

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

Завдання 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).

```
4. bash
(venv) Vlads-MacBook-Pro:01-solution myownreality$ python y03s02-imitmod-lab-03-solution.py

Введіть кількість деталей в партії n_1: 20
Введіть кількість стандартних деталей n_2: 5
Введіть, скільки деталей буде обрано: 2
```

Рис. 1: Результат роботи програми: вікно терміналу

В результаті роботи програма будує ряд розподілу числа X у вигляді графіку функції ймовірностей (рис. 2а), а також багатокутник розподілу (рис. 2б).

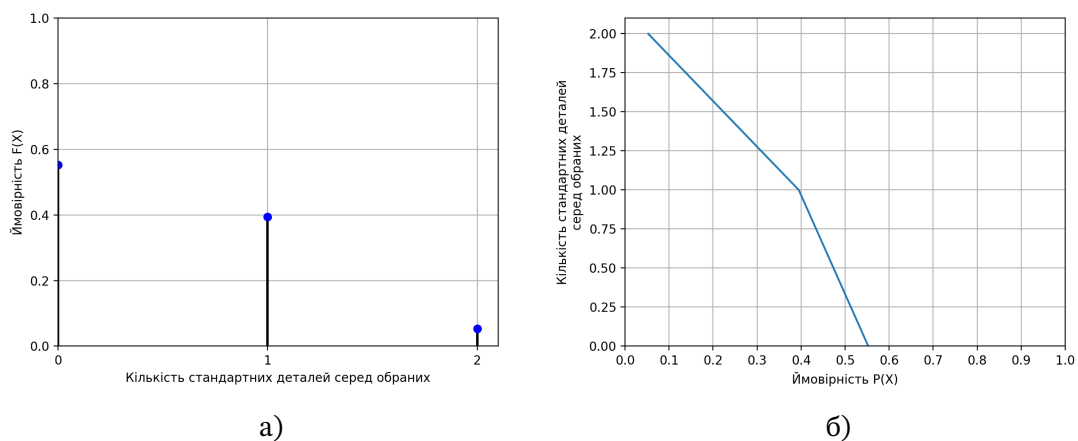


Рис. 2: Графіки, створені програмою: а — графік розподіл ймовірностей, б — багатокутник розподілу

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

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

```

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

```
