

# **Эмпирическая функция распределения вариационного ряда**

**Цель работы:** построить эмпирическую функцию распределения для дискретного и интервального вариационного ряда.

**Оборудование:** ПК, VS code, Python.

**Код программы:**

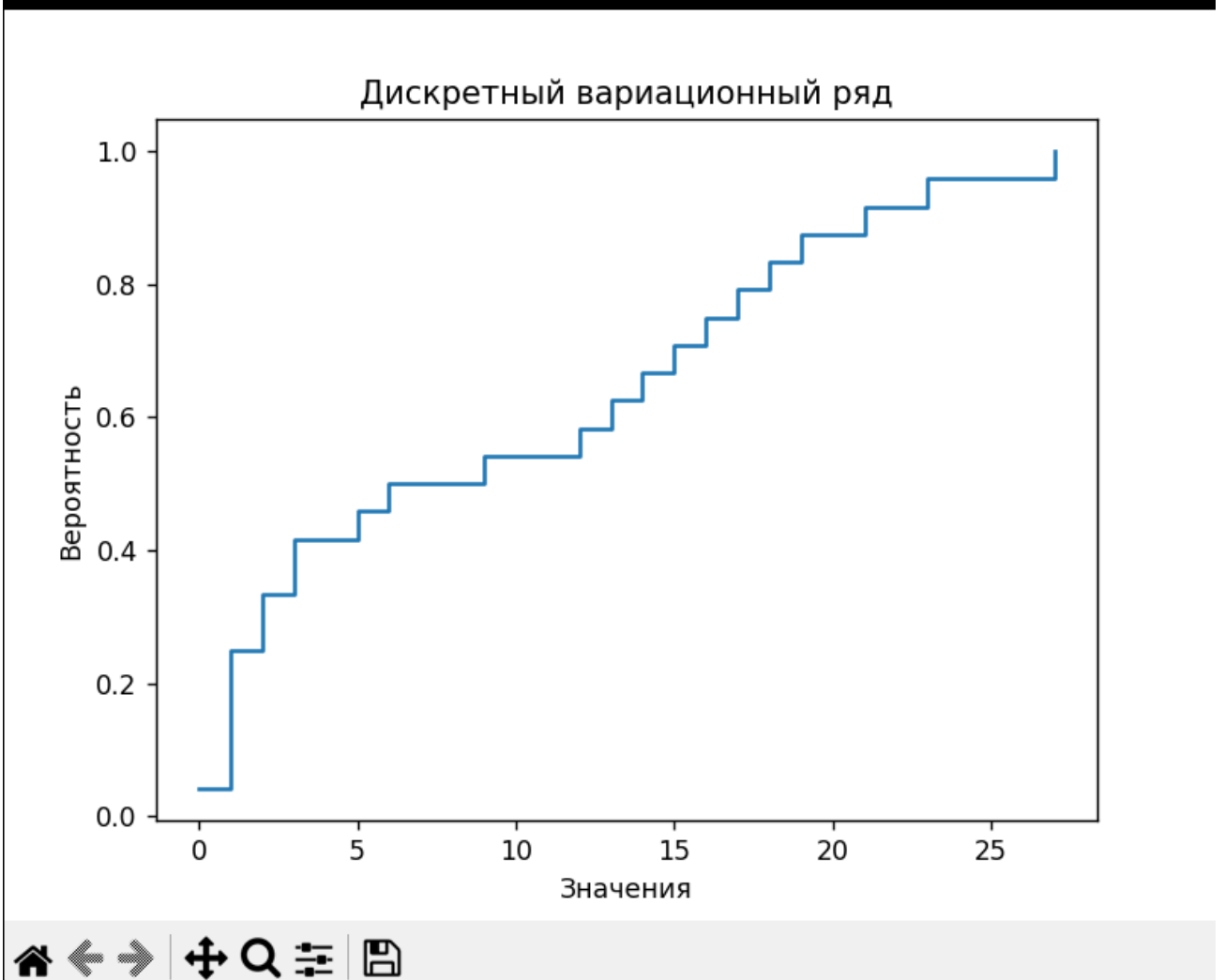
```

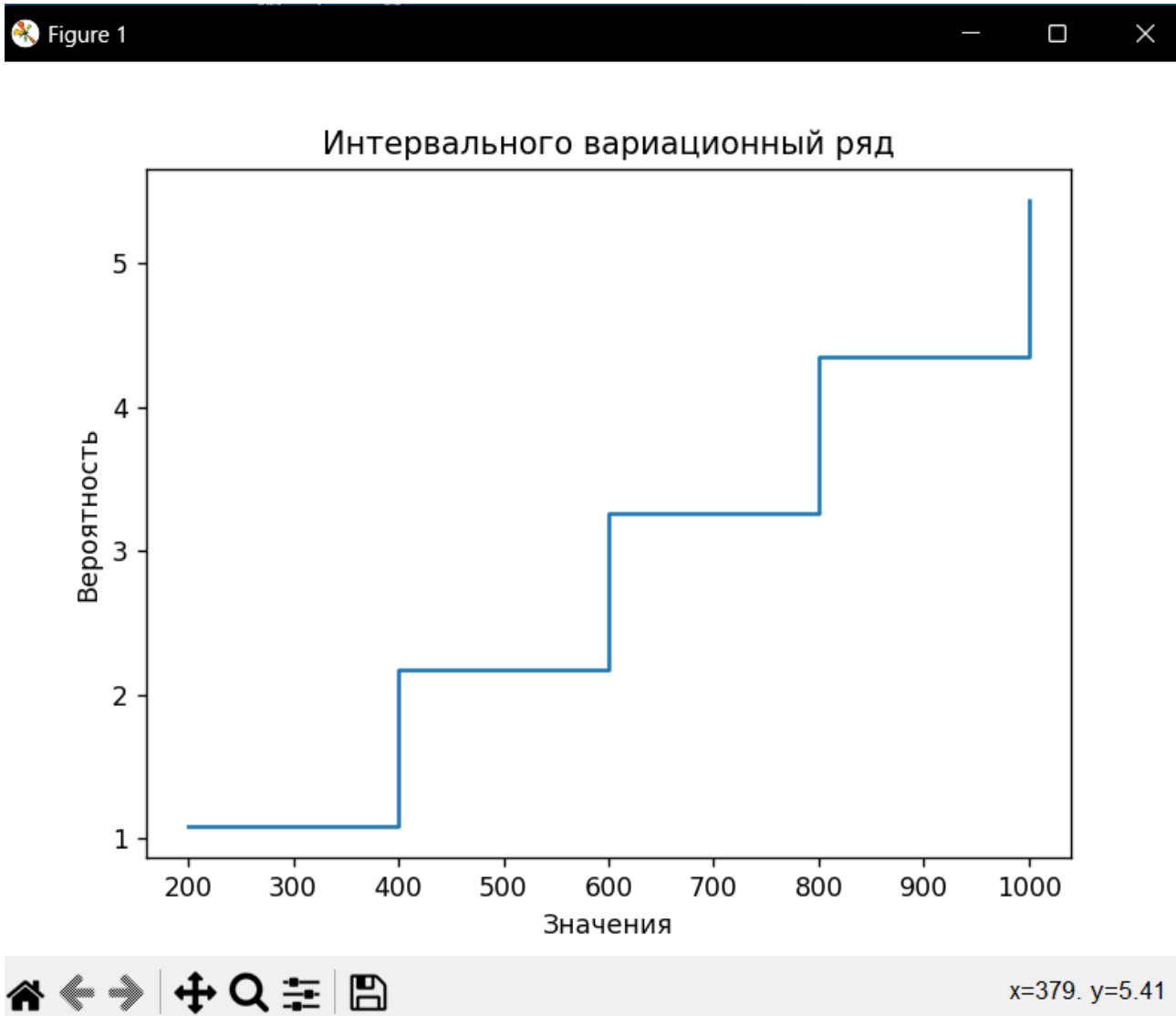
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  def discrete(data):
5      data_sorted = np.sort(data.flatten())
6      unique_values, counts = np.unique(data_sorted, return_counts=True)
7      probabilities = counts / len(data_sorted)
8      probabilities = np.cumsum(probabilities)
9      return unique_values, probabilities
10
11 def interval(data):
12     sorted_data = np.sort(data[0])
13     n = np.sum(data[1])
14     intervals = data[0]
15     interval_lengths = np.diff(intervals.flatten())
16     interval_lengths = np.cumsum(interval_lengths)
17     interval_probabilities = interval_lengths / n
18
19     return intervals, interval_probabilities
20
21 data_discrete = np.array([[9, 12, 13, 14, 15, 16, 17, 18, 19, 21, 23, 27],
22                           [1, 2, 3, 6, 5, 3, 2, 1, 1, 1, 1, 0]])
23 values_discrete, ecdf_discrete = discrete(data_discrete)
24
25 plt.step(values_discrete, ecdf_discrete, where='post')
26 plt.title('Дискретный вариационный ряд')
27 plt.xlabel('Значения')
28 plt.ylabel('Вероятность')
29 plt.show()
30
31
32 data_interval = np.array([[200, 400, 600, 800, 1000, 1200],
33                           [30, 38, 50, 31, 22, 13],
34                           [0.163, 0.207, 0.272, 0.168, 0.120, 0.07]])
35 intervals, ecdf_interval = interval(data_interval)
36
37 plt.step(intervals.flatten()[:-1], ecdf_interval, where='post')
38 plt.title('Интервального вариационный ряд')
39 plt.xlabel('Значения')
40 plt.ylabel('Вероятность')
41 plt.show()

```

**Результаты выполнения работы:**

Figure 1





#### Вывод:

Программа успешно выполнила работу, был построен график эмпирической функции распределения для дискретного и интервального вариационного ряда.