2. Лабораторна робота. Застосування мови Python до статистичних задач у теорії ймовірностей

Мета: засвоїти можливості роботи мови Python до застосування розв'язування статистичних задач теорії ймовірностей.

Python є потужним інструментом для розв'язання статистичних задач завдяки бібліотекам, таким як NumPy, Pandas, SciPy, Statsmodels, Scikit-learn, Matplotlib та Seaborn. Нижче наведені приклади задач зі статистики, які можна вирішити за допомогою Python.

Методичні рекомендації до написання коду

1. Описова статистика

Задача. Обчислити середнє значення, медіану, моду, дисперсію та стандартне відхилення.

```
import numpy as np
from scipy import stats

data = [12, 15, 14, 10, 8, 15, 18, 20]

mean = np.mean(data)
median = np.median(data)
mode = stats.mode(data)
variance = np.var(data, ddof=1)
std_dev = np.std(data, ddof=1)

print(f"Середнє: {mean}, Медіана: {median}, Мода: {mode.mode}, Дисперсія: {variance}, Стандартне відхилення: {std_dev}")
```

Результат:

Середнє: 14.0, Медіана: 14.5, Мода: 15, Дисперсія: 15.714285714285714, Стандартне відхилення: 3.9641248358604595

2. Тестування статистичних гіпотез

Задача: Виконати *t*-тест для перевірки середнього значення вибірки.

```
from scipy.stats import ttest_1samp

data = [2.3, 2.5, 2.8, 3.0, 2.9, 3.1]
t_stat, p_value = ttest_1samp(data, 3.0)  # Нульова гіпотеза: середнє значення = 3.0

print(f"T-статистика: {t_stat}, P-значення: {p_value}")
```

Результат:

T-статистика: -1.8576071235199598, P-значення: 0.12234647325914844

Задача: Виконати t-тест для двох вибірок.

```
from scipy.stats import ttest_ind

group1 = [23, 20, 22, 24, 25]
group2 = [30, 28, 27, 25, 26]
t_stat, p_value = ttest_ind(group1, group2)

print(f"T-статистика: {t_stat}, P-значення: {p_value}")
```

Результат:

T-статистика: -3.616777720717859, P-значення: 0.006814354918278612

3. Кореляція та регресія

Задача: Обчислити коефіцієнт кореляції між двома наборами даних.

```
from scipy.stats import pearsonr, spearmanr

x = [10, 20, 30, 40, 50]
y = [12, 24, 36, 48, 60]

pearson_corr, _ = pearsonr(x, y)
spearman_corr, _ = spearmanr(x, y)

print(f"Кореляція Пірсона: {pearson_corr}, Кореляція Спірмана: {spearman_corr}")
```

Результат:

Кореляція Пірсона: 0.9999999999999998, Кореляція Спірмана: 0.9999999999999999

Задача: Виконати лінійну регресію.

```
from scipy.stats import linregress

x = [1, 2, 3, 4, 5]
y = [2, 4, 5, 4, 5]

slope, intercept, r_value, p_value, std_err = linregress(x, y)

print(f"Нахил: {slope}, Перетин: {intercept}, R-квадрат: {r_value**2}, P-значення: {p_value}")
```

Результат:

Нахил: 0.6000000000000001, Перетин: 2.1999999999999997, R-квадрат: 0.6000000000000002, P-значення: 0.12402706265755441

4. Візуалізація статистичних даних

Задача: Побудувати гістограму (див. рис. 2.1) та ящик boxplot (див. рис. 2.2).

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns

data = [12, 15, 14, 10, 8, 15, 18, 20]

plt.hist(data, bins=5, color='blue', alpha=0.7)
plt.title('Гістограма')
plt.xlabel('Значення')
plt.ylabel('Частота')
plt.show()

sns.boxplot(data)
plt.title('Boxplot')
plt.show()
```

Результат:

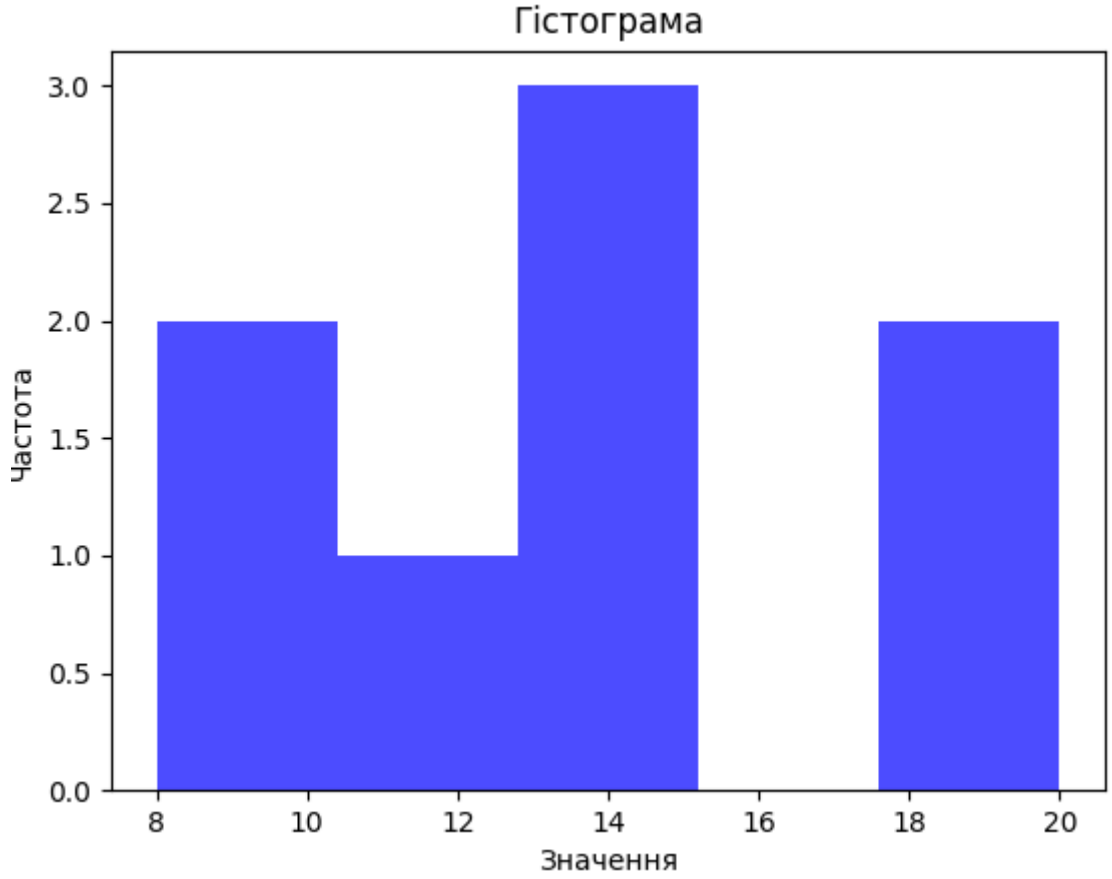

Рисунок 2.1 – Вивід гістограми засобами `matplotlib.pyplot`

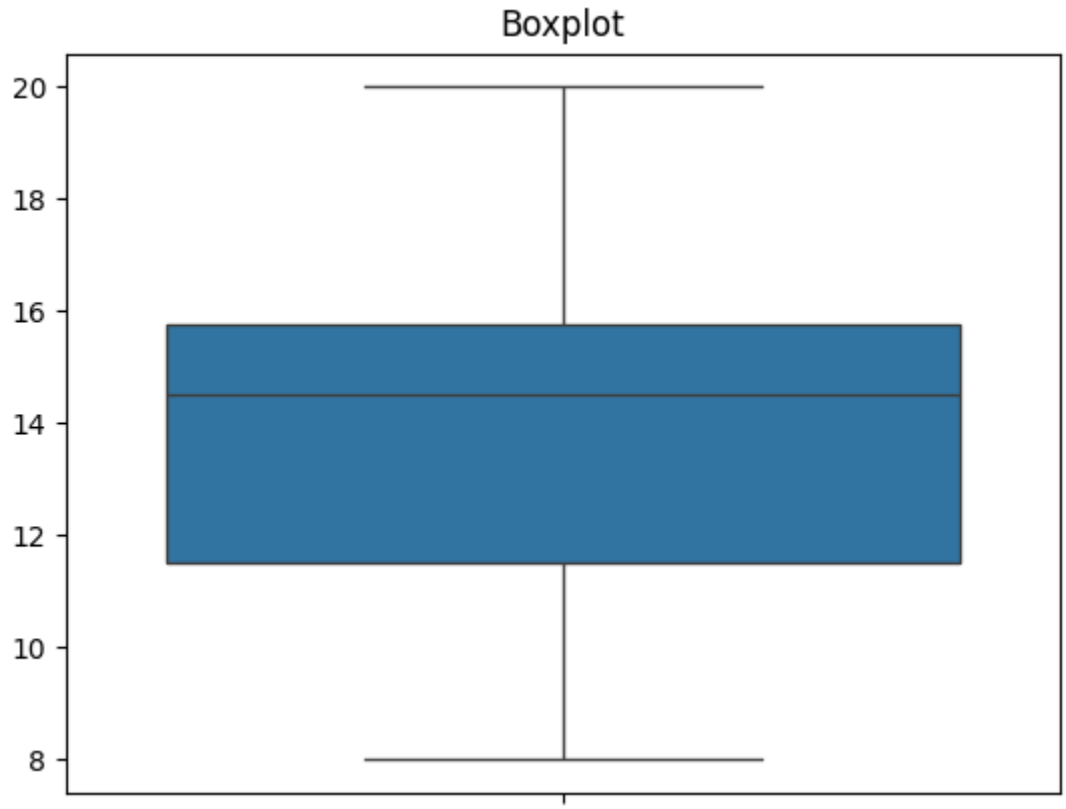

Рисунок 2.2 – Вивід boxplot

5. Аналіз розподілу

Задача: Перевірити, чи відповідає вибірка нормальному розподілу.

```
from scipy.stats import shapiro

data = [12, 15, 14, 10, 8, 15, 18, 20]
stat, p = shapiro(data)
print(f"Статистика Шапіро-Уілка: {stat}, P-значення:
{p}")
if p > 0.05:
    print("Дані відповідають нормальному розподілу.")
else:
    print("Дані не відповідають нормальному розподілу.")
```

Результат:

Статистика Шапіро-Уілка: 0.9781675297855071, P-значення: 0.9532741883063235

Дані відповідають нормальному розподілу.

6. Статистичне моделювання

Задача: Генерація вибірок та моделювання розподілів (див. рис. 2.3).

```
import numpy as np
import matplotlib.pyplot as plt

# Генерація нормального розподілу
```

```
data = np.random.normal(loc=0, scale=1, size=1000)

plt.hist(data, bins=30, density=True, alpha=0.6,
color='g')
plt.title('Нормальний розподіл')
plt.show()
```

Результат:

Рисунок 2.3 – Вивід нормального розподілу

7. Аналіз варіації (ANOVA)

Односторонній дисперсійний аналіз (ANOVA) – це статистичний метод, який використовується для перевірки значущих відмінностей між середніми значеннями груп даних. Він зазвичай використовується в експериментальних дослідженнях для порівняння впливу різних методів лікування або втручань на певний результат.

Основна ідея ANOVA полягає в тому, щоб розділити загальну варіабельність даних на два компоненти: варіабельність між групами (внаслідок лікування) і варіабельність всередині кожної групи (внаслідок випадкової варіабельності та індивідуальних відмінностей). Тест ANOVA обчислює F-статистику, яка є відношенням міжгрупової варіації до внутрішньогрупової варіації.

Односторонній дисперсійний аналіз – це статистичний тест, який використовується для визначення того, чи існують значущі відмінності між середніми значеннями двох або більше незалежних груп. Він використовується для перевірки нульової гіпотези про те, що середні всіх груп рівні, проти

альтернативної гіпотези про те, що принаймні одне середнє відрізняється від інших.

Задача: Провести однофакторний ANOVA-аналіз.

```
from scipy.stats import f_oneway

group1 = [23, 20, 22, 24, 25]
group2 = [30, 28, 27, 25, 26]
group3 = [20, 21, 22, 23, 24]

f_stat, p_value = f_oneway(group1, group2, group3)
print(f"F-статистика: {f_stat}, P-значення: {p_value}")
```

Результат:

F-статистика: 11.878787878787882, P-значення: 0.0014284961417153712

8. Байєсівський аналіз

Задача: Оцінити ймовірності за допомогою формули Байєса.

```
# Ймовірність A та B
p_a = 0.3  # P(A)
p_b = 0.4  # P(B)
p_b_given_a = 0.7  # P(B|A)

# P(A|B)
p_a_given_b = (p_b_given_a * p_a) / p_b
print(f"P(A|B): {p_a_given_b}")
```

Результат:

P(A|B): 0.5249999999999999

9. Кластеризація даних

Scikit-learn – це модуль Python для машинного навчання, створений на основі SciPy і розповсюджується за ліцензією 3-Clause BSD.

Задача: Виконати кластеризацію методом k-means.

```
from sklearn.cluster import KMeans
import numpy as np

data = np.array([[1, 2], [1, 4], [1, 0],
                 [10, 2], [10, 4], [10, 0]])

kmeans = KMeans(n_clusters=2, random_state=0).fit(data)
print(f"Кластери: {kmeans.labels_}")
```

Результат:

Кластери: [1 1 1 0 0 0]

Ці задачі демонструють можливості Python для аналізу, обробки даних і вирішення статистичних проблем, забезпечуючи точні результати та можливість легкої візуалізації.

Завдання до лабораторної роботи

1. Згенерувати випадкову вибірку випадкових величин, об’ємом 100 елементів (цілі числа). Використовуючи бібліотеку NumPy, обчислити середнє значення, медіану, моду, дисперсію та стандартне відхилення.

2. Згенерувати випадкову вибірку випадкових величин, об’ємом 50 елементів (дробові числа, які містять один знак після коми). Виконати *t*-тест для перевірки середнього значення вибірки. Пояснити отриманий результат.

3. Згенерувати дві випадкові вибірки випадкових величин, об’ємом 20 елементів (цілі числа). Обчислити коефіцієнт кореляції між двома наборами даних (кореляція Пірсона, кореляція Спірмана). Пояснити отриманий результат.

4. Згенерувати випадкову вибірку випадкових величин, об’ємом 25 елементів (цілі числа). Зробити візуалізація статистичних даних за допомогою бібліотек Matplotlib і Seaborn.

5. Згенерувати випадкову вибірку випадкових величин, об’ємом 75 елементів (цілі числа). Перевірити, чи відповідає вибірка нормальному розподілу. Пояснити отриманий результат.

6. Згенерувати три випадкові вибірки випадкових величин, об’ємом 20 елементів (цілі числа). Провести односторонній дисперсійний аналіз (ANOVA-аналіз). Пояснити отриманий результат.

Питання для самоконтролю

1. Наведіть приклади сучасних бібліотек мови Python, які використовуються для розв’язання статистичних задач. Охарактеризуйте їх можливості.
2. Сформулюйте означення таких понять: середнє значення вибірки, медіана, мода, дисперсія та стандартне відхилення вибірки.
3. Як можна обчислити середнє значення, медіану, моду, дисперсію та стандартне відхилення вибірки?
4. Що називають коефіцієнтом кореляції між двома наборами даних?
5. Як робиться візуалізація статистичних даних за допомогою бібліотек Matplotlib і Seaborn?
6. Охарактеризуйте коефіцієнт кореляції Пірсона та коефіцієнт кореляції Спірмана. Що вони означають?
7. Що таке односторонній дисперсійний аналіз?