

**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 №17**

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

Выполнил:

Студент 3-ого курса

Учебной группы БИСО-02-22

Зубарев В.С.

Задание №1:

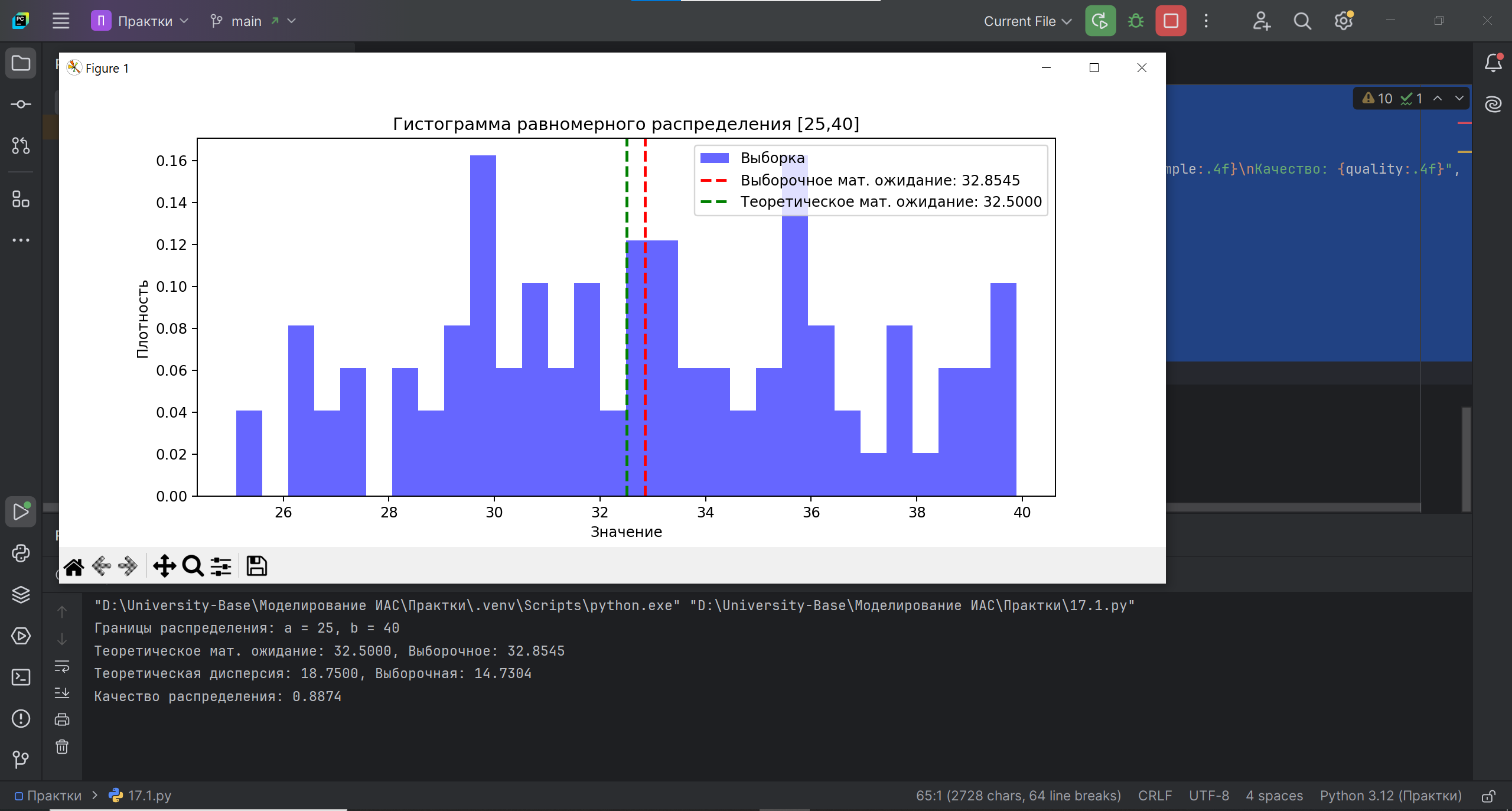
Смоделировать выборку (последовательность не менее 20-ти) равномерно распределённых случайных чисел – СЧ РР (0,1) и чисел, распределённых на отрезке [a,b], методом обратных функций.

Проверить качество полученного распределения, сравнив выборочные статистические характеристики с теоретическими параметрами.

Решение (листинг кода программы):

# Вариант 5, Зубарев  
import numpy as np  
import matplotlib.pyplot as plt  
  
  
def uniform\_inverse\_transform(size, a=0, b=1):  
 *"""Генерация равномерно распределенных чисел методом обратных функций."""* r = [np.random.random() for \_ in range(size)] # Генерация случайных чисел в [0,1]  
 return [a + (b - a) \* ri for ri in r] # Преобразование в диапазон [a,b]  
  
  
def compute\_statistics(sample):  
 *"""Вычисление выборочных статистических характеристик."""* n = len(sample)  
 mean\_sample = sum(sample) / n # Вычисление выборочного среднего  
 variance\_sample = sum((x - mean\_sample) \*\* 2 for x in sample) / n # Вычисление дисперсии  
 return mean\_sample, variance\_sample  
  
  
def evaluate\_quality(mean\_sample, mean\_theoretical, variance\_sample, variance\_theoretical):  
 *"""Оценка качества распределения (0 - плохо, 1 - идеально)."""* mean\_diff = abs(mean\_sample - mean\_theoretical) / abs(mean\_theoretical)  
 variance\_diff = abs(variance\_sample - variance\_theoretical) / abs(variance\_theoretical)  
 quality = max(0, 1 - (mean\_diff + variance\_diff) / 2)  
 return quality  
  
  
size = 100 # Размер выборки  
a, b = 25, 40 # Границы отрезка для равномерного распределения  
  
sample\_ab = uniform\_inverse\_transform(size, a, b) # [a,b]  
  
# Теоретические характеристики  
mean\_theoretical = (a + b) / 2 #Вычисление теоритического мат.ожидания  
variance\_theoretical = ((b - a) \*\* 2) / 12 # Вычисление теоритической квадратичной дисперсии  
  
# Выборочные характеристики  
mean\_sample, variance\_sample = compute\_statistics(sample\_ab)  
quality = evaluate\_quality(mean\_sample, mean\_theoretical, variance\_sample, variance\_theoretical)  
  
# Вывод результатов  
print(f"Границы распределения: a = {a}, b = {b}")  
print(f"Теоретическое мат. ожидание: {mean\_theoretical:.4f}, Выборочное: {mean\_sample:.4f}")  
print(f"Теоретическая дисперсия: {variance\_theoretical:.4f}, Выборочная: {variance\_sample:.4f}")  
print(f"Качество распределения: {quality:.4f}")  
  
# Гистограмма распределения  
plt.hist(sample\_ab, bins=30, density=True, alpha=0.6, color='b', label="Выборка")  
plt.axvline(mean\_sample, color='r', linestyle='dashed', linewidth=2,  
 label=f"Выборочное мат. ожидание: {mean\_sample:.4f}")  
plt.axvline(mean\_theoretical, color='g', linestyle='dashed', linewidth=2,  
 label=f"Теоретическое мат. ожидание: {mean\_theoretical:.4f}")  
  
# Подпись теоретической и выборочной дисперсии  
plt.text(b - 0.5, 0.8,  
 f"Теор. дисперсия: {variance\_theoretical:.4f}\nВыбор. дисперсия: {variance\_sample:.4f}\nКачество: {quality:.4f}",  
 fontsize=10, bbox=dict(facecolor='white', alpha=0.5))  
  
plt.title(f"Гистограмма равномерного распределения [{a},{b}]")  
plt.xlabel("Значение")  
plt.ylabel("Плотность")  
plt.legend()  
plt.show()

Результат работы:



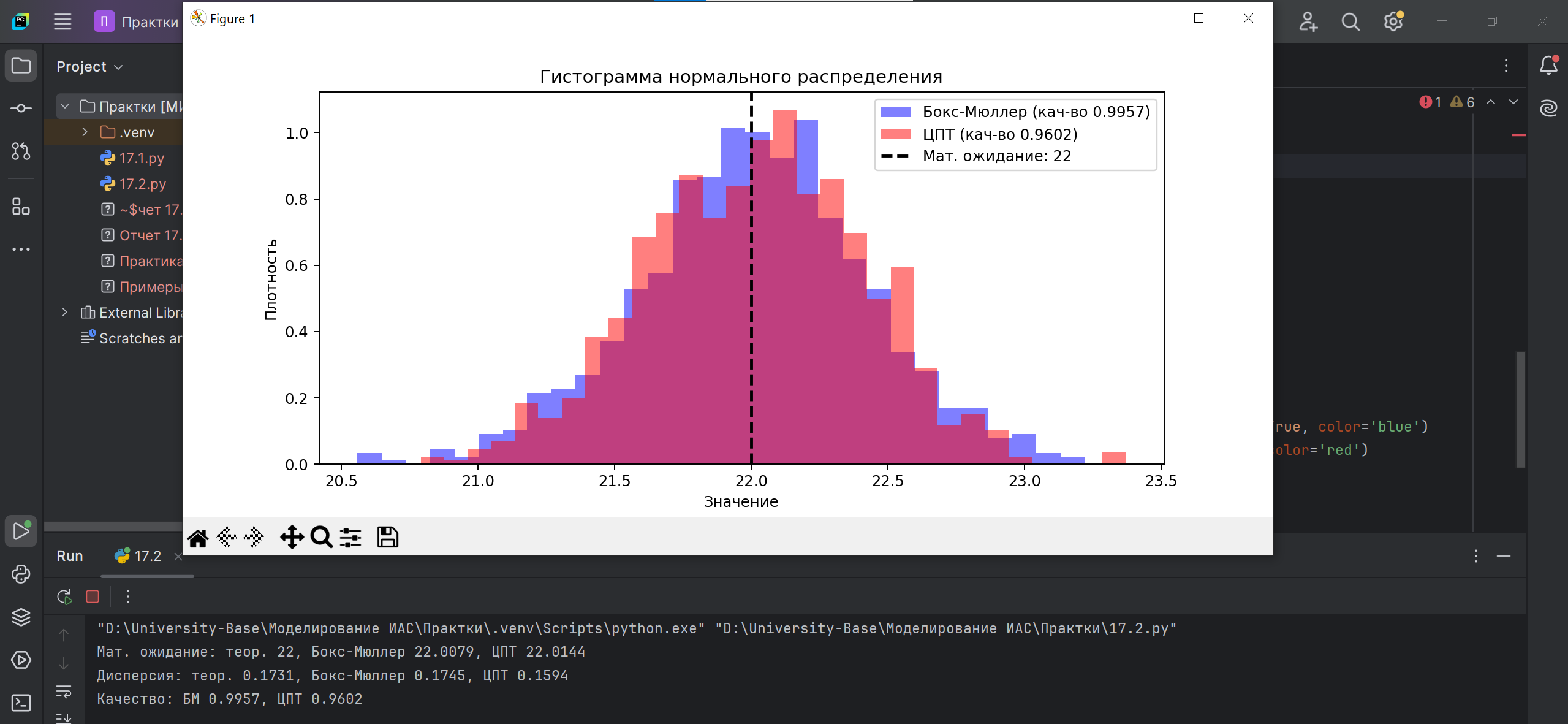
Задание №2:

Промоделировать методом ЦПТ и Бокса-Мюллера случайные числа (последовательность не менее 20-ти), распределённые непрерывно по нормальному закону, используя компьютерный ГПСЧрр(0,1) и заданные значения mx и σx (значения выбираем произвольно). Сравнить выборочные статистические характеристики с теоретическими характеристиками распределения.

Решение (листинг кода программы):

# Вариант 5, Зубарев  
import random  
import math  
import matplotlib.pyplot as plt  
  
def box\_muller(size, mx, dx):  
 *"""Генерация нормальных случайных чисел методом Бокса-Мюллера."""* numbers = []  
 for \_ in range(size):  
 u1, u2 = random.random(), random.random()  
 z = math.sqrt(-2 \* math.log(u1)) \* math.cos(2 \* math.pi \* u2)  
 numbers.append(mx + dx \* z)  
 return numbers  
  
def clt(size, mx, dx):  
 *"""Генерация нормальных случайных чисел методом ЦПТ."""* numbers = []  
 for \_ in range(size):  
 sum\_uniform = sum(random.random() for \_ in range(12)) - 6 # Центрируем сумму  
 numbers.append(mx + dx \* sum\_uniform / math.sqrt(12 / 12))  
 return numbers  
  
def compute\_stats(data):  
 *"""Вычисление среднего и дисперсии."""* mean\_sample = sum(data) / len(data)  
 var\_sample = sum((x - mean\_sample) \*\* 2 for x in data) / len(data)  
 return mean\_sample, var\_sample  
  
def quality\_score(mean\_sample, mean\_true, var\_sample, var\_true):  
 *"""Оценка качества (0 - плохо, 1 - идеально)."""* mean\_err = abs(mean\_sample - mean\_true) / abs(mean\_true)  
 var\_err = abs(var\_sample - var\_true) / abs(var\_true)  
 return max(0, 1 - (mean\_err + var\_err) / 2)  
  
size = 1000  
mx, dx = 22, 0.416  
  
sample\_bm = box\_muller(size, mx, dx)  
sample\_clt = clt(size, mx, dx)  
  
mean\_bm, var\_bm = compute\_stats(sample\_bm)  
mean\_clt, var\_clt = compute\_stats(sample\_clt)  
  
var\_theoretical = dx \*\* 2  
quality\_bm = quality\_score(mean\_bm, mx, var\_bm, var\_theoretical)  
quality\_clt = quality\_score(mean\_clt, mx, var\_clt, var\_theoretical)  
  
print(f"Мат. ожидание: теор. {mx}, Бокс-Мюллер {mean\_bm:.4f}, ЦПТ {mean\_clt:.4f}")  
print(f"Дисперсия: теор. {var\_theoretical:.4f}, Бокс-Мюллер {var\_bm:.4f}, ЦПТ {var\_clt:.4f}")  
print(f"Качество: БМ {quality\_bm:.4f}, ЦПТ {quality\_clt:.4f}")  
  
plt.hist(sample\_bm, bins=30, alpha=0.5, label=f"Бокс-Мюллер (кач-во {quality\_bm:.4f})", density=True, color='blue')  
plt.hist(sample\_clt, bins=30, alpha=0.5, label=f"ЦПТ (кач-во {quality\_clt:.4f})", density=True, color='red')  
plt.axvline(mx, color='black', linestyle='dashed', linewidth=2, label=f"Мат. ожидание: {mx}")  
  
plt.title("Гистограмма нормального распределения")  
plt.xlabel("Значение")  
plt.ylabel("Плотность")  
plt.legend()  
plt.show()

Результат работы:



Задание №3:

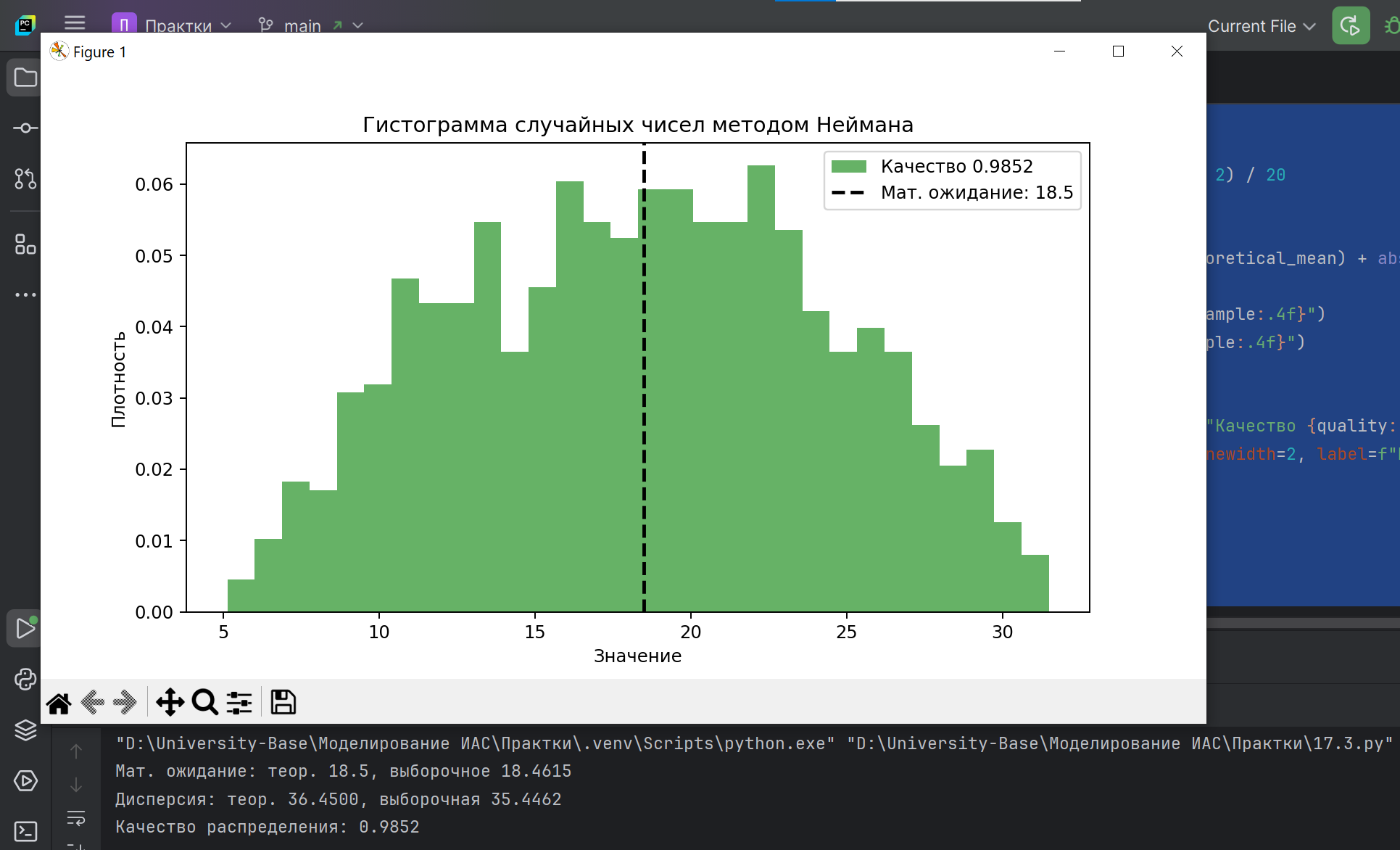
Смоделировать методом усечения – Неймана случайные числа (последовательность не менее 20-ти) по непрерывному закону распределения, заданного в ограниченной области y ∈ [a,b] плотностью распределения.

Сравнить выборочные статистические характеристики с теоретическими параметрами закона распределения.

Решение (листинг кода программы)

#Вариант 5, Зубарев  
import random  
import math  
import matplotlib.pyplot as plt  
  
a, b = 5, 32  
m = (a + b) / 2  
  
size = 1000  
data = []  
  
while len(data) < size:  
 y = random.uniform(a, b)  
 u = random.uniform(0, 1)  
 f\_y = (6 / (4 \* (b - a))) \* (1 - 4 \* ((y - m) / (b - a)) \*\* 2)  
 if u <= f\_y:  
 data.append(y)  
  
mean\_sample = sum(data) / len(data)  
squared\_mean\_sample = sum(x \*\* 2 for x in data) / len(data)  
var\_sample = squared\_mean\_sample - mean\_sample \*\* 2  
  
theoretical\_mean = m  
theoretical\_squared\_mean = (20 \* theoretical\_mean \*\* 2 + (b - a) \*\* 2) / 20  
theoretical\_var = theoretical\_squared\_mean - theoretical\_mean \*\* 2  
  
quality = max(0, 1 - (abs(mean\_sample - theoretical\_mean) / abs(theoretical\_mean) + abs(var\_sample - theoretical\_var) / abs(theoretical\_var)) / 2)  
  
print(f"Мат. ожидание: теор. {theoretical\_mean}, выборочное {mean\_sample:.4f}")  
print(f"Дисперсия: теор. {theoretical\_var:.4f}, выборочная {var\_sample:.4f}")  
print(f"Качество распределения: {quality:.4f}")  
  
plt.hist(data, bins=30, density=True, alpha=0.6, color='g', label=f"Качество {quality:.4f}")  
plt.axvline(theoretical\_mean, color='black', linestyle='dashed', linewidth=2, label=f"Мат. ожидание: {theoretical\_mean}")  
plt.title("Гистограмма случайных чисел методом Неймана")  
plt.xlabel("Значение")  
plt.ylabel("Плотность")  
plt.legend()  
plt.show()

Результат работы:



Задание №4:

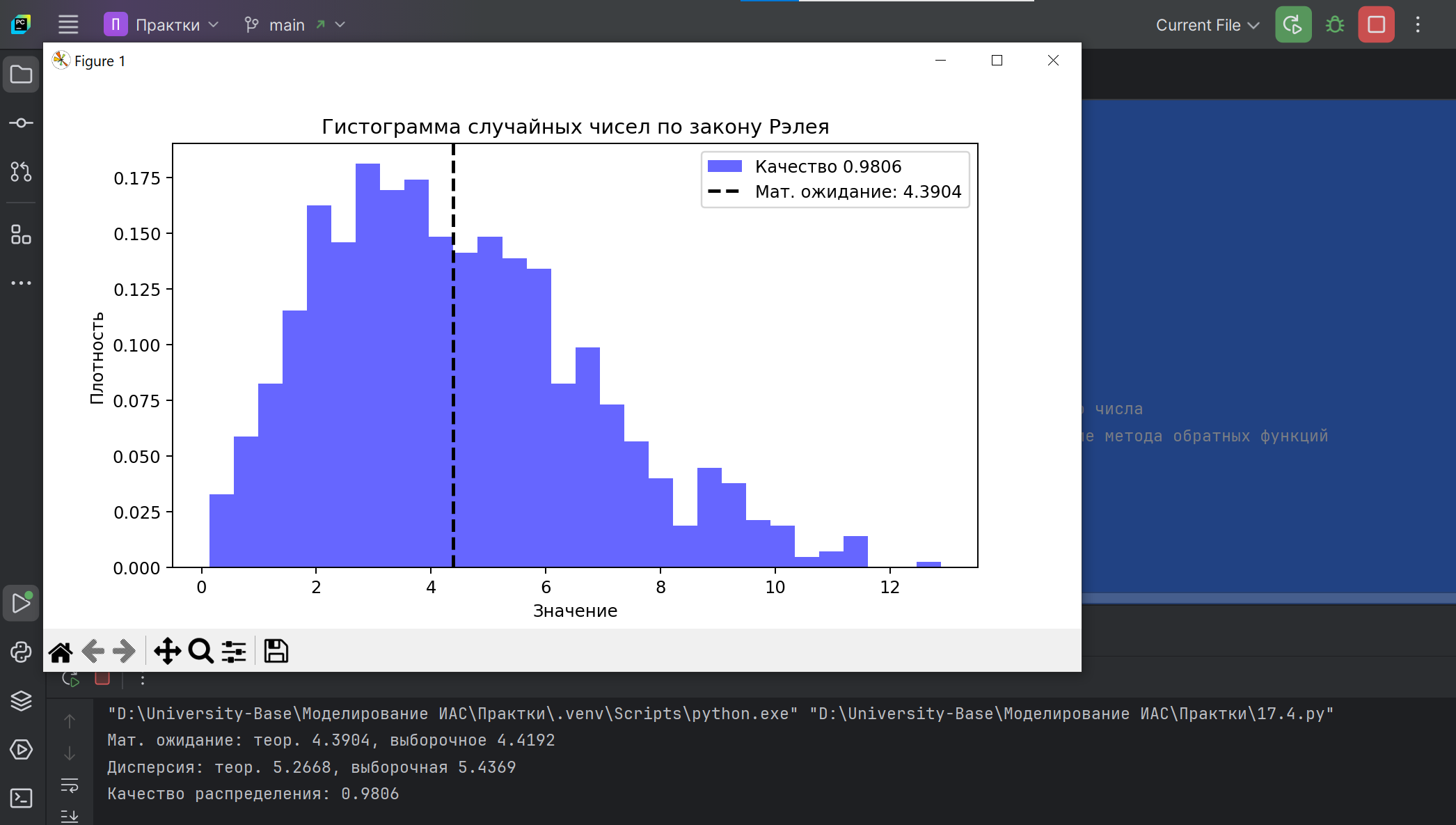
Смоделировать методом обратных функций случайные числа (последовательность не менее 20-ти), распределённые непрерывно по закону Рэлея.

Сравнить выборочные статистические характеристики с теоретическими характеристиками распределения.

Решение (листинг кода программы):

#Вариант 5, Зубарев  
import random  
import math  
import matplotlib.pyplot as plt  
  
sigma = 3.503 # Параметр распределения Рэлея  
size = 1000  
data = []  
  
# Генерация случайных чисел методом обратных функций  
for \_ in range(size):  
 u = random.random() # Генерация равномерного случайного числа  
 y = sigma \* math.sqrt(-2 \* math.log(1 - u)) # Применение метода обратных функций  
 data.append(y)  
  
# Вычисление выборочного математического ожидания  
mean\_sample = sum(data) / len(data)  
  
# Вычисление выборочного среднего квадрата  
squared\_mean\_sample = sum(x \*\* 2 for x in data) / len(data)  
  
# Вычисление выборочной дисперсии  
var\_sample = squared\_mean\_sample - mean\_sample \*\* 2  
  
# Теоретическое математическое ожидание  
theoretical\_mean = sigma \* math.sqrt(math.pi / 2)  
  
# Теоретическая дисперсия  
theoretical\_var = (2 - math.pi / 2) \* sigma \*\* 2  
  
# Оценка качества распределения  
quality = max(0, 1 - (abs(mean\_sample - theoretical\_mean) / abs(theoretical\_mean) + abs(var\_sample - theoretical\_var) / abs(theoretical\_var)) / 2)  
  
# Вывод результатов  
print(f"Мат. ожидание: теор. {theoretical\_mean:.4f}, выборочное {mean\_sample:.4f}")  
print(f"Дисперсия: теор. {theoretical\_var:.4f}, выборочная {var\_sample:.4f}")  
print(f"Качество распределения: {quality:.4f}")  
  
# Построение гистограммы  
plt.hist(data, bins=30, density=True, alpha=0.6, color='b', label=f"Качество {quality:.4f}")  
plt.axvline(theoretical\_mean, color='black', linestyle='dashed', linewidth=2, label=f"Мат. ожидание: {theoretical\_mean:.4f}")  
plt.title("Гистограмма случайных чисел по закону Рэлея")  
plt.xlabel("Значение")  
plt.ylabel("Плотность")  
plt.legend()  
plt.show()

Результат работы:



Задание №5:

Смоделировать методом «розыгрыша жребия» случайные числа (последовательность не менее 20-ти), распределённые по дискретному геометрическому закону.

Вычислить выборочные характеристики и сравнить с теоретическими.

Решение (листинг кода программы):

#Вариант 5, Зубарев  
import random  
import math  
import matplotlib.pyplot as plt  
  
p = 0.595 # Параметр геометрического распределения  
q = 1 - p  
size = 1000  
data = []  
  
# Генерация случайных чисел методом "розыгрыша жребия"  
for \_ in range(size):  
 u = random.random() # Генерация равномерного случайного числа  
 k = 1 # Начальное значение k  
 cumulative\_prob = p # Начальная суммарная вероятность  
 while u > cumulative\_prob:  
 k += 1  
 cumulative\_prob += (q \*\* (k - 1)) \* p # Вычисление границ отрезка для метода жребия  
 data.append(k)  
  
# Вычисление выборочного математического ожидания  
mean\_sample = sum(data) / len(data)  
  
# Вычисление выборочного среднего квадрата  
squared\_mean\_sample = sum(x \*\* 2 for x in data) / len(data)  
  
# Вычисление выборочной дисперсии  
var\_sample = squared\_mean\_sample - mean\_sample \*\* 2  
  
# Теоретическое математическое ожидание  
theoretical\_mean = 1 / p  
  
# Теоретическая дисперсия  
theoretical\_var = q / (p \*\* 2)  
  
# Оценка качества распределения  
quality = max(0, 1 - (abs(mean\_sample - theoretical\_mean) / abs(theoretical\_mean) + abs(var\_sample - theoretical\_var) / abs(theoretical\_var)) / 2)  
  
# Вывод результатов  
print(f"Мат. ожидание: теор. {theoretical\_mean:.4f}, выборочное {mean\_sample:.4f}")  
print(f"Дисперсия: теор. {theoretical\_var:.4f}, выборочная {var\_sample:.4f}")  
print(f"Качество распределения: {quality:.4f}")  
  
# Построение гистограммы  
plt.hist(data, bins=30, density=True, alpha=0.6, color='b', label=f"Качество {quality:.4f}")  
plt.axvline(theoretical\_mean, color='black', linestyle='dashed', linewidth=2, label=f"Мат. ожидание: {theoretical\_mean:.4f}")  
plt.title("Гистограмма случайных чисел по геометрическому распределению")  
plt.xlabel("Значение")  
plt.ylabel("Плотность")  
plt.legend()  
plt.show()

Результат работы:

