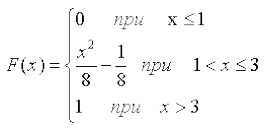
**Лабораторная работа №3**

**Функции распределения**

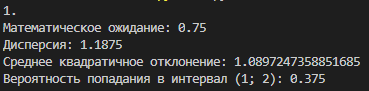
Задача №1

Постановка задачи: Для случайной величины, заданной интегральной функцией распределения



Найти: дифференциальную функцию случайной величины X; математическое ожидание, дисперсию, среднее квадратичной отклонение случайной величины X; вероятность попадания случайной величины в интервал (1;2).

Результаты:



Код программы:

import math

def task1():

    def f\_int(x):

        if x <= 1:

            return 0

        elif x > 3:

            return 1

        else:

            return x \* x / 8 - 1 / 8

    a = 1

    b = 2

    # Вероятность попадания в интервал (1; 2)

    # P(1 < x < 2)

    p = f\_int(b) - f\_int(a)

    # Дифференциальная функция распределения

    # def f\_dif(x):

    #     if x <= 3 or x > 3:

    #         return 0

    #     else:

    #         return x / 4

    # Математическое ожидание

    func = lambda x: x \*\* 2 / 4

    mx = func(b) - func(a)

    # Дисперсия

    func = lambda x: x \*\* 3 / 4

    dx = func(b) - func(a) - mx\*\*2

    # Среднее квадратичное отклонение

    delta\_x = math.sqrt(dx)

    print(f'1.\n'

          f'Математическое ожидание: {mx}\n'

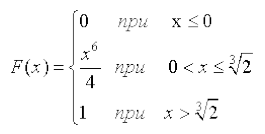
          f'Дисперсия: {dx}\n'

          f'Среднее квадратичное отклонение {delta\_x}\n'

          f'Вероятность попадания в интервал ({a}; {b}) = {p}')

Задача №2

Постановка задачи: Для интегральной функции случайной величины X



Найти вероятность того, что в результате шести испытаний случайная величина X два раза примет значение, принадлежащее интервалу (0;1).

Результаты:



Код программы:

import math

def task2():

    def f(x):

        if x <= 0:

            return 0

        elif x > 2 \*\* (1 / 3):

            return 1

        else:

            return x \*\* 6 / 4

    a = 0

    b = 1

    k = 2  # Кол-во наступлений события

    n = 6  # Кол-во испытаний

    #  Вероятность для одного

    p = f(b) - f(a)

    q = 1 - p

    # Бернулли

    c\_k\_n = math.factorial(n) / (math.factorial(k) \* math.factorial(n - k))

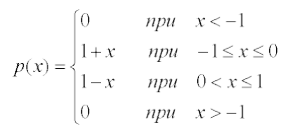
    p\_k\_n = c\_k\_n \* p\*\*k \* q\*\*(n - k)

    print(f'2.\n'

          f'Вероятность: {p\_k\_n}')

Задача №3

Постановка задачи: Для плотности распределения вероятностей, заданной следующим образом



Подсчитать вероятность того, что соответствующая величина примет значение от -0.5 до 1.

Результаты:



Код программы:

import math

def task3():

    def f(x):

        if x < -1 or x > 1:

            return 0

        elif -1 <= x <= 0:

            return 1 + x

        else:

            return 1 - x

    def f\_int(x):

        if x < -1 or x > 1:

            return 0

        elif -1 <= x <= 0:

            return x\*\*2 / 2 + x

        else:

            return x - x\*\*2 / 2

    a = -0.5

    b = 1

    p = f\_int(b) - f\_int(a)

    print(f'3.\n'

          f'Вероятность: {p}')