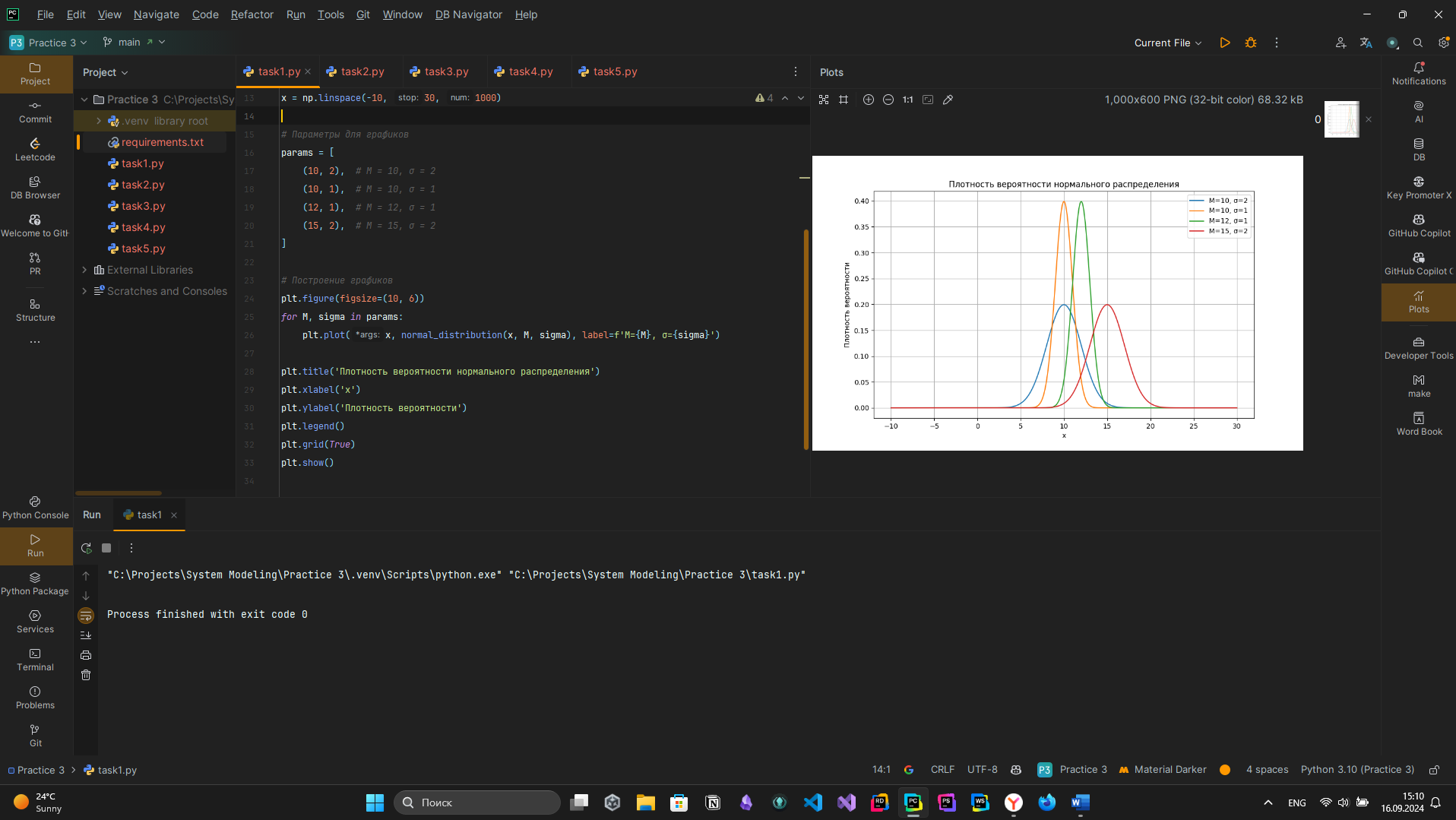
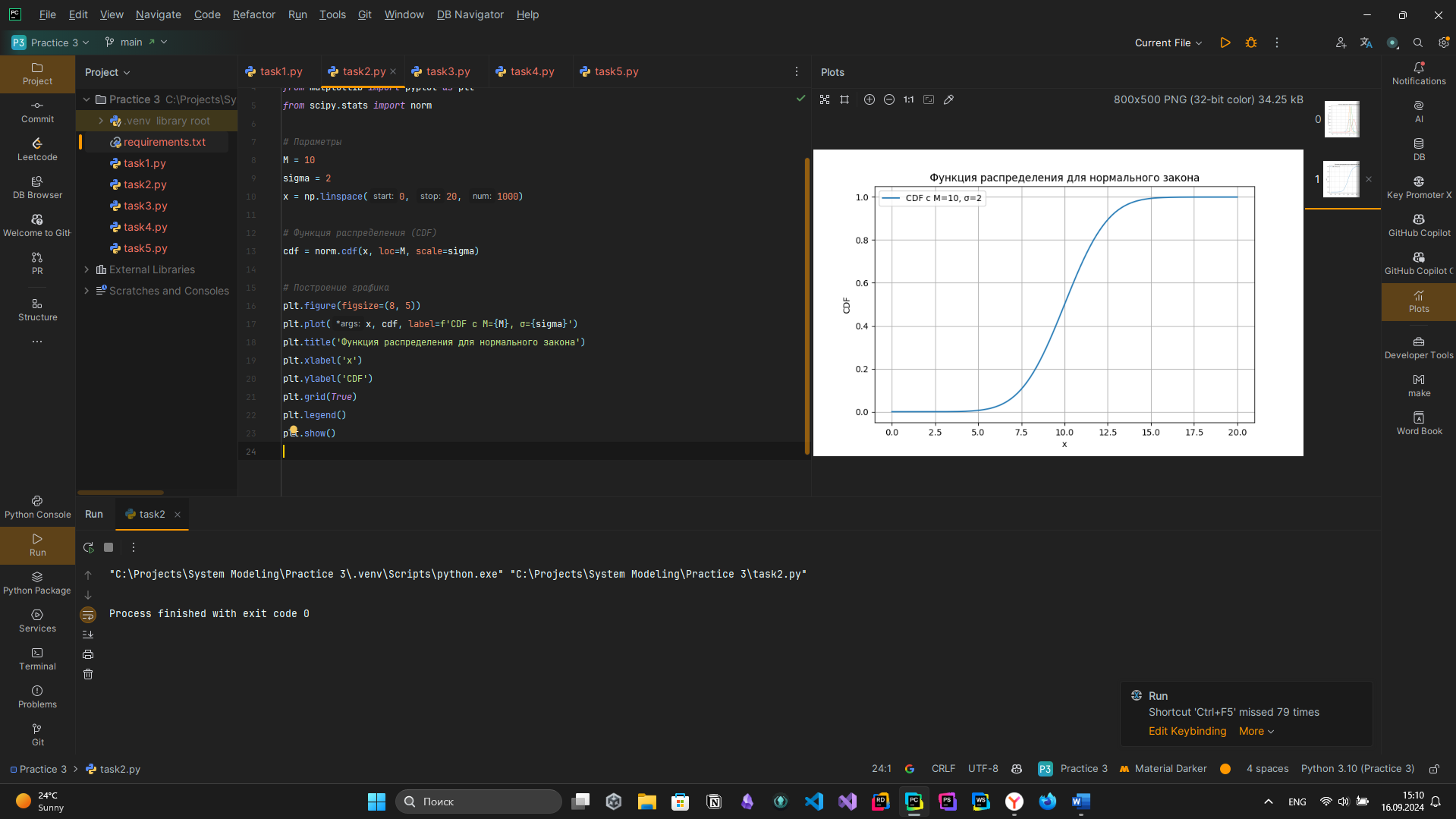
Практика 3

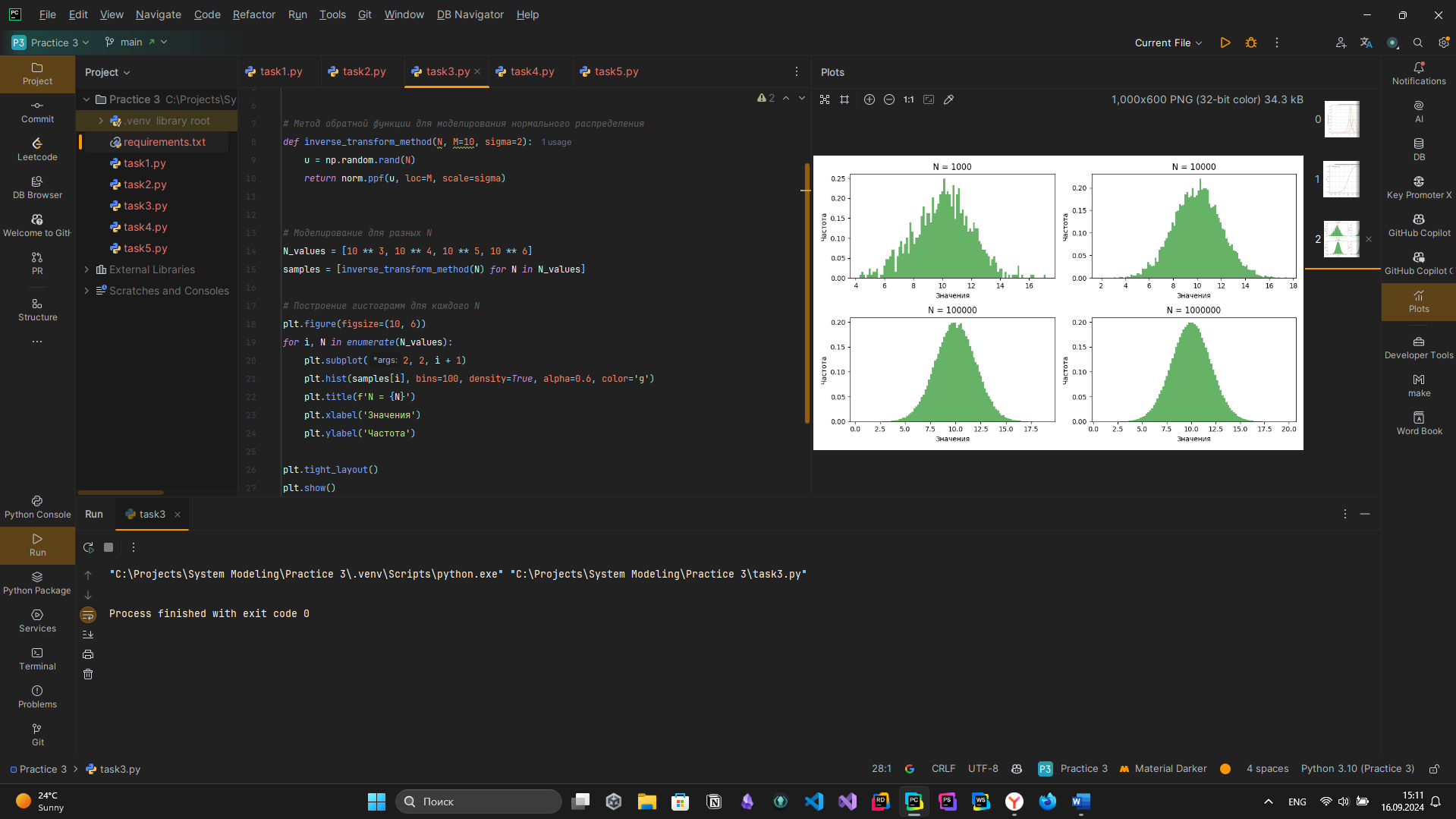
БСБО-09-23

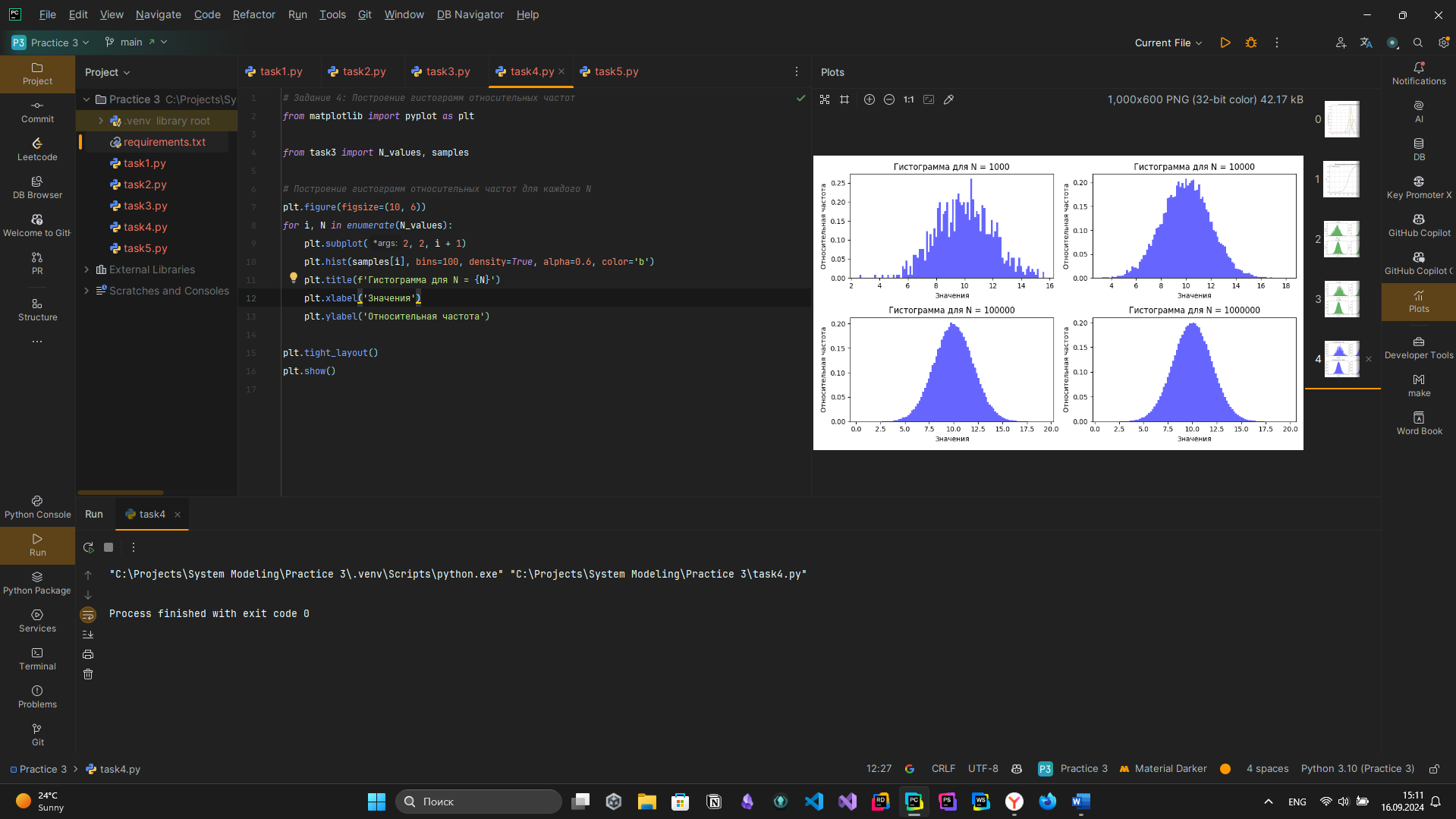
Шутов Кирилл Сергеевич

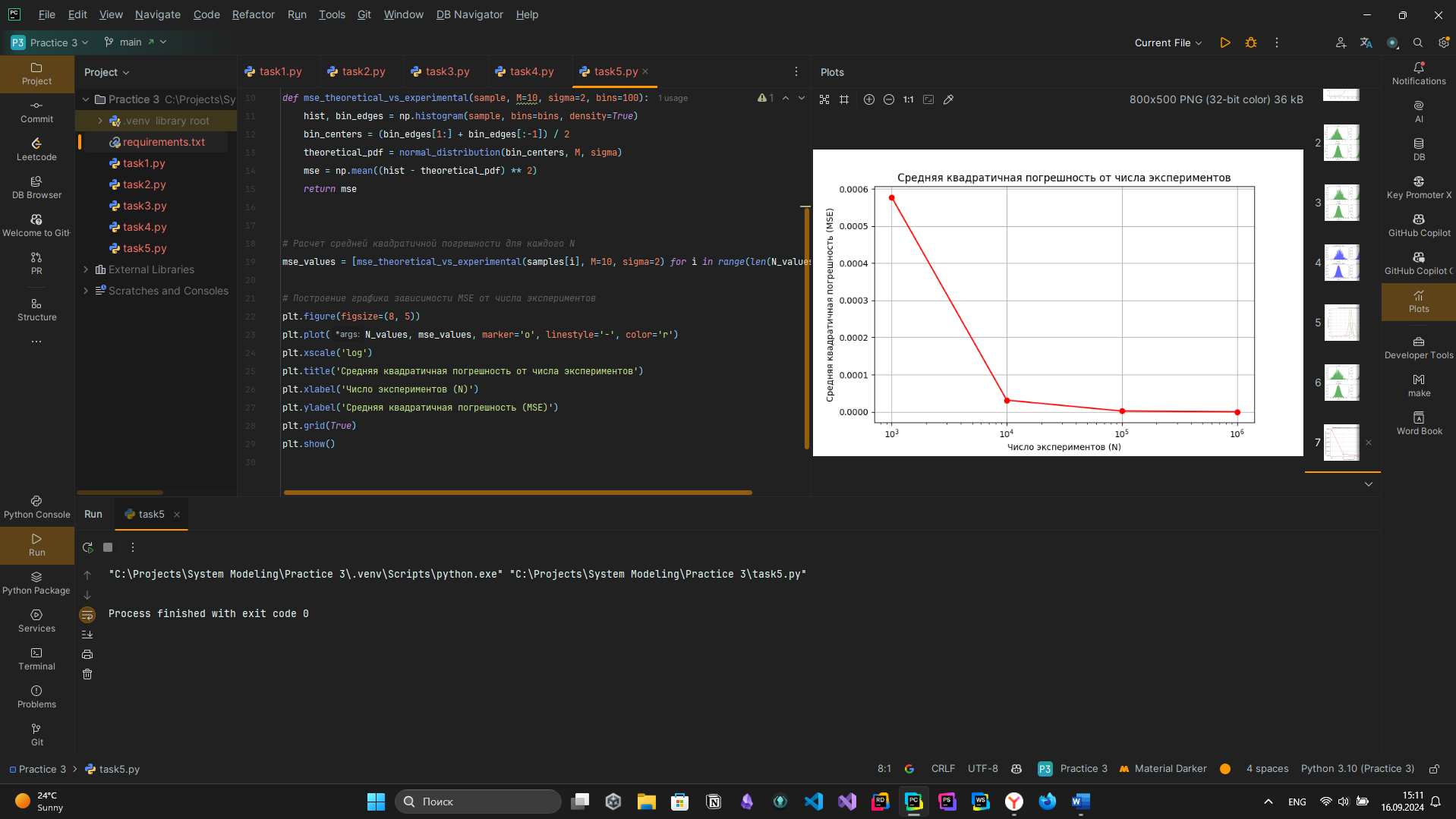
Пример работы











Листинг кода

|  |
| --- |
| *# Задание 1: Плотность вероятности нормального распределения* **import** numpy **as** np **import** matplotlib.pyplot **as** plt   *# Функция плотности вероятности нормального распределения* **def** normal\_distribution(x, M, sigma):  **return** (1 / (sigma \* np.sqrt(2 \* np.pi))) \* np.exp(-((x - M) \*\* 2) / (2 \* sigma \*\* 2))   *# Параметры* x = np.linspace(-10, 30, 1000)  *# Параметры для графиков* params = [  (10, 2), *# M = 10, σ = 2* (10, 1), *# M = 10, σ = 1* (12, 1), *# M = 12, σ = 1* (15, 2), *# M = 15, σ = 2* ]  *# Построение графиков* plt.figure(figsize=(10, 6)) **for** M, sigma **in** params:  plt.plot(x, normal\_distribution(x, M, sigma), label=**f'M={**M**}, σ={**sigma**}'**)  plt.title(**'Плотность вероятности нормального распределения'**) plt.xlabel(**'x'**) plt.ylabel(**'Плотность вероятности'**) plt.legend() plt.grid(**True**) plt.show() |
| *# Задание 2: Функция распределения (CDF) для нормального закона* **import** numpy **as** np **from** matplotlib **import** pyplot **as** plt **from** scipy.stats **import** norm  *# Параметры* M = 10 sigma = 2 x = np.linspace(0, 20, 1000)  *# Функция распределения (CDF)* cdf = norm.cdf(x, loc=M, scale=sigma)  *# Построение графика* plt.figure(figsize=(8, 5)) plt.plot(x, cdf, label=**f'CDF с M={**M**}, σ={**sigma**}'**) plt.title(**'Функция распределения для нормального закона'**) plt.xlabel(**'x'**) plt.ylabel(**'CDF'**) plt.grid(**True**) plt.legend() plt.show() |
| *# Задание 3: Моделирование нормального распределения методом обратной функции* **import** numpy **as** np **from** matplotlib **import** pyplot **as** plt **from** scipy.stats **import** norm   *# Метод обратной функции для моделирования нормального распределения* **def** inverse\_transform\_method(N, M=10, sigma=2):  u = np.random.rand(N)  **return** norm.ppf(u, loc=M, scale=sigma)   *# Моделирование для разных N* N\_values = [10 \*\* 3, 10 \*\* 4, 10 \*\* 5, 10 \*\* 6] samples = [inverse\_transform\_method(N) **for** N **in** N\_values]  *# Построение гистограмм для каждого N* plt.figure(figsize=(10, 6)) **for** i, N **in** enumerate(N\_values):  plt.subplot(2, 2, i + 1)  plt.hist(samples[i], bins=100, density=**True**, alpha=0.6, color=**'g'**)  plt.title(**f'N = {**N**}'**)  plt.xlabel(**'Значения'**)  plt.ylabel(**'Частота'**)  plt.tight\_layout() plt.show() |
| *# Задание 4: Построение гистограмм относительных частот* **from** matplotlib **import** pyplot **as** plt  **from** task3 **import** N\_values, samples  *# Построение гистограмм относительных частот для каждого N* plt.figure(figsize=(10, 6)) **for** i, N **in** enumerate(N\_values):  plt.subplot(2, 2, i + 1)  plt.hist(samples[i], bins=100, density=**True**, alpha=0.6, color=**'b'**)  plt.title(**f'Гистограмма для N = {**N**}'**)  plt.xlabel(**'Значения'**)  plt.ylabel(**'Относительная частота'**)  plt.tight\_layout() plt.show() |
| *# Задание 5: Средняя квадратичная погрешность* **import** numpy **as** np **from** matplotlib **import** pyplot **as** plt  **from** task1 **import** normal\_distribution **from** task3 **import** samples, N\_values   *# Функция для вычисления средней квадратичной погрешности* **def** mse\_theoretical\_vs\_experimental(sample, M=10, sigma=2, bins=100):  hist, bin\_edges = np.histogram(sample, bins=bins, density=**True**)  bin\_centers = (bin\_edges[1:] + bin\_edges[:-1]) / 2  theoretical\_pdf = normal\_distribution(bin\_centers, M, sigma)  mse = np.mean((hist - theoretical\_pdf) \*\* 2)  **return** mse   *# Расчет средней квадратичной погрешности для каждого N* mse\_values = [mse\_theoretical\_vs\_experimental(samples[i], M=10, sigma=2) **for** i **in** range(len(N\_values))]  *# Построение графика зависимости MSE от числа экспериментов* plt.figure(figsize=(8, 5)) plt.plot(N\_values, mse\_values, marker=**'o'**, linestyle=**'-'**, color=**'r'**) plt.xscale(**'log'**) plt.title(**'Средняя квадратичная погрешность от числа экспериментов'**) plt.xlabel(**'Число экспериментов (N)'**) plt.ylabel(**'Средняя квадратичная погрешность (MSE)'**) plt.grid(**True**) plt.show() |