Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ

ЗВІТ
з лабораторної роботи №1
з навчальної дисципліни «Технології Data Science»

Тема:

ПІДГОТОВКА ТА АНАЛІЗ ДАНИХ ДЛЯ СТАТИСТИЧНОГО НАВЧАННЯ

Виконав:
Студент 4 курсу кафедри ФІОТ,
Навчальної групи ІП-11
Олександр Головня

Перевірив:
Професор кафедри ОТ ФІОТ
Олексій Писарчук

Київ 2024

І. Мета:

Виявити дослідити та узагальнити особливості застосування методів статистичного навчання для задач визначення статистичних характеристик вхідного потоку даних з використанням спеціалізованих пакетів мови програмування Python.

ІІ. Завдання:

Варіант (порядковий номер в списку групи)	**ІІ рівень складності** **8 балів**
5	Закон зміни похибки – нормальний, рівномірний; Закон зміни досліджуваного процесу (тренду) – лінійний, квадратичний. Комбінаторика похибка / тренд – довільна. Реальні дані – 3 показники.

Завдання ІІ рівня – максимально 8 балів.

Розробити програмний скрипт мовою Python що забезпечує аналіз властивостей і характеристик вихідних даних відповідно до етапів:

1. Модель генерації випадкової величини за заданим у табл.1 додатку 1 закону розподілу;
2. Модель зміни (ідеальний тренд) досліджуваного процесу за заданим у табл.1 додатку 1 законом;
3. Адитивна модель статистичної вибірки відповідно до синтезованих в п.1,2 моделей випадкової (стохастична) і невипадкової складових. Параметри закону розподілу та закону зміни досліджуваного процесу обрати самостійно.
4. Визначення статистичних (числових) характеристик сформованих в п.1,3 вибірок (дисперсія, середньоквадратичне відхилення, математичне очікування, гістограма закону розподілу).
5. Визначення статистичних характеристик реальних даних, заданих файлом Oschadbank (USD).xls за умов табл. 1 додатку 1.
6. Провести аналіз отриманих результатів та верифікацію розробленого скрипта.

III. Результати виконання лабораторної роботи.

3.1. Синтезована математична модель перетворень графічних об’єктів відповідно до індивідуального завдання.

Синтезована математична модель в даній лабораторній роботі включає в себе дві основні компоненти:

1. Модель генерації випадкової величини: Ця модель використовується для створення випадкових чисел згідно з вказаними законами розподілу, такими як нормальний та експоненційний закони. Вона служить для створення помилок або шуму, які додаються до ідеального тренду.

Рівномірний закон розподілу:

***1. Щільність розподілу ймовірностей*:**

$$f(x)=\begin{cases}0, & при \quad x<a,\\ \dfrac{1}{b-a}, & при \quad a\le x\le b,\\ 0, & при \quad x>b.\end{cases}$$

де: x – випадкова величина; a, b – межі реалізації ВВ, параметри закону розподілу ВВ.

2. Числові характеристики:

Математичне сподівання:

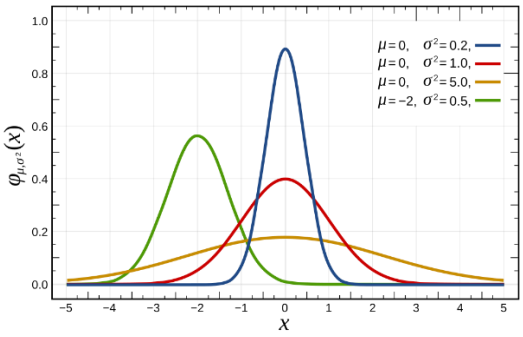
$$m=\frac{a+b}{2}$$

Дисперсія:

$$\sigma^2=\frac{\sum(x-\bar{x})^2}{n}$$

Нормальний закон розподілу:

1. ***Щільність розподілу ймовірностей*:**



$$f(x;\mu,\sigma)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

де μ — математичне сподівання, σ^2 — дисперсія випадкової величини. Параметр σ також відомий, як стандартне відхилення.

2. ***Числові характеристики:***

Математичне сподівання:

$$m_x=\frac{1}{n}\sum_{i=1}^{n}x_i,$$

Дисперсія:

$$D_x=\frac{1}{n}\sum_{i=1}^{n}(x_i-m_x)^2.$$

2. Модель ідеального тренду: Ця модель представляє собою ідеальний сценарій або закон зміни досліджуваного процесу. У лабораторній роботі реалізовані два види трендів: постійний (константний) та квадратичний. Модель ідеального тренду використовується для створення основної залежності вихідних даних.

Формула лінійної моделі виглядає так:

$$y(t) = a \cdot t + b$$

Де:

- $y(t)$ — значення залежної змінної в момент часу t,
- a — коефіцієнт нахилу (показує, як швидко змінюється залежна змінна зі зміною часу),
- b — початкове значення залежної змінної (коли $t = 0$).

Ця модель описує пряму лінію, де зміна значення відбувається з постійною швидкістю.

Формула квадратичної моделі виглядає так:

$$y(t) = a \cdot t^2 + b \cdot t + c$$

Де:

- $y(t)$ — значення залежної змінної в момент часу t,
- a — коефіцієнт при t^2 (визначає "викривленість" параболи),
- b — коефіцієнт при t (лінійний компонент),
- c — початкове значення залежної змінної.

Квадратична модель описує параболічну залежність, тобто процес з прискоренням або уповільненням.

Ці моделі можна використовувати для різних типів трендів у залежності від характеру змін досліджуваного процесу.

Методика визначення статистичних характеристик стохастичних даних.

1. Виділення систематичної складової (тренду).
2. Обчислення середнього значення (математичного очікування)
3. Обчислення Дисперсії. Дисперсія показує, наскільки дані розкидані відносно середнього значення. Чим більша дисперсія, тим більше варіативність даних.
4. Середньоквадратичне відхилення. Це квадратний корінь із дисперсії. Воно показує, наскільки значення в середньому відхиляються від математичного очікування.
5. Коефіцієнт асиметрії (скос). Асиметрія вимірює ступінь "нерівності" розподілу даних. Якщо асиметрія дорівнює нулю, то розподіл симетричний.
6. Гістограма — це графічне представлення розподілу даних, яке показує, як часто зустрічаються певні значення. Її можна використовувати для оцінки форми розподілу й визначення, чи відповідає він, наприклад, нормальному розподілу.

Визначення статистичних характеристик стохастичних даних дозволяє зробити висновки про властивості процесу та його випадкову природу. Такий аналіз допомагає краще зрозуміти поведінку даних та прогнозувати їх у майбутньому.

3.2. Блок схема алгоритму та її опис.

Рис.1 - Блок-схема алгоритму програми

Опис алгоритму:

Початок програми.

1. Ініціалізація параметрів:

Визначення обсягу вибірки (кількість даних) n. Задаються значення констант: обсяг вибірки, кількість реалізацій, коефіцієнт аномальних викидів, параметри нормального розподілу тощо.

2. Визначення моделей:

Викликаються функції моделей

Модель тренду:

- Квадратична модель.
- Лінійна модель.

Шуми:

- Нормальні помилки.
- Рівномірні помилки.

3. Моделювання даних з шумами:

Створюються моделі з трендом та шумами:

- Нормальні помилки + тренд.
- Рівномірні помилки + тренд.

Для цих моделей будуються графіки та проводиться статистичний аналіз.

4. Моделювання з аномальними викидами :

Створюються моделі тренду з нормальними помилками та АВ.

Створюються моделі тренду з рівномірними помилками та АВ.

Будуються графіки та виконується статистичний аналіз цих моделей.

5. Аналіз реальних даних:

Завантажуються реальні дані з архіву Ощадбанку, проводиться обробка та виведення результатів:

- Коливання курсу USD в 2022 році.

- Аналіз статистичних характеристик реальних даних.

6. Завершення програми.

3.3. Опис структури проекту програми в середовищі PyCharm.

Для реалізації розробленого алгоритму мовою програмування Python сформовано проєкт.

Проєкт базується на лінійній бізнес-логіці функціонального програмування та має таку структуру.

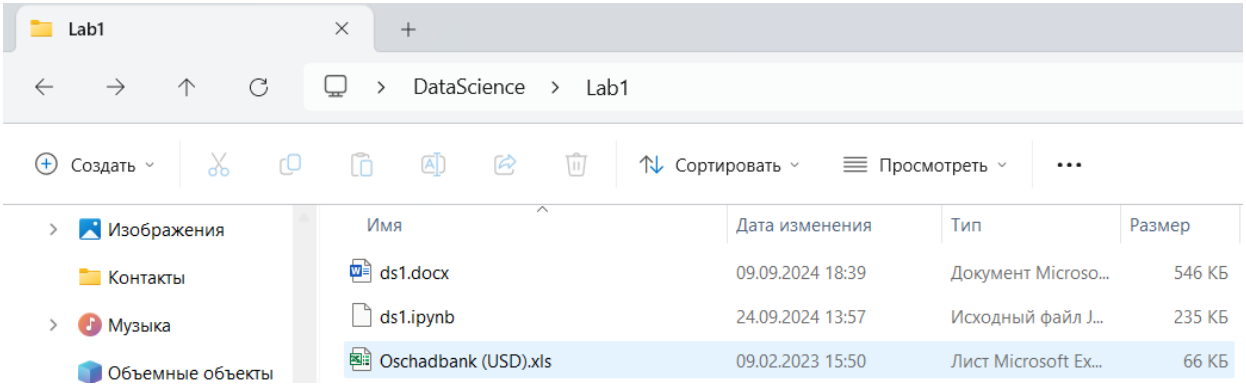


Рис.2. Структура проекту.

ds1.ipynb – файл програмного коду лабораторної роботи;
ds1.doc – файл звіту лабораторної роботи.
Oschadbank (USD).xls – файл xls з реальними даними.

3.4. Результати роботи програми відповідно до завдання.

Результатом роботи програми є:

Діаграми: Програма генерує різні діаграми, які візуалізують дані.

Числові характеристики: Програма розраховує числові характеристики для вибірок, такі як математичне сподівання (середнє значення), дисперсія (середньоквадратичне відхилення) та інші. Ці характеристики надають кількісні оцінки розподілу даних.

Вивід результатів: Результати виводяться на екран у вигляді текстових повідомлень, діаграм і графіків. Вони можуть бути використані для аналізу та порівняння різних даних та моделей.

```
Characteristics of Normal Distribution
Mean: -0.01673460549351011
Variance: 24.658480035381817
Standard Deviation: 4.965730564114591
------------------------
```

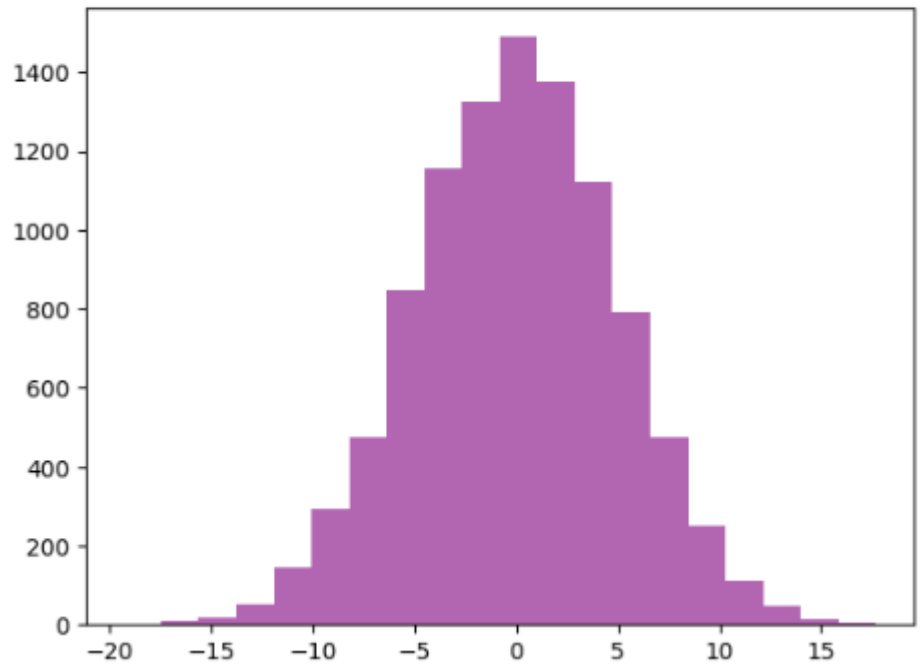


Рис.3. – Статистичні характеристики Нормального розподілу.

```
Characteristics of Uniform Distribution
Mean: 4928.875613753784
Variance: 8245611.13156134
Standard Deviation: 2871.5172177024015
--------------------
```

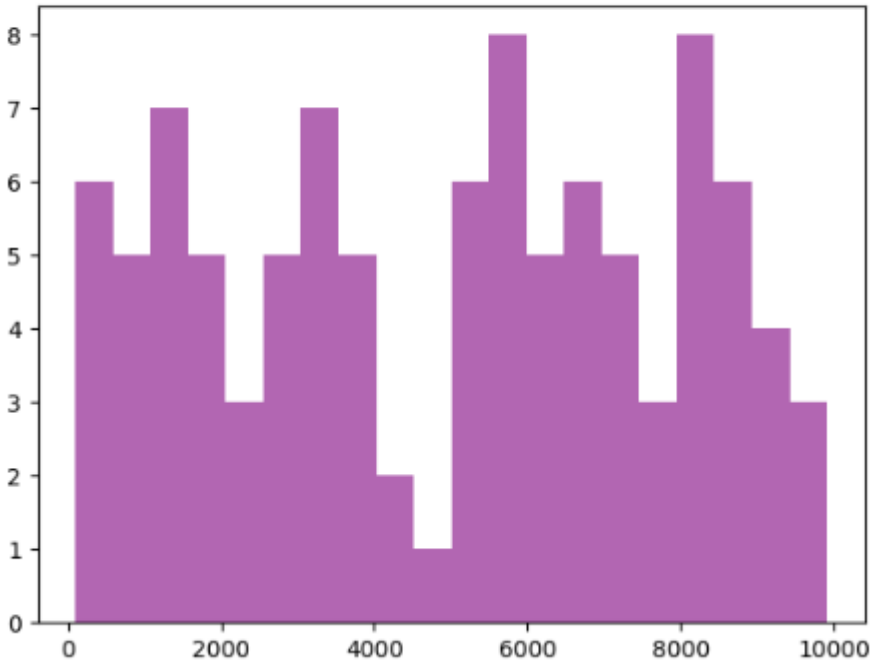
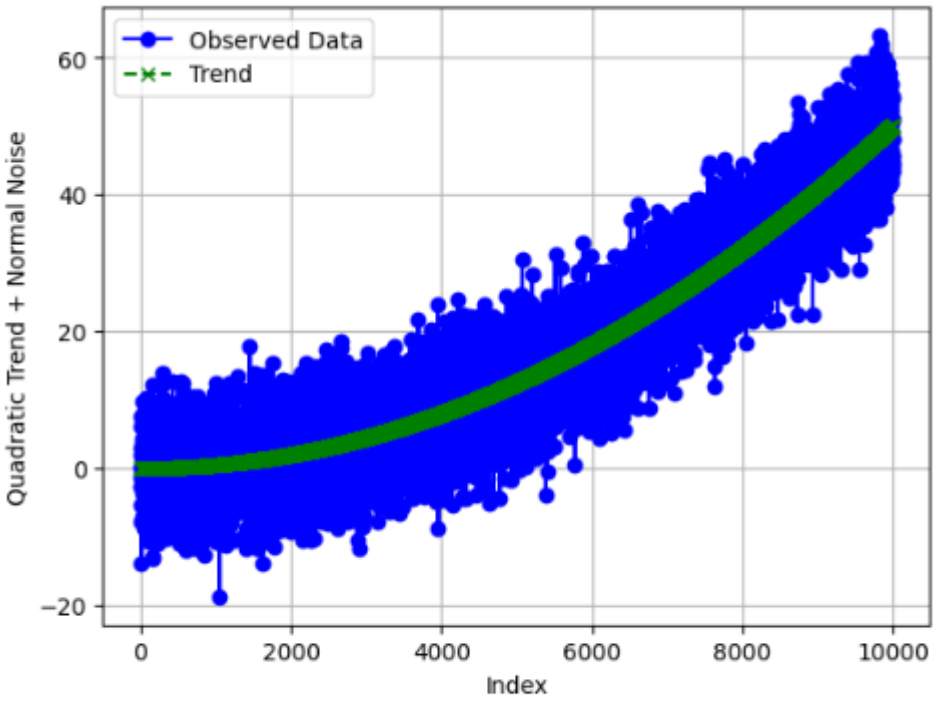
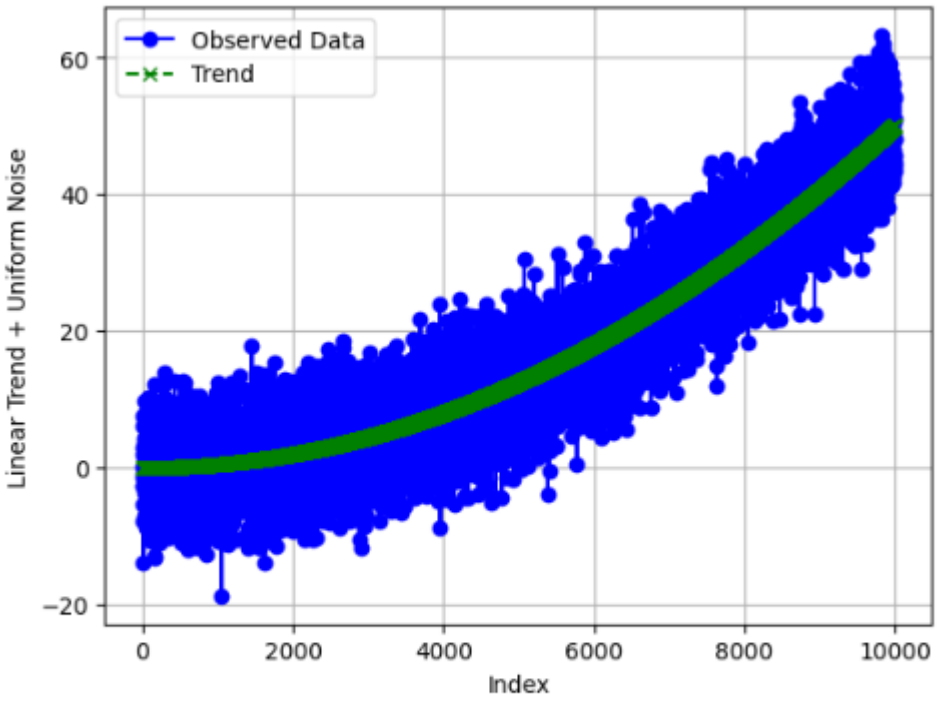


Рис.4. – Статистичні характеристики Рівномірного розподілу.



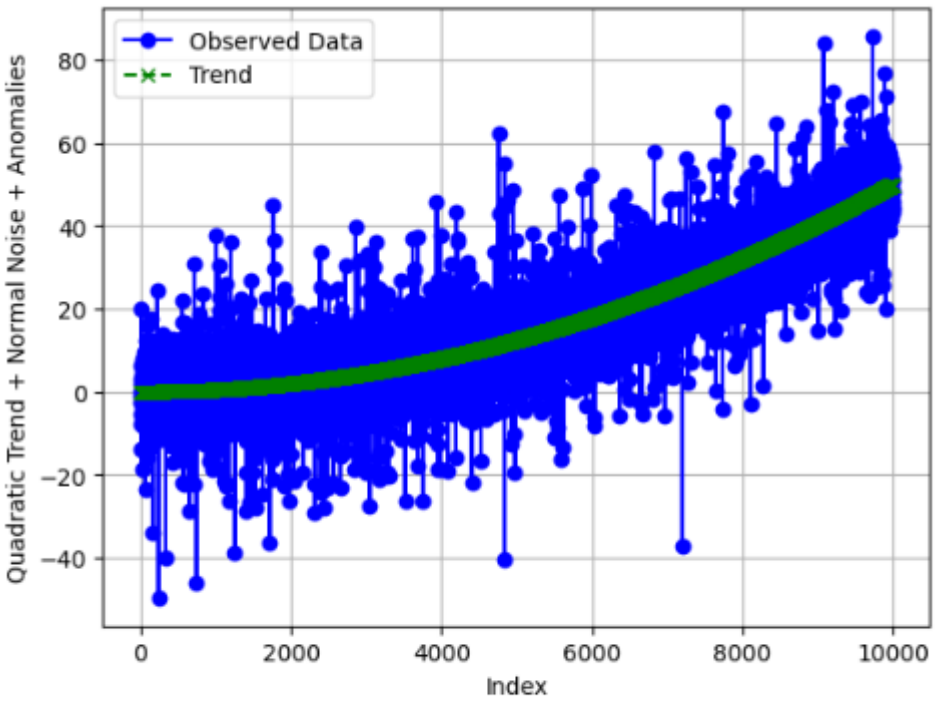

```
Sample Statistics
Median: 0.08871231007981062
Variance: 24.657895132358263
Standard Deviation: 4.965671669810466
```

Рис.5. – Квадратична модель з Норм. шумом



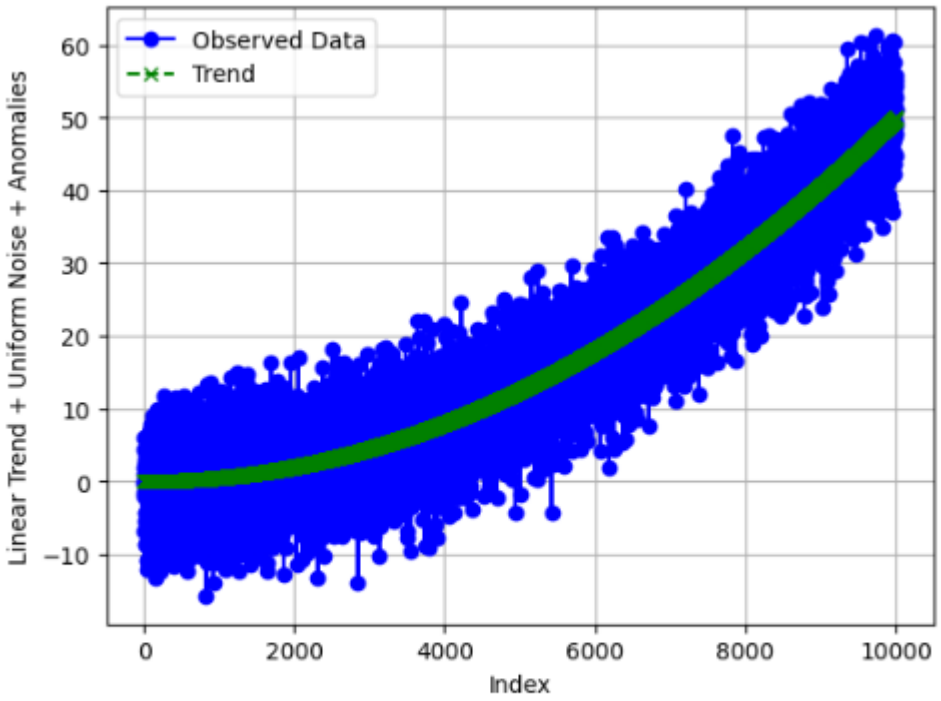

```
Sample Statistics
Median: 0.08871231007981062
Variance: 24.657895132358263
Standard Deviation: 4.965671669810466
```

Рис.6. – Лінійна модель з рівномірним шумом



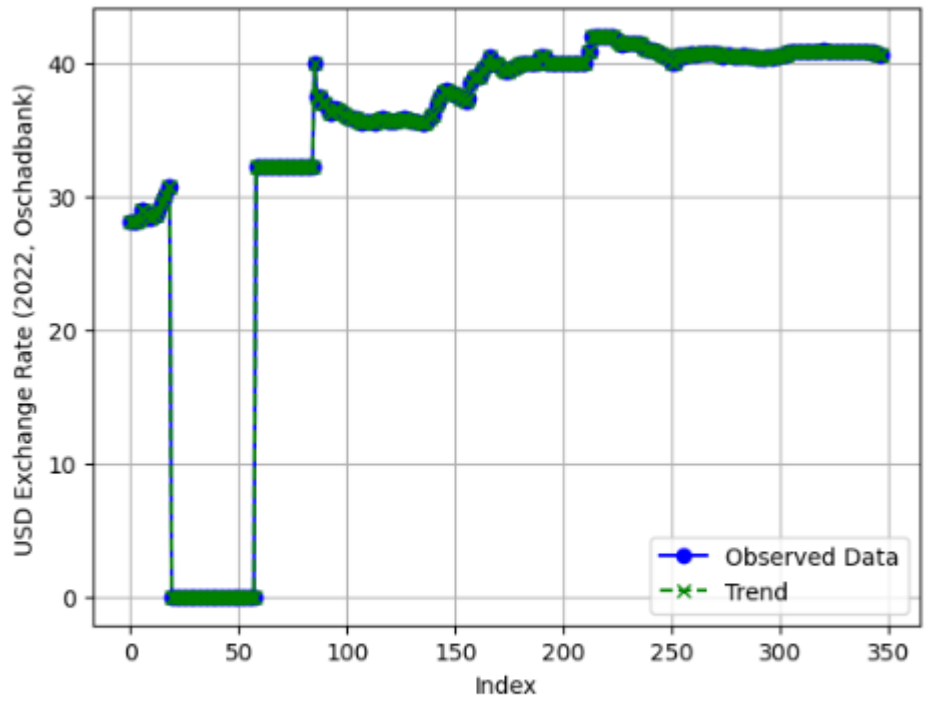

```
Sample Statistics
Median: 0.06706486716489168
Variance: 45.78644371829933
Standard Deviation: 6.766568090125107
```

Рис.5. – Квадратична модель з Норм. шумом + АВ




```
Sample Statistics
Median: 0.21196483871090083
Variance: 24.531819663812403
Standard Deviation: 4.952960696776465
```

Рис.6. – Лінійна модель з Рівномірним шумом + АВ




```
Sample Statistics
Median: 0.2243231691173122
Variance: 63.896746894523595
Standard Deviation: 7.993544075973035
```

Рис. 7. – Визначення статистичних характеристик реальних даних, заданих файлом Oschadbank (USD).xls

3.5. Програмний код.

Програмний код послідовно реалізує алгоритм рис.1 та спрямовано на отримання результатів, поданих вище. При цьому використано можливості Python бібліотек: pip; pandas; numpy; xlrd; matplotlib. Контексні коментарі пояснюють сутність окремих скриптів наведеного коду програми. (Повний лістинг коду у файлі ds1.py або github)

```python
if __name__ == '__main__':
    num_samples = 10000
    iteration = int(num_samples)
    anomaly_factor = 3
    anomaly_percent = 10
    anomaly_count = int((iteration * anomaly_percent) / 100)
    mean_val = 0
    std_dev = 5

    quadratic_trend = quadratic_model(num_samples)
    linear_trend = linear_model(num_samples)

    normal_noise = generate_random_normal(mean_val, std_dev, iteration)
    uniform_noise = generate_random_uniform(100, iteration)

    noisy_quadratic = add_normal_noise(normal_noise, quadratic_trend, num_samples)
    plot_data(quadratic_trend, noisy_quadratic, 'Quadratic Trend + Normal Noise')
    calculate_statistics(noisy_quadratic, 'Normal Noise Sample')

    noisy_linear = noisy_model(normal_noise, linear_trend, num_samples)
    plot_data(linear_trend, noisy_linear, 'Linear Trend + Uniform Noise')
    calculate_statistics(noisy_linear, 'Uniform Noise Sample')

    anomalies_quadratic = add_anomalies_normal(normal_noise, noisy_quadratic, anomaly_count, anomaly_factor)
    plot_data(quadratic_trend, anomalies_quadratic, 'Quadratic Trend + Normal Noise + Anomalies')
    calculate_statistics(anomalies_quadratic, 'Sample with Anomalies')

    anomalies_linear = trend_with_anomalies(linear_trend, noisy_linear, anomaly_count, anomaly_factor)
    plot_data(linear_trend, anomalies_linear, 'Linear Trend + Uniform Noise + Anomalies')
    calculate_statistics(anomalies_linear, 'Sample with Anomalies')

    usd_data = parse_data('https://www.oschadbank.ua/rates-archive', 'Oschadbank (USD).xls', 'Продаж')
    plot_data(usd_data, usd_data, 'USD Exchange Rate (2022, Oschadbank)')
    calculate_statistics(usd_data, 'USD Exchange Rate (2022, Oschadbank)')
```

```python
def linear_model(num_points):
    linear_data = np.zeros((num_points))
    for idx in range(num_points):
        linear_data[idx] = 0.0000005 * idx * idx
    return linear_data

def noisy_model(base_model, noise_data, num_points):
    result_model = np.zeros(num_points)
    for idx in range(num_points):
        result_model[idx] = base_model[idx] + noise_data[idx]
    return result_model

def trend_with_anomalies(base_model, noisy_data, anomaly_count, anomaly_factor):
    anomaly_samples = np.zeros((anomaly_count))
    anomaly_indices = np.random.randint(0, len(base_model), anomaly_count)

    for i in range(anomaly_count):
        anomaly_samples[i] = mth.ceil(np.random.randint(1, anomaly_factor))
        noisy_data[anomaly_indices[i]] = base_model[anomaly_indices[i]] + anomaly_factor * anomaly_samples[i]
    return noisy_data

def quadratic_model(num_points):
    quadratic_data = np.zeros(num_points)
    for idx in range(num_points):
        quadratic_data[idx] = 0.0000005 * idx * idx
    return quadratic_data

def add_normal_noise(noise_data, model_data, num_points):
    final_data = np.zeros(num_points)
    for idx in range(num_points):
        final_data[idx] = model_data[idx] + noise_data[idx]
    return final_data

def add_anomalies_normal(noise_data, model_data, anomaly_count, anomaly_factor):
    anomaly_samples = np.random.normal(0, anomaly_factor * 5, anomaly_count)
    anomaly_indices = np.random.randint(0, len(model_data), anomaly_count)

    for idx in range(anomaly_count):
        model_data[anomaly_indices[idx]] = model_data[anomaly_indices[idx]] + anomaly_samples[idx]
    return model_data
```

```python
import pandas as pd
import math as mth
import matplotlib.pyplot as plt

def parse_data(url, file_path, column_name):
    df = pd.read_excel(file_path)
    for _, col_values in df[[column_name]].items():
        print(col_values)
    data_array = np.zeros((len(col_values)))
    for idx in range(len(col_values)):
        data_array[idx] = col_values[idx]
    return data_array

```

```python
def generate_random_normal(mean_val, std_dev, num_samples):
    sample_data = np.random.normal(mean_val, std_dev, num_samples)
    mean_sample = np.mean(sample_data)
    var_sample = np.var(sample_data)
    stddev_sample = mth.sqrt(var_sample)

    print('Characteristics of Normal Distribution')
    print('Mean:', mean_sample)
    print('Variance:', var_sample)
    print('Standard Deviation:', stddev_sample)
    print('-------------------------')

    plt.hist(sample_data, bins=20, color="purple", alpha=0.6)
    plt.show()
    return sample_data
```

[15]
```python
def generate_random_uniform(size, low=0, high=1):
    uniform_samples = np.random.uniform(low, high, size)
    mean_uniform = np.mean(uniform_samples)
    var_uniform = np.var(uniform_samples)
    stddev_uniform = mth.sqrt(var_uniform)

    print('Characteristics of Uniform Distribution')
    print('Mean:', mean_uniform)
    print('Variance:', var_uniform)
    print('Standard Deviation:', stddev_uniform)
    print('--------------------')

    plt.hist(uniform_samples, bins=20, color="purple", alpha=0.6)
    plt.show()
    return uniform_samples
```

3.6. Аналіз результатів відлагодження та верифікації результатів роботи програми.

Результати відлагодження та тестування довели працездатність розробленого коду. Це підтверджується результатами розрахунків, які не суперечать теоретичним положенням.

Верифікація функціоналу програмного коду, порівняння отриманих результатів з технічними умовами завдання на лабораторну роботу доводять, що усі завдання виконані у повному обсязі.

IV. Висновки.

В результаті виконаної лабораторної роботи був розроблений програмний скрипт мовою Python який забезпечує аналіз різних характеристик та властивостей вибірок даних, включаючи синтез випадкових величин, моделювання тренду та визначення статистичних параметрів.

Синтезовані дані є корисними для тестування і верифікації алгоритмів їх обробки і будуть використані в подальших дослідженнях часових рядів.

Виконав: студент Олександр Головня