МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет
«Харківський авіаційний інститут»

О. В. Лучшева, В. О. Захаренко

ОСНОВИ СТАТИСТИКИ В PYTHON

Навчальний посібник до практичних робіт

Харків «ХАІ» 2026

УДК 519.2:004.43(076.5)
Л87

Рецензенти: д-р техн. наук, проф. М. О. Волк,
д-р техн. наук, проф. Ю. О. Романенков

Лучшева, О. В.
Л87 Основи статистики в Python [Електронний ресурс] : навч. посіб. до практ. робіт / О. В. Лучшева, В. О. Захаренко. – Харків : Нац. аерокосм. ун-т «Харків. авіац. ін-т», 2026. – 111 с.

ISBN 978-966-996-105-1

Наведено теоретичні відомості та комплекс практичних робіт для самостійного виконання, що дає змогу сформувати системні знання та практичні навички застосування методів математичної статистики за допомогою сучасних програмних засобів.

Викладено ключові поняття теорії ймовірностей та математичної статистики, наведено основні формули, а також детально розглянуто приклади їх реалізації в середовищі програмування Python та в табличному процесорі MS Excel. Матеріал супроводжується ілюстраціями, фрагментами коду та поясненнями, що полегшує засвоєння як теоретичних основ, так і практичних аспектів аналізу даних. Практичні роботи побудовано на основі прикладних задач з різних галузей, що дає змогу не лише закріпити обчислювальні навички, але й навчитися інтерпретувати отримані статистичні результати для прийняття обґрунтованих рішень.

Для здобувачів освіти інженерних, комп'ютерних та економічних спеціальностей, аспірантів, а також широкого кола фахівців — аналітиків, інженерів та науковців, — які прагнуть опанувати сучасний інструментарій для статистичного аналізу даних.

Іл. 1. Бібліогр.: 14 назв

УДК 519.2:004.43(076.5)

ISBN 978-966-996-105-1

ЗМІСТ

ВСТУП

Теорія ймовірностей та математична статистика є фундаментальними дисциплінами, що лежать в основі сучасних методів аналізу даних, інженерних досліджень та процесів прийняття рішень в умовах невизначеності. У добу цифрової трансформації володіння теоретичним апаратом цих наук стає недостатнім. Ключовою компетенцією сучасного фахівця є здатність застосовувати математичні методи на практиці, використовуючи для цього ефективні програмні інструменти.

Метою цього навчального посібника є формування у здобувачів освіти стійких практичних навичок застосування основних методів математичної статистики для розв'язання прикладних задач. Посібник орієнтовано на широке коло майбутніх фахівців: інженерів, аналітиків даних, економістів, науковців та менеджерів, які у своїй професійній діяльності будуть стикатися з необхідністю приймати обґрунтовані рішення на основі отриманих даних. Матеріали посібника спрямовано на формування сучасної аналітичної культури та надання інструментарію для роботи з реальними даними.

Особливістю посібника є інтегрований підхід, що передбачає паралельне використання табличного процесора MS Excel та мови програмування Python. Такий підхід має важливу дидактичну перевагу. Виконання розрахунків у MS Excel дає змогу візуалізувати логіку обчислень та сформувати інтуїтивне розуміння математичного апарату. Подальша реалізація тих самих завдань у Python не лише навчає автоматизації, але й є потужним засобом взаємної перевірки. Збіг результатів, отриманих двома різними інструментами, надає впевненості у правильності як аналітичних міркувань, так і розробленого програмного коду.

У посібнику послідовно розглянуто шлях від базових операцій з випадковими подіями до перевірки статистичних гіпотез, включно з первинним обробленням даних, обчисленням числових характеристик, методами оцінювання та графічним аналізом. Вибір Python зумовлений його провідною роллю в галузі Data Science та наявністю потужних бібліотек (NumPy, SciPy, Matplotlib). У виданні також враховано сучасні тенденції, зокрема використання інтелектуальних асистентів як допоміжного засобу для оптимізації процесу написання коду. Особливу увагу приділено не лише отриманню числових результатів, але і їх інтерпретації в контексті поставленного завдання, що розвиває навички критичного мислення.

Виконання практичних робіт, наведених у посібнику, сприятиме підвищенню статистичної грамотності та розвитку навичок програмування, що є міцним фундаментом для вивчення дисциплін у галузі машинного навчання, аналізу даних та моделювання складних систем.

ПОЛОЖЕННЯ ТА ВИМОГИ ДО ВИКОНАННЯ ПРАКТИЧНИХ РОБІТ

У посібнику наведено типовий варіант завдання для ознайомлення з алгоритмом розрахунків та самостійного виконання. Повний перелік варіантів індивідуальних завдань для кожного здобувача освіти надається викладачем безпосередньо на практичному занятті в електронному форматі.

Загальні критерії оцінювання практичних робіт

Підсумкова оцінка за кожну практичну роботу формується на основі таких компонентів та їх вагових коефіцієнтів:

1. **Правильність коду та аналітичних розрахунків (40 %):**

— програма мовою Python працює без помилок і видає коректні результати відповідно до алгоритму;

— виконання розрахунків уручну або в MS Excel використовується як обов'язкова альтернативна перевірка (верифікація) для підтвердження правильності результатів, отриманих програмним шляхом.

2. **Якість результатів та візуалізацій (20 %):**

— результати розрахунків подано у зручному для аналізу форматі (структуровані таблиці або форматоване виведення);

— графіки згенеровані програмним шляхом за допомогою бібліотек Python (допускається додаткова візуалізація в MS Excel для порівняння та підтвердження коректності);

— графічні матеріали мають належне оформлення (назви, підписи осей, легенду) та дають змогу однозначно інтерпретувати отримані статистичні дані.

3. **Глибина технічних висновків (20 %):**

— висновки логічно випливають з отриманих результатів та містять статистичну інтерпретацію (наприклад, оцінку форми розподілу, значущості гіпотез або точності інтервалів);

— здобувач освіти демонструє розуміння прикладного значення аналізу, пояснюючи, які властивості досліджуваного процесу відображають отримані цифри та графіки.

4. **Аналіз використання ШІ та рефлексія (10 %):**

— чітко вказано назву та версію моделі ШІ, що використовувалася як допоміжний інструмент;

— наведено приклади промптів (запитів) та критичний аналіз отриманих відповідей (виявлення помилок, оптимізація коду);

— зроблено висновок про те, як саме використання ШІ допомогло в опануванні матеріалу та перевірці власних припущень.

5. **Оформлення звіту та відповіді на запитання (10 %):**

— звіт структуровано й оформлено згідно з вимогами;

— надано чіткі та правильні відповіді на контрольні запитання, що підтверджує засвоєння теоретичного базису практичної роботи.

Загальний порядок здачі робіт

Підготовка та форматування звіту

Звіт є основним документом, що підтверджує факт виконання практичної роботи, фіксує отримані результати та містить їх наукову інтерпретацію. Звітну документацію необхідно готувати виключно у форматі .docx (Microsoft Word) для забезпечення можливості рецензування викладачем.

Під час підготовки текстової частини, розрахункових таблиць та графічних матеріалів здобувач освіти має суворо дотримуватися технічних стандартів, викладених у методичних вказівках кафедри (див. окремий файл «Вимоги до оформлення звітів»). Особливу увагу слід приділяти якості оформлення графіків та повноті підписів до них. Неналежний стиль форматування або відсутність обов'язкових структурних елементів є підставою для зниження підсумкової оцінки або повернення роботи на доопрацювання.

Формування повного комплекту звітних файлів

Для забезпечення прозорості та можливості повної перевірки достовірності розрахунків здобувач освіти має підготувати цілісний пакет матеріалів, до якого обов'язково входять:

1. *Текстовий звіт* про виконання практичної роботи (повна версія зі скриншотами та висновками).
2. *Файли з вихідним програмним кодом* у форматі .py (стандартні скрипти) або .ipynb (інтерактивні блокноти Jupyter/Colab). Програмний код має бути структурованим, містити змістовні коментарі до ключових етапів аналізу (імпорт даних, розрахунок метрик, візуалізація) та бути повністю готовим до запуску в робочому середовищі без додаткових виправлень.
3. *Допоміжні розрахункові файли* у форматі .xlsx. Ці файли додаються як обов'язковий елемент у разі проведення альтернативної верифікації результатів, що дає змогу порівняти точність роботи алгоритмів Python із класичними табличними методами оброблення даних.

Процедура завантаження та технічні вимоги до архіву

Усі зазначені матеріали мають бути зібрані в одну папку та подані у вигляді єдиного архівного файлу стандартного формату (zip або 7z), що дає змогу зберегти цілісність структури коду та уникнути пошкодження файлів під час передачі.

1. *Завантаження архіву* здобувач освіти здійснює самостійно через систему дистанційного навчання Moodle безпосередньо у відповідний розділ курсу.
2. Робота вважається прийнятою до розгляду лише за наявності **повного комплекту файлів**. У разі виявлення помилок у коді або невідповідності результатів у звіті та програмних файлах викладач має право призначити додатковий захист роботи у синхронному режимі.

Практична робота № 1
ОПЕРАЦІЇ НАД ПОДІЯМИ ТА ЙМОВІРНІСНЕ МОДЕЛЮВАННЯ

Мета роботи:

— закріпити теоретичні знання про простір елементарних подій та операції над ними (об'єднання, перетин, різниця);

— навчитися аналітично та програмно знаходити результати операцій над множинами подій;

— опанувати метод статистичних випробувань (метод Монте-Карло) для емпіричного оцінювання ймовірностей;

— освоїти техніку візуалізації результатів симуляції за допомогою Python.

Компетентності, що формуються

Після виконання цієї роботи здобувач освіти зможе:

— формалізувати умови задач з теорії ймовірностей, визначаючи простір подій та їх множини;

— виконувати базові операції над множинами: об'єднання, перетин, різниця, доповнення;

— пояснювати суть методу Монте-Карло та умови його застосування;

— розробляти програмні скрипти на Python для моделювання випадкових експериментів;

— обчислювати ймовірності аналітично (для дискретних подій) та емпірично (методом Монте-Карло);

— використовувати бібліотеки numpy та matplotlib для генерації випадкових чисел та візуалізації даних.

Необхідне програмне забезпечення та бібліотеки:

1. Середовище розроблення: Google Colab, Jupyter Notebook або будь-яке IDE для Python.
2. Python версії 3.8 або новішої з бібліотеками: numpy, matplotlib.
3. MS Excel для аналітичних розрахунків та перевірки.

Теоретичні відомості

Основні поняття теорії ймовірностей

В основі теорії ймовірностей лежить поняття випадкового експерименту – дії, результат якої неможливо передбачити однозначно. Простір елементарних подій (Ω) – це множина всіх можливих, взаємовиключних наслідків експерименту. Наприклад, при киданні кубика $\Omega = \{1, 2, 3, 4, 5, 6\}$.

Випадкова подія ($A, B, \ldots$) – це будь-яка підмножина простору Ω. Подія відбувається, якщо результат експерименту є елементом цієї множини.

Достовірна подія: подія, яка відбувається завжди ($A = \Omega$).

Неможлива подія: подія, яка не може відбутися (A = ∅, порожня множина).

Класичне означення ймовірності

Якщо простір Ω складається зі скінченної кількості рівноможливих елементарних подій, то ймовірність події A обчислюється як відношення кількості сприятливих для A подій (m) до загальної кількості подій (n):

$$P(A) = \frac{m}{n}.$$

Наприклад, ймовірність випадіння парного числа на кубику: $A = \{2, 4, 6\}$, отже, $m = 3$, $n = 6$ і $P(A) = 3/6 = 0.5$.

Операції над подіями та їх імовірності

Оскільки події є множинами, над ними можна виконувати стандартні теоретико-множинні операції:

1. Об'єднання подій ($A \cup B$ або $A + B$): подія, що полягає в настанні *хоча б однієї* з подій A або B.
2. Перетин подій ($A \cap B$ або $A \cdot B$): подія, що полягає в настанні *обох подій* A і B одночасно.
3. Різниця подій ($A \setminus B$): подія, що полягає в настанні події A, але *ненастанні* події B.
4. Доповнення (протилежна подія) ($\bar{A}$): подія, що полягає в тому, що подія A *не відбулася*. $\bar{A} = \Omega \setminus A$. Ймовірність протилежної події: $P(\bar{A}) = 1 - P(A)$.

Теорема додавання ймовірностей

Ймовірність об'єднання двох подій дорівнює сумі їх імовірностей без імовірності їх спільного настання:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Реалізація операцій в Python

Приклад програми:

```
# Визначення простору елементарних подій (гральний кубик)
omega = {1, 2, 3, 4, 5, 6}
# Подія A: випало парне число
A = {2, 4, 6}
# Подія B: випало число, яке більше трьох
B = {4, 5, 6}
# 1. Об'єднання подій (A ∪ B) — відбувається хоча б одна з подій.
# В Python використовується оператор вертикальної риски "|"
union_AB = A | B
print(f"Об'єднання A ∪ B: {union_AB}")
# 2. Перетин подій (A ∩ B) — події відбуваються одночасно.
# В Python використовується оператор амперсанда "&"
intersection_AB = A & B
print(f"Перетин A ∩ B: {intersection_AB}")
# 3. Різниця подій (A \ B) — подія A відбувається, а B — ні.
# В Python використовується оператор мінуса "-"
difference_AB = A - B
# Використовуємо подвійний слеш або r-рядок для уникнення SyntaxWarning
print(f"Різниця A \\ B: {difference_AB}")
```

```
# 4. Доповнення до події A (A') — подія A не відбулася.
# Визначається як різниця між універсальною множиною omega та подією A
complement_A = omega - A
print(f"Доповнення до A: {complement_A}")
```

Результат роботи програми:

```
Об'єднання A U B: {2, 4, 5, 6}
Перетин A ∩ B: {4, 6}
Різниця A \ B: {2}
Доповнення до A: {1, 3, 5}
```

Геометрична ймовірність та метод Монте-Карло

Геометрична ймовірність застосовується, коли простір Ω є неперервною геометричною областю (відрізок, площина, об'єм). Ймовірність потрапляння точки в деяку підобласть g усередині загальної області G визначається як відношення їх мір (довжин, площ, об'ємів):

$$P(A) = \frac{m(g)}{m(G)}.$$

Метод Монте-Карло – це числовий метод, що дає змогу знайти наближене значення ймовірності шляхом проведення великої кількості (N) випадкових експериментів. Для оцінювання геометричної ймовірності:

— генерується N випадкових точок, рівномірно розподілених у великій області G;

— підраховується кількість k точок, що потрапили всередину цільової області g;

— ймовірність наближено дорівнює відношенню $P \approx k / N$.

Згідно з законом великих чисел при необмеженому збільшенні кількості випробувань ($N \to \infty$) відносна частота k/N збігається до істинної ймовірності P. Це означає, що чим більше точок генеруємо, тим точнішою, в середньому, буде оцінка.

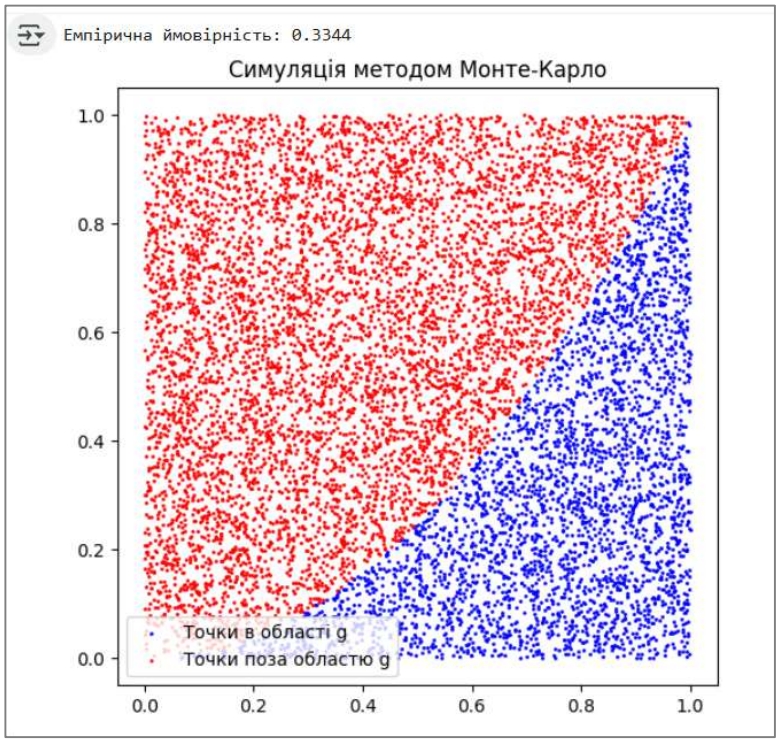
Реалізація в Python

Приклад реалізації:

```
import numpy as np
import matplotlib.pyplot as plt
# Кількість випадкових точок (експериментів)
N = 10000
# Генеруємо N випадкових точок (x, y) в квадраті [0,1] x [0,1]
points = np.random.rand(N, 2)
x, y = points[:, 0], points[:, 1]
# Умова потрапляння в цільову область g (наприклад, y < x^2)
condition = y < x**2
# Кількість точок, що задовольняють умову
k = np.sum(condition)
# Емпірична ймовірність
probability = k / N
print(f"Емпірична ймовірність: {probability:.4f}")
# Візуалізація
plt.figure(figsize=(6, 6))
plt.scatter(x[condition], y[condition], s=1, c='blue', label='Точки в області
g')
```

```
    plt.scatter(x[~condition], y[~condition], s=1, c='red', label='Точки поза областю g')
    plt.legend()
    plt.title("Симуляція методом Монте-Карло")
    plt.show()
```

Результат роботи програми:



Порядок виконання роботи

Теоретична підготовка

1. Опрацювати фундаментальні поняття теорії ймовірностей: випадковий експеримент, подія та аксіоматична побудова простору елементарних подій Ω.
2. Вивчити визначення основних операцій над подіями (об'єднання, перетин, різниця), розглядаючи їх як відповідні операції над множинами.
3. Розглянути класичне та геометричне означення ймовірності, визначивши межі їх застосування.
4. Дослідити принципи методу статистичних випробувань (Монте-Карло) та його математичний зв'язок із законом великих чисел.
5. Ознайомитися з програмною реалізацією операцій над множинами (тип даних set) у мові Python, а також із функціональними можливостями бібліотек *numpy.random* (генерація чисел) та *matplotlib.pyplot* (візуалізація).

Аналітичне розв'язання (MS Excel / Вручну)

На цьому етапі проводиться апріорне дослідження задачі для отримання результатів, що будуть еталоном для верифікації коду.

1. Для *завдання 1*:

— чітко формалізувати елементи простору елементарних подій Ω;

— визначити склад множин, що відповідають подіям A, B, C згідно з варіантом;

— виконати задані операції над множинами аналітичним шляхом, зафіксувавши вислідні множини.

2. Для *завдання 2*:

— обчислити точні значення мір (площ) цільової області g та базової області G, використовуючи методи інтегрування або відповідні геометричні формули;

— розрахувати точне значення геометричної ймовірності за формулою $P = m(g)/m(G)$.

Програмна реалізація (Python)

Створити новий Python-скрипт (.py) або інтерактивний блокнот (.ipynb) та виконати імпорт необхідних бібліотек.

1. Для *завдання 1*:

— ініціалізувати об'єкти типу *set*, що відповідають подіям A, B, C ;

— програмно реалізувати обчислення заданих виразів за допомогою операторів |, &, −;

— забезпечити форматоване виведення результатів для подальшого порівняльного аналізу.

2. Для *завдання 2:*

— розробити функцію, що реалізує алгоритм статистичного моделювання методом Монте-Карло;

— провести серію експериментів для вибірок обсягом $N = 1000$ та $N = 10000$ випадкових точок;

— для кожного експерименту обчислити та вивести на екран наближене значення ймовірності;

— побудувати графічну ілюстрацію симуляції (візуалізацію точок) для наочного відображення розподілу випадкових величин.

Аналіз результатів та формування висновків

1. Порівняти результати операцій над множинами, отримані аналітично та за допомогою Python.

На цьому етапі необхідно провести детальне зіставлення отриманих множин. У разі виявлення розбіжностей здобувач освіти має ідентифікувати їх причини: чи це була помилка в логіці побудови Python-коду, чи похибка в аналітичних розрахунках «уручну».

2. Порівняти точне значення геометричної ймовірності (з етапу 2) з наближеними значеннями, отриманими методом Монте-Карло для $N = 1000$ та $N = 10000$.

Слід оцінити абсолютне відхилення емпіричного результату від теоретичного еталона. Це дає змогу наочно переконатися у працездатності розробленого алгоритму симуляції.

3. Проаналізувати, як збільшення кількості випробувань N вплинуло на точність емпіричної оцінки ймовірності.

Необхідно зробити висновок про характер збіжності результатів: чи наблизилося значення ймовірності до точного при збільшенні вибірки в 10 разів, та як це відобразилося на візуальній щільності точок на графіках.

4. Сформулювати загальні висновки до роботи, підсумувавши виконані дії та отримані результати.

Висновок має бути комплексним: від підтвердження засвоєння операцій над множинами до оцінки ефективності використання методу Монте-Карло для розв'язання статистичних задач.

Вимоги до звіту та його вміст

Звіт є підсумковим документом, що фіксує результати проведеного дослідження, його структура має містити такі обов'язкові компоненти:

1. Титульна сторінка.

Оформлюється згідно з установленим зразком. Має містити назву дисципліни, номер роботи, дані здобувача освіти та викладача.

2. Тема, мета роботи та номер варіанта.

Формулюються чітко відповідно до методичних вказівок.

3. Постановка завдання.

Необхідно повністю скопіювати умову завдання 1 та завдання 2 зі свого індивідуального варіанта.

4. Результати виконання.

Це основний розділ звіту, що складається з трьох частин:

а) розрахунки (MS Excel / Вручну):

— для *завдання 1:* початкові множини та вислідні множини після кожної операції;

— для *завдання 2:* розрахунок точних площ та аналітичної ймовірності;

б) код програми: повний програмний код на Python з коментарями для обох завдань;

в) результати роботи програми:

— для *завдання 1:* скріншот текстового виведення (для множин);

— для *завдання 2:* скріншот виведення ймовірностей для $N = 1000$ та $N = 10000$ разом із графіком візуалізації.

5. Аналіз використання ШІ-асистента.

У цьому розділі здобувач освіти має продемонструвати навички критичної взаємодії з технологіями штучного інтелекту:

а) вибір програмного інструментарію ШІ. Вказати повну назву та версію використаної моделі (наприклад, Gemini 1.5 Flash, ChatGPT 4.0, Claude 3.5);

б) роль ШІ у виконанні роботи.

Навести 2–3 конкретні приклади запитів (промптів) та проаналізувати отримані відповіді. Наприклад, як ШІ допоміг з генерацією коду для візуалізації, пояснив синтаксис операцій над множинами або допоміг виправити помилки (debug) у логіці програми.

6. Висновки.

Розділ має містити підсумок проведеного дослідження, поділений на два аспекти:

а) технічний висновок: необхідно провести компаративний аналіз результатів:

— для *завдання 1:* підтвердити збіжність аналітичних та програмних розрахунків;

— для *завдання 2*: зіставити точне значення геометричної ймовірності з емпіричними оцінками та зробити аргументований висновок про те, як збільшення кількості випробувань N впливає на точність методу Монте-Карло;

б) рефлексія щодо використання ШІ.

Критично оцінити ефективність допомоги асистента. Описати, чи сприяв ШІ глибшому розумінню суті методу Монте-Карло, чи його роль обмежилася технічним пришвидшенням написання та налагодження коду.

Завдання для самостійного виконання

Завдання 1. В експерименті з киданням двох гральних кубиків простір подій Ω складається з пар (i, j). Розглядають такі події:

— A – сума очок дорівнює 8;

— B – хоча б на одному кубику випало 5 очок;

— C – на обох кубиках випали парні числа.

Знайти: $A \cup B$, $A \cap C$, $B \setminus C$.

Завдання 2. У квадрат $[0, 2]$ x $[0, 2]$ навмання кидається точка. Знайти ймовірність того, що вона потрапить в область, обмежену кривими $y = x^2$ та $x = y^2$.

Контрольні запитання

1. Що таке простір елементарних подій? Наведіть приклад для експерименту з киданням двох монет.

2. Дайте визначення об'єднання, перетину та різниці двох подій. Яким логічним сполучникам («І», «АБО») вони відповідають?

3. Що таке протилежна подія (доповнення)? Яка формула пов'язує ймовірність події та її доповнення?

4. Сформулюйте класичне визначення ймовірності та назвіть умови, за яких його можна застосовувати.

5. Запишіть теорему додавання ймовірностей для двох довільних подій. Коли ця формула спрощується до простого додавання $P(A + B) = P(A) + P(B)$?

6. У чому полягає суть методу Монте-Карло для обчислення геометричної ймовірності?

7. Який фундаментальний закон теорії ймовірностей обґрунтовує, що метод Монте-Карло дає результат, близький до істинного?

8. Як і чому кількість випробувань N впливає на точність оцінки ймовірності методом Монте-Карло?

9. Як у мові Python реалізуються операції об'єднання, перетину та різниці для об'єктів типу *set*?

10. Які функції з бібліотеки *numpy* використовуються для генерації випадкових чисел у заданому діапазоні? Наведіть приклад.

11. Як формулюється визначення геометричної ймовірності? Чим воно відрізняється від класичного визначення і яку роль у ньому відіграють міри (довжина, площа, об'єм)?

12. Поясніть різницю між статистичною частотою події (отриманою внаслідок моделювання) та її теоретичною ймовірністю. Чому ці значення не завжди збігаються і від чого залежить величина похибки?

13. Як за допомогою логічних умов у Python визначити, чи потрапила згенерована точка (x, y) всередину фігури (наприклад, кола $x^2 + y^2 \leq R^2$)?

Практична робота № 2
РЕАЛІЗАЦІЯ ФОРМУЛ ПОВНОЇ ЙМОВІРНОСТІ ТА БАЙЄСА В PYTHON

Мета роботи:

— закріпити теоретичні знання про систему гіпотез, формулу повної ймовірності та теорему Байєса;

— опанувати методику програмної реалізації ймовірнісних розрахунків у середовищі Python;

— розвинути навички аналізу апостеріорних даних та інтерпретації результатів для прийняття рішень в умовах невизначеності.

Компетентності, що формуються

Після виконання цієї роботи здобувач освіти зможе:

— формалізувати прикладні задачі, визначаючи повну групу несумісних подій (гіпотез);

— розраховувати повну ймовірність складної події, що залежить від низки факторів;

— застосовувати формулу Байєса для переоцінки (уточнення) ймовірностей гіпотез на основі нових даних;

— створювати Python-скрипти для автоматизації розрахунків повної та апостеріорної ймовірностей;

— використовувати можливості ШІ-асистентів для пояснення логіки байєсівського виведення та налагодження коду.

Необхідне програмне забезпечення та бібліотеки

1. Середовище розроблення: Google Colab, Jupyter Notebook або будь-яке інше інтегроване середовище розроблення (IDE) для Python (наприклад, VS Code, PyCharm).
2. Мова програмування Python (версія 3.8 або новіша).
3. Табличний процесор MS Excel (або аналоги) для верифікації розрахунків.

Теоретичні відомості

Основні поняття та ідея методу

У багатьох задачах нас цікавить ймовірність певної події A, яка може відбутися внаслідок різних, незалежних одна від одної причин. Ці причини в теорії ймовірностей називають **гіпотезами**.

Гіпотези ($\boldsymbol{H_1, H_2, \ldots, H_n}$) – це припущення про умови, в яких відбувається експеримент. Вони повинні утворювати **повну групу подій**, що означає:

1. Несумісність (жодні дві гіпотези не можуть відбутися одночасно).
2. Вичерпність (обов'язково відбудеться одна з цих гіпотез). Звідси випливає, що сума їх ймовірностей дорівнює 1:

$$P(H_1) + P(H_2) + \cdots + P(H_n) = 1.$$

Ймовірності гіпотез $P(H_\text{i})$ до початку експерименту (до того, як отримали нову інформацію) називаються апріорними (додослідними).

Формула повної ймовірності

Ця формула відповідає на запитання: «Яка загальна ймовірність настання події A, якщо вона може бути спричинена будь-якою з гіпотез H_i?». Формулу загальной ймовірності події A можна записати так:

$$P(A) = P(H_1)P(A \mid H_1) + P(H_2)P(A \mid H_2) + \cdots + P(H_n)P(A \mid H_n) = \\ = \sum_{i=1}^{n} P(H_i)P(A \mid H_i),$$

де $P(A)$ – повна ймовірність події A;

$P(H_\text{i})$ – апріорна ймовірність і-ї гіпотези;

$P(A|H_\text{i})$ – умовна ймовірність події A за умови, що настала і-та гіпотеза.

Логіка формули: знаходимо суму добутків ймовірності події А за кожної з умов H_i ($P(A|H_i)$) на ймовірність самої цієї умови ($P(H_i)$).

Формула Байєса (теорема гіпотез)

Ця формула є логічним продовженням попередньої і відповідає на обернене запитання: «Подія А вже відбулася. Яка тепер імовірність того, що її спричинила конкретна гіпотеза H_i?».

Формула Байєса дає змогу оновити знання про ймовірності гіпотез після отримання нових даних (факту настання події А):

$$P(H_i \mid A) = \frac{P(H_i) \cdot P(A \mid H_i)}{P(A)},$$

де $P(H_i|A)$ – апостеріорна (післядосвідна) ймовірність гіпотези H_i. Це те, що необхідно знайти;

$P(H_i)$ – апріорна ймовірність гіпотези H_i;

$P(A|H_i)$ – умовна ймовірність події А при гіпотезі H_i;

$P(A)$ – повна ймовірність події А (розраховується за попередньою формулою і є нормувальним множником).

Реалізація в Python

Для реалізації цих формул найзручніше використовувати списки (list) або масиви numpy.

Кроки реалізації:

— визначити вхідні дані (створити два списки: один для апріорних ймовірностей гіпотез, інший – для відповідних умовних ймовірностей);

— обчислити повну ймовірність (використовуючи цикл або генератор списку, поелементно перемножити два списки і знайти суму добутків);

— обчислити апостеріорні ймовірності (для кожної гіпотези обчислити її апостеріорну ймовірність за формулою Байєса, використовуючи вже розраховану повну ймовірність).

Приклад коду:

```
#Вхідні дані (приклад для трьох гіпотез)
# Ймовірності гіпотез P(H_i)
probabilities_h = [0.25, 0.40, 0.35]
# Умовні ймовірності P(A|H_i)
conditional_probabilities_a = [0.03, 0.02, 0.05]

# 1. Розрахунок повної ймовірності P(A)
# Використовуємо генератор списку та функцію sum для лаконічності
total_probability_a = sum(p_h * p_a_h for p_h, p_a_h in zip(probabilities_h,
conditional_probabilities_a))
print(f"Повна ймовірність події A: {total_probability_a:.4f}")
# 2. Розрахунок за формулою Байєса P(H_i|A)
print("\nАпостеріорні ймовірності гіпотез:")
posterior_probabilities_h = []
for i in range(len(probabilities_h)):
    # Чисельник формули Байєса
    numerator = probabilities_h[i] * conditional_probabilities_a[i]
    # Ділення на повну ймовірність
```

```
        p_h_given_a = numerator / total_probability_a
        posterior_probabilities_h.append(p_h_given_a)
        print(f"Ймовірність гіпотези H{i+1} після події A: {p_h_given_a:.4f}")
    # Перевірка: сума апостеріорних ймовірностей має дорівнювати 1
    print(f"\nПеревірочна сума апостеріорних ймовірностей: {sum(posterior_probabilities_h):.4f}")
```

Результат роботи програми:

```
Повна ймовірність події A: 0.0330

Апостеріорні ймовірності гіпотез:
Ймовірність гіпотези H1 після події A: 0.2273
Ймовірність гіпотези H2 після події A: 0.2424
Ймовірність гіпотези H3 після події A: 0.5303

Перевірочна сума апостеріорних ймовірностей: 1.0000
```

Реалізація в MS Excel

Excel дає змогу наочно структурувати дані та розрахунки в табличній формі:

	A	B	C	D	E
1	**Гіпотеза**	**P(H)**	**P(A|H)**	**P(H) * P(A|H)**	**P(H|A)**
2	H_1	0,25	0,03	0,0075	0,22727273
3	H_2	0,4	0,02	0,008	0,24242424
4	H_3	0,35	0,05	0,0175	0,53030303
5					
6	**P(A) =**	0,033		**Перевірка:**	1

— стовпець «P(H) * P(A|H)» (у цьому стовпці для кожного рядка обчисліть добуток відповідних значень зі стовпців «P(H)» та «P(A|H)»);

— обчислення повної ймовірності P(A) (під таблицею знайдіть суму всіх значень зі стовпця «P(H) * P(A|H)», отримане число і є повною ймовірністю події A);

— стовпець «P(H|A)»: для розрахунку апостеріорної ймовірності для кожної гіпотези візьміть відповідне значення зі стовпця «P(H) * P(A|H)» і розділіть його на загальну повну ймовірність P(A).

При створенні формули для першого рядка зробіть посилання на комірку з повною ймовірністю абсолютним (додавши знаки $ перед літерою стовпця та номером рядка, наприклад, B6). Це дасть змогу просто «протягнути» формулу вниз для інших гіпотез без помилок.

Перевірка: сума всіх значень у стовпці «P(H|A)» повинна дорівнювати 1, це підтверджує, що всі розрахунки були виконані правильно.

Порядок виконання роботи

Теоретична підготовка

1. Опрацювати фундаментальні поняття: повна група несумісних подій, апріорна та апостеріорна ймовірності.

2. Вивчити умови застосування формули повної ймовірності та теореми Байєса.

3. Підготувати інструментарій: перевірити налаштування середовища Python та створити шаблон таблиці в MS Excel (або аналогах, наприклад, Google Sheets).

4. Обрати індивідуальний варіант завдання згідно зі списком.

Аналітичне розв'язання (MS Excel / Вручну)

На цьому етапі виконується розрахунок еталонних значень (на папері або в електронній таблиці) для подальшої перевірки коду.

1. Формалізація даних:

— визначити перелік гіпотез H_i та відповідні ймовірності ($P(H_i)$, $P(A|H_i)$);

— занести дані у таблицю MS Excel або виписати їх для ручного розрахунку.

2. Розрахунок повної ймовірності:

— обчислити добутки $P(H_i) \cdot P(A|H_i)$ для кожної гіпотези;

— знайти суму цих добутків – це і є повна ймовірність $P(A)$ (в Excel використати функцію суми).

3. Переоцінка гіпотез:

— розрахувати апостеріорні ймовірності $P(H_i|A)$, поділивши проміжний добуток на знайдену повну ймовірність;

— перевірити коректність: сума всіх отриманих апостеріорних імовірностей має дорівнювати 1.

Програмна реалізація та використання ШІ

Створити новий скрипт (.py) або блокнот (.ipynb) та виконати такі дії:

1. Підготовка даних: створити списки (або масиви) для зберігання вхідних значень: апріорних імовірностей $P(H_i)$ та умовних імовірностей $P(A|H_i)$ згідно з варіантом.

2. Залучення ШІ-асистента:

— використати ШІ (Gemini, ChatGPT) для оптимізації роботи над кодом;

— завдання для ШІ: попросити згенерувати функцію, яка приймає два списки ймовірностей і повертає результати розрахунків, АБО попросити пояснити синтаксис специфічних конструкцій Python (наприклад, логіку функції zip() для паралельної ітерації).

3. Реалізація алгоритму:

— написати (або адаптувати згенерований) код для обчислення повної ймовірності $P(A)$ та апостеріорних імовірностей $P(H_i|A)$;

— додати коментарі до ключових рядків коду, що пояснюють математичний зміст операцій.

4. Виведення результатів: забезпечити зрозуміле текстове виведення результатів з точністю до чотирьох знаків після коми.

Аналіз результатів та формування висновків

1. Верифікація розрахунків:

— порівняти числові значення повної ймовірності та апостеріорних імовірностей, отримані в програмі (Python), з результатами попереднього етапу (MS Excel / ручний розрахунок);

— переконатися, що розбіжність відсутня (або пояснюється мінімальною похибкою округлення).

2. Інтерпретація результатів:

— проаналізувати динаміку змін: порівняти апріорні ймовірності $P(H_i)$ з апостеріорними $P(H_i|A)$;

— визначити «гіпотезу-переможця»: яка причина стала найбільш вірогідною після того, як подія відбулася.

3. Рефлексія щодо використання ШІ: у висновках коротко описати свій досвід взаємодії з ШІ-асистентом: чи допоміг він зрозуміти логіку коду, чи довелося виправляти згенерований скрипт, наскільки корисними були пояснення.

Вимоги до звіту та його вміст

Звіт є підсумковим документом, що фіксує результати проведеного дослідження, його структура має містити такі обов'язкові компоненти:

1. Титульна сторінка.

Оформлюється згідно з установленим зразком. Має містити назву дисципліни, номер роботи, дані здобувача освіти та викладача.

2. Тема, мета роботи та номер варіанта.

Формулюються чітко відповідно до методичних вказівок.

3. Постановка завдання.

Необхідно повністю скопіювати умову задачі зі свого індивідуального варіанта (опис ситуації, статистичні дані по групах/гіпотезах).

4. Результати виконання.

Це основний розділ звіту, що складається з трьох частин:

а) розрахунки (MS Excel / Вручну): таблиця або детальний опис розрахунків, що містить: перелік гіпотез (H_i), їх апріорні ймовірності ($P(H_i)$), умовні ймовірності ($P(A|H_i)$), проміжні добутки та фінальний розрахунок апостеріорних ймовірностей ($P(H_i|A)$);

б) код програми: повний програмний код на Python із коментарями, що пояснюють кроки обчислення повної ймовірності та переоцінки гіпотез;

в) результати роботи програми: скріншот текстового виведення консолі (або комірки Jupyter Notebook) з отриманими числовими значеннями.

5. Аналіз використання ШІ-асистента.

У цьому розділі здобувач освіти має продемонструвати навички критичної взаємодії з технологіями штучного інтелекту:

а) вибір програмного інструментарію ШІ. Вказати повну назву та версію використаної моделі (наприклад, Gemini 1.5 Flash, ChatGPT 4.0, Claude 3.5);

б) роль ШІ у виконанні роботи. Навести 2–3 конкретні приклади запитів (промптів) та проаналізувати отримані відповіді. Наприклад, як ШІ допоміг структурувати дані задачі, пояснив логіку функції zip() для ітерації по списках ймовірностей або допоміг інтерпретувати отриманий результат.

6. Висновки.

Розділ має містити підсумок проведеного дослідження, розділений на два аспекти:

а) технічний висновок:

— провести компаративний аналіз: підтвердити збіжність результатів ручного розрахунку (Excel) та програмного коду;

— інтерпретувати результат: вказати, яка гіпотеза стала найбільш імовірною після настання події, і пояснити, чому так сталося (математичне обґрунтування);

б) рефлексія щодо використання ШІ.

Критично оцінити ефективність допомоги асистента. Описати, чи сприяв ШІ глибшому розумінню різниці між апріорною та апостеріорною ймовірностями, чи його роль обмежилася лише технічним генеруванням коду.

Завдання для самостійного виконання

Завдання. Фабрика виробляє електронні компоненти на трьох виробничих лініях: X, Y та Z. Відомі частки виробництва кожної лінії та відсоток браку для кожної з них.

Лінія X: виробляє 25 % усієї продукції, ймовірність браку — 3 %.

Лінія Y: виробляє 40 % усієї продукції, ймовірність браку — 2 %.

Лінія Z: виробляє 35 % усієї продукції, ймовірність браку — 5 %.

Уся продукція надходить на спільний склад. Навмання вибирають один виріб. Необхідно обчислити:

1. Повну ймовірність того, що вибраний компонент є бракованим (подія A).

2. Апостеріорну ймовірність того, що цей (уже відомо, що бракований) компонент був вироблений саме на лінії X, використовуючи формулу Байєса.

Контрольні запитання

1. Що таке повна група несумісних подій (гіпотез)?

2. Сформулюйте умову, за якої застосовується формула повної ймовірності.

3. У чому полягає суть теореми Байєса?

4. Що таке апріорна та апостеріорна ймовірності?

5. Наведіть приклад практичної задачі з медицини, де може використовуватися теорема Байєса.

6. Як програмно подати набір гіпотез та умовних імовірностей у Python?

7. Чому знаменник у формулі Байєса є повною ймовірністю?

8. Опишіть, як змінюється впевненість у гіпотезі після отримання нових даних.

9. Чи може апостеріорна ймовірність бути меншою за апріорну? Якщо так, то за яких умов?

10. Як ШІ може допомогти уникнути «помилки базового відсотка» (base rate fallacy) під час інтуїтивного оцінювання ймовірностей?

11. Чому сума всіх апостеріорних ймовірностей $\sum P(H_i|A)$ завжди має дорівнювати 1? Як це використовується для верифікації розрахунків?

12. Що станеться з апостеріорною ймовірністю гіпотези H_k, якщо умовна ймовірність події для неї $P(A|H_k) = 0$? Поясніть логічний зміст цього результату.

13. Які переваги використання списків (або масивів) у Python порівняно з окремими змінними, якщо кількість гіпотез значно збільшиться (наприклад, до 50)?

14. Як за допомогою ШІ-асистента можна згенерувати тестові дані (mock data) для перевірки програми на граничних випадках?

15. Які типові помилки можуть виникнути при зіставленні списків апріорних $P(H_i)$ та умовних $P(A|H_i)$ імовірностей у коді (наприклад, проблема індексації)?

Практична робота № 3
СТАТИСТИЧНИЙ АНАЛІЗ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ

Мета роботи:

— навчитися проводити первинне оброблення статистичних даних (побудова варіаційних та інтервальних рядів);

— опанувати методи розрахунку числових характеристик вибірки та параметрів групування (правило Стерджеса);

— освоїти техніки візуалізації розподілів (гістограми, полігони частот, емпіричні функції) засобами Python;

— розвинути навички використання бібліотек numpy, pandas, matplotlib та генеративного ШІ для автоматизації статистичного аналізу.

Компетентності, що формуються:

— здатність структурувати «сирі» дані та перетворювати їх на інформативні статистичні ряди;

— уміння вибирати коректний тип візуалізації залежно від характеру даних;

— навички програмної реалізації статистичних алгоритмів мовою Python;

— ефективне використання ШІ-асистентів для генерації коду, налагодження та інтерпретації форми розподілу.

Необхідне програмне забезпечення та бібліотеки:

1. Середовище розроблення: Google Colab, Jupyter Notebook або будь-яке інше інтегроване середовище розроблення (IDE) для Python (наприклад, VS Code, PyCharm).
2. Python версії 3.8 або новішої з бібліотеками: numpy, matplotlib, pandas.
3. MS Excel для проведення аналітичних розрахунків та перевірки.

Теоретичні відомості

Варіаційний ряд та розподіл частот

Для первинного аналізу вибірки необхідно впорядкувати дані та визначити частоту появи кожного значення.

Варіаційний ряд – це послідовність значень вибірки, записаних у порядку зростання ($x_1 \leq x_2 \leq \cdots \leq x_n$).

Частота (n_i) – число, яке показує, скільки разів значення $\boldsymbol{x_i}$ зустрічається у вибірці. Сума всіх частот дорівнює обсягу вибірки $\boldsymbol{N}$.

Відносна частота (w_i) – відношення частоти до загального обсягу вибірки: $w_i = \frac{n_i}{N}$. Сума відносних частот завжди дорівнює 1.

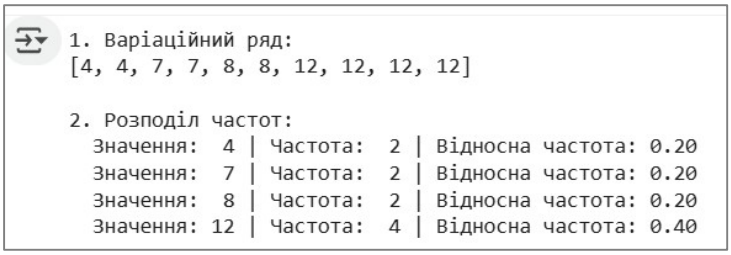
Реалізація в Python

Для оброблення дискретних даних доцільно використовувати вбудовані засоби мови. Функція *sorted*() впорядковує масив, а клас *Counter* з модуля *collections* дає змогу миттєво підрахувати частоти.

Приклад реалізації:

```
import collections
# Вхідні дані
data = [7, 4, 4, 8, 12, 12, 12, 7, 8, 12]
N = len(data)
# 1. Створення варіаційного ряду
sorted_data = sorted(data)

# 2. Обчислення частот: Counter створює словник {значення: частота}
frequency = collections.Counter(sorted_data)
# 3. Обчислення відносних частот

relative_frequency = {key: value / N for key, value in frequency.items()}
# --- Виведення результатів для цього блока ---
print("1. Варіаційний ряд:")
print(sorted_data)
print("\n2. Розподіл частот:")
for value, freq in sorted(frequency.items()):
    rel_freq = relative_frequency[value]
    print(f"   Значення: {value:2} | Частота: {freq:2} | Відносна частота:
{rel_freq:.2f}")
```

Результат роботи програми:



```
1. Варіаційний ряд:
[4, 4, 7, 7, 8, 8, 12, 12, 12, 12]

2. Розподіл частот:
  Значення:  4 | Частота:  2 | Відносна частота: 0.20
  Значення:  7 | Частота:  2 | Відносна частота: 0.20
  Значення:  8 | Частота:  2 | Відносна частота: 0.20
  Значення: 12 | Частота:  4 | Відносна частота: 0.40
```

Реалізація в MS Excel

Варіаційний ряд (введіть дані в стовпець, виділіть його та на вкладці *Data* натисніть *Sort A to Z*).

Частоти (використовуйте функцію *COUNTIF* для підрахунку кількості кожного унікального значення, наприклад: *=COUNTIF(A1:A10; C1),* де $A1{:}\,A100$ – діапазон вибірки, а $C1$ – унікальне значення).

Відносні частоти (поділіть отриману частоту на загальну кількість елементів (функція *COUNT*).

Інтервальний ряд (групування даних)

Для неперервних випадкових величин або великих вибірок дані групують в інтервали. Оптимальна кількість інтервалів (k) визначається за правилом Стерджеса:

$$k \approx 1 + 3.322 \cdot log_{10}(N).$$

Ширина інтервалу (h) розраховується як відношення розмаху вибірки ($R = x_{max} - x_{min}$) до кількості інтервалів:

$$h = \frac{R}{k}.$$

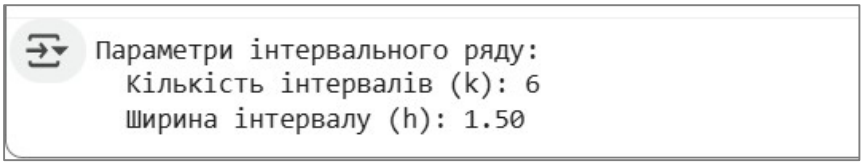
Реалізація в Python

Для математичних операцій, як-от логарифм, зручно використовувати бібліотеку *numpy*.

Приклад коду:

```
import numpy as np
# Припустимо, ми працюємо з більшим набором даних
data = [7, 4, 4, 8, 12, 12, 12, 7, 8, 12, 8, 12, 4, 12, 12, 4, 12, 4, 12, 12,
6, 9, 5, 3, 6, 6, 9, 3, 5, 6, 9, 5, 6, 6, 9, 6, 9, 6, 6, 6]
N = len(data)
R = max(data) - min(data)

# Визначаємо кількість інтервалів за формулою Стерджеса
k = round(1 + 3.322 * np.log10(N))

# Розраховуємо ширину кожного інтервалу
h = R / k

# --- Виведення результатів для цього блока ---
print("Параметри інтервального ряду:")
print(f"  Кількість інтервалів (k): {k}")
print(f"  Ширина інтервалу (h): {h:.2f}")
```

Результат роботи програми:



Реалізація в MS Excel

1. Обчисліть N, x_{max}, x_{min} за допомогою функцій $COUNT$, MAX, MIN відповідно.

2. Розрахуйте k за формулою Стерджеса, записавши в рядку формул такий вираз: «$= 1 + 3{,}322 * LOG10(N)$». Округліть результат до цілого числа за допомогою функції $ROUND$.

3. Розрахуйте h, записавши в рядку формул « $= (MAX - MIN)/k$».

Графічне подання даних

Візуалізація варіаційного ряду дає змогу оцінити відповідність емпіричного розподілу теоретичним законам (найчастіше – нормальному розподілу).

Основні інструменти візуалізації:

1. *Гістограма* – стовпчаста діаграма, що відображає щільність розподілу частот за інтервалами.

2. *Полігон частот* – ламана лінія, що з'єднує середини інтервалів, демонструючи неперервну тенденцію розподілу.

При аналізі графіків (гістограми та полігону) ключовими є характеристики форми розподілу:

1. *Асиметрія* (Skewness) – характеризує ступінь зміщення розподілу відносно математичного сподівання. Якщо коефіцієнт асиметрії $A_s \approx 0$, розподіл вважається *симетричним*. *Додатне* значення вказує на *правосторонню* асиметрію (довгий правий «хвіст»), *від'ємне* – на *лівосторонню*.

2. *Ексцес* (Kurtosis) – міра гостровершинності графіка щільності ймовірності порівняно з нормальним розподілом. Додатний ексцес ($E_k > 0$) свідчить про надмірну концентрацію значень навколо середнього («гострий пік»), від'ємний ($E_k < 0$) – про більш плоский розподіл.

Реалізація в Python

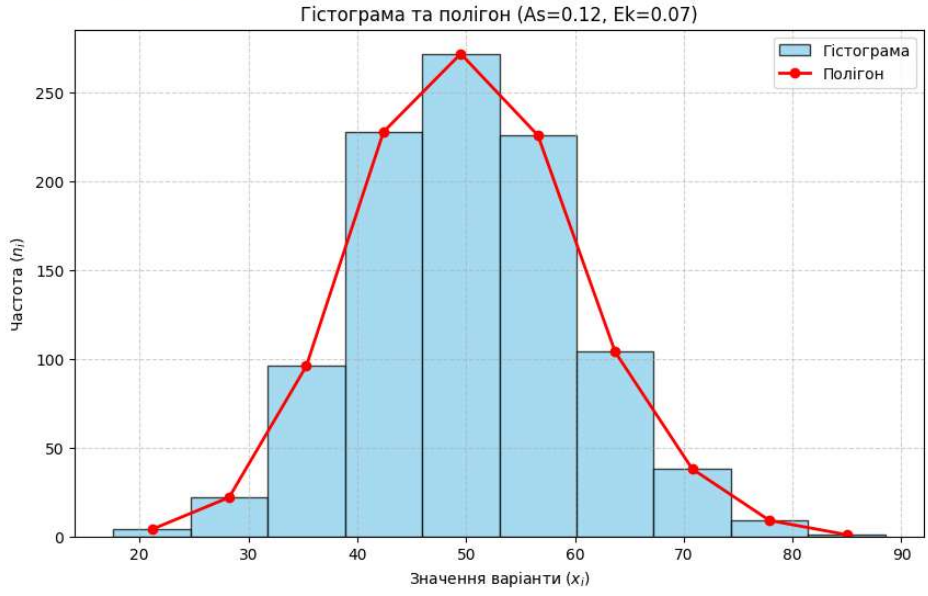
Для комплексної візуалізації використовується бібліотека $matplotlib$. Особливість реалізації полягає в розбіжності координат для побудови графіків: функція $plt.hist$ автоматично розбиває дані на інтервали ($bins$) та будує стовпці за їх межами, тоді як полігон частот має будуватися за геометричними центрами цих інтервалів. Тому необхідно програмно «перехопити» розраховані межі, обчислити їх середини та з'єднати ці точки лінією. Для точного розрахунку числових характеристик форми (A_s, E_k) використовується модуль $scipy.stats$.

Приклад коду:

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import skew, kurtosis
# --- БЛОК 1: Підготовка даних ---
# Генеруємо 1000 випадкових чисел (нормальний розподіл)
# loc=50 (середнє), scale=10 (стандартне відхилення)
np.random.seed(42) # Фіксація генератора для відтворюваності
data = np.random.normal(loc=50, scale=10, size=1000)
# Розрахунок кількості інтервалів (k) за правилом Стерджеса
k = int(1 + 3.322 * np.log10(len(data)))
print(f"Кількість даних: {len(data)}, Кількість інтервалів: {k}")
# --- БЛОК 2: Візуалізація ---
plt.figure(figsize=(10, 6))
# 1. Побудова гістограми
# counts - висота стовпців, bins - межі інтервалів
counts, bins, patches = plt.hist(data, bins=k, edgecolor='black', alpha=0.75,
label='Гістограма', color='skyblue')
# 2. Побудова полігону частот
# Знаходимо центри інтервалів
bin_centers = 0.5 * (bins[:-1] + bins[1:])
# Будуємо червону лінію по центрах
plt.plot(bin_centers,  counts,  color='red',  marker='o',  linestyle='-',
linewidth=2, label='Полігон')
# --- БЛОК 3: Статистика ---
# Розрахунок характеристик форми
sk_val = skew(data)
ku_val = kurtosis(data)
print(f"Коефіцієнт асиметрії (Skewness): {sk_val:.4f}")
print(f"Коефіцієнт ексцесу (Kurtosis):  {ku_val:.4f}")
# --- БЛОК 4: Оформлення ---
plt.title(f'Гістограма та полігон (As={sk_val:.2f}, Ek={ku_val:.2f})')
plt.xlabel('Значення варіанти ($x_i$)')
plt.ylabel('Частота ($n_i$)')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```

Результат роботи програми:

```
Кількість даних: 1000, Кількість інтервалів: 10
Коефіцієнт асиметрії (Skewness): 0.1168
Коефіцієнт ексцесу (Kurtosis):  0.0662
```



Реалізація в MS Excel

У середовищі електронних таблиць комплексний аналіз форми розподілу виконується у два етапи: розрахунок числових характеристик за допомогою вбудованих функцій та візуалізація через інструменти побудови діаграм.

1. Розрахунок показників форми.

Для кількісного оцінювання відхилення емпіричного розподілу від нормального закону використовуються стандартизовані статистичні функції. Коефіцієнт асиметрії розраховується за допомогою функції $= SKEW(\text{діапазон})$, результат якої вказує на наявність та напрямок зміщення вершини розподілу. Для визначення гостровершинності графіка застосовується функція Excel «$= KURT(\text{діапазон})$», що дає змогу порівняти концентрацію значень навколо середнього з еталонним нормальним розподілом.

2. Побудова графіків розподілу.

Побудова гістограми найефективніше реалізується через інструментарій пакету аналізу. Для виклику функції необхідно перейти на вкладку $Data$ та у блоці $Analyze$ обрати пункт $Data\ Analysis$. У переліку інструментів слід вибрати $Histogram$.

У діалоговому вікні налаштувань процедури необхідно задати параметри вхідних даних:

— у полі $Input\ Range$ вказується посилання на масив вихідних емпіричних даних;

— у полі $Bin\ Range$ зазначається діапазон комірок із заздалегідь розрахованими верхніми межами інтервалів;

— обов'язковою умовою є активація опції $Chart\ Output$, що забезпечує автоматичну генерацію стовпчастої діаграми частот разом із таблицею розподілу.

Візуалізація *полігону частот* виконується на основі сформованої таблиці розподілу. Оскільки полігон відображає функціональну залежність частоти від центру інтервалу, необхідно додатково створити стовпець значень середин інтервалів (x_i). Для графічного подання використовується тип діаграми $Scatter\ with\ Straight\ Lines\ and\ Markers$. При побудові вісь абсцис (X) має відповідати розрахованим серединам інтервалів, а вісь ординат (Y) – відповідним абсолютним частотам (n_i).

Емпірична функція та накопичені частоти

Емпірична функція розподілу є вибірковим аналогом теоретичної функції розподілу ймовірностей $F(x)$. Вона дає змогу оцінити, яка частка елементів вибірки не перевищує задане значення.

Накопичена частота (N_i) – це сума частот усіх варіант, що менші або дорівнюють поточному значенню x_i. Вона показує, скільки спостережень накопичено до певного моменту ($N_i = \sum_{j=1}^{i} n_j$).

Емпірична функція розподілу ($F_n(x)$) – це ступінчата функція, що зростає від 0 до 1, значення якої дорівнює відносній накопиченій частоті:

$$F_n(x) = \frac{\text{кількість елементів} \leq x}{n}.$$

Реалізація в Python

Побудова графіка емпіричної функції має свої особливості, оскільки математично це кусково-стала функція (зберігає постійне значення на інтервалі та робить «стрибок» у точці появи нового значення).

Алгоритм реалізації:

1. **Сортування (Sorting) –** функція $F(x)$ визначена для впорядкованих даних (варіаційного ряду). Тому масив даних обов'язково сортується за зростанням.

2. **Генерація осі Y** – для кожного ***i***-го елемента відсортованого ряду значення функції дорівнює i/n, де n – загальний обсяг вибірки. Створюємо послідовність від $1/n$ до 1.

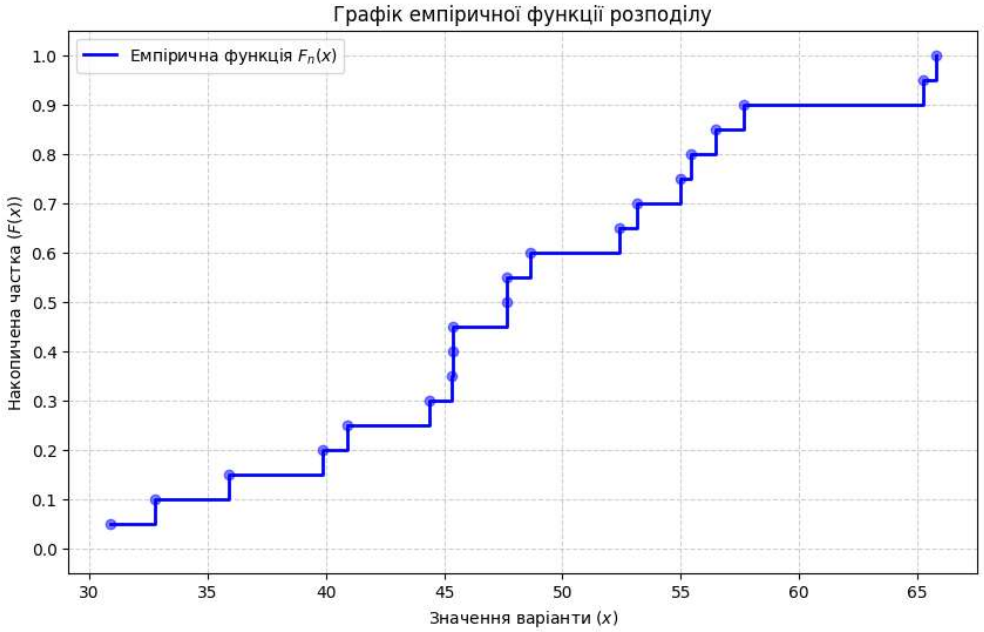
3. **Візуалізація сходинок –** використовується функція $plt.step()$. Важливим є параметр $where = 'post'$, який вказує, що «сходинка» малюється *після* точки даних (горизонтальна лінія, потім вертикальний стрибок), що відповідає математичному визначенню функції розподілу.

Приклад коду:

```
import matplotlib.pyplot as plt
import numpy as np
# --- БЛОК 1: Підготовка даних ---
# Генеруємо тестові дані (або використовуємо ваші реальні дані)
np.random.seed(42)
data = np.random.normal(loc=50, scale=10, size=20) # Мала вибірка для наочності сходинок
# 1. Сортування даних (Варіаційний ряд)
# Функція F(x) показує накопичення, тому порядок даних критично важливий.
sorted_data = np.sort(data)
# 2. Розрахунок накопичених частот (вісь Y)
# n - загальна кількість спостережень
n = len(sorted_data)
# Створюємо масив [1, 2, 3, ..., n] і ділимо на n.
# Отримуємо частки: [1/n, 2/n, ..., 1.0]
y_values = np.arange(1, n + 1) / n
# --- БЛОК 2: Візуалізація ---
plt.figure(figsize=(10, 6))
# 3. Побудова ступінчастого графіка
# plt.plot() з'єднав би точки похилими лініями, що є помилкою для F(x).
# plt.step() малює правильні сходинки.
# where='post': значення змінюється стрибком у точці даних.
plt.step(sorted_data, y_values, where='post', label='Емпірична функція $F_n(x)$', color='blue', linewidth=2)
# Додаткові налаштування для наочності
plt.plot(sorted_data, y_values, 'o', color='blue', alpha=0.5) # Позначаємо точки стрибків маркерами
plt.title('Графік емпіричної функції розподілу')
plt.xlabel('Значення варіанти ($x$)')
plt.ylabel('Накопичена частка ($F(x)$)')
plt.grid(True, linestyle='--', alpha=0.6)
# Фіксація меж та кроку сітки по осі Y для зручності читання частот (0.0, 0.1, ... 1.0)
```

```
plt.yticks(np.arange(0, 1.1, 0.1))
plt.ylim(-0.05, 1.05) # Невеликий відступ зверху і знизу
plt.legend()
plt.show()
```

Результат роботи програми:



Реалізація в MS Excel

У середовищі електронних таблиць побудова емпіричної функції розподілу вимагає попередньої підготовки даних, оскільки стандартний інструментарій не має спеціалізованого типу діаграми $Step\ Chart$. Реалізація виконується через апроксимацію точковою діаграмою.

1. Підготовка варіаційного ряду.

Перед побудовою необхідно сформувати таблицю, що відображає залежність між значенням варіанти та накопиченою часткою:

а) сортування даних – розмістіть вихідні дані в одному стовпці. Виділіть діапазон та на вкладці $Data$ оберіть інструмент $Sort$ (від мінімального до максимального). Це є обов'язковою умовою для коректного відображення функції $F(x)$;

б) індексація (i): у сусідньому стовпці пронумеруйте елементи вибірки від 1 до n (де n – обсяг вибірки);

в) розрахунок накопиченої частоти ($F_n(x)$): у наступному стовпці обчисліть відносну частоту за виразом, який треба записати у рядок формул Excel:

$$= \text{Поточний\_номер} \ / \ \text{Загальна\_кількість}$$

(наприклад: $= B2/COUNT(\$A\$2{:}\$A\$21)$). Значення мають зростати від $1/n$ до 1.

2. Побудова графіка.

Для візуалізації використовується тип діаграми, що показує функціональну залежність Y від X:

— виділіть два стовпці: відсортовані значення даних (вісь X) та розраховані накопичені частоти (вісь Y);

— перейдіть на вкладку $Insert$, у групі $Charts$ натисніть кнопку $Insert\ Scatter$;

— виберіть підтип $Scatter\ with\ Straight\ Lines\ and\ Markers$.

Стандартна діаграма Excel з'єднує точки найкоротшим шляхом (похилими лініями), тоді як теоретична емпірична функція є ступінчастою. Для навчальних цілей така апроксимація є допустимою, однак слід розуміти, що графік демонструє загальну тенденцію зростання накопиченої ймовірності.

Щільність частот

Щільність відносної частоти – це ключовий показник, який використовується для побудови нормованих гістограм. На відміну від звичайної гістограми, де висота стовпця відображає кількість елементів (n_i), у нормованій гістограмі висота стовпця розраховується таким чином, щоб загальна площа всіх стовпців дорівнювала 1.

Це дає можливість:

— порівнювати розподіли, побудовані на вибірках різного обсягу;

— накладати на гістограму графік теоретичної функції щільності розподілу (наприклад, криву нормального розподілу) для візуальної перевірки гіпотез.

Щільність частоти (d_i) для i-го інтервалу розраховується як відношення відносної частоти до ширини інтервалу:

$$d_i = \frac{n_i}{N \cdot h},$$

де n_i – абсолютна частота (кількість елементів) в i-му інтервалі;

N – загальний обсяг вибірки ($\sum n_i$);

h – ширина інтервалу (крок).

Реалізація в Python

Побудова гістограми щільності виконується бібліотекою $matplotlib$. Ключовим моментом є використання параметра $density = True$ у функції $plt.hist$, який змінює принцип побудови графіка:

— автоматична нормалізація – функція самостійно перераховує висоту кожного стовпця, ділячи частоту інтервалу на загальний обсяг вибірки (N) та на ширину інтервалу (h);

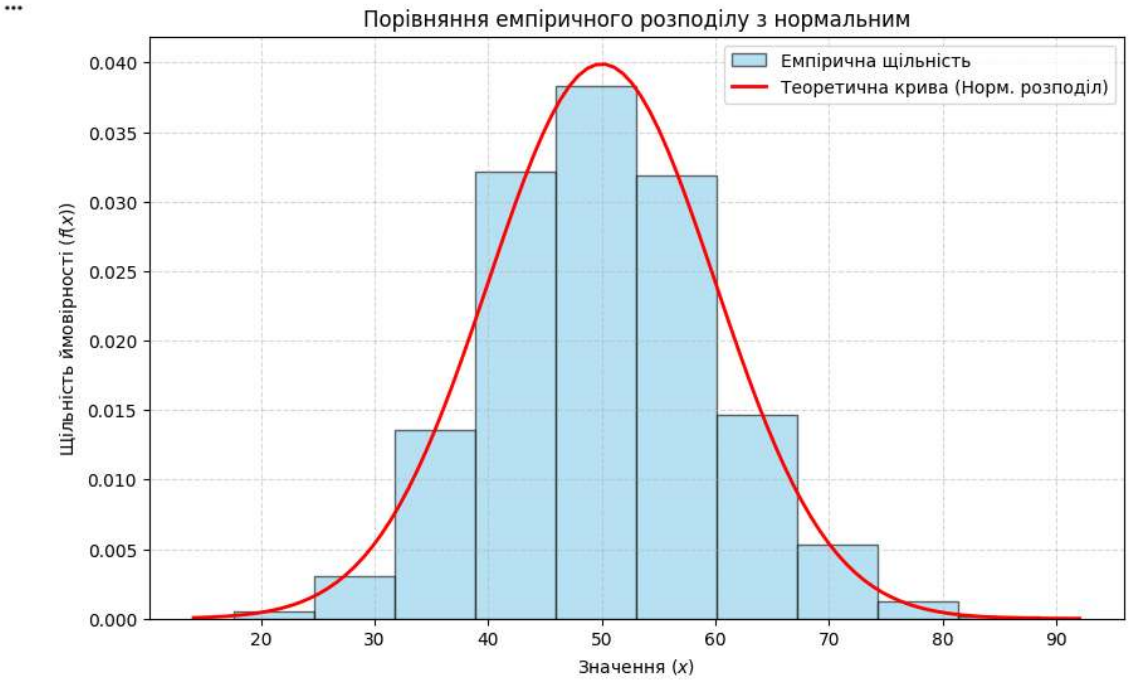
— нормування площі – сумарна площа всіх стовпців отриманої гістограми завжди дорівнює одиниці. Це дає змогу інтерпретувати вісь ординат (Y) як щільність імовірності;

— теоретичний аналіз – нормований вигляд дає змогу коректно накласти поверх гістограми графік теоретичної функції (наприклад, криву нормального розподілу) для візуального оцінювання відповідності емпіричних даних теоретичному закону.

Приклад реалізації:

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm  # Імпорт для теоретичної кривої
# --- БЛОК 1: Генерація даних ---
# Генеруємо 1000 чисел із нормальним розподілом
# Середнє (mu) = 50, Стандартне відхилення (sigma) = 10
np.random.seed(42)
mu, sigma = 50, 10
data = np.random.normal(mu, sigma, 1000)
# Розрахунок кількості інтервалів за правилом Стерджеса
k = int(1 + 3.322 * np.log10(len(data)))
plt.figure(figsize=(10, 6))
# --- БЛОК 2: Гістограма щільності ---
# density=True перетворює вісь Y з "кількості" на "щільність"
# alpha=0.6 робить стовпці напівпрозорими
count, bins, ignored = plt.hist(data, bins=k,
density=True, alpha=0.6, color='skyblue', edgecolor='black',
label='Емпірична щільність')
# --- БЛОК 3: Теоретична крива нормального розподілу ---
# Створюємо масив точок X від мінімального до максимального значення
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
# Розраховуємо теоретичну щільність (PDF - Probability Density Function)
# Використовуємо параметри нашої вибірки (mu, sigma)
p = norm.pdf(x, mu, sigma)
# Малюємо криву поверх гістограми
plt.plot(x, p, 'r', linewidth=2, label='Теоретична крива (Норм. розподіл)')
# --- БЛОК 4: Оформлення ---
plt.title('Порівняння емпіричного розподілу з нормальним')
plt.xlabel('Значення ($x$)')
plt.ylabel('Щільність імовірності ($f(x)$)')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

Результат роботи програми:



Реалізація в MS Excel

Excel не має вбудованої функції для автоматичної побудови гістограми щільності, тому розрахунок висоти стовпців виконується вручну.

1. Підготовка даних.

Сформуйте таблицю частотного розподілу, що містить межі інтервалів та частоти (n_i). У окремі клітинки винесіть константи: загальний обсяг вибірки (N) та ширину інтервалу (h).

2. Розрахунок щільності.

Створіть новий стовпець «Щільність». Введіть у рядок формул для першого інтервалу такий вираз:

$$= \text{Частота} \,/\, (\text{Обсяг\_вибірки} * \text{Ширина\_інтервалу}).$$

Приклад: якщо частота у комірці $C2$, обсяг N у $F1$, а ширина h у $F2$, вираз у рядку формул має такий вигляд:

$$= C2/(\$F\$1 * \$F\$2).$$

Використовуйте абсолютні посилання (символ «$» для комірок з N та h, щоб при копіюванні формули вниз посилання на них не зміщувалися.

3. Побудова діаграми:

— виділіть стовпець підписів інтервалів (вісь X) та розрахований стовпець «Щільність» (вісь Y);

— перейдіть на вкладку $Insert \rightarrow Column\ Chart \rightarrow Clustered\ Column$;

— зробіть форматування під гістограму: клікніть правою кнопкою миші на стовпцях діаграми $\rightarrow$ $Format\ Data\ Series$. Встановіть параметр $Gap\ Width$ на $0\,\%$ або $5\,\%$, щоб стовпці торкалися один одного, імітуючи неперервний розподіл.

Порядок виконання роботи

Теоретична підготовка

1. Опрацювати фундаментальні поняття: варіаційний ряд, частоти та відносні частоти, інтервальний розподіл.
2. Вивчити умови застосування правила Стерджеса для визначення оптимальної кількості інтервалів.
3. Підготувати інструментарій: перевірити налаштування середовища Python та створити шаблон таблиці в MS Excel для розрахунків.
4. Вибрати індивідуальний варіант завдання згідно зі списком.

Аналітичне розв'язання (MS Excel / Вручну)

На цьому етапі слід виконати розрахунок еталонних значень (в електронній таблиці) для подальшої перевірки коду та розуміння природи даних.

1. Первинне оброблення даних:

— побудувати дискретний варіаційний ряд (впорядкувати вибірку за зростанням);

— розрахувати частоти та відносні частоти для кожного унікального значення.

2. Групування даних (інтервальний ряд):

— обчислити розмах вибірки та визначити оптимальну кількість інтервалів (крок) за правилом Стерджеса;

— сформувати інтервали та підрахувати частоти потрапляння випадкової величини в кожен з них.

3. Підготовка до візуалізації:

— розрахувати накопичені частоти для побудови кумуляти;

— підготувати табличні дані (середини інтервалів, висоти стовпчиків) для побудови гістограми та полігону частот.

Програмна реалізація та використання ШІ

Створити новий скрипт (.py) або блокнот (.ipynb) та виконати такі дії:

1. Підготовка даних:

— створити масиви (або використати pandas.Series) для зберігання вхідної вибірки згідно з варіантом;

— імпортувати необхідні бібліотеки (numpy, matplotlib, pandas).

2. Залучення ШІ-асистента:

— використати ШІ для оптимізації роботи над кодом;

— завдання для ШІ: попросити згенерувати код для автоматичного розрахунку кількості інтервалів за формулою Стерджеса АБО попросити допомоги у налаштуванні стилізації графіків бібліотеки matplotlib (підписи осей, легенда, сітка).

3. Реалізація алгоритму:

— написати (або адаптувати згенерований) код, який будує варіаційний ряд та розраховує інтервальні характеристики;

— реалізувати функції для побудови графіків: гістограми частот, полігону та емпіричної функції розподілу (кумуляти).

4. Виведення результатів:

— забезпечити виведення основних числових характеристик (min/max значення, крок інтервалу) у текстовому вигляді;

— відобразити графіки в одній області (subplots) або послідовно.

Аналіз результатів та формування висновків

1. Верифікація розрахунків:

— порівняти числові значення (межі інтервалів, частоти), отримані в програмі (Python), з результатами попереднього етапу (MS Excel);

— переконатися, що візуальний вигляд графіків у Python відповідає табличним даним з Excel.

2. Інтерпретація результатів:

— сформулювати фінальні, обґрунтовані висновки щодо форми розподілу даних на основі візуалізацій (симетричність, наявність «хвостів», схожість на нормальний розподіл);

— оцінити, наскільки коректно вибрана кількість інтервалів відображає структуру даних.

3. Рефлексія щодо використання ШІ: у висновках коротко описати свій досвід взаємодії з ШІ-асистентом: чи коректно він застосував правило Стерджеса, чи допомогли його підказки покращити візуалізацію графіків.

Вимоги до звіту та його вміст

Звіт є підсумковим документом, що фіксує результати проведеного дослідження, його структура має містити такі обов'язкові компоненти:

1. Титульна сторінка.

Оформлюється згідно з установленим зразком. Має містити назву дисципліни, номер роботи, дані здобувача освіти та викладача.

2. Тема, мета роботи та номер варіанта.

Формулюються чітко відповідно до методичних вказівок.

3. Постановка завдання.

Необхідно повністю скопіювати умову завдання зі свого індивідуального варіанта (вихідні дані вибірки).

4. Результати виконання.

Це основний розділ звіту, що складається з трьох частин:

а) розрахунки (MS Excel / Вручну):

— таблиця частотного розподілу (інтервали, частоти, накопичені частоти);

— розрахунок числових характеристик форми розподілу: асиметрії (A_s) та ексцесу (E_k);

— таблиця розрахунку щільності відносної частоти;

б) код програми: повний програмний код на Python з коментарями, що реалізує розрахунки та побудову всіх типів графіків;

в) результати роботи програми:

— скріншот текстового виведення розрахованих коефіцієнтів ($Skewness$, $Kurtosis$);

— графік 1: гістограма частот, суміщена з полігоном розподілу;

— графік 2: емпірична функція розподілу (ступінчастий графік);

— графік 3: гістограма щільності ймовірності з накладеною теоретичною кривою нормального розподілу.

5. Аналіз використання ШІ-асистента.

У цьому розділі здобувач освіти має продемонструвати навички критичної взаємодії з технологіями штучного інтелекту:

а) вибір програмного інструментарію ШІ. Вказати повну назву та версію використаної моделі (наприклад, Gemini 1.5 Flash, ChatGPT 4.0, Claude 3.5);

б) роль ШІ у виконанні роботи. Навести 2–3 конкретні приклади запитів (промптів) та проаналізувати отримані відповіді. Наприклад, як ШІ допоміг розібратися з параметром $density = True$ у бібліотеці $matplotlib$, пояснив різницю між гістограмою та полігоном або допоміг виправити помилки при побудові ступінчастої функції.

6. Висновки.

Розділ має містити підсумок проведеного дослідження, поділений на два аспекти:

а) технічний висновок.

Необхідно провести аналіз форми розподілу. На основі отриманих графіків та коефіцієнтів $(A_s,\ E_k)$ визначити тип розподілу (симетричний/асиметричний, гостроверхий/плосковерхий). Зробити аргументований висновок про те, наскільки емпіричні дані узгоджуються з теоретичним нормальним законом розподілу;

б) рефлексія щодо використання ШІ.

Критично оцінити ефективність допомоги асистента. Описати, чи сприяв ШІ глибшому розумінню статистичних концепцій (таких як щільність імовірності або емпірична функція), чи його роль обмежилася технічним пришвидшенням написання коду візуалізації.

Завдання для самостійного виконання

Завдання 1. Варіаційний та інтервальний ряд, частоти та гістограма.

Дані для оброблення: 7, 4, 4, 8, 12, 12, 12, 7, 8, 12, 8, 12, 4, 12, 12, 4, 12, 4, 12, 12, 6, 9, 5, 3, 6, 6, 9, 3, 5, 6, 9, 5, 6, 6, 9, 6, 9, 6, 6, 6.

1. Побудуйте варіаційний ряд.
2. Складіть розподіл частот та відносних частот.
3. Розрахуйте ширину інтервалу за правилом Стерджеса та побудуйте інтервальний ряд.
4. Створіть гістограму частот.

Завдання 2. Емпірична функція розподілу та полігон частот.

Дані для оброблення: 2.41, 2.62, 2.73, 2.52, 2.54, 2.41, 2.81, 2.53, 2.64, 2.61, 2.72, 2.45, 2.52, 2.1, 2.64, 2.52, 2.5, 2.33, 2.24, 2.4, 2.72, 2.51, 2.4, 2.61, 2.42, 2.43, 2.65, 2.54, 2.35, 2.54, 2.62, 2.9, 2.75, 2.24, 2.65, 2.45, 2.53, 2.32, 2.24, 2.55.

1. Розрахуйте ширину інтервалу за правилом Стерджеса та складіть інтервальний ряд.
2. Побудуйте полігон частот.
3. Знайдіть емпіричну функцію розподілу та створіть її графік.

Завдання 3. Групований розподіл і статистичний аналіз.

Дані для оброблення: 52, 54, 70, 73, 75, 52, 82, 93, 95, 82, 60, 62, 50, 79, 92, 91, 81, 53, 80, 70, 52, 82, 93, 95, 82, 55, 60, 62, 50, 90, 52, 54, 70, 73, 75, 52, 82, 93, 95, 82, 82, 82, 93, 55, 60, 62, 50, 79, 28, 100.

1. Запишіть впорядкований розподіл частот та відносних частот.
2. Розрахуйте ширину інтервалу за правилом Стерджеса та побудуйте згрупований розподіл частот.
3. Побудуйте полігон накопичених частот та графік щільності відносних частот.

Контрольні запитання

1. Що таке варіаційний ряд і чим він відрізняється від інтервального?
2. Для чого використовується правило Стерджеса?

3. У чому принципова різниця між гістограмою та полігоном частот?
4. Що показує емпірична функція розподілу і як вона будується?
5. Що таке накопичена частота?
6. Яка функція в бібліотеці matplotlib відповідає за побудову гістограми?
7. Як вибір кількості інтервалів впливає на вигляд гістограми?
8. Наведіть приклад даних, для яких краще будувати полігон, а не гістограму.
9. Що таке щільність відносної частоти?
10. Як ШІ може допомогти інтерпретувати форму розподілу, отриману на гістограмі?
11. Що характеризують коефіцієнти асиметрії (A_s) та ексцесу (E_k) і про що свідчить їх значне відхилення від нуля?
12. Чому при побудові гістограми щільності (density plot) сумарна площа всіх стовпців завжди дорівнює одиниці?
13. З якою метою на гістограму емпіричного розподілу накладається теоретична крива нормального закону?
14. Чому попереднє сортування даних (створення варіаційного ряду) є обов'язковим технічним етапом для розрахунку емпіричної функції розподілу?
15. У чому полягає особливість (обмеження) побудови графіка емпіричної функції в MS Excel порівняно з бібліотекою matplotlib?
16. Який аргумент функції plt.hist() у Python необхідно активувати, щоб висота стовпців відображала не кількість елементів, а щільність імовірності?
17. Як наявність двох вершин (бімодальність) на гістограмі впливає на висновок про відповідність емпіричних даних нормальному закону розподілу?

Практична робота № 4
ОБЧИСЛЕННЯ ОСНОВНИХ СТАТИСТИЧНИХ ХАРАКТЕРИСТИК ТА ГРАФІЧНЕ ПОДАННЯ ДАНИХ

Мета роботи:

— закріпити теоретичні знання про вибірковий метод та числові характеристики вибірки (міри центральної тенденції та розсіювання);

— навчитися аналітично та програмно знаходити основні статистичні показники: середнє, дисперсію, моду, медіану;

— опанувати методи побудови варіаційних рядів та таблиць частот для згрупованих даних;

— вдосконалити навички візуалізації даних (полігони, гістограми) для оцінювання їх розподілу та навчитися будувати графік емпіричної функції розподілу.

Компетентності, що формуються:

— здатність обчислювати ключові міри центральної тенденції (середнє, мода, медіана) та розсіювання (дисперсія, розмах) для кількісного аналізу вибірки;

— уміння структурувати вибіркові дані, формуючи варіаційні та інтервальні ряди розподілу;

— навички програмної реалізації графічних моделей (полігони, гістограми, емпірична функція розподілу) засобами бібліотек numpy, pandas, matplotlib;

— ефективне використання ШІ-асистентів для генерації коду, його налагодження та глибшого пояснення статистичних закономірностей.

Необхідне програмне забезпечення та бібліотеки:

1. Середовище розроблення: Google Colab, Jupyter Notebook або будь-яке інше інтегроване середовище розроблення (IDE) для Python (наприклад, VS Code, PyCharm).
2. Python версії 3.8 або новішої з бібліотеками: numpy, scipy, matplotlib, pandas.
3. MS Excel для проведення паралельних аналітичних розрахунків та верифікації результатів.

Теоретичні відомості

Статистичний аналіз вибіркових даних базується на обчисленні числових характеристик, за допомогою яких можна описати властивості генеральної сукупності. За функціональним призначенням ці характеристики поділяються на міри центральної тенденції (вказують на центр розподілу значень випадкової величини) та міри розсіювання (кількісно характеризують варіабельність даних).

Характеристики положення (міри центральної тенденції)

Ці показники визначають типовий рівень ознаки у досліджуваній сукупності.

Вибіркове середнє ($\bar{x}$)

Цей показник є ключовою характеристикою центральної тенденції, що визначає узагальнене значення вибірки. Методика його розрахунку залежить від типу даних: для незгрупованих рядів середнє обчислюється як сума всіх варіант, поділена на їх загальну кількість. У випадку ж згрупованих даних (інтервального ряду) застосовується інший підхід – середнє визначається як зважена сума середин інтервалів, де роль вагових коефіцієнтів виконують відповідні частоти.

Формули обчислення:

— для незгрупованих даних

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i ;$$

— для інтервального варіаційного ряду (де x_i^* – середини інтервалів)

$$\bar{x} = \frac{\sum_{i=1}^{k} x_i^* \cdot n_i}{n}.$$

Реалізація в Python

Для автоматизації обчислень використовується бібліотека $numpy$. Для незгрупованих вибірок застосовується стандартна функція $mean()$, а для інтервальних рядів (де необхідно враховувати частоти) – функція $average()$ із параметром $weights$.

Приклад реалізації:

```
import numpy as np
# а) Для незгрупованих даних
data_ungrouped = [10, 12, 15, 18, 20]
mean_val = np.mean(data_ungrouped)
# б) Для згрупованих даних (інтервальний ряд)
midpoints = [12.5, 17.5, 22.5, 27.5, 32.5]  # Середини інтервалів (xi*)
frequencies = [5, 8, 12, 15, 5]             # Частоти (ni)
# Використання weights дає змогу обчислити зважене середнє
mean_weighted = np.average(midpoints, weights=frequencies)
print(f"Середнє (незгруповані): {mean_val}")
print(f"Середнє (згруповані): {mean_weighted:.2f}")
```

Результат роботи програми:

```
Середнє (незгруповані): 15.0
Середнє (згруповані): 23.28
```

Реалізація в MS Excel

Для *незгрупованих* даних застосовується стандартна статистична функція $= AVERAGE(\text{діапазон\_даних})$.

Для *згрупованих* даних (інтервальний ряд) розрахунок виконується поетапно, оскільки вбудована функція для зваженого середнього відсутня:

— сформувати допоміжний стовпець, що містить добутки середин інтервалів на відповідні частоти ($x_i^* \cdot n_i$);

— обчислити суму значень цього стовпця;

— отриману суму поділити на загальний обсяг вибірки (суму частот).

Медіана (Me)

Медіана – це структурна середня величина, яка ділить ранжований (впорядкований за зростанням) варіаційний ряд на дві рівні за обсягом частини. Це означає, що 50 % елементів вибірки є меншими за медіану, а 50 % – більшими.

Ключовою особливістю медіани є її робастність (стійкість) до аномальних викидів. На відміну від середнього арифметичного, медіана не змінюється під впливом екстремальних значень на краях розподілу, що робить її надійною характеристикою центру для асиметричних вибірок.

Алгоритм обчислення: якщо обсяг вибірки n непарний, то медіаною є варіанта, що знаходиться точно посередині ряду; якщо обсяг вибірки n

парний, то медіана визначається як середнє арифметичне двох центральних варіант.

Реалізація в Python

Для обчислення медіани в середовищі Python використовується функція $numpy.median()$. Вона автоматизує процес ранжування (сортування) даних та вибору центрального значення. Алгоритм функції самостійно адаптується до обсягу вибірки.

Приклад програми:

```
import numpy as np
# а) Непарна кількість елементів
data_odd = [10, 12, 15, 18, 20]
median_odd = np.median(data_odd)
print(f"Медіана (непарна к-ть): {median_odd}")
# б) Парна кількість елементів
data_even = [10, 12, 15, 18, 20, 22]
# Медіана = (15 + 18) / 2 = 16.5
median_even = np.median(data_even)
print(f"Медіана (парна к-ть): {median_even}")
```

Результат роботи програми:

```
Медіана (непарна к-ть): 15.0
Медіана (парна к-ть): 16.5
```

Реалізація в MS Excel

Для автоматизованого розрахунку медіани застосовується стандартна статистична функція «$= MEDIAN(\text{масив даних})$», алгоритм якої передбачає самостійну адаптацію до обсягу вибірки (n). Такий підхід дає змогу виконувати всі операції, включно з сортуванням даних та перевіркою парності, програмно, звільняючи користувача від необхідності попереднього ручного оброблення масиву.

Мода (Mo)

Мода – це значення ознаки, яке зустрічається у вибірці з найбільшою частотою. Це єдина міра центральної тенденції, яку можна застосовувати не лише до кількісних, а й до якісних (номінальних) даних.

Типологія розподілів за модою:

— унімодальний (має одне чітко виражене пікове значення);

— бімодальний/полімодальний (має два або більше піків з однаковою (або близькою) максимальною частотою);

— антимодальний (усі значення трапляються з однаковою частотою, тобто мода відсутня).

Реалізація в Python

Стандартна бібліотека $numpy$ не містить прямої функції для обчислення моди. Для цього використовується модуль $scipy.stats$, який дає змогу отримати як саме значення моди, так і кількість його повторень.

Приклад програми:

```
from scipy import stats
import numpy as np
# Вхідний масив (унімодальний приклад)
data = [12, 15, 15, 18, 20, 20, 20, 25]
# keepdims=True — параметр для коректного формату виведення
mode_info = stats.mode(data, keepdims=True)
# Результат повертається у вигляді об'єкта з двома масивами: mode та count
print(f"Значення моди: {mode_info.mode[0]}")
print(f"Частота появи: {mode_info.count[0]}")
```

Результат роботи програми:

```
Значення моди: 20
Частота появи: 3
```

Реалізація в MS Excel

У сучасних версіях табличного процесора для розрахунку моди передбачено дві спеціалізовані функції, вибір яких залежить від очікуваної структури розподілу:

— для унімодальних розподілів використовується функція «$= MODE.SNGL(\text{діапазон})$». Якщо у вибірці є кілька мод, функція поверне лише *першу* знайдену, ігноруючи інші;

— для полімодальних розподілів застосовується функція «$= MODE.MULT(\text{діапазон})$». Ця функція повертає вертикальний масив значень. У старих версіях Excel для її коректної роботи необхідно виділити діапазон комірок і натиснути комбінацію клавіш $Ctrl + Shift + Enter$ (формула масиву).

Розрахунок моди та медіани для інтервального ряду

Для згрупованих даних (інтервального розподілу) конкретні значення варіант невідомі, тому застосовується метод лінійної інтерполяції всередині визначеного інтервалу.

Формула медіани (Me)

Спочатку визначається медіанний інтервал – перший інтервал, у якому накопичена частота (S_i) перевищує половину суми всіх частот ($\sum f\ /2$):

$$Me = x_0 + h\ \frac{0.5\sum f - S_{Me-1}}{f_{Me}},$$

де x_0 – нижня межа медіанного інтервалу;

h – величина (крок) інтервалу;

$\sum f$ – сума всіх частот (обсяг вибірки);

S_{Me-1} – накопичена частота інтервалу, що передує медіанному;

f_{Me} – частота самого медіанного інтервалу.

Формула моди (Mo)

Спочатку визначається модальний інтервал – інтервал, що має найбільшу частоту:

$$Mo = x_0 + h \frac{f_{Mo} - f_{Mo-1}}{(f_{Mo} - f_{Mo-1}) + (f_{Mo} - f_{Mo+1})},$$

де x_0 – нижня межа модального інтервалу;
h – величина (крок) інтервалу;
f_{Mo} – частота модального інтервалу;
f_{Mo-1} – частота інтервалу, що передує модальному;
f_{Mo+1} – частота інтервалу, наступного за модальним.

Реалізація в Python (для інтервальних даних)

Оскільки стандартні бібліотеки працюють переважно із «сирими» даними, для інтервальних рядів необхідно створити власні функції на основі наведених вище формул:

```
def grouped_mode(lower_bounds, frequencies, h):
    """Обчислення моди для інтервального ряду."""
    # Знаходимо індекс інтервалу з найбільшою частотою
    idx = frequencies.index(max(frequencies))
        f_mo = frequencies[idx]
    f_prev = frequencies[idx - 1] if idx > 0 else 0
    f_next = frequencies[idx + 1] if idx < len(frequencies) - 1 else 0
        x0 = lower_bounds[idx]
        # Формула моди
    mo = x0 + h * ((f_mo - f_prev) / ((f_mo - f_prev) + (f_mo - f_next)))
    return mo
def grouped_median(lower_bounds, frequencies, h):
    """Обчислення медіани для інтервального ряду."""
    total_count = sum(frequencies)
    half_count = total_count / 2
    cumulative_freq = 0
    idx = 0
        # Шукаємо медіанний інтервал
    for i, f in enumerate(frequencies):
        cumulative_freq += f
        if cumulative_freq >= half_count:
            idx = i
            break
        # Визначаємо накопичену частоту попереднього інтервалу
    s_prev = cumulative_freq - frequencies[idx]
    f_me = frequencies[idx]
    x0 = lower_bounds[idx]
        # Формула медіани
    me = x0 + h * ((half_count - s_prev) / f_me)
    return me
# Приклад даних: інтервали 10-20, 20-30, 30-40
L = [10, 20, 30]  # Нижні межі (x0)
F = [5, 15, 10]   # Частоти (fi)
step = 10         # Ширина (h)
print(f"Мода (інтервальна): {grouped_mode(L, F, step):.2f}")
print(f"Медіана (інтервальна): {grouped_median(L, F, step):.2f}")
```

Результат роботи програми:

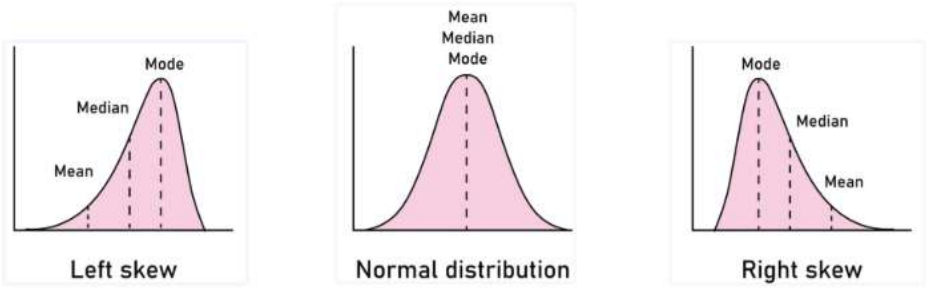
```
Мода (інтервальна): 26.67
Медіана (інтервальна): 26.67
```

Реалізація в MS Excel

В Excel немає вбудованих функцій для цих формул. Розрахунок виконується поетапно в комірках або створенням таблиці, де окремо вираховуються накопичені частоти для пошуку медіанного інтервалу.

Співвідношення мір центральної тенденції

Порівняння середнього ($\bar{x}$), медіани (Me) та моди (Mo) дає змогу визначити форму розподілу та наявність асиметрії без побудови графіка:



Симетричний розподіл: $\bar{x} \approx Me \approx Mo$.

Дані розподілені рівномірно навколо центру («дзвоноподібна» крива).

Правостороння (позитивна) асиметрія: $\bar{x} > Me > Mo$.

«Хвіст» розподілу витягнутий управо, у бік великих значень. Середнє зміщується в бік викидів сильніше, ніж медіана.

Лівостороння (негативна) асиметрія: $\bar{x} < Me < Mo$.

«Хвіст» розподілу витягнутий уліво. Середнє занижується через малі екстремальні значення.

Характеристики розсіювання (міри варіації)

Ці показники кількісно описують ступінь відхилення варіант від центральної тенденції та однорідність сукупності.

Розмах варіації (R)

Розмах варіації (Range) – це найпростіша абсолютна міра розсіювання, що визначається як різниця між максимальним ($\boldsymbol{x_{max}}$) та мінімальним ($\boldsymbol{x_{min}}$) значеннями елементів вибірки:

$$R = x_{max} - x_{min}.$$

Цей показник характеризує лише граничні межі варіювання ознаки (амплітуду). Розмах є дуже чутливим до аномальних викидів: одне екстремальне значення може суттєво викривити уявлення про варіацію всієї сукупності.

Реалізація в Python

У бібліотеці $numpy$ для цього існує спеціалізована функція $ptp()$ (peak-to-peak), яка працює швидше за окремий виклик максимуму та мінімуму на великих масивах.

Приклад коду:

```
import numpy as np
data = [10, 12, 15, 18, 20]
# Спосіб 1: Використання функції NumPy
range_val = np.ptp(data)
# Спосіб 2: Класичний підхід (через вбудовані функції Python)
# range_val = max(data) - min(data)
print(f"Розмах варіації: {range_val}")
```

Результат роботи програми:

```
Розмах варіації: 10
```

Реалізація в MS Excel

В Excel немає єдиної готової функції для обчислення розмаху. Розрахунок виконується шляхом комбінації функцій пошуку екстремумів за допомогою такого виразу в рядку формул:

$$= MAX(\text{діапазон\_даних}) - MIN(\text{діапазон\_даних}).$$

Вибіркова дисперсія (S^2) та стандартне відхилення (S)

Ці показники є основними мірами варіації, що описують, наскільки далеко значення вибірки «розкидані» відносно їх середнього арифметичного.

Вибіркова дисперсія (S^2) – це середнє арифметичне квадратів відхилень кожного значення від середнього. Вона показує ступінь мінливості даних, але вимірюється у «квадратних одиницях» (наприклад, грн^2, кг^2), що ускладнює інтерпретацію.

Стандартне відхилення (S) (середньоквадратичне відхилення) – це корінь квадратний з дисперсії. Воно вимірюється в тих самих одиницях, що й вихідні дані, і показує середню величину помилки, яку допускаємо, замінюючи будь-який елемент вибірки її середнім.

Для отримання незсуненої оцінки (unbiased estimator) генеральної дисперсії на основі вибірки, суму квадратів ділять не на n, а на кількість ступенів вільності $n - 1$ (поправка Бесселя).

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2;$$

$$S = \sqrt{S^2}.$$

Реалізація в Python

Бібліотека $numpy$ за замовчуванням обчислює дисперсію для *генеральної* сукупності (ділить на $\boldsymbol{n}$). Щоб отримати *вибіркову* дисперсію (як в Excel функція $VAR.S$), необхідно явно вказати параметр $ddof = 1$ ($Delta\ Degrees\ of\ Freedom$).

Приклад коду:

```
import numpy as np
data = [10, 12, 15, 18, 20]
# Обчислення вибіркової дисперсії (Variance)
# ddof=1 вмикає ділення на n-1
variance = np.var(data, ddof=1)
# Обчислення стандартного відхилення (Standard Deviation)
std_dev = np.std(data, ddof=1)
print(f"Вибіркова дисперсія: {variance:.2f}")
print(f"Стандартне відхилення: {std_dev:.2f}")
```

Результат роботи програми:

```
Вибіркова дисперсія: 17.00
Стандартне відхилення: 4.12
```

Реалізація в MS Excel

В Excel існують окремі функції для *генеральної* сукупності (застарілі або з суфіксом .P) та для *вибірки* (з суфіксом .S). Для статистичного аналізу вибіркових даних слід використовувати такі вирази у рядку формул:

— дисперсію: «$= VAR.S(\text{діапазон\_даних})$». Функція «$= VAR(\dots)$» у старих версіях також рахує за *вибіркою*;

— стандартне відхилення: «$= STDEV.S(\text{діапазон\_даних})$». Функція «$= STDEV(\dots)$» у старих версіях також рахує за *вибіркою*.

Дисперсія та стандартне відхилення для згрупованих даних

Якщо дані подано у вигляді таблиці частот або інтервалів, розрахунок мір розсіювання базується на серединах інтервалів. Припускаємо, що всі варіанти всередині інтервалу рівномірно розподілені, тому їх можна замінити одним центральним значенням.

Алгоритм розрахунку:

1. Знайти середину кожного інтервалу: $x_i' = \frac{x_{min}+x_{max}}{2}$.
2. Обчислити зважене середнє арифметичне: $\overline{x_{wa}} = \frac{\sum f_i x_i'}{\sum f_i}$.
3. Розрахувати вибіркову дисперсію, зважуючи квадрат відхилення середини інтервалу на його частоту.

Формула вибіркової дисперсії для згрупованих даних

$$S^2 = \frac{\sum f_i (x_i' - \bar{x})^2}{(\sum f_i) - 1}.$$

Формула стандартного відхилення

$$S = \sqrt{S^2}.$$

Реалізація в Python

Стандартні функції $numpy.var$ або $numpy.std$ не підтримують роботу з вагами частот напряму так, щоб коректно врахувати поправку на вибірку $(n-1)$. Тому обчислення виконується поетапно або за допомогою власної функції.

Приклад коду:

```
import numpy as np
def grouped_statistics(lower_bounds, upper_bounds, frequencies):
    """
    Розрахунок середнього, дисперсії та станд. відхилення для інтервалів.
    """
    # 1. Знаходимо середини інтервалів
    midpoints = (np.array(lower_bounds) + np.array(upper_bounds)) / 2
    freqs = np.array(frequencies)
    n = np.sum(freqs) # Загальний обсяг вибірки
        # 2. Зважене середнє
    weighted_mean = np.average(midpoints, weights=freqs)
        # 3. Дисперсія (сума зважених квадратів відхилень / n-1)
    variance = np.sum(freqs * (midpoints - weighted_mean)**2) / (n - 1)
        # 4. Стандартне відхилення
    std_dev = np.sqrt(variance)
        return weighted_mean, variance, std_dev
# Вхідні дані: Інтервали 0-10, 10-20, 20-30
L = [0, 10, 20]   # Нижні межі
U = [10, 20, 30]  # Верхні межі
F = [5, 12, 3]    # Частоти
mean, var, std = grouped_statistics(L, U, F)
print(f"Середнє: {mean:.2f}")
print(f"Дисперсія: {var:.2f}")
print(f"Стандартне відхилення: {std:.2f}")
```

Результат роботи програми:

```
Середнє: 14.00
Дисперсія: 41.05
Стандартне відхилення: 6.41
```

Реалізація в MS Excel

В Excel немає вбудованої функції для зваженої дисперсії. Розрахунок виконується через створення допоміжних стовпчиків у таблиці.

Припустимо, у вас є таблиця:

	A	B	C
1	Нижня межа	Верхня межа	Частота f
2			

1. Створити стовпчик «Середина» (D): $= (A2 + B2)/2$.
2. Розрахувати *середнє зважене*.

Нехай результат буде в комірці G1. У рядок формул небхідно ввести такий вираз: «$= SUMPRODUCT(D{:}D;\ C{:}C)\ /\ SUM(C{:}C)$ ()».

3. Розрахувати *дисперсію*.

Можна створити допоміжний стовпчик E з виразом у рядку формул для кожного рядка: « $= (D2 - \$G\$1)\text{^}2\ *\ C2$».

Тоді дисперсія: $= SUM(E{:}E)\ /\ (SUM(C{:}C)\ -\ 1)$.

Альтернатива (однією складною формулою без стовпчика E):

$= SUMPRODUCT\big(C2{:}C10; (D2{:}D10 - G1)\ \text{^}2\ \ \big)/(SUM(C2{:}C10) - 1)$,

де D – діапазон середин; C – діапазон частот; G1 – середнє.

Емпірична функція розподілу

Емпірична функція розподілу – це статистичний аналог теоретичної функції розподілу $F(x)$, побудований на основі вибіркових даних. Вона визначає для кожного значення x відносну частоту появи подій, менших або таких, що дорівнюють цьому значенню.

Позначається як $F^*(x)$ або $\widehat{F}_n(x)$. Формула розрахунку

$$F^*(x) = \frac{n_x}{n},$$

де n_x – кількість елементів вибірки, значення яких менші або дорівнюють x (накопичена частота);

n – загальний обсяг вибірки.

Емпірична функція розподілу є нормованою та неспадною: її значення завжди знаходяться в межах $0 \leq F^*(x) \leq 1$, а зі збільшенням аргументу x накопичена частота лише збільшується або залишається незмінною. Графічно така залежність відображається у вигляді розривної східчастої лінії, де «стрибки» відбуваються безпосередньо у точках, що відповідають спостережуваним значенням варіант. При цьому висота кожної сходинки дорівнює $1/n$ (або кратна цій величині, якщо у вибірці присутні значення, що повторюються).

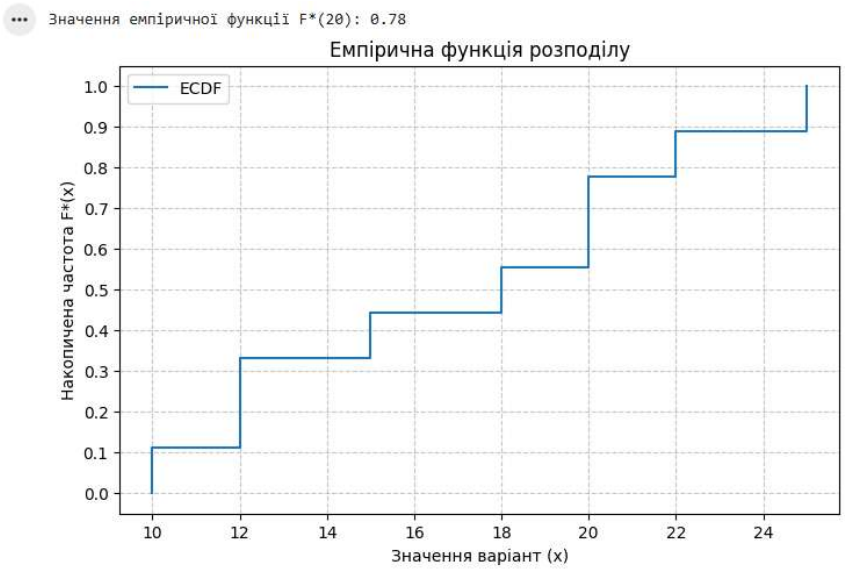
Реалізація в Python

Для роботи з емпіричним розподілом найзручніше використовувати бібліотеку $statsmodels$. Клас ECDF створює повноцінний об'єкт-функцію, який можна використовувати як для отримання ймовірностей, так і для побудови графіка.

Приклад коду:

```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.distributions.empirical_distribution import ECDF
# Вихідні дані
data = [10, 12, 12, 15, 18, 20, 20, 22, 25]
# 1. Створення об'єкта-функції ECDF
ecdf = ECDF(data)
# 2. Аналітичне використання:
# Отримання значення функції в конкретній точці.
# Запитання: "Яка частка значень не перевищує 20?" (P(X <= 20))
prob = ecdf(20)
print(f"Значення емпіричної функції F*(20): {prob:.2f}")
# Результат: 0.78 (оскільки 7 з 9 чисел <= 20)
# 3. Графічна побудова
# Використовуємо функцію step() для створення східчастого графіка
plt.figure(figsize=(8, 5))
plt.step(ecdf.x, ecdf.y, where='post', label='ECDF')
# Оформлення графіка
plt.title("Емпірична функція розподілу")
plt.xlabel("Значення варіант (x)")
plt.ylabel("Накопичена частота F*(x)")
plt.yticks(np.arange(0, 1.1, 0.1))  # Крок сітки по Y
plt.grid(True, linestyle='--', alpha=0.7)
plt.legend()
plt.show()
```

Результат роботи програми:

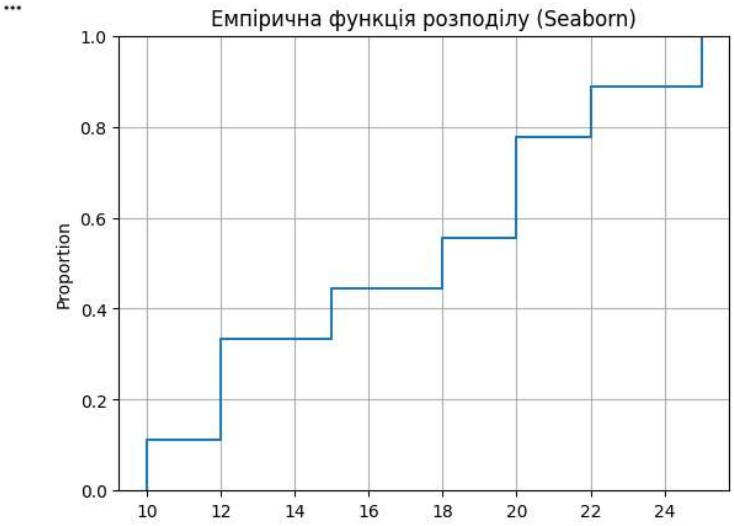


Для миттєвої побудови графіка без створення проміжних об'єктів використовується функція $ecdfplot$ бібліотеки *Seaborn*. Вона автоматично сортує дані та обчислює накопичені частки.

Приклад реалізації:

```
import seaborn as sns
import matplotlib.pyplot as plt
data = [10, 12, 12, 15, 18, 20, 20, 22, 25]
# Побудова графіка однією командою
sns.ecdfplot(data=data)
# Налаштування відображення (опціонально)
plt.title("Емпірична функція розподілу (Seaborn)")
plt.grid(True)
plt.show()
```

Результат роботи програми:



Цей метод вирізняється лаконічністю та автоматизацією, оскільки бібліотека $seaborn$ самостійно налаштовує стиль графіка та осі, не потребуючи ручних розрахунків чи імпорту додаткових модулів

(наприклад, $statsmodels$). За замовчуванням функція візуалізує накопичену частку (у діапазоні від 0 до 1), проте за необхідності відобразити абсолютну кількість спостережень достатньо додати до виклику параметр $stat = 'count'$.

Реалізація в MS Excel

В Excel відсутній інструмент для автоматичної побудови емпіричної функції розподілу однією кнопкою, тому це відбувається через побудову таблиці накопичених частот.

1. Необхідно відсортувати дані в одному стовпці за зростанням, це є критично важливим для коректного накопичення частот.
2. Виконати ранжування: у сусідньому стовпці пронумерувати елементи від 1 до n (n – обсяг вибірки).
3. У третьому стовпці виконати розрахунок частот за допомогою такого виразу у рядку формул «$= \text{Ранг} / \text{Загальна кількість}$».
4. Побудувати графік, виділивши стовпчик зі значеннями (вісь X) і стовпчик з розрахованими частками (вісь Y), та вставити діаграму типу $Scatter\ with\ Straight\ Lines$.

Для отримання значення емпіричної функції розподілу без побудови допоміжної таблиці (альтернативний варіант) використовується функція в рядку формул

$$= PERCENTRANK.INC(\text{масив};\ x).$$

Ця функція повертає ранг значення у відсотках (від 0 до 1), що математично відповідає значенню $F^*(x)$. Суфікс $.INC$ (inclusive) означає, що функція додає межі масиву ($0\ \%$ та $100\ \%$) у розрахунок, що відповідає визначенню кумуляти.

Порядок виконання роботи

Теоретична підготовка

1. Опрацювати фундаментальні поняття математичної статистики: генеральна сукупність та вибірка, варіаційний ряд, частоти та кумуляти (накопичені частоти).
2. Вивчити визначення та аналітичні формули для розрахунку мір центральної тенденції (середнє арифметичне, мода, медіана) та мір розсіювання (дисперсія, середнє квадратичне відхилення, розмах, коефіцієнт варіації).
3. Розглянути методику побудови інтервального варіаційного ряду та алгоритм побудови емпіричної функції розподілу $F^*(x)$ (кумулятивної кривої).
4. Ознайомитися з програмною реалізацією статистичних методів у мові Python, зокрема з функціональними можливостями бібліотек pandas (обробка рядів), numpy (обчислення) та matplotlib.pyplot (візуалізація гістограм та $F^*(x)$).

5. Підготувати необхідні інструменти: MS Excel та середовище Python.

6. Вибрати свій індивідуальний варіант завдання.

Аналітичне розв'язання (MS Excel / Вручну)

На цьому етапі проводиться апріорне дослідження завдання для отримання результатів, що будуть еталоном для верифікації коду.

1. Для структурування даних:

— чітко формалізувати варіаційні ряди: виконати ранжування вибірок (для пунктів з незгрупованими даними) та зафіксувати інтервальні межі (для згрупованих даних);

— побудувати розширені таблиці частот, визначивши абсолютні ($\boldsymbol{n_i}$), відносні (w_i) та накопичені частоти, необхідні для подальшої візуалізації.

2. Для статистичного аналізу:

— обчислити точні значення мір центральної тенденції (середнє $\bar{x}$, мода Mo, медіана Me), застосовуючи відповідні формули для дискретних та інтервальних рядів;

— розрахувати точні значення мір розсіювання: дисперсії (D), середнього квадратичного відхилення (σ), розмаху (R) та коефіцієнта варіації (V);

— сформулювати попередню гіпотезу про характер розподілу (симетрія/асиметрія) для кожного пункту варіанта на основі отриманих числових показників.

Програмна реалізація та використання ШІ (Python)

Створити новий Python-скрипт (.py) або інтерактивний блокнот (.ipynb) та виконати імпорт необхідних бібліотек (pandas, numpy, matplotlib).

1. Для оброблення даних та обчислень:

— ініціалізувати вхідні дані: створити масиви/списки для пунктів з незгрупованими вибірками та відповідні структури для пункту з інтервальними даними;

— програмно реалізувати розрахунок повного спектра статистичних характеристик для кожного пункту варіанта: середнього, моди, медіани, дисперсії, стандартного відхилення та розмаху;

— забезпечити форматоване виведення результатів у консоль для порівняння з розрахунками, виконаними на попередньому етапі.

2. Для візуалізації та інтеграції ШІ:

— розробити код для побудови графіків згідно з вимогами кожного пункту: емпіричної функції розподілу $F^*(x)$ та полігону частот (для незгрупованих даних); гістограми та полігону частот (для інтервальних даних);

— залучити ШІ-асистента для оптимізації коду візуалізації (наприклад, для коректного відображення сходинкового вигляду $F^*(x)$ у matplotlib) або для допомоги в реалізації алгоритму обчислення медіани;

— використати ШІ для інтерпретації результатів: пояснення відмінностей між модою та середнім значенням або аналізу форми розподілу отриманих вибірок.

Аналіз результатів та оформлення звіту

1. Порівняти числові характеристики, отримані аналітично (в MS Excel) та за допомогою Python.

На цьому етапі необхідно провести детальне зіставлення обчислених значень (середнього, дисперсії, моди, медіани). У разі виявлення розбіжностей здобувач освіти має ідентифікувати їх причини: чи це була помилка в логіці Python-коду, чи похибка округлення в ручних розрахунках, чи особливості реалізації алгоритмів у бібліотеках.

2. Проаналізувати форму розподілу для кожного пункту варіанта.

На основі взаємного розташування мір центральної тенденції (середнього $\bar{x}$, моди Mo та медіани Me) визначити тип симетрії або асиметрії розподілу. Слід зробити висновок: чи є розподіл симетричним, правостороннім (позитивна асиметрія) або лівостороннім (негативна асиметрія).

3. Оцінити інформативність графічних моделей (полігону частот, гістограми, емпіричної функції $F^*(x)$.

Необхідно співставити візуальний вигляд побудованих графіків із числовими висновками пункту 2. Графіки мають наочно підтверджувати характер розподілу даних та демонструвати щільність частот у відповідних інтервалах.

4. Сформулювати загальні висновки до роботи, підсумувавши виконані дії та отримані результати.

Висновок має бути комплексним: від підтвердження навичок розрахунку статистичних показників (уручну та програмно) до оцінки ефективності використання Python-бібліотек (pandas, matplotlib) для автоматизації оброблення експериментальних даних.

Вимоги до звіту та його вміст

Звіт є підсумковим документом, що фіксує результати проведеного дослідження, його структура має містити такі обов'язкові компоненти:

1. Титульна сторінка.

Оформлюється згідно з установленим зразком. Має містити назву дисципліни, номер роботи, дані здобувача освіти та викладача.

2. Тема, мета роботи та номер варіанта.

Формулюються чітко відповідно до методичних вказівок.

3. Постановка завдання.

Необхідно повністю скопіювати умову завдання зі свого індивідуального варіанта (включно з вихідною вибіркою даних).

4. Результати виконання.

Це основний розділ звіту, що складається з трьох частин:

а) розрахунки (MS Excel / Вручну). Навести таблиці з проміжними та кінцевими розрахунками:

— побудова варіаційних рядів (дискретного або інтервального);

— розрахунок мір центральної тенденції (середнє, медіана, мода);

— розрахунок мір розсіювання (дисперсія, стандартне відхилення);

б) код програми: повний програмний код на Python з коментарями, що пояснюють ключові етапи оброблення даних та вибір бібліотек;

в) результати роботи програми:

— скріншот текстового виведення (консолі) з розрахованими статистичними показниками;

— згенеровані графіки з підписами осей та назвами (полігон частот, гістограма, емпірична функція розподілу).

5. Аналіз використання ШІ-асистента.

У цьому розділі здобувач освіти має продемонструвати навички критичної взаємодії з технологіями штучного інтелекту:

а) вибір програмного інструментарію ШІ. Вказати повну назву та версію використаної моделі (наприклад, Gemini 1.5 Pro, ChatGPT 4o, Claude 3.5 Sonnet);

б) роль ШІ у виконанні роботи. Навести 2–3 конкретні приклади запитів (промптів) та проаналізувати отримані відповіді. Наприклад, як ШІ допоміг з вибором типу графіка для дискретних/неперервних даних, пояснив фізичний зміст дисперсії або допоміг виправити помилки (debug) у коді.

6. Висновки.

Розділ має містити підсумок проведеного дослідження, розділений на два аспекти:

а) технічний висновок: необхідно провести компаративний аналіз результатів:

— підтвердити збіжність розрахунків, виконаних уручну (в Excel) та програмно;

— описати форму розподілу (симетричний, асиметричний) та оцінити ступінь розсіювання даних (однорідність сукупності);

— зробити аргументований висновок про те, наскільки отримані вибіркові оцінки (середнє, дисперсія) є надійними для характеристики генеральної сукупності;

б) рефлексія щодо використання ШІ.

Критично оцінити ефективність допомоги асистента. Описати, чи сприяв ШІ глибшому розумінню статистичних методів, чи його роль обмежилася технічним пришвидшенням написання коду та розрахунків.

Завдання для самостійного виконання

Завдання 1. Аналіз незгрупованих даних.

Дані (час виконання алгоритму, мс): 12, 15, 18, 20, 22, 22, 25, 30, 32, 35, 40, 40, 45, 50, 55, 60, 65, 70, 75, 80.

Проаналізувати: обчислити середнє, дисперсію, моду, медіану, розмах; побудувати емпіричну функцію розподілу та полігон частот.

Завдання 2. Аналіз згрупованих даних.

У межах аналізу часу відгуку бази даних було проведено вимірювання. Час відгуку (у мілісекундах) був згрупований у інтервали. Отримані дані наведено нижче:

$[x_i; x_{i+1})$	[5; 10)	[10; 15)	[15; 20)	[20; 25)	[25; 30)
n_i	6	8	5	4	2

Проаналізувати: обчислити середнє, дисперсію, стандартне відхилення, моду, медіану, розмах; побудувати гістограму та полігон частот.

Завдання 3. Комплексний аналіз вибірки.

Дані (100 вимірювань часу оброблення, мс): 2, 5, 8, 6, 4, 5, 8, 6, 8, 4, 8, 5, 8, 10, 2, 5, 8, 6, 8, 6, 8, 4, 8, 5, 8, 6, 8, 4, 8, 5, 8, 10, 5, 8, 6, 8, 6, 5, 8, 6, 8, 4, 8, 5, 8, 10, 2, 5, 8, 10, 4, 8, 5, 8, 10, 5, 8, 6, 8, 6, 5, 8, 6, 8, 4, 8, 5, 8, 10, 2, 5, 8, 10, 2, 5, 8, 6, 4, 5, 8, 6, 8, 4, 8, 5, 8, 10, 2, 5, 8, 6, 8, 6, 8, 4, 8, 5, 8, 6, 8.

Проаналізувати: виконати повний статистичний аналіз: побудувати варіаційний та статистичні ряди (частот, відносних, накопичених), емпіричну функцію, гістограму, полігон та обчислити всі числові характеристики.

Контрольні запитання

1. Що таке варіаційний ряд і для чого він використовується?
2. У чому різниця між модою та медіаною? Коли доцільно використовувати кожну з них?
3. Що характеризує дисперсія, а що – середньоквадратичне відхилення?
4. Для яких даних будують гістограму, а для яких – полігон частот?
5. Що показує емпірична функція розподілу? Які її властивості?
6. Яка функція в бібліотеці numpy дає змогу обчислити середнє значення та медіану?
7. Як у matplotlib налаштувати кількість стовпців (бінів) для гістограми?
8. Наведіть приклад, коли вибіркове середнє може бути поганою характеристикою центральної тенденції.
9. Що таке розмах вибірки і яку інформацію він дає про дані?
10. Як ШІ може допомогти у виборі оптимальної кількості інтервалів для побудови гістограми?

Практична робота № 5
ТОЧКОВІ ТА ІНТЕРВАЛЬНІ ОЦІНКИ ПАРАМЕТРІВ РОЗПОДІЛУ

Мета роботи:

— навчитися обчислювати точкові оцінки параметрів розподілу (зокрема методом моментів);

— дослідити різницю між зміщеними та незміщеними оцінками дисперсії;

— опанувати побудову довірчих інтервалів для середнього та дисперсії за допомогою Python та MS Excel;

— навчитися визначати мінімальний обсяг вибірки для заданої точності розрахунків.

Компетентності, що формуються:

— здатність обчислювати точкові оцінки параметрів розподілу (середнє, дисперсія) за допомогою методу моментів;

— уміння розрізняти зміщені та незміщені оцінки дисперсії та обґрунтовувати умови їх застосування;

— навички побудови довірчих інтервалів для математичного сподівання та дисперсії з використанням t-розподілу Стьюдента та χ^2-розподілу;

— здатність визначати мінімально необхідний обсяг вибірки для досягнення заданої точності оцінювання;

— навички програмної реалізації статистичних розрахунків та побудови інтервальних оцінок засобами бібліотек numpy та scipy.stats;

— ефективне використання ШІ-асистентів для генерації коду, його налагодження та пояснення складних статистичних концепцій.

Необхідне програмне забезпечення та бібліотеки:

1. Середовище розроблення: Google Colab, Jupyter Notebook або будь-яке інше інтегроване середовище розроблення (IDE) для Python (наприклад, VS Code, PyCharm).
2. Python версії 3.8 або новішої з бібліотеками: numpy, scipy, matplotlib, pandas.
3. MS Excel із встановленою надбудовою «Аналіз даних» (Data Analysis ToolPak).

Теоретичні відомості

Основи оцінювання параметрів розподілу

У математичній статистиці процес переходу від спостережень (вибірки) до висновків про всю групу (генеральну сукупність) базується на оцінюванні параметрів. Оцінювання параметрів – це знаходження

невідомих характеристик розподілу (таких як математичне сподівання a чи дисперсія σ^2) за даними вибірки.

За формою подання результату розрізняють два типи оцінок.

Точкові оцінки – це конкретні числові значення, що розраховуються за вибіркою. Вони є «найкращим наближенням» справжнього параметра, але мають недолік: не дають уявлення про можливу помилку.

Інтервальні оцінки (довірчі інтервали) – це діапазони значень, які з певною наперед заданою ймовірністю (надійністю γ) містять справжнє значення параметра. Вони дають змогу оцінити точність та надійність дослідження.

Метод моментів

Метод моментів є одним із класичних методів оцінювання параметрів розподілу. Він базується на прирівнюванні теоретичних моментів розподілу (математичних сподівань степенів випадкової величини) до їх вибіркових аналогів. Для випадкової величини X з розподілом, що залежить від параметрів $\theta 1, \theta 2, \ldots, \theta k$, метод моментів передбачає:

1. Обчислення перших k вибіркових моментів:

$$m_r = \frac{1}{n}\sum_{i=1}^{n} x_i^r, \quad r = 1, 2, \ldots, k,$$

де x_i – елементи вибірки; n – обсяг вибірки; r – порядок моменту.

2. Прирівнювання вибіркових моментів до теоретичних моментів $M_r = E(X^r)$, які є функціями невідомих параметрів.

3. Розв'язання системи рівнянь для знаходження оцінок параметрів $\widehat{\Theta}_1, \widehat{\Theta}_2, \ldots, \widehat{\Theta}_k$.

Особливості застосування для типових розподілів

Для нормального розподілу $N(a,\ \sigma^2)$:

— перший момент $E(X) = a$, тому вибіркове середнє $\bar{x}$ є оцінкою $\hat{a}$;

— другий центральний момент $E(X-a)^2 = \sigma^2$, тому оцінка дисперсії може бути отримана через вибіркову дисперсію.

Для *розподілу Пуассона* з параметром *λ*

$$E(X) = \lambda,$$

тому вибіркове середнє $\bar{x}$ є оцінкою *λ*.

Для *біноміального розподілу B(n, p)*

$$E(X) = np,\ D(X) = np(1-p).$$

Використовуючи вибіркове середнє $\bar{x}$ та вибіркову дисперсію, можна отримати оцінки $\hat{n}$ та $\hat{p}$.

Властивості оцінок: незміщеність і стійкість

Оцінка параметра вважається незміщеною, якщо її математичне сподівання дорівнює істинному значенню параметра:

$$E(\widehat{\theta}) = \theta.$$

Наприклад, вибіркове середнє $\bar{x}$ є незміщеною оцінкою математичного сподівання *a* нормального розподілу, оскільки $E(\bar{x}) = a$.

Оцінка є стійкою (або консистентною), якщо при збільшенні обсягу вибірки n→∞ вона збігається за ймовірністю до істинного значення параметра:

$$\hat{\theta} \xrightarrow{P} \theta.$$

Стійкість забезпечує надійність оцінок при великих вибірках, що є важливим для прогнозування та прийняття рішень.

Зміщена та незміщена оцінки дисперсії

Для нормального розподілу зміщена оцінка дисперсії обчислюється як

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

Вона є зміщеною, оскільки $E(s^2) = \frac{n-1}{n}\sigma^2$. Незміщена оцінка дисперсії має такий вигляд:

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2,$$

оскільки $E(\hat{\sigma})^2 = \sigma^2$. Незміщена оцінка є кращою для аналізу мінливості, тоді як зміщена оцінка може використовуватися як консервативна оцінка.

Довірчі інтервали для параметрів нормального розподілу

Довірчий інтервал для середнього значення *a* нормального розподілу з невідомою дисперсією σ^2 базується на t-розподілі Стьюдента. Для вибірки обсягом n з середнім $\bar{x}$ і вибірковою дисперсією $\hat{\sigma}^2$ довірчий інтервал з надійністю $\gamma = 1 - \alpha$ має вигляд

$$\bar{x} \pm t_{\alpha/2,n-1} \cdot \frac{\hat{\sigma}}{\sqrt{n}},$$

де $t_{\alpha/2,n-1}$ – квантиль t-розподілу з $n-1$ ступенями вільності.

Довірчий інтервал для дисперсії σ^2 базується на χ^2-розподілі Пірсона. Для вибіркової дисперсії $\hat{\sigma}^2$

$$\frac{(n-1)\hat{\sigma}^2}{\chi^2_{\alpha/2,n-1}}, \frac{(n-1)\hat{\sigma}^2}{\chi^2_{1-\alpha/2,n-1}},$$

де $\chi^2_{\alpha/2,n-1}$ і $\chi^2_{1-\alpha/2,n-1}$ – квантилі χ^2-розподілу Пірсона.

Визначення обсягу вибірки

Одним із ключових етапів планування статистичного дослідження є визначення того, скільки спостережень необхідно провести для отримання надійного результату.

Для оцінювання середнього значення a нормального розподілу з відомою дисперсією σ^2, щоб похибка оцінювання не перевищувала δ з надійністю $\gamma = 1 - \alpha$, мінімальний обсяг вибірки n визначається за формулою

$$n \geq (\frac{z_{\alpha/2} \cdot \sigma}{\delta})^2,$$

де $z_{\alpha/2}$ – квантиль стандартного нормального розподілу. Ця формула дає змогу оптимізувати витрати на збір даних, забезпечуючи необхідну точність оцінки.

Практичне застосування методів оцінювання

Статистичні оцінки параметрів розподілу є фундаментом для прийняття обґрунтованих рішень у різних сферах — від логістики до маркетингу. Ключові сценарії застосування методів, розглянутих у цій роботі:

1. Оптимізація сервісу (очікування в черзі).

Розрахунок точкових оцінок середнього часу та дисперсії дає змогу бізнесу моделювати системи масового обслуговування. Це необхідно для оптимізації кількості робочих кас, прогнозування «пікових» навантажень та мінімізації часу очікування клієнтів.

2. Контроль якості (кількість дефектів).

Оцінювання параметра λ розподілу Пуассона (який описує рідкісні події) допомагає встановити середню інтенсивність появи браку на виробничій лінії. Це дає змогу оцінити стабільність техпроцесу та вчасно впровадити заходи з контролю якості.

3. Маркетингова аналітика (успішність рекламної кампанії).

Використання параметрів біноміального розподілу (n, p) дає змогу оцінити ймовірність конверсії (успішної дії клієнта). Це критично для аналізу ефективності різних стратегій просування та прогнозування майбутнього прибутку.

4. Логістика та безпека (аналіз авіаційних вантажів).

Побудова довірчих інтервалів для середньої ваги вантажу та його дисперсії є питанням безпеки. Це дає змогу авіакомпаніям точно планувати завантаження літаків, враховуючи можливі відхилення від середнього значення, та гарантувати стабільність рейсу.

5. Операційне планування (час польоту).

За допомогою визначення мінімально необхідного обсягу вибірки можна з високою точністю оцінити середній час перебування літака в повітрі. Це сприяє створенню максимально щільного та ефективного розкладу рейсів, що мінімізує простої в аеропортах.

Бібліотеки та методи Python для виконання завдання

Реалізація статистичних алгоритмів у Python базується на використанні спеціалізованих бібліотек, які забезпечують високу точність обчислень та зручність маніпулювання даними.

Бібліотека NumPy

$NumPy$ (Numerical Python) – фундаментальний пакет для наукових обчислень. Він дає змогу працювати з великими масивами даних, містить оптимізовані функції для розрахунку основних статистичних метрик.

Методи бібліотеки та їх функціональне призначення:

— $numpy.array(data)$ використовується для перетворення вхідних послідовностей (списків чи кортежів) у масив $NumPy$, що дає змогу ефективно застосовувати до даних математичні операції;

— $numpy.mean(data)$ призначено для обчислення вибіркового середнього $\bar{x}$, яке береться як точкова оцінка параметра a для нормального розподілу або λ для розподілу Пуассона;

— $numpy.var(data, ddof = 0)$ призначається для розрахунку зміщеної вибіркової дисперсії s^2 (з дільником n), що дає змогу проводити консервативний аналіз стабільності процесів;

— $numpy.var(data, ddof = 1)$ дає змогу отримати незміщену вибіркову дисперсію $s^2_{unbiased}$ (з дільником $n-1$) для об'єктивного оцінювання мінливості генеральної сукупності;

— $numpy.std(data, ddof = 1)$ дає змогу розрахувати виправлене стандартне відхилення s, необхідне для визначення точності оцінки при побудові довірчих інтервалів;

— $numpy.sum(data)$ виконує обчислення суми елементів вибірки, що важливо для перевірки розрахунків при роботі з таблицями частот;

— $numpy.average(data, weights)$ забезпечує розрахунок зваженого середнього значення, де параметр $weights$ відповідає частотам значень у варіаційному ряді.

Бібліотека SciPy (модуль stats)

Бібліотека $SciPy$ містить спеціалізовані модулі для роботи з різними типами ймовірнісних розподілів та обчислення їх критичних точок (квантилів).

Методи модуля scipy.stats та їх роль у розрахунках:

— $scipy.stats.t.ppf(q, df)$ використовується для знаходження квантиля t-розподілу Стьюдента, що є базовим кроком при побудові довірчого інтервалу для середнього значення за невідомої дисперсії;

— $scipy.stats.chi2.ppf(q, df)$ дає можливість визначити квантилі χ^2-розподілу, необхідні для встановлення нижньої та верхньої меж довірчого інтервалу дисперсії;

— $scipy.stats.norm.ppf(q)$ призначено для отримання квантиля стандартного нормального розподілу, що застосовується для визначення мінімального обсягу вибірки при заданій точності.

Допоміжні модулі Python

Допоміжні модулі виконують необхідні службові дії при застосуванні основних бібліотек:

— $collections.Counter(data)$ – метод модуля $Collections$, що дає змогу автоматизувати підрахунок унікальних значень у вибірці та сформувати на їх основі таблицю частот;

— $math.ceil(x)$ – функція модуля $Math$, яка виконує округлення значення вгору до найближчого цілого, що є обов'язковою умовою при визначенні достатнього обсягу вибірки n.

Приклад програмної реалізації (NumPy)

У наведеному нижче фрагменті коду продемонстровано процес оброблення одновимірного масиву даних: від завантаження вибірки до отримання точкових оцінок параметрів розподілу.

Приклад коду:

```
import numpy as np
# Формування масиву вхідних даних (вибірки)
sample = np.array([7, 12, 9, 15, 4])
# --- Статистичні обчислення ---
# 1. Розрахунок вибіркового середнього (оцінка математичного сподівання)
mean = np.mean(sample)
# 2. Незміщена вибіркова дисперсія (оцінка генеральної дисперсії)
# Параметр ddof=1 (Delta Degrees of Freedom) забезпечує ділення суми квадратів на (n - 1)
variance_unbiased = np.var(sample, ddof=1)
# 3. Зміщена вибіркова дисперсія (оцінка розсіювання всередині вибірки)
# Параметр ddof=0 є значенням за замовчуванням (ділення на n)
variance_biased = np.var(sample, ddof=0)
# 4. Незміщене стандартне відхилення (виправлене середнє квадратичне відхилення)
std = np.std(sample, ddof=1)
# --- Форматоване виведення результатів ---
print("="*40)
print("  СТАТИСТИЧНІ ХАРАКТЕРИСТИКИ ВИБІРКИ")
print("="*40)
print(f"Вхідний масив даних: {sample}")
print(f"Обсяг вибірки (n):    {len(sample)}")
print("-"*40)
print(f"Вибіркове середнє (mean):          {mean:.4f}")
print(f"Незміщена дисперсія (ddof=1):      {variance_unbiased:.4f}")
print(f"Зміщена дисперсія (ddof=0):        {variance_biased:.4f}")
print(f"Стандартне відхилення (std):       {std:.4f}")
print("="*40)
```

Результат роботи програми:

```
========================================
  СТАТИСТИЧНІ ХАРАКТЕРИСТИКИ ВИБІРКИ
========================================
Вхідний масив даних: [ 7 12  9 15  4]
Обсяг вибірки (n):    5
----------------------------------------
Вибіркове середнє (mean):          9.4000
Незміщена дисперсія (ddof=1):      18.3000
Зміщена дисперсія (ddof=0):        14.6400
Стандартне відхилення (std):       4.2778
========================================
```

У цьому прикладі показано використання форматування рядків ($f-strings$) з обмеженням кількості знаків після коми (:.4f). У математичній статистиці це дає змогу уникнути надлишкової точності, яка не має фізичного змісту, та зробити результати зручними для порівняння з табличними даними.

Програмне знаходження критичних точок (SciPy)

Для знаходження меж довірчих інтервалів використовується метод $.ppf()$ (Percent Point Function), який є зворотним до інтегральної функції розподілу. Він дає змогу отримати значення випадкової величини (квантиль) для заданої ймовірності.

Приклад програми:

```
from scipy.stats import t, chi2, norm
# --- Вхідні параметри ---
n = 30           # Обсяг вибірки
alpha = 0.05     # Рівень значущості (відповідає надійності gamma = 0.95)

# --- Обчислення квантилів ---
# 1. Квантиль t-розподілу (Стьюдента)
#  Необхідний для побудови довірчого інтервалу середнього при невідомій дисперсії.
# Використовується n-1 ступенів свободи (df).
t_quantile = t.ppf(1 - alpha/2, df=n-1)
# 2. Квантилі chi2-розподілу (Хі-квадрат)
# Використовуються для побудови довірчого інтервалу дисперсії.
# Оскільки розподіл асиметричний, розраховуються окремо нижня та верхня межі.
chi2_upper = chi2.ppf(1 - alpha/2, df=n-1) # Верхня критична точка
chi2_lower = chi2.ppf(alpha/2, df=n-1)     # Нижня критична точка
# 3. Квантиль стандартного нормального розподілу (Z-оцінка)
# Застосовується для розрахунку довірчих інтервалів при великих вибірках
# або для визначення мінімально необхідного обсягу дослідження.
z_quantile = norm.ppf(1 - alpha/2)
# --- Форматоване виведення результатів ---
print("="*50)
print(f"РОЗРАХУНОК КВАНТИЛІВ (n={n}, alpha={alpha})")
print("="*50)
print(f"1. Квантиль Стьюдента (t):          {t_quantile:.4f}")
print(f"2. Нижня межа Хі-квадрат (chi2):    {chi2_lower:.4f}")
print(f"3. Верхня межа Хі-квадрат (chi2):   {chi2_upper:.4f}")
print(f"4. Квантиль нормальний (z-score):   {z_quantile:.4f}")
print("="*50)
```

Результат роботи програми:

```
==================================================
РОЗРАХУНОК КВАНТИЛІВ (n=30, alpha=0.05)
==================================================
1. Квантиль Стьюдента (t):          2.0452
2. Нижня межа Хі-квадрат (chi2):    16.0471
3. Верхня межа Хі-квадрат (chi2):   45.7223
4. Квантиль нормальний (z-score):   1.9600
==================================================
```

Для двостороннього довірчого інтервалу аргументом функції ppf використано рівень $1 - alpha/2$, щоб відсікти половину рівня значущості з кожного «хвоста» розподілу. Це забезпечує симетрію ймовірності відносно центру розподілу.

Допоміжні інструменти оброблення даних

Окрім спеціалізованих статистичних пакетів, Python містить вбудовані модулі, що дають змогу ефективно групувати дані та виконувати операції округлення згідно з вимогами статистичної точності.

Модуль Collections (підрахунок частот)

Метод $Counter$ незамінний при роботі з дискретними даними (наприклад, із кількістю дефектів або покупок), коли вихідну вибірку потрібно подати у вигляді таблиці частот:

```
from collections import Counter
# Вихідна вибірка (наприклад, кількість дефектів на різних дільниках)
defects = [3, 5, 2, 1, 4, 3, 2, 2, 5]
# Побудова таблиці частот
freq_table = Counter(defects)
# Виведення результату у форматі "Значення: Частота"
print("Таблиця частот (x_i: n_i):", freq_table)
# Результат: Counter({2: 3, 3: 2, 5: 2, 1: 1, 4: 1})
```

Результат роботи програми:

```
••• Таблиця частот (x_i: n_i): Counter({2: 3, 3: 2, 5: 2, 1: 1, 4: 1})
```

Модуль Math (коректне округлення)

Під час розрахунку мінімального обсягу вибірки n математичні правила округлення змінюються на статистичні: будь-яке дробове значення має округлятися до більшого цілого, щоб гарантувати дотримання заданої похибки ϵ.

Приклад програми:

```
import math

Розраховане теоретичне значення обсягу вибірки
n_theoretical = 10.3

# Округлення вгору до найближчого цілого
n_final = math.ceil(n_theoretical)

print(f"Теоретичний обсяг: {n_theoretical}")
print(f"Необхідний обсяг (округлений): {n_final}")

# Результат: 11
```

Результат роботи програми:

```
••• Теоретичний обсяг: 10.3
    Необхідний обсяг (округлений): 11
```

Оброблення даних за таблицею частот (NumPy)

Якщо вхідні дані вже згруповані (наприклад, у задачі про рекламну кампанію), для розрахунку параметрів використовуються зважені обчислення:

```
import numpy as np

# x - значення (кількість подій), n - частоти (кількість спостережень)
x = np.array([0, 1, 2, 3, 4])
n = np.array([3, 10, 16, 34, 17])

# Обчислення зваженого середнього
mean = np.average(x, weights=n)
# Обчислення зваженої дисперсії

variance = np.average((x - mean)**2, weights=n)
print(f"Середнє за таблицею частот: {mean:.4f}")
print(f"Дисперсія за таблицею частот: {variance:.4f}")
```

Результат роботи програми:

```
Середнє за таблицею частот: 2.6500
Дисперсія за таблицею частот: 1.1275
```

Під час роботи з великими вибірками, поданими у вигляді частотних таблиць, використання $np.average$ з параметром $weights$ є значно продуктивнішим за «розгортання» таблиці у повний масив.

Порядок виконання роботи

Теоретична підготовка

1. Опрацювати фундаментальні поняття теорії оцінювання: точкові та інтервальні оцінки, метод моментів, рівень значущості α та надійність γ.
2. Вивчити властивості статистичних оцінок: незміщеність, стійкість (консистентність) та ефективність. Особливу увагу звернути на математичне обґрунтування різниці між зміщеною та незміщеною дисперсіями.
3. Розглянути алгоритми побудови довірчих інтервалів для параметрів нормального розподілу (математичного сподівання при невідомому σ та дисперсії) з використанням розподілів Стьюдента (t) та Пірсона (χ^2).
4. Ознайомитися з програмною реалізацією статистичних функцій у бібліотеці $scipy.stats$ (модулі $t.ppf$, $chi2.ppf$, $norm.ppf$) та методами оброблення зважених даних у $numpy$.
5. Підготувати інструментарій: MS Excel (надбудова «Аналіз даних») та середовище Python (Google Colab / IDE).
6. Обрати індивідуальний варіант завдання.

Аналітичне розв'язання (MS Excel / Вручну)

На цьому етапі формується еталонна база результатів для подальшої верифікації програмного коду.

1. Точкове оцінювання за методом моментів:

— для вибірок з незгрупованими даними обчислити вибіркові моменти та отримати точкові оцінки параметрів;

— для згрупованих даних (таблиць частот) виконати розрахунок зваженого середнього та дисперсії;

— розрахувати два види дисперсії: зміщену (для оцінки розсіювання в поточній вибірці) та незміщену (як оцінку генеральної дисперсії).

2. Інтервальне оцінювання та планування:

— використовуючи таблиці критичних точок або функції Excel, знайти відповідні квантилі розподілів для заданої надійності γ;

— побудувати довірчі інтервали для досліджуваних параметрів;

— розрахувати мінімально необхідний обсяг вибірки n для забезпечення заданої граничної похибки ϵ .

Програмна реалізація та використання ШІ (Python)

1. Оброблення даних та обчислення:

— ініціалізувати вхідні масиви даних та структури для частотних таблиць;

— реалізувати алгоритм оцінювання параметрів, використовуючи можливості numpy.mean, numpy.var (з параметром $ddof$) та numpy.average (для зважених оцінок);

— програмно обчислити квантилі та межі довірчих інтервалів за допомогою методів $scipy.stats$.

2. Візуалізація та інтеграція ШІ:

— залучити ШІ-асистента для написання функцій, що автоматизують розрахунок довірчих інтервалів для різних типів розподілів;

— використати ШІ для налагодження коду при побудові гістограм частот із накладеними межами довірчих інтервалів;

— за допомогою ШІ-інструментів проаналізувати вплив обсягу вибірки на ширину довірчого інтервалу та точність точкових оцінок.

Аналіз результатів та оформлення звіту

1. Верифікація результатів: порівняти числові показники, отримані аналітично та програмно. У разі розбіжностей (наприклад, при обчисленні квантилів χ^2) проаналізувати причини: похибки округлення або відмінності в аргументації функцій (рівень значущості α проти надійності γ).

2. Аналіз властивостей оцінок: на основі проведених розрахунків підтвердити властивість незміщеності виправленої дисперсії та прокоментувати стійкість вибіркового середнього при збільшенні N.

3. Інтерпретація довірчих інтервалів: пояснити практичний зміст отриманих інтервалів (наприклад, у контексті максимального завантаження літака або часу очікування) та оцінити, чи є знайдена похибка допустимою для конкретної прикладної задачі.

4. Сформулювати загальні висновки, підсумувавши ефективність методу моментів та надійність побудованих інтервальних моделей.

Вимоги до звіту та його вміст

Звіт є підсумковим документом, що фіксує результати проведеного дослідження, його структура має містити такі обов'язкові компоненти:

1. Титульна сторінка.

Оформлюється згідно з установленим зразком. Має містити назву дисципліни, номер роботи, дані здобувача освіти та викладача.

2. Тема, мета роботи та номер варіанта.

Формулюються чітко відповідно до методичних вказівок.

3. Постановка завдання.

Необхідно повністю скопіювати умову завдання зі свого індивідуального варіанта (включно з вихідною вибіркою даних).

4. Результати виконання.

Це основний розділ звіту, що складається з трьох частин:

а) розрахунки (MS Excel / Вручну). Навести таблиці з проміжними та кінцевими розрахунками:

— обчислення точкових оцінок за методом моментів;

— знаходження критичних точок ($t_{\alpha/2}$, $\chi^2_{\alpha/2}$) за таблицями;

— розрахунок меж довірчих інтервалів для середнього та дисперсії;

б) код програми: повний програмний код на Python з коментарями.

Важливо показати використання $numpy.mean()$, $numpy.var(ddof = 1)$ та функцій $scipy.stats.t.ppf$ або $scipy.stats.chi2.ppf$;

в) результати роботи програми:

— скріншот консолі з отриманими інтервалами;

— графіки: гістограма вибірки з накладеною теоретичною щільністю розподілу та візуалізація довірчого інтервалу.

5. Аналіз використання ШІ-асистента.

У цьому розділі здобувач освіти має продемонструвати навички критичної взаємодії з технологіями штучного інтелекту:

а) вибір програмного інструментарію ШІ. Вказати повну назву та версію (наприклад, Gemini 3 Flash, ChatGPT 4o);

б) роль ШІ у виконанні роботи.

Навести приклади промптів: як ШІ пояснив різницю між z-оцінкою та t-розподілом, або як допоміг розрахувати ступені вільності $df = n - 1$.

6. Висновки.

Розділ має містити підсумок проведеного дослідження:

а) технічний висновок: компаративний аналіз (збіжність Excel та Python).

Оцінка точності: як ширина інтервалу залежить від обсягу вибірки n. Чи є отримані за методом моментів оцінки стійкими?

б) рефлексія щодо використання ШІ.

Описати, чи допоміг асистент зрозуміти зміст поняття «довірча ймовірность», чи лише пришвидшив написання коду.

Завдання для самостійного виконання

Завдання 1. Аналіз часу очікування в черзі.

Досліджується час очікування клієнтів у черзі супермаркету (в хвилинах), який, як вважається, підкоряється нормальному розподілу.

Дані вибірки (30 клієнтів): 7, 12, 9, 15, 4, 11, 8, 16, 13, 6, 10, 14, 5, 18, 9, 3, 12, 17, 11, 8, 13, 19, 10, 7, 15, 14, 6, 20, 9, 11.

1. За методом моментів знайти точкові оцінки: середнього часу очікування та незміщеної дисперсії.
2. Теоретично перевірити, чи є оцінка середнього незміщеною та стійкою, та пояснити практичне значення цих властивостей.
3. Обчислити зміщену оцінку дисперсії та пояснити її значення для аналізу стабільності черг.

Завдання 2. Оцінювання параметрів розподілів.

1. Кількість дефектів на виробництві (розподіл Пуассона).

Аналізується кількість дефектних деталей за годину.

Дані (100 годин спостережень): 3, 5, 2, 1, 4, 0, 6, 3, 2, 8, 4, 5, 1, 3, 7, 2, 0, 3, 4, 5, 9, 2, 3, 1, 4, 6, 3, 2, 5, 1, 3, 4, 0, 2, 3, 5, 4, 1, 3, 2, 6, 3, 4, 5, 2, 1, 3, 7, 4, 0, 3, 5, 2, 1, 4, 8, 3, 2, 5, 1, 3, 4, 6, 2, 3, 5, 4, 1, 3, 2, 5, 3, 4, 1, 2, 3, 6, 5, 4, 3, 2, 1, 3, 5, 4, 2, 3, 1, 5, 3, 4, 2, 6, 3, 5, 1, 4, 2, 3, 7.

Необхідно сформувати таблицю частот та за методом моментів знайти точкову оцінку параметра λ (середню кількість дефектів).

2. Успішність рекламної кампанії (біноміальний розподіл).

Аналізується кількість покупок, здійснених одним клієнтом.

Дані (опитування 100 клієнтів):

Кількість покупок (x)	0	1	2	3	4	5	6	7	8
Кількість клієнтів (n_i)	3	10	16	34	17	11	5	3	1

Необхідно за методом моментів знайти точкові оцінки параметрів n (максимальна кількість спроб) та p (ймовірність покупки).

Завдання 3. Аналіз авіаційних вантажів.

1. Максимальне завантаження літака (нормальний розподіл).

Аналізується максимальне завантаження літака (в тоннах).

Дані (50 вимірювань): 75, 82, 87, 90, 93, 78, 85, 88, 91, 94, 79, 83, 86, 89, 92, 77, 84, 88, 90, 95, 81, 86, 89, 93, 76, 80, 85, 87, 91, 94, 78, 83, 88, 90, 92, 79, 84, 86, 89, 95, 82, 87, 91, 93, 77, 85, 88, 90, 92, 80.

Необхідно (з надійністю $\gamma = 0{,}95$):

— знайти довірчий інтервал для середнього максимального завантаження (при невідомій дисперсії);

— знайти довірчий інтервал для дисперсії максимального завантаження.

2. Планування обсягу вибірки.

Авіакомпанія хоче оцінити середній час польоту з високою точністю. Відомо, що час польоту підкоряється нормальному розподілу зі стандартним відхиленням $\sigma = 2$ год.

Необхідно визначити мінімальний обсяг вибірки n, необхідний для оцінювання середнього часу польоту з похибкою не більше 0.5 години та надійністю γ = 0,9.

Контрольні запитання

1. У чому різниця між точковою та інтервальною оцінками?
2. Що таке незміщена оцінка параметра? Наведіть приклад.
3. Поясніть суть методу моментів для знаходження оцінок.
4. Чому вибіркова дисперсія, обчислена з дільником n, є зміщеною оцінкою?
5. Як інтерпретувати 99 % довірчий інтервал для математичного сподівання?
6. Який розподіл використовується для побудови довірчого інтервалу для середнього при невідомій дисперсії?
7. Яка функція в scipy.stats дає змогу обчислити довірчий інтервал для середнього?
8. Для чого потрібно визначати мінімальний обсяг вибірки?
9. Наведіть приклад практичної задачі, де важливо оцінити параметр λ розподілу Пуассона.
10. Як ШІ може допомогти уникнути помилок при виборі формули для довірчого інтервалу (наприклад, коли дисперсія відома, а коли – ні)?
11. Як зміна надійності γ (наприклад, з 0.95 на 0.99) впливає на ширину довірчого інтервалу при незмінному обсязі вибірки?
12. Чому довірчий інтервал для дисперсії, побудований за допомогою χ^2-розподілу Пірсона, є асиметричним відносно точкової оцінки s^2?

Практична робота № 6
ПЕРЕВІРКА СТАТИСТИЧНИХ ГІПОТЕЗ ДЛЯ НОРМАЛЬНОГО РОЗПОДІЛУ

Мета роботи:

— освоїти алгоритм перевірки гіпотез від формулювання H_0 та H_1 до прийняття рішення за критичними значеннями та p-значенням;

— навчитися застосовувати t-тест (для середніх) та χ^2-тест (для дисперсії) у задачах з однією та двома вибірками;

— опанувати інструменти $scipy.stats$ для автоматизації розрахунків у Python та графічну візуалізацію критичних областей;

— закріпити навички верифікації розрахунків через порівняння результатів Excel та Python із залученням ШІ для інтерпретації висновків.

Компетентності, що формуються:

— здатність формулювати гіпотези H_0 та H_1 щодо середнього значення та дисперсії нормального розподілу;

— уміння вибирати статистичний критерій (z-тест, t-тест, χ^2-тест) залежно від обсягу вибірки та відомих параметрів;

— навички розрахунку статистики тесту, меж критичних областей та обчислення p-значення;

— здатність обґрунтовано приймати рішення щодо гіпотез та інтерпретувати результати у прикладних задачах;

— навички програмної реалізації тестів у $scipy.stats$ та візуалізації розподілів за допомогою $matplotlib$;

— ефективне використання ШІ для генерації функцій автоматизації, перевірки коду та пояснення концепцій.

Необхідне програмне забезпечення та бібліотеки:

1. Середовище розроблення: Google Colab, Jupyter Notebook або будь-яке інше інтегроване середовище розроблення (IDE) для Python.
2. Python версії 3.8 або новішої з встановленими бібліотеками: numpy, scipy, matplotlib, pandas.
3. Пакет MS Excel із активованою надбудовою «Аналіз даних» (Data Analysis ToolPak) для проведення порівняльних розрахунків.

Теоретичні відомості

Основні визначення

Випадкова величина, що підкоряється нормальному розподілу $(\mathrm{N}(\mu, \sigma^2))$, характеризується математичним сподіванням (μ) та дисперсією (σ^2).

Вибірка – це набір спостережень такої величини, отриманих у процесі дослідження. Вибіркове середнє $(\bar{x})$ обчислюється як середнє арифметичне значень вибірки за формулою

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

де x_i – значення спостережень;

n – обсяг вибірки.

Вибіркова дисперсія (s^2) визначається як

$$\mathrm{s}^2 = \frac{1}{\mathrm{n}-1}\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{x_i} - \bar{\mathrm{x}})^2$$

і є оцінкою дисперсії генеральної сукупності.

Рівень значущості (α) – це ймовірність помилки першого роду, тобто відхилення істинної нульової гіпотези. У роботі застосовуються значення $(\alpha = 0.01, 0.02, 0.025, 0.05, 0.10)$ залежно від вимог точності в предметній області.

Критична область – це множина значень статистики тесту, за яких нульова гіпотеза відхиляється. *p-значення* – ймовірність отримати статистику, не менш екстремальну, ніж спостережувана, за умови істинності нульової гіпотези; якщо p-значення менше (α), нульову гіпотезу відхиляють.

Перевірка середнього однієї вибірки (t-тест)

Для перевірки гіпотези про середнє значення вибірки, коли дисперсія генеральної сукупності невідома, використовується t-тест Ст'юдента. Згідно з «нульовою гіпотезою» $(\mathrm{H}_0\colon \mu = \mu_0)$ середнє вибірки $(\bar{\mathrm{x}})$ дорівнює заданому значенню (μ_0). Альтернативна гіпотеза може бути двобічною $(\mathrm{H}_1\colon \mu \neq \mu_0)$ або однобічною $(\mathrm{H}_1\colon \mu > \mu_0)$ або $(\mathrm{H}_1\colon \mu < \mu_0)$.

Статистика тесту обчислюється за формулою

$$\mathrm{t} = \frac{\bar{\mathrm{x}} - \mu_0}{\mathrm{s}/\sqrt{\mathrm{n}}},$$

де s – вибіркове стандартне відхилення; n – обсяг вибірки.

Ця статистика підкоряється t-розподілу Ст'юдента з $(n-1)$ ступенями свободи.

Для двобічного тесту критична область визначається як $\left(|\mathrm{t}| > \mathrm{t}_{\alpha/2,\mathrm{n}-1}\right)$, для правобічного – $\left(\mathrm{t} > \mathrm{t}_{\alpha,\mathrm{n}-1}\right)$, для лівобічного – $\left(\mathrm{t} < -\mathrm{t}_{\alpha,\mathrm{n}-1}\right)$.

Рішення приймається шляхом порівняння обчисленого значення $(|t|)$ з критичним значенням із таблиці t-розподілу або перевірки, чи p-значення менше (α).

Якщо умова виконується, нульову гіпотезу відхиляють. Метод застосовується, наприклад, для перевірки середнього часу оброблення замовлень на складі, тривалості роботи електролампочок або атмосферного тиску.

Перевірка дисперсії однієї вибірки (χ^2-тест)

Для перевірки гіпотези про дисперсію вибірки використовується (χ^2)-тест. Згідно з нульовою гіпотезою $(\mathrm{H}_0\colon \sigma^2 = \sigma_0^2)$ дисперсія вибірки (s^2) дорівнює заданому значенню (σ_0^2).

Альтернативна гіпотеза може бути двобічною $(H_1\colon \sigma^2 \neq \sigma_0^2)$ або однобічною: $(\mathrm{H}_1\colon \sigma^2 > \sigma_0^2)$ або $(\mathrm{H}_1\colon \sigma^2 < \sigma_0^2)$.

Статистика тесту обчислюється за формулою

$$\chi^2 = \frac{(\mathrm{n}-1)\mathrm{s}^2}{\sigma_0^2},$$

де n – обсяг вибірки.

Ця статистика підкоряється (χ^2)-розподілу з $k = (n-1)$ ступенями свободи. Критична область для двобічного тесту визначається як $\chi^2 < \chi^2_{\alpha/2,\mathrm{n}-1}$або $\chi^2 > \chi^2_{1-\alpha/2,\mathrm{n}-1}$; для правобічного – $\chi^2 > \chi^2_{1-\alpha,\mathrm{n}-1}$; для лівобічного – $\chi^2 < \chi^2_{\alpha,\mathrm{n}-1}$.

Рішення приймається шляхом порівняння (χ^2) з критичними значеннями із таблиці (χ^2)-розподілу або перевірки, чи p-значення менше (α). Якщо умова виконується, нульову гіпотезу відхиляють.

Метод застосовується для оцінювання стабільності напруги в електромережі, точності вимірювань датчиків або мінливості міцності матеріалів.

Порівняння двох незалежних вибірок

Для коректного порівняння середніх спочатку перевіряють рівність дисперсій двох вибірок за допомогою F-тесту Фішера ($H_0: \sigma_1^2 = \sigma_2^2$).

Статистика тесту

$$F = \frac{s_{max}^2}{s_{min}^2},$$

де s_{max}^2 – більша з двох вибіркових дисперсій.

Статистика підкоряється розподілу Фішера з $k_1 = n_{max} - 1$ та $k_2 = n_{min} - 1$ ступенями свободи.

Після цього використовується t-тест для порівняння середніх ($H_0: a_1 = a_2$). Статистика

$$t = \frac{\bar{x} - \bar{y}}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

де $s_p^2 = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}$ – об'єднана дисперсія.

Метод застосовується для порівняння доходів працівників, витрат електроенергії тощо.

Рівень значущості та прийняття рішення

Рівень значущості α визначає допустиму ймовірність *помилки першого роду* («хибна тривога»). Використовуються значення $0.01, 0.05, 0.1$, що відповідають вимогам у логістиці, медицині чи економіці.

Також існує *помилка другого роду* (β) – прийняття хибної H_0 («пропуск цілі»).

Рішення приймається шляхом порівняння статистики з критичними значеннями ($t_{\text{крит}}$, $\chi^2_{\text{крит}}$, $F_{\text{крит}}$) або перевірки, чи p-значення менше α. Якщо умова виконується, нульову гіпотезу відхиляють, що свідчить про значущі відмінності.

Використання Python для виконання завдань

Для автоматизації обчислень статистичних тестів використовується мова програмування Python із бібліотеками $scipy.stats$ (математичні розподіли) та $numpy$ (оброблення масивів).

Одновибірковий *t*-тест

У бібліотеці $scipy.stats$ використовується об'єкт t, який надає метод ppf (Percent Point Function) для визначення критичних значень та метод sf (Survival Function) для обчислення p-значень. Для двобічного тесту рівень значущості α ділиться навпіл.

Бібліотека $numpy$ використовується для знаходження вибіркових характеристик зі згрупованих даних. Для обчислення зваженого середнього використовується функція np.average, де параметр weights вказує на частоти спостережень.

Приклад реалізації:

```
import numpy as np
from scipy.stats import t
# 1. Оброблення даних за допомогою numpy
# x - значення, n - частоти спостережень
x_bar = np.average(x, weights=n)
n_total = n.sum()
df = n_total - 1
# 2. Розрахунок критичного значення (двобічний тест)
# Використовуємо ppf для рівня (1 - alpha/2)
t_crit = t.ppf(1 - alpha/2, df)
# 3. Розрахунок p-значення
# Використовуємо sf для знаходження площі "хвоста" і множимо на 2
p_value = 2 * t.sf(abs(t_stat), df)
```

Перевірка дисперсії (χ^2-тест)

У бібліотеці $scipy.stats$ використовується об'єкт $chi2$, який надає метод ppf для визначення критичних значень, та методи cdf (Cumulative Distribution Function) і sf для обчислення p-значень. Оскільки χ^2-розподіл є несиметричним, для двобічного тесту критична область визначається двома окремими точками на лівому та правому «хвостах» розподілу.

Бібліотека $numpy$ використовується для обчислення виправленої вибіркової дисперсії s^2. При роботі з масивами даних важливо використовувати параметр $ddof = 1$ (Delta Degrees of Freedom), щоб отримати незміщену оцінку дисперсії, яка необхідна для розрахунку статистики тесту.

Приклад реалізації:

```
import numpy as np
from scipy.stats import chi2
# 1. Оброблення даних за допомогою numpy
# Обчислення виправленої дисперсії (ddof=1 забезпечує поділ на n-1)
s_sq = np.var(data, ddof=1)
df = len(data) - 1
# 2. Розрахунок критичних значень (двобічний тест)
# Через несиметричність розподілу знаходимо обидві межі окремо
crit_left = chi2.ppf(alpha / 2, df)
crit_right = chi2.ppf(1 - alpha / 2, df)
# 3. Розрахунок p-значення
# Для двобічного тесту вибирається мінімальне значення з двох хвостів і подвоюється
p_value = 2 * min(chi2.cdf(chi2_stat, df), chi2.sf(chi2_stat, df))
```

Порівняння двох незалежних вибірок (t-тест)

У бібліотеці $scipy.stats$ використовується об'єкт t для перевірки гіпотези про рівність середніх двох сукупностей. Для двобічного тесту ($H_1: a_1 \neq a_2$) критичне значення та p-значення обчислюються з урахуванням сумарної кількості ступенів свободи $k = n_1 + n_2 - 2$.

Бібліотека $numpy$ застосовується для попереднього аналізу кожної вибірки: обчислення їх середніх значень та виправлених дисперсій. Ці показники необхідні для подальшого розрахунку об'єднаної дисперсії та статистики t, яка показує ступінь розбіжності між двома групами:

```
import numpy as np
from scipy.stats import t
# 1. Оброблення характеристик двох вибірок через numpy
x1_bar = np.average(x1, weights=n1)
x2_bar = np.average(x2, weights=n2)
df = n1.sum() + n2.sum() - 2
# 2. Розрахунок критичного значення (двобічний тест)
# Визначає межу, за якою розбіжність вважається значущою
t_crit = t.ppf(1 - alpha / 2, df)
# 3. Розрахунок p-значення
# Використовуємо abs(t_stat), щоб коректно оцінити відхилення в будь-який бік
p_value = 2 * t.sf(abs(t_stat), df)
```

Реалізація в MS Excel

MS Excel може бути використаний як допоміжний інструмент для підготовки вхідних даних (розрахунку проміжних сум n_i, $x_i \cdot n_i$), а також для верифікації результатів, отриманих за допомогою мови Python.

Основні статистичні функції

Для швидкої перевірки обчислених статистик і p-значень рекомендується використовувати такі вбудовані функції у рядку формул:

а) для перевірки середнього (t-тест):

$= T.INV.2T(alpha; df)$ – двобічне критичне значення $t_{\text{крит}}$;

$= T.DIST.2T(ABS(t_s tat); df)$ –двобічне p-значення для отриманої статистики;

б) для перевірки дисперсії (χ^2-тест):

$= CHISQ.INV.RT(alpha/2;\ df)$ – критичне значення правої межі;

$= CHISQ.DIST.RT(chi\_stat;\ df)$ – значення правобічного «хвоста» для розрахунку p-value;

в) для порівняння двох вибірок доцільно використовувати надбудову $Data\ Analysis\ ToolPak$: t-тест: двовибірковий з однаковими дисперсіями.

Використання функцій Excel дає змогу оперативно контролювати правильність роботи програмного коду на Python. Наприклад, значення p-value, отримане через $= T.DIST.2T$, має повністю збігатися з результатом методу $t.sf(abs(t\_stat), df) * 2$.

Порядок виконання роботи

Теоретична підготовка

1. Опрацювати засади статистичної перевірки гіпотез: поняття нульової (H_0) та альтернативної (H_1) гіпотез, рівень значущості α, потужність критерію та p-значення.

2. Вивчити умови застосування параметричних критеріїв: z-критерію (при відомій дисперсії), t-критерію Стьюдента (при невідомій дисперсії) та χ^2-критерію Пірсона для аналізу варіації.

3. Розглянути геометричний зміст критичних областей для односторонніх та двосторонніх альтернатив, а також алгоритми їх побудови залежно від типу розподілу статистики.

4. Ознайомитися з методами обчислення p-значень у бібліотеці $scipy.stats$ та функціями для знаходження критичних точок розподілів (ppf, isf).

5. Підготувати інструментарій: MS Excel (надбудова «Аналіз даних») та середовище Python (Google Colab / IDE).

6. Обрати індивідуальний варіант завдання.

Аналітичне розв'язання (MS Excel / Вручну)

На цьому етапі формується еталонна база результатів для подальшої верифікації програмного коду.

1. Перевірка гіпотез про середнє та дисперсію однієї вибірки:

— сформулювати H_0 про рівність параметра еталону та відповідну H_1 (двосторонню або односторонню);

— обчислити спостережуване значення статистики тесту ($t_{\text{спост}}$ або $\boldsymbol{\chi^2_{\text{спост}}}$);

— знайти критичні значення за таблицями або функціями Excel ($T.INV$, $CHISQ.INV$) та визначити, чи потрапляє статистика в критичну область.

2. Порівняння двох незалежних вибірок:

— виконати розрахунок вибіркових середніх та об’єднаної дисперсії для двох груп даних;

— розрахувати емпіричне значення t-статистики для перевірки гіпотези про рівність математичних сподівань.

Програмна реалізація та використання ШІ (Python)

1. Автоматизація статистичних тестів:

— реалізувати введення даних та обчислення статистики, використовуючи $scipy.stats.ttest\_1samp$ (для однієї вибірки) та $ttest\_ind$ (для двох вибірок);

— програмно обчислити p-значення та порівняти його з заданим рівнем значущості α.

2. Візуалізація та інтеграція ШІ:

— залучити ШІ-асистента для побудови графіків щільності розподілу з автоматичним зафарбовуванням критичних областей за допомогою $matplotlib$;

— використати ШІ для написання функцій, що автоматизують прийняття рішення (виведення тексту «Відхилити H_0» або «Немає підстав для відхилення»);

— за допомогою ШІ проаналізувати чутливість результатів тесту до зміни рівня значущості $\boldsymbol{\alpha}$.

Аналіз результатів та оформлення звіту

1. Верифікація результатів: порівняти p-значення, отримані в Python, із результатами порівняння статистики та критичних точок в Excel. Проаналізувати причини можливих розбіжностей.

2. Інтерпретація статистичних висновків: пояснити практичний зміст прийнятого рішення (наприклад, чи є відхилення в точності датчиків критичним, чи є різниця в доходах заводів суттєвою).

3. Рефлексія щодо використання ШІ: оцінити ефективність використання ШІ при інтерпретації складних випадків (наприклад, коли p-значення дуже близьке до α) та при дебагінгу логічних помилок у коді.

4. Сформулювати загальні висновки, підсумувавши результати перевірки всіх висунутих гіпотез.

Вимоги до звіту та його вміст

Звіт є підсумковим документом, що фіксує результати проведеного статистичного дослідження щодо перевірки гіпотез. Його структура має містити такі обов'язкові компоненти:

1. Титульна сторінка.

Оформлюється згідно з установленим зразком. Має містити назву дисципліни, номер роботи, дані здобувача освіти та викладача.

2. Тема, мета роботи та номер варіанта.

Формулюються чітко відповідно до методичних вказівок.

3. Постановка завдання.

Необхідно повністю скопіювати умову завдання зі свого індивідуального варіанта (вихідні дані вибірки, гіпотетичні значення a_0 або σ_0^2, рівень значущості α).

4. Результати виконання.

Це основний розділ звіту, що складається з трьох частин:

а) розрахунки (MS Excel / Вручну):

— таблиця розрахунку вибіркових характеристик (середнє $\bar{x}$, виправлена дисперсія s^2, обсяг вибірки n);

— розрахунок спостережуваного значення критерію ($t_{\text{спост}}$, $\chi^2_{\text{спост}}$ або $F_{\text{спост}}$);

— визначення критичних точок розподілу для заданого рівня значущості α;

б) код програми: повний програмний код на Python з коментарями, що реалізує обчислення статистик, знаходження p-значень та побудову графіків візуалізації;

в) результати роботи програми:

— скріншот текстового виведення розрахованих показників (статистика тесту, p-value, критичні значення);

— *графік 1*: щільність імовірності відповідного розподілу (t, chi^2 або F) із заштрихованою критичною областю;

— *графік 2*: положення спостережуваного значення статистики на кривій розподілу відносно критичних меж.

5. Аналіз використання ШІ-асистента.

У цьому розділі здобувач освіти має продемонструвати навички критичної взаємодії з технологіями штучного інтелекту:

а) вибір програмного інструментарію ШІ. Вказати повну назву та версію використаної моделі (наприклад, Gemini 2.0 Flash, ChatGPT 4o, Claude 3.5);

б) роль ШІ у виконанні роботи.

Навести 2–3 конкретні приклади запитів (промптів) та проаналізувати отримані відповіді. Наприклад, як ШІ допоміг вибрати метод $t.sf$ замість $t.cdf$, пояснив логіку подвоєння p-значення для двобічного тесту або допоміг виправити помилку у формулі об'єднаної дисперсії.

6. Висновки.

Розділ має містити підсумок проведеного дослідження, розділений на два аспекти:

а) технічний висновок: необхідно провести аналіз результатів тестування. На основі порівняння статистики з критичним значенням (або $p-$value з α) зробити аргументований висновок про прийняття або відхилення нульової гіпотези. Пояснити практичне значення результату (наприклад, чи є підстави вважати обладнання несправним або зміну доходу значущою);

б) рефлексія щодо використання ШІ.

Критично оцінити ефективність допомоги асистента. Описати, чи сприяв ШІ глибшому розумінню логіки статистичного висновку (помилки I та II роду, рівень значущості), чи його роль обмежилася технічним написанням коду для бібліотеки $scipy.stats$.

Завдання для самостійного виконання

Завдання 1. Аналіз часу оброблення замовлень на складі.

Досліджується час оброблення замовлень (у хвилинах), який підкоряється нормальному розподілу.

Дані вибірки:

x_i, *хв*	4.2	6.2	8.2	10.2	12.2
n_i	6	8	12	8	2

При рівні значущості α = 0,05 перевірити гіпотезу про те, що середній час оброблення становить 8 хвилин ($H_0: a = 8$) проти альтернативної гіпотези ($H_1: a \neq 8$). Пояснити практичне значення результату.

Завдання 2. Перевірка точності датчиків температури.

Перевіряється дисперсія вимірювань датчиків температури, які підпорядковуються нормальному розподілу.

Дані вибірки: обсяг $n = 20$, виправлена вибіркова дисперсія $s^2 = 7.5$.

При рівні значущості $\alpha = 0{,}01$ перевірити нульову гіпотезу про відповідність стандарту (H_0: $\sigma^2 = 5$) проти трьох альтернативних:

— H_1: $\sigma^2 \neq 5$ (відхилення від стандарту);

— H_1: $\sigma^2 > 5$ (надмірна мінливість);

— H_1: $\sigma^2 < 5$ (вища точність).

Пояснити, як результати вплинуть на рішення щодо калібрування датчиків.

Завдання 3. Порівняння доходів працівників двох заводів.

Порівнюються місячні доходи працівників двох заводів, які є незалежними та нормально розподіленими величинами.

Дані вибірок:

Завод 1 (x_i, грн)	150.6	160.6	170.6	180.6	190.6
Кількість працівників n_i	12	28	40	18	2
Завод 2 (y_i, грн)	140.8	160.8	180.8	200.8	220.8
Кількість працівників n_i	2	6	32	8	2

При рівні значущості $\alpha = 0{,}05$ перевірити гіпотезу про рівність середніх доходів (H_0: $E[X] = E[Y]$) проти альтернативної гіпотези (H_1: $E[X] \neq E[Y]$). Пояснити практичне значення результату.

Контрольні запитання

1. Що таке нульова та альтернативна гіпотези в контексті перевірки закону розподілу?
2. Що таке нульова та альтернативна гіпотези? Наведіть приклад.
3. У чому різниця між z-критерієм та t-критерієм для перевірки гіпотез про середнє?
4. Що таке рівень значущості (α) та потужність критерію?
5. Як визначається кількість ступенів свободи для t-тесту та χ^2-тесту?
6. Поясніть, що таке p-значення і як його використовувати для прийняття рішення.
7. Для перевірки якої гіпотези використовується критерій χ^2?
8. Яка функція в бібліотеці *scipy.stats* використовується для проведення t-тесту для двох незалежних вибірок?
9. Як візуально зобразити критичну область на графіку розподілу?
10. Що таке помилка першого та другого роду в статистиці?
11. Як ШІ може допомогти уникнути поширених помилок при інтерпретації результатів статистичних тестів?
12. У чому полягає особливість двосторонньої критичної області порівняно з односторонньою при розрахунку критичного значення?
13. Чому при розрахунку вибіркової дисперсії в Python (бібліотека NumPy) важливо використовувати параметр ddof=1?
14. Які умови мають виконуватися для вибіркових даних, щоб результати t-тесту вважалися достовірними?

15. Як зміниться ймовірність помилки другого роду (β), якщо зменшити рівень значущості α (наприклад, з 0.05 до 0.01)?

16. Як за допомогою ШІ-асистента можна перевірити стійкість отриманого статистичного висновку до аномальних значень (викидів) у вибірці?

Практична робота № 7
ЗАСТОСУВАННЯ КРИТЕРІЮ УЗГОДЖЕНОСТІ ПІРСОНА (X^2)

Мета роботи:

— ознайомитися з перевіркою статистичних гіпотез про відповідність даних теоретичним законам розподілу;

— опанувати методику розрахунку статистичних характеристик вибірки та теоретичних частот;

— навчитися застосовувати критерій узгодженості Пірсона (χ^2) для прийняття обґрунтованих рішень;

— оволодіти методами візуальної верифікації гіпотез через побудову суміщених графіків частот.

Компетентності, що формуються:

— здатність коректно формулювати гіпотези H_0 та H_1 щодо законів розподілу досліджуваних даних;

— навички розрахунку емпіричних та теоретичних частот для основних типів статистичних розподілів;

— уміння застосовувати алгоритм критерію χ^2 для оцінювання міри розбіжності між даними;

— здатність обґрунтовувати статистичні висновки на основі аналізу p-value та критичних меж;

— володіння інструментарієм Python ($scipy$, $numpy$, $matplotlib$) для аналізу та візуалізації даних;

— ефективне використання ШІ-асистентів для написання, перевірки та інтерпретації коду.

Необхідне програмне забезпечення та бібліотеки:

1. Середовище розроблення: Google Colab, Jupyter Notebook або будь-яке інше інтегроване середовище розроблення (IDE) для Python (наприклад, VS Code, PyCharm).
2. Python версії 3.8 або новішої з бібліотеками: $numpy$, $scipy$, $matplotlib$, $pandas$.
3. MS Excel із встановленою надбудовою $Data\ Analysis\ ToolPak$.

Теоретичні відомості

У статистиці важливо визначити, якому закону розподілу відповідають емпіричні дані, наприклад, нормальному, рівномірному, показниковому, біноміальному чи розподілу Пуассона. Для перевірки відповідності даних

певному розподілу використовують різні критерії, одним із найпоширеніших є критерій Пірсона (χ^2), який порівнює емпіричні частоти з теоретичними, оцінюючи їх узгодженість.

Критерій узгодженості Пірсона (χ^2)

Критерій Пірсона є універсальним методом перевірки статистичних гіпотез про відповідність емпіричних даних теоретичному розподілу. Його функціональне призначення полягає в оцінюванні розбіжностей між фактичними частотами, отриманими внслідок спостережень, та очікуваними частотами, розрахованими на основі гіпотетичної моделі.

Статистика критерію: для оцінки узгодженості використовується сума квадратів відхилень, нормованих відносно теоретичних частот:

$$\chi^2_{\text{спост}} = \sum_{i=1}^{k} \frac{(n_i - n_i')^2}{n_i'} \ ,$$

де n_i – емпіричні частоти (дані вибірки);

n_i' – теоретичні частоти (очікувані значення);

k – кількість інтервалів (категорій) групування.

Процедура проведення перевірки реалізується за таким алгоритмом:

1. Формулюють гіпотези:

— H_0: дані відповідають вибраному теоретичному розподілу;

— H_1: дані не відповідають вибраному розподілу.

2. Розраховують теоретичні частоти (n_i'): на основі обсягу вибірки n та теоретичних ймовірностей p_i визначають значення $n_i' = n \cdot p_i$.

3. Перевіряють умови застосовності:

— критерій коректно працює за умови $n_i' \geq 5$ для кожного інтервалу;

— якщо ця умова не виконується, малочисельні інтервали об'єднують із сусідніми, відповідно зменшуючи кількість категорій k.

4. Визначають кількість ступенів свободи (df) за формулою

$$df = k - m - 1,$$

де m – кількість параметрів розподілу, що оцінювалися за вибіркою.

Незалежно від вибраного закону розподілу (рівномірний, показниковий, біноміальний чи Пуассона), під символом k завжди слід розуміти кількість груп (інтервалів), що залишилися після об'єднання.

5. Приймають рішення щодо відповідності вибраному занону розподілу.

Обчислене значення $\chi^2_{\text{спост}}$ порівнюють із критичним $\chi^2_{\alpha,df}$, знайденим за таблицями для заданого рівня значущості α:

— якщо $\chi^2_{\text{спост}} \leq \chi^2_{\text{крит}}$, то нульова гіпотеза H_0 не відхиляється;

— якщо $\chi^2_{\text{спост}} > \chi^2_{\text{крит}}$, дані не відповідають вибраному закону розподілу.

Інтерпретація та обмеження:

а) критерій дає достовірні результати при $n \geq 50$. Якщо обсяг вибірки $n < 30$ або теоретичні частоти $n_i' < 5$, результати можуть бути статистично неточними, що вимагає об'єднання інтервалів;

б) невідхилення H_0 не доводить абсолютну ідентичність розподілів, а лише свідчить про те, що наявні розбіжності мають випадковий характер і не є статистично значущими;

в) якщо гіпотеза H_0 відхиляється, це може вказувати на некоректний вибір теоретичного закону розподілу, наявність у вибірці аномальних викидів (екстремальних значень) або порушення незалежності спостережень.

Приклад застосування: перевірка часу очікування клієнтів у черзі на відповідність показниковому розподілу. Якщо дані підтверджують гіпотезу, це дає змогу використовувати апарат теорії масового обслуговування для розрахунку оптимальної кількості кас або операторів.

Нормальний розподіл

Нормальний (гаусівський) розподіл є основою багатьох статистичних методів завдяки своїм унікальним властивостям і центральній граничній теоремі, згідно з якою сума багатьох незалежних випадкових величин із будь-яким розподілом наближається до нормального за великої кількості доданків. Він характеризується симетричним дзвоноподібним розподілом, де більшість значень зосереджені біля середнього, а ймовірність крайніх значень швидко зменшується.

Нормальний розподіл є базовим для аналізу даних у біології (наприклад, зріст, вага), техніці (похибки вимірювань), економіці (фінансові показники).

Властивості та параметри:

— розподіл є симетричним відносно середнього значення $\bar{x}$;

— 68 % даних лежать у межах одного стандартного відхилення ($\pm\sigma$) від середнього, 95 % – у межах двох (2σ), 99.7 % – у межах трьох (3σ) – правило трьох сигм;

— математичне сподівання $M(x) = a$ (де a – параметр центру, що для нормального закону збігається з модою та медіаною);

— дисперсія $D(x) = \sigma^2$ (характеризує ступінь «розмитості» дзвона).

— нормальний розподіл є граничним для багатьох інших розподілів (наприклад, біноміального за великих n).

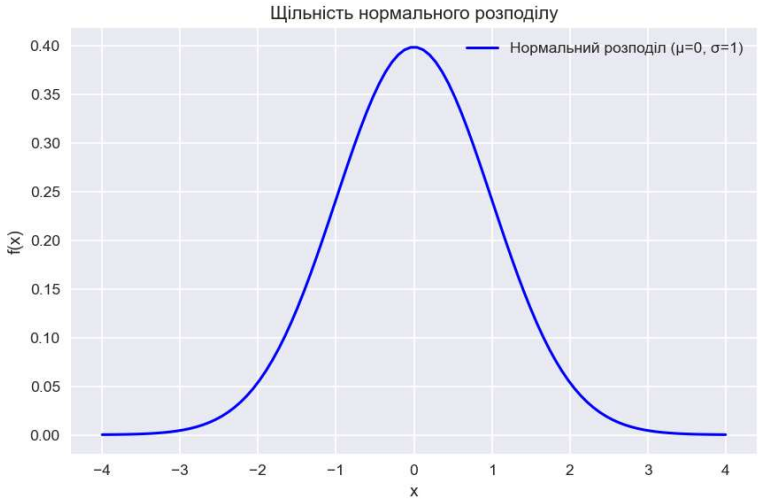
Щільність ймовірності

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}},$$

де $\bar{x} = \frac{1}{n}\sum x_i n_i$ – вибіркове середнє (оцінка математичного сподівання);

$\sigma^2 = \frac{1}{n}\sum(x_i - \bar{x})^2 n_i$ – дисперсія (оцінка варіації даних);

$\sigma = \sqrt{\sigma^2}$ – стандартне відхилення.



Застосування критерію Пірсона для нормального закону

При перевірці відповідності емпіричних даних нормальному розподілу процедура має такі етапи:

— за вибіркою обчислюють $\bar{x}$ і σ (параметризація);

— для кожного інтервалу $[x_i, x_{i+1}]$ обчислюють нормовані межі:

$$z_i = \frac{x_i - \bar{x}}{\sigma} \text{ (нормування меж);}$$

— розраховуються теоретичні частоти:

$$n_i' = n \cdot [\Phi(z_{i+1}) - \Phi(z_i)],$$

де $\Phi(z)$ – функція Лапласа, що відповідає стандартному нормальному розподілу $(N(0,1))$;

— проводиться оброблення даних шляхом порівняння емпіричних n_i та теоретичних n_i' частот за статистикою χ^2. Якщо частоти менші за 5, сусідні інтервали об'єднують для забезпечення надійності критерію.

Кількість ступенів свободи

Для нормального розподілу кількість ступенів свободи df розраховується за формулою

$$df = k - m - 1,$$

де k – кількість інтервалів групування вибірки; m — кількість параметрів, що оцінюються за вибіркою.

Вибір значення m залежить від повноти вхідних даних:

— $df = k - 3$ $(m = 2)$: якщо обчислюється вибіркове середнє $(\bar{x})$ та вибіркова дисперсія (s^2) за даними спостережень. Це стандарт для більшості робіт;

— $df = k - 1$ $(m = 0)$: якщо параметри теоретичного розподілу (математичне сподівання a та стандартне відхилення σ) задані заздалегідь

в умові задачі. У цьому випадку вибірка лише перевіряється на відповідність уже готовому еталону.

Практичні аспекти

1. Нормальний розподіл часто використовується як припущення в статистичних тестах (t-тест, ANOVA).
2. Відхилення від нормальності може вказувати на наявність аномалій або потребу в іншій моделі.
3. Для великих вибірок оцінки параметрів є точними, але для малих вибірок можуть бути упередженими.

Рівномірний розподіл

Рівномірний розподіл характеризується однаковою ймовірністю для всіх значень у межах інтервалу $[a, b]$. Він моделює ситуації, де немає переваги для певних значень, наприклад, випадковий вибір числа, час прибуття транспорту за графіком або стабільний розподіл продажів. Рівномірний розподіл є простим, але рідко зустрічається в реальних даних через ідеалізовану природу.

Властивості та параметри:

— усі значення в інтервалі $[a, b]$ мають однакову щільність імовірності;

— математичне сподівання $M(x) = \frac{a+b}{2}$;

— дисперсія $D(x) = \frac{(b-a)^2}{12}$;

— є базовим у комп'ютерних симуляціях (генерація псевдовипадкових чисел).

Щільність ймовірності

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

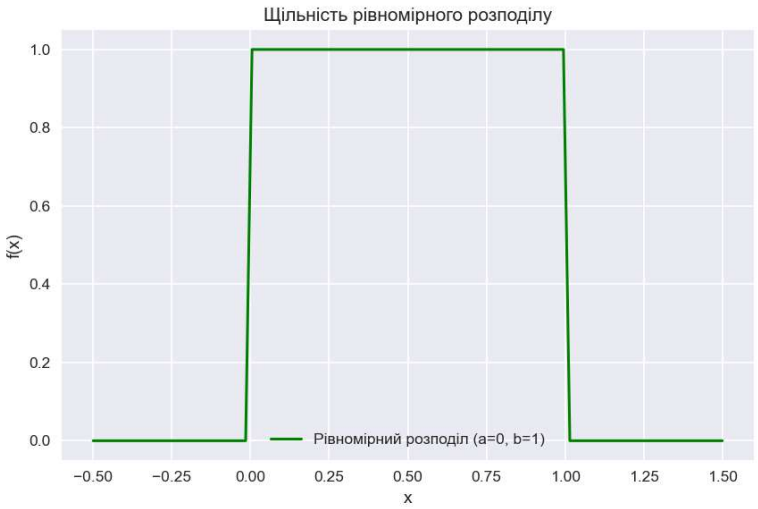
де замість теоретичних меж a та b використовують їх вибіркові оцінки a^* та b^*, тобто наближені значення, отримані за формулами, а не істинні параметри моделі:

$$a^* = \bar{x} - \sqrt{3s^2},$$
$$b^* = \bar{x} + \sqrt{3s^2}.$$

Для їх розрахунку обчислюють характеристики вибірки:

— $\bar{x} = \frac{\sum x_i n_i}{\sum n_i}$ – вибіркове середнє;

— $s^2 = \frac{\sum (x_i - \bar{x})^2 n_i}{\sum n_i}$ – вибіркова дисперсія.



Застосування критерію Пірсона для рівномірного закону

При перевірці відповідності емпіричних даних рівномірному розподілу процедура має такі етапи:

— за даними вибірки обчислюють $\bar{x}$ і σ, після чого оцінюють межі a^* і b^* (параметризація);

— визначають теоретичні частоти, пропорційні довжині інтервалів:

$$n_i' = \frac{n \cdot (x_{i+1} - x_i)}{b^* - a^*};$$

— проводиться оброблення даних шляхом порівняння емпіричних n_i та теоретичних n_i' частот за статистикою χ^2. Якщо частоти менші за 5, сусідні інтервали об'єднують для забезпечення надійності критерію.

Кількість ступенів свободи

Для рівномірного розподілу кількість ступенів свободи df розраховується за формулою

$$df = k - m - 1,$$

де k – кількість інтервалів групування вибірки; m – кількість параметрів, що оцінюються за вибіркою.

Значення df залежить від умов задачі:

— $df = k - 3$ $(m = 2)$: якщо межі інтервалу a та b не були задані, і їх оцінки (a^* та b^*) обчислювались через вибіркове середнє ($\bar{x}$) та вибіркову дисперсію (s^2). Це найбільш розповсюджений випадок при опрацюванні результатів експерименту;

— $df = k - 1$ $(m = 0)$: якщо теоретичні межі інтервалу $[a, b]$ були задані заздалегідь в умові задачі (наприклад, технічні характеристики приладу або часовий інтервал руху транспорту). У цьому випадку параметри за вибіркою не оцінюються, а лише перевіряється відповідність даних встановленим межам.

Практичні аспекти

1. Рівномірний розподіл рідко точно описує реальні дані, але є корисним для моделювання стабільних процесів.

2. Оцінка меж a^* і b^* може бути чутливою до викидів у вибірці, що вимагає попередньої перевірки даних.

3. Застосовується в криптографії для перевірки генераторів випадкових чисел.

Показниковий (експоненціальний) розподіл

Показниковий розподіл є неперервним і моделює час між послідовними подіями в пуассонівському процесі, наприклад, час між поломками обладнання, дзвінками до кол-центру або запитами до сервера. Його ключова властивість – «безпам'ятність»: імовірність події не залежить від часу, що минув. Показниковий розподіл є частковим випадком гамма-розподілу з параметром форми 1.

Властивості та параметри:

— математичне сподівання та дисперсія:

$$M(x) = \frac{1}{\lambda}, \quad D(x) = \frac{1}{\lambda^2},$$

де λ (лямбда) – параметр інтенсивності (середня кількість подій, що відбуваються за одиницю часу);

— стандартне відхилення дорівнює математичному сподіванню ($\sigma = M(x) = 1/\lambda$). Це головна ознака, за якою можна ідентифікувати показниковий розподіл у вибірці;

— на відміну від нормального розподілу, показниковий має найбільшу щільність імовірності біля нуля, яка стрімко спадає із зростанням x. Це відображає високу ймовірність появи коротких інтервалів між подіями;

— тісний зв'язок із розподілом Пуассона: якщо кількість подій за одиницю часу описується законом Пуассона з параметром λ, то час між цими подіями завжди має показниковий розподіл з тим самим параметром.

Щільність ймовірності

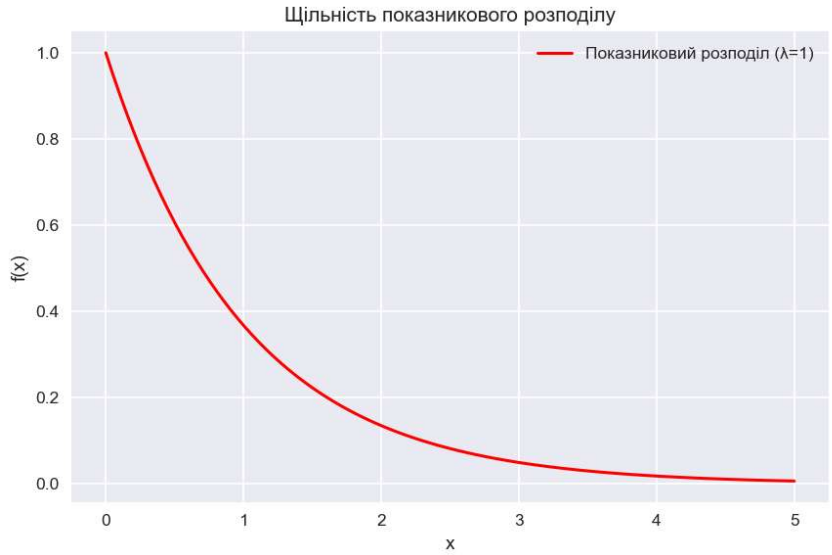
$$f(x) = \begin{cases} 0, & x < 0, \\ \lambda e^{-\lambda x}, & x \geq 0, \end{cases}$$

де замість теоретичного параметра інтенсивності λ використовують його вибіркову оцінку λ^*, тобто наближене значення, отримане за формулою, а не істинний параметр моделі:

$$\lambda^* = \frac{1}{\bar{x}}.$$

Для його розрахунку обчислюють характеристику вибірки:

$\bar{x} = \frac{\sum x_i n_i}{\sum n_i}$ – вибіркове середнє (середній час очікування події).



Застосування критерію Пірсона для показникового закону

При перевірці відповідності емпіричних даних показниковому розподілу процедура має такі етапи:

— за даними вибірки обчислюють вибіркове середнє $\bar{x}$, після чого оцінюють параметр інтенсивності λ^* (параметризація):

$$\lambda^* = \frac{1}{\bar{x}};$$

— визначають теоретичні частоти, використовуючи функцію розподілу. Для кожного j-го інтервалу $[x_j; x_{j+1}]$ теоретична частота обчислюється як

$$n'_j = n \cdot \left(e^{-\lambda^* x_j} - e^{-\lambda^* x_{j+1}}\right);$$

— проводиться оброблення даних шляхом порівняння емпіричних n_j та теоретичних n'_j частот за статистикою χ^2. Якщо частоти менші за 5, сусідні інтервали об'єднують для забезпечення надійності критерію.

Кількість ступенів свободи

Для показникового розподілу кількість ступенів свободи df розраховується за формулою

$$df = k - m - 1,$$

де k – кількість інтервалів групування вибірки; m – кількість параметрів, що оцінюються за вибіркою.

Значення df залежить від умов задачі:

– $df = k - 2$ $(m = 1)$: якщо параметр інтенсивності λ^* не був заданий, і його оцінку обчислювали через вибіркове середнє $(\bar{x})$. Оскільки показниковий розподіл визначається лише одним параметром, це найбільш розповсюджений випадок при опрацюванні результатів експерименту;

– $df = k - 1$ $(m = 0)$: якщо теоретичний параметр інтенсивності λ був заданий заздалегідь в умові задачі (наприклад, середня інтенсивність

відмов, указана в паспорті обладнання). У цьому випадку параметр за вибіркою не оцінюється, а лише перевіряється відповідність даних встановленому еталону.

Практичні аспекти

1. Показниковий розподіл ідеально описує тривалість інтервалів між незалежними подіями, що відбуваються з постійною середньою інтенсивністю.
2. На відміну від рівномірного, цей розподіл має виражену асиметрію: більшість значень зосереджена біля нуля (короткі інтервали), а довгі періоди очікування трапляються значно рідше.
3. Оцінка параметра λ через вибіркове середнє $\bar{x}$ є дуже простою, проте вона чутлива до «бездіяльності» системи – якщо процес зупинявся, це спотворить результати.
4. Ключова властивість – відсутність післядії: ймовірність того, що подія відбудеться в наступну хвилину, не залежить від того, скільки вже чекали.

Біноміальний розподіл

Біноміальний розподіл є дискретним і описує кількість успіхів у (n) незалежних випробуваннях, де кожне випробування має два результати (успіх із ймовірністю (p), невдача з ймовірністю $(1-p)$). Він застосовується для моделювання кількості дефектних виробів, кліків на рекламу або позитивних відповідей у опитуванні. За великих (n) і помірного (p) біноміальний розподіл наближається до нормального, а за малого (p) – до пуассонівського.

Властивості та параметри

— математичне сподівання та дисперсія:

$$M(x) = np, \quad D(x) = npq,$$

де n – кількість незалежних випробувань; p – ймовірність успіху в одному випробуванні; $q = 1 - p$ – ймовірність невдачі;

— на відміну від показникового розподілу, тут дисперсія завжди менша за математичне сподівання ($D(x) < M(x)$), оскільки $q < 1$;

— форма розподілу при $p = 0.5$ є симетричною. Якщо n велике, він стає схожим на нормальний (дзвоноподібний), але залишається дискретним;

— при дуже малому p та великому n біноміальний розподіл наближається до розподілу Пуассона, що дає змогу спрощувати розрахунки для рідкісних подій.

Ймовірність події (закон розподілу)

Замість щільності (як у неперервних розподілах) тут використовують формулу Бернуллі для розрахунку ймовірності отримання рівно k успіхів:

$$P_n(k) = C_n^k \cdot p^k \cdot q^{n-k},$$

де замість теоретичної ймовірності p використовують її вибіркову оцінку p^*, отриману за формулою

$$p^* = \frac{\bar{x}}{n}.$$

Для розрахунку обчислюють характеристики вибірки:

— $\bar{x} = \frac{\sum x_i n_i}{\sum n_i}$ – вибіркова середня кількість успіхів;

— n – загальна кількість випробувань в одній серії.

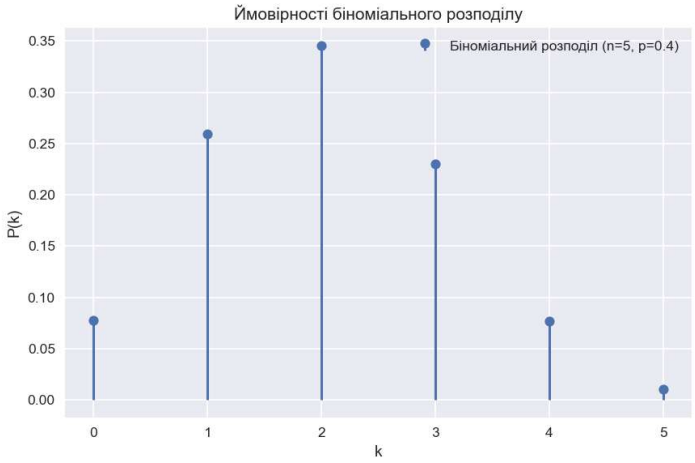
Кількість ступенів свободи

Для біноміального розподілу кількість ступенів свободи df розраховується за формулою

$$df = k - m - 1;$$

— $df = k - 2$ $(m = 1)$: якщо ймовірність p^* оцінювалася за даними вибірки через вибіркове середнє;

— $df = k - 1$ $(m = 0)$: якщо ймовірність p була відома заздалегідь (наприклад, імовірність випадання герба при підкиданні монети $p = 0.5$).



Застосування критерію Пірсона для біноміального закону

При перевірці відповідності емпіричних даних біноміальному розподілу процедура має такі етапи:

— за даними вибірки обчислюють вибіркове середнє $\bar{x}$ (середню кількість успіхів), після чого оцінюють імовірність успіху p^*(параметризація):

$$p^* = \frac{\bar{x}}{n},$$

де n – кількість випробувань у кожній серії;

— визначають теоретичні частоти для кожного значення k (кількості успіхів від 0 до n):

$$n'_k = N\, P_n(k) = N\, (C_n^k \cdot (p^*)^k \cdot (1 - p^*)^{n-k}),$$

де N – загальна кількість серій випробувань;

— проводиться оброблення даних шляхом порівняння емпіричних n_kта теоретичних n'_k частот за статистикою χ^2. Якщо теоретичні частоти для певних значень k менші за 5, ці варіанти об'єднують із сусідніми для забезпечення надійності критерію.

Практичні аспекти

1. Біноміальний розподіл є базовим для контролю якості, коли кожен виріб класифікується як «придатний» або «бракований».
2. На відміну від неперервних розподілів, тут працюємо з цілими числами (кількістю успіхів), тому графік розподілу – це набір окремих стовпчиків (гістограма), а не суцільна лінія.

Пуассонівський розподіл

Пуассонівський розподіл є дискретним і описує кількість рідкісних, незалежних подій за фіксований інтервал часу або простору, наприклад, кількість дзвінків у кол-центрі, дефектів на одиницю продукції або аварій за день. Він є граничним випадком біноміального розподілу за великих (n) і малого (p), де $(\mathrm{np} = \lambda)$.

Властивості та параметри

— математичне сподівання та дисперсія:

$$M(x) = \lambda, \quad D(x) = \lambda,$$

де λ (лямбда) – параметр інтенсивності (середня кількість подій за одиницю часу або простору);

— унікальною ознакою розподілу Пуассона є рівність математичного сподівання та дисперсії $(M(x) = D(x))$. Це ключовий індикатор для вибору цього закону при аналізі вибірки;

— при малих значеннях λ розподіл має сильну правосторонню асиметрію. Зі зростанням λ розподіл стає більш симетричним і при $\lambda > 20$ наближається до нормального;

— закон Пуассона часто називають «законом рідкісних подій», оскільки він описує біноміальний розподіл, у якому кількість випробувань $n \to \infty$, а ймовірність успіху $p \to 0$ при стабільному добутку $np = \lambda$.

Ймовірність події (закон розподілу)

Ймовірність того, що за фіксований інтервал відбудеться рівно k подій, обчислюється за формулою

$$P(k) = \frac{\lambda^k \cdot e^{-\lambda}}{k!},$$

де замість теоретичного параметра λ використовують його вибіркову оцінку λ^*:

$$\lambda^* = \overline{x}.$$

Для розрахунку обчислюють характеристику вибірки: $\bar{x} = \frac{\sum x_i n_i}{\sum n_i}$ – вибіркове середнє (середня кількість подій у вибірці).

Застосування критерію Пірсона

При перевірці відповідності даних закону Пуассона етапи процедури такі:

— за даними вибірки обчислюють вибіркове середнє $\bar{x}$, що одночасно є оцінкою параметра інтенсивності (параметризація):

$$\lambda^* = \bar{x};$$

— визначають теоретичні частоти для кожного значення k (кількості подій), використовуючи закон Пуассона:

$$n'_k = N \cdot \frac{(\lambda^*)^k \cdot e^{-\lambda^*}}{k!},$$

де N – загальний обсяг вибірки (сума всіх емпіричних частот);

— проводиться оброблення даних шляхом порівняння емпіричних n_k та теоретичних n'_k частот за статистикою χ^2. Якщо частоти (емпіричні або теоретичні) для окремих значень k менші за 5, ці групи об'єднують із сусідніми для забезпечення надійності критерію.

Кількість ступенів свободи

Для розподілу Пуассона кількість ступенів свободи df розраховується за формулою

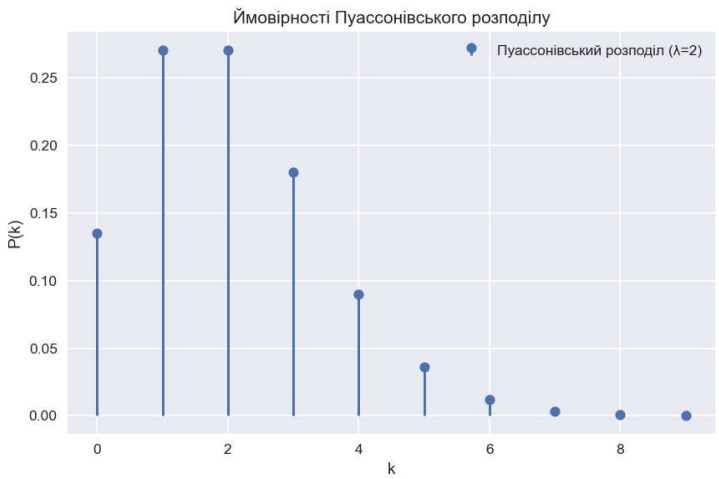
$$df = k - m - 1,$$

де k – кількість груп (категорій) подій у вибірці, а m – кількість параметрів, що оцінюються.

Значення df залежить від походження параметра λ:

— $df = k - 2$ $(m = 1)$: якщо параметр λ^* оцінювався за даними вибірки через вибіркове середнє $(\bar{x})$. Це стандартна ситуація для більшості робіт;

— $df = k - 1$ $(m = 0)$: якщо параметр λ був відомий заздалегідь (наприклад, заданий в умові як історично встановлена норма інтенсивності подій).



Практичні аспекти та приклади

Розподіл використовується для моделювання потоків подій, де кожна подія не залежить від попередньої.

Приклади: кількість друкарських помилок на сторінці книги, кількість викликів швидкої допомоги за годину, кількість дефектів на одному метрі тканини або кількість частинок, що вилітають при радіоактивному розпаді.

Застосування Python для виконання завдань

Для виконання завдань практичної роботи Python надає ефективні інструменти для оброблення даних, статистичних обчислень і візуалізації. Розглянемо ключові етапи виконання завдань із використанням можливостей Python для аналізу даних.

Організація та підготовка даних

Цей етап передбачає введення даних, які складаються з інтервалів (для неперервних розподілів) або категорій (для дискретних розподілів) із відповідними частотами. Бібліотека **Pandas** забезпечує зручне створення табличних структур, тоді як **NumPy** дає змогу працювати з масивами меж інтервалів.

Приклад реалізації:

```
import pandas as pd
import numpy as np

# Створення таблиці з даними

data = pd.DataFrame({
'interval': ['[0-10]', '[10-20]', '[20-30]', '[30-40]'],
'frequency': [15, 25, 20, 10]})
bin_edges = np.array([0, 10, 20, 30, 40])
# межі інтервалів
```

Така організація даних є основою для подальшого статистичного аналізу.

Оцінювання параметрів розподілу

Для перевірки відповідності даних розподілу необхідно оцінити його параметри, такі як вибіркове середнє ($\bar{x}$) , дисперсія (s^2), параметр інтенсивності (λ) або ймовірність успіху (p). Бібліотека **NumPy** забезпечує точні обчислення на основі середин інтервалів і частот.

Приклад реалізації:

```
# Нормальний розподіл
midpoints = (bin_edges[:-1] + bin_edges[1:]) / 2
mean = np.sum(midpoints * data['frequency']) / data['frequency'].sum()
variance = np.sum(data['frequency'] * (midpoints - mean)**2) /
data['frequency'].sum()
std = np.sqrt(variance)

# Рівномірний розподіл
a_est = mean - np.sqrt(3 * variance)
```

```
b_est = mean + np.sqrt(3 * variance)

# Показниковий розподіл
lambda_est = 1 / mean

# Біноміальний розподіл
n = 10  # кількість випробувань
p_est = mean / n

# Пуассонівський розподіл
lambda_est = mean
```

Ці оцінки формують основу для обчислення теоретичних частот.

Розрахунок теоретичних частот

Теоретичні частоти $(n_i' = n \cdot p_i)$ розраховуються на основі ймовірностей (p_i), що залежать від типу розподілу. Модуль **SciPy.stats** містить функції для обчислення ймовірностей для кожного розподілу, що значно полегшує цей процес.

Приклад реалізації:

```
from scipy.stats import norm, uniform, expon, binom, poisson

# Нормальний розподіл
z_scores = (bin_edges - mean) / std
p_i = norm.cdf(z_scores[1:]) - norm.cdf(z_scores[:-1])
expected = data['frequency'].sum() * p_i

# Рівномірний розподіл
p_i = uniform.cdf(bin_edges[1:], a_est, b_est-a_est) - uniform.cdf(bin_edges[:-1], a_est, b_est-a_est)
expected = data['frequency'].sum() * p_i

# Показниковий розподіл
p_i = expon.cdf(bin_edges[1:], scale=1/lambda_est) - expon.cdf(bin_edges[:-1], scale=1/lambda_est)
expected = data['frequency'].sum() * p_i

# Біноміальний розподіл
# k має відповідати числовим значенням категорій (0, 1, 2...)
k = np.arange(len(data['frequency']))
p_i = binom.pmf(k, n, p_est)
expected = data['frequency'].sum() * p_i

# Пуассонівський розподіл
p_i = poisson.pmf(k, lambda_est)
expected = data['frequency'].sum() * p_i
```

Отримані частоти використовуються для порівняння з емпіричними даними.

Застосування критерію Пірсона

Критерій Пірсона оцінює відповідність емпіричних частот (n_i) теоретичним (n_i') шляхом обчислення статистики $\chi 2$. NumPy використовується для розрахунків, а SciPy.stats – для визначення критичного значення та прийняття рішення щодо гіпотези.

Приклад реалізації:

```
from scipy.stats import chi2
import numpy as np

chi2_stat = np.sum((data['frequency'] - expected)**2 / expected)
k = len(data['frequency'])
# Вибір кількості параметрів залежно від розподілу
m = 2 # 2 для норм./рівномірн., 1 для Пуассона/показник.
df = k - 1 - m
alpha = 0.05
chi2_crit = chi2.ppf(1 - alpha, df)
print(f"Статистика χ²: {chi2_stat:.2f}, Критичне значення: {chi2_crit:.2f}")
if chi2_stat < chi2_crit:
    print("Дані відповідають гіпотетичному розподілу.")
else:
    print("Дані не відповідають гіпотетичному розподілу.")
```

Цей етап завершує аналіз відповідності даних заданому розподілу.

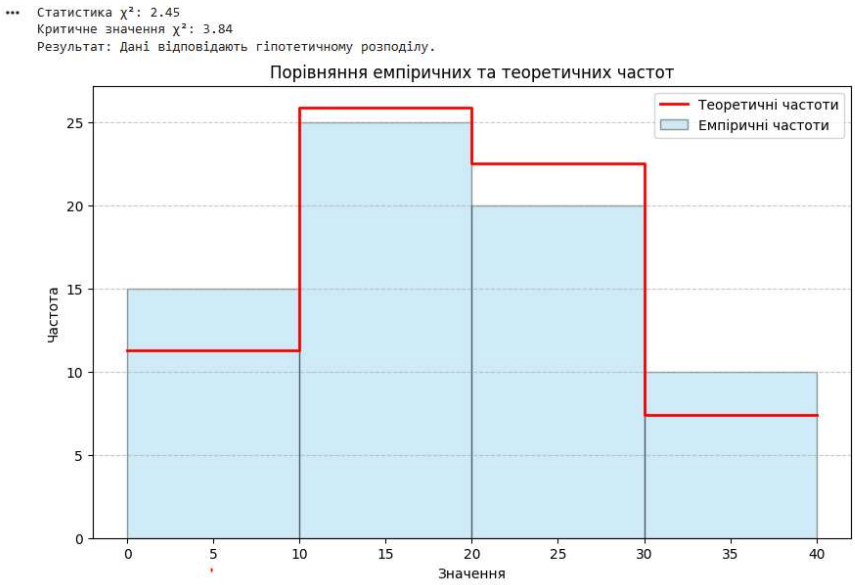
Візуалізація результатів

Для наочного подання результатів створені гістограми або стовпчикові діаграми, що порівнюють емпіричні та теоретичні частоти. Бібліотека Matplotlib забезпечує гнучкі інструменти для побудови таких графіків.

Приклад реалізації:

```
import matplotlib.pyplot as plt
import numpy as np

# Неперервні розподіли
plt.figure(figsize=(8, 5))
# Додано align='edge', щоб стовпчик займав увесь інтервал від лівої межі
plt.bar(bin_edges[:-1], data['frequency'], width=np.diff(bin_edges),
        alpha=0.5,       label='Емпіричні       частоти',       align='edge',
edgecolor='black')

# Теоретичні частоти (малюємо вужчими стовпчиками по центру для порівняння)
plt.bar(bin_edges[:-1]        +        np.diff(bin_edges)/2,        expected,
width=np.diff(bin_edges)/2,
        alpha=0.7, label='Теоретичні частоти', align='center', color='red')
plt.xlabel('Значення')
plt.ylabel('Частота')
plt.title('Порівняння для неперервного розподілу')
plt.legend()
plt.show()

# Дискретні розподіли
plt.figure(figsize=(8, 5))
x = np.arange(len(data['frequency']))
width = 0.35  # Ширина стовпчиків

# Зміщуємо стовпчики вліво та вправо від центру x, щоб вони стояли поруч
plt.bar(x  -  width/2,  data['frequency'],  width,  alpha=0.6,  label='Емпіричні
частоти')
plt.bar(x + width/2, expected, width, alpha=0.6, label='Теоретичні частоти',
color='red')
plt.xlabel('Кількість подій (категорії)')
plt.ylabel('Частота')
plt.title('Порівняння для дискретного розподілу')
plt.xticks(x)  # Встановлюємо позначки на осі X згідно з категоріями
plt.legend()
plt.show()
```

Після виконання розрахунків та побудови графіка програма видає підсумкові статистичні дані та візуалізацію.



Методика верифікації розрахунків у MS Excel

Для підтвердження правильності програмної реалізації та глибшого розуміння алгоритму перевірки гіпотез необхідно виконати дублюючий розрахунок у MS Excel.

1. Розрахунок теоретичних імовірностей (p_i).

Вибір функції, яку вводимо в рядок формул, залежить від типу розподілу, вказаного в індивідуальному завданні:

— нормальний розподіл:

$= NORM.DIST(x_{right};\ mean;\ std;\ TRUE) - NORM.DIST(x_{left};\ mean;\ std;\ TRUE);$

— рівномірний розподіл: імовірність для кожного інтервалу розраховується як $p_i = \frac{\Delta x}{b-a}$, де Δx — ширина інтервалу, a та b — межі розподілу;

— показниковий розподіл:

$= EXPON.DIST(x\_right;\ lambda;\ TRUE) - EXPON.DIST(x\_left;\ lambda;\ TRUE);$

— пуассонівський розподіл:

$= POISSON.DIST(k; lambda; FALSE)$ – розраховується для кожного цілого значення k;

— біноміальний розподіл:

$= BINOM.DIST(k;\ n;\ p;\ FALSE)$ – розраховується для кожного цілого числа успіхів k.

2. Алгоритм обчислення статистики χ^2.

Для кожного інтервалу (або значення k) виконайте такі кроки:

— обчисліть теоретичні частоти (n_i'): $n_i' = n \cdot p_i$, де n – загальний обсяг вибірки (сума всіх емпіричних частот);

— обчисліть складові статистики: у новому стовпці розрахуйте значення за формулою «$= (n\_i - n\_i')\text{^}2 / n\_i'$»;

— знайдіть спостережуване значення (χ^2_{stat}): підсумуйте значення отриманого стовпця за допомогою функції «$= SUM()$»;

— визначте критичне значення: використовуйте функцію «$= CHISQ.INV.RT(alpha; df)$», де df — кількість ступенів свободи.

3. Прийняття рішення.

Порівняйте значення χ^2_{stat} із χ^2_{crit}. Якщо спостережуване значення менше за критичне, гіпотеза про вибраний закон розподілу не відхиляється. Отримані результати мають збігатися з виведенням програми на Python.

Порядок виконання роботи

Теоретична підготовка

1. Опрацювати фундаментальні поняття: статистична гіпотеза, нульова (H_0) та альтернативна (H_1) гіпотези, рівень значущості (α), критична область.
2. Вивчити алгоритм застосування критерію Пірсона (χ^2) та умови його використання (мінімальний обсяг вибірки, частоти груп $n_i \geq 5$).
3. Підготувати інструментарій: перевірити налаштування середовища Python та створити шаблон таблиці в MS Excel для розрахунку теоретичних частот.
4. Вибрати індивідуальний варіант завдання згідно зі списком.

Аналітичне розв'язання (MS Excel / Вручну)

На цьому етапі виконується розрахунок еталонних значень для розуміння природи статистичної перевірки.

1. Первинне оброблення та оцінювання: розрахувати вибіркові характеристики (середнє, дисперсію), необхідні для оцінювання параметрів гіпотетичного розподілу.
2. Розрахунок частот:

— сформувати інтервальний ряд та підрахувати емпіричні частоти n_i;

— обчислити теоретичні частоти n_i' для вибраного закону розподілу. За необхідності виконати об'єднання малочисельних інтервалів.

3. Статистична перевірка:

— обчислити значення статистики критерію χ^2 за формулою

$$\sum \frac{(n_i - n_i')^2}{n_i'};$$

— визначити кількість ступенів свободи df та знайти критичне значення $\chi^2_{\text{крит}}$ за таблицями.

Програмна реалізація та використання ШІ

Створити скрипт або блокнот та виконати такі дії:

1. Підготовка даних: створити масиви для вхідної вибірки та імпортувати бібліотеки ($numpy$, $scipy.stats$, $matplotlib$).

2. Залучення ШІ-асистента: використати ШІ для генерації коду обчислення p-значення за допомогою $scipy.stats.chisquare$ або для стилізації графіків.

3. Реалізація алгоритму:

— написати код для автоматичного розрахунку теоретичних частот та статистики χ^2;

— реалізувати побудову гістограми з накладеною теоретичною кривою щільності розподілу.

4. Виведення результатів: забезпечити виведення розрахованого χ^2, критичного значення та p −value у консоль.

Аналіз результатів та формування висновків

1. Верифікація розрахунків: порівняти значення статистики χ^2, отримані в Python, з ручними розрахунками в Excel.

2. Інтерпретація результатів: сформулювати рішення щодо прийняття або відхилення нульової гіпотези на основі порівняння $\chi^2_{\text{спост}}$ з $\chi^2_{\text{крит}}$ або аналізу p-value.

3. Рефлексія щодо використання ШІ: описати досвід взаємодії з ШІ: чи правильно він підказав функцію для розрахунку ступенів свободи та чи допоміг в інтерпретації p-значення.

Вимоги до звіту та його зміст

Звіт є підсумковим документом, що фіксує результати проведеного статистичного дослідження щодо перевірки гіпотез. Його структура має містити такі обов'язкові компоненти:

1. Титульна сторінка.

Оформлюється згідно з установленим зразком. Має містити назву дисципліни, номер роботи, дані здобувача освіти та викладача.

2. Тема, мета роботи та номер варіанта.

Формулюються чітко відповідно до методичних вказівок.

3. Постановка завдання.

Необхідно повністю скопіювати умову завдання зі свого індивідуального варіанта (вихідні дані вибірки, передбачуваний закон розподілу, рівень значущості α).

4. Результати виконання.

Це основний розділ звіту, що складається з трьох частин:

а) розрахунки (MS Excel / Вручну):

— таблиця розрахунку вибіркових характеристик (середнє $\bar{x}$, дисперсія s^2, групові частоти);

— таблиця розрахунку теоретичних частот для заданого розподілу;

— обчислення спостережуваного значення критерію χ^2_{stat};

— визначення критичної точки χ^2_{crit} для заданого рівня значущості α та кількості ступенів свободи df;

б) код програми: повний програмний код на Python з коментарями, що реалізує оцінку параметрів, обчислення теоретичних частот, статистику Пірсона та візуалізацію;

в) результати роботи програми:

— скріншот текстового виведення розрахованих показників (статистика χ^2, критичне значення, результат перевірки гіпотези);

— графік: порівняльна гістограма емпіричних та теоретичних частот (із використанням $align = 'edge'$ для неперервних даних).

5. Аналіз використання ШІ-асистента.

У цьому розділі здобувач освіти має продемонструвати навички критичної взаємодії з технологіями штучного інтелекту:

а) вибір програмного інструментарію ШІ. Вказати повну назву та версію використаної моделі (Gemini 2.0 Flash, ChatGPT 4o, Claude 3.5);

б) роль ШІ у виконанні роботи. Навести 2–3 конкретні приклади запитів (промптів) та проаналізувати отримані відповіді. Наприклад, як ШІ допоміг вибрати кількість параметрів m для розрахунку ступенів вільності, пояснив різницю між pmf та cdf у SciPy або допоміг виправити помилку в аргументах функції $plt.bar$.

6. Висновки.

Розділ має містити підсумок проведеного дослідження, поділений на два аспекти:

а) технічний висновок:

— провести аналіз результатів тестування. На основі порівняння χ^2_{stat} з χ^2_{crit} зробити аргументований висновок про прийняття або відхилення нульової гіпотези;

— пояснити практичне значення результату (чи підтверджено припущення про закон розподілу досліджуваної величини);

б) рефлексія щодо використання ШІ.

Критично оцінити ефективність допомоги асистента. Описати, чи сприяв ШІ глибшому розумінню логіки критерію згоди Пірсона, чи його роль обмежилася лише технічним генеруванням коду.

Завдання для самостійного виконання

Завдання 1. Мобільні ігри.

Компанія, що розробляє мобільні ігри, тестує новий сервер для оброблення запитів гравців. Для оптимізації продуктивності вони вимірювали час відповіді сервера (мс) на 100 запитів під час пікового навантаження. Потрібно перевірити, чи час відповіді відповідає нормальному розподілу, щоб правильно налаштувати серверну інфраструктуру.

Дані: [100-120]: 5, [120-140]: 10, [140-160]: 20, [160-180]: 30, [180-200]: 20, [200-220]: 10, [220-240]: 5.

За критерієм Пірсона (α=0.05) перевірте, чи час відповіді сервера має нормальний розподіл. Обчисліть вибіркове середнє, дисперсію, теоретичні частоти вручну або в Excel. Реалізуйте в Python, побудуйте гістограму для порівняння емпіричних і теоретичних частот.

Завдання 2. Доставка продуктів.

Логістична компанія, що доставляє продукти, прагне оптимізувати графік роботи складів. Вони проаналізували час очікування вантажівок на розвантаження (хв) протягом 50 операцій. Менеджери припускають, що час очікування розподілений рівномірно (розвантаження відбувається за чітким графіком). Це допоможе спланувати роботу складів без затримок.

Дані: [0-5]: 12, [5-10]: 10, [10-15]: 11, [15-20]: 9, [20-25]: 8.

За критерієм Пірсона (α=0.05) перевірте, чи час очікування розподілений рівномірно. Обчисліть параметри розподілу, теоретичні частоти вручну або в Excel. Реалізуйте в Python, побудуйте гістограму.

Завдання 3. Активність користувачів.

Популярна соціальна мережа аналізує активність користувачів, зокрема кількість скарг на некоректний контент за день. За 200 днів підрахували, скільки скарг надходило щодня. Команда модераторів хоче перевірити, чи кількість скарг відповідає пуассонівському розподілу, щоб передбачити пікові навантаження та оптимізувати роботу.

Дані: 0: 50, 1: 80, 2: 40, 3: 20, 4: 10.

За критерієм Пірсона (α=0.05) перевірте, чи кількість скарг має пуассонівський розподіл. Обчисліть параметр λ, теоретичні частоти вручну або в Excel. Реалізуйте в Python, побудуйте гістограму.

Контрольні запитання

1. Що таке нульова та альтернативна гіпотези в контексті перевірки закону розподілу?
2. У чому полягає суть критерію узгодженості Пірсона (χ^2)?
3. Як визначається кількість ступенів свободи при перевірці гіпотези про нормальний розподіл?
4. У якому випадку необхідно об'єднувати інтервали (категорії) при розрахунку χ^2 і для чого це робиться?
5. Що таке p-значення (p-value) і як його інтерпретувати?
6. Яка функція в бібліотеці scipy.stats дає змогу виконати перевірку гіпотези за критерієм Пірсона?
7. Чим відрізняється розрахунок теоретичних частот для рівномірного та показникового розподілів?
8. Як візуально за допомогою гістограми можна оцінити узгодженість емпіричних та теоретичних даних?
9. Наведіть приклад практичної задачі, де може знадобитися перевірка гіпотези про пуассонівський розподіл.
10. Які переваги надає використання ШІ для інтерпретації результатів статистичних тестів порівняно з простим аналізом коду?

Практична робота № 8
КОРЕЛЯЦІЙНИЙ АНАЛІЗ І РЕГРЕСІЙНІ МОДЕЛІ

Мета роботи:

— ознайомитися з методами аналізу статистичних залежностей між змінними;

— опанувати техніки побудови лінійних, нелінійних та множинних регресійних моделей за допомогою методу найменших квадратів;

— навчитися обчислювати коефіцієнти кореляції та детермінації вручну, в MS Excel та Python;

— освоїти побудову графіків кореляційного поля та регресійних кривих для оцінки й інтерпретації зв’язків у даних.

Компетентності, що формуються:

— здатність оцінювати силу та напрямок зв'язку між змінними за допомогою коефіцієнтів кореляції;

— уміння вибирати оптимальну математичну модель (лінійну, нелінійну або множинну) для адекватного опису залежностей у даних;

— навички програмної реалізації методу найменших квадратів та обчислення показників якості моделей мовою Python;

— ефективне використання ШІ-асистентів для генерації коду регресійних моделей, їх перевірки та змістовної інтерпретації результатів.

Необхідне програмне забезпечення та бібліотеки:

1. Середовище розроблення: Google Colab, Jupyter Notebook або будь-яке інше інтегроване середовище розроблення (IDE) для Python (наприклад, VS Code, PyCharm).
2. Python версії 3.8 або новішої з бібліотеками: numpy, scipy, matplotlib, pandas, scikit-learn.
3. MS Excel із встановленою надбудовою Data Analysis ToolPak.

Теоретичні відомості

Кореляційний та регресійний аналіз є фундаментальними методами статистики, за допомогою яких можна виявляти та моделювати зв’язки між змінними, оцінювати їх силу й прогнозувати майбутні значення. Ці методи знаходять широке застосування в економіці, технологіях, медицині, маркетингу та інших галузях для аналізу даних, оптимізації процесів і прийняття обґрунтованих рішень. Кореляційний аналіз допомагає визначити, наскільки тісно пов’язані змінні, тоді як регресійний аналіз створює математичні моделі для опису цих залежностей.

Для виконання таких аналізів широко використовуються програмні середовища MS Excel і Python. Excel пропонує зручний інтерфейс і вбудовані інструменти для швидкого аналізу даних, тоді як Python забезпечує гнучкість, точність і потужні бібліотеки для оброблення великих обсягів даних і побудови складних моделей.

Кореляційний аналіз

Кореляційний аналіз – це статистичний метод, що використовується для оцінювання сили та напрямку зв’язку між двома або більше змінними. Його мета – визначити, чи існує залежність між змінними, і якщо так, то наскільки вона сильна. Кореляція не вказує на причинно-наслідковий зв’язок, а лише на статистичну асоціацію.

Коефіцієнт кореляції Пірсона

Коефіцієнт кореляції Пірсона (r) вимірює силу та напрямок *лінійного* зв’язку між двома змінними (x) і (y).

Формула

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}},$$

де $(\bar{x}, \bar{y})$ – середні значення змінних;

(x_i, y_i) – індивідуальні значення змінних.

Інтерпретація:

- $(r = 1)$: ідеальний прямий лінійний зв’язок;
- $(r = -1)$: ідеальний обернений лінійний зв’язок;
- $(r \approx 0)$: відсутність лінійного зв’язку;
- $(|r| \geq 0.7)$: сильний зв’язок;
- $(0.3 \leq |r| < 0.7)$: помірний зв’язок;
- $(|r| < 0.3)$: слабкий зв’язок.

Кореляційне поле

Кореляційне поле – це графік розсіювання точок (x_i, y_i), який дає змогу візуально оцінити форму, силу та напрямок зв’язку між змінними. На основі форми можна вибрати тип регресії (лінійна, параболічна тощо).

Приклади візуалізації різних типів зв'язку наведено на рис. 8.1. Аналіз таких полів дає змогу аналітику ще до початку розрахунків зрозуміти, чи є зв'язок лінійним, зворотним або ж він зовсім відсутній.

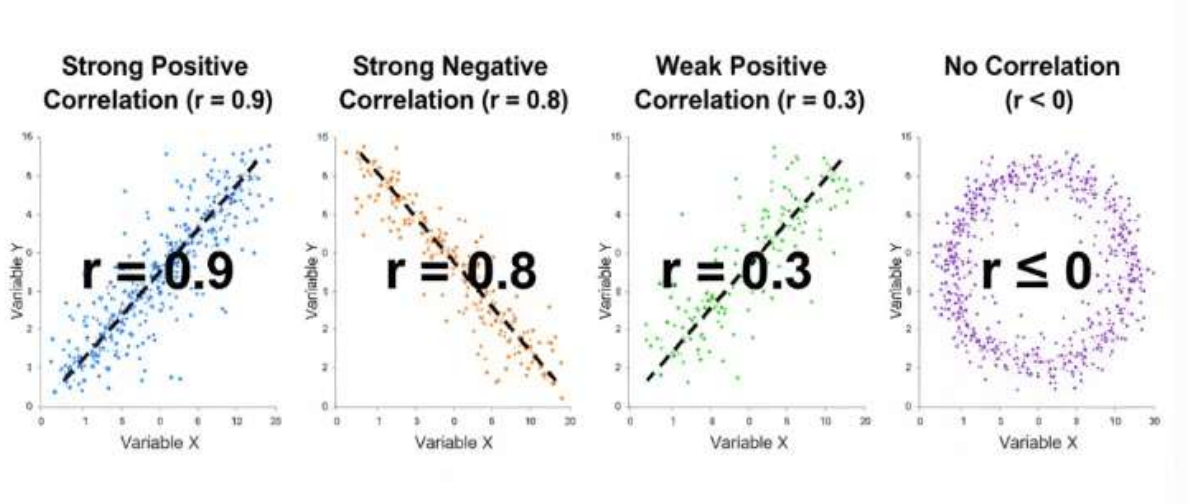


Рис. 8.1. Типові види кореляційних полів залежно від сили та напрямку зв’язку

Регресійний аналіз

Регресійний аналіз – це розділ математичної статистики, присвячений побудові та аналізу моделей залежності між результативною ознакою (залежною змінною y) та одним або групою факторних показників (незалежних змінних x).

На відміну від кореляційного аналізу, який лише оцінює силу асоціації, регресійний аналіз дає змогу отримати аналітичний вираз (формулу) зв'язку, за допомогою якого можна прогнозувати значення результативної ознаки за заданих значень факторів.

Завданням регресійного аналізу є:

— вибір форми зв'язку (визначення математичної функції, яка найкраще апроксимує емпіричні дані);

— оцінювання параметрів моделі (обчислення коефіцієнтів рівняння, зазвичай за допомогою методу найменших квадратів);

— статистичне оцінювання моделі (тобто перевірка адекватності моделі реальним даним та оцінювання значущості отриманих результатів).

Залежно від кількості включених факторів та характеру математичної залежності розрізняють такі види регресії:

1. Парна регресія (дослідження впливу одного фактора x):

— лінійна: $y = ax + b + \varepsilon$. Описує рівномірну зміну результату;

— нелінійна (параболічна): $y = ax^2 + bx + c$, яка використовується, коли залежність має точку екстремуму або змінює темп зростання;

— інші види: гіперболічна ($y = a/x + b$), показникові тощо.

2. Множинна регресія (дослідження впливу декількох факторів $x_1, x_2, \dots, x_n$):

— множинна лінійна: $y = b + a_1x_1 + a_2x_2 + \cdots + a_nx_n$. Дає змогу оцінити ізольований вплив кожного фактора на результат;

— множинна нелінійна: *поліноміальна (*враховує нелінійний внесок кожного фактора, наприклад, квадратичні члени x_i^2), *мультиплікативна (степенева):* $y = b \cdot x_1^{a_1} \cdot x_2^{a_2} \dots x_n^{a_n}$. Описує складну взаємодію факторів (ефект масштабу).

Інтерпретація параметрів моделі

Економіко-статистичне зростання параметрів визначається вибраною математичною формою рівняння:

— коефіцієнти регресії (a_i): характеризують чутливість результату y до зміни фактора x_i. У лінійних моделях вони показують абсолютну зміну y при зміні x на одиницю (мінімально прийняту цілу міру), а в нелінійних – визначають характер динаміки (прискорення, еластичність) та напрямок впливу;

— вільний член (b або a_0): відображає узагальнений вплив факторів, не включених до моделі. Геометрично це значення y у точці $x = 0$.

Метод найменших квадратів (МНК)

Метод найменших квадратів є класичним математичним апаратом для оцінювання параметрів регресії. Його метою є знаходження таких значень коефіцієнтів, при яких сума квадратів відхилень фактичних значень (y_i) від теоретичних ($\hat{y}_i$), розрахованих за моделлю, буде мінімальною:

$$\boldsymbol{S = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \rightarrow min.}$$

Для лінійної парної регресії ($y = ax + b$) параметри обчислюються на основі системи нормальних рівнянь, що має такий вигляд:

— коефіцієнт нахилу (a): $a = \frac{n\sum xy \ \ \sum x \sum y}{n\sum x^2 - (\sum x)^2}$;

— вільний член (b): $b = \bar{y} - a\bar{x}$.

Коефіцієнт детермінації (R^2)

Для оцінювання адекватності побудованої моделі та її прогнозної сили використовують коефіцієнт детермінації (R^2). Він визначає частку варіації залежної змінної, яка пояснюється впливом включених у модель факторів.

Математично він розраховується як відношення поясненої дисперсії до загальної:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}.$$

Інтерпретація значень:

— $R^2 = 1$: модель ідеально описує дані (всі точки лежать на лінії регресії);

— $R^2 = 0$: модель не пояснює варіацію результату (зв'язок між змінними відсутній);

— $0 < R^2 < 1$: значення показує відсоток варіації y, зумовлений впливом фактора x. Наприклад, $R^2 = 0.85$ означає, що 85 % змін результату пояснюються моделлю, а 15 % – іншими факторами.

Особливості множинної регресії

У випадку множинної регресії звичайний R^2 завжди збільшується при додаванні нових змінних, навіть якщо вони не є суттєвими. Тому використовується скоригований коефіцієнт детермінації ($\overline{R^2}$ або R^2_{adj}), який враховує кількість ступенів вільної варіації та накладає «штраф» за надлишковість моделі.

Формула розрахунку

$$\overline{R^2} = 1 - (1 - R^2)\frac{n-1}{n-k-1},$$

де n – обсяг вибірки (кількість спостережень);

k – кількість незалежних змінних (факторів) у моделі.

Застосування Python для виконання завдань

Python є універсальним інструментом для виконання кореляційного та регресійного аналізу завдяки своїм бібліотекам, які забезпечують високу точність, гнучкість і можливість автоматизації обчислень. Використання Python дає змогу обробляти великі обсяги даних, будувати складні моделі та створювати якісні візуалізації.

Для реалізації кореляційно-регресійного аналізу в Python використовується стандартний стек бібліотек, кожна з яких відповідає за свій етап оброблення даних:

— $numpy$: використовується для числових обчислень, зокрема для побудови лінійної та поліноміальної регресії через функцію $polyfit$, за допомогою якої можна швидко знаходити параметри моделей;

— $scipy$: містить інструменти для нелінійної регресії ($optimize.curve_fit$) та обчислення коефіцієнта кореляції Пірсона ($stats.pearsonr$), включаючи p-значення для перевірки значущості;

— $pandas$: призначена для зручного оброблення даних, їх організації у форматі таблиць ($DataFrames$) і підготовки до аналізу;

— $matplotlib$: використовується для створення графіків, таких як кореляційне поле та регресійні криві, з можливістю детального форматування;

— $scikit-learn$: надає інструменти для множинної регресії ($LinearRegression$) та оцінювання якості моделей через коефіцієнт детермінації (R^2).

Кореляційний аналіз

Кореляційний аналіз у Python виконується за допомогою функції $pearsonr$ з бібліотеки $scipy.stats$. Вона повертає коефіцієнт кореляції Пірсона (r) та p-значення, яке дає змогу оцінити статистичну значущість зв'язку (якщо p-значення менше 0.05, зв'язок вважається статистично значущим). За допомогою цього методу можна швидко оцінити силу лінійного зв'язку між змінними.

Наприклад, розглянемо дані про кількість годин навчання (x) та оцінки здобувачів освіти (y):

```
from scipy.stats import pearsonr
x = [10, 15, 20, 25]  # Години навчання
y = [60, 75, 85, 90]   # Оцінки
r, p_value = pearsonr(x, y)
print(f"Коефіцієнт кореляції: {r:.3f}, p-значення: {p_value:.3f}")
```

Результат:

```
••• Коефіцієнт кореляції: 0.976, p-значення: 0.024
```

Лінійна регресія

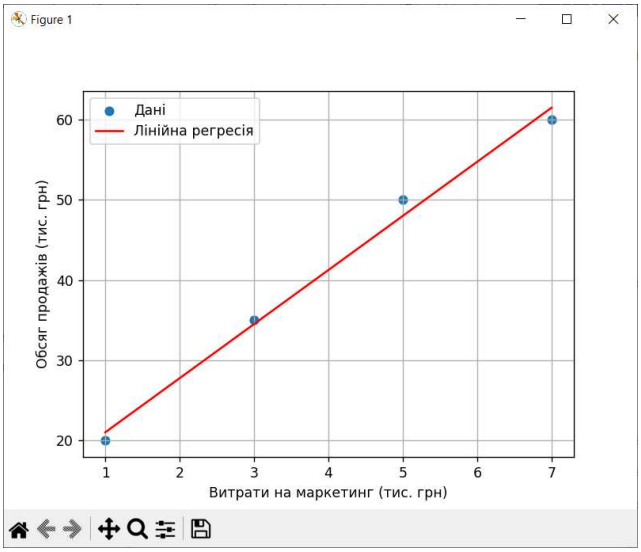
Для побудови лінійної регресії використовується функція *numpy.polyfit* із параметром ступеня 1. Вона обчислює коефіцієнти регресії ($\hat{a}, \hat{b}$) за

методом найменших квадратів. Після цього можна візуалізувати дані та регресійну пряму за допомогою matplotlib.

Наприклад, для даних про витрати на маркетинг (x) та обсяг продажів (y):

```
import numpy as np
import matplotlib.pyplot as plt
x = np.array([1, 3, 5, 7])        # Витрати на маркетинг (тис. грн)
y = np.array([20, 35, 50, 60])    # Обсяг продажів (тис. грн)
coeffs = np.polyfit(x, y, 1)       # Ступінь 1 для лінійної регресії
b, a = coeffs                      # b — нахил, a — вільний член
plt.scatter(x, y, label="Дані")
plt.plot(x, b*x + a, "r-", label="Лінійна регресія")
plt.xlabel("Витрати на маркетинг (тис. грн)")
plt.ylabel("Обсяг продажів (тис. грн)")
plt.legend()
plt.grid(True)
plt.show()
```

Цей код створює кореляційне поле з точками даних і додає червону лінію регресії, що дає змогу оцінити відповідність моделі даним:

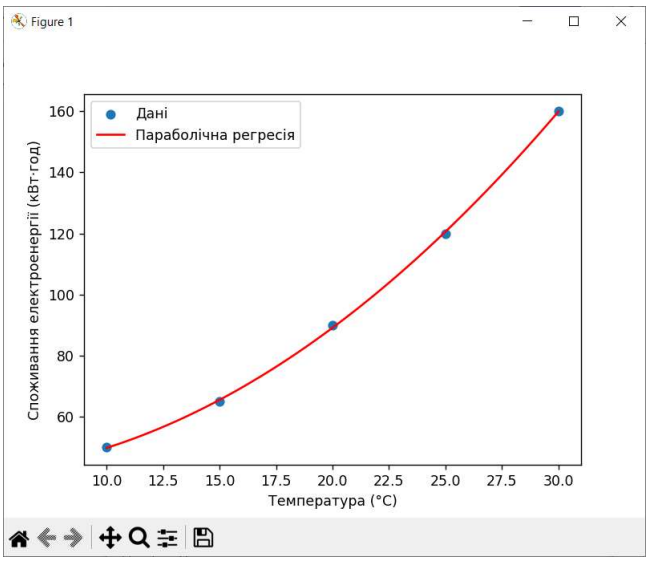


Нелінійна регресія

Нелінійна регресія застосовується, коли зв'язок між змінними не є лінійним. Для параболічної регресії ($\hat{y} = \hat{a} + \hat{b}x + \hat{c}x^2$) використовується *numpy.polyfit* зі ступенем 2.

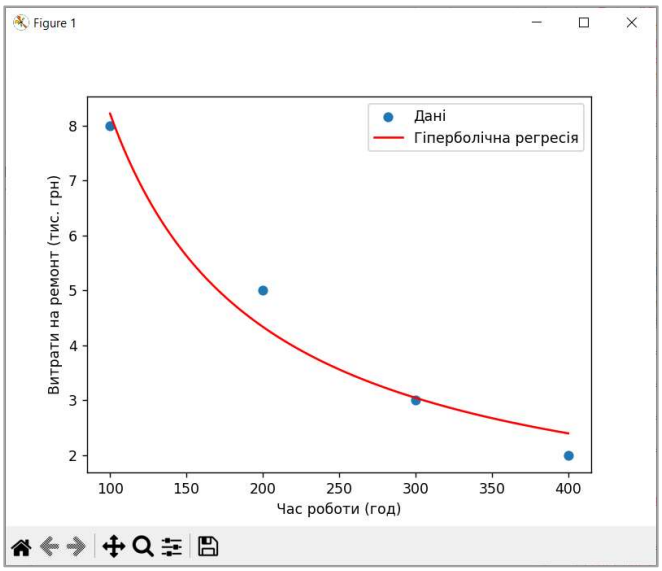
Наприклад, для даних про температуру (x) та споживання електроенергії (y):

```
x = np.array([10, 15, 20, 25, 30])  # Температура (°C)
y = np.array([50, 65, 90, 120, 160]) # Споживання електроенергії (кВт·год)
coeffs = np.polyfit(x, y, 2)         # Ступінь 2 для параболічної регресії
c, b, a = coeffs                     # a + bx + cx^2
x_fit = np.linspace(min(x), max(x), 100)
y_fit = a + b*x_fit + c*x_fit**2
plt.scatter(x, y, label="Дані")
plt.plot(x_fit, y_fit, "r-", label="Параболічна регресія")
plt.xlabel("Температура (°C)")
plt.ylabel("Споживання електроенергії (кВт·год)")
plt.legend()
plt.show()
```



Для складніших моделей, таких як гіперболічна ($\hat{y} = \hat{a} + \frac{\hat{b}}{x}$), використовується *scipy.optimize.curve_fit*. Наприклад, для даних про час роботи обладнання (x) та витрати на ремонт (y):

```
from scipy.optimize import curve_fit
x = np.array([100, 200, 300, 400])  # Час роботи (год)
y = np.array([8, 5, 3, 2])          # Витрати на ремонт (тис. грн)
def hyperbola(x, a, b):
    return a + b/x
params, _ = curve_fit(hyperbola, x, y)
a, b = params
x_fit = np.linspace(min(x), max(x), 100)
y_fit = hyperbola(x_fit, a, b)
plt.scatter(x, y, label="Дані")
plt.plot(x_fit, y_fit, "r-", label="Гіперболічна регресія")
plt.xlabel("Час роботи (год)")
plt.ylabel("Витрати на ремонт (тис. грн)")
plt.legend()
plt.show()
```



Цей підхід забезпечує гнучкість у моделюванні нелінійних залежностей.

Множинна регресія

Множинна регресія, яка враховує кілька незалежних змінних, реалізується за допомогою *sklearn.linear_model.LinearRegression*. Наприклад, для даних про продажі (y), залежні від витрат на рекламу (x_1) та кількості відвідувачів сайту (x_2):

```
from sklearn.linear_model import LinearRegression
import numpy as np
X = np.array([[0.5, 100], [1.0, 150], [1.5, 200], [2.0, 250]])  # [Витрати на рекламу (тис. грн), Відвідувачі (осіб)]
y = np.array([20, 30, 45, 60])                # Продажі (тис. грн)
model = LinearRegression()
model.fit(X, y)
print(f"Коефіцієнти: {model.coef_}, Вільний член: {model.intercept_}")
print(f"R^2: {model.score(X, y):.3f}")
```

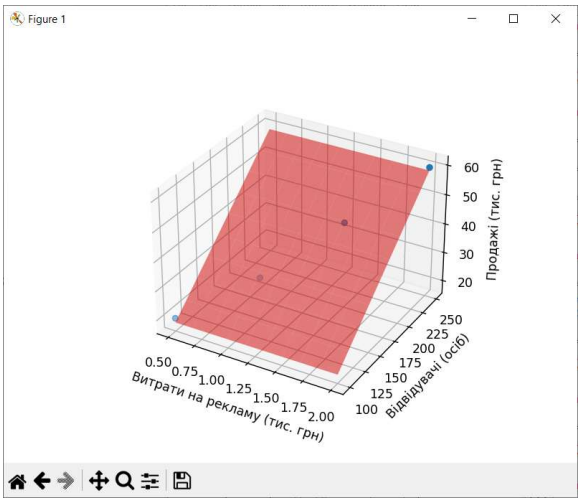
Результат:

```
Коефіцієнти: [0.00269973 0.269973  ], Вільний член: -8.498650134986484
R^2: 0.992
```

Для візуалізації множинної регресії використовується 3D-графік, що показує площину регресії:

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
ax.scatter(X[:, 0], X[:, 1], y)
x1, x2 = np.meshgrid(np.linspace(min(X[:, 0]), max(X[:, 0]), 10),
                     np.linspace(min(X[:, 1]), max(X[:, 1]), 10))
y_pred = model.intercept_ + model.coef_[0]*x1 + model.coef_[1]*x2
ax.plot_surface(x1, x2, y_pred, color="r", alpha=0.5)
ax.set_xlabel("Витрати на рекламу (тис. грн)")
ax.set_ylabel("Відвідувачі (осіб)")
ax.set_zlabel("Продажі (тис. грн)")
plt.show()
```

Цей код створює тривимірне зображення, яке допомагає оцінити, як дві незалежні змінні впливають на залежну.



Обчислення (R^2)

Коефіцієнт детермінації (R^2) оцінює якість регресійної моделі. Для лінійної та нелінійної регресії (R^2) обчислюється вручну:

```
y_pred = b*x + a  # Для лінійної регресії
ss_tot = np.sum((y - np.mean(y))**2)
ss_res = np.sum((y - y_pred)**2)
r_squared = 1 - ss_res/ss_tot
print(f"R^2: {r_squared:.3f}")
```

Для множинної регресії використовується метод *score* об'єкта *LinearRegression*, який автоматично повертає (R^2). Високе значення (R^2) (близьке до 1) свідчить про хорошу відповідність моделі даним.

Форматування графіків

Для створення інформативних графіків важливо додавати підписи осей, заголовки, легенди та сітку. Наприклад, у коді для лінійної регресії використовуються *plt.xlabel*, *plt.ylabel*, *plt.legend* і *plt.grid(True)*. Різні кольори для точок даних і регресійної кривої (наприклад, синій для точок і червоний для лінії) полегшують сприйняття. Щоб зберегти графік для звіту, використовується така команда:

```
plt.savefig("plot.png").
```

Це дає змогу включити зображення у звіт у високій якості.

Використання MS Excel

MS Excel є потужним інструментом для кореляційного та регресійного аналізу завдяки вбудованим функціям і надбудові «Аналіз даних». Основні етапи:

Кореляційний аналіз

Функція $CORREL$ використовується для обчислення коефіцієнта кореляції Пірсона між двома наборами даних. Він показує, наскільки сильно пов'язані ці змінні лінійною залежністю.

Використовуючи інструмент $Correlation$ надбудови *Data Analysis*, можна отримати кореляційну матрицю, яка показує, наскільки сильно взаємопов'язані змінні.

У таблиці кореляції діагональні елементи дорівнюють 1, бо змінна ідеально корелює сама з собою. Позадіагональні елементи – це коефіцієнти кореляції Пірсона між парами змінних, що показують силу та напрямок зв'язку.

Регресійний аналіз

1. Лінійна регресія.

У надбудові $Data\ Analysis$ вибрати *Regression,* вказати діапазони для (y) (залежна змінна) та (x) (незалежна змінна).

Результати, які можна отримати: коефіцієнти регресії ($\hat{a}, \hat{b}$), (R^2), стандартні похибки, p-значення для перевірки значущості (α).

2. Нелінійна регресія.

Вибрати діаграму розсіювання $(Scatter\ Plot)$ з додаванням лінії тренду: створити діаграму розсіювання $(Scatter\ Plot)$ для даних, у контекстному

меню вибрати пункт $Add\ Trendline$, вказати тип тренду $(Trend\ Type)$, увімкнути опцію $Display\ Equation\ on\ Chart$ та $Display\ R^2\ Value\ on\ Chart$.

Для складних нелінійних моделей використовують $Solver$ для мінімізації суми квадратів відхилень.

Побудова графіків

1. Створіть діаграму розсіювання *(Scatter Plot)* для пар (x_i, y_i) – кореляційне поле.

2. Додайте лінію тренду до діаграми розсіювання, вибравши відповідний тип регресії – регресійна крива.

3. Додайте підписи осей, заголовок, легенду – остаточне форматування.

Перевірка значущості

Для перевірки значущості коефіцієнта кореляції $(\alpha = 0.05)$ використовуйте t-критерій:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}.$$

Порівняйте (t) з критичним значенням із таблиці t-розподілу для $(n-2)$ ступенів свободи.

Порядок виконання роботи

Теоретична підготовка

1. Опрацювати фундаментальні поняття: кореляційна залежність, метод найменших квадратів (МНК), коефіцієнт кореляції Пірсона (r), коефіцієнт детермінації (R^2).
2. Вивчити види регресійних моделей: лінійна, параболічна $(y = ax^2 + bx + c)$, гіперболічна $(y = a/x + b)$ та множинна.
3. Підготувати інструментарій: перевірити наявність бібліотек scikit-learn, scipy та matplotlib у середовищі Python; активувати надбудову Analysis ToolPak у MS Excel.
4. Обрати індивідуальний варіант завдання згідно зі списком.

Аналітичне розв'язання (MS Excel / Вручну)

На цьому етапі виконується розрахунок еталонних значень для розуміння математичної логіки побудови моделей.

1. Первинний аналіз даних: побудувати діаграму розсіювання $(Scatter\ Plot)$ для візуального визначення форми зв'язку (лінійна чи нелінійна).
2. Параметричне оцінювання:

— обчислити параметри вибраної регресійної моделі (a, b, c) за допомогою функцій $SLOPE$, $INTERCEPT$ або інструмента «Regression»;

— розрахувати коефіцієнт кореляції (r) та детермінації (R^2) для оцінювання тісноти зв'язку та якості моделі.

3. Побудова лінії тренду: додати на графік лінію тренду, вивести її рівняння та значення R^2 безпосередньо на діаграму.

Програмна реалізація та використання ШІ

Створити скрипт або блокнот та виконати такі дії:

1. Підготовка даних: сформувати масиви (або $DataFrame$) для незалежних (X) та залежних (Y) змінних.

2. Залучення ШІ-асистента: використати ШІ для генерації коду побудови нелінійних моделей (наприклад, перетворення даних для гіперболічної регресії) або для налаштування дизайну графіків.

3. Реалізація алгоритму:

— написати код для автоматичного обчислення параметрів моделі за допомогою LinearRegression з sklearn або polyfit з numpy;

— реалізувати візуалізацію: кореляційне поле з накладеною лінією регресії (теоретичною кривою).

4. Виведення результатів: забезпечити виведення рівняння регресії, значень r та R^2 у консоль.

Аналіз результатів та формування висновків

1. Верифікація розрахунків.

Порівняти значення коефіцієнтів та R^2, отримані в Python, з результатами в MS Excel. Проаналізувати причини можливих розбіжностей.

2. Інтерпретація результатів.

Оцінити статистичну значущість зв’язку. Сформулювати висновок про те, наскільки побудована модель адекватно описує досліджуваний процес.

3. Рефлексія щодо використання ШІ.

Описати досвід взаємодії з ШІ: чи допоміг він коректно підготувати дані (reshape) для scikit-learn та чи сприяв глибшому розумінню суті коефіцієнта детермінації.

Вимоги до звіту та його вміст

Звіт є підсумковим документом, що фіксує результати побудови та аналізу статистичних моделей. Його структура має містити такі обов'язкові компоненти:

1. Титульна сторінка.

Оформлюється згідно з установленим зразком. Має містити назву дисципліни, номер роботи, дані здобувача освіти та викладача.

2. Тема, мета роботи та номер варіанта.

Формулюються чітко відповідно до методичних вказівок щодо вивчення взаємозв'язків між величинами.

3. Постановка завдання.

Необхідно повністю скопіювати умову завдання зі свого індивідуального варіанта (таблиці значень факторів x та результатів y).

4. Результати виконання.

Основний розділ звіту, що складається з трьох частин:

а) розрахунки (MS Excel / Вручну):

— таблиця проміжних обчислень для МНК ($\sum x, \sum y, \sum x^2, \sum xy, \ldots$);

— розрахунок параметрів лінійної регресії (a, b) за формулами;

— розрахунок коефіцієнта кореляції Пірсона (r) та коефіцієнта детермінації (R^2);

б) код програми: повний програмний код на Python з коментарями, що реалізує обчислення параметрів для лінійної, нелінійної та множинної регресій, а також візуалізацію результатів;

в) результати роботи програми:

— скріншот текстового виведення: розраховані коефіцієнти, p-value та значення R^2;

— графік 1: кореляційне поле ($scatter\ plot$) з накладеною лінією лінійної регресії;

— графік 2: порівняння фактичних даних із кривою нелінійної (параболічної або гіперболічної) регресії;

— графік 3 (для множинної регресії): 3D-візуалізація точок даних та площини регресії.

5. Аналіз використання ШІ-асистента.

У цьому розділі здобувач освіти має продемонструвати навички критичної взаємодії з технологіями штучного інтелекту:

а) вибір програмного інструментарію ШІ. Вказати повну назву та версію використаної моделі (наприклад, Gemini 1.5 Flash, ChatGPT 4.0, Claude 3.5);

б) роль ШІ у виконанні роботи.

Навести 2–3 конкретні приклади запитів (промптів) та проаналізувати отримані відповіді. Наприклад, як ШІ допоміг інтерпретувати від'ємний коефіцієнт регресії, пояснив різницю між R^2 та скоригованим R^2 або допоміг налаштувати 3D-візуалізацію в matplotlib.

6. Висновки.

Розділ має містити підсумок проведеного дослідження, розділений на два аспекти:

а) технічний висновок.

Для кожного завдання описати характер зв'язку (тісний/слабкий, прямий/зворотний). На основі показника R^2 зробити висновок про адекватність побудованої моделі. Визначити, яка з моделей (лінійна чи нелінійна) краще описує наявні дані.

б) рефлексія щодо використання ШІ.

Критично оцінити ефективність допомоги асистента. Описати, чи сприяв ШІ розумінню суті регресійного аналізу, чи його роль обмежилася лише технічною допомогою у написанні коду для складних графіків.

Завдання для самостійного виконання

Ви – аналітик даних у компанії, що працює в інноваційній галузі. Ваше завдання – дослідити зв'язки між ключовими показниками (інвестиції, продажі, виробництво) за допомогою кореляційного та регресійного аналізу, щоб оцінити ефективність і запропонувати стратегії розвитку.

Ви працюєте в «StarLink Innovations», компанії, що розробляє малі супутники (CubeSats) для зв'язку та моніторингу Землі. Ваш аналіз допоможе оптимізувати інвестиції та продажі.

Завдання 1. Лінійна регресія (економіка космічних технологій).

Компанія інвестує в обладнання для виробництва супутників (3D-принтери, роботизовані лінії, тестові стенди). Потрібно з'ясувати, як ці інвестиції (млн грн) впливають на прибуток від продажів даних (млн грн), які використовуються для моніторингу посівів, прогнозування погоди та навігації. Ваші висновки покажуть, чи варто збільшувати бюджет на обладнання.

Дані:

x: інвестиції (млн грн): [2, 4, 6, 8]

y: прибуток (млн грн): [0.81, 2.3, 3.4, 4.54]

Необхідно:

— побудувати лінійну регресію $\hat{y} = \hat{a} + \hat{b}x$ вручну або в Excel;

— обчислити r, R^2, перевірити значущість ($\alpha = 0.05$);

— у Python (numpy.polyfit, matplotlib) побудувати кореляційне поле та регресійну криву;

— зробити висновки: чи сильний зв'язок? Чи можна прогнозувати прибуток? Дати рекомендації.

Приклад висновку: сильний зв'язок ($r \approx 0.98$) показує, що інвестиції підвищують прибуток. Модель підходить для прогнозування, але врахуйте конкуренцію.

Завдання 2. Параболічна регресія (виробництво супутників)

Виробництво компонентів для супутників (антени, сонячні панелі, плати) залежить від інвестицій у фонди (млн грн). Зв'язок може бути нелінійним через обмеження складів і логістики. Потрібно визначити оптимальний рівень інвестицій для максимізації виробництва.

Дані:

x: інвестиції (млн грн): [0, 1, 2, 3, 4, 5, 6, 7, 8]

y: обсяг виробництва (млн грн): [0.1, 0.48, 0.81, 1.26, 2.3, 2.85, 3.4, 3.96, 4.54]

Необхідно:

— побудувати параболічну регресію $\hat{y} = \hat{a} + \hat{b}x + \hat{c}x^2$ у Python (numpy.polyfit, deg=2) або Excel;

— обчислити r, R^2;

— у Python побудувати кореляційне поле та криву (matplotlib);

— зробити висновки: чи нелінійний зв’язок? Який оптимальний рівень інвестицій? Дати рекомендації.

Приклад висновку: параболічна модель ($R^2 \approx 0.99$) показує сповільнення після 6–7 млн грн. Інвестуйте в логістику для зростання.

Завдання 3. Множинна регресія (торгівля супутниковими даними)

Компанія продає дані через виставкові центри, де клієнти (аграрні фірми, урядові організації) тестують продукти. Товарообіг (млн у. о.) залежить від торгової площі (тис.м^2) і потоку клієнтів (тис. чол./день). Ваш аналіз допоможе спланувати нові центри.

Дані:

Товарообіг (y, млн у.о.)	Торгова площа (x_1, тис. м^2)	Потік клієнтів (x_2, тис. чол./день)
2.93	0.31	10.24
5.27	0.98	7.15
6.85	1.21	10.81
7.01	1.29	9.89
7.02	1.12	13.72

Необхідно:

— побудувати множинну регресію ($\hat{y} = \widehat{a_0} + \widehat{a_1}x_1 + \widehat{a_2}x_2$) у Python (sklearn);

— обчислити R^2, оцінити якість;

— побудувати 3D-графік (matplotlib, Axes3D);

— зробити висновки: який фактор важливіший? Чи можна прогнозувати? Дати рекомендації.

Приклад висновку: потік клієнтів має більший вплив. Модель ($R^2 \approx 0.95$) підходить для прогнозування. Інвестуйте в маркетинг.

Контрольні запитання

1. Що таке кореляційний аналіз і яка його мета?
2. Як інтерпретується коефіцієнт кореляції Пірсона, якщо він дорівнює -0.9, 0.1 та 0?
3. У чому полягає суть методу найменших квадратів?
4. Що показує коефіцієнт детермінації $R^2 = 0.85$?
5. Яка функція в бібліотеці numpy використовується для побудови поліноміальної регресії?
6. У якому випадку доцільно використовувати параболічну регресію замість лінійної?
7. Чим множинна регресія відрізняється від простої лінійної регресії?
8. Як у matplotlib побудувати діаграму розсіювання (кореляційне поле)?
9. Для чого використовується p-значення, яке повертає функція scipy.stats.pearsonr?

10. Які переваги та недоліки використання ШІ для вирішення завдань регресійного аналізу?

11. Що таке вільний член (b) у рівнянні лінійної регресії $y = ax + b$ і як його інтерпретувати в економічних задачах?

12. Чим відрізняється звичайний коефіцієнт детермінації (R^2) від скоригованого ($\overline{R^2}$) і коли слід звертати увагу саме на другий?

13. Які умови (постулати) мають виконуватися для залишків регресії, щоб модель вважалася якісною?

14. Яку роль відіграє функція np.linspace() при візуалізації нелінійної регресії в Python?

15. Що таке мультиколінеарність у множинній регресії та чому вона є небажаною при побудові моделі?

БІБЛІОГРАФІЧНИЙ СПИСОК

Васильків, І. М. Основи теорії ймовірностей і математичної статистики [Електронний ресурс] : навч. посіб. / І. М. Васильків. – Львів : ЛНУ імені Івана Франка, 2020. – 184 с. – Режим доступу: https://new.mmf.lnu.edu.ua/wp-content/uploads/2020/04/Vasyl-kiv-I.M.-IMS_CHASTYNA_1.pdf

Веригіна, І. В. Теорія ймовірностей та математична статистика: Ч. 1. Випадкові події: Лекції і практикум [Електронний ресурс] : навч. посіб. для студ. спец. 143 «Атомна енергетика», спеціалізації «Атомні електричні станції» / І. В. Веригіна, О. В. Островська. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 57 с. – Режим доступу: https://ela.kpi.ua/bitstream/123456789/23501/1/NP(T_Ym)_1.pdf

Єрьоменко, В. О. Практикум з теорії ймовірностей та математичної статистики [Текст] / В. О. Єрьоменко, М. І. Шинкарик, Р. М. Бабій, А. І. Процик. – Тернопіль : Екон. думка, 2015. – 317 с.

Жлуктенко, В. І. Теорія ймовірностей і математична статистика [Текст] : навч.- метод. посіб. У 2 ч. Ч. 1. Теорія ймовірностей / В. І. Жлуктенко, С. І. Наконечний. – Київ : КНЕУ, 2015. – 304 с.

Жлуктенко, В. І. Теорія ймовірностей і математична статистика [Текст] : навч.- метод. посіб. У 2 ч. Ч. 2. Теорія ймовірностей / В. І. Жлуктенко, С. І. Наконечний. – Київ : КНЕУ, 2016. – 336 с.

Кармелюк, Г. І. Теорія ймовірностей та математична статистика. Посібник для розв’язування задач [Текст] : навч. посіб. / Г. І. Кармелюк. – Київ : Центр учбової літератури, 2019. – 567 с.

Кравченко, І. В. Інформаційні технології: Системи комп’ютерної математики [Електронний ресурс] : навч. посіб. для студ. спец. «Автоматизація та комп’ютерно-інтегровані технології» / І. В. Кравченко, В. І. Микитенко. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 243 с. – Режим доступу: https://ooep.kpi.ua/downloads/disc/inf_t/posibn_Krav_Myk.pdf

Найко, Д. А. Теорія ймовірностей та математична статистика [Електронний ресурс] : навч. посіб. / Д. А. Найко, О. Ф. Шевчук. – Вінниця : ВНАУ, 2020. – 382 с. – Режим доступу: http://repository.vsau.org/getfile.php/24513.pdf

Огірко, О. І. Теорія ймовірностей та математична статистика [Електронний ресурс] : навч. посіб. / О. І. Огірко, Н. В. Галайко. – Львів : ЛьвДУВС, 2017. – 292 с. – Режим доступу: https://dspace.lvduvs.edu.ua/bitstream/1234567890/629/1/теорія ймовірностей підручник.pdf

Практикум з теорії імовірностей та математичної статистики : навч. посіб. для студ. екон. спец. [Електронний ресурс] / А. М. Алілуйко, Н. В. Дзюбановська, В. О. Єрьоменко, О. М. Мартинюк, М. І. Шинкарик. – Тернопіль : Підручники і посібники, 2018. — 352 с.

– Режим доступу: http://dspace.wunu.edu.ua/bitstream/316497/32236/1/Практикум_Теорія ймовірностей_2018.pdf

Теорія ймовірностей та математична статистика (конспект лекцій + тести) [Електронний ресурс]: навч. посіб. / Я. Т.Соловко, П. Г. Остафійчук, О. З. Гарпуль, С. А. Войтик. – Вид. 2-ге, доп. – Івано-Франківськ : Репозитарій / ЗВО «Університет Короля Данила», 2021. – 150 с. – Режим доступу: http://repository.ukd.edu.ua/bitstream/handle/123456789/152/ТЙ_Навчальний посібник_2е видання.pdf?sequence=1&isAllowed=y

Barlow, M. T. Lectures on probability theory and statistics [Electronic resource] / M. T. Barlow, D. Nualart. – Berlin : Springer, 2018. — 244 p. – Mode of access: https://www.researchgate.net/publication/283913234_Lectures_on_Probability_Theory_and_Statistics

Durrett, R. Probability, Theory and Examples [Electronic resource] / R. Durrett. – Wadsworth Publishing, 2016. – Mode of access: https://services.math.duke.edu/~rtd/PTE/PTE5_011119.pdf

Gine, E. Lectures on probability theory and statistics [Text] / E. Gine, G. R. Grimmett, L. Saloff-Coste. – Berlin : Springer, 2016. – 264 p.

Навчальне видання

Лучшева Оксана Вадимівна
Захаренко Володимир Олександрович

ОСНОВИ СТАТИСТИКИ В PYTHON

Редактор Н. В. Мазепа

Зв. план, 2026
Підписано до видання 08.09.2026
Ум. друк. арк. 6,2. Обл.-вид. арк. 6,94. Електронний ресурс

Видавець і виготовлювач
Національний аерокосмічний університет
«Харківський авіаційний інститут»
61070, Харків-70, вул. Вадима Манька, 17
http://www.khai.edu

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготовлювачів і розповсюджувачів
видавничої продукції сер. ДК № 391 від 30.03.2001