Міністерство освіти і науки
України Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського” Факультет
інформатики та обчислювальної техніки Кафедра автоматики та
управління в технічних системах

Лабораторна робота №1

Теорія ймовірності і математична статистика

«Описова статистика»

Варіант 65

Виконали
студенти групи
ІА-11:

Юраш Б.В.
Воробей А.О.
Нікіфоров М.С.
Мельник В.О.

Перевірив:

ас. Цимбал С. І.

Київ 2022

Мета роботи

Навчитись розраховувати числові характеристики вибірки програмними засобами.

Теоретичні відомості

Множину однорідних об’єктів називають *статистичною сукупністю*. *Вибірковою сукупністю (вибіркою)* називають сукупність випадково взятих об’єктів із статистичної сукупності.

Генеральною називають сукупність об’єктів, з яких зроблено вибірку. *Об’ємом* сукупності (вибіркової або генеральної) називають кількість об’єктів цієї сукупності.

Полігоном частот вибірки називають ламану з вершинами в точках (x_i, n_i). *Полігоном відносних частот* вибірки називають ламану з вершинами в точках $(x_i, n_i/n)$. Полігони частот є аналогами щільності ймовірностей.

Гістограмою частот називають ступінчасту фігуру, яка складається з прямокутників, основами яких є інтервали варіант довжиною $h = x_i - x_{i-1}$, а висоти дорівнюють n_i/h.

Нехай $x_1, x_2, \dots, x_n$ – спостереження (значення величини X) у вибірці з об’ємом n. Тоді *вибіркове середнє* можна знайти за формулою:

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

Медіана вибірки, що має n відсортованих значень, визначається як центральне значення, якщо n непарне число, або як середнє значення двох центральних значень, якщо n парне число.

Мода вибірки визначається як значення, що зустрічається найчастіше у вибірці. Вибірка даних може мати більше однієї моди, і в цьому випадку вона називається *мультимодальною вибіркою*.

Якщо $x_1, x_2, \dots, x_n$ є вибіркою з об’ємом n, тоді *вибіркова дисперсія*

розраховується за формулою:

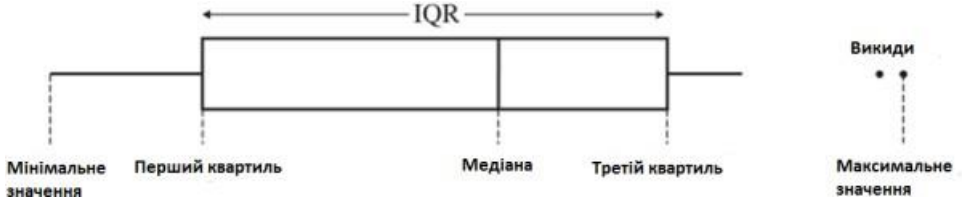
$$s^2 = \frac{\sum_{i=1}^{n} \left(x_i - \overline{x} \right)^2}{n-1}$$

або

$$s^2 = \frac{\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i \right)^2}{n}}{n-1} = \frac{\sum_{i=1}^{n} x_i^2 - n\overline{X}^2}{n-1}$$

Значення $s = \sqrt{s^2}$ називається *вибірковим середнім квадратичним відхиленням*.

Діаграма розмаху або коробкова діаграма – це схематичне представлення положення даних, включаючи найменші та найбільші значення, нижню та верхню чверть вибірки (нижній та верхній квартилі), медіану та статистичні викиди. Виглядає діаграма наступним чином:



Діаграма Парето – це діаграма, де категорії розташовані в порядку зменшення частоти. Діаграма Парето використовується для контролю якості та вдосконалення процесів з метою визначення декількох основних причин більшості проблем та першочергового їх вирішення.

Кругова діаграма або секторна діаграма створюється діленням кола на сектори, де розмір сектора відображає відносну частоту категорії у відсотках.

Завдання

1. Згенерувати вибірку об’ємом n (див. варіанти завдань нижче) з нормальної популяції. Значення математичних сподівань обрати самостійно.

2. Написати програму, що:

а) будує полігон та гістограму частот;

б) розраховує вибіркове середнє, медіану, моду, вибіркові дисперсію та середньоквадратичне відхилення заданої вибірки (написати власні реалізації розрахунків відповідних характеристик);

в) будує діаграми розмаху, Парето та кругову;

г) виводить результати пунктів а)-в).

3. Скласти звіт до виконаної роботи, в якому навести значення математичних сподівань, згенерованої вибірки, скріншоти результатів відповідно до п. 2 та посилання на репозиторій з кодом (лінк з qr-кодом на останній сторінці звіту).

Хід роботи

Завдання А
(будує полігон та гістограму частот)

```
def frequencies_polygon(sample, n):
    sample.sort()
    l = 1

    for i in range(len(sample)):
        number = round(sample[i], 1)
        if i + 1 < n:
            if number < round(sample[i + 1], 1):
                l += 1
            else:
                continue

    polygon_sample = [0] * l
    m = 0

    for i in range(len(sample)):
        number = round(sample[i], 1)
        if i + 1 < n:
            if round(sample[i + 1], 1) == number:
                continue
        polygon_sample[m] = number
        m += 1

    frequencies = [0] * l
    k = 0

    for i in range(len(sample)):
        number = round(sample[i], 1)
        if i + 1 < n:
            if round(sample[i + 1], 1) == number:
                continue
        temp = 0
        for j in range(len(sample)):
            if number == round(sample[j], 1):
                temp += 1
        frequencies[k] = temp
        k += 1
    plt.xlabel('value')
    plt.ylabel('frequency')
    x = plt.hist(sample, bins=l)
    x_array = valueForFrequenciPoligon(x[1])
    plt.xlabel('value')
    plt.ylabel('frequency')
    plt.plot(x_array, frequencies)
    plt.show()


def valueForFrequenciPoligon(x_array):
    result = []

    for j in range(len(x_array) - 1):
        result.append((x_array[j] + x_array[j + 1]) / 2)

    return result
```

Завдання В
(розраховує вибіркове середнє, медіану, моду, вибіркові дисперсію та середньоквадратичне відхилення заданої вибірки*)*

```python
# Вибіркове середнє
def sample_mean(arr):
    arr_sum = 0
    for number in arr:
        arr_sum += number
    return arr_sum / len(arr)


# Медіана
def median(arr):
    arr.sort()
    arr_len = len(arr)
    center_index = int(arr_len / 2)
    if arr_len % 2 == 0:
        right_central_index = center_index
        left_central_index = center_index - 1
        return (arr[right_central_index] + arr[left_central_index]) / 2
    return arr[center_index]


# Мода
def mode(arr):
    numbers_quantity = {}
    for number in arr:
        if number not in numbers_quantity:
            numbers_quantity[number] = 1
        else:
            numbers_quantity[number] = numbers_quantity[number] + 1
    return most_often_number(numbers_quantity)



def most_often_number(numbers_quantity):
    maximum = None
    for key, value in numbers_quantity.items():
        if maximum is not None:
            if value > maximum:
                maximum = value
        else:
            maximum = value

    res = []
    for key, value in numbers_quantity.items():
        if value == maximum:
            res.append(key)

    if len(res) == 1:
        return res[0]
    return res


# Вибіркова дисперпсія
def sample_variance(arr):
    square_number_sum = 0
    s_mean = sample_mean(arr)
    for number in arr:
        square_number_sum += number ** 2
    numerator = square_number_sum - len(arr) * (s_mean ** 2)
    denominator = len(arr) - 1
    return numerator / denominator


# Вибіркове середньоквадратичне відхилення
def sample_mean_squared_deviation(arr):
    return math.sqrt(sample_variance(arr))
```

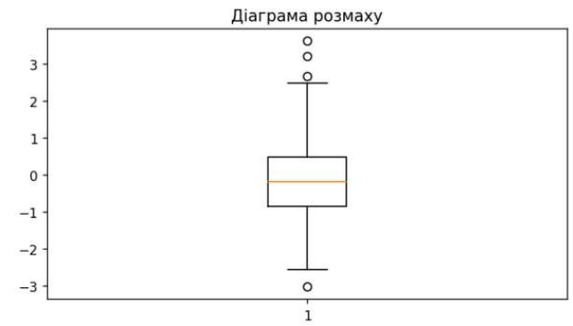
Завдання С
(будує діаграми розмаху, Парето та кругову)

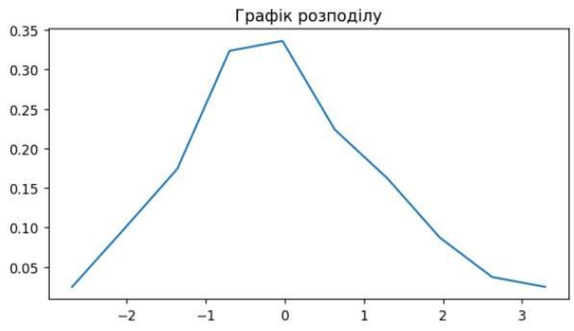
```python
# Helping functions
def ProbabilityDensityFunction(k, arr):
    pdf_x = []
    pdf_y = []
    n = len(arr)
    h = (max(arr) - min(arr)) / k
    a = min(arr)
    for i in range(0, k):
        count = 0
        for j in arr:
            if (a + i * h) < j < (a + (i * h) + h):
                count = count + 1
        pdf_x.append(a + i * h + h / 2)
        pdf_y.append(count / (n * h))
    pdf_x = list(map(lambda el: round(el, 2), pdf_x))
    d = {'x': pdf_x, 'y': pdf_y}
    return d


def sortValues(x, y):
    df = pd.DataFrame({'numbers': y})
    df.index = x
    df = df.sort_values(by='numbers', ascending=False)
    pareto_x = []
    for i in range(0, len(df.index.values)):
        x = string.ascii_uppercase[i] + f' {df.index.values[i]}'
        pareto_x.append(x)
    pareto = {'x': pareto_x, 'y': df["numbers"].values}
    return pareto


# Function to build graphics
def buildboxPlot(arr):
    plt.subplot(2, 2, 1)
    plt.title("Діаграма розмаху")
    plt.boxplot(arr)


def buildPDF(x, y):
    plt.subplot(2, 2, 2)
    plt.title("Графік розподілу")
    plt.plot(x, y)


def buildPareto(x, y):
    pareto = sortValues(x, y)
    plt.subplot(2, 2, 3)
    plt.title("Діаграма Парето")
    plt.bar(pareto["x"], pareto["y"])


def buildPieDensity(x, y):
    explode = (0.4, 0.3, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.4, 0.4)
    plt.subplot(2, 2, 4)
    plt.title("Кругова діаграма розподілу")
    plt.pie(y, explode=explode, labels=x, autopct='%1.1f%%')


def build_graphics(number_of_intervals, arr):
    coordinates = ProbabilityDensityFunction(number_of_intervals, arr)

    buildboxPlot(arr)
    buildPDF(coordinates["x"], coordinates["y"])
    buildPareto(coordinates["x"], coordinates["y"])
    buildPieDensity(coordinates["x"], coordinates["y"])

    plt.show()
```

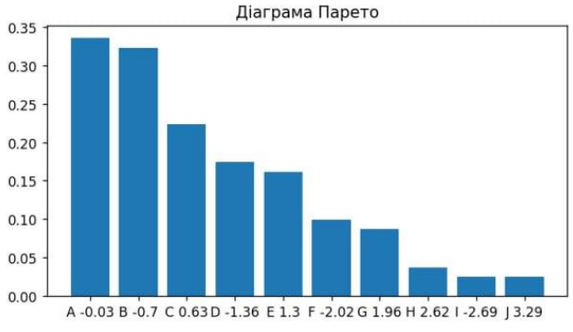
Завдання D
(виводить результати пунктів а)-в))

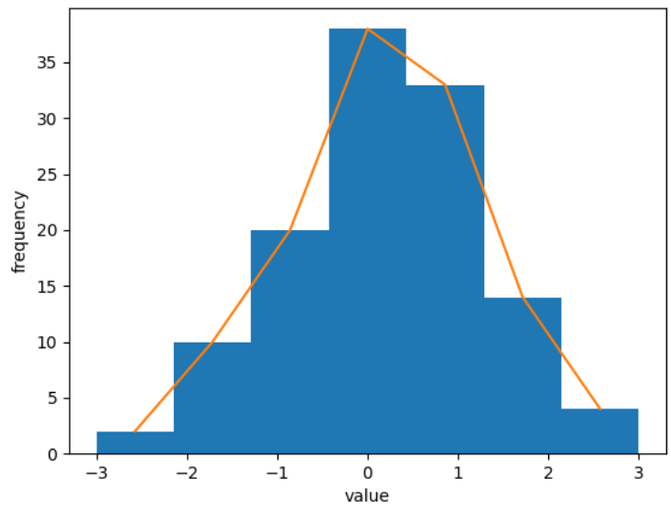
```
n = 121
mean_square_deviation = 1.3
sample = np.random.normal(0.0, mean_square_deviation, n).round(decimals=0)
# Variant
print("Variant: ", 120 % 11 + 11 * 5)

# Task A
print("------------Task A------------")
print("Це діаграма")
frequencies_polygon(sample, n)

# Task B
print("------------Task B------------")
print("Вибіркове середнє: ", round(sample_mean(sample), 3))
print("Медіана: ", median(sample))
print("Мода: ", mode(sample))
print("Вибіркова дисперсія: ", round(sample_variance(sample), 3))
print("Вибіркове середньоквадратичне відхилення: ", round(sample_mean_squared_deviation(sample), 3))

# Task C
print("------------Task C------------")
print("Це діаграма")
build_graphics(10, sample)
```

```
Variant:  65
------------Task A------------
Це діаграма
------------Task B------------
Вибіркове середнє:  0.223
Медіана:  -0.0
Мода:  -0.0
Вибіркова дисперсія:  1.641
Вибіркове середньоквадратичне відхилення:  1.281
------------Task C------------
Це діаграма
```









Висновок – на лабораторній роботі, ми навчилися застосовувати знання з теорії ймовірності у програмування Python. Написали програмний код, який будує полігон та гістограму частот; розраховує вибіркове середнє, медіану, моду, вибіркові дисперсію та середньоквадратичне відхилення заданої вибірки (написати власні реалізації розрахунків відповідних характеристик); будує діаграми розмаху, Парето та кругову;