# Міністерство освіти і науки України Національний авіаційний університет Навчально-науковий інститут комп'ютерних інформаційних технологій Кафедра комп'ютеризованих систем управління

Лабораторна робота №3 з дисципліни «Імітаційне моделювання» на тему «Моделювання дискретних випадкових величин»

> Виконав: студент ННІКІТ групи СП-325 Клокун В. Д. Перевірила: Марченко Н. Б.

Київ 2019

### 1. МЕТА РОБОТИ

Ознайомитись з алгоритмом побудови ряду розподілу дискретних випадкових величин та його графічним зображенням; побудувати імітаційну модель отримання системи дискретних випадкових величин.

## 2. ХІД РОБОТИ

Для виконання роботи поставлені такі завдання:

- 1. Побудувати імітаційну модель отримання системи дискретних випадкових величин (СДВВ). Відповідно до варіанту завдання побудувати ряд розподілу дискретної випадкової величини X.
- 2. На основі СДВВ в створеній програмі побудувати многокутник розподілу та графік функції розподілу F(x).

ЗАВДАННЯ ЗА ВАРІАНТОМ В партії з  $n_1$  деталей є  $n_2$  стандартних деталей. Навмання з усієї партії вибирається 2 деталі. Побудувати ряд розподілу числа X — числа стандартних деталей серед відібраних. Скористатись гіпергеометричним розподілом.

Під час виконання роботи була розроблена імітаційна модель для виконання поставлених завдань і реалізована у вигляді відповідного програмного засобу (ліст. А.1). Реалізований програмний засіб був запущений на моделювання і надавав стабільний і очікуваний результат (рис. 1).

Рис. 1: Результат роботи програми: вікно терміналу

У результаті роботи програма будує ряд розподілу у вигляді таблиці. Перевіримо його значення:

$$\sum_{i=1}^{n} P_i = 0.35 + 0.50 + 0.15 = 1.$$

Як бачимо, ймовірності сходяться. Також програма будує графік функції ймовірностей (рис. 2), а також многокутник розподілу (рис. 3).

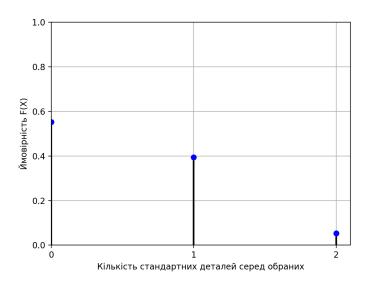


Рис. 2: Графік функції ймовірностей

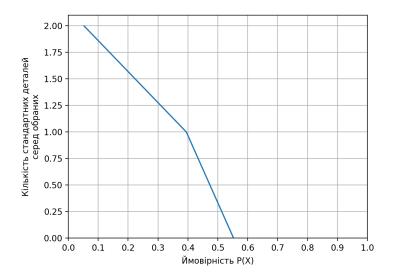


Рис. 3: Многокутник розподілу

### 3. Висновок

Виконуючи дану лабораторну роботу, ми ознайомились з алгоритмами моделювання результатів випробувань випадкових подій; побудували імітаційну модель функціонування системи протягом деякого часу.

# А. Повний початковий код програмної реалізації

Лістинг А.1: Повний початковий код програмної реалізації

```
import numpy as np
2 import scipy.stats as sp_stats
   import matplotlib.pyplot as plt
4
5
   def split_list_of_points(lofp):
 6
        lx, ly = zip(*lofp)
7
        return lx, ly
8
9
10
   def main():
11
        n_1 = float(input('Введіть кількість деталей в партії n_1: '))
12
        n_2 = float(input('Введіть кількість стандартних деталей n_2: '))
13
        num_samples = float(input(('Введіть, скільки деталей буде обрано: ')))
14
15
        total = n 1 # Total number of outcomes
16
                    # Number of good outcomes
        good = n_2
17
18
        rand_var = sp_stats.hypergeom(total, good, num_samples)
19
        possible_outcomes = np.arange(num_samples+1)
20
        # pmf --- probability mass function
21
        pmf_details = rand_var.pmf(possible_outcomes)
22
23
        # Probability function
24
25
        fig = plt.figure(1)
26
        ax = fig.add subplot(111)
27
        ax.plot(possible_outcomes, pmf_details, 'bo')
28
        # Plot vlines from 0 to the probability value
29
30
        ax.vlines(possible_outcomes, 0, pmf_details, lw=2)
        ax.set_xticks(possible_outcomes)
31
        ax.set_xlim(0.0, None)
32
        ax.set_ylim(0.0, 1.0)
33
        ax.set_xlabel('Кількість стандартних деталей серед обраних')
34
        ax.set_ylabel('\Bar{M}moвірність F(X)')
35
36
        plt.grid()
```

```
37
        # Probability distribution polygon
38
        fig = plt.figure(2)
39
        plt.subplot(111)
40
        ax = fig.add_subplot(111)
41
        # Create a list of graph points sorted by X value (pmf_details)
42
        points = [i for i in sorted(zip(pmf_details, possible_outcomes))]
43
        # Split a list of points into tuples of X and Y coordinates
44
        x, y = split_list_of_points(points)
45
        ax.plot(x, y)
46
        ax.set_xlim([0.0, None])
47
48
        ax.set_ylim([0.0, None])
        ax.set_xticks(np.arange(1.1, step=0.1))
49
        ax.set_xlabel('Ймовірність P(X)')
50
        ax.set_ylabel('Кількість стандартних деталей\nсеред обраних')
51
52
53
        plt.grid()
        plt.show()
54
55
56
    if __name__ == '__main__':
57
        main()
58
59
```