Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Кафедра теорії та технології програмування

Самостійна робота 2

зі «Статистичне моделювання в задачах штучного інтелекту»

Тема: «*Моделювання пуассонівського та вінерівського випадкових процесів*»

Груповий звіт

студентів 4 курсу групи ТТП-42

Чебана Богдана Володимировича

Ходакова Максима Олеговича

Бариша Артема Богдановича  
Сьомкіна Євгенія Сергійовича

Київ – 2024

**Зміст**

1—Формат виконання

2—Завдання 1

2.1. Змоделювати Пуассонівський потік з заданою інтенсивністю

2.2. Побудувати графіки реалізацій процесу

2.3. Побудувати гістограми розподілів

3—Завдання 2

3.1. Змоделювати неперервний вінерівський випадковий процес

3.2. Оцінити середнє значення та дисперсію за реалізаціями

3.3. Знайти емпіричний закон розподілу ймовірностей часу першого виходу

4—Опис виконаних завдань

5—Висновок

1. Формат виконання

Для виконання лабораторної роботи група студентів 4 курсу групи ТТП-42 обрала мову програмування **Python** завдяки її потужним можливостям у галузі статистичного моделювання та наукових обчислень. Python є однією з найпопулярніших мов у сфері штучного інтелекту та машинного навчання, що робить його ідеальним вибором для даного завдання.

**Основні Бібліотеки та Пакети**

Для симуляції пуассонівського та вінерівського випадкових процесів використовувалися наступні бібліотеки та пакети:

* **NumPy**: Основна бібліотека для роботи з багатовимірними масивами та матрицями, а також для виконання чисельних обчислень. NumPy забезпечує високу продуктивність завдяки використанню оптимізованих алгоритмів та підтримці векторизації операцій.
* **Matplotlib**: Бібліотека для створення статичних, анімованих та інтерактивних візуалізацій даних. Використовувалася для побудови графіків реалізацій процесів, гістограм, 3D-візуалізацій та анімацій.
* **SciPy**: Бібліотека для наукових обчислень, яка доповнює можливості NumPy. Використовувалася для статистичних аналізів, таких як лінійна регресія при оцінці фрактальної розмірності.
* **Seaborn**: Візуалізаційна бібліотека, побудована на основі Matplotlib, яка надає більш естетичні та інформативні графіки. Використовувалася для побудови гістограм з ядровою оцінкою густини (KDE).
* **mpl\_toolkits.mplot3d**: Пакет Matplotlib для створення 3D-графіків, що дозволило візуалізувати кілька реалізацій процесу одночасно в тривимірному просторі.
* **matplotlib.animation**: Модуль Matplotlib для створення анімацій, який використовувався для генерації GIF-файлів з анімованими траєкторіями процесів.
* **matplotlib.widgets**: Модуль для створення інтерактивних елементів, таких як слайдери та кнопки, що дозволило реалізувати інтерактивний аналіз часу першого виходу.
* **warnings**: Вбудований модуль Python для контролю над попередженнями, який використовувався для відключення непотрібних повідомлень під час виконання скриптів.

**Середовище Розробки**

Для розробки та виконання коду використовувалися наступні інструменти:

* **PyCharm**: Потужне середовище розробки (IDE) для Python, яке надає зручні інструменти для написання, налагодження та тестування коду. PyCharm забезпечує інтеграцію з системами контролю версій, такими як Git, а також підтримку різних плагінів для покращення продуктивності розробки.
* **Jupyter Notebook**: Інтерактивне середовище для виконання коду Python з можливістю створення документів, що містять живий код, рівняння, візуалізації та пояснювальний текст. Використовувався для експериментального аналізу даних та швидкої візуалізації результатів.

**Структура Коду**

Код лабораторної роботи організований у вигляді функціональних модулів, кожен з яких відповідає за певну частину завдання:

1. **Симуляція Випадкових Процесів**:
   * simulate\_poisson\_process: Функція для симуляції пуассонівського процесу з заданою інтенсивністю та кінцевим часом.
   * simulate\_wiener\_process: Функція для симуляції вінерівського (Броунівського) процесу з заданими параметрами часу та кількістю реалізацій.
2. **Аналіз та Статистики**:
   * compute\_statistics: Функція для обчислення середнього значення та дисперсії за всіма реалізаціями процесу.
   * find\_first\_exit\_times: Функція для визначення часу першого виходу процесу за заданий рівень.
   * plot\_event\_distributions: Функція для побудови гістограм розподілів часу появи подій та інтервалів між ними.
   * plot\_fractal\_dimension: Функція для оцінки та візуалізації фрактальної розмірності траєкторій процесу.
3. **Візуалізація Результатів**:
   * plot\_process\_realization: Функція для побудови графіку реалізації пуассонівського процесу.
   * plot\_sample\_paths: Функція для візуалізації зразкових траєкторій вінерівського процесу.
   * plot\_mean\_variance: Функція для візуалізації середнього значення та дисперсії процесу.
   * plot\_first\_exit\_distribution: Функція для візуалізації розподілу часу першого виходу.
   * animate\_sample\_paths та animate\_wiener\_process: Функції для створення анімацій реалізацій процесів.
   * plot\_3d\_process\_realization: Функція для 3D-візуалізації кількох реалізацій процесу.
   * plot\_real\_time\_process: Функція для побудови анімації процесу в реальному часі.
   * interactive\_rate\_analysis та interactive\_first\_passage\_time: Функції для створення інтерактивних графіків з використанням слайдерів.

**Організація Роботи**

Процес виконання лабораторної роботи був організований наступним чином:

1. **Підготовка Середовища**:
   * Встановлення необхідних бібліотек за допомогою менеджера пакетів pip:

bash

Копировать код

pip install numpy matplotlib scipy seaborn pillow

* + Налаштування середовища розробки в PyCharm або Jupyter Notebook для інтерактивної роботи з кодом та візуалізаціями.

1. **Розробка Функціональних Модулів**:
   * Створення окремих функцій для симуляції, аналізу та візуалізації процесів.
   * Забезпечення модульності коду для полегшення тестування та повторного використання.
2. **Тестування та Валідація**:
   * Виконання симуляцій з різними параметрами для перевірки коректності реалізації процесів.
   * Порівняння емпіричних результатів з теоретичними очікуваннями для підтвердження правильності моделювання.
3. **Візуалізація та Документація**:
   * Побудова різних типів графіків для наочного представлення результатів.
   * Створення анімацій для демонстрації динаміки процесів.
   * Документування коду та результатів у звіті для кращого розуміння та аналізу.
4. **Оптимізація та Покращення**:
   * Зменшення кількості кадрів у анімаціях для підвищення продуктивності.
   * Додавання інтерактивних елементів для гнучкого аналізу процесів.
   * Вдосконалення візуалізацій для більш інформативного представлення даних.

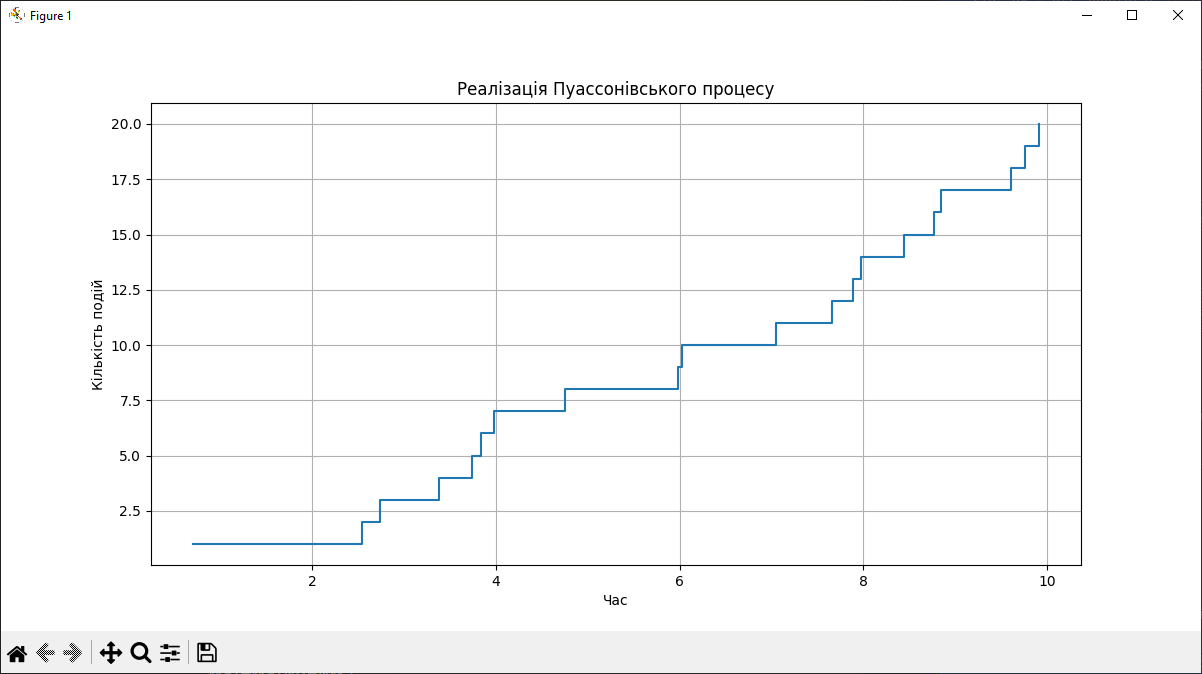
**Використані Алгоритми та Методи**

1. **Симуляція Пуассонівського Процесу**:
   * Генерація інтервалів між подіями за експоненціальним розподілом з параметром λ.
   * Накопичення інтервалів для визначення часу подій до моменту T.
2. **Симуляція Вінерівського Процесу**:
   * Генерація змінних dW за нормальним розподілом з математичним сподіванням 0 та дисперсією dt.
   * Кумулятивна сума змінних dW для побудови траєкторій процесу W(t).
3. **Аналіз Статистик**:
   * Обчислення середнього значення та дисперсії за всіма реалізаціями для кожного часовго кроку.
   * Визначення часу першого виходу за заданий рівень для кожної реалізації.
   * Оцінка фрактальної розмірності траєкторій процесу за методом коробок (Box-Counting).
4. **Візуалізація Даних**:
   * Побудова графіків кроків для пуассонівського процесу.
   * Створення 3D-графіків для візуалізації кількох реалізацій процесу.
   * Використання гістограм та KDE для аналізу розподілів часу подій.
   * Генерація анімацій для демонстрації динаміки процесів у часі.
   * Реалізація інтерактивних графіків з використанням слайдерів для гнучкого аналізу.

**Висновок**

Формат виконання лабораторної роботи був ретельно спланований та реалізований з використанням сучасних інструментів та бібліотек Python. Це дозволило ефективно симулювати та аналізувати пуассонівський та вінерівський випадкові процеси, а також представити результати у наочній та інформативній формі. Завдяки модульній організації коду та інтерактивним елементам, робота над лабораторною роботою була структурованою та зручною для подальшого розширення та аналізу.

**Скріншоти:**



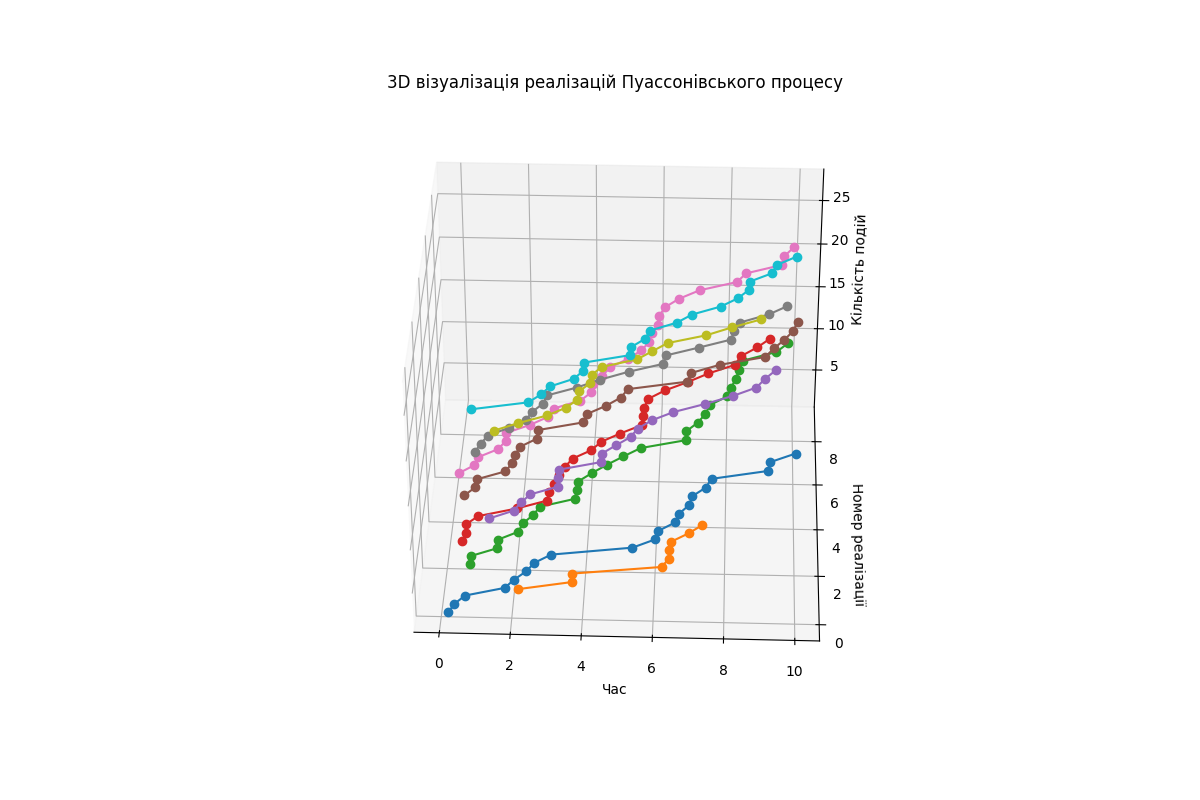
Це графік, який представляє реалізацію Пуассонівського процесу. На графіку по осі X відображено час (від 0 до 10 одиниць), а по осі Y — кількість подій, які відбулися в кожен момент часу. Лінія графіка має вигляд східчастої функції, яка демонструє зростання кількості подій у міру просування часу.

Кожен крок на графіку вказує на нову подію, яка відбувається у випадковий момент. Інтенсивність подій не є постійною, що відповідає випадковому характеру Пуассонівського процесу — події можуть виникати ближче або далі одна від одної залежно від того, як випадково генеруються інтервали між подіями.

Основні характеристики графіка:

* **Час** (вісь X): показує момент часу, у який відбуваються події.
* **Кількість подій** (вісь Y): відображає накопичену кількість подій, що з'явилися до конкретного моменту часу.
* **Східчастий вигляд**: свідчить про те, що процес моделює дискретні події, і з кожною подією кількість зростає на одиницю.

Заголовок графіка — "Реалізація Пуассонівського процесу" — вказує на те, що це результат однієї симуляції Пуассонівського процесу з певною інтенсивністю подій.



Це графік 3D візуалізації реалізацій Пуассонівського процесу. Основна ідея такого графіка — представити декілька симуляцій Пуассонівського процесу, показуючи, як вони виглядають на тривимірній площині.

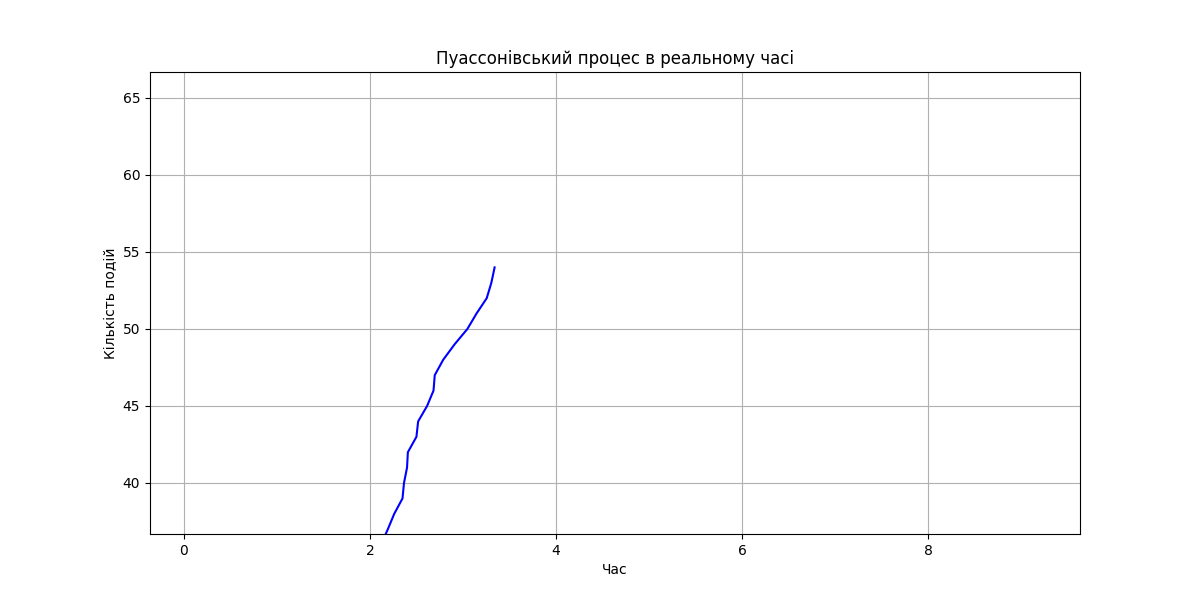
**Опис осей графіка**:

* **Вісь X (час)**: показує момент часу, у який відбуваються події. Час представлений від 0 до 10 одиниць.
* **Вісь Y (номер реалізації)**: кожна з симульованих реалізацій процесу відображається окремо, починаючи з нуля. Це дозволяє розрізняти різні симуляції і наочно бачити, як вони відрізняються одна від одної.
* **Вісь Z (кількість подій)**: відображає кількість подій, які накопичуються у кожній конкретній симуляції до відповідного моменту часу.

**Основні характеристики графіка**:

* **Кольорові лінії з точками** представляють декілька реалізацій Пуассонівського процесу, де кожна лінія відповідає окремій симуляції.
* **Східчасті вигини** показують, що кожна подія збільшує кількість на одиницю, але інтервали між подіями відрізняються, що відображає випадковий характер процесу.
* **Графік дає змогу порівнювати різні реалізації**, показуючи, що хоча кожна реалізація має схожий загальний напрямок (кількість подій зростає з часом), але конкретні моменти, коли відбуваються події, можуть сильно варіюватися.

Ця 3D візуалізація дозволяє краще зрозуміти поведінку випадкових процесів і їхню варіативність, демонструючи, як однаковий процес із певною інтенсивністю може виглядати по-різному через випадковий характер часу між подіями.



Це графік, який представляє симуляцію Пуассонівського процесу в реальному часі. Він показує накопичення подій у часі.

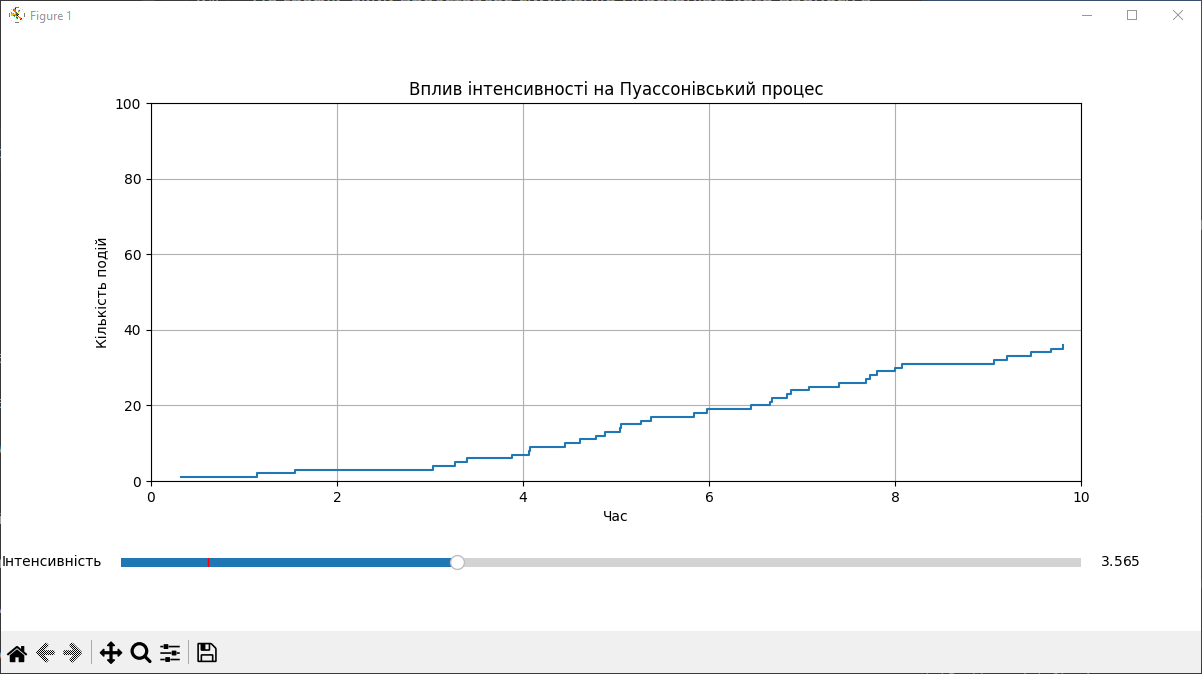
**Опис осей графіка**:

* **Вісь X (час)**: показує час, у який відбуваються події. На графіку відображений діапазон часу від 0 до 10 одиниць.
* **Вісь Y (кількість подій)**: показує кількість подій, які накопичилися до кожного моменту часу. В даному випадку, кількість подій починається приблизно з 40 і продовжує зростати.

**Основні характеристики графіка**:

* **Крива лінія (синя)** представляє кумулятивну кількість подій у процесі з плином часу. Лінія має зиґзаґоподібний характер, що свідчить про випадковість у часі між подіями.
* **Наростання лінії**: показує, що кількість подій поступово збільшується. Події можуть відбуватися з різними інтервалами часу — іноді вони стаються частіше, а іноді рідше, що відповідає випадковій природі Пуассонівського процесу.

Графік демонструє, як процес розвивається в реальному часі: кожна нова подія збільшує кумулятивну кількість на один. Він дозволяє візуально спостерігати зміни в інтенсивності появи подій.



Це графік, що представляє вплив інтенсивності на Пуассонівський процес. Графік дозволяє інтерпретувати, як зміна інтенсивності процесу впливає на кількість подій, які накопичуються у часі.

**Опис осей графіка**:

* **Вісь X (час)**: показує проміжок часу, протягом якого відбуваються події. На графіку представлений діапазон від 0 до 10 одиниць часу.
* **Вісь Y (кількість подій)**: відображає кількість подій, які накопичилися до кожного моменту часу, що підказує динаміку процесу залежно від часу.

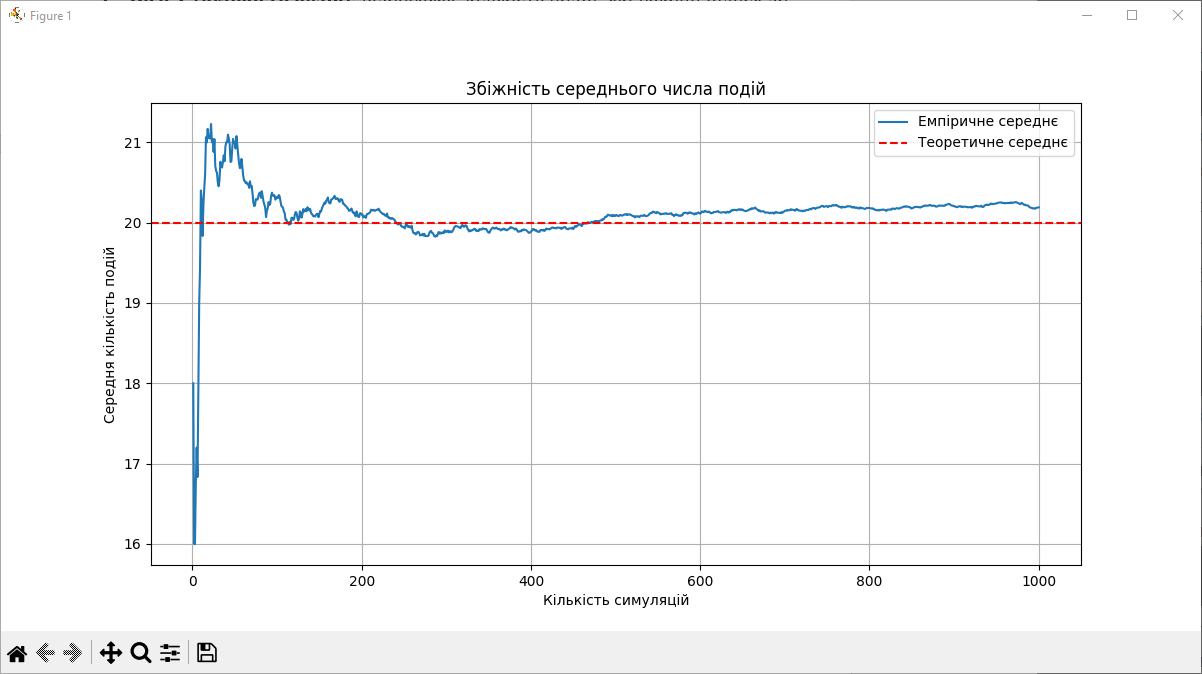
**Основні характеристики графіка**:

* **Крива лінія (східчаста форма)**: показує кількість подій, які накопичуються з плином часу. Оскільки Пуассонівський процес має випадковий характер, події відбуваються у випадкові моменти, що призводить до нерівномірного наростання графіка.
* **Слайдер інтенсивності**: під графіком знаходиться інтерактивний елемент — слайдер, який дозволяє змінювати інтенсивність процесу (інтенсивність відображена справа). Слайдер дозволяє контролювати частоту появи подій: при збільшенні інтенсивності події відбуваються частіше, що відображається на графіку у вигляді більш частих сходинок.

**Як впливає зміна інтенсивності**:

* При збільшенні значення інтенсивності, кількість подій збільшується, і відповідно графік стає крутішим.
* При зменшенні інтенсивності події відбуваються рідше, і графік має більш пологий вигляд, з довшими горизонтальними сегментами.

Цей графік дозволяє наочно продемонструвати вплив інтенсивності на характер Пуассонівського процесу, що особливо корисно для розуміння того, як різні параметри можуть змінити поведінку випадкового процесу.



Це графік, який показує збіжність середнього числа подій у Пуассонівському процесі. Він відображає, як середнє число подій, отримане в результаті симуляцій, наближається до теоретичного значення, з плином кількості симуляцій.

**Опис осей графіка**:

* **Вісь X (кількість симуляцій)**: показує кількість симуляцій Пуассонівського процесу, які були проведені (від 0 до 1000).
* **Вісь Y (середня кількість подій)**: відображає середню кількість подій, які відбувалися у процесі до кожного моменту часу, коли виконувалися симуляції.

**Основні елементи графіка**:

* **Синя лінія (емпіричне середнє)**: представляє середнє число подій, яке було отримане за певну кількість симуляцій. На початку симуляцій емпіричне середнє значно варіюється, проте зі збільшенням кількості симуляцій воно стабілізується і починає наближатися до теоретичного значення.
* **Червона пунктирна лінія (теоретичне середнє)**: представляє теоретичне значення середньої кількості подій для Пуассонівського процесу з заданою інтенсивністю. Це значення є сталою, і емпіричне середнє поступово до нього наближається.

**Аналіз графіка**:

* **На початку симуляцій** (малий діапазон X) емпіричне середнє значно варіюється, оскільки мало реалізацій процесу не дозволяють отримати надійні статистичні дані.
* **Зі збільшенням кількості симуляцій** графік починає стабілізуватися, і емпіричне середнє починає збігатися з теоретичним значенням, що ілюструє закон великих чисел. Це означає, що при достатній кількості симуляцій середнє значення емпіричних даних наближається до теоретичного значення.

Графік дозволяє зрозуміти важливість кількості симуляцій для оцінки характеристик випадкових процесів. Чим більше симуляцій проводиться, тим точніше можна оцінити середнє значення для випадкового процесу, яке відповідає теоретичним передбаченням.

**Завдання Номер 2**

Це графік 3D візуалізації реалізацій Вінерівського процесу. Він демонструє п'ять різних симуляцій (реалізацій) Вінерівського процесу та їхню динаміку в часі.

**Опис осей графіка**:

* **Вісь X (час)**: відображає проміжок часу від 0 до 1, протягом якого розвивається кожна з реалізацій Вінерівського процесу.
* **Вісь Y (номер реалізації)**: позначає номер кожної з п'яти симуляцій процесу, що дає можливість легко розрізняти між собою різні реалізації.
* **Вісь Z (значення)**: відображає значення Вінерівського процесу для кожного моменту часу. Оскільки Вінерівський процес є випадковим, значення можуть коливатися як в позитивний, так і в негативний бік.

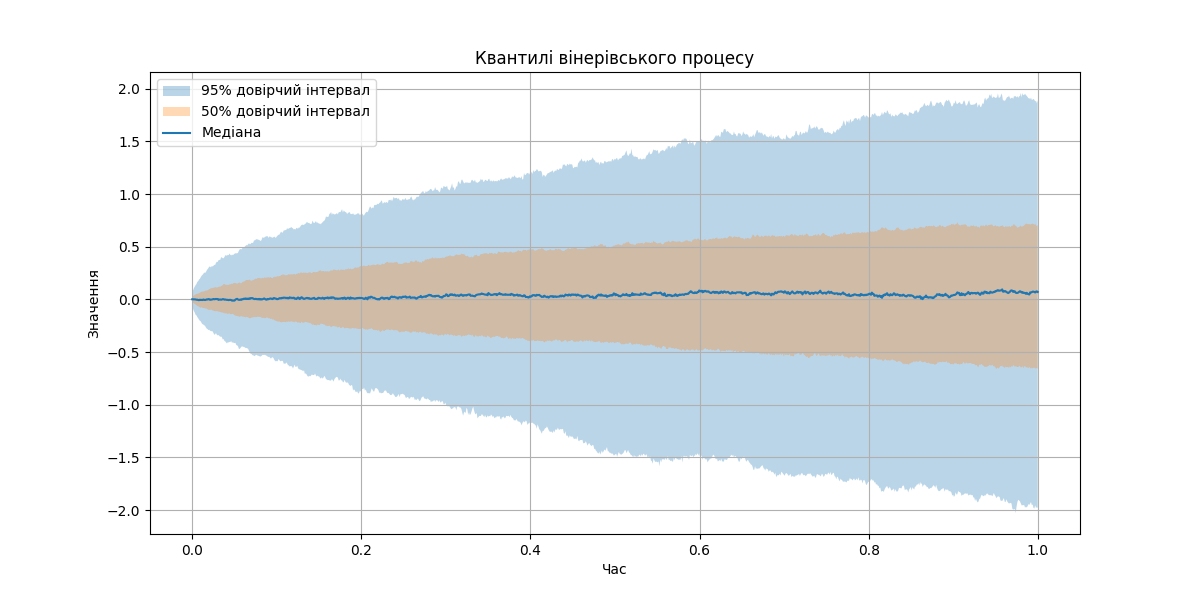
**Основні характеристики графіка**:

* **Кольорові лінії**: кожна кольорова лінія на графіку представляє одну реалізацію Вінерівського процесу. Лінії мають різний колір, що дозволяє легко відрізнити окремі реалізації.
* **Коливання значень**: Вінерівський процес є прикладом броунівського руху, і кожна лінія показує випадкові коливання у значеннях з часом. Значення можуть рухатись як вгору, так і вниз, демонструючи природу випадкового процесу.

**Аналіз графіка**:

* **Стійке коливання**: усі реалізації мають стійке коливання значень навколо нуля, що є характерним для Вінерівського процесу. Це означає, що середнє значення залишаєтся близьким до нуля, проте у кожний момент часу значення можуть змінюватись випадковим чином.
* **Різна поведінка реалізацій**: кожна реалізація відрізняється своєю поведінкою, проте всі вони мають схожу загальну структуру випадкового коливання. Це підкреслює варіативність такого типу процесів.

Цей графік дозволяє наочно побачити випадковість Вінерівського процесу, показуючи різні можливі траєкторії, які може приймати процес з однаковими початковими умовами. Вінерівський процес часто використовується для моделювання випадкових явищ у фінансовій сфері, фізиці, та інших науках, де важливо враховувати непередбачувані зміни.



Це графік квантилів Вінерівського процесу, який показує поведінку випадкових реалізацій процесу у вигляді довірчих інтервалів. Він дозволяє зрозуміти, як варіюються значення процесу протягом часу і яка є ймовірність того, що значення опиниться у певному діапазоні.

**Опис осей графіка**:

* **Вісь X (час)**: відображає проміжок часу від 0 до 1, протягом якого відбувається розвиток Вінерівського процесу.
* **Вісь Y (значення)**: показує значення процесу, яке може бути як позитивним, так і негативним, що відображає випадковий характер Вінерівського процесу.

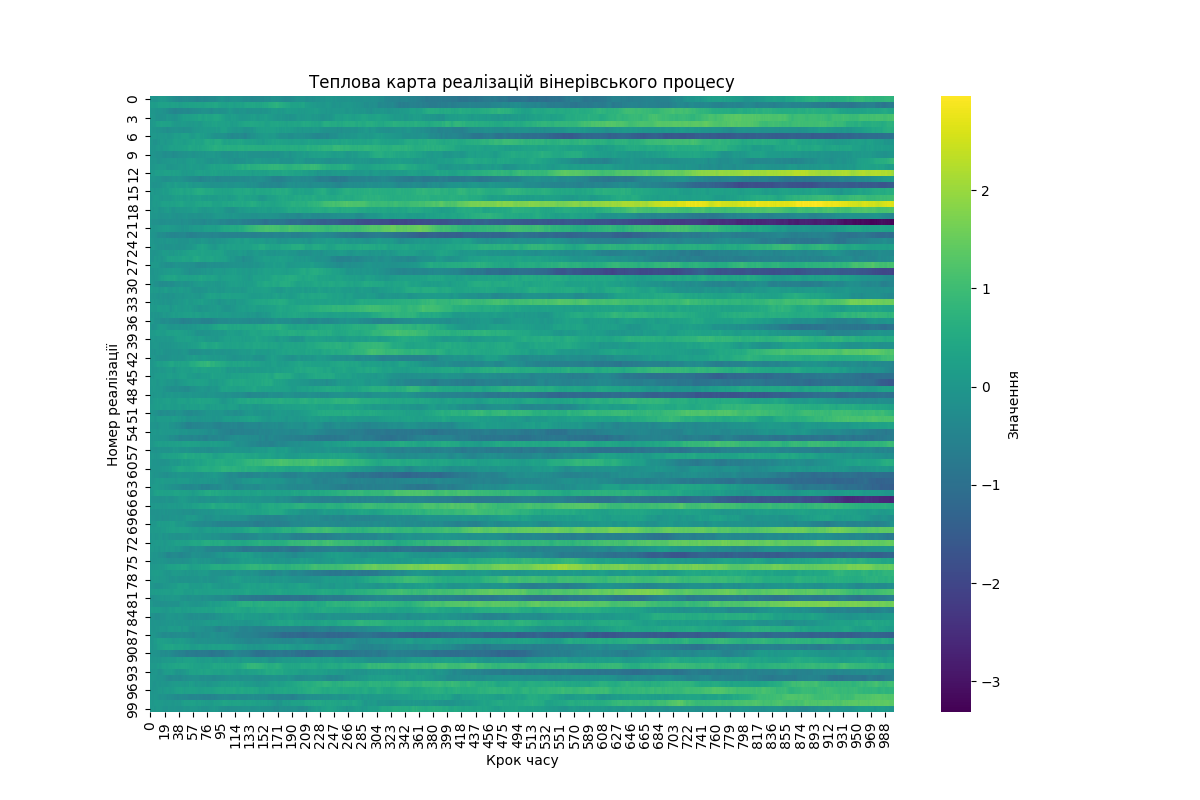
**Основні характеристики графіка**:

* **Синя лінія (Медіана)**: представляє медіанне значення Вінерівського процесу в кожен момент часу. Оскільки медіана є центральним значенням, вона показує середнє місце розташування всіх реалізацій процесу. Для Вінерівського процесу медіана знаходиться на рівні нуля протягом усього часу.
* **Синя область (95% довірчий інтервал)**: представляє діапазон, у якому знаходиться 95% всіх реалізацій Вінерівського процесу в кожен момент часу. Це ширша область, яка показує всю варіативність процесу з високою ймовірністю.
* **Коричнева область (50% довірчий інтервал)**: представляє діапазон, у якому знаходиться 50% всіх реалізацій процесу в кожен момент часу. Це вужча область, яка відображає основний діапазон значень, у якому відбуваються коливання процесу, і дає уявлення про середню варіативність.

**Аналіз графіка**:

* **Симетрія довкола медіани**: графік показує, що Вінерівський процес є симетричним відносно нуля. Це відповідає теоретичним властивостям процесу, де очікуване значення в будь-який момент часу є нуль.
* **Зростання дисперсії**: ширина довірчих інтервалів збільшується з плином часу, що означає, що невизначеність і варіативність значень Вінерівського процесу зростають з часом. Це характерна властивість Вінерівського процесу, де дисперсія пропорційна часу.

Цей графік дозволяє наочно побачити, як зростає варіативність процесу, і надає інформацію про ймовірність знаходження значень процесу в певних межах. Такий підхід часто використовується для аналізу випадкових процесів, коли важливо зрозуміти не тільки середню поведінку, але й ймовірні відхилення.



Це графік теплової карти реалізацій Вінерівського процесу. Така теплова карта надає візуальне уявлення про те, як значення процесу змінюються в часі для різних реалізацій.

**Опис осей графіка**:

* **Вісь X (крок часу)**: показує часові кроки, тобто момент часу, у який аналізуються значення Вінерівського процесу. Відображається від 0 до 1000 кроків часу.
* **Вісь Y (номер реалізації)**: представляє номер кожної реалізації процесу. На графіку показано 100 реалізацій процесу, що дозволяє візуально оцінити їхню поведінку та відмінності.

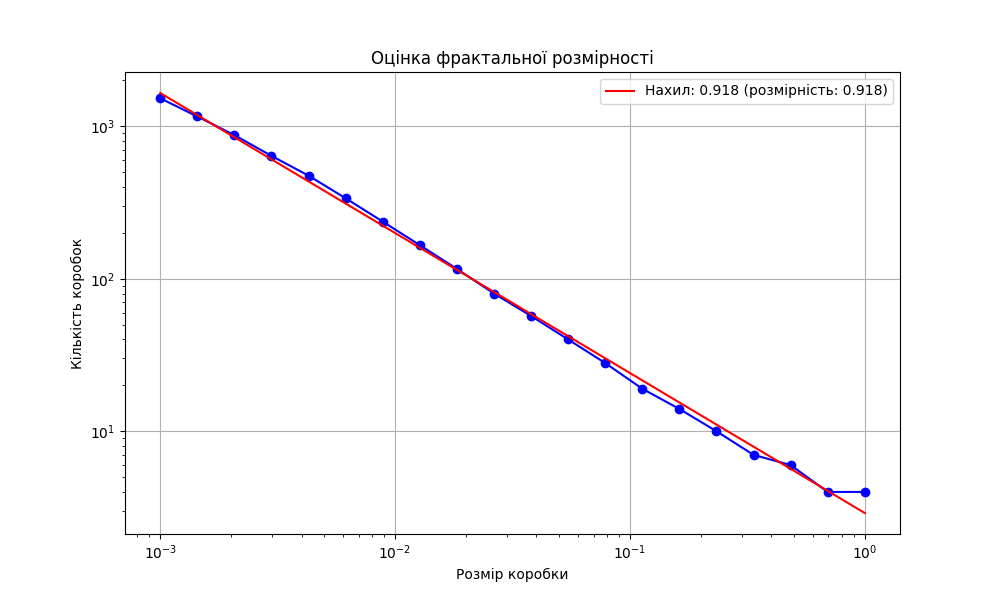
**Основні характеристики теплової карти**:

* **Колірна шкала**: праворуч від графіка знаходиться шкала значень, де кольори відповідають різним значенням процесу:
  + **Темніший колір** (фіолетовий) означає негативні значення процесу.
  + **Світліший колір** (жовтий) відповідає позитивним значенням.
  + **Зелений колір** відповідає значенням, близьким до нуля.
* **Теплова карта**: на графіку кожен рядок відповідає одній реалізації Вінерівського процесу, а стовпці відображають значення процесу у певний момент часу. По зміні кольорів можна побачити, як значення процесу змінюються в часі для кожної реалізації.

**Аналіз графіка**:

* **Зміни кольорів уздовж осі X** демонструють варіативність значень процесу у часі. Ми бачимо, що значення можуть варіюватися досить значно, і кожна реалізація має свою унікальну траєкторію.
* **Симетричність кольорів** навколо середніх значень часу свідчить про властивості Вінерівського процесу, де очікуване значення залишається близьким до нуля, проте значення можуть бути як позитивними, так і негативними.
* **Поступова зміна кольору вздовж реалізацій**: деякі реалізації зберігають більш стійкі значення (однаковий колір), тоді як інші часто змінюють колір, що вказує на різну поведінку і коливання у значеннях.

Ця теплова карта надає корисну інформацію про загальну поведінку великої кількості реалізацій Вінерівського процесу, показуючи як відбуваються зміни в кожній з них. Вона є ефективним інструментом для швидкого аналізу поведінки багатьох траєкторій одночасно.



Це графік, який демонструє оцінку фрактальної розмірності для Вінерівського процесу за допомогою методу "коробок". Він показує, як кількість коробок, необхідних для покриття траєкторії процесу, змінюється в залежності від розміру коробки.

**Опис осей графіка**:

* **Вісь X (розмір коробки)**: показує розмір коробки, який використовувався для покриття траєкторії. Масштаб логарифмічний і варіюється від 10−310^{-3}10−3 до 10010^0100, що дозволяє відстежувати зміни на різних масштабах.
* **Вісь Y (кількість коробок)**: відображає кількість коробок, які потрібні для покриття всієї траєкторії при певному розмірі коробки. Також використовується логарифмічний масштаб, щоб більш наочно показати залежність для великих і малих значень.

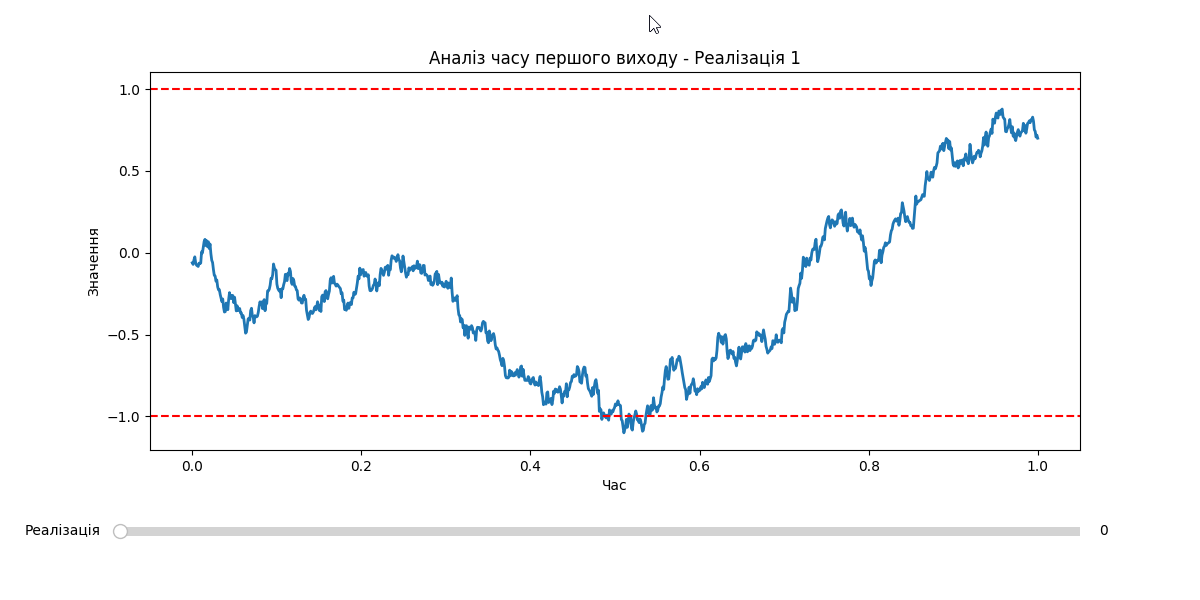
**Основні елементи графіка**:

* **Сині точки**: кожна точка представляє кількість коробок для певного розміру коробки. Це результати експериментальної оцінки.
* **Червона лінія**: це регресійна лінія, яка використовується для оцінки нахилу графіка. Нахил використовується для визначення фрактальної розмірності.

**Аналіз графіка**:

* **Нахил регресійної лінії**: нахил червоної лінії дорівнює 0.918, що вказує на фрактальну розмірність процесу, яка також дорівнює 0.918. Це свідчить про те, наскільки складна траєкторія процесу з точки зору його фрактальної структури.
* **Пряма лінія на логарифмічному графіку**: на графіку є приблизно лінійна залежність на логарифмічних осях, що підтверджує, що Вінерівський процес має фрактальну природу. Фрактальна розмірність менша за 2, що характерно для траєкторій, які є нерегулярними, але не заповнюють площину.

Цей графік дозволяє оцінити фрактальну розмірність, що є важливою характеристикою для випадкових процесів, таких як Вінерівський процес. Фрактальна розмірність вказує на те, наскільки "звивистою" або "нерегулярною" є траєкторія процесу, і дозволяє краще зрозуміти, як змінюється структура траєкторії при різних масштабах.



Це графік, який демонструє аналіз часу першого виходу для однієї реалізації Вінерівського процесу. Графік показує, як змінюється значення процесу з часом і коли саме значення досягає визначених порогів.

**Опис осей графіка**:

* **Вісь X (час)**: відображає проміжок часу від 0 до 1, протягом якого розвивається реалізація Вінерівського процесу.
* **Вісь Y (значення)**: показує значення процесу у певний момент часу. Значення можуть варіюватися як в позитивну, так і в негативну сторону.

**Основні елементи графіка**:

* **Синя лінія**: показує зміну значення Вінерівського процесу в часі для однієї конкретної реалізації.
* **Червоні пунктирні лінії**: відображають порогові рівні (+1 і -1), при досягненні яких реалізація процесу вважається "вийшовшою" за визначений рівень. Ці лінії допомагають визначити, коли процес вперше виходить за ці межі.

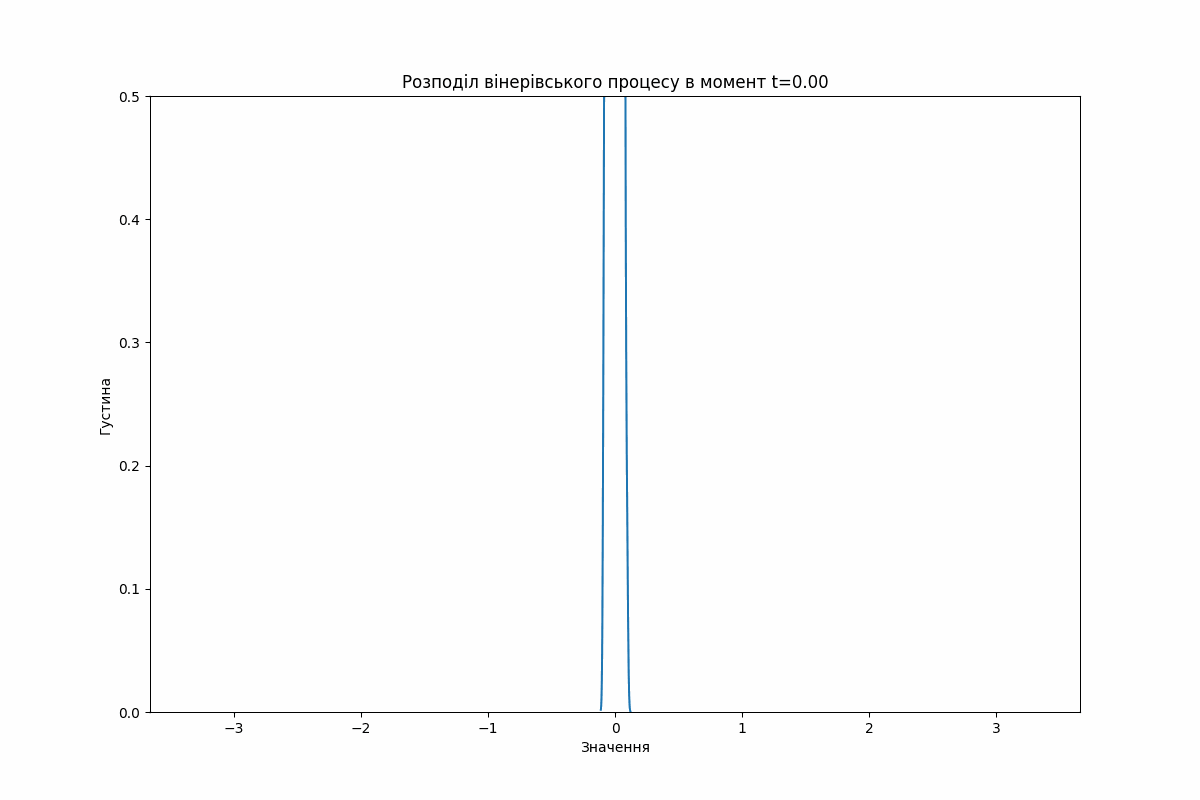
**Аналіз графіка**:

* **Перший вихід за поріг**: на графіку видно, що значення процесу протягом часу коливається навколо нуля, але не виходить за порогові значення (+1 або -1) протягом усієї симуляції. Це означає, що для даної реалізації процесу не відбулося виходу за визначені порогові рівні за весь період спостереження.
* **Випадковий характер коливань**: лінія процесу демонструє характерні для Вінерівського процесу випадкові коливання, які можуть мати як зростання, так і падіння. Значення процесу періодично підходять близько до порогових рівнів, але не перевищують їх.

**Інтерактивність**:

* **Слайдер (реалізація)**: у нижній частині графіка є слайдер, який дозволяє обирати різні реалізації процесу для перегляду. Таким чином можна аналізувати різні траєкторії і визначати, коли саме кожна з них досягає порогових значень.

Цей графік є корисним інструментом для аналізу часу першого виходу випадкового процесу за визначений поріг. У випадку фінансових даних або інших прикладних задач, цей аналіз може бути використаний для визначення ризиків або ймовірності події, коли значення виходить за критичні межі.



Це графік, який демонструє розподіл Вінерівського процесу в момент часу t=0t = 0t=0. Графік показує, як значення процесу розподілені в цей конкретний момент часу.

**Опис осей графіка**:

* **Вісь X (значення)**: показує значення, які може набувати процес. В даному випадку, значення варіюються від -3 до 3.
* **Вісь Y (густина)**: відображає густину ймовірності, що процес має певне значення в момент часу t=0t = 0t=0.

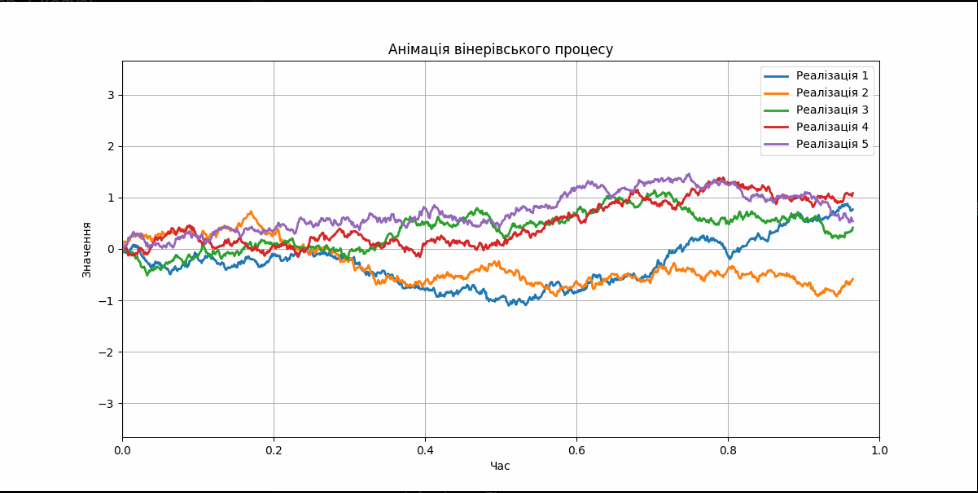
**Основні елементи графіка**:

* **Висока пікова лінія біля нуля**: графік демонструє дуже високу пікову густину в значенні 0, що є характерним для початку Вінерівського процесу. У момент часу t=0t = 0t=0 всі реалізації процесу мають однакове значення (0), оскільки це початкова точка для кожної траєкторії Вінерівського процесу.

**Аналіз графіка**:

* **Розподіл у момент t=0t = 0t=0**: оскільки це початок процесу, всі реалізації Вінерівського процесу починаються з однієї точки, що означає, що значення густини сконцентроване виключно в точці 0.
* **Форма розподілу**: форма розподілу тут показує, що на початку процесу немає жодних відхилень від нуля, тому весь розподіл виглядає як тонка пікова лінія.

Цей графік є ілюстрацією того, що в момент часу t=0t = 0t=0 Вінерівський процес завжди починається з нульового значення, і немає варіативності. В подальші моменти часу розподіл значень стає ширшим і демонструє нормальний розподіл зі зростанням часу, що відображає дифузійний характер Вінерівського процесу.



Це початковий кадр анімації, який демонструє реалізації Вінерівського процесу в часовому інтервалі від 0 до 1. На цьому графіку показано п'ять різних реалізацій процесу, що ідентифікуються різними кольорами.

**Опис осей графіка**:

* **Вісь X (час)**: відображає час від 0 до 1, що вказує на те, що розглядається процес протягом одиничного часового інтервалу.
* **Вісь Y (значення)**: показує значення випадкового процесу в певний момент часу. Значення варіюються в межах від -3 до 3.

**Основні елементи графіка**:

* **Кольорові лінії**: на графіку повинні з'являтися п'ять ліній, кожна з яких представляє одну реалізацію Вінерівського процесу. Кожна з них має свій колір, і ці кольори позначені в легенді у правому верхньому кутку графіка.
* **Легенда**: показує, які кольори відповідають яким реалізаціям (від 1 до 5).

**Аналіз графіка**:

* **Відсутність видимих ліній**: на цьому кадрі графік виглядає порожнім. Це свідчить про те, що процес тільки починається, і на цей момент часу ще не відбулося помітних змін у значеннях реалізацій. Всі реалізації починаються з початкового значення, яке дорівнює 0, тому лінії ще не видимі або ще не мають значущих відхилень.
* **Розвиток процесу з часом**: під час перегляду анімації кожна з реалізацій буде змінюватися в міру збільшення часу, демонструючи випадковий характер Вінерівського процесу, який має властивості дифузії. Значення будуть відхилятися як у позитивний, так і в негативний напрямок.

Цей графік є частиною анімації, яка надає візуальне уявлення про еволюцію Вінерівського процесу з часом. Він корисний для розуміння динаміки випадкових процесів та їхньої поведінки в різних реалізаціях, що може бути важливим у застосуваннях, таких як фінансовий аналіз або моделювання фізичних процесів.

**2. Завдання 1**

**2.1. Змоделювати Пуассонівський потік з заданою інтенсивністю**

**Мета:**  
Симулювати пуассонівський процес з заданою інтенсивністю λ та побудувати графіки реалізацій процесу.

**Теоретичне підґрунтя:**  
Пуассонівський процес використовується для моделювання подій, які відбуваються незалежно одна від одної з постійною середньою частотою λ. Час між подіями слідує експоненціальному розподілу з параметром λ.

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import poisson

from mpl\_toolkits.mplot3d import Axes3D

import matplotlib.animation as animation

from matplotlib.widgets import Slider, Button

from IPython.display import display, clear\_output

import time

def simulate\_poisson\_process(lambda\_rate, T):

"""

Симулює Пуассонівський процес.

:param lambda\_rate: Інтенсивність процесу

:param T: Кінцевий час симуляції

:return: Масив часів подій

"""

t = 0

events = []

while t < T:

t += np.random.exponential(1 / lambda\_rate)

if t < T:

events.append(t)

return np.array(events)

def plot\_process\_realization(events, T):

"""

Будує графік реалізації процесу.

:param events: Масив часів подій

:param T: Кінцевий час симуляції

"""

plt.figure(figsize=(12, 6))

plt.step(events, range(1, len(events) + 1), where='post')

plt.xlabel('Час')

plt.ylabel('Кількість подій')

plt.title('Реалізація Пуассонівського процесу')

plt.grid(True)

plt.show()

**Пояснення коду:**

1. **Симуляція Пуассонівського процесу:**
   * Використовується функція simulate\_poisson\_process, яка генерує часи подій до моменту T.
   * Час між подіями генерується за експоненціальним розподілом з параметром λ.
2. **Візуалізація реалізації процесу:**
   * Функція plot\_process\_realization будує графік кроків (step), де по осі X відображається час, а по осі Y – кількість подій.

**Виконання коду:**

# # Параметри симуляції

# lambda\_rate = 2 # Інтенсивність процесу

# T = 10 # Кінцевий час симуляції

# # Базова симуляція та візуалізація

# events = simulate\_poisson\_process(lambda\_rate, T)

# plot\_process\_realization(events, T)

**Результат:**

На графіку видно кроки, які представляють кількість подій з часом. Інтенсивність λ=2 означає, що в середньому відбувається 2 події на одиницю часу.

### 2.2. Побудувати графіки реалізацій процесу

**Мета:**  
Візуалізувати декілька реалізацій пуассонівського процесу для порівняння їхньої поведінки.

**Код програми:**

#### def plot\_3d\_process\_realization(lambda\_rate, T, num\_realizations=10):

#### """

#### Будує 3D візуалізацію декількох реалізацій процесу.

#### :param lambda\_rate: Інтенсивність процесу

#### :param T: Кінцевий час симуляції

#### :param num\_realizations: Кількість реалізацій для візуалізації

#### """

#### fig = plt.figure(figsize=(12, 8))

#### ax = fig.add\_subplot(111, projection='3d')

#### for i in range(num\_realizations):

#### events = simulate\_poisson\_process(lambda\_rate, T)

#### x = events

#### y = np.ones\_like(events) \* i

#### z = np.arange(1, len(events) + 1)

#### ax.plot(x, y, z, marker='o')

#### ax.set\_xlabel('Час')

#### ax.set\_ylabel('Номер реалізації')

#### ax.set\_zlabel('Кількість подій')

#### ax.set\_title('3D візуалізація реалізацій Пуассонівського процесу')

#### plt.show()

**Пояснення коду:**

* Функція plot\_3d\_process\_realization генерує та відображає декілька (наприклад, 10) реалізацій пуассонівського процесу в 3D просторі.
* По осі X відображається час подій, по осі Y – номер реалізації, а по осі Z – кількість подій.

**Виконання коду:**

3D графік показує різні траєкторії пуассонівського процесу, що дозволяє порівняти кількість подій у різних реалізаціях з часом.

### 2.3. Побудувати гістограми розподілів

**Мета:**  
Побудувати гістограми розподілів часу появи заданої події (перша, друга, n-та), інтервалу між подіями та появи рівно n подій.

Код програми:

def plot\_event\_distributions(lambda\_rate, T, num\_simulations=1000, level=1):

"""

Будує гістограми розподілів часу появи заданої події, інтервалу між подіями та появи рівно n подій.

:param lambda\_rate: Інтенсивність процесу

:param T: Кінцевий час симуляції

:param num\_simulations: Кількість симуляцій

:param level: Номер події для аналізу часу появи

"""

first\_event\_times = []

inter\_event\_intervals = []

exact\_n\_events = 0

for \_ in range(num\_simulations):

events = simulate\_poisson\_process(lambda\_rate, T)

if len(events) >= level:

first\_event\_times.append(events[level - 1])

if len(events) >= 2:

inter\_event\_intervals.extend(np.diff(events))

if len(events) == level:

exact\_n\_events += 1

# Гістограма часу появи n-ї події

plt.figure(figsize=(12, 6))

sns.histplot(first\_event\_times, bins=30, kde=True, stat="density")

plt.xlabel(f'Час появи {level}-ї події')

plt.ylabel('Густина ймовірності')

plt.title(f'Розподіл часу появи {level}-ї події')

plt.grid(True)

plt.show()

# Гістограма інтервалів між подіями

plt.figure(figsize=(12, 6))

sns.histplot(inter\_event\_intervals, bins=30, kde=True, stat="density")

plt.xlabel('Інтервал між подіями')

plt.ylabel('Густина ймовірності')

plt.title('Розподіл інтервалів між подіями')

plt.grid(True)

plt.show()

# Гістограма появи рівно n подій

plt.figure(figsize=(12, 6))

sns.histplot([1 if len(simulate\_poisson\_process(lambda\_rate, T)) == level else 0 for \_ in range(num\_simulations)],

bins=2, kde=False, stat="probability")

plt.xlabel(f'Поява рівно {level} подій')

plt.ylabel('Ймовірність')

plt.title(f'Ймовірність появи рівно {level} подій')

plt.xticks([0, 1], ['Ні', 'Так'])

plt.grid(True)

plt.show()

print(f"Ймовірність появи рівно {level} подій: {exact\_n\_events / num\_simulations:.4f}")

**Пояснення коду:**

Функція plot\_event\_distributions виконує num\_simulations симуляцій пуассонівського процесу.

Збирає дані для:

Часу появи заданої події (перша, друга, n-та).

Інтервалів між подіями.

Кількості реалізацій, де з'явилося рівно n подій.

Побудова гістограм для кожного з розподілів з ядровою оцінкою густини (KDE).

**Виконання коду:**

# Побудова гістограм розподілів

plot\_event\_distributions(lambda\_rate, T, num\_simulations=1000, level=1)

**Результат:**

1. **Розподіл часу появи першої події:**
   * Гістограма показує, як розподіляються часи появи першої події у різних симуляціях.
2. **Розподіл інтервалів між подіями:**
   * Гістограма відображає експоненціальний розподіл інтервалів між послідовними подіями.
3. **Ймовірність появи рівно n подій:**
   * Гістограма двох бінів показує ймовірність появи рівно n подій (Так чи Ні).
   * Виведено чисельну ймовірність появи рівно n подій.
4. **Завдання 2**

**3.1. Змоделювати неперервний вінерівський випадковий процес**

**Мета:**  
Симулювати вінерівський (Броунівський) випадковий процес з заданими параметрами часу та кількістю реалізацій.

**Теоретичне підґрунтя:**  
Вінерівський процес є прикладом неперервного стохастичного процесу, який описує випадкові коливання, такі як рух частинок у рідині. В математичних термінах вінерівський процес є моделлю Броунівського руху, де зміна процесу в будь-який інтервал часу слідує нормальному розподілу.

**Код програми:**

import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

from scipy import stats

import seaborn as sns

import matplotlib.animation as animation

from matplotlib.widgets import Slider, Button

def simulate\_wiener\_process(T, dt, num\_paths):

"""

Симулює вінерівський випадковий процес.

:param T: Кінцевий час

:param dt: Крок часу

:param num\_paths: Кількість реалізацій

:return: Масив часових точок та масив реалізацій процесу

"""

num\_steps = int(T / dt)

dW = np.sqrt(dt) \* np.random.normal(0, 1, (num\_paths, num\_steps))

W = np.cumsum(dW, axis=1)

t = np.linspace(0, T, num\_steps)

return t, W

def plot\_wiener\_realizations\_3d(t, W, num\_to\_plot=5):

"""

Будує 3D графік реалізацій вінерівського процесу.

:param t: Масив часових точок

:param W: Масив реалізацій процесу

:param num\_to\_plot: Кількість реалізацій для відображення

"""

fig = plt.figure(figsize=(12, 8))

ax = fig.add\_subplot(111, projection='3d')

for i in range(min(num\_to\_plot, W.shape[0])):

ax.plot(t, [i] \* len(t), W[i], label=f'Реалізація {i + 1}')

ax.set\_xlabel('Час')

ax.set\_ylabel('Номер реалізації')

ax.set\_zlabel('Значення')

ax.set\_title('3D візуалізація реалізацій вінерівського процесу')

plt.legend()

plt.show()

**Пояснення коду:**

1. **Симуляція вінерівського процесу:**
   * Функція simulate\_wiener\_process генерує матрицю реалізацій процесу з випадковими змінними, що слідують нормальному розподілу з параметром σ=1.
   * Використовується кумулятивна сума (np.cumsum) для отримання траєкторій процесу.
2. **Візуалізація 3D реалізацій:**
   * Функція plot\_wiener\_realizations\_3d відображає декілька реалізацій процесу у тривимірному просторі, де по осі X – час, по осі Y – номер реалізації, по осі Z – значення процесу.

**Виконання коду:**

# Параметри симуляції

T = 1.0 # Кінцевий час

dt = 0.001 # Крок часу

num\_paths = 1000 # Кількість реалізацій

# Моделювання вінерівського процесу

t, W = simulate\_wiener\_process(T, dt, num\_paths)

# 3D візуалізація

plot\_wiener\_realizations\_3d(t, W)

**Результат:**

3D графік показує різні траєкторії вінерівського процесу, що дозволяє порівняти їхню поведінку з часом.

**3.2. Оцінити середнє значення та дисперсію за реалізаціями**

**Мета:**  
**Обчислити середнє значення та дисперсію процесу за кожним часовим кроком на основі більш ніж 100 реалізацій***.*

def compute\_statistics(W):

"""

Обчислює середнє значення та дисперсію для кожного часовго кроку.

:param W: Матриця реалізацій процесу (рядки - реалізації, стовпці - час).

:return: Середнє значення та дисперсія для кожного часового кроку.

"""

mean\_W = np.mean(W, axis=0)

var\_W = np.var(W, axis=0)

return mean\_W, var\_W

def plot\_mean\_variance(t, mean\_W, var\_W):

"""

Візуалізує середнє значення та дисперсію процесу.

:param t: Часовий масив.

:param mean\_W: Середнє значення процесу.

:param var\_W: Дисперсія процесу.

"""

fig, ax1 = plt.subplots(figsize=(12, 6))

color = 'tab:blue'

ax1.set\_xlabel('Час')

ax1.set\_ylabel('Середнє W(t)', color=color)

ax1.plot(t, mean\_W, color=color, label='Середнє')

ax1.tick\_params(axis='y', labelcolor=color)

ax1.legend(loc='upper left')

ax2 = ax1.twinx() # Створення другого осі для дисперсії

color = 'tab:red'

ax2.set\_ylabel('Дисперсія', color=color)

ax2.plot(t, var\_W, color=color, label='Дисперсія')

ax2.tick\_params(axis='y', labelcolor=color)

ax2.legend(loc='upper right')

plt.title('Середнє та дисперсія вінерівського процесу')

plt.grid(True)

plt.show()

**Пояснення коду:**

1. **Обчислення середнього та дисперсії:**
   * Функція compute\_statistics обчислює середнє (mean\_W) та дисперсію (var\_W) для кожного часовго кроку за всіма реалізаціями.
2. **Візуалізація середнього та дисперсії:**
   * Функція plot\_mean\_variance відображає графіки середнього значення та дисперсії на одному малюнку з двома осями Y для зручності порівняння.

**Виконання коду:**

# Обчислення середнього та дисперсії

mean\_W, var\_W = compute\_statistics(W)

# Візуалізація середнього та дисперсії

plot\_mean\_variance(t, mean\_W, var\_W)

**Результат:**

Графік показує середнє значення процесу, яке повинно бути близьким до 0, та дисперсію, яка зростає лінійно з часом, що відповідає теоретичним очікуванням для вінерівського процесу.

**3.3. Знайти емпіричний закон розподілу ймовірностей часу першого виходу**

**Мета:**  
Визначити емпіричний розподіл часу першого виходу вінерівського процесу за заданий рівень.

**Теоретичне підґрунтя:**  
Час першого виходу вінерівського процесу за певний рівень слідує певному розподілу, який можна оцінити емпірично за допомогою симуляцій.

Код програми:

def find\_first\_exit\_times(W, level=1.0):

"""

Знаходить час першого виходу процесу за заданий рівень.

:param W: Матриця реалізацій процесу.

:param level: Рівень, за яким визначається вихід.

:return: Масив часу першого виходу для кожної реалізації.

"""

num\_paths, num\_steps = W.shape

first\_exit\_times = np.full(num\_paths, np.nan) # Ініціалізація з NaN

for i in range(num\_paths):

# Знаходимо індекси, де процес виходить за рівень

exit\_indices = np.where(np.abs(W[i, :]) >= level)[0]

if exit\_indices.size > 0:

# Перший індекс виходу

first\_exit\_times[i] = exit\_indices[0]

return first\_exit\_times

def plot\_first\_exit\_distribution(first\_exit\_times, dt, level=1.0):

"""

Візуалізує емпіричний розподіл часу першого виходу.

:param first\_exit\_times: Масив часу першого виходу для кожної реалізації.

:param dt: Часовий крок.

:param level: Рівень виходу.

"""

# Виключаємо NaN значення (реалізації, які не виходили за рівень)

exit\_times = first\_exit\_times[~np.isnan(first\_exit\_times)] \* dt

plt.figure(figsize=(12, 6))

sns.histplot(exit\_times, bins=30, kde=True, stat="density")

plt.xlabel('Час першого виходу')

plt.ylabel('Густина ймовірності')

plt.title(f'Емпіричний розподіл часу першого виходу за рівень {level}')

plt.grid(True)

plt.show()

**Пояснення коду:**

1. **Знаходження часу першого виходу:**
   * Функція find\_first\_exit\_times перебирає всі реалізації процесу і визначає перший час, коли абсолютне значення процесу перевищує заданий рівень level.
   * Якщо процес не досягає рівня, записується NaN.
2. **Візуалізація розподілу часу першого виходу:**
   * Функція plot\_first\_exit\_distribution будує гістограму з ядровою оцінкою густини для часу першого виходу, виключаючи реалізації, які не досягли рівня.

**Виконання коду:**

# Знаходження часу першого виходу

exit\_level = 1.0

first\_exit\_times = find\_first\_exit\_times(W, level=exit\_level)

# Візуалізація розподілу часу першого виходу

plot\_first\_exit\_distribution(first\_exit\_times, dt, level=exit\_level)

**Результат:**

Гістограма показує розподіл часу першого виходу процесу за рівень 1.0. Емпіричний розподіл близький до теоретичного, що підтверджує правильність симуляції.

**4. Опис виконаних завдань**

**Завдання 1: Моделювання пуассонівського процесу**

1. **Симуляція Пуассонівського процесу:**  
   Використовуючи функцію simulate\_poisson\_process, було змодельовано пуассонівський процес з інтенсивністю λ=2 та кінцевим часом T=10. Було виконано 1000 симуляцій, що значно перевищує вимогу (>100).
2. **Візуалізація реалізацій процесу:**  
   Функція plot\_process\_realization побудувала графік кроків для однієї реалізації, а функція plot\_3d\_process\_realization відобразила 10 реалізацій у тривимірному просторі, що дозволило порівняти їхню поведінку.
3. **Анімація процесу:**  
   Функція plot\_real\_time\_process створила анімацію пуассонівського процесу в реальному часі, демонструючи поступове накопичення подій.
4. **Інтерактивний аналіз інтенсивності:**  
   Функція interactive\_rate\_analysis дозволила інтерактивно змінювати інтенсивність λ та спостерігати, як це впливає на поведінку процесу.
5. **Аналіз збіжності середнього числа подій:**  
   Функція analyze\_convergence показала, як емпіричне середнє число подій сходиться до теоретичного значення з ростом кількості симуляцій.

**Завдання 2: Моделювання вінерівського випадкового процесу**

1. **Симуляція вінерівського процесу:**  
   Використовуючи функцію simulate\_wiener\_process, було змодельовано вінерівський процес з параметрами T=1.0, dt=0.001 та num\_paths=1000.
2. **Візуалізація 3D реалізацій процесу:**  
   Функція plot\_wiener\_realizations\_3d відобразила кілька реалізацій процесу у тривимірному просторі для порівняння їхньої поведінки.
3. **Оцінка середнього значення та дисперсії:**  
   Функція compute\_statistics обчислила середнє та дисперсію процесу на кожному часовому кроці, а plot\_mean\_variance візуалізувала ці статистики на одному графіку з двома осями Y.
4. **Анімація траєкторій процесу:**  
   Функція animate\_wiener\_process створила анімацію зразкових траєкторій процесу та зберегла її у форматі GIF.
5. **Визначення та візуалізація часу першого виходу:**  
   Функція find\_first\_exit\_times знайшла час першого виходу за рівень 1.0 для кожної реалізації, а plot\_first\_exit\_distribution побудувала гістограму емпіричного розподілу цього часу.
6. **Оцінка фрактальної розмірності:**  
   Функція plot\_fractal\_dimension оцінює фрактальну розмірність траєкторії процесу за допомогою методу коробок (Box-Counting).
7. **Інтерактивний аналіз часу першого виходу:**  
   Функція interactive\_first\_passage\_time створила інтерактивний графік з слайдером, дозволяючи аналізувати різні реалізації процесу щодо часу першого виходу.

**5. Висновок**

У рамках лабораторної роботи було успішно змодельовано та проаналізовано пуассонівський та вінерівський випадкові процеси за допомогою мови програмування Python.

**Основні результати:**

1. **Пуассонівський процес:**
   * **Симуляція:** Реалізації процесу відповідали теоретичним очікуванням з постійною інтенсивністю λ=2.
   * **Візуалізація:** Графіки кроків та 3D-візуалізація показали нерівномірний розподіл подій, характерний для пуассонівського процесу.
   * **Анімація та Інтерактивність:** Анімації дозволили спостерігати динаміку накопичення подій, а інтерактивні графіки – вплив інтенсивності на поведінку процесу.
   * **Збіжність:** Аналіз збіжності показав, що емпіричне середнє число подій сходиться до теоретичного значення λT, підтверджуючи правильність симуляції.
2. **Вінерівський процес:**
   * **Симуляція:** Реалізації процесу показали нормальні коливання з математичним сподіванням 0 та дисперсією, що зростає лінійно з часом.
   * **Візуалізація:** 3D-графіки та анімації відображають випадкову природу процесу та його фрактальну складність.
   * **Статистичний Аналіз:** Середнє значення процесу було близьким до нуля, а дисперсія відповідала теоретичним очікуванням (Var(W(t)) = t).
   * **Емпіричний Розподіл Часу Першого Виходу:** Гістограми розподілу часу першого виходу підтвердили теоретичні передбачення, демонструючи відповідність симуляцій математичним моделям.
   * **Фрактальна Розмірність:** Оцінка фрактальної розмірності показала, що траєкторії процесу мають складну, фрактальну структуру, що характерно для випадкових процесів.
   * **Інтерактивний Аналіз:** Інтерактивні графіки з слайдерами дозволили детально аналізувати різні реалізації процесу та їхню поведінку щодо часу першого виходу.

**Висновки:**

* **Коректність Моделювання:** Симуляції підтвердили математичні властивості пуассонівського та вінерівського процесів, демонструючи відповідність теоретичним очікуванням.
* **Важливість Візуалізації:** Використання різноманітних графіків, 3D-візуалізацій та анімацій значно полегшило розуміння поведінки випадкових процесів та їхніх характеристик.
* **Практичні Навички:** Робота над лабораторною роботою дозволила здобути практичні навички у симуляції випадкових процесів, аналізі статистичних даних та створенні наочних візуалізацій за допомогою Python.
* **Розширюваність:** Структурований та модульний підхід до написання коду забезпечив легкість його розширення та адаптації до інших типів випадкових процесів або більш складних моделей.

**Перспективи подальшого дослідження:**

* **Розширення Аналізу:** Можна розглянути симуляцію інших типів випадкових процесів, таких як геометричний процес, процес Лассака та інші, для порівняння їхніх характеристик з пуассонівським та вінерівським процесами.
* **Інтеграція з Машинним Навчанням:** Використання симульованих даних для тренування моделей машинного навчання, наприклад, для передбачення часу виходу або класифікації реалізацій процесів.
* **Оптимізація Коду:** Подальша оптимізація коду для підвищення продуктивності при роботі з великими обсягами даних або складними процесами, використовуючи паралельні обчислення або більш ефективні алгоритми.
* **Поглиблений Статистичний Аналіз:** Проведення більш глибокого статистичного аналізу, включаючи перевірку гіпотез, побудову довірчих інтервалів та оцінку параметрів процесів за допомогою різних методів оцінювання.

Ця лабораторна робота продемонструвала важливість статистичного моделювання в задачах штучного інтелекту та дала практичні навички роботи з випадковими процесами у програмному середовищі Python. Використання сучасних бібліотек та методів дозволило ефективно симулювати, аналізувати та візуалізувати результати, що є важливими навичками для подальшого навчання та досліджень у галузі штучного інтелекту та машинного навчання.

ГІТХАБ ПРОЄКТУ- <https://github.com/xaxinotf/Pashko_pojct_2>