



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №1

Название Графики функций плотности и распределения СВ

Дисциплина Моделирование

Студент Прохорова Л. А.

Группа ИУ7-73Б

Оценка (баллы) _____

Преподаватель Рудаков И. В.

Москва — 2021 г.

1 Задание

Реализовать программу для построения графиков функции распределения и функции плотности вероятности для случайных величин, имеющих следующие распределения:

- равномерное распределение;
- распределение Пуассона.

2 Теоретические свойства

2.1 Равномерное распределение

Случайная величина X имеет непрерывное равномерное распределение $X \sim R(a, b)$ на отрезке $[a, b]$, где $a, b \in R$, если ее функция плотности $f(x)$ имеет следующий вид:

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in [a, b] \\ 0, & \text{иначе.} \end{cases} \quad (1)$$

Функция распределения $F(x) = \int_{-\inf}^x f(t)dt$ принимает вид:

$$F(x) = \begin{cases} \frac{x-a}{b-a}, & x \in [a, b] \\ 1, & x > b \\ 0, & x < a \end{cases} \quad (2)$$

2.2 Распределение Пуассона

Распределение Пуассона — распределение дискретного типа случайной величины, представляющей собой число событий, произошедших за фиксированное время, при условии, что данные события происходят с некоторой фиксированной средней интенсивностью и независимо друг от друга.

Распределение Пуассона имеет один параметр, обозначаемый символом λ – среднее количество успешных испытаний в заданной области возможных исходов. Дисперсия распределения Пуассона также равна λ , а его стандартное отклонение равно $\sqrt{\lambda}$. Количество успешных испытаний X пуассоновской случайной величины изменяется от 0 до бесконечности. Случайная величина Y имеет распределение Пуассона с математическим ожиданием λ записывается: $Y \sim P(\lambda)$.

Для дискретной случайной величины не существует функции плотности распределения вероятностей.

Функция вероятности при $\lambda > 0, k \geq 0$:

$$P(x = k) = \frac{\lambda^k}{k!} e^{-\lambda} \quad (3)$$

где

k — количество событий,

λ — математическое ожидание случайной величины (среднее количество событий за фиксированный промежуток времени),

$e=2,718281828\dots$ - основание натурального логарифма.

Функция распределения при $\lambda > 0, k \geq 0$:

$$F(x) = \sum_{k=0}^N \frac{\lambda^k}{k!} e^{-\lambda} \quad (4)$$

3 Результаты работы программы

Реализованная программа строит графики функций (1), (2), (3), (4) для параметров которые вводятся в консоль.

```
Введите(через пробел) границы интервала для равномерного распределения: 5 10
Введите(через пробел) значения параметров a и b для равномерного распределения: -3 5
Введите(через пробел) границы интервала и λ для распределения Пуассона: 0 20 2
```

Рисунок 1 – Пример ввода параметров.

Программа разработана в интегрированной среде разработки для языка программирования Python - PyCharm.

3.1 Равномерное распределение

На рисунках 2 и 3 приведены результаты построения графиков плотности вероятности $f(x)$ и распределения $F(x)$ соответственно, на отрезке $x \in [-5, 10]$, для случайной величины $X \sim R(-2, 5)$.

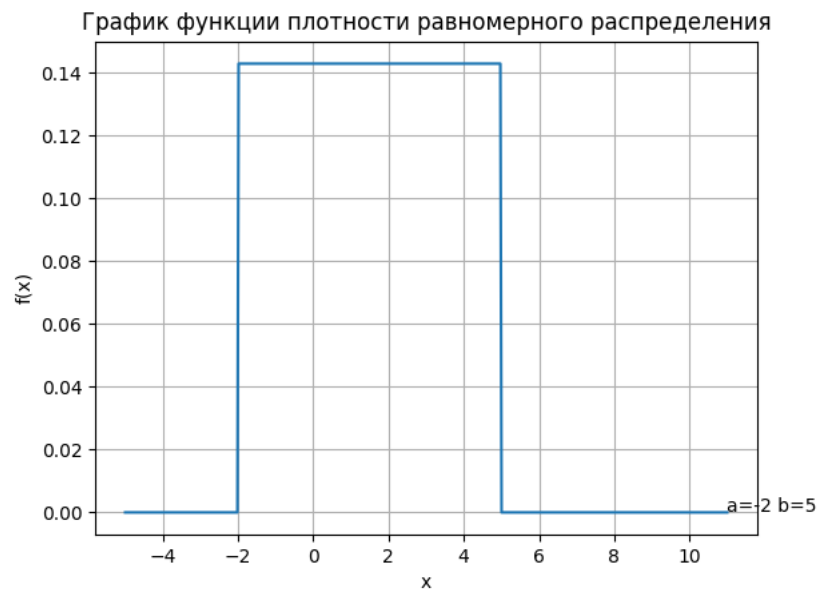


Рисунок 2 – График функции плотности распределения $f(x)$, $x \in [-5, 10]$, для случайной величины $X \sim R(-2, 5)$.

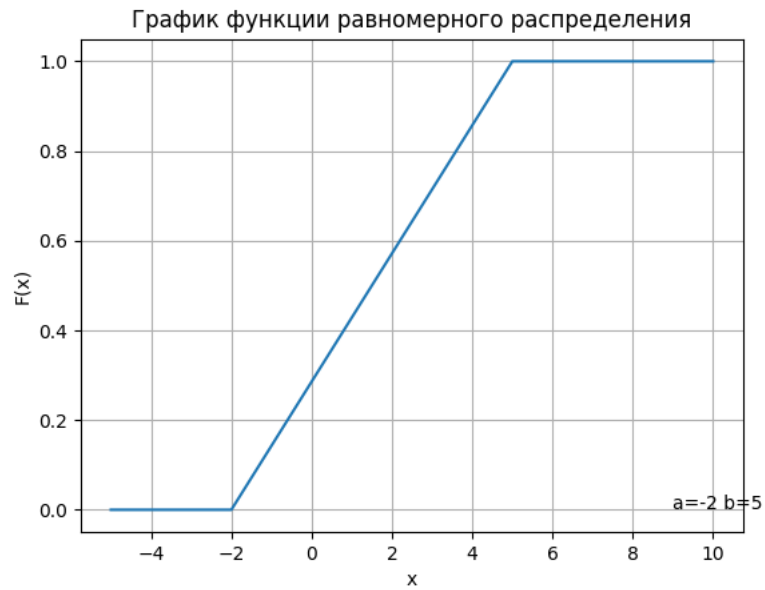


Рисунок 3 – График функции распределения $F(x)$, $x \in [-5, 10]$, для случайной величины $X \sim R(-2, 5)$.

3.2 Распределение Пуассона

На рисунках 4 и 5 приведены результаты построения графиков функции вероятности $P(x)$ и распределения $F(x)$ соответственно, на отрезке $x \in [0, 20]$ при $\lambda = 1$.

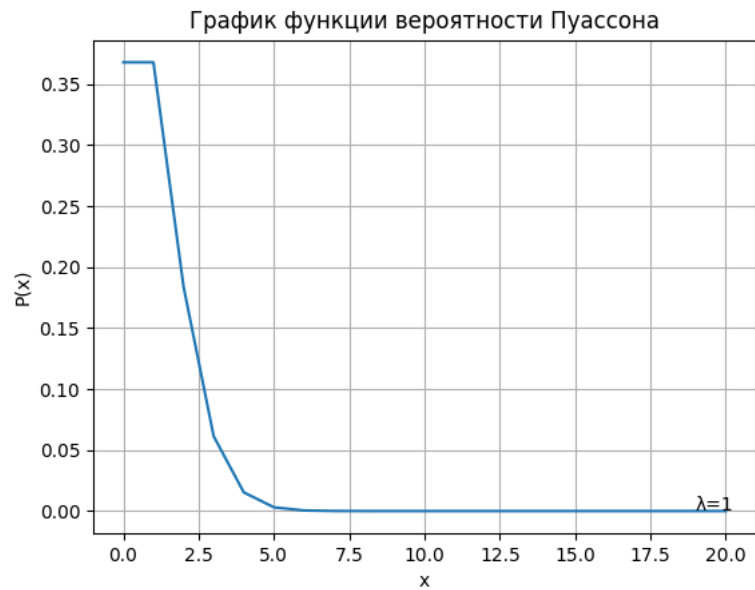


Рисунок 4 – График функции вероятности распределения Пуассона $P(x)$, $x \in [0, 20]$, при $\lambda = 1$.

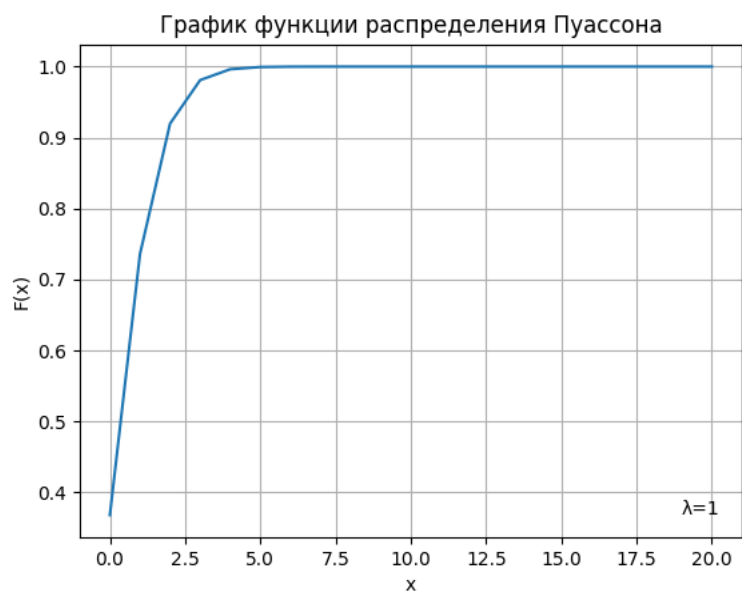


Рисунок 5 – График функции распределения Пуассона $P(x)$, $x \in [0, 20]$, при $\lambda = 1$.

На рисунках 6 и 7 приведены результаты построения графиков функции вероятности $P(x)$ и распределения $F(x)$ соответственно, на отрезке $x \in [0, 20]$ при $\lambda = 10$.

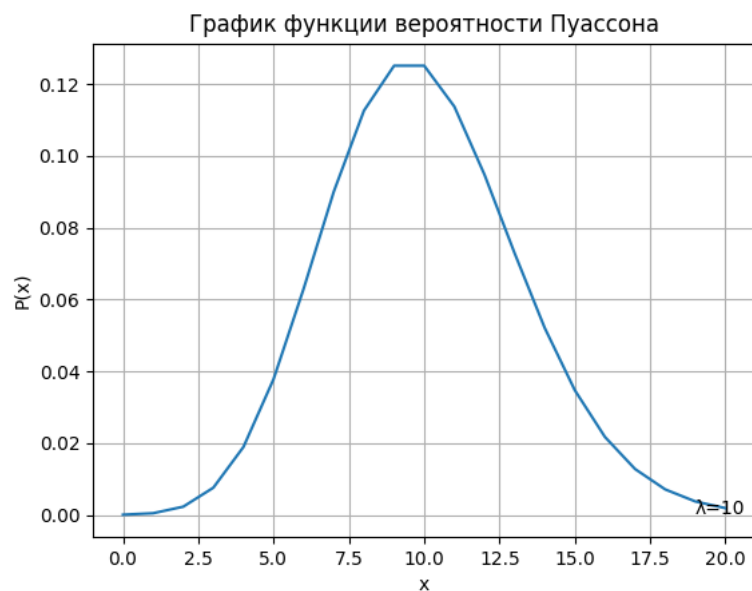


Рисунок 6 – График функции вероятности распределения Пуассона $P(x)$, $x \in [0, 20]$, при $\lambda = 10$.

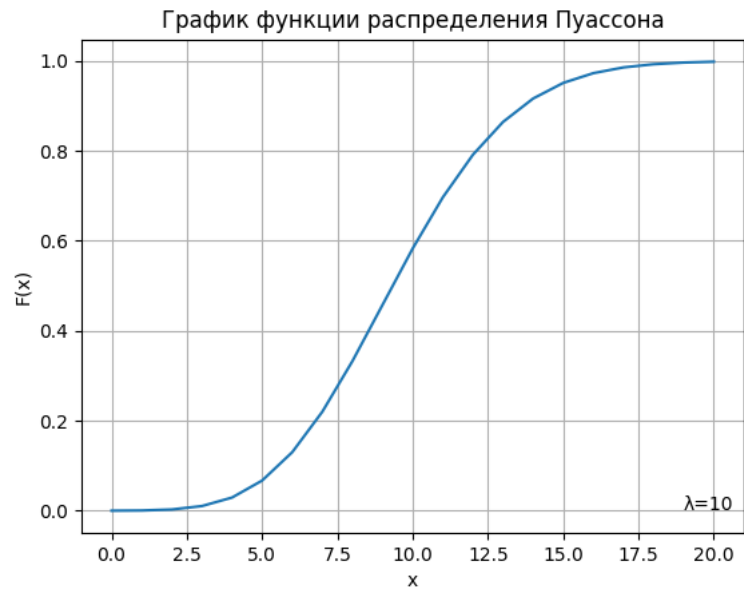


Рисунок 7 – График функции распределения Пуассона $P(x)$, $x \in [0, 20]$, при $\lambda = 10$.

4 Код программы

В листинге 4 приведена реализация лабораторной работы.

```

0 import matplotlib.pyplot as plt
1 from math import exp, factorial
2 import numpy
3 import sys
4
5 def uniform_density(a, b, x):
6     if x >= a and x <= b:
7         return 1 / (b - a)
8     return 0
9
10 def poisson_density(k, lmbd):
11     return exp(-lmbd) * pow(lmbd, k) / factorial(k)
12
13 def uniform_distribution(a, b, x):
14     if x < a:
15         return 0
16     elif x > b:
17         return 1

```

```

18     else:
19         return (x - a) / (b - a)
20
21 def poisson_distribution(k, lmbd):
22     sum = 0
23     for i in range(k + 1):
24         sum = sum + (pow(lmbd, i)) / factorial(i)
25     sum = sum * exp(-lmbd)
26     return sum
27
28 def poisson(k_start, k_stop, lmbd):
29     distribution_x = [i for i in range(k_start, k_stop + 1)]
30     distribution_y = [poisson_distribution(x, lmbd) for x in
31     distribution_x]
32     prepare_show_poisson(distribution_x, distribution_y, "График функции
33     распределения Пуассона", "x", "F(x)", lmbd)
34     plt.show()
35
36     density_x = [i for i in range(k_start, k_stop + 1)]
37     density_y = [poisson_density(x, lmbd) for x in density_x]
38     prepare_show_poisson(density_x, density_y, "График функции вероятности
39     Пуассона", "x", "P(x)", lmbd)
40     plt.show()
41
42 def uniform(start, stop, a, b):
43     distribution_x = [i for i in range(start, stop + 1)]
44     distribution_y = [uniform_distribution(a, b, x) for x in
45     distribution_x]
46     prepare_show_uniform(distribution_x, distribution_y, "График функции
47     равномерного распределения", "x", "F(x)", a, b)
48     plt.show()
49
50     density_x = numpy.linspace(start, stop + 1, num=500, endpoint=True,
51     retstep=False, dtype=None, axis=0)
52     density_y = [uniform_density(a, b, x) for x in density_x]
53     prepare_show_uniform(density_x, density_y, "График функции плотности
54     равномерного распределения", "x", "f(x)", a, b)
55     plt.show()
56
57 def prepare_show_poisson(x_value, y_value, title, x_label, y_label, lmbd):

```



```

51     plt.plot(x_value, y_value)
52     plt.grid(True)
53     plt.title(title)
54     plt.ylabel(y_label)
55     plt.xlabel(x_label)
56
57     label = "="
58     label += str(lmbd)
59     label_lmbd = label
60
61     plt.text(x_value[-2], min(y_value), label_lmbd, fontsize=10)
62
63 def prepare_show_uniform(x_value, y_value, title, x_label, y_label, a, b):
64     plt.plot(x_value, y_value)
65     plt.grid(True)
66     plt.title(title)
67     plt.ylabel(y_label)
68     plt.xlabel(x_label)
69
70     label = "a="+str(a)+" b="+str(b)
71     label_lmbd = label
72
73     plt.text(x_value[-2], min(y_value), label_lmbd, fontsize=10)
74
75 if __name__ == '__main__':
76     startU, stopU = map(int, input("Введите через ( пробел ) границы интервала для
    равномерного распределения: ").split())
77
78     if startU >= stopU:
79         print("Левая граница должна быть меньше правой")
80         sys.exit(0)
81
82     a, b = map(int, input("Введите через ( пробел ) значения параметров a и b для
    равномерного распределения: ").split())
83
84     if a >= b:
85         print("Параметр a должен быть меньше параметра b")
86         sys.exit(0)
87     uniform(startU, stopU, a, b)
88

```

```
89     start, stop, lmbd = map(int, input("Введите через пробел границы интервала
и для распределения Пуассона: ").split())
90     if start < 0 or stop < 0:
91         print("Границы интервала не могут быть отрицательными")
92         sys.exit(0)
93     elif start >= stop:
94         print("Левая граница должна быть меньше правой")
95         sys.exit(0)
96     elif lmbd < 0:
97         print(" должна быть больше 0")
98         sys.exit(0)
99     else:
100         poisson(start, stop, lmbd)
```