожного типу  
linear\_coeffs, linear\_r\_squared = compare\_regression(x\_data, y\_data, linear\_regression)  
hyperbolic\_coeffs, hyperbolic\_r\_squared = compare\_regression(x\_data, y\_data, hyperbolic\_regression)  
power\_coeffs, power\_r\_squared = compare\_regression(x\_data, y\_data, power\_regression)  
exponential\_coeffs, exponential\_r\_squared = compare\_regression(x\_data, y\_data, exponential\_regression)  
logarithmic\_coeffs, logarithmic\_r\_squared = compare\_regression(x\_data, y\_data, logarithmic\_regression)  
  
# Друк результатів  
print("Лінійна регресія: m =", linear\_coeffs[0], "b =", linear\_coeffs[1], "R^2 =", linear\_r\_squared)  
print("Гіперболічна регресія: a =", hyperbolic\_coeffs[0], "b =", hyperbolic\_coeffs[1], "R^2 =", hyperbolic\_r\_squared)  
print("Степенева регресія: a =", power\_coeffs[0], "b =", power\_coeffs[1], "R^2 =", power\_r\_squared)  
print("Показникова регресія: a =", exponential\_coeffs[0], "b =", exponential\_coeffs[1], "R^2 =", exponential\_r\_squared)  
print("Логарифмічна регресія: a =", logarithmic\_coeffs[0], "b =", logarithmic\_coeffs[1], "R^2 =", logarithmic\_r\_squared)  
  
# Побудова графіків  
plt.scatter(x\_data, y\_data, label='Табличні значення')  
plt.plot(x\_data, linear\_regression(x\_data, \*linear\_coeffs), label='Лінійна регресія')  
plt.plot(x\_data, hyperbolic\_regression(x\_data, \*hyperbolic\_coeffs), label='Гіперболічна регресія')  
plt.plot(x\_data, power\_regression(x\_data, \*power\_coeffs), label='Степенева регресія')  
plt.plot(x\_data, exponential\_regression(x\_data, \*exponential\_coeffs), label='Показникова регресія')  
plt.plot(x\_data, logarithmic\_regression(x\_data, \*logarithmic\_coeffs), label='Логарифмічна регресія')  
plt.xlabel('x')  
plt.ylabel('y')  
plt.legend()  
plt.grid(True)  
plt.show()

**Контрольні питання:**

1. **Загальна постановка задачі знаходження наближуючої функції**:
   * Задача полягає у знаходженні функції, яка найкращим чином апроксимує залежність між вхідними та вихідними даними.
2. **Суть наближення таблично заданої функції за методом найменших квадратів**:
   * Метод найменших квадратів полягає у мінімізації суми квадратів відхилень між спостережуваними значеннями та значеннями, передбаченими наближувальною функцією.
3. **Функції, які можуть бути використані в якості наближувальних**:
   * Лінійна, квадратична, показникова, степенева, логарифмічна, поліноміальна тощо.
4. **Знаходження відхилень виміряних значень від обчислених за формулою наближуючої функції**:
   * Відхилення обчислюються шляхом віднімання фактичних значень від значень, отриманих за допомогою наближувальної функції.
5. **Знаходження наближуючої функції у вигляді лінійної функції F(x,a,b)=ax+b**:
   * Це можна зробити за допомогою методу найменших квадратів, де коєфіцієнти aa і bb знаходяться шляхом мінімізації суми квадратів відхилень між фактичними значеннями і значеннями, обчисленими за формулою ax+bax+b.
6. **Знаходження наближуючої функції у вигляді квадратичної функції F(x,a,b,c)=ax²+bx+c**:
   * Аналогічно, метод найменших квадратів можна використати для знаходження коефіцієнтів aa, bb, і cc, які мінімізують відхилення між спостережуваними даними і значеннями, обчисленими за формулою ax2+bx+cax2+bx+c.
7. **Приведення показникової, степеневої, логарифмічної функцій до лінійної форми**:
   * Це можна зробити шляхом використання логарифмів, зміни змінних, або перетворень, які зводять функцію до лінійної форми.
8. **Як функція трьох змінних може приймати найменше значення**:
   * Мінімальне значення функції може бути знайдено шляхом вирішення системи рівнянь або застосування методів оптимізації, таких як градієнтний спуск.
9. **Коефіцієнт кореляції і його знаходження**:
   * Коефіцієнт кореляції вимірює силу та напрям зв'язку між двома змінними. Він знаходиться шляхом обчислення коваріації між двома змінними і ділення її на добуток стандартних відхилень цих змінних.
10. **Межі значень коефіцієнта кореляції і їх інтерпретація**:
    * Коефіцієнт кореляції може приймати значення від -1 до 1. Значення близькі до 1 вказують на сильний позитивний зв'язок, значення близькі до -1 вказують на сильний негативний зв'язок, а значення близькі до 0 вказують на відсутність зв'язку.
11. **Відхилення**:
    * Відхилення - це різниця між фактичним значенням і передбаченим (або обчисленим) значенням.
12. **Визначення правильності обраної функції**:
    * Правильність обраної функції може бути оцінена за допомогою коефіцієнта детермінації R2R2, який вказує на те, яка частина варіації вихідних даних пояснюється моделлю.

**Висновок:**

За допомогою методу найменших квадратів ми успішно побудували рівняння регресії для даних, що були задані таблично. Порівнявши різні види регресій, ми визначили, що краще всього для цих даних підходить гіперболічна регресія. Аналіз відхилень та порівняння отриманих результатів з табличними даними підтвердили правильність наших висновків. Використання Python спростило процес обробки даних та аналізу результатів. Таким чином, ми отримали практичні навички з обробки експериментальних даних та побудови регресійних моделей.