

**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

|  |
| --- |
| **РТУ МИРЭА** |
|  |
| **Институт кибербезопасности и цифровых технологий (ИКБ)** |
|  |
| КБ-2 «Информационно-аналитические системы кибербезопасности» |

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОГО ЗАДАНИЯ №2**

**В РАМКАХ ДИСЦИПЛИНЫ «МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННО-АНАЛИТИЧЕЧКИХ СИСТЕМ»**

Выполнил:

Студент 3-ого курса

Учебной группы БИСО-02-22

Зубарев В.С.

Задание

1). Постройте график функции вероятности треугольного распределения.

2). Смоделируйте случайный процесс.

3). Разработайте программу расчёта.

4). Получите выборку объёмом 200 значений.

5). Оцените выборочные характеристики и постройте гистограмму частотного распределения по выборке.

6). Сделайте выводы.

Решение (листинг кода)

#Зубарев, вариант 5  
import random  
import matplotlib.pyplot as plt  
import numpy as np  
  
# Входные параметры  
a1, b1 = 3, 16 # Параметры равномерного распределения S1  
a2, b2 = 3, 3 # Параметры равномерного распределения S2  
a3, b3, c = 4, 16, 8 # Параметры треугольного распределения S3  
Q = 5 # Коэффициент Q  
size = 200 # Размер выборки  
  
def uniform\_process(a, b, size):  
 *"""Генерация равномерного случайного процесса."""* return [(b - a) \* random.random() + a for \_ in range(size)]  
  
def triangular\_process(a, b, c, size):  
 *"""Генерация треугольного случайного процесса."""* result = []  
 for \_ in range(size):  
 u = random.random()  
 if u < (c - a) / (b - a):  
 result.append(a + ((b - a) \* (c - a) \* u) \*\* 0.5)  
 else:  
 result.append(b - ((b - a) \* (b - c) \* (1 - u)) \*\* 0.5)  
 return result  
  
def simulate\_process(size):  
 *"""Моделирование случайного процесса S."""* S1 = uniform\_process(a1, b1, size)  
 S2 = uniform\_process(a2, b2, size)  
 S3 = triangular\_process(a3, b3, c, size)  
 return [s1 + Q \* s2 + s3 for s1, s2, s3 in zip(S1, S2, S3)]  
  
# Генерация выборки случайного процесса  
samples = simulate\_process(size)  
  
# Вычисление выборочных характеристик  
mean\_val = sum(samples) / size  
std\_val = (sum((x - mean\_val) \*\* 2 for x in samples) / size) \*\* 0.5  
print(f"Выборочное среднее: {mean\_val:.4f}")  
print(f"Выборочное стандартное отклонение: {std\_val:.4f}")  
  
# Гистограмма частотного распределения  
hist\_bins = 15  
hist\_range = min(samples), max(samples)  
step = (hist\_range[1] - hist\_range[0]) / hist\_bins  
histogram = [0] \* hist\_bins  
for s in samples:  
 index = min(int((s - hist\_range[0]) / step), hist\_bins - 1)  
 histogram[index] += 1  
  
# Данные для графика функции вероятности треугольного распределения  
x = np.linspace(a3, b3, 1000)  
fx = np.where(x < c, 2 \* (x - a3) / ((b3 - a3) \* (c - a3)), 2 \* (b3 - x) / ((b3 - a3) \* (b3 - c)))  
  
# Построение графиков  
fig, axs = plt.subplots(3, 1, figsize=(8, 12))  
  
# График функции вероятности  
axs[0].plot(x, fx, label="PDF треугольного распределения", color='blue')  
axs[0].set\_xlabel("x")  
axs[0].set\_ylabel("Плотность вероятности")  
axs[0].set\_title("Функция вероятности треугольного распределения")  
axs[0].legend()  
axs[0].grid()  
  
# График треугольного распределения  
triangular\_sample = triangular\_process(a3, b3, c, size)  
axs[1].hist(triangular\_sample, bins=15, density=True, alpha=0.6, color='r', edgecolor='black')  
axs[1].set\_xlabel("Значения")  
axs[1].set\_ylabel("Частота")  
axs[1].set\_title("Гистограмма треугольного распределения")  
axs[1].grid()  
  
# Гистограмма выборки случайного процесса  
bin\_edges = [hist\_range[0] + i \* step for i in range(hist\_bins + 1)]  
axs[2].bar(bin\_edges[:-1], histogram, width=step, edgecolor='black', alpha=0.7, color='g')  
axs[2].set\_xlabel("Значения случайного процесса")  
axs[2].set\_ylabel("Частота")  
axs[2].set\_title("Гистограмма выборки")  
axs[2].grid()  
  
plt.tight\_layout()  
plt.show()

Результат работы программы



