Мигачев Павел Игоревич, группа 8.1

Лабораторная работа № 1

**Вариант № 7**

Моделирование случайных величин

**Цель работы**

Исследовать алгоритмы генерации случайных величин в среде Python. Научиться определять значения параметров случайной величины

**Задание**

**Описание варианта: Гамма-распределение (b = 2.2, c = 6.2)**

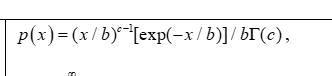
1. Постройте график зависимости значения выборочного математического ожидания от числа реализаций СВ. Так же отобразите на графике значение математического ожидания, вычисленное на основе соотношений из таблицы.

2. Постройте график зависимости значения выборочной дисперсии от числа реализаций СВ. Так же отобразите на графике значение дисперсии, вычисленное на основе соотношений из таблицы.

3. Выполните визуализацию эмпирической плотности распределения СВ (гистограммы с необходимой нормировкой). Так же отобразите на рисунке график плотности распределения, определённой на основе соотношений из таблицы. Постройте график зависимости значения средней абсолютной разности эмпирических и теоретических значений плотности распределения от числа реализаций СВ.

**Код программы**

import matplotlib.pyplot as plt  
import numpy as np  
import math  
  
*# Параметры распределения*b = 2.2  
c = 6.2  
  
*# Теоретические значения*m\_theory = c \* b *# Мат. Ожидание*D\_theory = c \* b\*\*2 *# Дисперсия*print(f"Теоретическое мат. ожидание: {m\_theory:.3f}")  
print(f"Теоретическая дисперсия: {D\_theory:.3f}")  
  
*# Генерация одной случайной величины гамма-распределения*def generate\_gamma(b, c):  
 return np.random.gamma(shape=c, scale=b)  
  
*# Плотность гамма-распределения в точке x*def gamma\_pdf(x, b, c):  
 *# Формула плотности: f(x) = (x^(c - 1) \* exp(-x / b)) / (b^c \* Γ(c))  
 # x - точка, в которой вычисляется плотность  
 # b - параметр масштаба, c - параметр формы* if x < 0:  
 return 0.0  
 return (x\*\*(c-1) \* np.exp(-x/b)) / (b\*\*c \* math.gamma(c))  
  
*# Генерация данных для анализа*np.random.seed(42) *# для воспроизводимости результатов*n\_max = 10000 *# объем выборки для анализа*n\_values = np.arange(1, n\_max + 1)  
  
*# Генерация всех значений сразу*all\_samples = np.array([generate\_gamma(b, c) for \_ in range(n\_max)])  
  
*# Расчет накапливающихся статистик*cumulative\_mean = np.cumsum(all\_samples) / n\_values *# выборочное мат. ожидание (среднее)*cumulative\_var = np.cumsum((all\_samples - cumulative\_mean)\*\*2) / np.maximum(n\_values - 1, 1) *# выборочная дисперсия как функция от числа реализаций  
  
# Задание 1: График выборочного мат. ожидания*plt.figure(figsize=(10, 6))  
plt.plot(n\_values, cumulative\_mean, label='Выборочное мат. ожидание')  
plt.axhline(y=m\_theory, color='r', linestyle='--', label='Теоретическое мат. ожидание')  
plt.xlabel('Число реализаций')  
plt.ylabel('Мат. ожидание')  
plt.title('Зависимость выборочного мат. ожидания от числа реализаций (гамма-распределение)')  
plt.legend()  
plt.grid(True)  
plt.show()  
  
*# Задание 2: График выборочной дисперсии*plt.figure(figsize=(10, 6))  
plt.plot(n\_values, cumulative\_var, label='Выборочная дисперсия')  
plt.axhline(y=D\_theory, color='r', linestyle='--', label='Теоретическая дисперсия')  
plt.xlabel('Число реализаций')  
plt.ylabel('Дисперсия')  
plt.title('Зависимость выборочной дисперсии от числа реализаций (гамма-распределение)')  
plt.legend()  
plt.grid(True)  
plt.show()  
  
*# Задание 3: Гистограмма и плотность распределения*plt.figure(figsize=(10, 6))  
n\_bins = 50  
counts, bin\_edges, \_ = plt.hist(all\_samples, bins=n\_bins, density=True,  
 alpha=0.7, label='Эмпирическая плотность')  
  
*# Теоретическая плотность*x = np.linspace(0, max(all\_samples), 1000) *# 1000 точек от 0 до максимума выборки*pdf\_values = [gamma\_pdf(x\_i, b, c) for x\_i in x] *# вычисление плотности в каждой точке*plt.plot(x, pdf\_values, 'r-', linewidth=2, label='Теоретическая плотность')  
  
plt.xlabel('Значение СВ')  
plt.ylabel('Плотность вероятности')  
plt.title('Эмпирическая и теоретическая плотности гамма-распределения')  
plt.legend()  
plt.grid(True)  
plt.show()  
  
*# Расчет средней абсолютной разности эмпирических и теоретических значений*def calculate\_mean\_abs\_diff(samples, b, c, n\_bins=50):  
 *# samples - массив случайных величин  
 # b, c - параметры гамма-распределения  
 # n\_bins - количество интервалов гистограммы* hist, bin\_edges = np.histogram(samples, bins=n\_bins, density=True) *# гистограмма с нормировкой* bin\_centers = (bin\_edges[:-1] + bin\_edges[1:]) / 2 *# центры интервалов гистограммы* theoretical\_pdf = np.array([gamma\_pdf(x, b, c) for x in bin\_centers]) *# теоретическая плотность в центрах* return np.mean(np.abs(hist - theoretical\_pdf)) *# средняя абсолютная разность (MAD)  
  
# График зависимости средней абсолютной разности*mad\_values = [] *# список для хранения значений средней абсолютной разности*for n in range(100, n\_max + 1, 100):  
 mad = calculate\_mean\_abs\_diff(all\_samples[:n], b, c) *# берём первые n элементов* mad\_values.append(mad)  
  
plt.figure(figsize=(10, 6))  
plt.plot(range(100, n\_max + 1, 100), mad\_values)  
plt.xlabel('Число реализаций')  
plt.ylabel('Средняя абсолютная разность')  
plt.title('Зависимость точности оценки плотности от числа реализаций (гамма-распределение)')  
plt.grid(True)  
plt.show()

Использовалась формула  


**Результаты выполнения задания**

1. График зависимости значения выборочного математического ожидания от числа реализаций СВ.

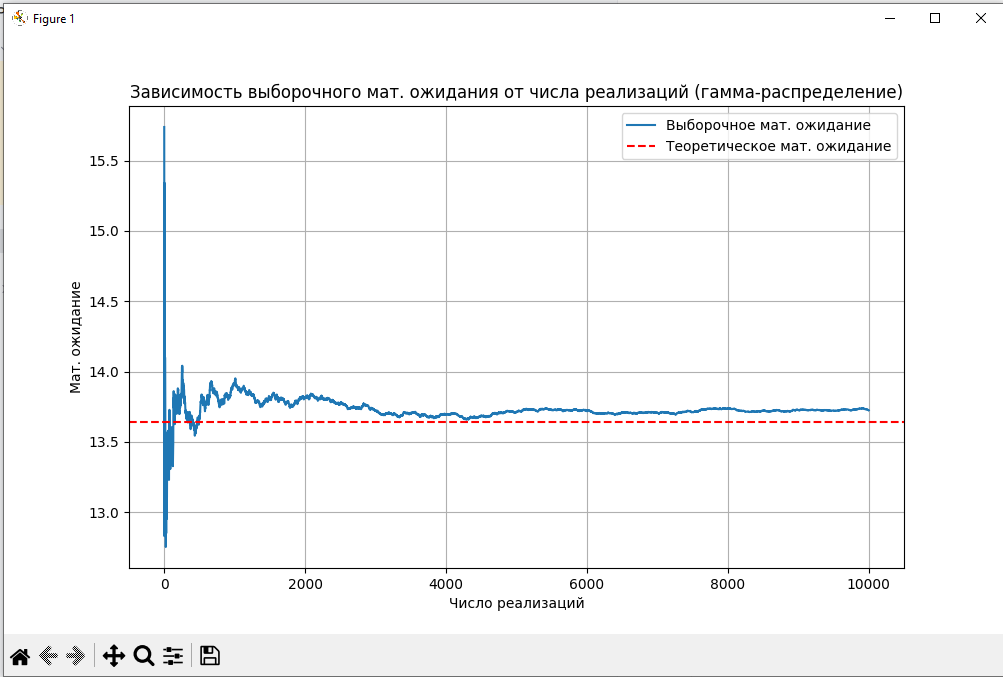


Рисунок 1.

1. График зависимости значения выборочной дисперсии от числа реализаций СВ.

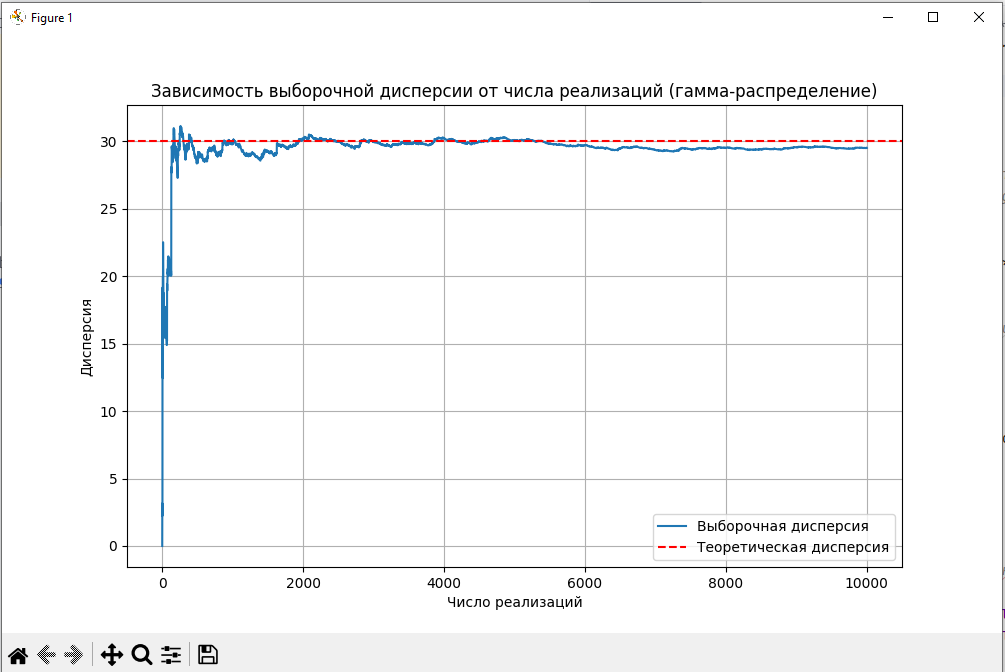


Рисунок 2.

1. Визуализация эмпирической плотности распределения СВ (гистограммы с необходимой нормировкой). График зависимости значения средней абсолютной разности эмпирических и теоретических значений плотности распределения от числа реализаций СВ.

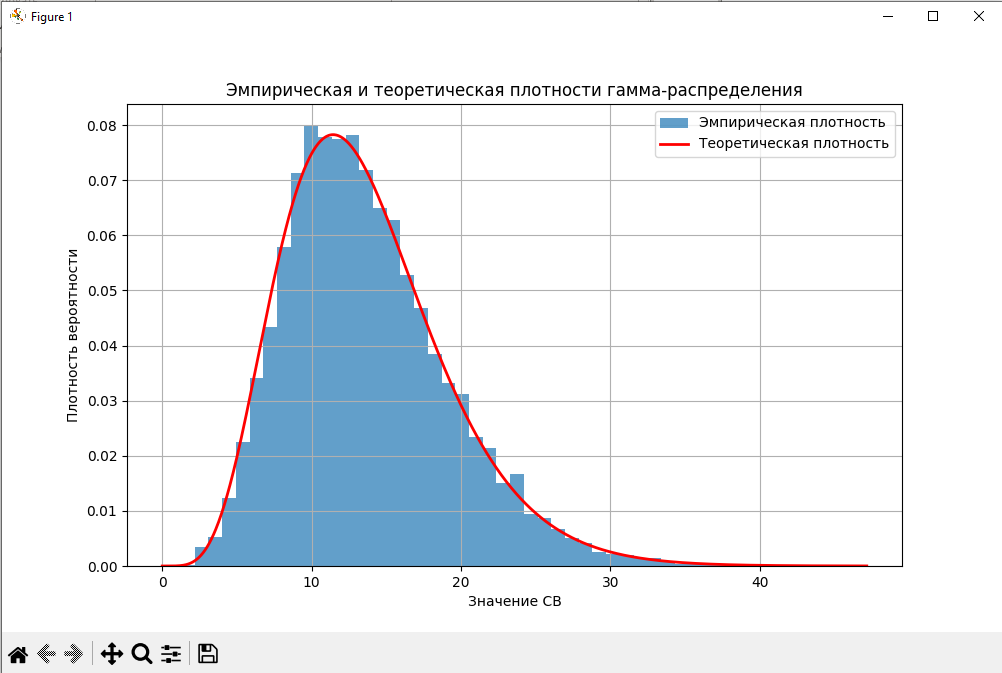


Рисунок 3.

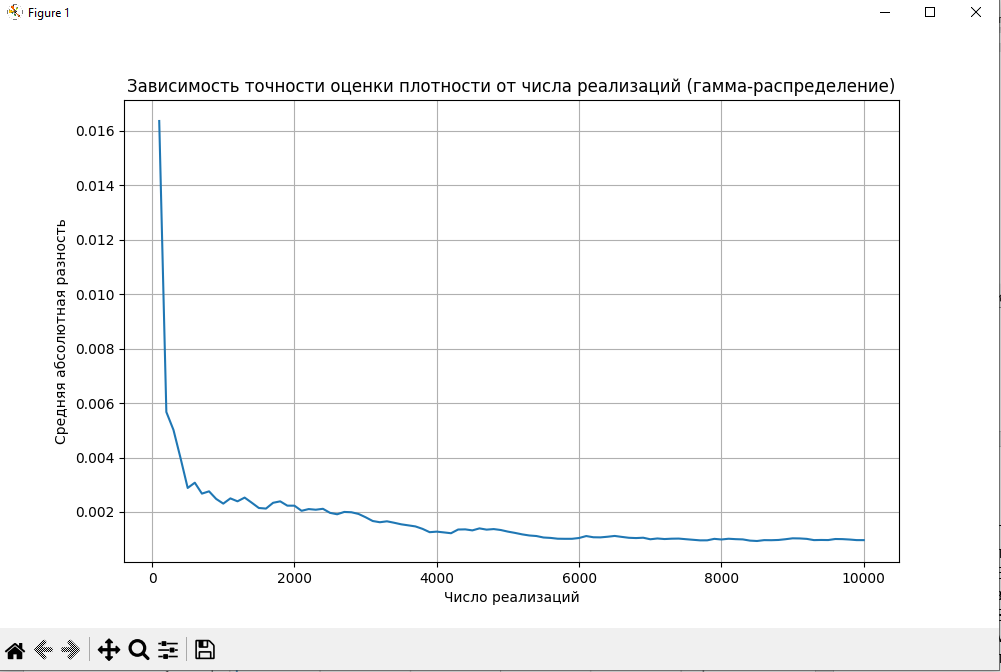
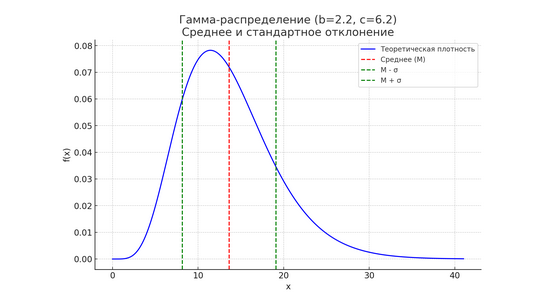


Рисунок 4.

# Ответы на контрольные вопросы

1. Если поменять параметры распределения, то изменится форма графиков: при увеличении с распределение становится более симметричным, при увеличении b — распределение растягивается вправо.
2. Если увеличить число реализаций, выборочные среднее и дисперсия будут точнее совпадать с теоретическими, а гистограмма станет ближе к теоретической плотности.
3. На графике плотности среднее отмечается вертикальной линией в центре, а отклонения ±σ — ещё двумя линиями слева и справа.



1. Гистограмма – графическое представление распределения данных, показывающее частоту попадания значений в определенные интервалы.
2. В Python **np.random.uniform(a, b, n),**  где a – нижняя граница диапазона, b – верхняя, а n – размер массива
3. Подматрицу выбирают срезами: matrix[r1:r2, c1:c2].

(Submatrix = matrix[row\_start:row\_end, col\_start:col\_end])

1. Можно использовать MSE (среднее значение квадратов разностей между прогнозируемыми и фактическими значениями), MAE (среднюю абсолютную ошибку между фактическими и прогнозируемыми значениями).
2. Перемножить матрицы: A @ B или np.dot (A, B).
3. Поэлементное умножение: A \* B.
4. **for value in vector** или **for I in range(len(vector))**
5. **plt.plot(x\_values, y\_values)**
6. **plt.hist(data, bins=50)**

# Выводы

В ходе работы было смоделировано гамма-распределение с параметрами b=2.2 с=6.2.  
Подтверждена сходимость выборочного математического ожидания и дисперсии к теоретическим значениям (μ=13.64 σ2=29.99).  
Построены графики, показывающие приближение выборочных характеристик к теоретическим при увеличении числа реализаций.  
Визуализирована эмпирическая и теоретическая плотности распределения, а также показано уменьшение ошибки оценки с ростом выборки.  
Алгоритм генерации реализован корректно, результаты эксперимента совпадают с теоретическими ожиданиями.