Дипломная работа по теме:

Сравнение различных библиотек для визуализации данных: Matplotlib, Seaborn и Plotly.  
Автор: Шляхова София Владимировна

**Содержание:  
1. Введение  
 Обоснование выбора темы  
 Определение цели и задач исследования  
2. Основные понятия и определения**3. Методы и подходы к разработкеВыбор данных и диаграмм для сравненияНаписание кодаПроверка кода4. Обзор инструментов (библиотек) для визуализации **Описание библиотек  
5.Проектирование  
 Выбор данных и диаграмм для сравнения  
 Основные требования  
 Технические требования  
6. Разработка в соответствии с созданной документацией  
 Планирование разработки** Разработка **Программа 1 (Круговая диаграмма)  
 Программа 2 (График функции)  
 Программа 3 (Диаграмма размаха)  
 Программа 4 (Столбчатая диаграмма)  
 Программа 5 (Диаграмма рассеяния)  
7. Анализ и интерпретация результатов  
 Анализ библиотек  
 Интерпретация результатов**

**Рекомендации по выбору библиотек в зависимости от проекта**

**8. Заключение**

**Обзор выполненной работы**

**Дальнейшие планы**

**1. Введение**

**Обоснование выбора темы:**

**1.Польза в современном мире:**

**В наше время с учетом огромного количества статистических данных, используемых в сферах маркетинга, финансов, политики, образования, торговли и т.д. И визуализация самый легкий способ упрощения и понимания связей разных данных друг с другом, а также, помогает четко видеть изменения**

**2.Практическая польза:**

**Во время учебы и работы ведение статистики очень важно для точного планирования и в последствии эффективного исполнения задач. Также визуализация может помочь в разных научных направлениях (физика, математика, химия и т.д).**

**3.Личный интерес:**

**Я очень люблю статистику и таблицы, тк очень часто сама пользуюсь большими объемами данных в своей жизни. Также визуализация данных ныне очень востребована в современном мире и умение работать с этими библиотеками может помочь при трудоустройстве.**

**Таким образом выбор темы «Сравнение различных библиотек для визуализации данных: Matplotlib, Seaborn и Plotly» обусловлено её актуальностью, востребованностью на рынке, практической значимостью, моей личной пользой и интересом**, **что делает эту тему подходящей для проведения дипломной работы.**

**Определение цели и задач исследования:**

Цель исследования:

Создать набор визуализаций с использованием Matplotlib, Seaborn и Plotly, сравнить их функциональность и удобство использования.

Задачи исследования:

1.Обзор различных библиотек для визуализации данных: Провести анализ существующих инструментов для визуализации, выбрать основные типы визуализации.

2. Разработка сценариев сравнения: Разработать алгоритм сравнения эффективности и наглядности использования библиотек визуализации, определить критерии сравнения и набрать минимальный набор данных для первичной проверки.

3. Создание программ и проведение тестирования: реализовать модели из разных видов визуализации в библиотеках (Matplotlib, Seaborn и Plotly), и оценить эффективность различных моделей, провести их тестирование.

4. Написание дипломной работы: Составить дипломную работу, включающую в себя введение, обзор литературы, методологию и результаты исследования, анализ результатов, выводы и рекомендации.

Цели и задачи помогут получить практические результаты и сравнить их, для понимания разницы между разными библиотеками для визуализации данных.

2. Основные понятия и определения

1. Визуализация данных — это представление данных в виде, который обеспечивает наиболее эффективную работу человека по их изучению.
2. График функции — геометрическое понятие в математике, дающее представление о геометрическом образе функции.
3. Диаграмма рассеяния — математическая диаграмма, изображающая значения двух переменных в виде точек на декартовой плоскости.
4. Диаграмма — графическое представление данных линейными отрезками или геометрическими фигурами, позволяющее быстро оценить соотношение нескольких величин.
5. Линейный график — это тип визуализации данных, представленный точками, соединёнными отрезками прямых линий.
6. Гистограмма — способ представления табличных данных в графическом виде — в виде столбчатой диаграммы.
7. Столбчатая диаграмма — диаграмма, представленная прямоугольными зонами, высоты или длины которых пропорциональны величинам, которые они отображают.
8. Кривая — непрерывная линия, которая не является прямой.
9. Функция — соответствие между двумя множествами, при котором каждому элементу одного множества соответствует единственный элемент другого.
10. Данные — формы представления информации, с которыми имеют дело информационные системы и их пользователи.
11. Круговая диаграмма — это круговая статистическая диаграмма, которая разделена на срезы для иллюстрации числовой пропорции.
12. Библиотека — набор файлов, модулей, классов, функций, в которых реализован весь функционал решения конкретной проблемы.
13. Ящик с усами, диаграмма размаха — график, использующийся в описательной статистике, компактно изображающий одномерное распределение вероятностей.

3. Методы и подходы к разработке

Выбор данных и диаграмм для сравнения:

В первую очередь нужно выбрать основные (5-6) графики и диаграммы, чтобы сравнить то как точно и понятно будут визуализированы данные. Главное, чтобы графики были во всех библиотеках. Т.к. неправильно будет сравнивать разные виды визуализации, для сравнения разных библиотек.

К примеру:

В Plotly есть анимационные и 3D модели, а в Seaborn есть только 2D модели, и если использовать 3D модель для Plotly и 2D модель для Seaborn их нельзя будет сравнивать.

После выбора нужно выбрать данные которые будут использованы при построении диаграмм. Данные должны состоять из небольшого количества пунктов (т.е. не более 15, т.к. тогда будет дольше и сложнее их использовать при проверке)

Написание кода:

Далее требуется написать программы, дополнительного ПО для графики не требуется, но требуется установка библиотек numpy и pandas. Сначала их нужно скачать через команду “pip install”.

Самое главное не делать код очень сложным т.к. требуется сравнить скорость и эффективность, а если программы будут написаны по-разному (имеется в виду большая разница или сильное упрощение/усложнение кода, сильно влияющие на скорость выполнения) чистое и правильное сравнение будет невозможно.

Проверка кода:

В проверке важнее всего не перепутать данные. А после получения результата их нужно очень подробно сравнить, после чего сравнить все отличия, а позже проанализировать.

4. Обзор инструментов (библиотек) для визуализации

Основные библиотеки — Matplotlib, Seaborn и Plotly.

Вспомогательные библиотеки (могут помочь с расчетами) — numpy и pandas.

Описание библиотек:

1.Matplotlib — библиотека на языке программирования Python для визуализации данных двумерной и трёхмерной графикой.

Основные возможности:

* Можно рассматривать графики и как одно целое и отдельно, частями, для точечного изменения графика
* Очень хорошо работает с numpy
* Matplotlib лежит в основе других библиотек, например, Seaborn
* У Matplotlib самый высокоуровневый (объектно-ориентированный) интерфейс с набором команд и функций
* Имеет возможность построить двумерные и трехмерные графики

Особенности:

* Большое количество графиков на выбор (двумерные и трехмерные)
* Объектно-ориентированный интерфейс

2. Seaborn — это библиотека для создания графиков на Python. Она основывается на matplotlib и взаимодействует со структурами данных pandas.

Основные возможности:

* Архитектура Seaborn позволяет быстро изучить и понять свои данные, довольно простой в понимании, позволяет быстро и эффективно работать с данными
* Поддерживает только двухмерную графику, и не предназначена для трехмерной графики и интерактивных графиков
* Имеет огромное количество видов графиков

Особенности:

* Основана на библиотеке matplotlib
* Есть только двумерные графики

3. Plotly — библиотека графического дизайна с открытым исходным кодом для Python

Основные возможности:

* Архитектура Plotly позволяет создавать отдельные, пользовательские элементы управления
* Поддерживает двухмерную и трехмерную графику, интерактивные графики, анимации
* Поддерживает виджеты Jupyter
* Является библиотекой с открытым кодом, и в случае ошибок можно их изменить для всех пользователей разом (т.е. предложить изменения)

Особенности:

* Очень открыта для изменений, взаимодействует с разными библиотеками и программами
* Самая вариативная в плане видов графиков и способов визуализации из представленных

4. NumPy — библиотека с открытым исходным кодом для языка программирования Python.

Основные возможности:

* Поддержка многомерных массивов (включая матрицы) Поддерживает только двухмерную графику.
* Поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами

Особенности:

* Для решения выражений из линейной алгебры использует код библиотеки LAPACK
* Позволяет математические операции с массивами

5. Pandas — программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня.

Основные возможности:

* Позволяет удалять, добавлять, соединять, срезать по индексу и изменять данные/наборы данных
* Работа с временными рядами: формирование временных периодов, изменение интервалов и т.д.
* Позволяет переформировывать данные и создавать таблицы
* Может манипулировать индексами двумерных массивов

Особенности:

* Есть открытый исходный код
* Нужна для обобщения информации, собранной об объекте исследования

5.Проектирование

Выбор данных и диаграмм для сравнения:

В первую очередь нужно выбрать основные (5-6) графики и диаграммы, чтобы сравнить то как точно и понятно будут визуализированы данные.

К примеру:

1. Круговая диаграмма
2. График функции (с математическим выражением) \*
3. Диаграмма размаха
4. Столбчатая диаграмма
5. Диаграмма рассеяния

Они являются основными для большинства сфер где есть потребность в визуализации, а также являются самыми популярными в непрофессиональной среде.

После выбора нужно выбрать данные которые будут использованы при построении диаграмм. Я возьму данные прироста населения города Красноярск в зависимости от года, для столбчатой диаграммы. Данные дохода по сферам, для круговой диаграммы. Данные по баллам за предмет математика на ЕГЭ, для диаграммы рассеяния. Данные о оценках по информатике при сдаче ОГЭ, для диаграммы размаха.

Для 2 пункта, использую функцию f(x)=2\*\*x – 2\*\*(x-1)

Основные требования:

1. Корректность написанных программ: Все программы должны быть написаны верно, без логических и синтаксических ошибок
2. Верные данные: Данные представляющие один из видов визуализации данных должны быть одинаковы (т.е. если представлена круговая диаграмма, то там должны быть данные для неё, а не данные для графика или столбчатой диаграммы. Иначе сравнение будет невозможно, из-за того, что данные друг от друга отличаются).
3. Результаты: Программа должна выдавать корректный и видный результат, примерно одинаковый у разных библиотек.

Технические требования:

Библиотеки: Matplotlib, Seaborn, Plotly, numpy и pandas.

Язык программирования: Python 3.11

Работа ведется через Pycharm, т.к. он является самой удобной для визуальных проектов средой разработки для Python.

6. Разработка в соответствии с созданной документацией

Планирование разработки

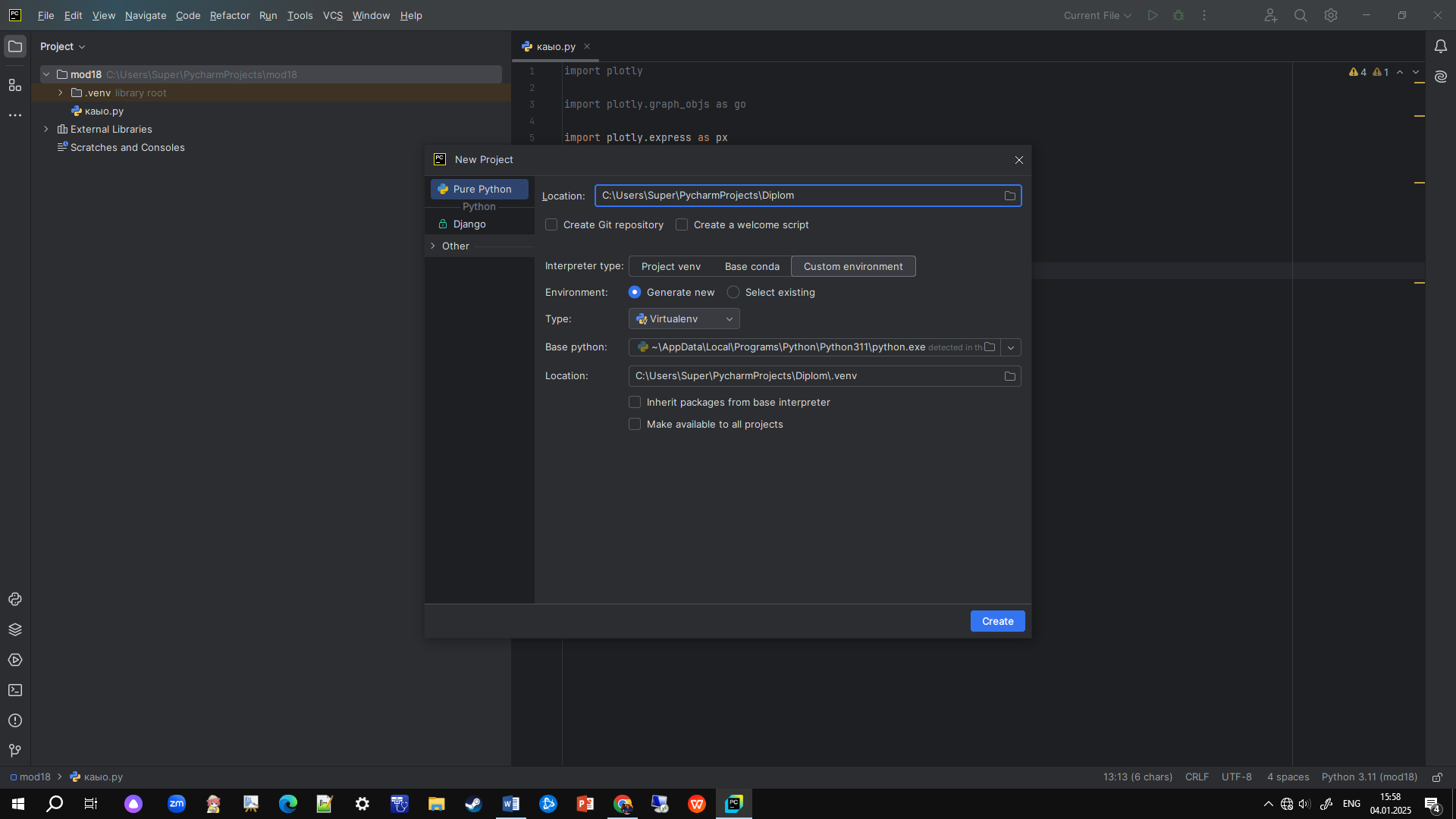
У разработки будет несколько этапов:

1. Создание проекта
2. В проекте потребуется создать несколько директорий, а в них файлы. Каждая директория нужна для заданий определенных типов (т.е. одна директория для круговой диаграммы, другая для графика функции и т.д.)
3. Подключение библиотек
4. Написание кода
5. Ввод и проверка данных
6. Проверка работоспособности, запуск программ
7. Сравнение и анализ
8. Написание вывода

Разработка:

Создание проекта:

Заходим в Pycharm и создаем новый проект, даем ему название и без изменения других настроек создаём его с помощью кнопки «Create»:

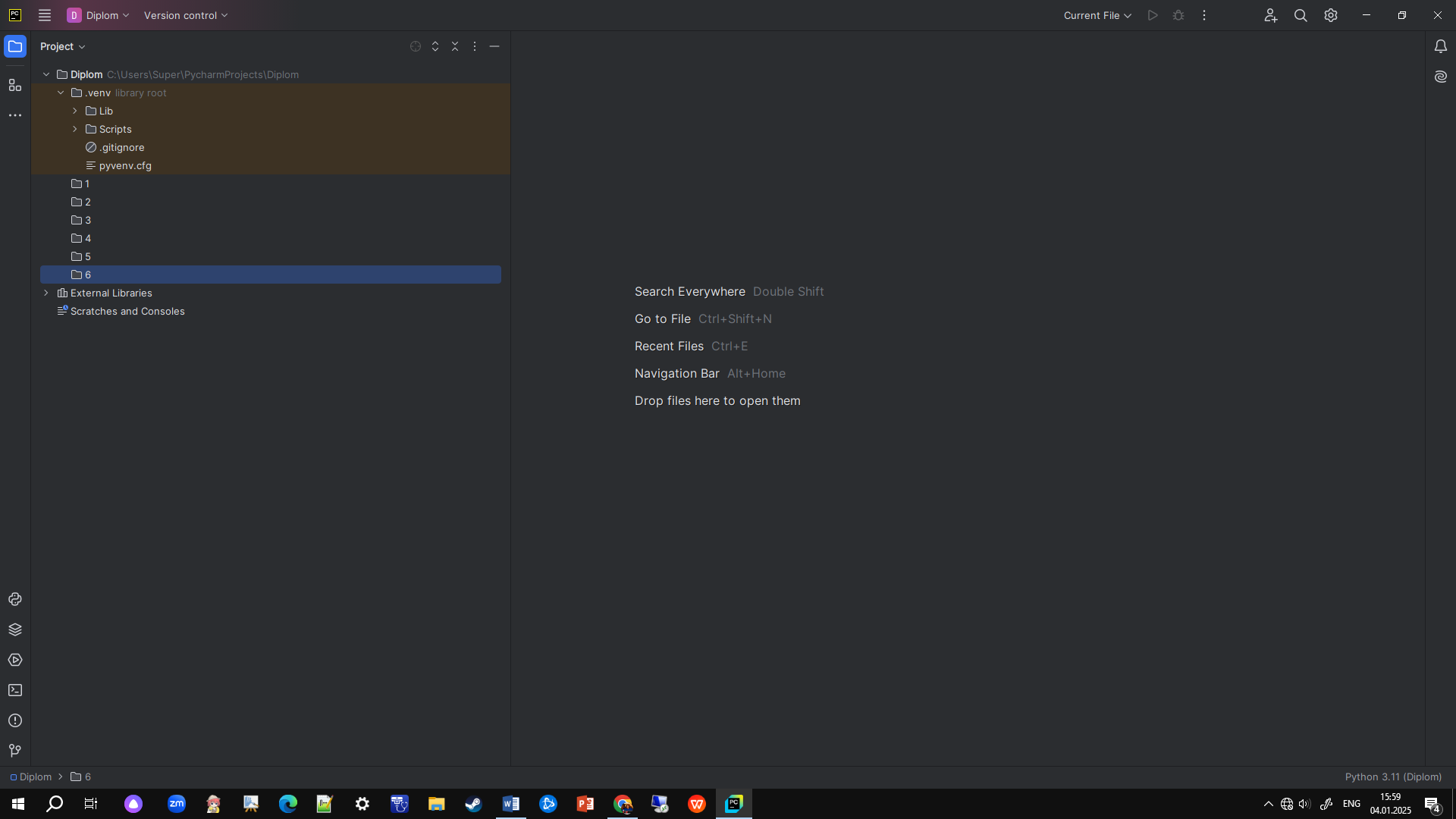


Создание директорий:

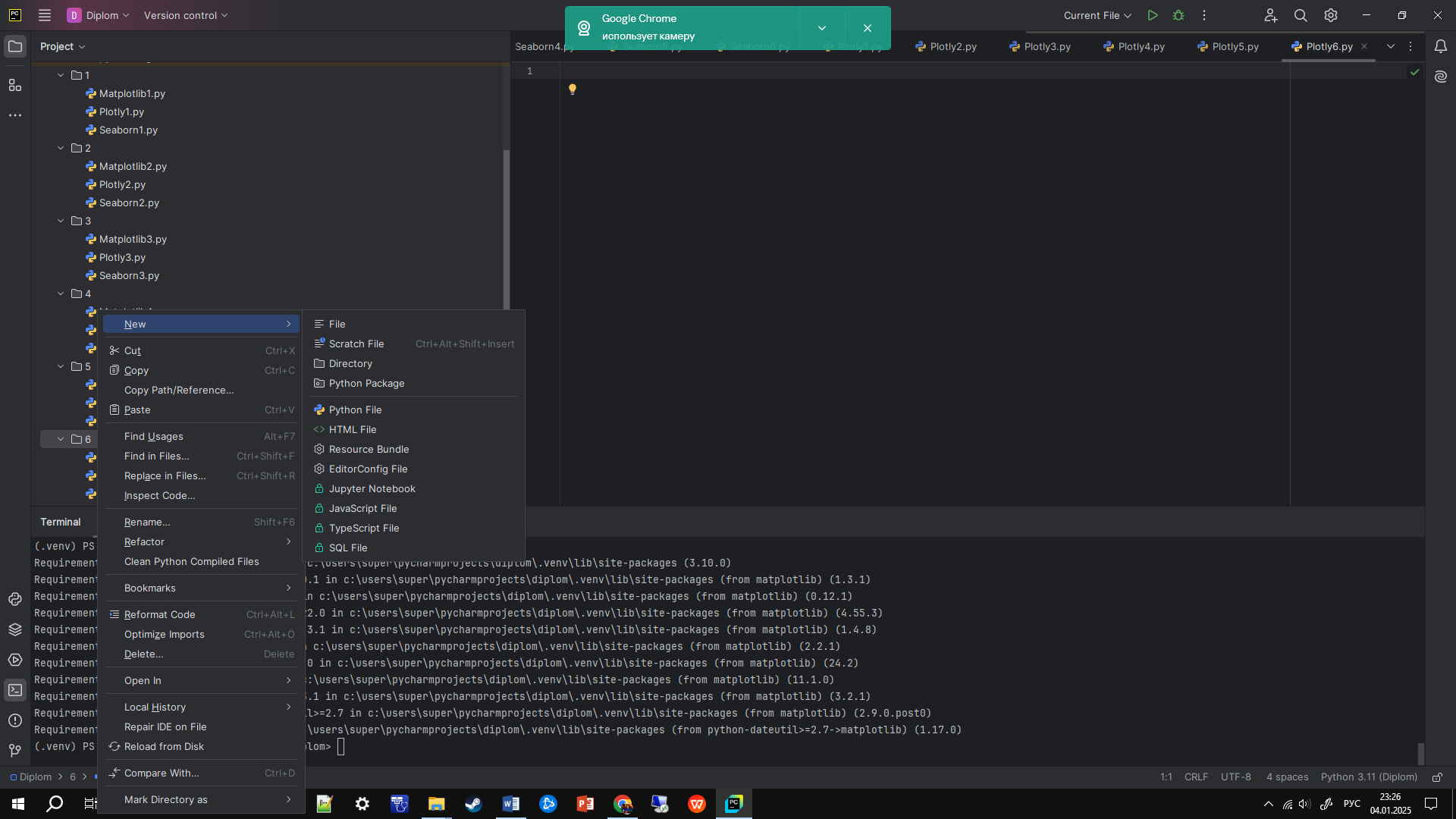
Далее требуется создать 4 директории внутри проекта:

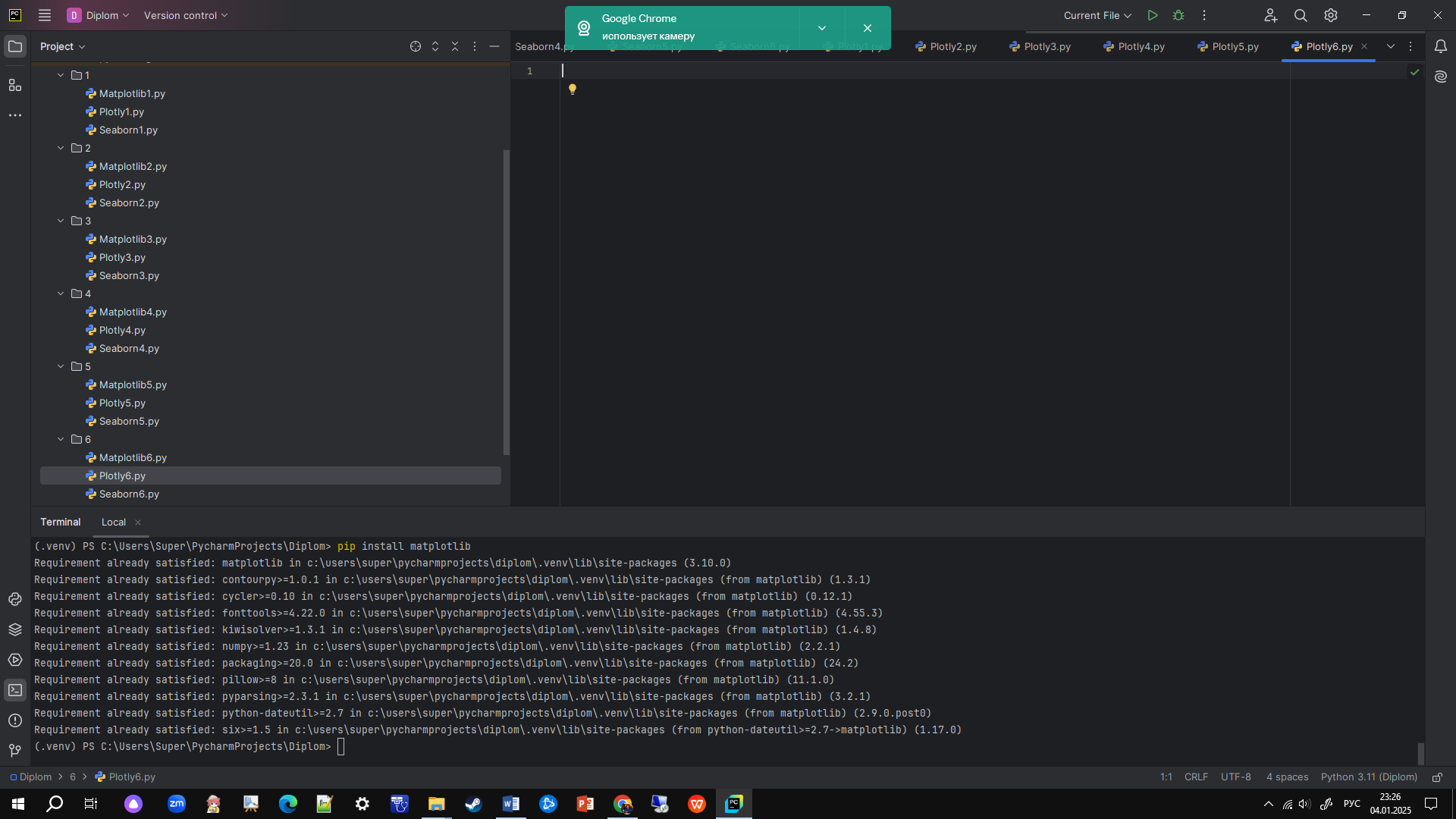
 (Создание директорий)

Названия директорий не важны, в моем случае это цифры от 1 до 5 (т.е. 5 штук и это совпадает с количеством примеров графиков и диаграмм):

 (Готовые директории в проекте)

После создания директорий, нужно создать файлы, по три файла в каждой:

 (Создание файла в Pycharm)

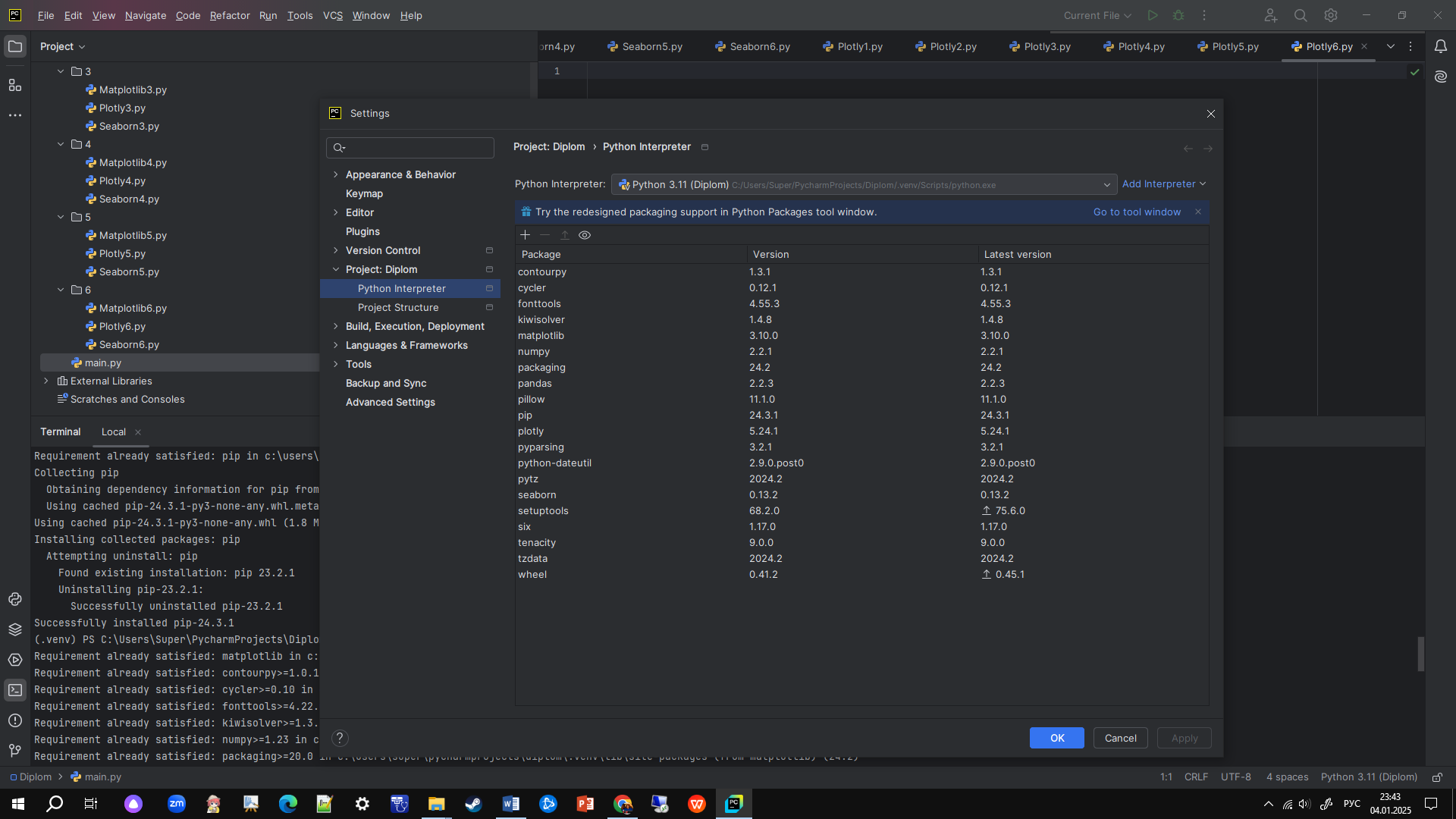
 (Готовые файлы в своих директориях)

Подключение библиотек:

Далее нужно подключить все библиотеки (Matplotlib, Seaborn, Plotly, numpy и pandas). Во всех файлах версии библиотек должны быть одинаковы и не должны конфликтовать друг с другом. Скачивание происходит с помощью команды в терминале

«pip install “название библиотеки” (если нужна определенная версия, после названия нужно написать «== “версия библиотеки”»)»

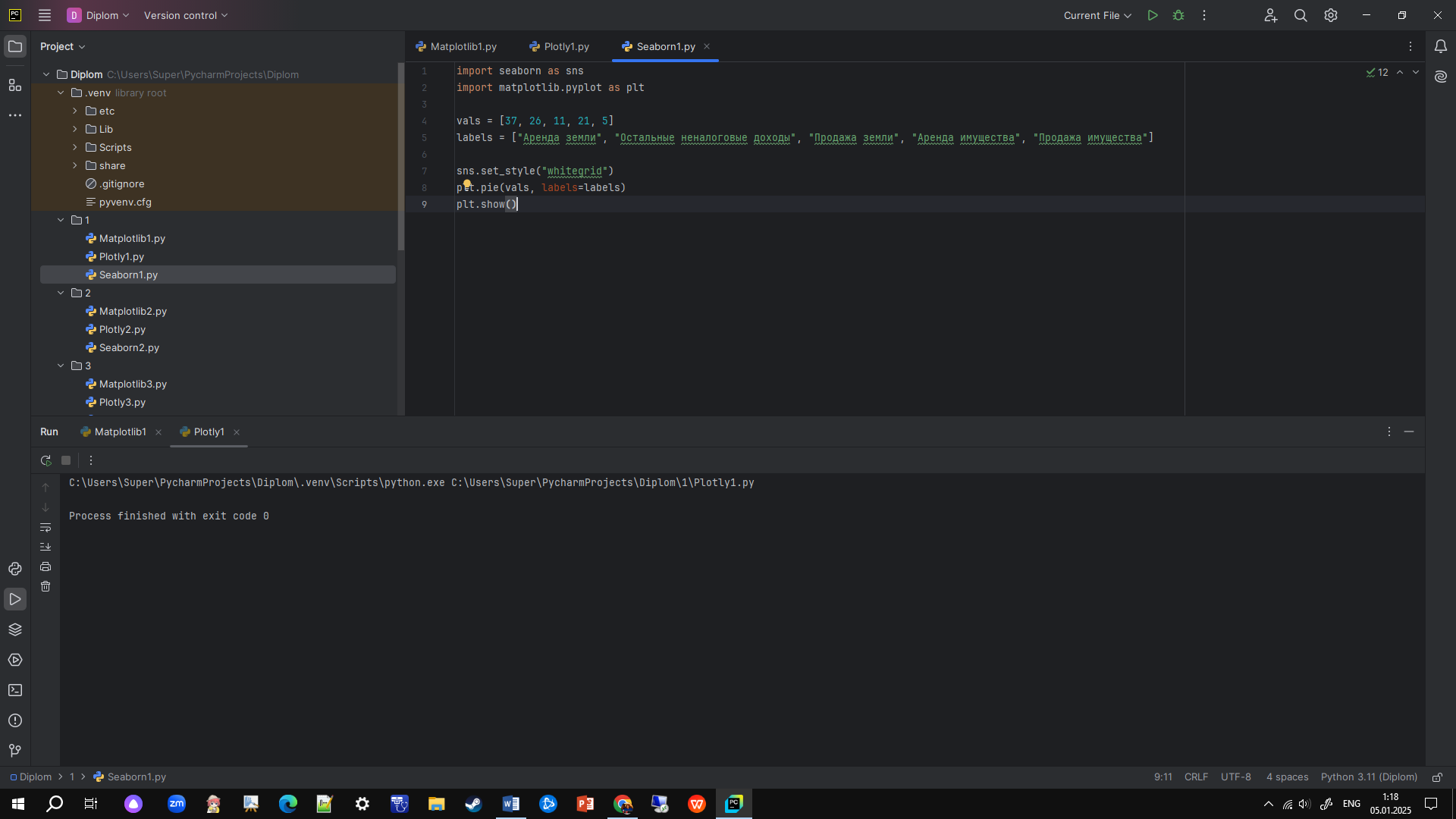
После скачивания библиотек Python Interpreter выглядит так:



Здесь написаны все имеющиеся и готовые к использованию библиотеки, их версии и последние версии из всех существующих.

Написание кода:

Изначально нужно импортировать библиотеки в файлы:

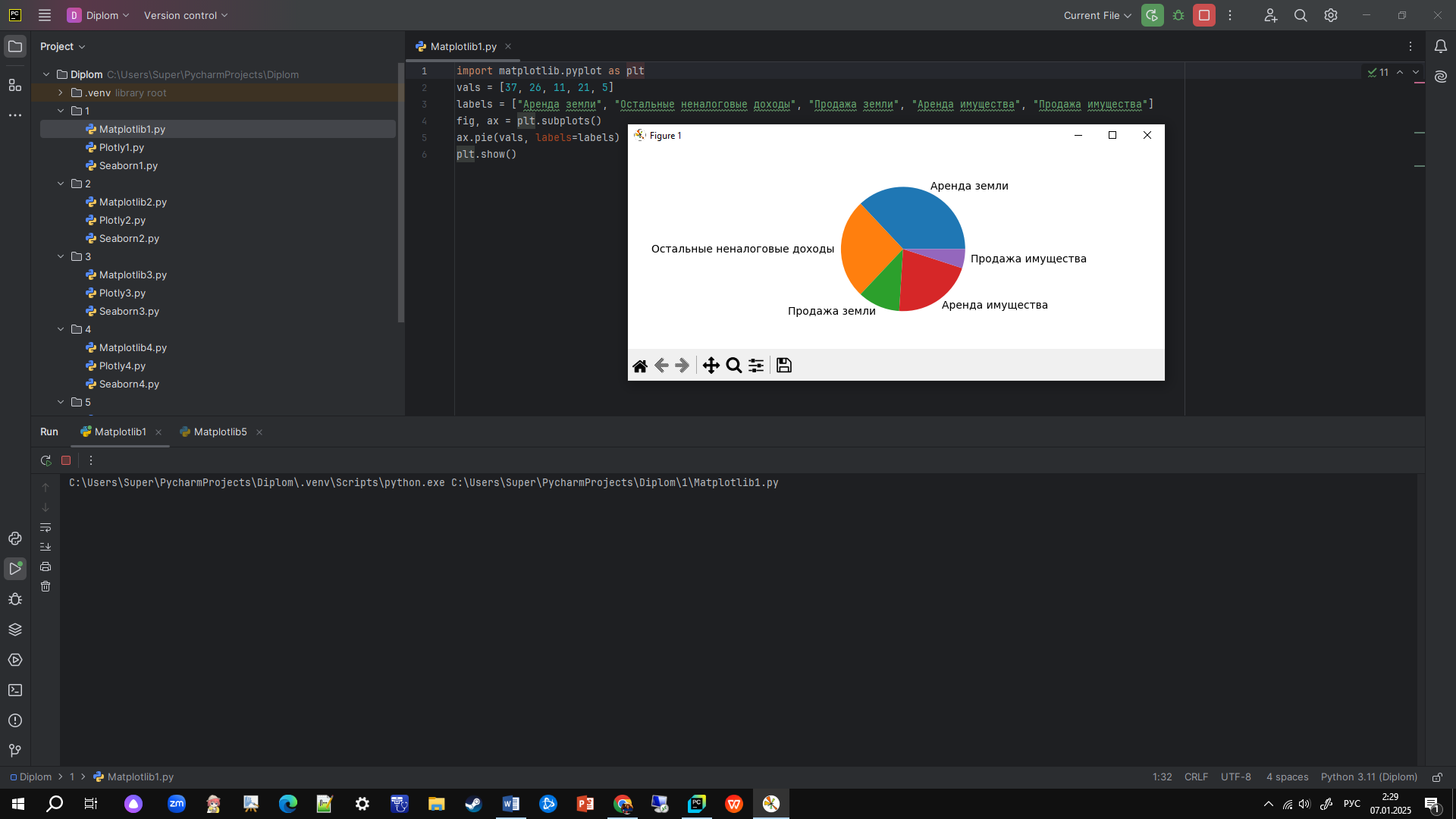


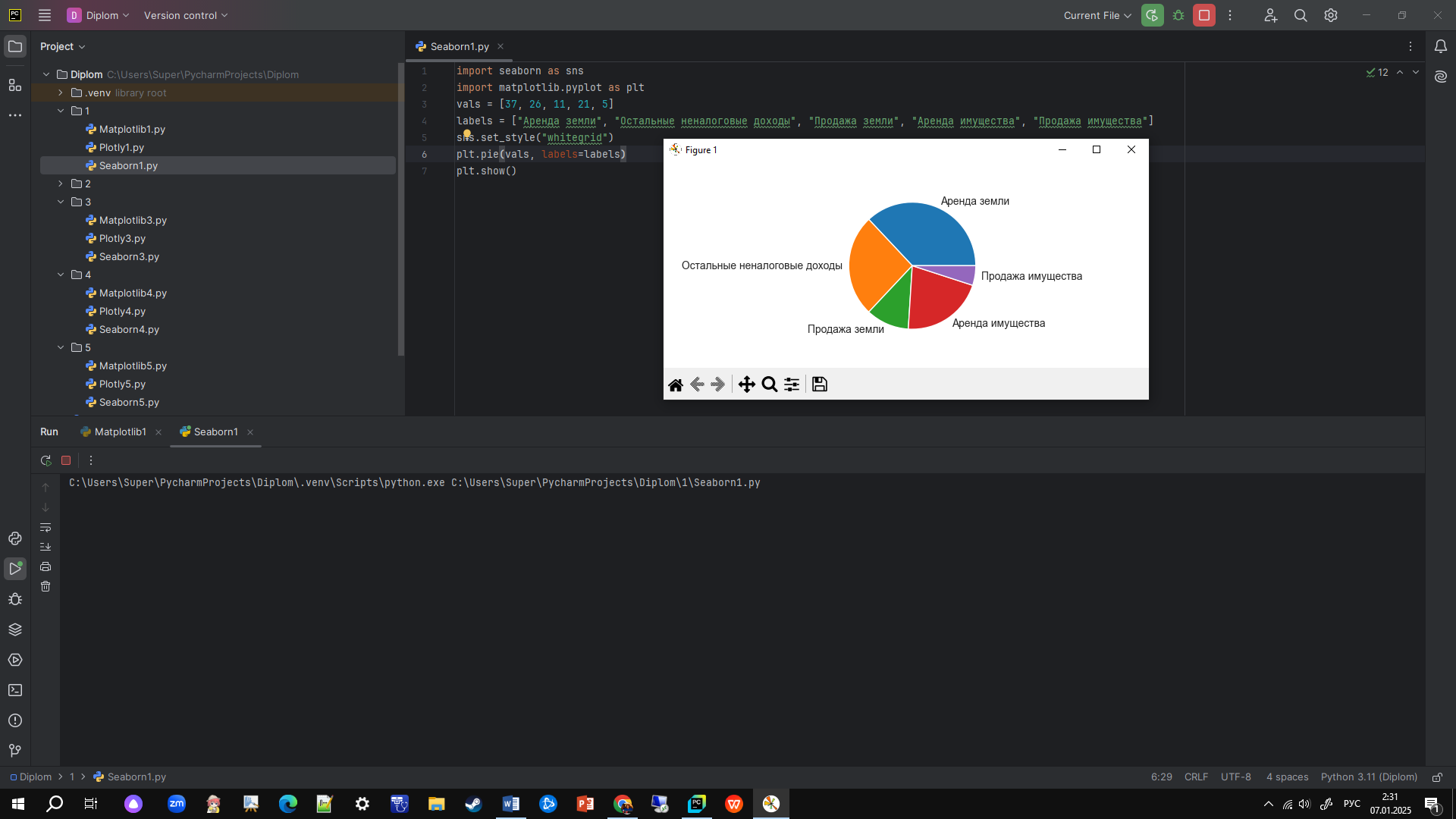
(Использую import … as …, так. позже будет удобнее работать)

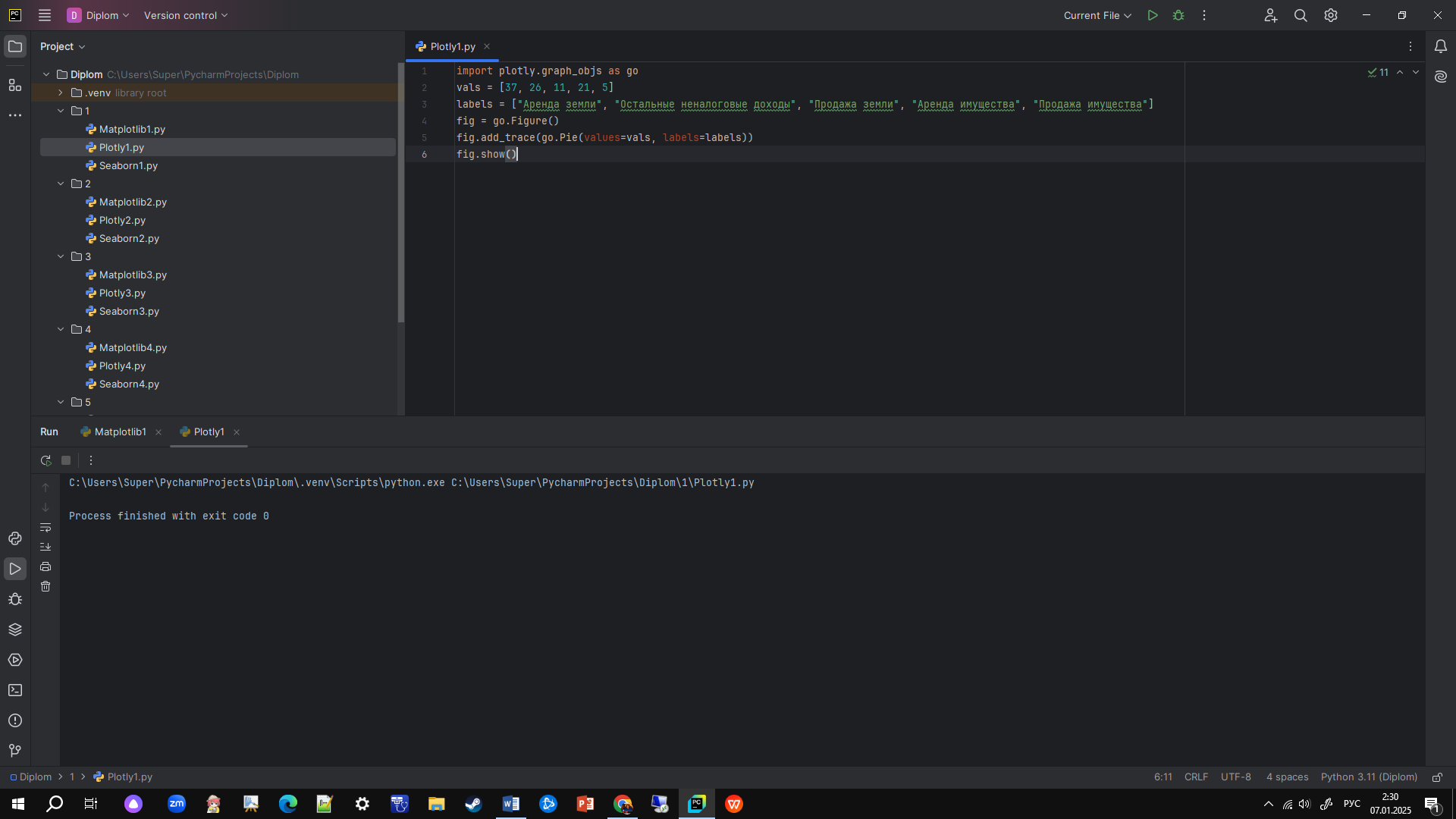
После подготовки, нужно написать код. Для поиска информации по библиотекам и понимания того как нужно писать код с использованием тех или иных библиотек я использую официальные сайты библиотек Plotly, Matplotlib и Seaborn, (то есть <https://plotly.com/graphing-libraries/>, <https://matplotlib.org/stable/> и <https://seaborn.pydata.org/index.html> соответственно); Сайт habr.com, где можно найти много интересной авторской информации по тем или иным темам, ведь в документации библиотек, часто информации не хватает для полноценной работы, для этих же целей можно использовать сайты medium.com, dev-gang.ru, sky.pro и skillbox.ru; Сайты stackoverflow.com и ru.stackoverflow.com для понимания того, что означают появившиеся ошибки и того как их исправить.

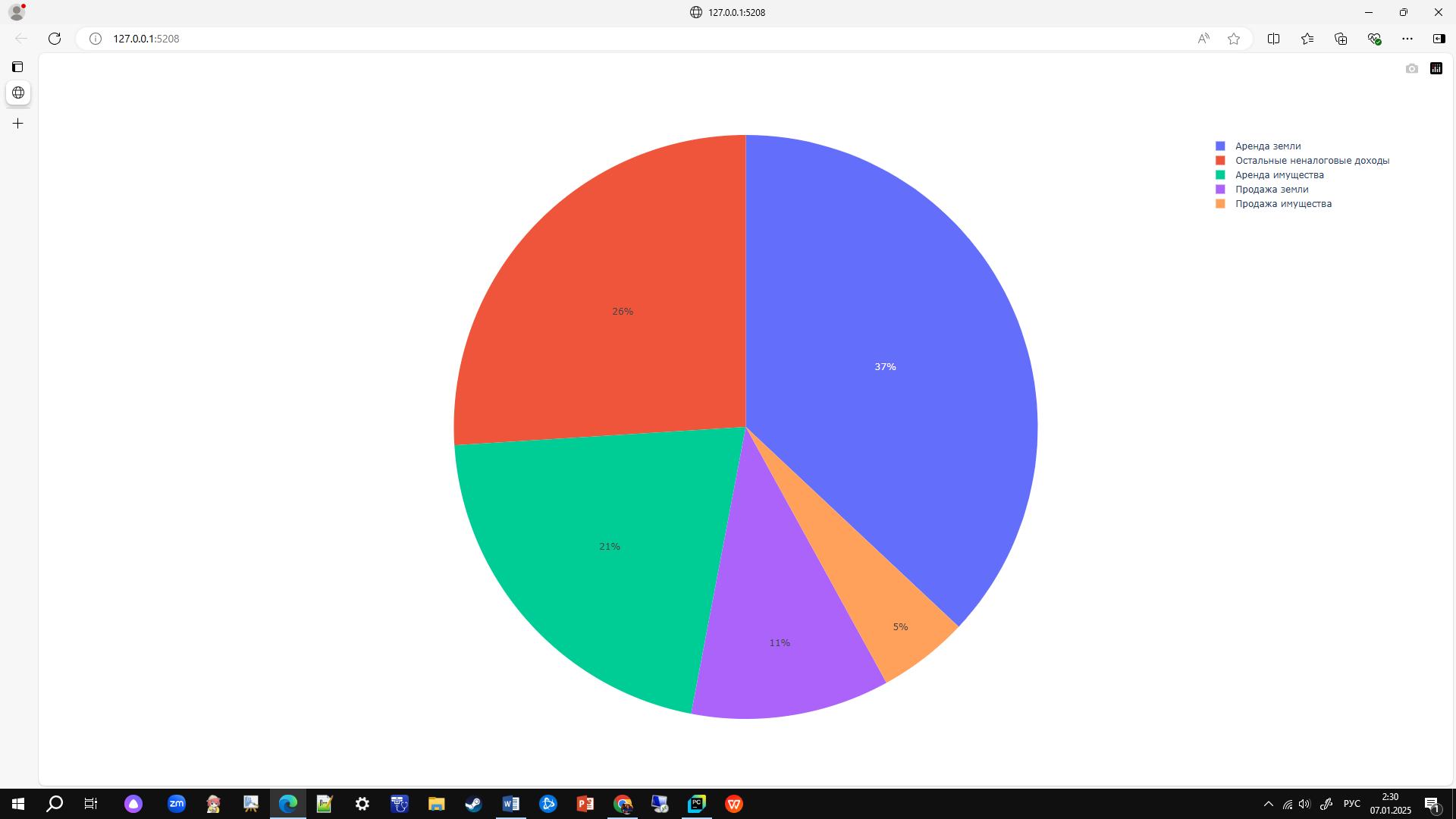
После поиска информации я написала код. В среднем длина программы от 7 до 9 строк. Самая длинная программа состоит из 21 строки, самая короткая из 5 строк.

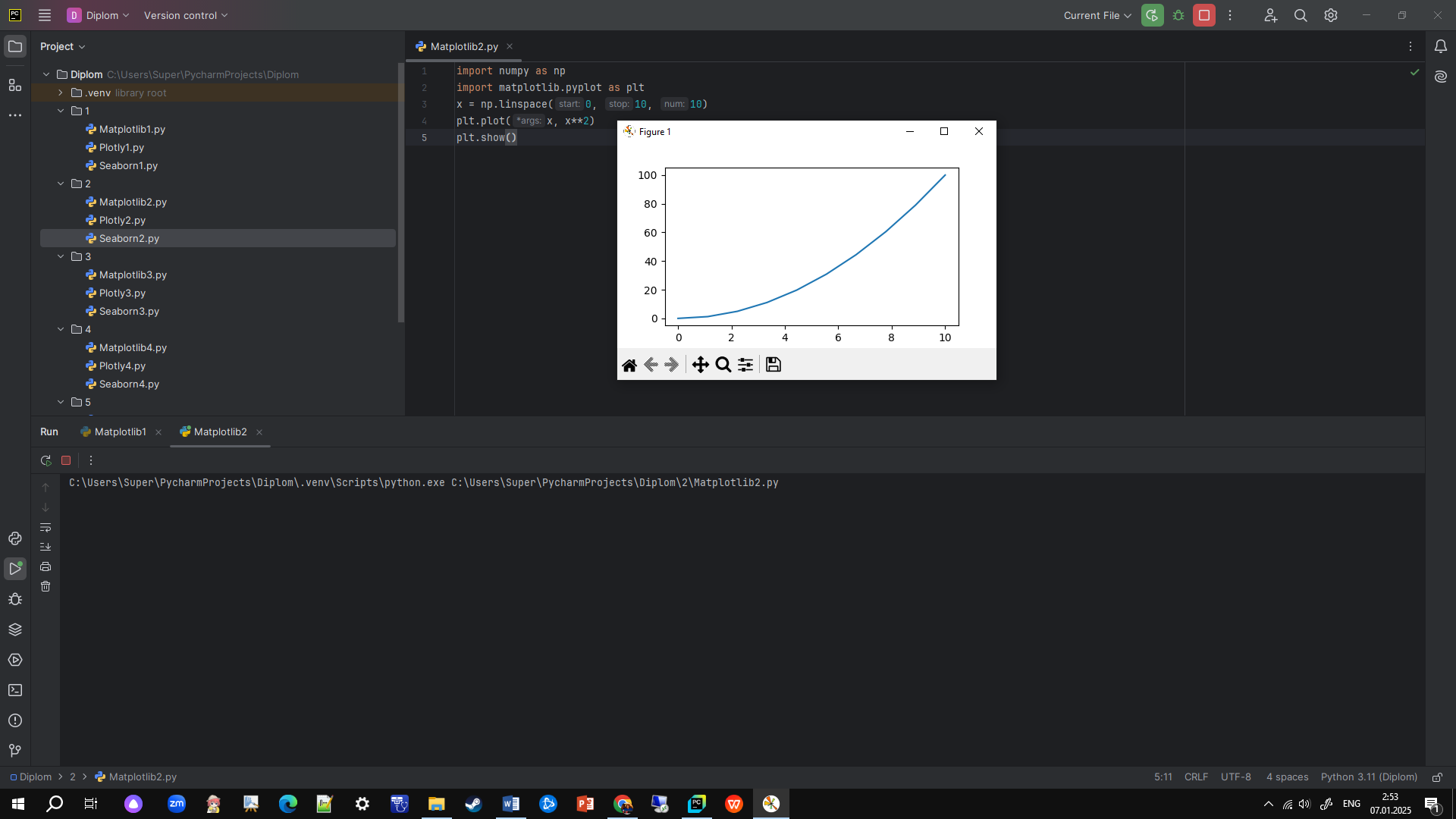
Программа 1 (Круговая диаграмма):

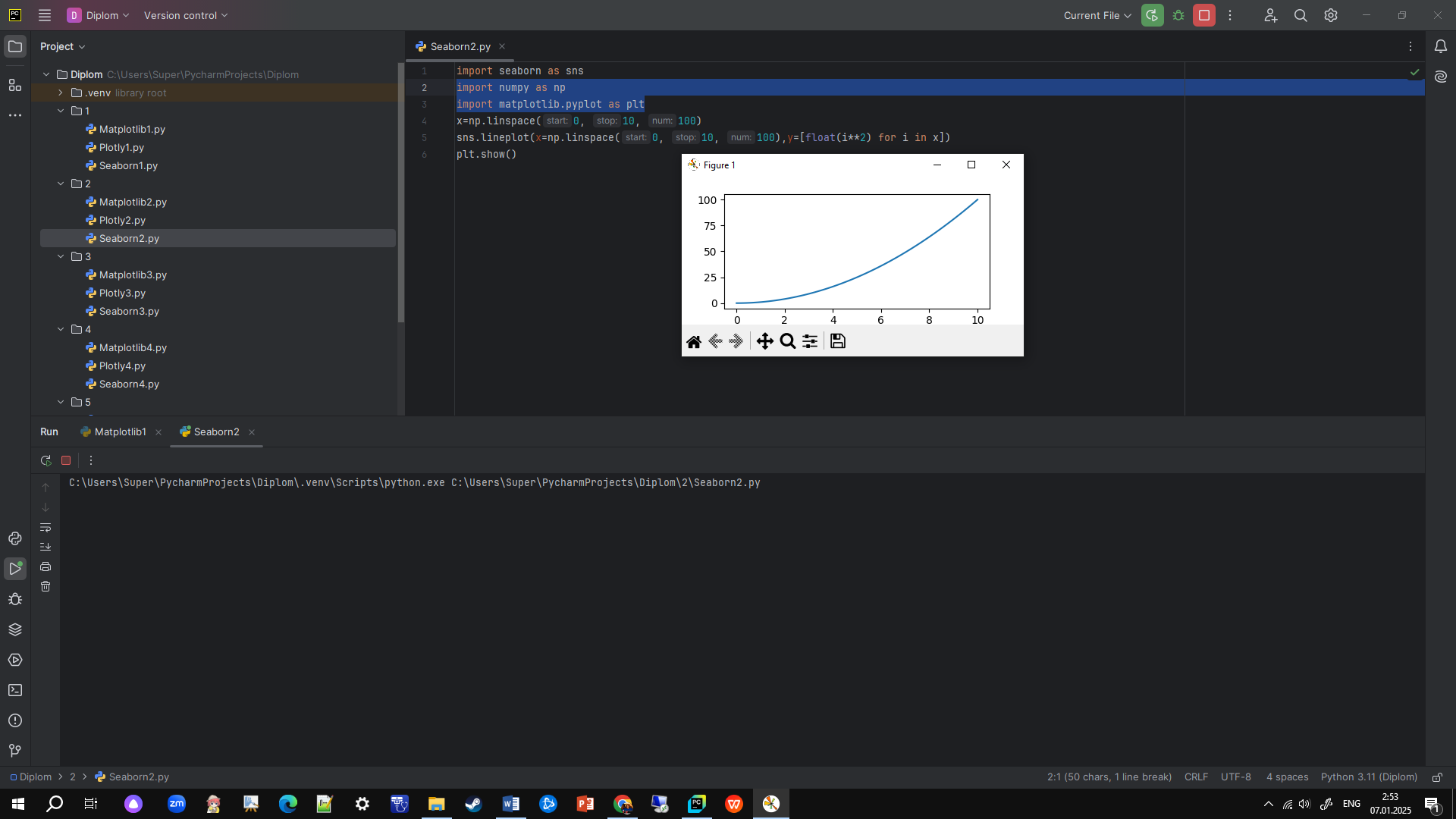
Matplotlib

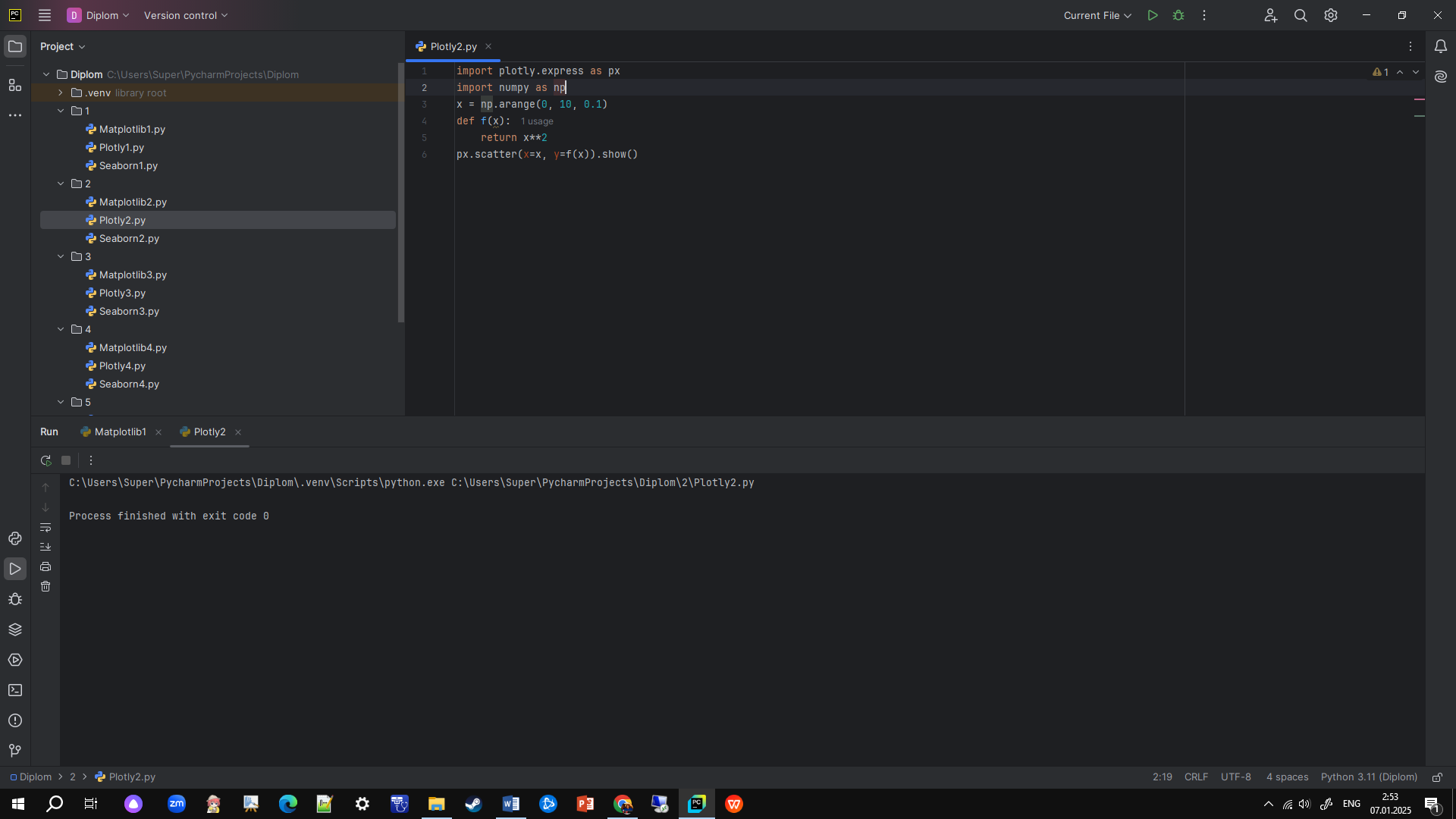
Seaborn

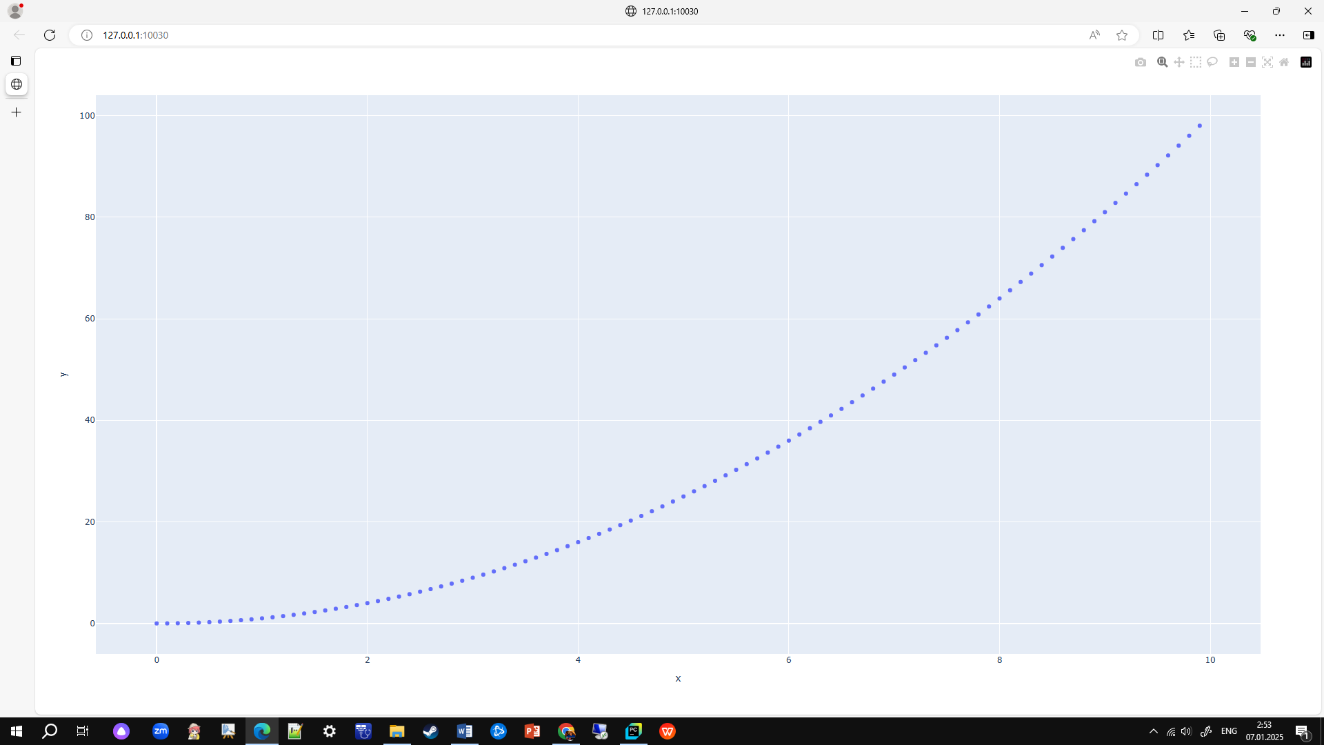


Plotly  
Программа 2 (График функции):

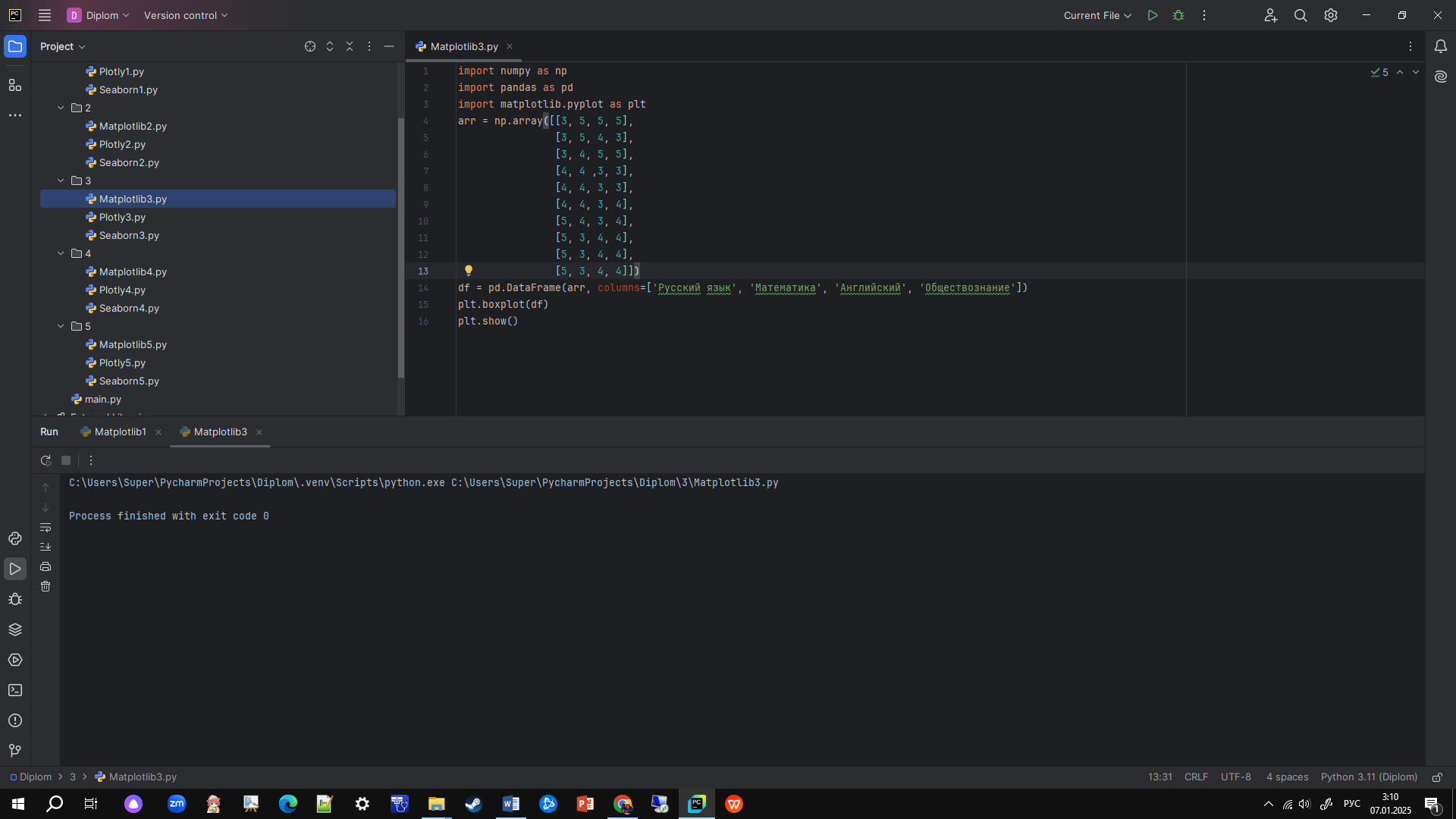
Matplotlib

Seaborn

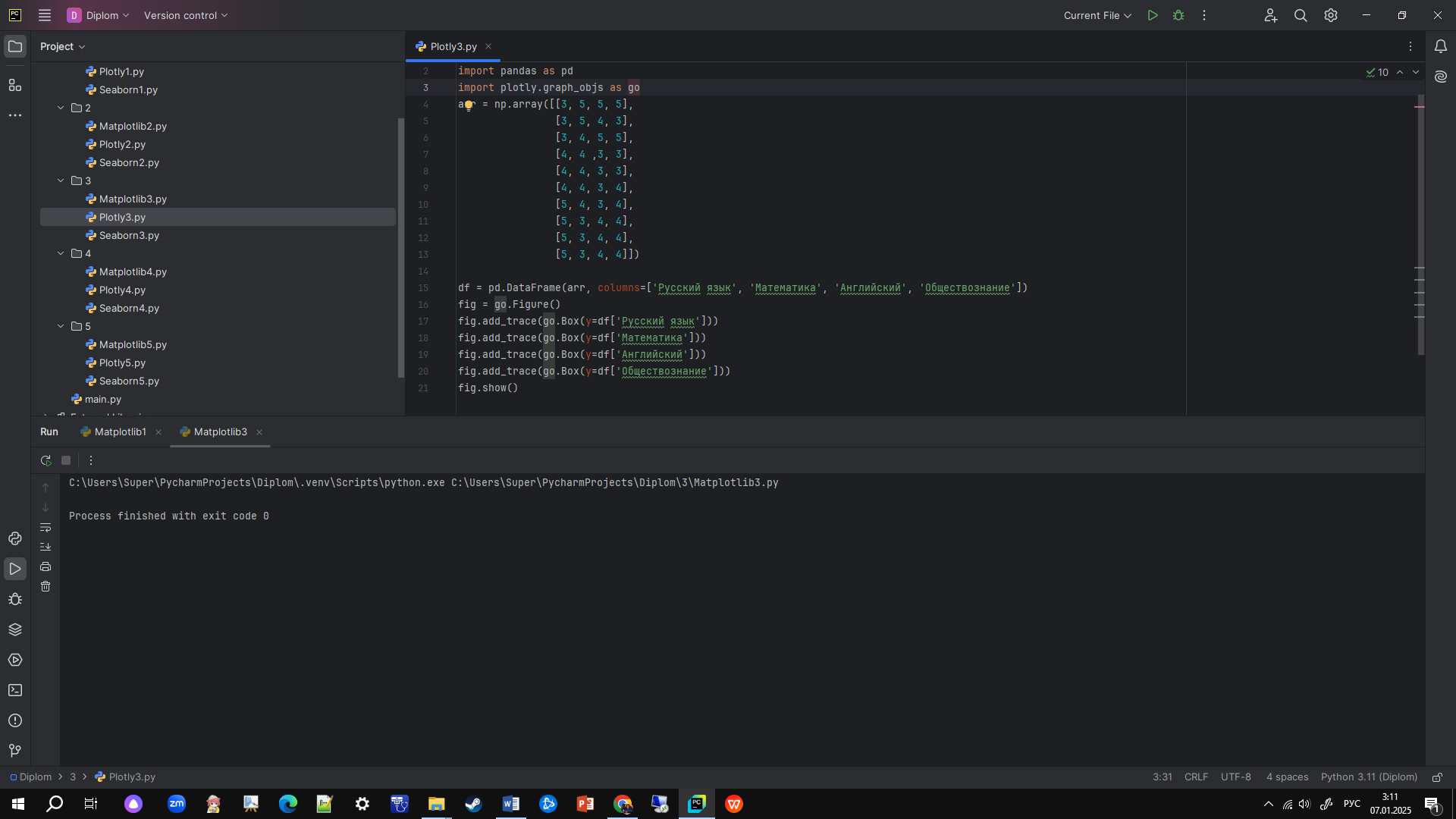


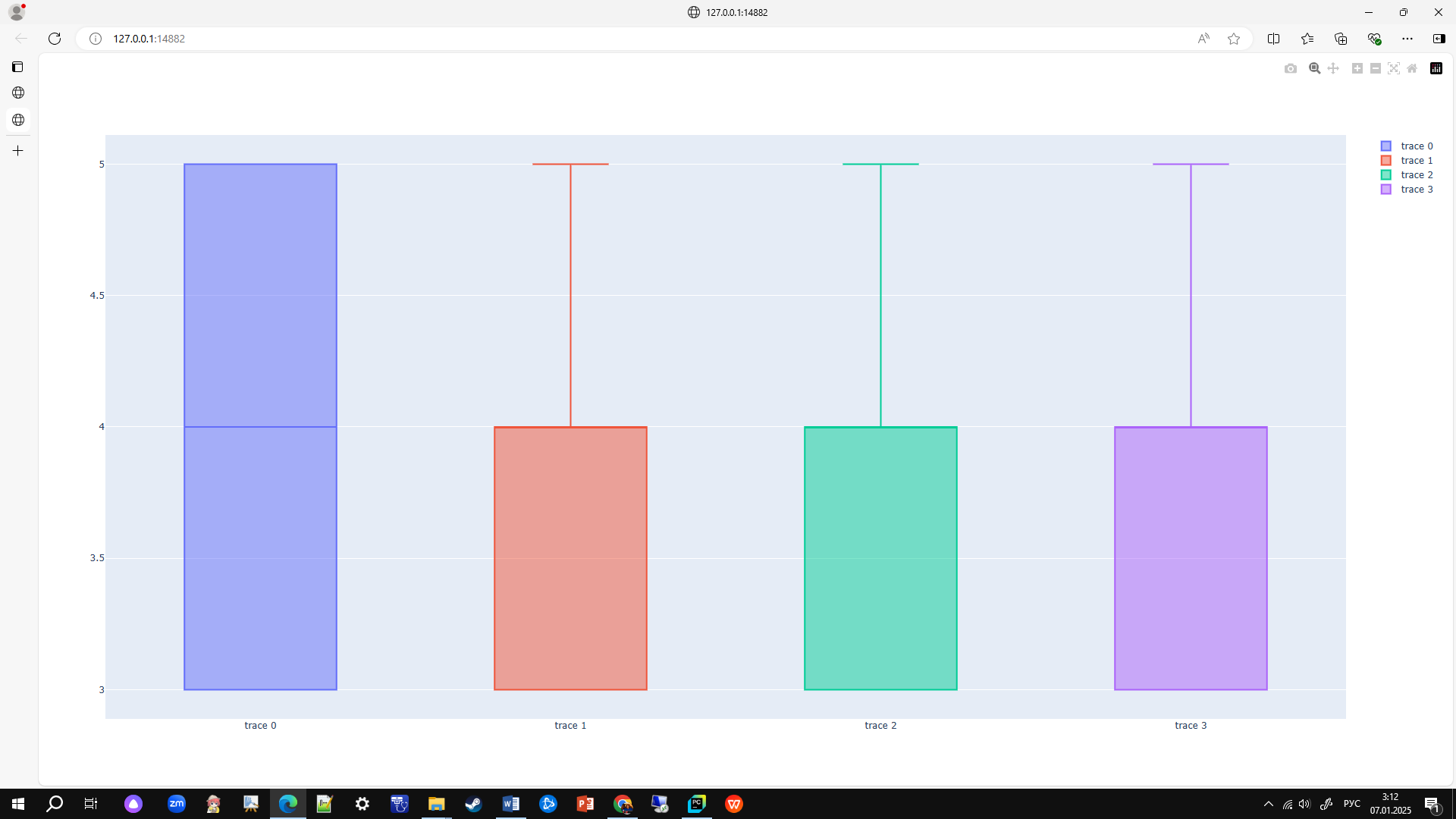
Plotly

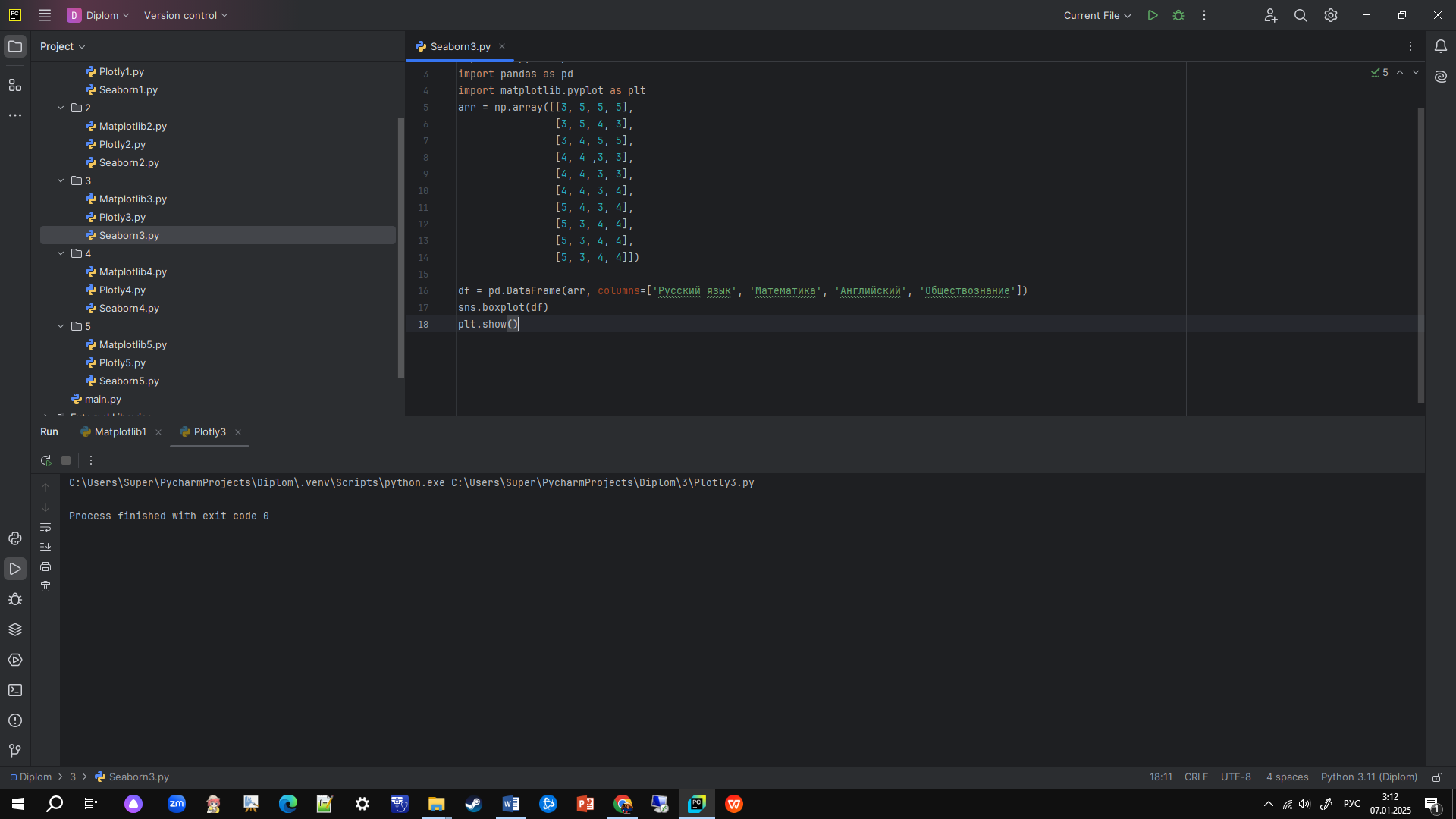
Программа 3 (Диаграмма размаха):

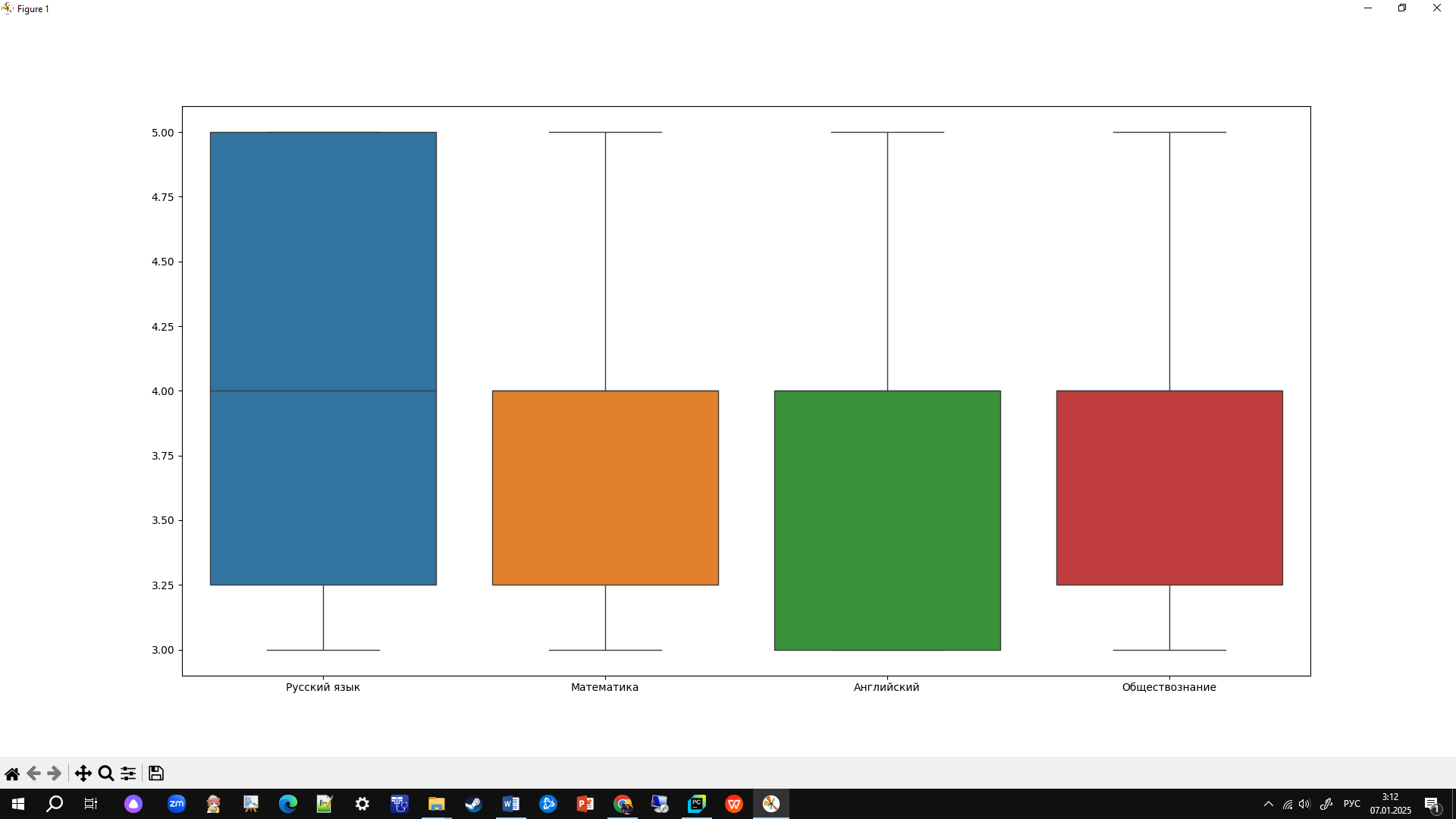


Matplotlib

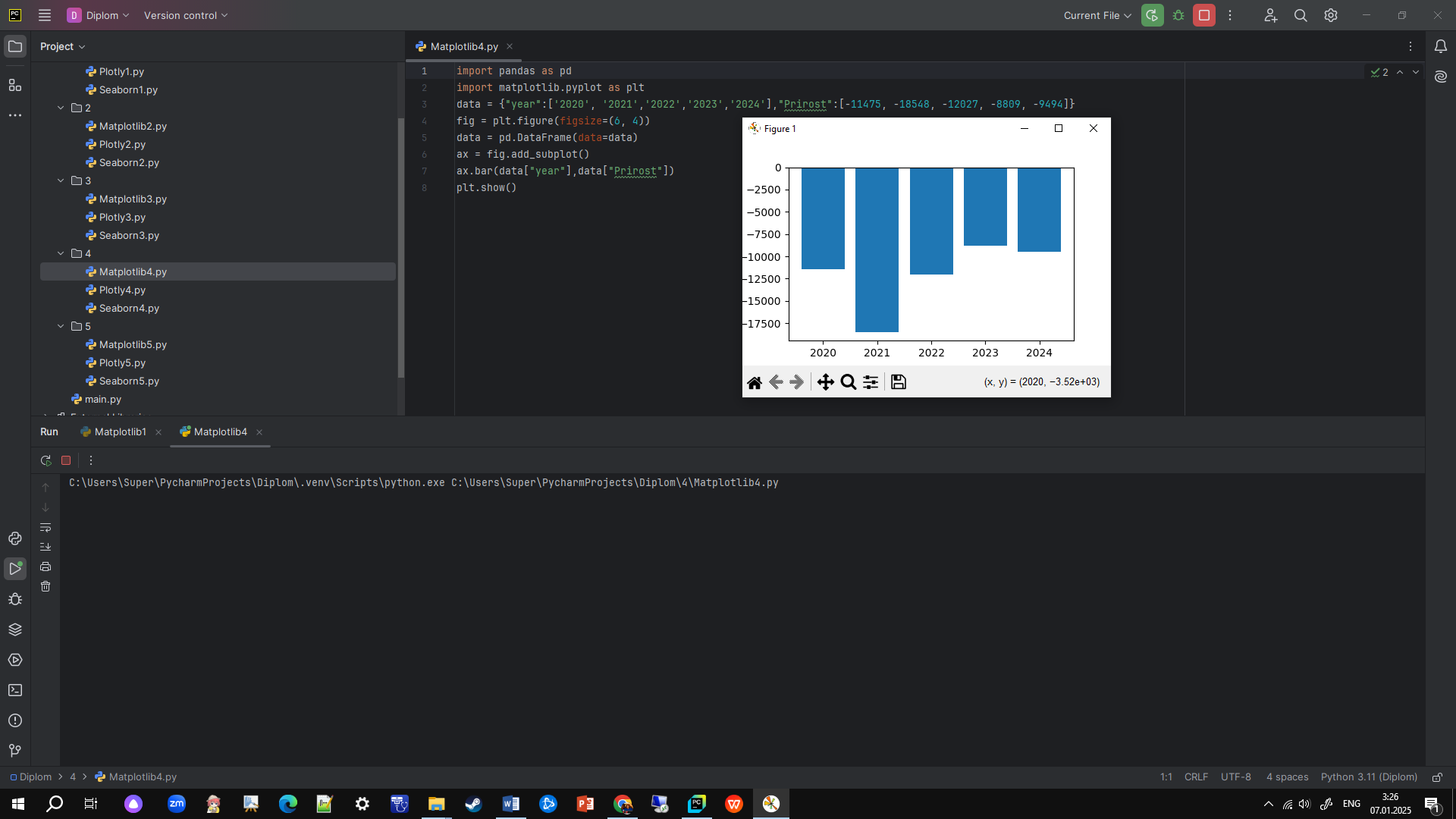


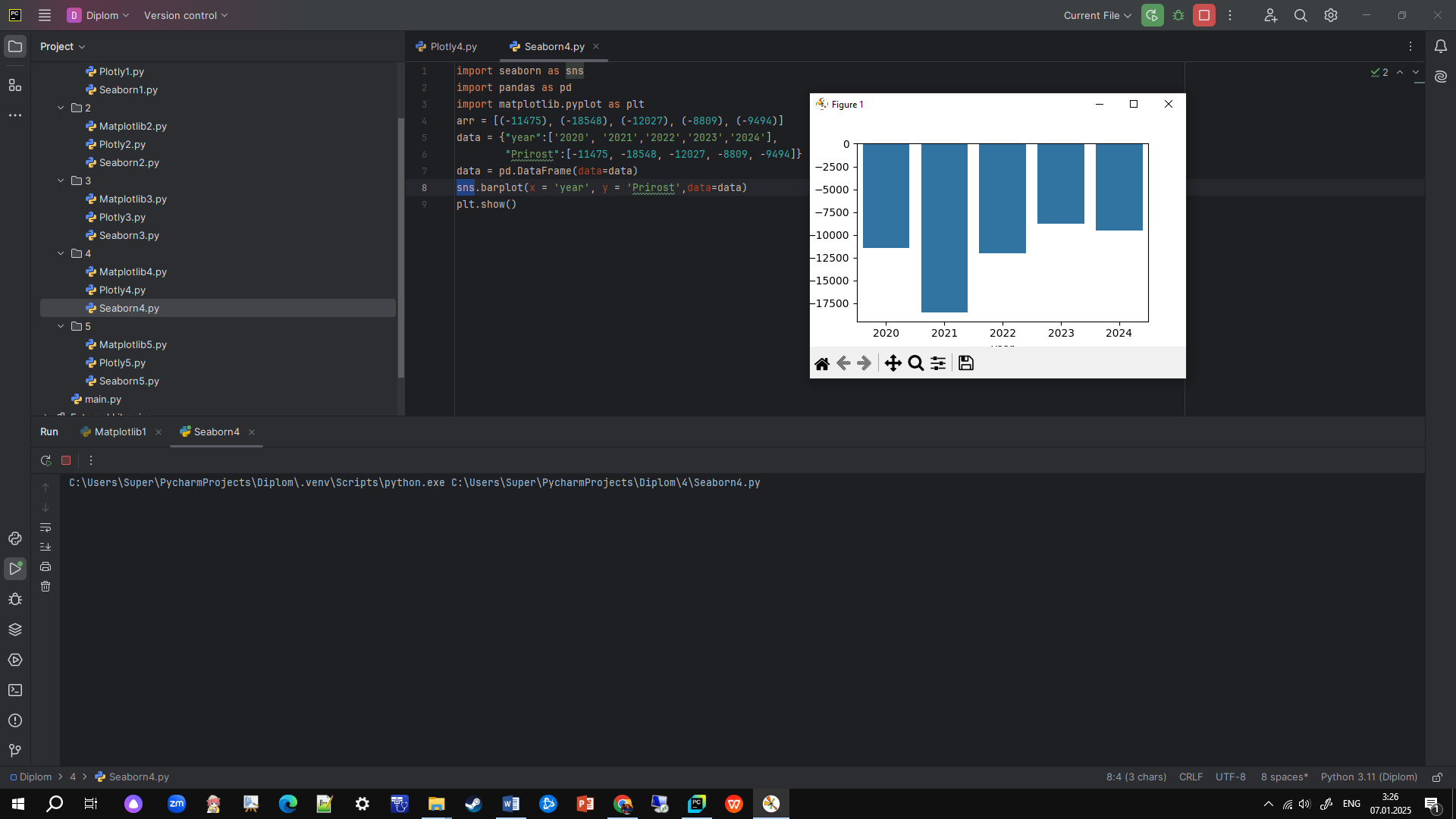
Plotly

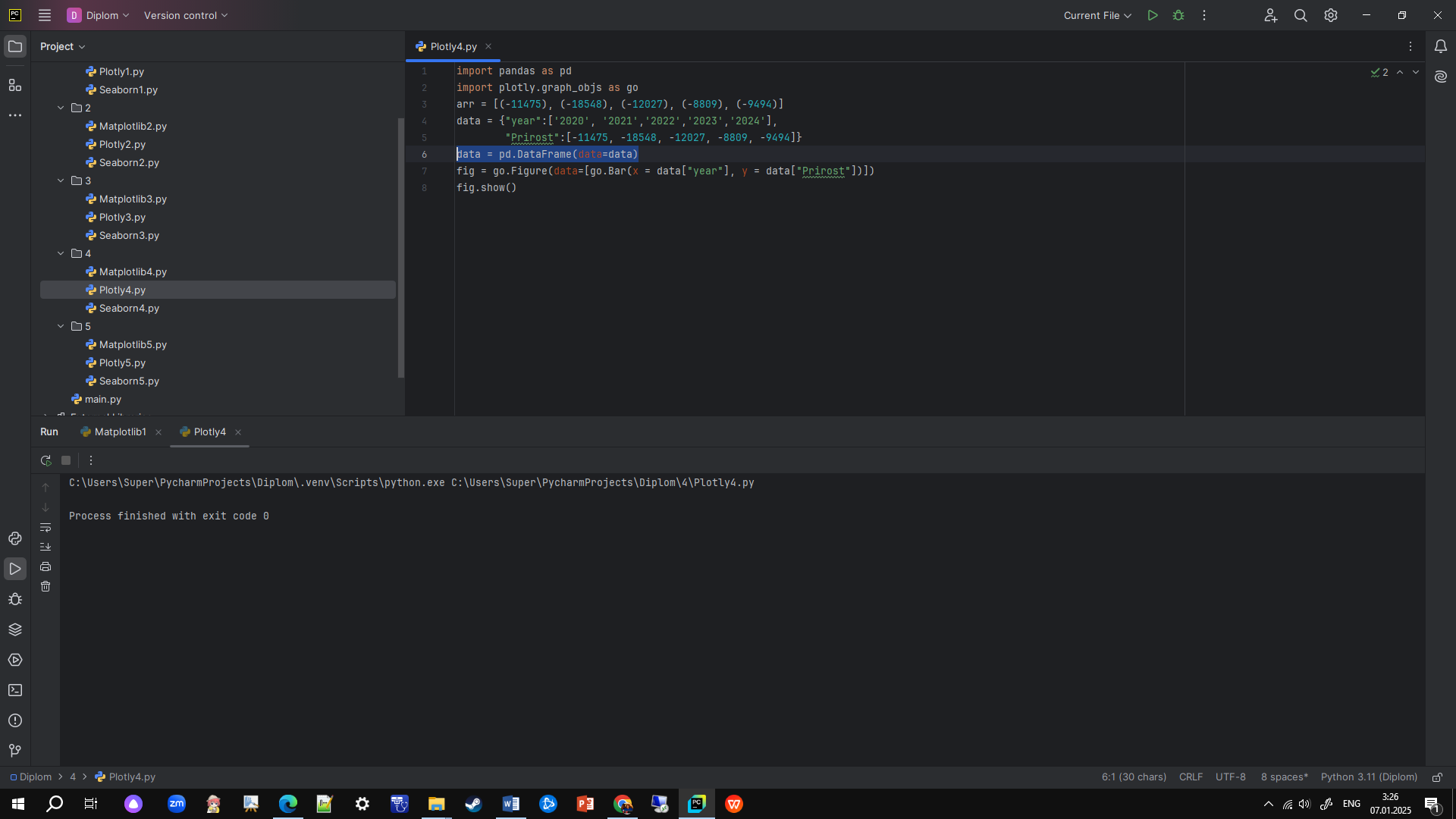


Seaborn

Программа 4 (Столбчатая диаграмма):

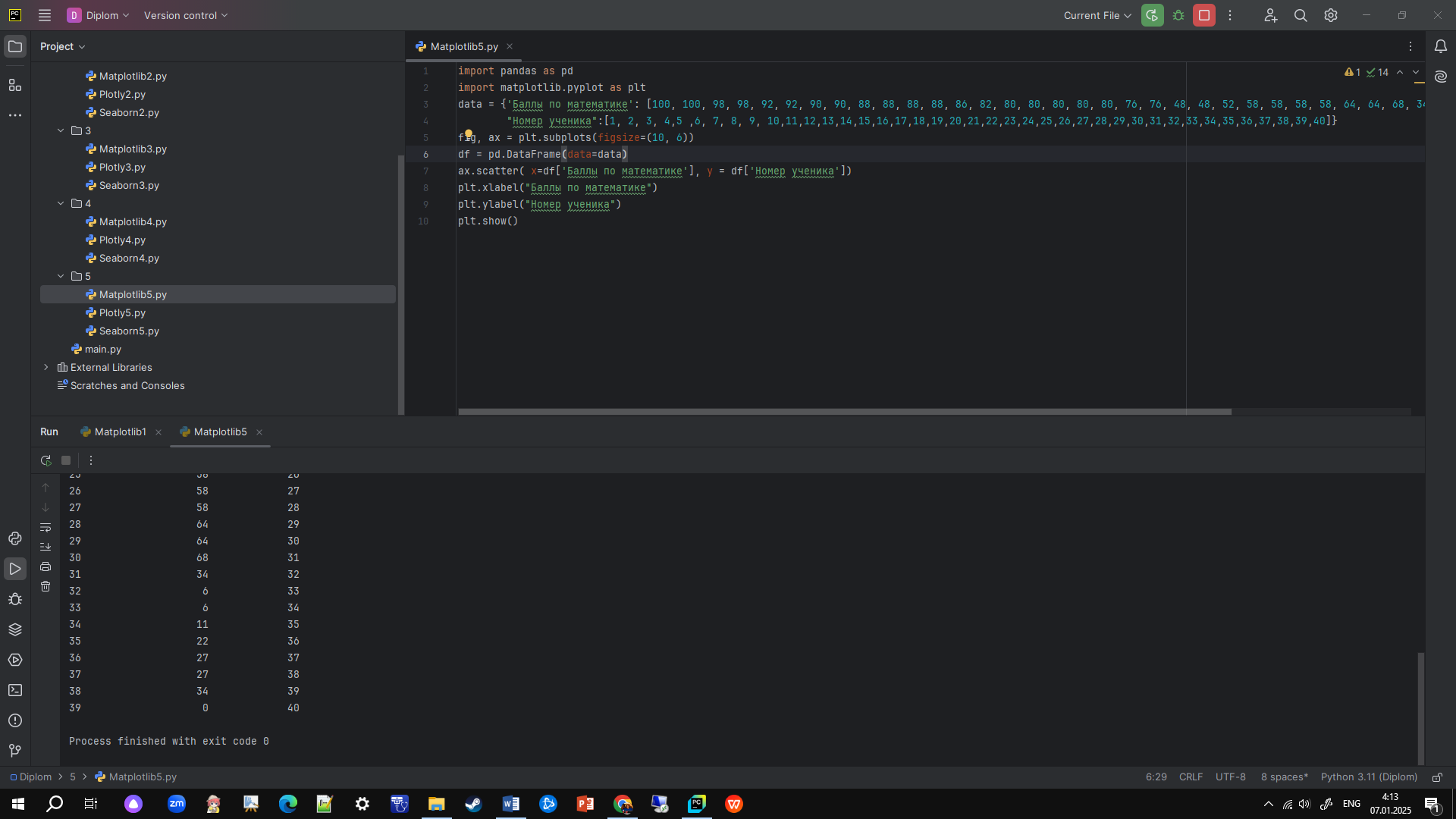
 Matplotlib

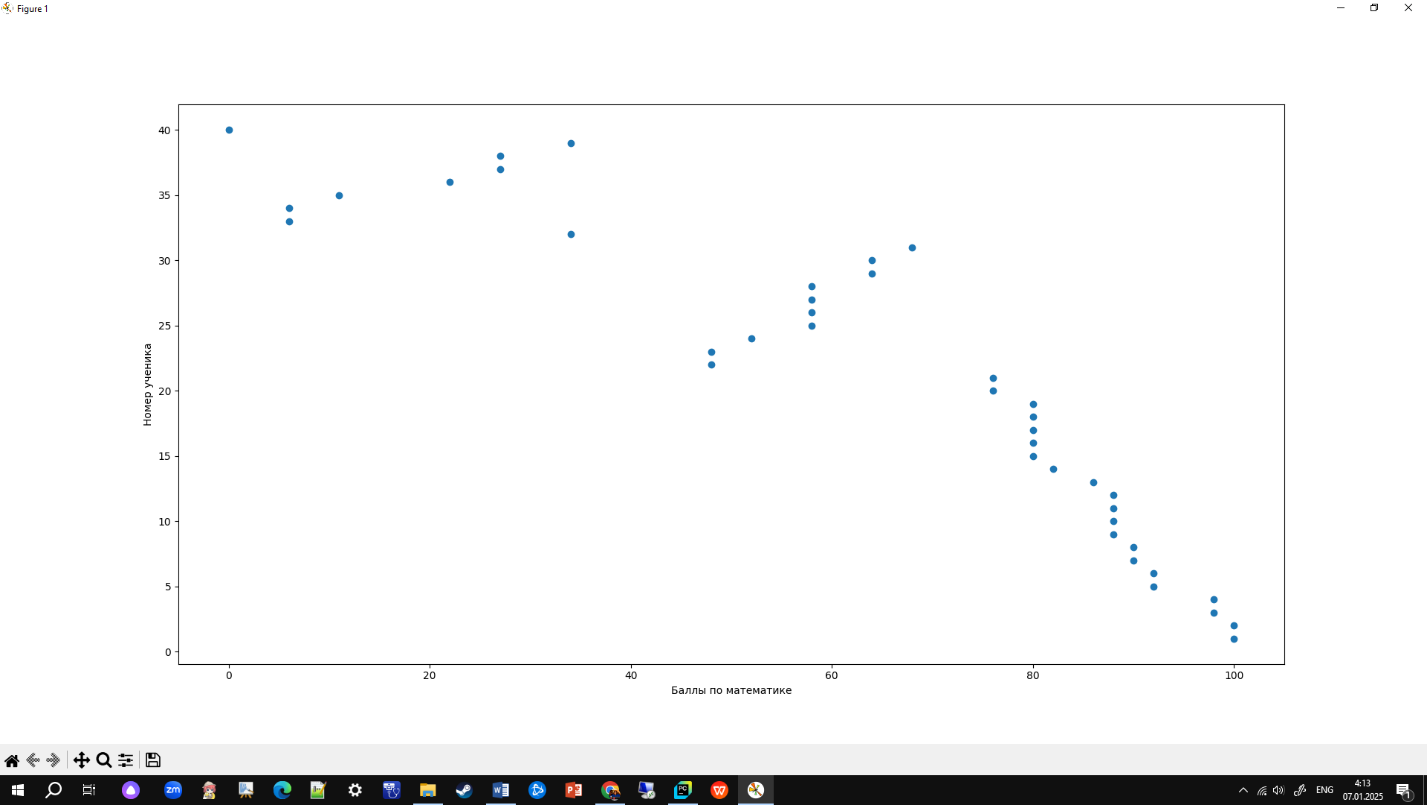
Seaborn

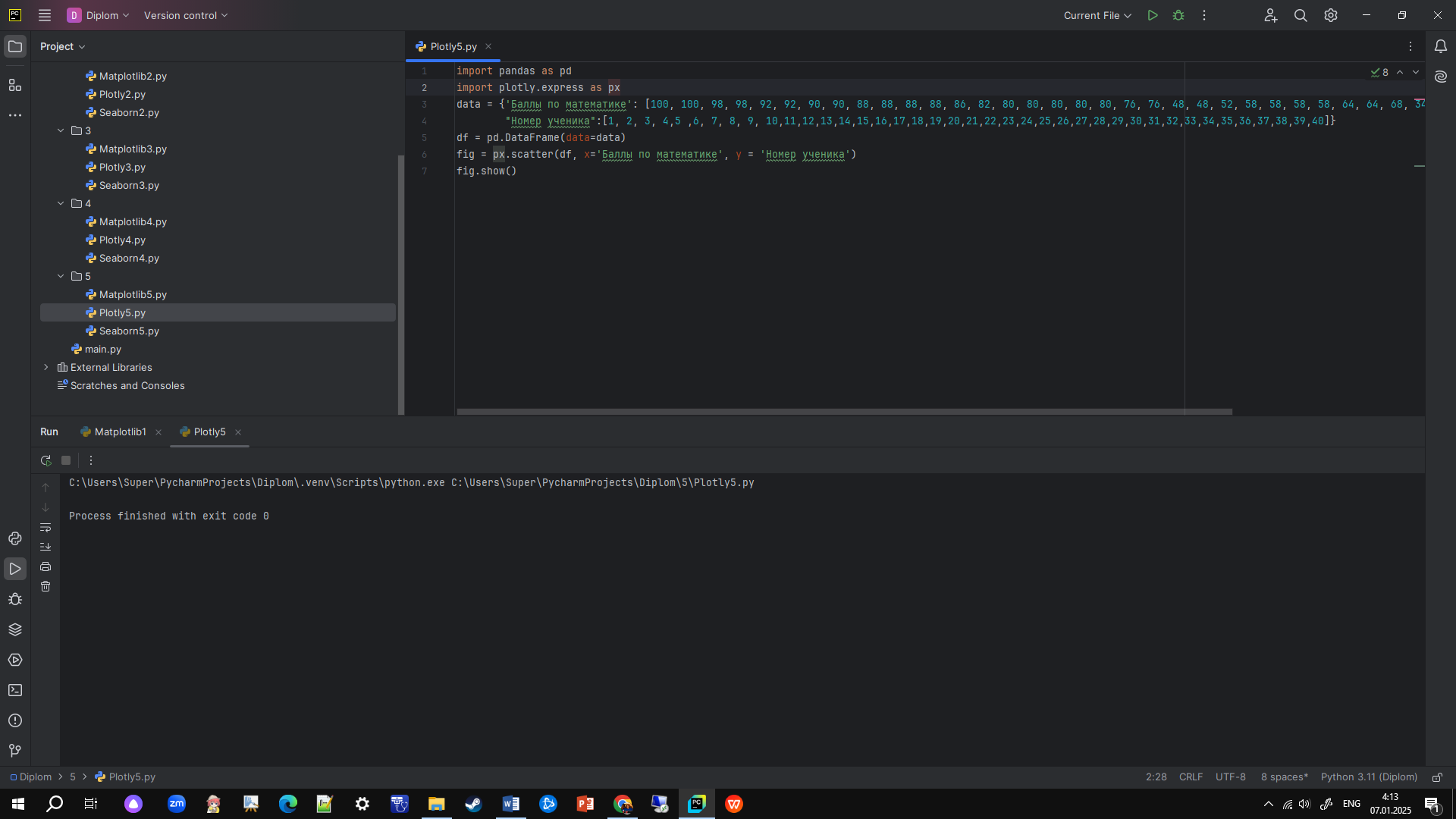


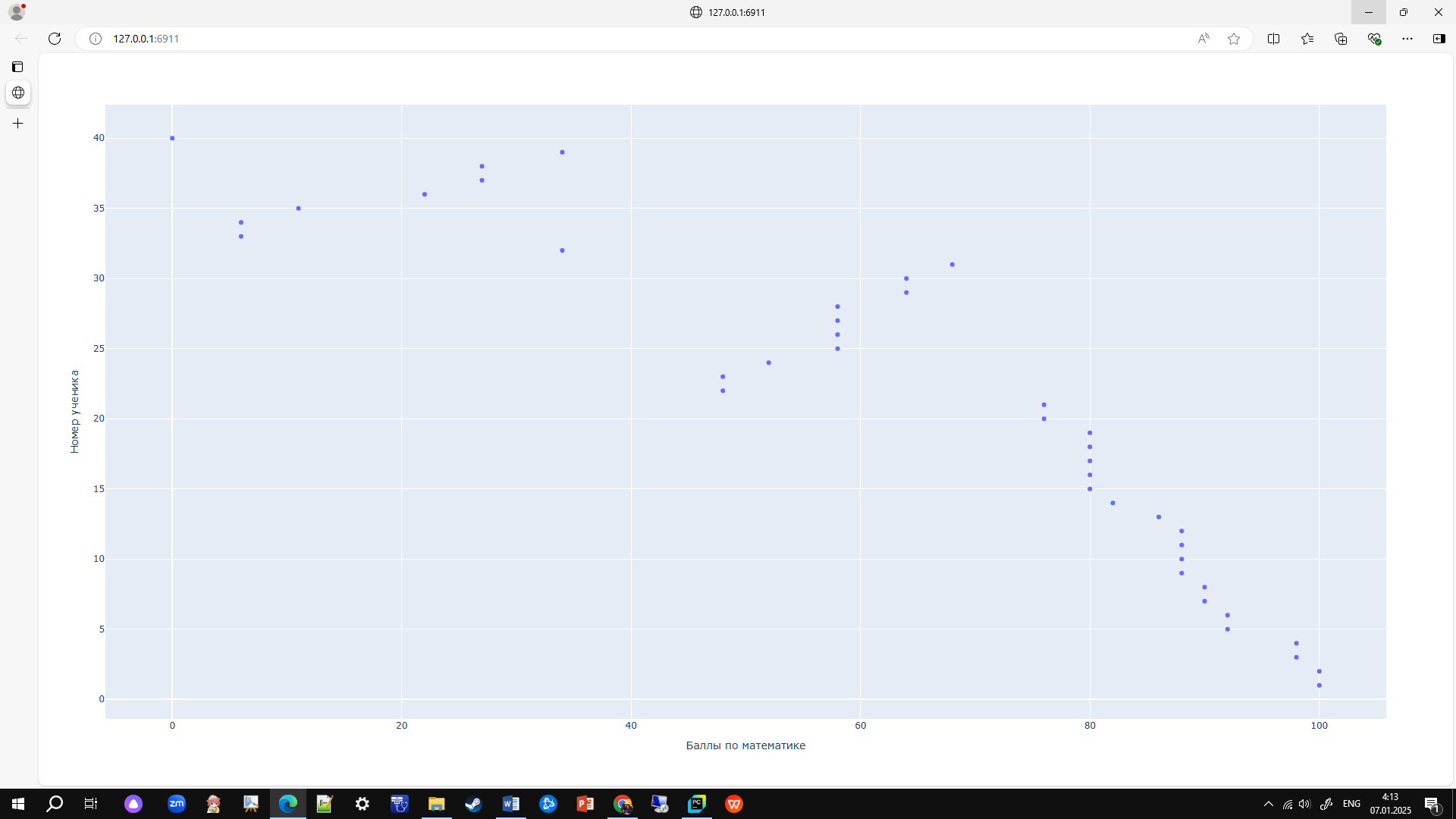
Plotly

Программа 5 (Диаграмма рассеяния):

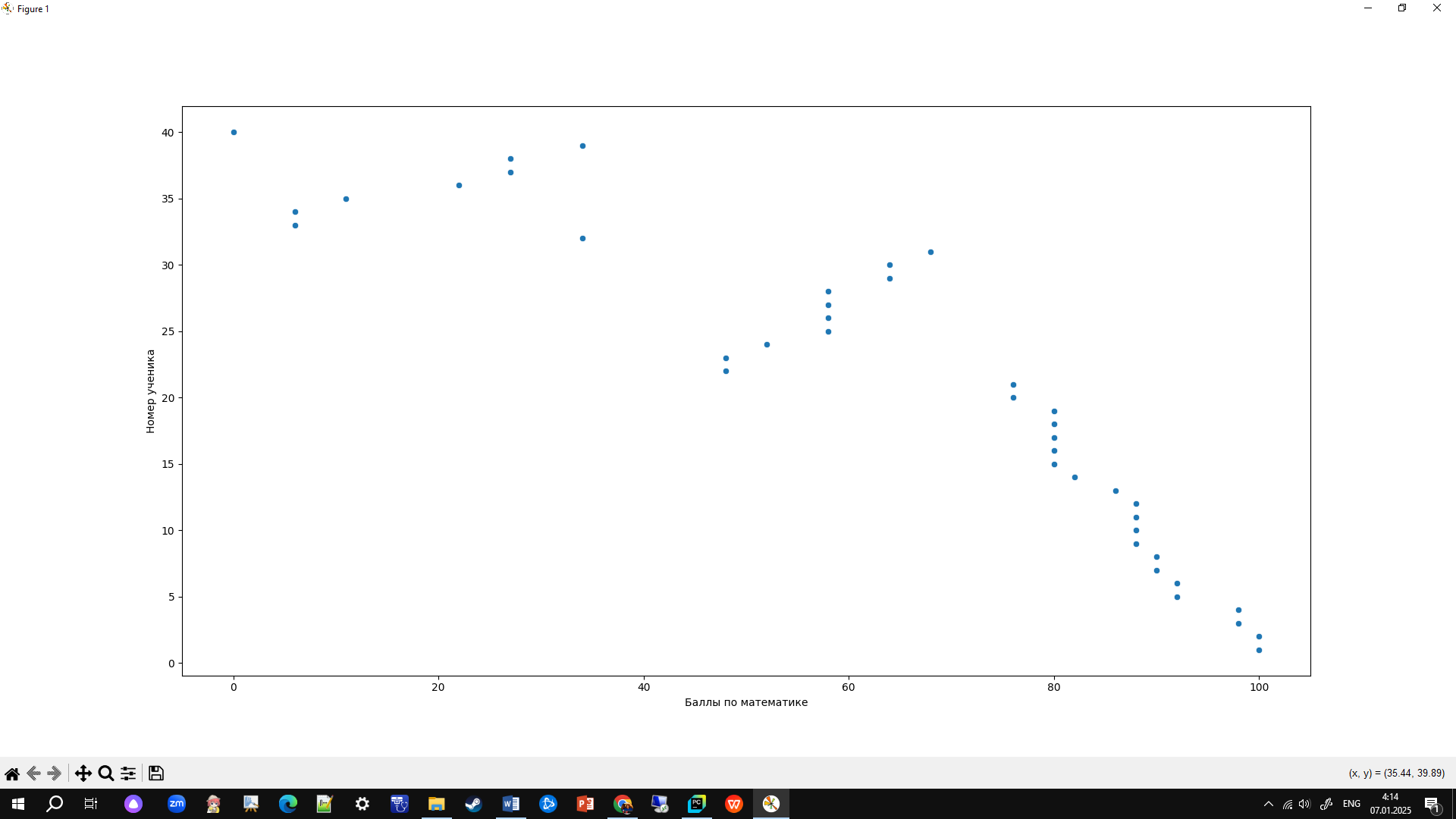


Matplotlib



Plotly



Seaborn

Результаты вышли везде одинаковые, все данные были вписаны верно и код работает корректно (все условия соблюдены).

Далее нужно провести анализ программ и сравнение библиотек.

7. Анализ и интерпретация результатов

Анализ библиотек:

Plotly:

Диаграммы и графики, созданные с помощью этой библиотеки, выглядят лучше, чем на двух других. Здесь визуализированные данные использовать удобнее всего, часто около диаграммы появляется легенда с обозначениями (автоматически). Также ориентация в диаграммах здесь самая удобная. Использовать команду show() можно напрямую через библиотеку Plotly, то есть для полноценной работы библиотеки другие не требуются (не считая математических Pandas/Numpy). Единственный недостаток библиотеки чуть более сложный синтаксис по сравнению с Matplotlib и Seaborn.

А главное и самое хорошее что есть в библиотеке это возможность точечно смотреть данные на диаграмме и также на диаграмме/графике можно посмотреть все нужные данные (те, которые были использованы для создания фрагмента).

Но также нельзя забывать о количестве и разнообразии видов визуализации данных в Plotly. Здесь представлены все возможные варианты от обычных линейных графиков до анимации нужной в самых сложных ситуациях и требующей определенного вида данные.

Если бы у меня была возможность выбора библиотеки для визуализации данных при работе над проектом, где мне потребуется показывать и использовать данные на диаграмме. Но здесь не так много функций в сравнении с Matplotlib, т.е. меньше способов изменить или что-то добавить в графике/диаграмме, что уменьшает функционал.

Из выше перечисленного, Plotly удобен в web проектах и в те моменты, когда нужно удобно, понятно и быстро взаимодействовать непосредственно с графиком/диаграммой.

Matplotlib:

В этой библиотеке не такой хороший визуал и нет такой наглядности в сравнении с Plotly. Но в библиотеке в разы больше способов изменения своего графика/диаграммы, есть больше изменяемых свойств и параметров. В библиотеке много функций и она работает автономно от других библиотек (не считая математических Pandas/Numpy).

Из недостатков можно назвать низкую интерактивность и не очень красивый внешний вид (сравнивая с Plotly). Так же можно назвать недостатком меньшее количество графиков и отсутствие анимационной визуализации данных. А самой большой проблемой является тяжелое управление и сложность освоения библиотеки.

Но есть и плюсы. Из основных: Графики выглядят просто, что помогает сконцентрироваться на данных (если исследовать большие наборы данных это удобней чем усложненные и интерактивные графики/диаграммы). Большое количество изменяемых элементов и возможностей корректировки внешнего вида графика могут помочь улучшить и упростить понимание взаимосвязей между визуализированной информацией.

Эта библиотека очень полезна при анализе больших данных и не предполагает использование в ситуациях, когда нужно узнать чуткую информацию о элементах через график/диаграмму. По большей части Matplotlib нужна для создания изображений.

Из выше перечисленного, Matplotlib удобен для сравнения больших объёмов данных. Это может пригодиться для презентаций или научных работ. Так как там требуется разбор тысяч или даже миллионов единиц данных.

Seaborn:

Seaborn основана на Matplotlib и можно сказать исправляет некоторые проблемы библиотеки. Она упрощает использование и позволяет пользователю заниматься визуалом, меньше думая о сопутствующих для создания графики моментах. Графики/диаграммы в Seaborn выглядят лучше графиков из Matplotlib, но хуже графиков Plotly. Библиотека достаточно многофункциональна и приятна в использовании.

Из недостатков можно назвать зависимость от Matplotlib для базового функционала, из-за чего нужно подгонять их под версии друг друга. Здесь так же нет анимационного варианта представления данных. А также Seaborn не такая гибкая в сравнении с Matplotlib.

Из плюсов можно назвать простоту использования и высокоуровневый интерфейс. Красивые графики по умолчанию в отличии от Matplotlib. Очень хорошая интеграция с pandas DataFrame (специальный вид данных организованных в двух измерениях: строках и столбцах, а также меток, соответствующих им). Поддержка сложной визуализаций без большой нагрузки на компьютер.

Seaborn можно использовать в тех же ситуациях, где используется Matplotlib. Но благодаря улучшенному визуалу графики/диаграммы будут выглядеть лучше и быть приятными глазу. Что дает ей преимущество, но вместе с этим функционал библиотеки меньше и нет такой вариативности.

Из выше перечисленного, Seaborn удобен для сравнения больших объёмов данных. Это может пригодиться для презентаций или научных работ. Так как там требуется разбор тысяч или даже миллионов единиц данных. Но из-за меньшего функционала некоторые моменты возможные в Matplotlibбудут невозможны в Seaborn. Однако визуально графики Seaborn будут выглядеть лучше и красочней.

Интерпретация результатов:

Сравнивая все библиотеки друг с другом, все они полезны если верно выбрать задание и набор данных. Seaborn и Matplotlib, нужны для обработки больших данных и подходят для быстрого и удобного визуального (образного) сравнения в тех ситуациях когда нам первоочередно нужно обобщить результаты и нет разницы какие данные расположены в определенных точках. Plotly же нужна для красивой визуализации для web представления или для интерактивной работы с данными.

В плане самостоятельной работы мне было удобней всего работать с графиками, сделанными в Plotly. Так как визуализация там более наглядна и на графиках/диаграммах можно смотреть данные (наводя мышку на часть нужную для просмотра и/или получения информации). Но Seaborn и Matplotlib тоже были не плохи, и, если бы данных было больше они были бы удобней Plotly.

Рекомендации по выбору библиотек в зависимости от проекта:

Если требуется график/диаграмма для web сайта или имеется маленькое количество данных (от 0 до 500 единиц), нужно использовать Plotly.

Если требуется график/диаграмма для обработки огромного количества данных и не требуется хорошего внешнего вида, но нужно много параметров для изменения и коррекции нужно использовать Matplotlib.

Если требуется график/диаграмма для обработки огромного количества данных и нужен хороший внешний вид, но не требуется так много параметров для изменения и коррекции нужно использовать Seaborn.

8. Заключение:

Обзор выполненной работы:

Все программы выдали нужный, хороший и главное правильный результат. Все графики вышли корректными и наглядными, но из-за того, что Matplotlib и Seaborn созданы для обработки большого количества данных, графики, созданные с помощью этих библиотек, были менее качественными и слишком минималистичными.

На создание всех программ и изучение библиотек у меня ушло около 5 дней, этот опыт довольно полезен для будущего, так как изучение библиотек в краткие сроки и поиск информации в сжатое время — это умение нужное для качественной работы и эффективности в будущей профессии.

Дальнейшие планы:

В будущем мне бы было интересно изучить больше библиотек для визуализации (например: mplcyberpunk, seaborn-style, plotnine, mpld3, Scikit-learn, Pattern и так далее).

Так же я продолжу изучение Matplotlib, Seaborn и Plotly, ведь все эти библиотеки очень полезны и, если ты их знаешь они могут дать больше возможностей в профессиональной деятельности, что даст возможность легче устроиться на работу.