**Міністерство освіти і науки України**

**Національний технічний університет України «КПІ» імені Ігоря Сікорського**

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи №2**

**з навчальної дисципліни «Вступ до технології Data Science»**

**Тема:**

**ДОСЛІДЖЕННЯ АЛГОРИТМІВ ЗГЛАДЖУВАННЯ**

**ЗА НАКОПИЧЕНОЮ ВИБІРКОЮ**

**Виконав:**

Студент X курсу кафедри ОТ ФІОТ,

Навчальної групи ІА-73

Петренко П.П.

**Перевірив:**

Професор кафедри ОТ ФІОТ

Писарчук О.О.

**Київ 2023**

**І. Мета:**

виявити дослідити та узагальнити особливості застосування методів первинної обробки експериментальних вибірок – виявлення аномальних вимірів та алгоритмів накопиченого згладжування з використанням спеціалізованих пакетів мови програмування Python.

**ІІ. Завдання:**

**І. Розробити універсальний скрипт мовою Python що реалізує:**

| **Варіант** (порядковий номер в списку групи) | **Алгоритм виявлення**  **аномальних вимірів** | **Метод усунення впливу**  **аномальних вимірів** |
| --- | --- | --- |
| 3, 18 | За коефіцієнтом старіння інформації | Відкидання вимірів |

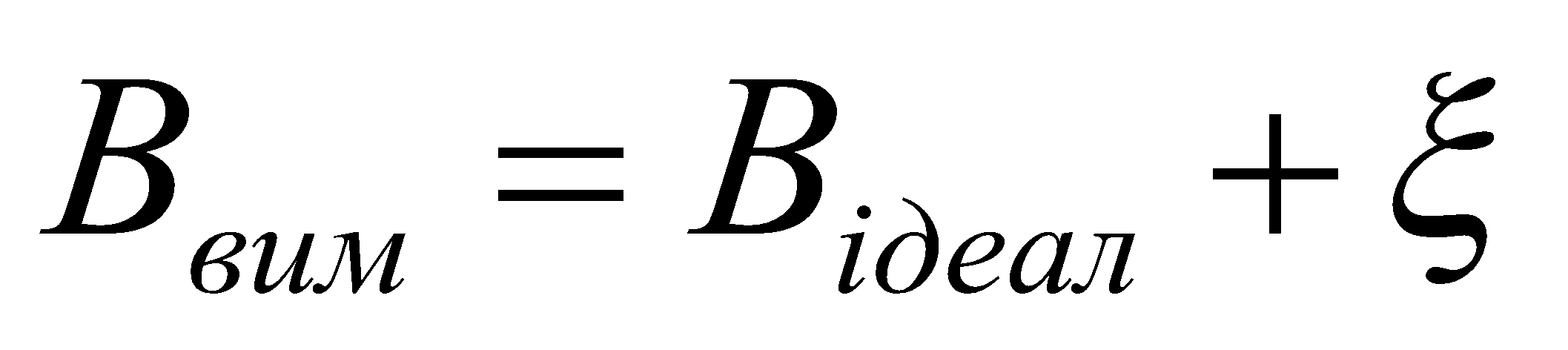
Лабораторія провідної ІТ-компанії реалізує масштабний проект розробки універсальної платформи з обробки Big Data масиву експериментальних даних поточного спостереження для виявлення закономірностей і прогнозування розвитку контрольованого процесу. Платформа передбачає розташування back-end компоненти на власному хмарному сервері з наданням повноважень користувачам заздалегідь адаптованого front-end функціоналу універсальної платформи.

Замовниками ресурсів платформи є: державні та комерційні компанії валютного трейдінгу для прогнозування динаміки зміни курсу валют та ціни інших товарів; метеорологічні служби для прогнозування параметрів метеоумов; департаменти охорони здоров’я для прогнозування зміни показників епідеміологічних ситуацій.

В продовження розвитку задач проекту минулого тижня (лабораторна робота №1) поточний перелік задач (tasks) для реалізації їх у межах лабораторної роботи №2 (Sprint – протягом тижня) для Вас, як Data Science [Engineer](https://jobs.dou.ua/companies/gravitum/vacancies/147764/) на проекті включає:

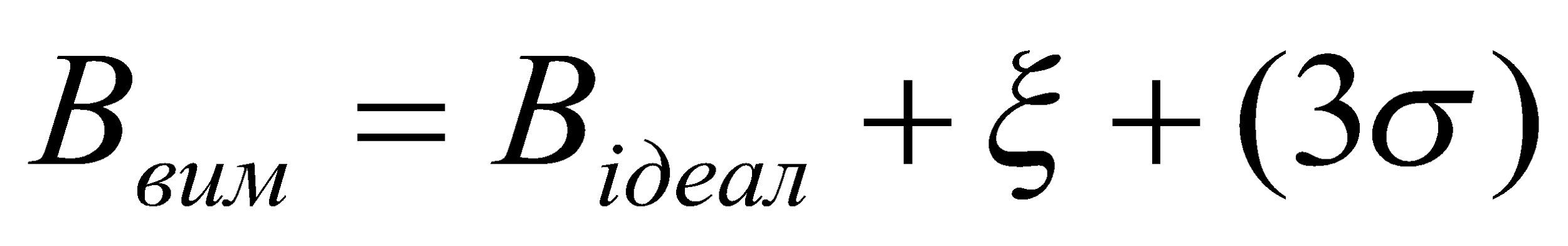
*1. Модель експериментальної вибірки з аномаліями відповідно до пунктів.*

1.1. Розробити модель дискретних значень виміряних параметрів експериментальної вибірки з характеристиками: трендова модель має квадратичний закон зміни; вибірка має 1000 вимірів; випадкова похибка вимірів розподілена за нормальним законом з нульовим середнім та змінним значенням середньоквадратичної похибки вимірювання; модель виміру – адитивна:

.

Для виконання даного завдання використати результати лабораторної роботи №1.

1.2. Модель генерації аномальних вимірів випадкової величини:

.

Аномальні виміри складають 10% від загальної кількості вимірів у експериментальної вибірки. Аномальні виміри рівномірно розташовані у межах дискретних значень експериментальної вибірки.

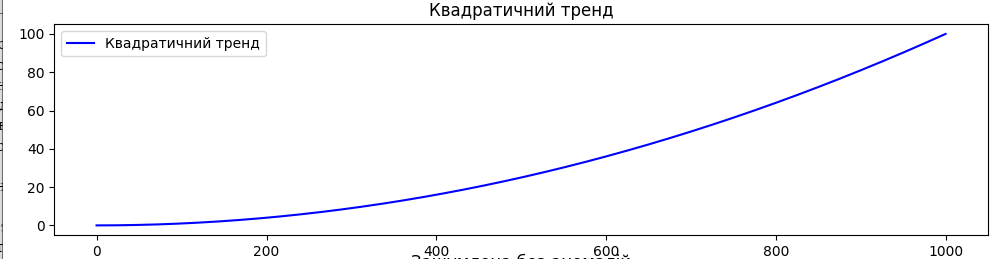


Рисунок 1. Квадратичний тренд

*2. Виявлення аномальних вимірів та усунення їх впливу на результати обробки.*

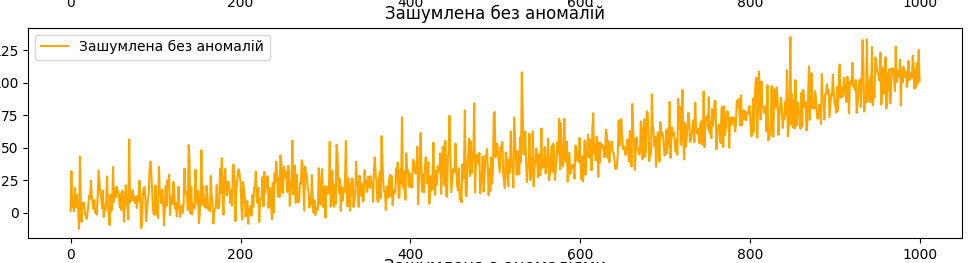


Рисунок 2. Адитивна модель ( квадратичного тренду та шуму за експоненційним розподілом )

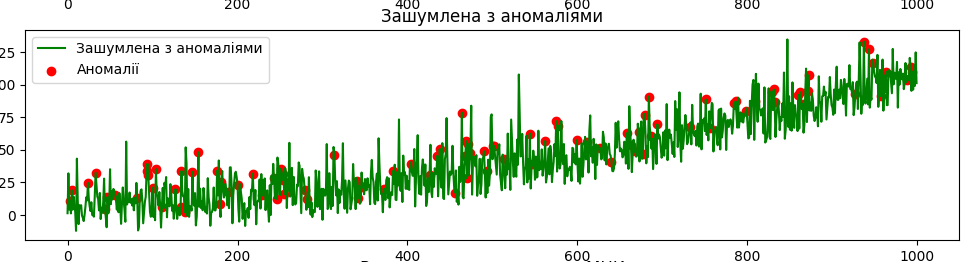


Рисунок 3. Визначення аномалій

*3. Здійснити згладжування експериментальної вибірки за відсутності аномальних вимірів відповідно до матричної форми метода найменших квадратів (МНК).*

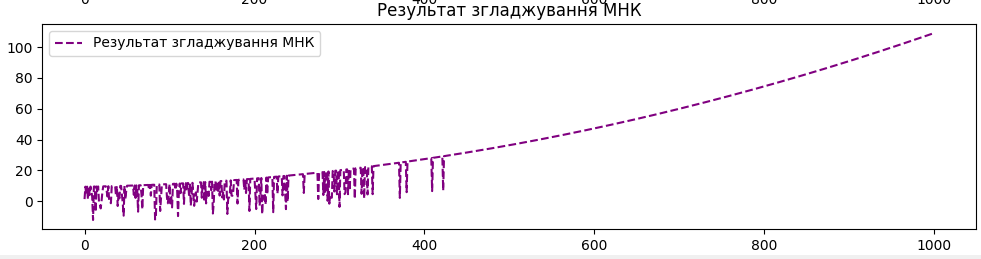


Рисунок 4. Згладжування МНК

5.1. Статистичні характеристики (математичне сподівання, середньоквадратичне відхилення) закону розподілу випадкової похибки вимірів; вхідної вибірки значень (зашумленої без аномальних вимірів); аномальної вибірки (зашумленої з аномальними вимірами); результатів згладжування МНК – у консолі та у формі таблички у звіті з лабораторної роботи.

*4. З використанням методу Монте-Карло дослідити статистичні характеристики (математичне сподівання, середньоквадратичне відхилення, гістограма закону розподілу):* закону розподілу випадкової похибки вимірів; вхідної вибірки значень (зашумленої без аномальних вимірів); аномальної вибірки (зашумленої з аномальними вимірами); результатів згладжування МНК.

*5. Відобразити результати розрахунків:*

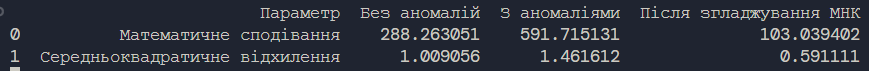


Рисунок 5. Статистичні характеристики методом монте-карло

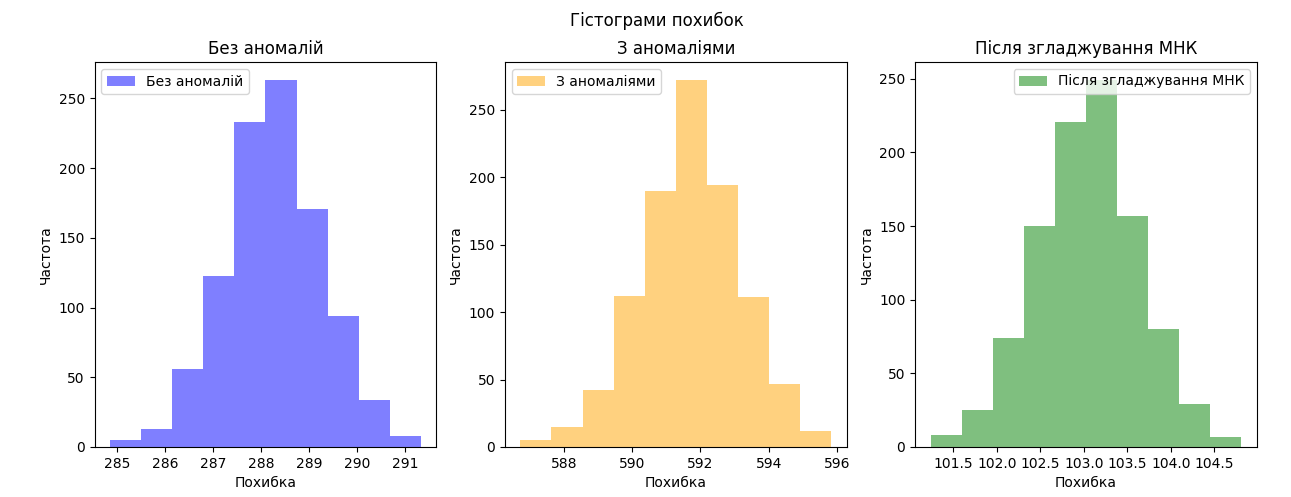


Рисунок 6. Гістограми розподілу

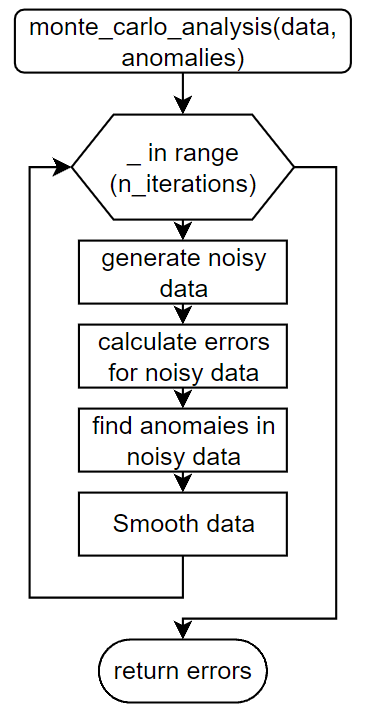


Рисунок 7. Блоксхема методу монте-карло

5.2. Графіки (в одному графічному вікні): квадратичного тренду; зашумленої без аномальних вимірів вибірки; зашумленої з аномальними вимірами вибірки; результатів згладжування МНК.

5.3. Гістограми (в одному графічному вікні) похибок: зашумленої без аномальних вимірів вибірки; зашумленої з аномальними вимірами вибірки; результатів згладжування МНК.

**I.I. Блоксхема**

**ІІ. Провести дослідження зміни статистичних характеристик сформованих моделей при зміні стохастичних характеристик помилки експериментальних даних.**

1. Зміна розподілу

* Нормальний розподіл
  + persentage anomalies: 0.1

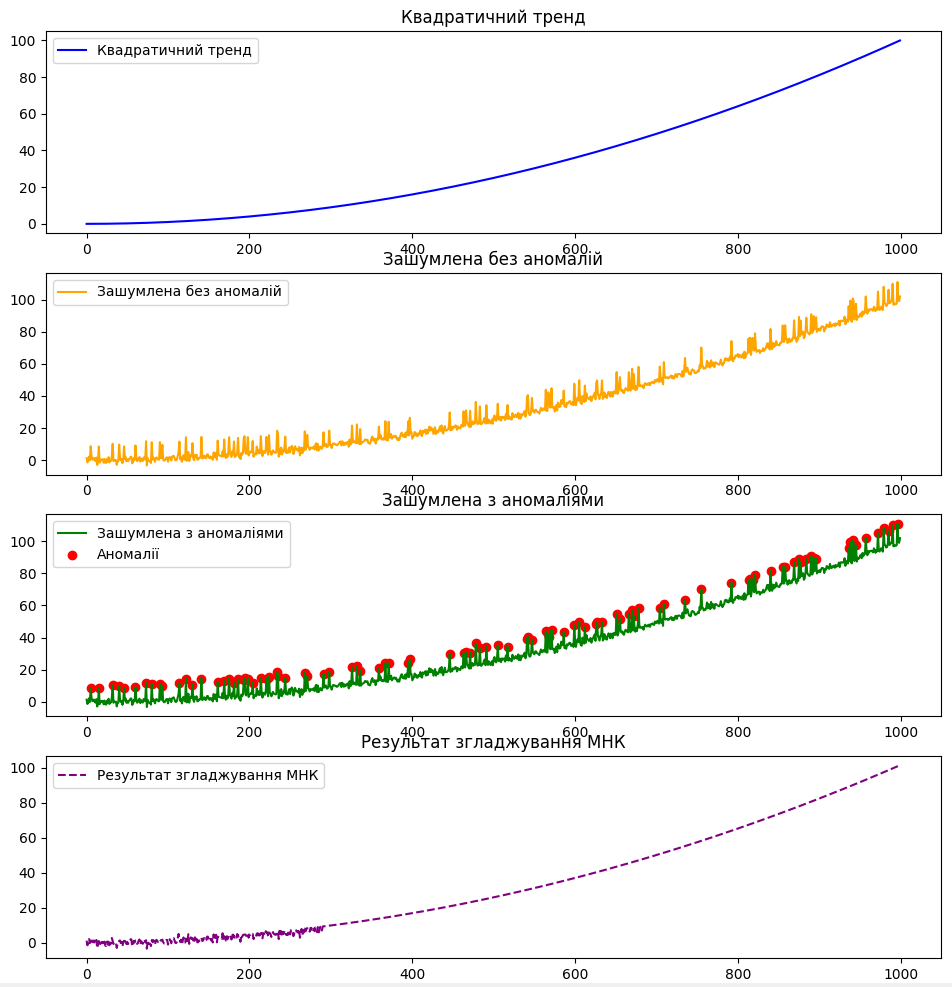


Рисунок 9. Графіки при нормальному розподілі

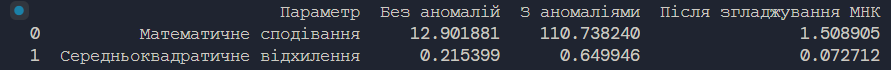


Рисунок 10. Статистичні дані при нормальному розподілі

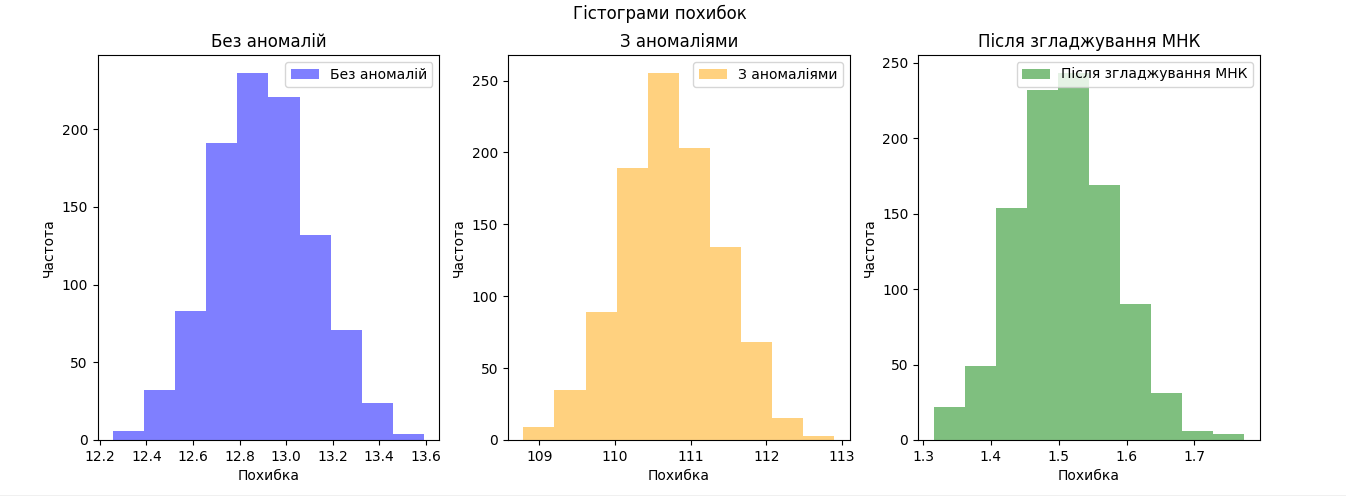


Рисунок 11. Гістограми розподілу ( при адитивній моделі з нормальним розподілом )

1. Зміна лямбди експоненційного розподілу

* lambda: 0.001

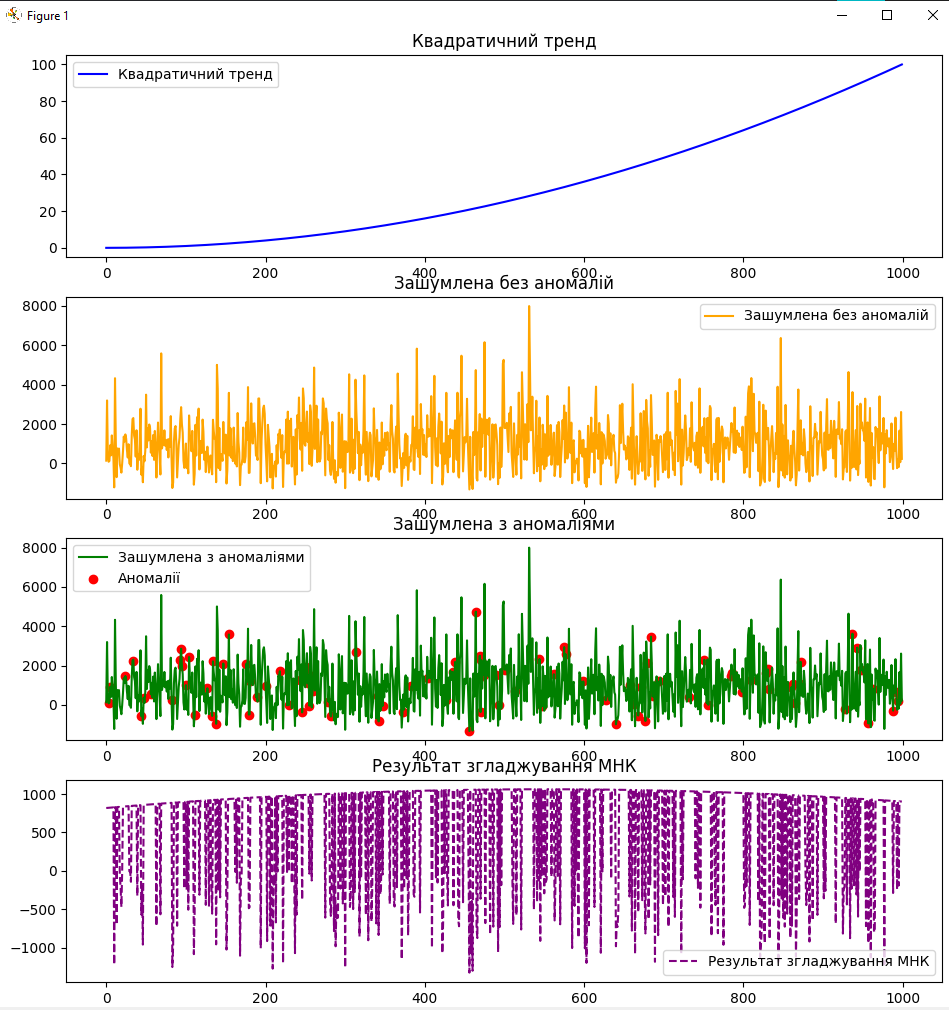


Рисунок 12.

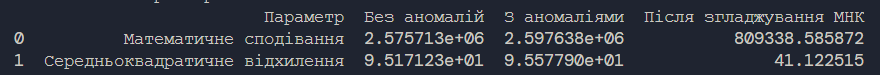


Рисунок 13.

* lambda: 4

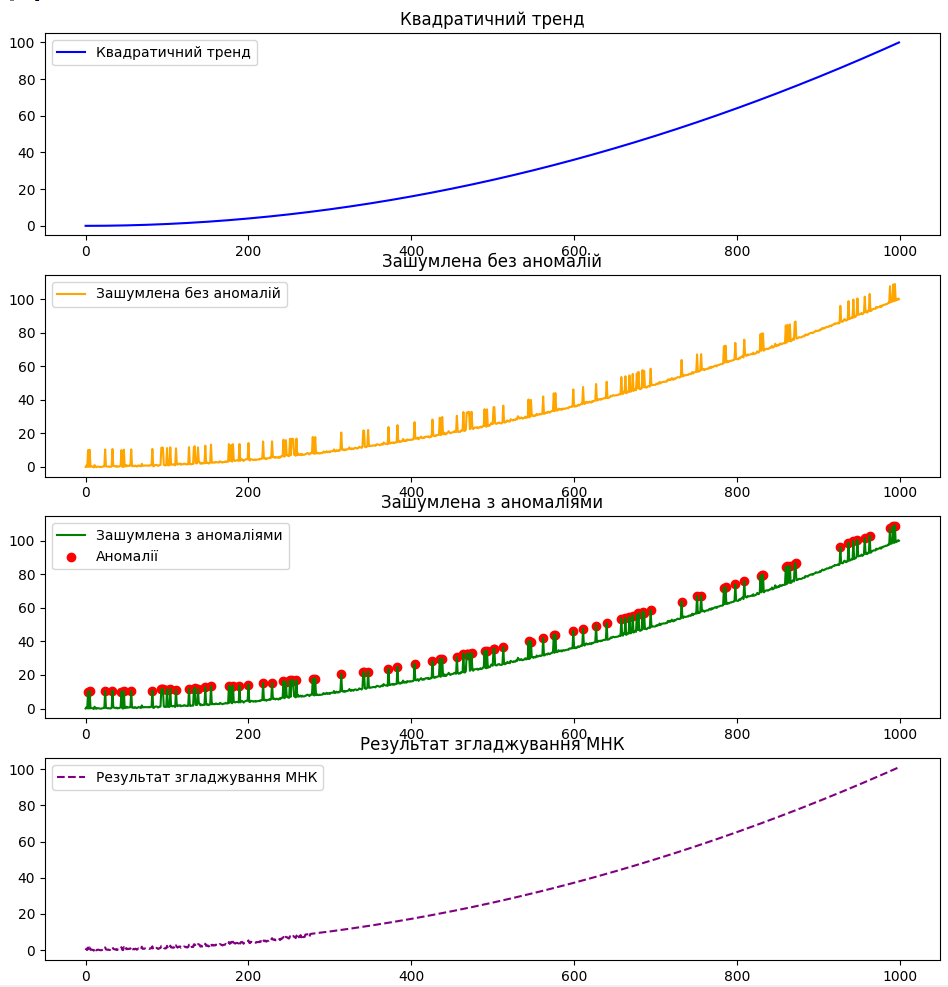


Рисунок 14.

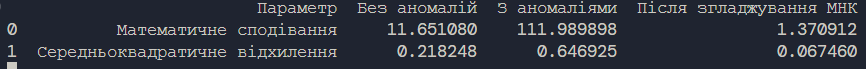


Рисунок 15.

1. Нормальний Розподіл:

* Рівномірний Шум: Нормальний розподіл випадкового компоненту генерує рівномірний шум, який може допомогти згладжувати дані. Це пов'язано із тим, що нормальний розподіл є основним розподілом для шуму в багатьох природних процесах, і він добре відображає реальні умови багатьох систем.
* Відновлення Вимірів: Також важливо відзначити, що нормальний розподіл дозволяє ефективно відновлювати виміри та враховувати особливості нормальних умов, що робить його добрим вибором для багатьох задач моделювання.

1. Експоненційний Розподіл:

* Зайві Аномалії: Виправданням для того, що експоненційний розподіл може генерувати зайві аномалії, може бути те, що він має тенденцію генерувати великі значення з меншою ймовірністю, що може призвести до виражених аномалій у випадковому шумі.
* Використання експоненційного розподілу може бути обґрунтовано для визначених умов або завдань.

1. Зміна Лямбди експоненційного розподілу та щільності ймовірності:

* Збільшення λ: Зменшує середній час між подіями, що може призвести до більшої концентрації значень ближче до нуля.
* Зменшення λ: Збільшує середній час між подіями, що може призвести до розтягнення графіка та більше значень, віддалених від нуля.
* Зміна λ може впливати на те, як інтерпретуються часові або інші інтервальні дані.
* Низьке λ може вказувати на велику інтенсивність подій або короткий час між подіями.

Вибір розподілу випадкового компоненту повинен залежати від конкретної задачі та природи даних. Важливо розуміти, як розподіл впливає на результати та як його властивості відповідають умовам задачі.

**ІІІ. Довести адекватність сформованих моделей та працездатність розробленого скріпта.**

**Розроблений код повинен бути раціональним та відповідати вимогам до чистого коду.**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import pandas as pd**

***# Алгоритм виявлення аномальних вимірів: Метод найменших квадратів***

**def detect\_anomalies(data):**

**"""**

**Виявляє аномальні виміри за допомогою порогового методу.**

**Parameters:**

**- data: numpy array, вхідні виміри**

**Returns:**

**- anomalies: numpy array, індекси аномальних вимірів**

**"""**

**threshold = 7.5**

**anomalies = np.where(data > threshold)[0]**

**return anomalies**

**def detect\_anomalies\_decay(data, alpha=0.05):**

**"""**

**Алгоритм виявлення аномалій за коефіцієнтом старіння інформації.**

**Parameters:**

**- data: numpy array, вхідні виміри**

**- alpha: float, коефіцієнт старіння інформації**

**Returns:**

**- anomalies: numpy array, індекси аномальних вимірів**

**"""**

**anomalies = []**

**for i in range(1, len(data)):**

**if abs(data[i] - data[i-1]) > alpha:**

**anomalies.append(i)**

**return np.array(anomalies)**

***# Метод усунення впливу аномальних вимірів: Відновлення вимірів методом найменших квадратів***

**def restore\_measurements(data, anomalies):**

**"""**

**Метод усунення впливу аномальних вимірів (відновлення).**

**Parameters:**

**- data: numpy array, вхідні виміри**

**- anomalies: numpy array, індекси аномальних вимірів**

**Returns:**

**- restored\_data: numpy array, виміри після відновлення аномалій методом найменших квадратів**

**"""**

**restored\_data = np.copy(data)**

**restored\_data[anomalies] = np.nan *# Позначте аномалії для відновлення***

**valid\_indices = np.where(~np.isnan(restored\_data))[0]**

**restored\_data[anomalies] = np.polyval(np.polyfit(valid\_indices, restored\_data[valid\_indices], 2), anomalies)**

**return restored\_data**

***# Здійснення згладжування методом найменших квадратів***

**def smooth\_data(data, anomalies):**

**"""**

**Згладжує виміри методом найменших квадратів.**

**Parameters:**

**- data: numpy array, вхідні виміри**

**- anomalies: numpy array, індекси аномальних вимірів**

**Returns:**

**- smoothed\_data: numpy array, виміри після згладжування аномалій методом найменших квадратів**

**"""**

**valid\_indices = np.where(~np.isnan(data))[0]**

**smoothed\_data = np.copy(data)**

**smoothed\_data[anomalies] = np.polyval(np.polyfit(valid\_indices, data[valid\_indices], 2), anomalies)**

**return smoothed\_data**

***# Здійснення Монте-Карло аналізу***

**def monte\_carlo\_analysis(data, anomalies):**

**"""**

**Монте-Карло аналіз для оцінки результатів згладжування.**

**Parameters:**

**- data: numpy array, вхідні виміри**

**- anomalies: numpy array, індекси аномальних вимірів**

**Returns:**

**- results: dictionary, результати Монте-Карло аналізу**

**"""**

**n\_iterations = 1000**

**errors\_noisy = []**

**errors\_anomalous = []**

**errors\_smoothed = []**

**for \_ in range(n\_iterations):**

**noisy\_data = data + np.random.normal(0, 1, len(data))**

**errors\_noisy.append(np.mean((noisy\_data - ideal\_values) \*\* 2))**

**anomalous\_data = np.copy(noisy\_data)**

**anomalous\_data[anomalies] += 10**

**errors\_anomalous.append(np.mean((anomalous\_data - ideal\_values) \*\* 2))**

**smoothed = smooth\_data(noisy\_data, anomalies)**

**errors\_smoothed.append(np.mean((smoothed - ideal\_values) \*\* 2))**

**return {**

**'errors\_noisy': errors\_noisy,**

**'errors\_anomalous': errors\_anomalous,**

**'errors\_smoothed': errors\_smoothed**

**}**

***# Відображення результатів***

**def display\_results(ideal, noisy, noisy\_with\_anomalies, smoothed, errors\_noisy, errors\_anomalous, errors\_smoothed):**

**"""**

**Відображає результати експерименту.**

**Parameters:**

**- ideal: numpy array, ідеальні виміри**

**- noisy: numpy array, зашумлені виміри без аномалій**

**- noisy\_with\_anomalies: numpy array, зашумлені виміри з аномаліями**

**- smoothed: numpy array, виміри після згладжуван**

**- errors\_noisy: numpy array, масив квадратичних похибок для зашумлених вимірів без аномалій**

**- errors\_anomalous: numpy array, масив квадратичних похибок для зашумлених вимірів з аномаліями**

**- errors\_smoothed: numpy array, масив квадратичних похибок для вимірів після згладжування**

**"""**

**statistics\_table = pd.DataFrame({**

**'Параметр': ['Математичне сподівання', 'Середньоквадратичне відхилення'],**

**'Без аномалій': [np.mean(errors\_noisy), np.std(errors\_noisy)],**

**'З аномаліями': [np.mean(errors\_anomalous), np.std(errors\_anomalous)],**

**'Після згладжування МНК': [np.mean(errors\_smoothed), np.std(errors\_smoothed)]**

**})**

**print(statistics\_table)**

**plt.figure(figsize=(10, 15))**

***# Перший графік - Квадратичний тренд***

**plt.subplot(4, 1, 1)**

**plt.plot(ideal, label='Квадратичний тренд', color='blue')**

**plt.legend()**

**plt.title('Квадратичний тренд')**

***# Другий графік - Зашумлена без аномалій***

**plt.subplot(4, 1, 2)**

**plt.plot(noisy, label='Зашумлена без аномалій', color='orange')**

**plt.legend()**

**plt.title('Зашумлена без аномалій')**

***# Третій графік - Зашумлена з аномаліями***

***#plt.title('Зашумлена з аномаліями')***

**plt.subplot(4, 1, 3)**

**plt.plot(noisy\_with\_anomalies, label='Зашумлена з аномаліями', color='green')**

***# Перевірка, чи аномалії в межах вимірів***

**valid\_anomalies = anomalies[anomalies < len(noisy\_with\_anomalies)]**

**plt.scatter(valid\_anomalies, noisy\_with\_anomalies[valid\_anomalies], color='red', label='Аномалії')**

**plt.legend()**

**plt.title('Зашумлена з аномаліями')**

***# Четвертий графік - Результат згладжування МНК***

**plt.subplot(4, 1, 4)**

**plt.plot(smoothed, label='Результат згладжування МНК', linestyle='--', color='purple')**

**plt.legend()**

**plt.title('Результат згладжування МНК')**

***# Додаткові налаштування***

**plt.tight\_layout()**

**plt.show()**

**fig, axs = plt.subplots(1, 3, figsize=(15, 5))**

***# Гістограма для без аномалій***

**axs[0].hist(errors\_noisy, alpha=0.5, color='blue', label='Без аномалій')**

**axs[0].set\_title('Без аномалій')**

***# Гістограма для з аномаліями***

**axs[1].hist(errors\_anomalous, alpha=0.5, color='orange', label='З аномаліями')**

**axs[1].set\_title('З аномаліями')**

***# Гістограма після згладжування МНК***

**axs[2].hist(errors\_smoothed, alpha=0.5, color='green', label='Після згладжування МНК')**

**axs[2].set\_title('Після згладжування МНК')**

***# Додаткові налаштування***

**for ax in axs:**

**ax.legend()**

**ax.set\_xlabel('Похибка')**

**ax.set\_ylabel('Частота')**

**plt.suptitle('Гістограми похибок')**

**plt.show()**

***# Виведення таблички у консоль***

**print("\nСтатистичні характеристики:")**

**print(statistics\_table)**

***# Закон зміни похибки – експонентційний;***

**def exp\_d(size=100000, lambda\_val=0.1):**

**""" Exponential Distribution array generator """**

**return np.random.exponential(scale=1.0 / lambda\_val, size=size)**

***# Генерація експериментальної вибірки***

**np.random.seed(42)**

**n\_measurements = 1000**

**percentage\_anomalies = 0.1**

**ideal\_values = np.linspace(0, 10, n\_measurements) \*\* 2**

***#noise = np.random.normal(0, 1, n\_measurements)***

**noise = exp\_d(n\_measurements, 4)**

**anomalies = np.random.choice(n\_measurements, int(n\_measurements \* percentage\_anomalies), replace=False)**

**measurements = ideal\_values + noise + 3 \* np.std(noise) \* (np.random.rand(n\_measurements) - 0.5)**

**measurements[anomalies] += 10 *# Додавання аномальних вимірів***

***# Алгоритм виявлення аномальних вимірів: Метод найменших квадратів***

**anomalies\_detected\_var1 = detect\_anomalies(measurements)**

***# Метод усунення впливу аномальних вимірів: Відновлення вимірів***

**restored\_measurements\_var1 = restore\_measurements(measurements, anomalies\_detected\_var1)**

***# Здійснення згладжування методом найменших квадратів***

**smoothed\_data\_var1 = smooth\_data(measurements, anomalies\_detected\_var1)**

***# Монте-Карло аналіз***

**monte\_carlo\_results\_var1 = monte\_carlo\_analysis(measurements, anomalies\_detected\_var1)**

**III. Висновок:**

виявив дослідив та узагальнив особливості застосування методів первинної обробки експериментальних вибірок – виявлення аномальних вимірів та алгоритмів накопиченого згладжування з використанням спеціалізованих пакетів мови програмування Python.

Виконав: студент Петренко П.П.