Общество с ограниченной ответственностью «ЭДЭКС»

Дипломная работа по теме:

**Сравнение различных библиотек для визуализации данных: Matplotlib, Seaborn и Plotly**

Автор: Кошелев Владислав Романович

**ОГЛАВЛЕНИЕ**

[Введение 3](#_Toc184652979)

[1. Введение в визуализацию данных 4](#_Toc184652980)

[1.1. Обзор библиотек для визуализации данных в Python 4](#_Toc184652981)

[2. Методология 7](#_Toc184652982)

[2.1. Визуализации с использованием Matplotlib 8](#_Toc184652983)

[2.1.1. Построение линейного графика. 8](#_Toc184652984)

[2.1.2. Построение столбчатой диаграммы. 10](#_Toc184652985)

[2.1.3. Построение круговой диаграммы. 11](#_Toc184652986)

[2.2. Визуализации с использованием Seaborn 12](#_Toc184652987)

[2.2.1. Построение линейного графика 12](#_Toc184652988)

[2.2.2. Построение столбчатых диаграмм 13](#_Toc184652989)

[2.2.3. Построение круговых диаграмм 15](#_Toc184652990)

[2.3. Визуализации с использованием Plotly 16](#_Toc184652991)

[2.3.1. Построение линейного графика 16](#_Toc184652992)

[2.3.1. Построение столбчатых диаграмм 17](#_Toc184652993)

[2.3.1. Построение круговых диаграмм 18](#_Toc184652994)

[3. Сравнительный анализ 20](#_Toc184652995)

[3.1. Функциональность библиотек 20](#_Toc184652996)

[3.2. Удобство использования 21](#_Toc184652997)

[3.3. Визуальное качество графиков 21](#_Toc184652998)

[3.4. Примеры применения в реальных задачах 22](#_Toc184652999)

[4. Обсуждение результатов 24](#_Toc184653000)

[4.1. Итоги сравнительного анализа 24](#_Toc184653001)

[4.2. Рекомендации по выбору библиотек 25](#_Toc184653002)

[Заключение 26](#_Toc184653003)

[Список использованных источников 28](#_Toc184653004)

[Приложение 29](#_Toc184653005)

# ****Введение****

**В эпоху больших данных, когда объемы информации растут с каждым днем, визуализация данных становится неотъемлемой частью анализа и интерпретации информации. Эффективные способы представления сложных наборов данных позволяют выявлять тренды, делать прогнозы и помогать в принятии решений в самых различных сферах — от бизнеса до науки и образования. В связи с этим, выбор соответствующих инструментов для визуализации становится ключевым этапом в процессе анализа данных.**

**Существует множество библиотек для визуализации данных в языке программирования Python, среди которых Matplotlib, Seaborn и Plotly являются одними из самых популярных и функциональных. Каждая из этих библиотек обладает уникальными особенностями, которые делают её применимой в разных контекстах. В условиях растущей сложности задач, стоящих перед аналитиками данных, важно понимать, какие инструменты предлагают наилучшие решения для визуализации, анализа и интерпретации данных.**

**Целью данной дипломной работы является сравнительный анализ библиотек визуализации данных Matplotlib, Seaborn и Plotly. Для достижения этой цели необходимо решить следующие задачи:**

* **изучить основные характеристики и функциональные возможности каждой из библиотек;**
* **создать набор визуализаций с использованием каждой библиотеки;**
* **сравнить созданные визуализации с точки зрения их функциональности, удобства использования и визуального качества;**
* **выработать рекомендации по выбору наиболее подходящего инструмента для конкретных задач визуализации данных.**

# 1. Введение в визуализацию данных

Визуализация данных — это ключевой аспект аналитической работы, предоставляющий возможность преобразовывать сложные наборы информации в наглядные визуальные формы, такие как графики, диаграммы и карты. Этот процесс играет важную роль в понимании и интерпретации данных, позволяя пользователям легче выявлять паттерны, тенденции и аномалии, которые могут быть незаметны в агрегированных числовых данных.

С помощью визуализации можно не только анализировать данные, но и эффективно передавать результаты анализа другим заинтересованным сторонам. Графическое представление информации значительно улучшает восприятие и интерпретацию данных, упрощая процесс принятия решений на основе полученных выводов. В условиях большого объема информации, с которым сталкиваются современные организации, необходимость в инструментах, позволяющих быстро и наглядно представить данные, становится все более очевидной.

Существует множество подходов к визуализации данных, и выбор конкретного метода зависит от поставленных задач, типа данных и желаемых результатов. Эффективная визуализация помогает не только в научных исследованиях, но и в бизнесе, крупных данных, образовании и медицины, открывая новые горизонты для анализа и представления информации.

## 1.1. Обзор библиотек для визуализации данных в Python

Python является одним из самых популярных языков программирования для анализа данных, и существует множество библиотек, предоставляющих инструменты для визуализации. В этом разделе мы рассмотрим три из наиболее используемых библиотек: Matplotlib, Seaborn и Plotly.

**Matplotlib** — это основная библиотека для создания графиков в Python, которая предоставляет широкий спектр возможностей для построения статических, анимационных и интерактивных графиков. Она была разработана в начале 2000-х годов, и с тех пор ее функциональность значительно расширилась. Matplotlib позволяет создавать различные типы графиков, включая линии, столбцы, круговые диаграммы и многое другое. Одна из основных особенностей Matplotlib — это гибкость в настройке графиков. Пользователи могут детально управлять всеми аспектами визуализации, включая размеры, цвета, метки и легенды.

**Seaborn** — это библиотека, созданная на основе Matplotlib, предлагающая более высокоуровневый интерфейс для визуализации данных. Она упрощает создание сложных визуализаций и делает акцент на статистических графиках. Seaborn поставляется с набором встроенных стилей и палитр, что позволяет быстро получать эстетически привлекательные графики. Библиотека также предлагает функции для визуализации корреляций, распределений и других статистических значений, что делает ее идеальным выбором для анализов на основе статистики.

**Plotly** – это библиотека, которая выделяется на фоне других библиотек за счет своей способности создавать интерактивные графики. Это делает Plotly особенно полезным для веб-приложений и презентаций, позволяя пользователям взаимодействовать с данными в режиме реального времени. Plotly поддерживает широкий набор графиков — от простых до сложных 3D-объектов. Удобный интерфейс и возможность генерации графиков, работающих в браузере, делают Plotly популярным выбором среди разработчиков и аналитиков.

В литературе представлено множество исследований, посвященных различным аспектам визуализации данных, включая сравнение библиотек и инструментов. Некоторые работы фокусируются на производительности различных библиотек, другие акцентируют внимание на эстетических или функциональных качествах визуализаций. Однако не так много исследований охватывают комплексный анализ функциональности и удобства использования Matplotlib, Seaborn и Plotly, что подчеркивает актуальность данной работы. Сравнение этих библиотек на основе практического применения и пользовательского опыта поможет в дальнейшем выборе инструмента для специфических задач визуализации данных.

# 2. Методология

Для проведения сравнительного анализа библиотек визуализации данных Matplotlib, Seaborn и Plotly необходимо использовать набор данных, который позволяет продемонстрировать разнообразие возможностей каждой из библиотек.

Для проведения качественного сравнительного анализа будет разработан набор критериев, по которым планируется оценивать функциональность и удобство использования каждой из библиотек.

К основным критериям относятся:

* функциональность: возможности создания различных типов графиков и визуализаций;
* удобство использования: простота и интуитивность интерфейса;
* качество визуализаций: эстетические и функциональные параметры графиков;
* поддержка и документация: наличие обучающих материалов и активность сообщества;
* применимость к реальным задачам: насколько библиотека подходит для решения практических задач анализа данных.

Эти критерии будут использоваться для систематизации результатов анализа и формирования выводов о том, какая библиотека лучше подходит для различных сценариев использования.

Строить графики будем на основе данных отдельных параметров финансовой отчетности Microsoft Corporation с 2011 по 2024 года, за основу будем брать следующие поля:

* год (year);
* выручка (earnings);
* чистая прибыль (revenue);
* валовая прибыль (gross\_profit).

## 2.1. Визуализации с использованием Matplotlib

В данном разделе подробно рассмотрим процесс создания графиков с использованием Matplotlib, настройки их внешнего вида и параметров, которые помогут пользователю увидеть, какие аспекты визуализации можно изменить с помощью этой библиотеки.

Создадим несколько базовых графиков:

* линейный график;
* столбчатую диаграмму;
* круговую диаграмму.

### 2.1.1. Построение линейного графика.

Первый шаг — это импорт необходимых библиотек и данных для визуализации. Так же мы извлекаем необходимые столбцы из DataFrame и назначаем их переменным на рис.1.

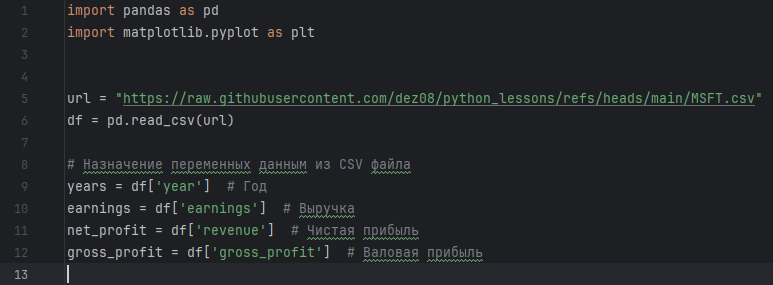


Рисунок 1. Импорт библиотек и данных

Теперь мы создаем линейный график для визуализации финансовых показателей за выбранный период на рис.2. и рис.3.

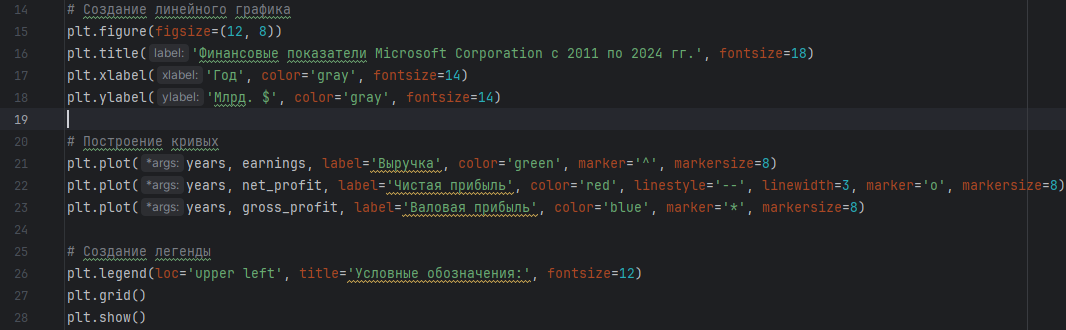
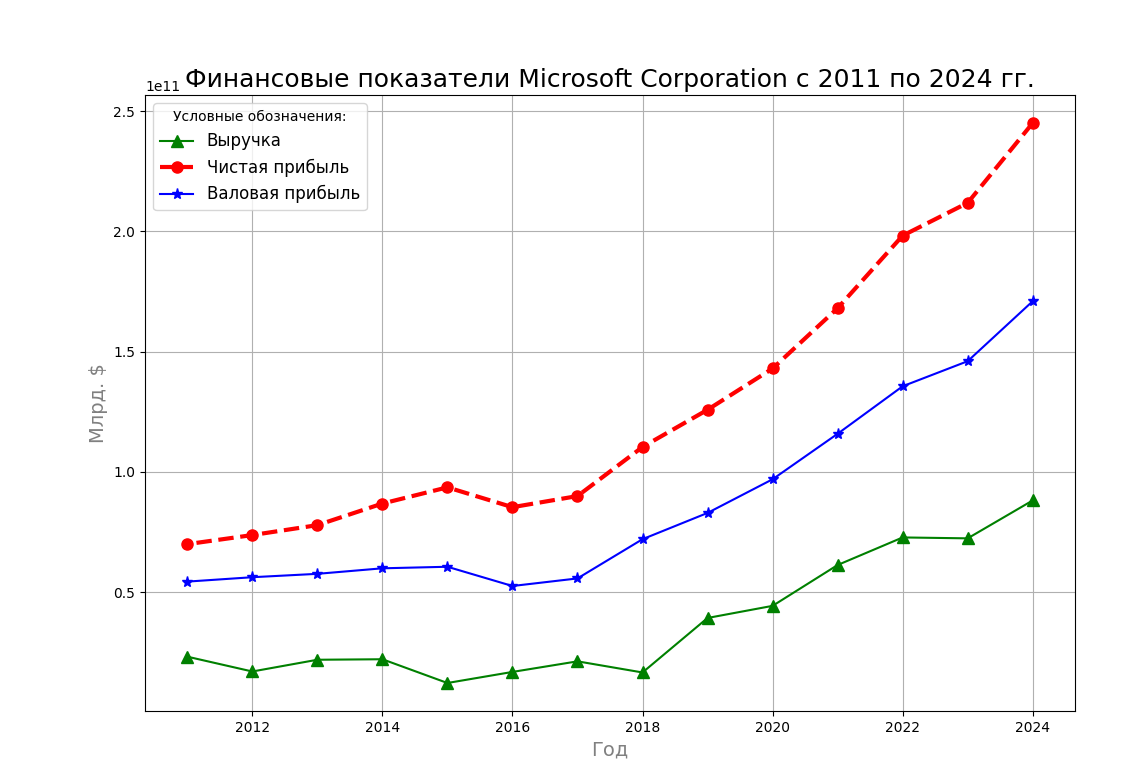


Рисунок 2. Код для создания линейного графика

Рисунок 3. Линейный график с использованием Matplotlib

Линейный график отображает финансовые показатели Microsoft за период с 2011 по 2024 годы.

На оси X расположены годы, а на оси Y — значения в миллиардах долларов.

График содержит три линии:

* зеленая линия для выручки;
* красная линия для чистой прибыли;
* синяя линия для валовой прибыли.

Каждая линия имеет свой маркер и стиль, что упрощает восприятие информации.

### 2.1.2. Построение столбчатой диаграммы.

Создаем столбчатую диаграмму для сравнения финансовых показателей, процесс показан на рис.4. и рис.5.

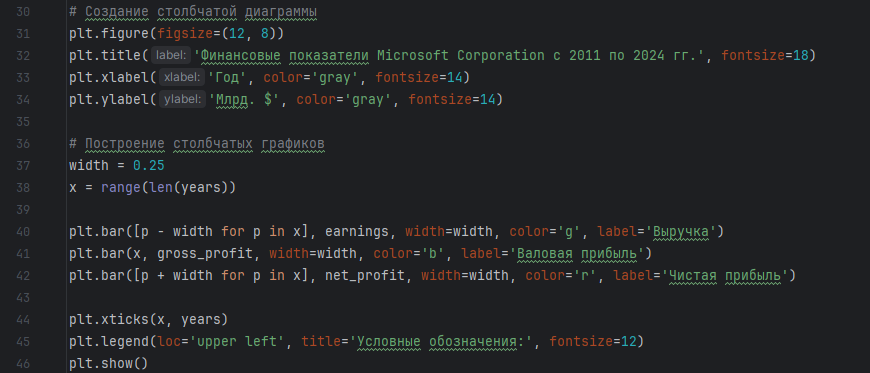


Рисунок 4. Код для создания столбчатой диаграммы

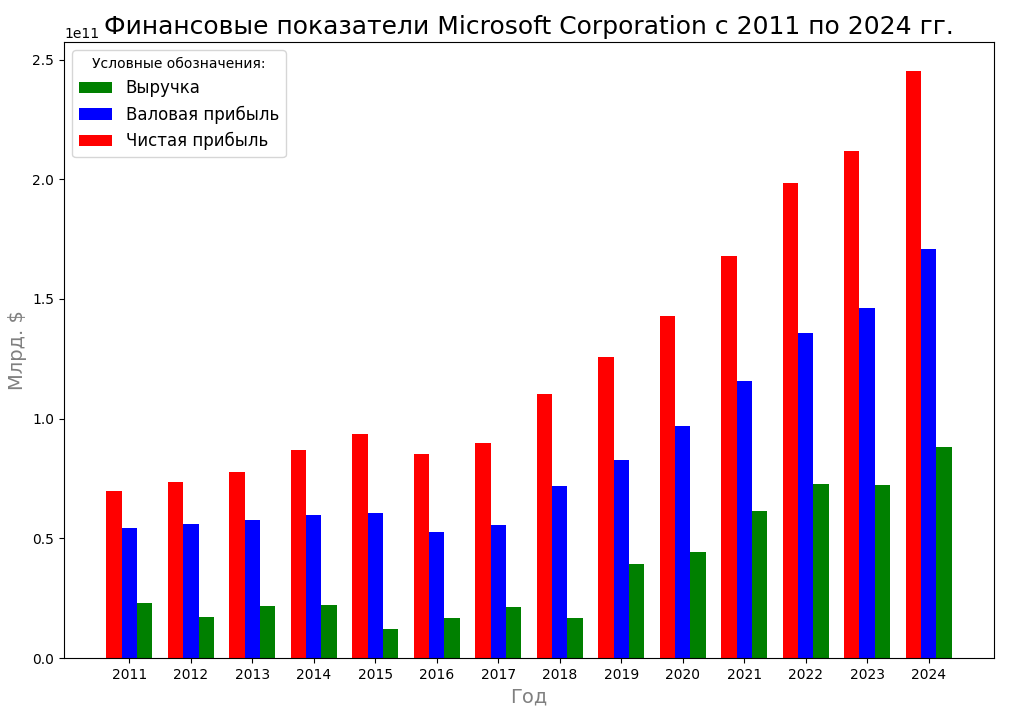


Рисунок 5. Столбчатая диаграмма с использованием Matplotlib

Столбчатая диаграмма позволяет наглядно сопоставлять значения выручки, валовой и чистой прибыли по годам.

Каждая группа столбцов соответствует конкретному году, что облегчает восприятие динамики изменений по показателям.

### 2.1.3. Построение круговой диаграммы.

Теперь мы создаем круговую диаграмму, представляющею доли финансовых показателей на рис.6. и рис.7.

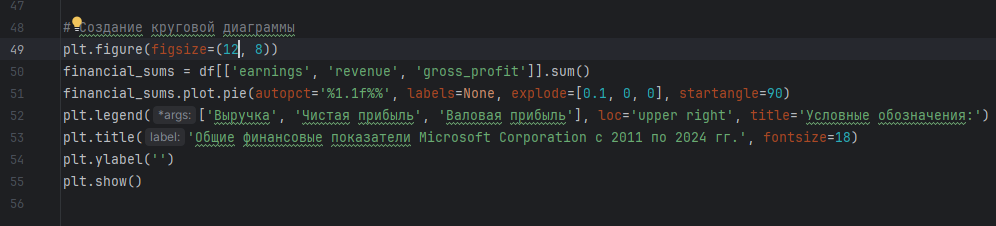


Рисунок 6. Код построения круговой диаграммы

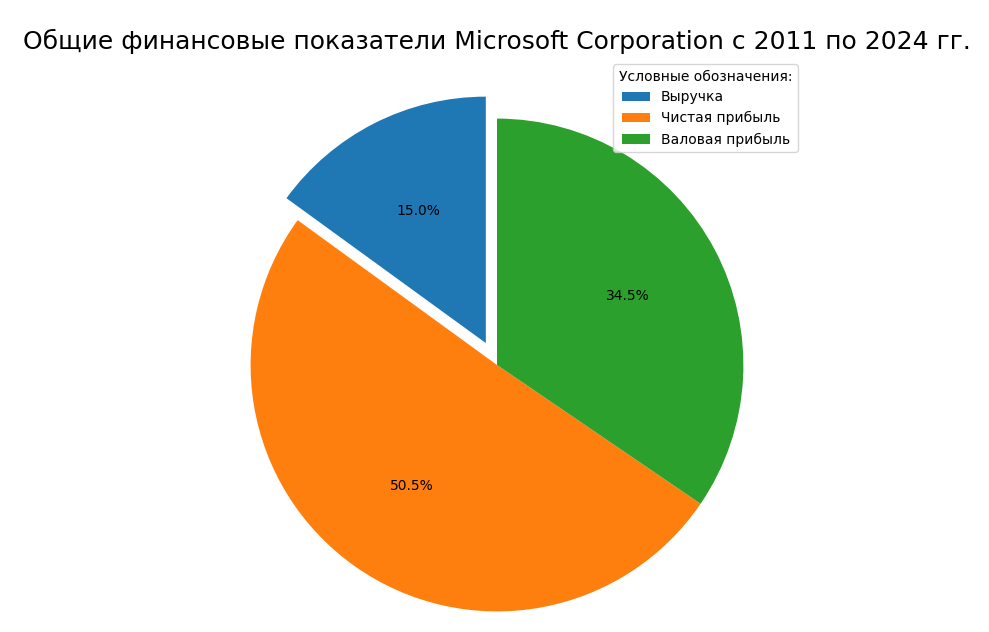


Рисунок 7. Круговая диаграмма с использованием Matplotlib

Круговая диаграмма отображает процентное соотношение между выручкой, чистой и валовой прибылью за весь рассматриваемый период.

Используя параметр autopct, мы можем видеть доли каждого показателя в общем контексте, что помогает лучше понять их взаимосвязь.

## 2.2. Визуализации с использованием Seaborn

Seaborn — это мощная библиотека для визуализации данных, построенная на основе Matplotlib, которая предоставляет высокоуровневые функции для создания эстетически привлекательных и сложных графиков с минимальными усилиями.

В рамках Seaborn мы рассмотрим использование встроенных функций для генерации различных типов визуализаций, включая линейные, столбчатые и круговые диаграммы. Эти графики помогут эффективно представлять данные и выявлять закономерности.

### 2.2.1. Построение линейного графика

Импорт необходимых библиотек и данных для визуализации приведен на рис.8.

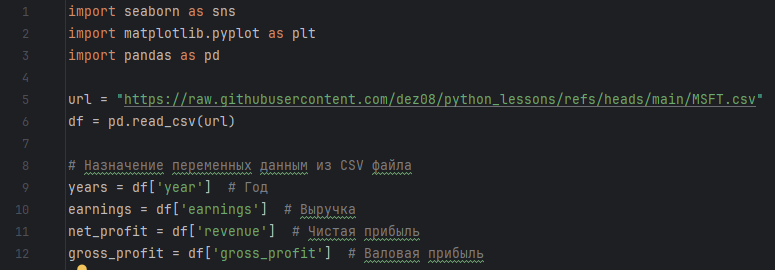


Рисунок 8. Импорт библиотек и данных

С помощью Seaborn мы можем легко создавать линейные графики, отображающие временные ряды и тенденции. Используя функцию sns.lineplot(), мы визуализируем динамику выручки, чистой и валовой прибыли на рис.9. и рис.10.



Рисунок 9. Код построения графика

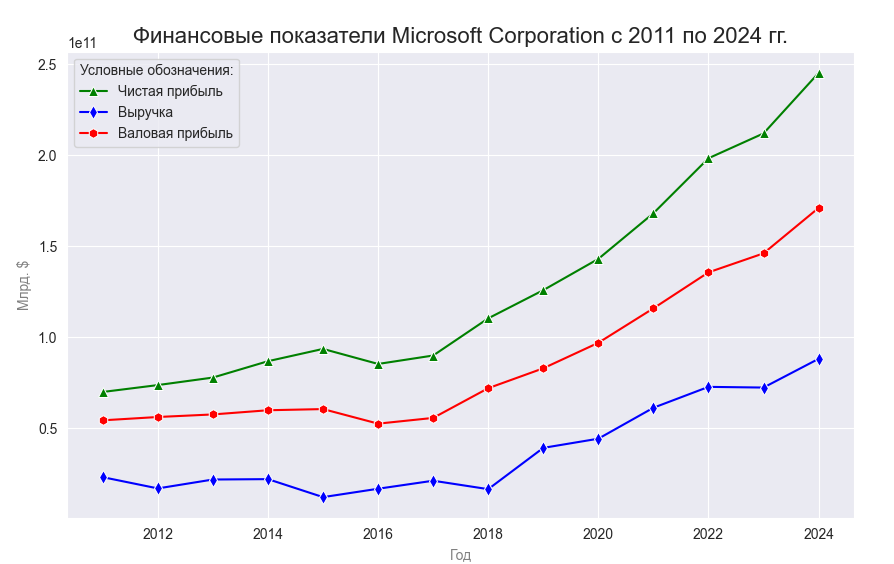


Рисунок 10. Линейный график с использованием Seaborn

Линейные диаграммы в Seaborn позволяют легко добавлять элементы такие как маркеры, стиль линий и цвета, что делает графики более информативными и наглядными. Это помогает быстро выявить тенденции и колебания в данных.

### 2.2.2. Построение столбчатых диаграмм

Seaborn предлагает удобные функции для создания столбчатых диаграмм с помощью sns.barplot(). Мы можем визуализировать финансовые показатели Microsoft по годам следующим образом указанным на рис.11. и рис.12.

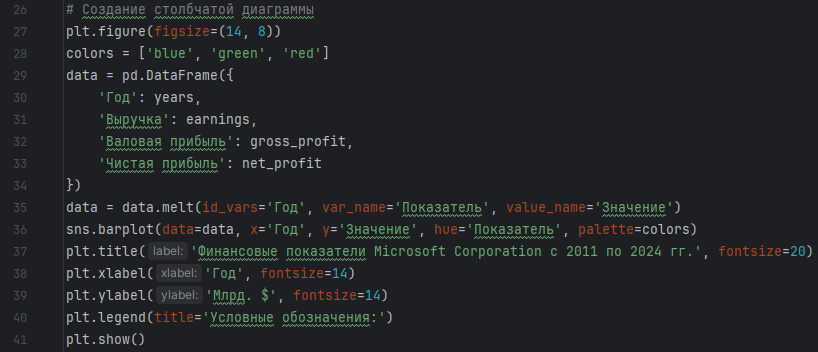


Рисунок 11. Код создания столбчатой диаграммы

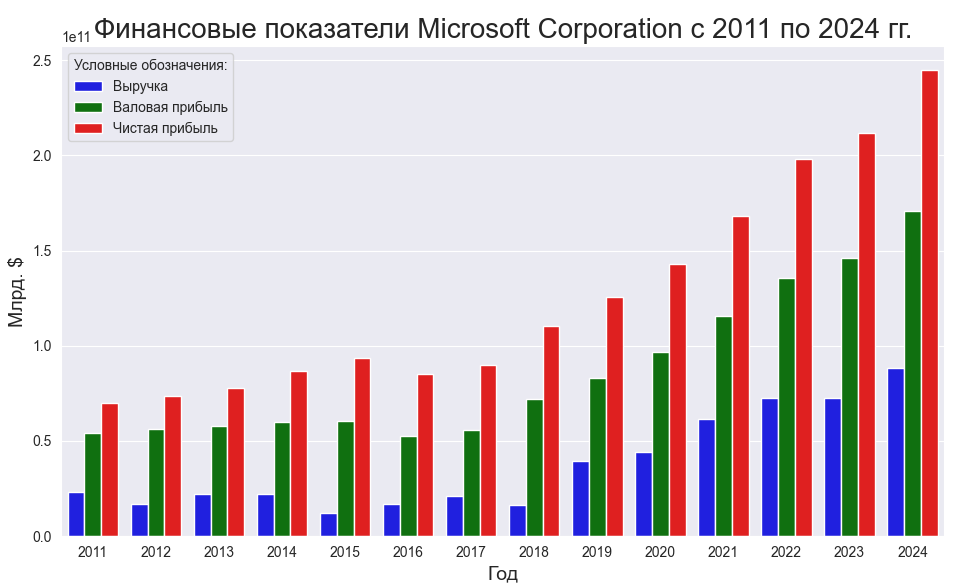


Рисунок 12. Столбчатая диаграмма с использованием Seaborn

При использовании столбчатых диаграмм в Seaborn можно легко настроить цвета и прозрачность столбцов, что позволяет легко сопоставлять различные показатели. Это особенно полезно для сравнения категориальных данных по годам.

### 2.2.3. Построение круговых диаграмм

Seaborn не предоставляет прямых функций для создания круговых диаграмм, однако мы можем использовать Matplotlib для их построения, сохраняя стиль Seaborn. Мы можем использовать sns.set() для настройки визуального стиля на рис.13. и рис.14.

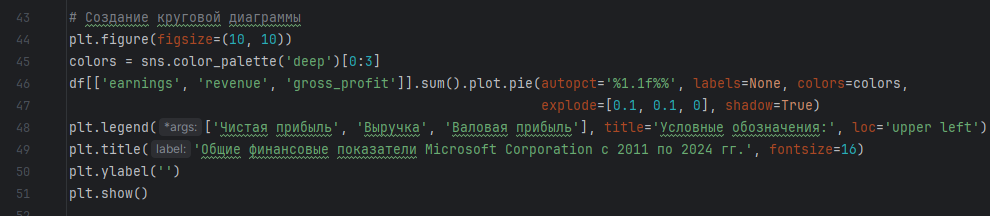


Рисунок 13. Код создания круговой диаграммы

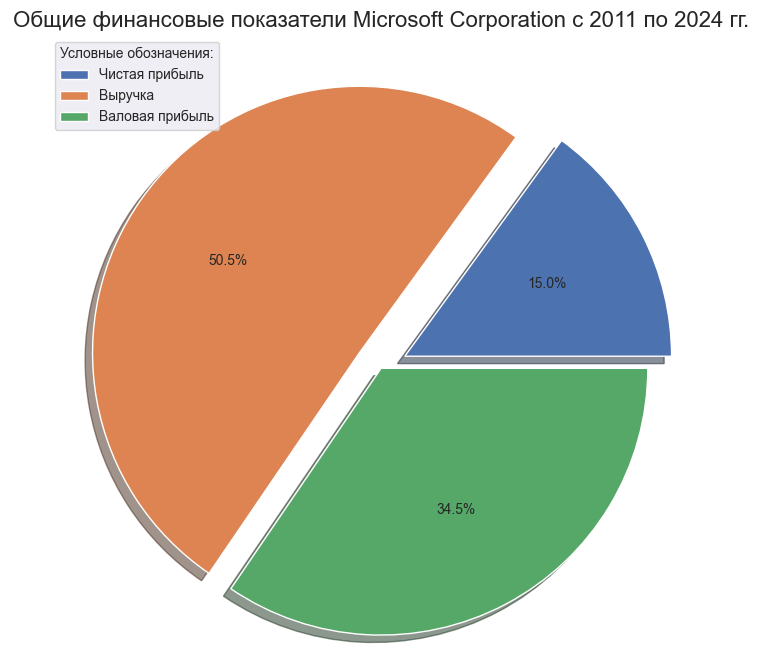


Рисунок 14. Круговая диаграмма с использованием Seaborn

Несмотря на то, что круговые диаграммы обычно создаются с помощью Matplotlib, использование стиля Seaborn может улучшить общую презентацию графика.

Использование Seaborn для визуализации финансовых показателей Microsoft Corporation позволяет создавать наглядные и информативные графики, включая линейные, столбчатые и круговые диаграммы. Линейные диаграммы подчеркивают тенденции, столбчатые диаграммы помогают в сравнении, а круговые диаграммы визуализируют доли, что в целом создает целостное представление о финансовом состоянии компании в исследуемый период.

## 2.3. Визуализации с использованием Plotly

Plotly — это мощная библиотека для создания интерактивных графиков и визуализаций, которые позволяют пользователям исследовать данные более глубоко.

Plotly позволяет создавать интерактивные графики, и в этом разделе рассмотрим примеры создания диаграмм с возможностью взаимодействия. Будут описаны различные типы графиков и их настройки, а также продемонстрированы преимущества интерактивности, которую предлагает эта библиотека.

### 2.3.1. Построение линейного графика

Импорт необходимых библиотек и данных для визуализации на рис.15.

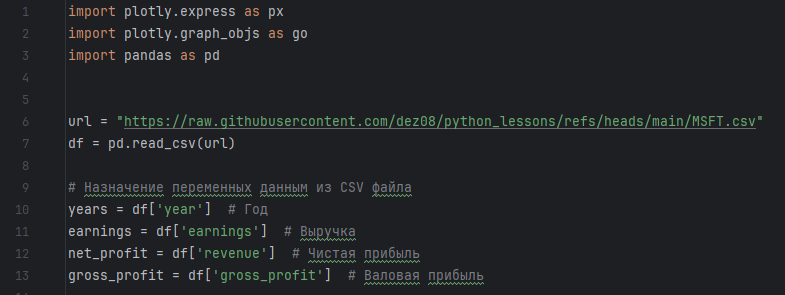


Рисунок 15. Импорт библиотек и данных

Создание интерактивной линейной диаграммы с использованием Plotly позволяет пользователю перемещаться по графику и получать дополнительную информацию о показателях. Пример кода для визуализации динамики выручки, чистой и валовой прибыли на рис.16. и рис.17.

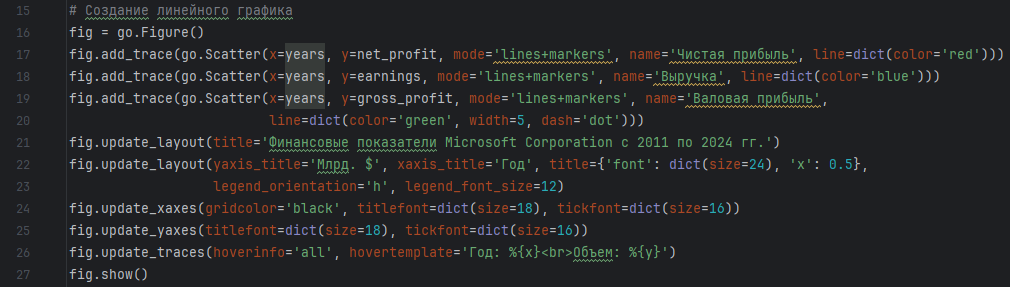


Рисунок 16. Код создания линейного графика

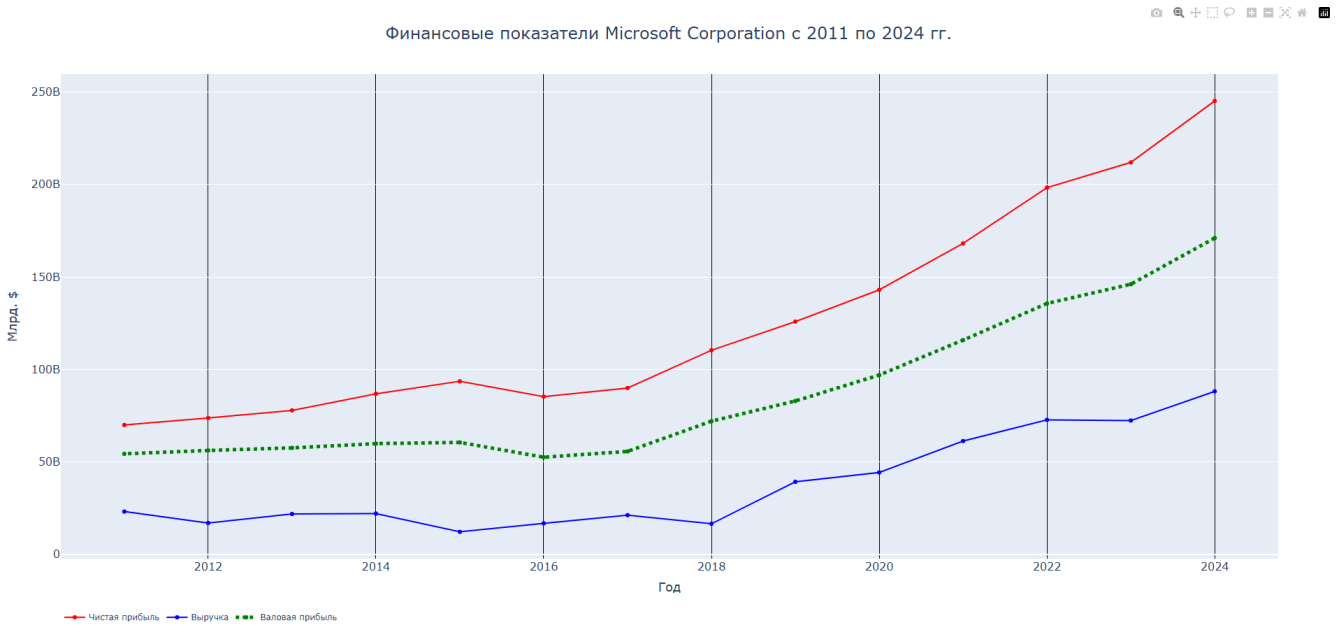


Рисунок 17. Линейный график с использованием Plotly

Этот график позволяет пользователю взаимодействовать с линиями, наводя курсор мыши, и получая информацию о значениях в каждом году.

### 2.3.1. Построение столбчатых диаграмм

Столбчатые диаграммы также можно легко создать с помощью Plotly, обеспечивая интерактивное сравнение финансовых показателей. Вот как можно создать столбчатую диаграмму на рис.18. и рис.19.

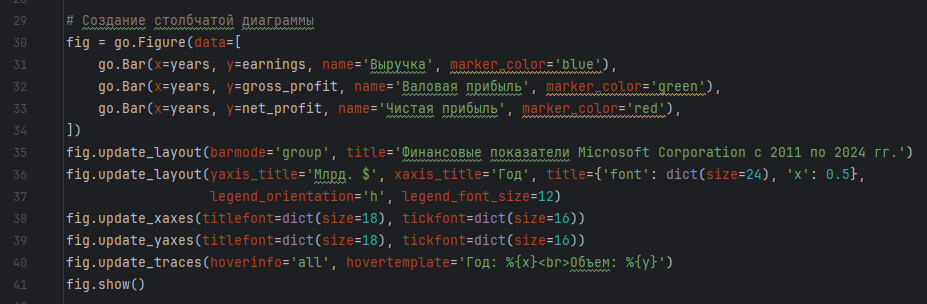


Рисунок 18. Код создания столбчатой диаграммы

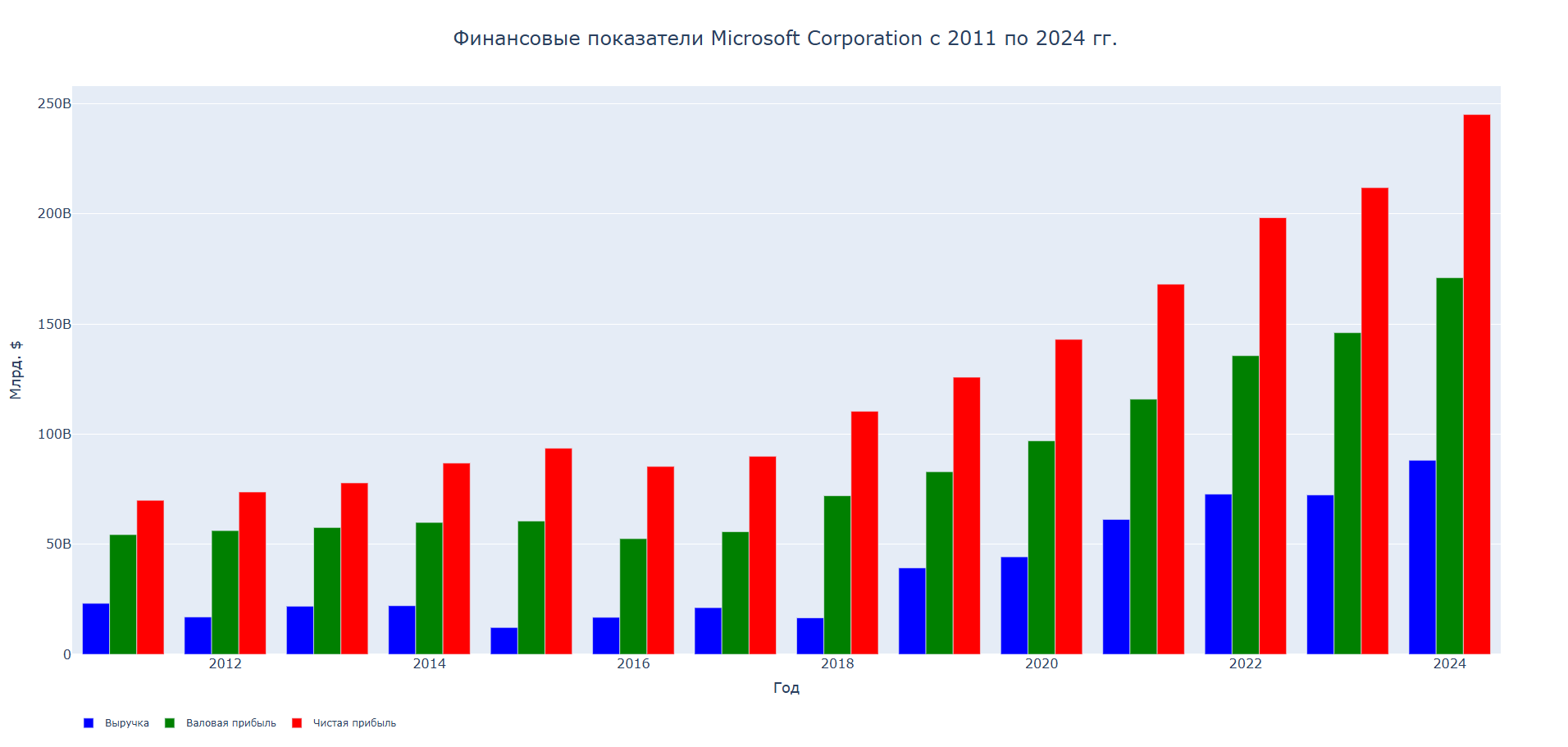


Рисунок 19. Столбчатая диаграмма с использованием Plotly

Группированные столбчатые диаграммы позволяют удобно сравнивать значения для разных показателей в каждом году и дают пользователю возможность взаимодействовать с графиком.

### 2.3.1. Построение круговых диаграмм

Круговые диаграммы в Plotly также обеспечивают интерактивный способ визуализации долей различных показателей, пример реализации на рис.20. и рис.21.

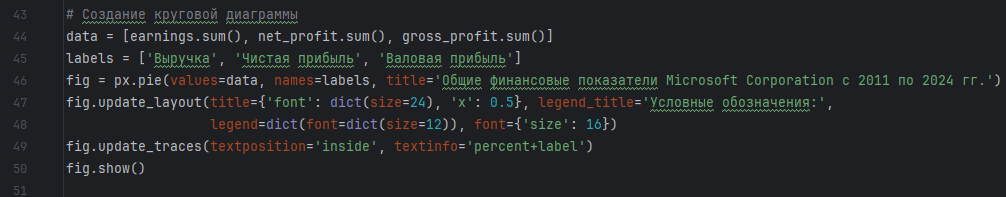


Рисунок 20. Код создания круговой диаграммы

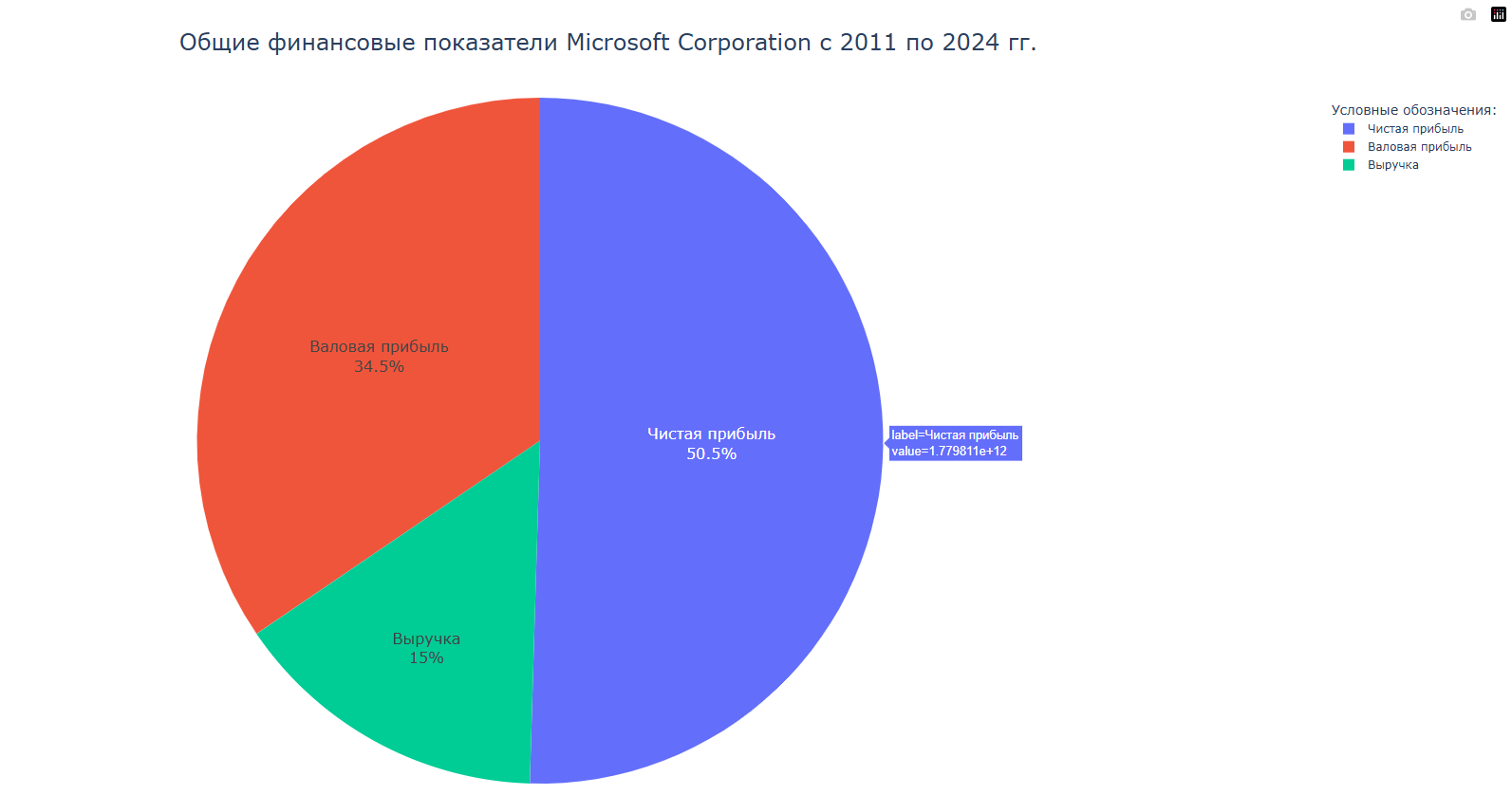


Рисунок 21. Круговая диаграмма с использованием Plotly

Круговая диаграмма отображает доли каждого показателя, позволяя пользователям наводить курсор и получать значение процентов и самих показателей на графике.

Использование Plotly для визуализации финансовых показателей Microsoft Corporation позволяет создавать интерактивные графики: линейные диаграммы для отображения тенденций, столбчатые диаграммы для сравнения и круговые диаграммы для представления долей. Эти визуализации не только делают данные более доступными, но и обеспечивают более глубокое понимание финансовых показателей компании в рассматриваемый период.

# 3. Сравнительный анализ

## 3.1. Функциональность библиотек

В этом разделе мы проведем сравнительный анализ функциональности библиотек Matplotlib, Seaborn и Plotly, рассматривая, какие типы визуализаций каждая библиотека может предложить и насколько они гибки в настройке.

**Matplotlib** обеспечивает широкий спектр возможностей для создания простых и сложных графиков, включая линии, бары, точки и т.д. Она также предлагает глубинные настройки внешнего вида графиков, что делает её мощным инструментом для опытных пользователей. Однако для новичков может быть сложно освоить синтаксис и настройки.

**Seaborn**, построенная на основе Matplotlib, предоставляет более высокоуровневый интерфейс, сосредоточенный на построении статистических визуализаций, таких как коробочные диаграммы, ядерные оценки плотности и парные графики. Она делает акцент на простоте использования и предустановленных стилях графиков, что позволяет быстро создавать эстетически привлекательные визуализации.

**Plotly** выделяется своей способностью создавать интерактивные визуализации. Она идеально подходит для работы с данными, где взаимодействие пользователя имеет значение, позволяя пользователям исследовать графики и получать дополнительную информацию, наведя курсор на элементы. Тем не менее, ограничения могут заключаться в сложности настройки визуализаций в сравнении с другими библиотеками.

Таким образом, выбор библиотеки зависит от конкретных целей и требований пользователя: Matplotlib хорошо подходит для глубокой настройки, Seaborn — для быстрых и стильных статистических графиков, а Plotly — для интерактивности.

## 3.2. Удобство использования

Удобство использования библиотек также является важным критерием для их выбора.

**Matplotlib** может представлять некоторую сложность для новичков из-за своего синтаксиса и необходимости предварительных настроек для достижения желаемого результата. Тем не менее, для пользователей, имеющих опыт работы с ней, она обеспечивает мощные инструменты для комплексного анализа и визуализации данных.

**Seaborn**, в свою очередь, существенно упрощает создание визуализаций за счет упрощенного синтаксиса и интуитивно понятной передачи данных. Пользователи могут сосредоточиться на анализе, а не на технических деталях, что делает эту библиотеку привлекательной для тех, кто хочет быстро визуализировать данные и получить результат.

**Plotly** предлагает простые в использовании функции для создания интерактивных графиков. Однако, полные возможности библиотеки могут потребовать больше знаний и усилий на начальном этапе. С другой стороны, возможность взаимодействия с визуализациями и представления данных в более доступной форме часто оправдывает эту сложность.

Следовательно, выбор подходящей библиотеки зависит от уровня подготовки пользователя и целей проекта в целом.

## 3.3. Визуальное качество графиков

Визуальное качество графиков — это еще один важный аспект, который стоит учитывать. **Matplotlib** позволяет пользователям создавать графики высочайшего качества с точки зрения настраиваемости, т.е. шрифтов, цветовых схем, размеров и прочего. Однако изначально графики могут выглядеть довольно простыми, что требует дополнительной работы для достижения желаемого эстетического уровня.

**Seaborn** обеспечивает более стилизованные графики по умолчанию. Она автоматически применяет лучшее для визуализации цветов и шрифтов, что значительно упрощает процесс создания привлекательных графиков. Это делает Seaborn хорошим выбором для тех, кто хочет быстро создать визуализации с высоким визуальным качеством без больших усилий.

**Plotly** также предлагает высококачественные визуализации, особенно в интерактивном режиме. Краткий интерфейс позволяет пользователям легко добавлять интерактивные элементы, такие как всплывающие окна и анимации, что существенно повышает восприятие графиков и взаимодействие с пользователем. Однако стоит отметить, что визуальное качество может зависеть от уровня сложности графиков.

Итак, если для вас важны как эстетика, так и функциональность, Seaborn и Plotly могут стать предпочтительными вариантами, тогда как Matplotlib стоит рассматривать для глубоких и настраиваемых визуализаций.

## 3.4. Примеры применения в реальных задачах

Каждая из библиотек находит свое применение в различных областях и задачах.

**Matplotlib** часто используется в научной и инженерной среде для создания публикаций, где требуются точные и настраиваемые графики. Например, математики и физики используют её для визуализации сложных данных и графиков функций.

**Seaborn** показывает свою эффективность в области статистики и анализа данных, особенно в ряде библиотек для анализа данных в Python, таких как Pandas. Она позволяет исследователям и аналитикам быстро получать визуальные представления данных, что важно при проведении предварительного анализа и визуализации результатов.

**Plotly** часто применяется в веб-разработке и бизнес-аналитике, где необходима интерактивность. Она позволяет создать дашборды и отчеты, которые пользователи могут активно исследовать, что показывает, как данные меняются в различном контексте. Примеры применения включают интерактивные графики для финансовых данных или анализов продаж.

Таким образом, выбор библиотеки для визуализации данных следует делать, исходя из конкретных задач и контекста их применения, чтобы максимально эффективно использовать доступные инструменты.

# 4. Обсуждение результатов

## 4.1. Итоги сравнительного анализа

В данном разделе рассматриваются ключевые выводы, сделанные в результате сравнительного анализа функциональности, удобства использования, визуального качества графиков и примеров применения библиотек Matplotlib, Seaborn и Plotly.

Первый важный вывод заключается в том, что каждая библиотека имеет свои уникальные сильные и слабые стороны.

**Matplotlib** является наиболее универсальной и мощной библиотекой, обеспечивая широкие возможности по созданию настраиваемых визуализаций. Тем не менее, она требует больше усилий для освоения, что может затруднить ее использование для новичков.

**Seaborn**, в свою очередь, упрощает процесс визуализации, предлагая высокоуровневый интерфейс и предустановленные стили для создания статистических графиков. Это делает её отличным вариантом для быстрой визуализации и анализа данных с акцентом на эстетику.

**Plotly** выделяется своей интерактивностью, что делает ее идеальным решением для поставленных задач, где нужна вовлеченность пользователя и динамичность представления данных. Тем не менее, осваивать Plotly может потребовать больше времени, особенно для пользователей, которые привыкли к статическим графикам.

Итак, выбор подходящей библиотеки зависит от конкретных целей проекта: если требуется гибкость и глубина анализа, лучше использовать Matplotlib; если акцент на быстром создании качественных статистических графиков — Seaborn; для создания интерактивных визуализаций целесообразно выбирать Plotly.

## 4.2. Рекомендации по выбору библиотек

На основании проведенного анализа, можно дать несколько рекомендаций по выбору библиотек для визуализации данных:

1. **Если вы новичок в области визуализации данных:** рекомендуется начать с Seaborn. Его простота в использовании и предустановленные стили помогут быстро создать красивые графики без необходимости изучать сложный синтаксис.
2. **Если вы работаете над проектом, требующим глубокой настройки и сложных графиков:** выбирайте Matplotlib. Эта библиотека позволит вам создавать уникальные и сложные визуализации, полностью адаптированные под ваши потребности.
3. **Если вам нужно создать интерактивные визуализации:** остановитесь на Plotly. Интерактивность этой библиотеки позволит вашим пользователям легче анализировать данные и исследовать различные аспекты визуализации.
4. **При работе в мультидисциплинарной команде:** рассмотрите возможность использовать Seaborn или Plotly, так как оба этих инструмента позволяют быстро создавать визуализации, которые могут быть понятны не только специалистам по данным, но и тем, кто работает в других областях.
5. **При подготовке данных для публикаций:** Matplotlib поможет вам создать графики, соответствующие специфическим требованиям и стандартам научных изданий.

Следуя этим рекомендациям, пользователи смогут выбирать наиболее подходящие инструменты для визуализации данных в зависимости от своих целей, уровня подготовки и контекста проектирования. Это обеспечит эффективность и эффективность проведения анализа данных и последующей визуализации результатов.

# Заключение

В ходе проведенного исследования были подробно рассмотрены три основные библиотеки для визуализации данных в Python: Matplotlib, Seaborn и Plotly. Каждая из них обладает уникальными характеристиками, подходящими для различных задач и уровней подготовки пользователей.

Основные выводы включают:

**Матplotlib** является наиболее универсальной библиотекой, позволяющей создавать разнообразные графики с высокой степенью настройки, но требует определенных навыков и времени для освоения.

**Seaborn** обеспечивает упрощенный интерфейс для создания качественных статистических визуализаций с акцентом на эстетику и простоту использования, что делает её отличным выбором для новичков.

**Plotly** выделяется благодаря своим возможностям по созданию интерактивных визуализаций, что делает его идеальным для задач, где требуется вовлеченность пользователя и динамичность представления данных.

Также было отмечено, что выбор библиотеки должен основываться на специфике задач: если необходима интерактивность, следует выбирать Plotly, если быстрое создание статистических графиков — Seaborn, а для глубоких настраиваемых визуализаций — Matplotlib.

Исследование визуализации данных открывает множество возможностей для дальнейших разработок и улучшений. В будущем можно рассмотреть следующие направления:

1. **Сравнительный анализ новых библиотек:** Появляются новые инструменты для визуализации данных, такие как Altair и Bokeh, которые могут стать альтернативой рассматриваемым библиотекам. Проведение более глубокого анализа их функциональности и удобства использования будет полезным.
2. **Интеграция с другими библиотеками:** Исследование взаимодействия между библиотеками визуализации и библиотеками для анализа данных (например, Pandas и NumPy) для создания более эффективных рабочих процессов.
3. **Обучение и документация:** Разработка учебных материалов и курсов, направленных на обучение пользователей эффективному использованию различных библиотек для визуализации данных и комбинированию их в зависимости от задач.
4. **Применение в новых областях:** Мониторинг и изучение применения библиотек визуализации данных в различных областях, таких как медицина, экология, социальные науки и бизнес, что может помочь в выявлении лучших практик и примеров использования.

Таким образом, перспектива значительно широка, и дальнейшие исследования в области визуализации данных имеют потенциал для улучшения инструментов и повышения их доступности для пользователей различных уровней подготовки и интересов.

# Список использованных источников

1. www.matplotlib.org
2. www.plotly.com
3. www.seaborn.pydata.org
4. https://pythonim.ru/libraries/biblioteka-matplotlib-v-python
5. https://habr.com/ru/companies/otus/articles/540526
6. https://python-school.ru/blog/визуализация-данных/plotly-basics
7. https://marketcap.ru/stocks/MSFT/financial-statements/income-statement
8. Абдрахманов М. И. Библиотека Matplotlib
9. Коритес Б. Графика на Python
10. Python. Визуализация данных

# Приложение

Так же представлю еще несколько вариантов абстрактных графиков из библиотек: Matplotlib, Seaborn и Plotly.

**Matplotlib -** создадим диаграмму рассеяния и 3D график:  
Импорт необходимых библиотек на рис.22



Рисунок 22. Импорт библиотек

Пример кода для визуализации диаграммы рассеяния на рис.23. и рис.24.

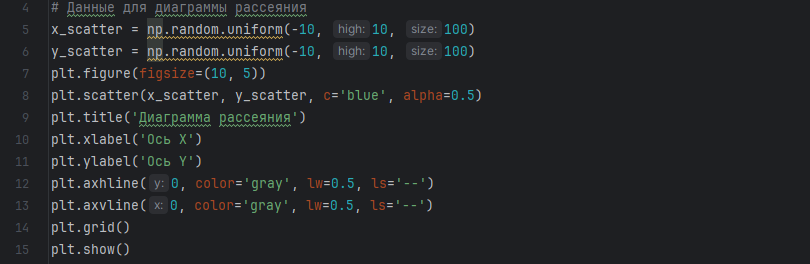


Рисунок 23. Код создания диаграммы рассеяния

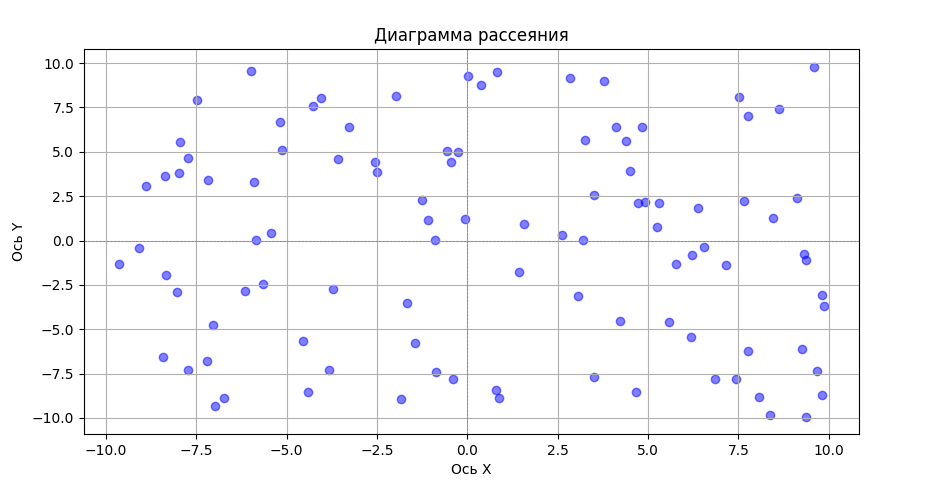


Рисунок 24. Диаграмма рассеяния

Пример кода для визуализации 3D графика параболы на рис.25. и рис.26.

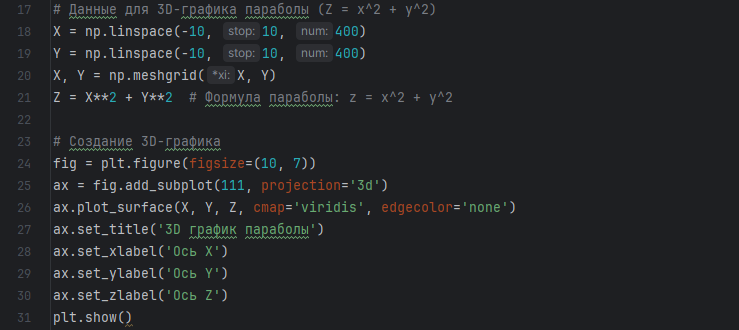


Рисунок 25. Код создания 3D графика параболы

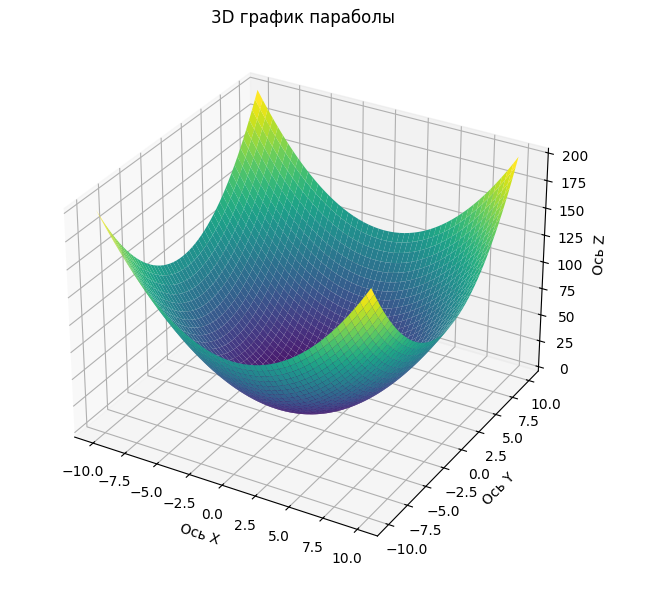


Рисунок 26. 3D график параболы

**Seaborn -** создадим график рассеяния и коробочный график:

Импорт необходимых библиотек на рис.27

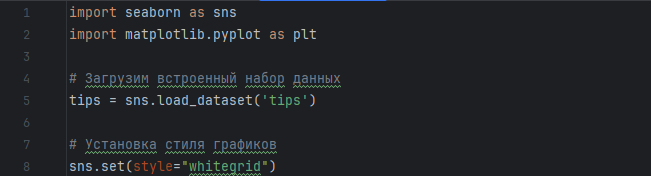


Рисунок 27. Импорт библиотек и данных

Пример кода для визуализации диаграммы рассеяния на рис.28. и рис.29.

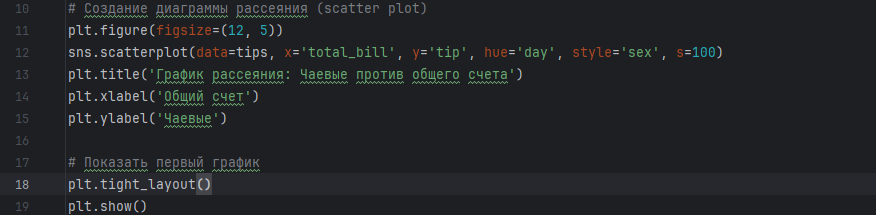


Рисунок 28. Код создания диаграммы рассеяния

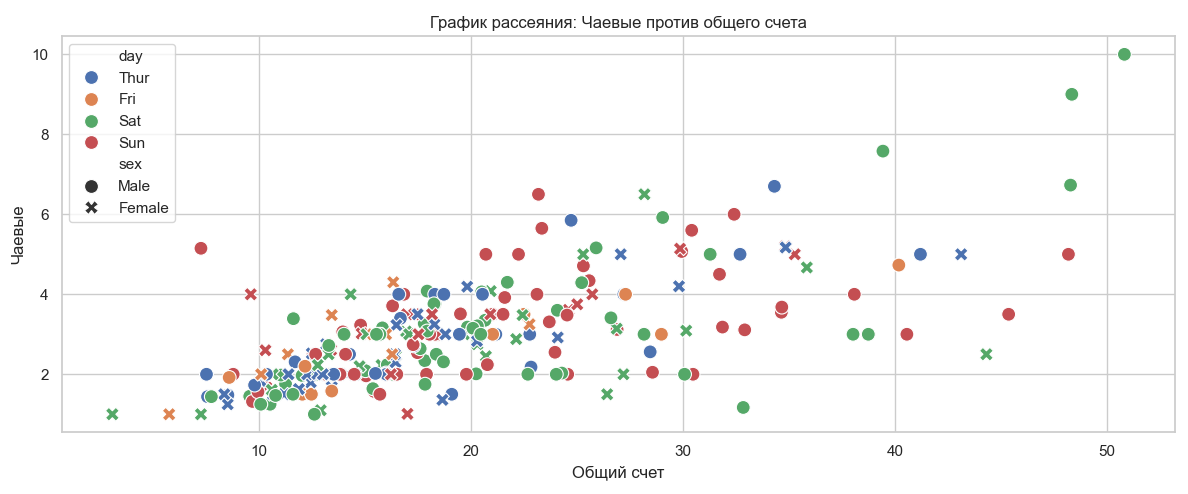


Рисунок 29. Диаграмма рассеяния

Пример кода для визуализации коробочного графика на рис.30. и рис.31.

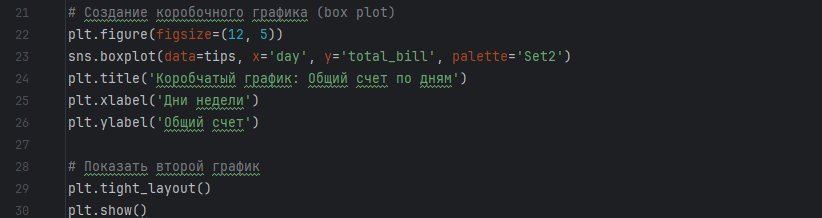


Рисунок 30. Код создания коробочной графика

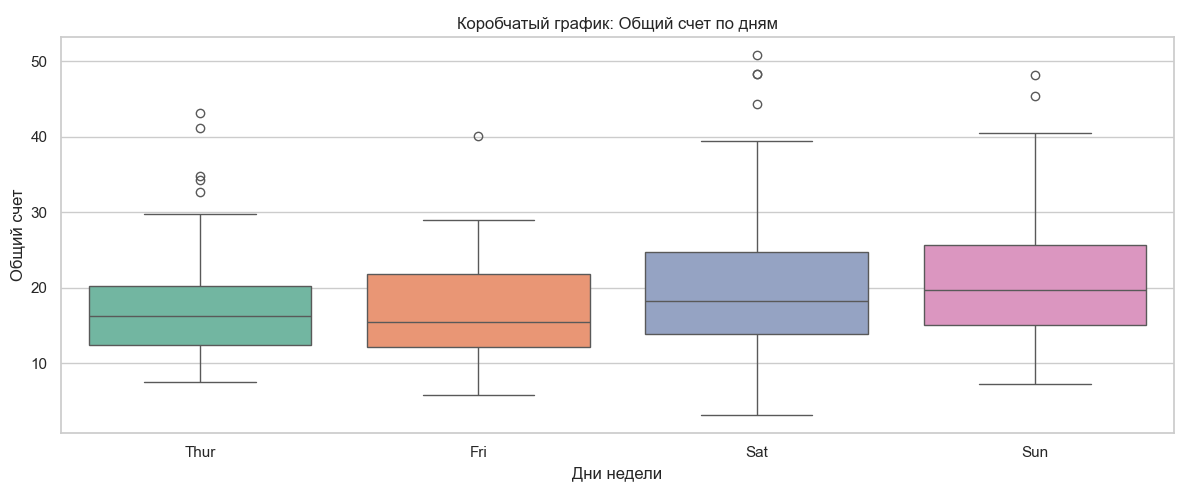


Рисунок 31. Коробочный график

**Plotly** - создадим диаграмму рассеяния и 3D график:

Импорт необходимых библиотек на рис.32



Рисунок 32. Импорт библиотек

Пример кода для визуализации диаграммы рассеяния на рис.33. и рис.34.

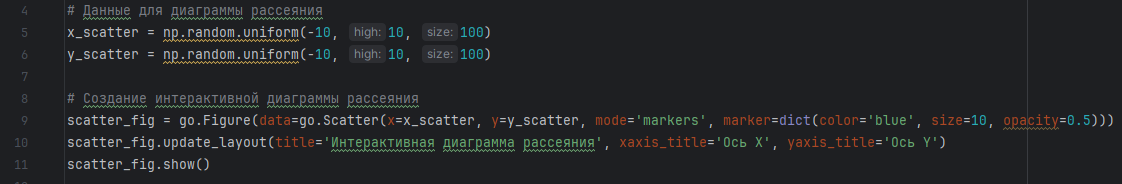


Рисунок 33. Код создания диаграммы рассеяния

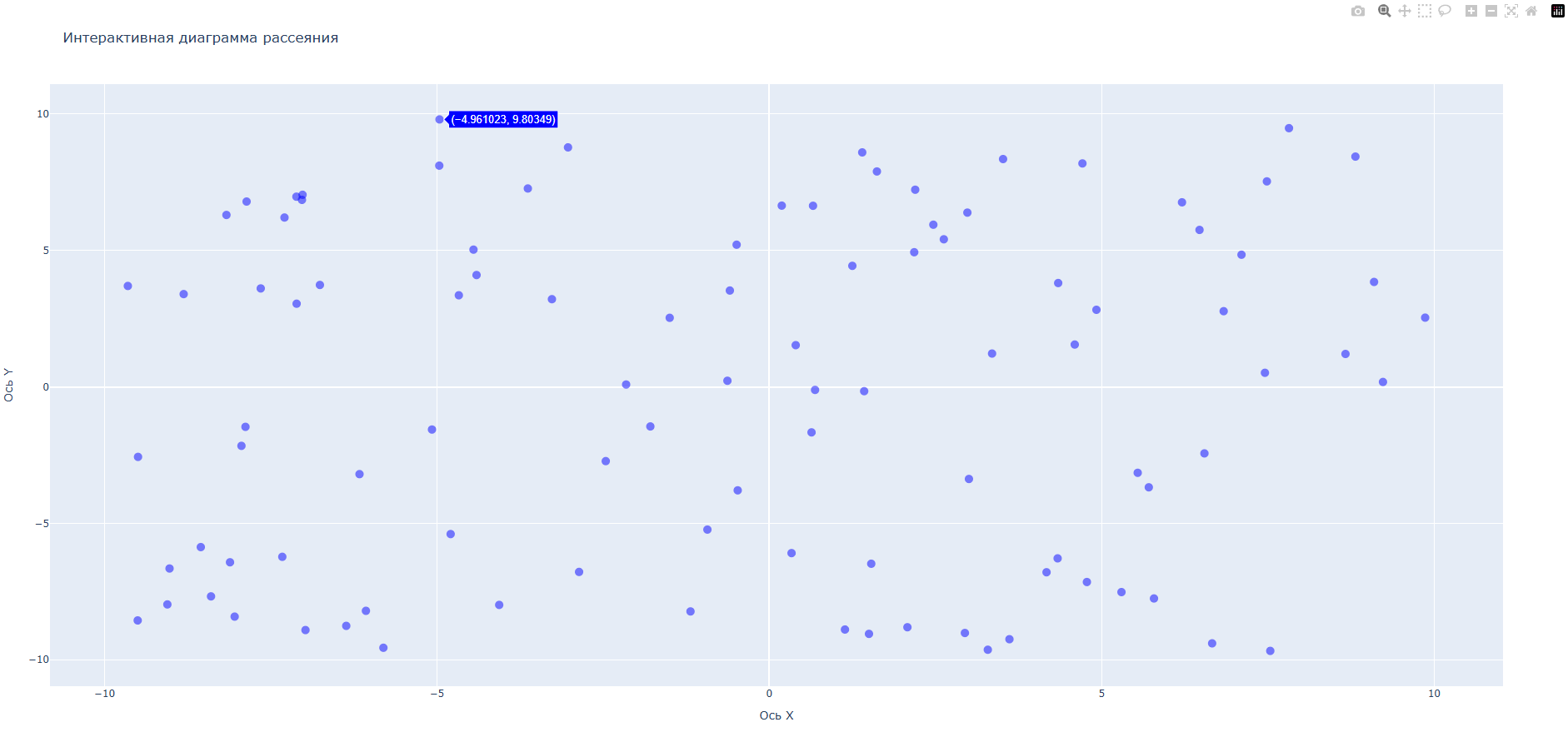


Рисунок 34. Диаграмма рассеяния

Пример кода для визуализации 3D графика на рис.35. и рис.36.

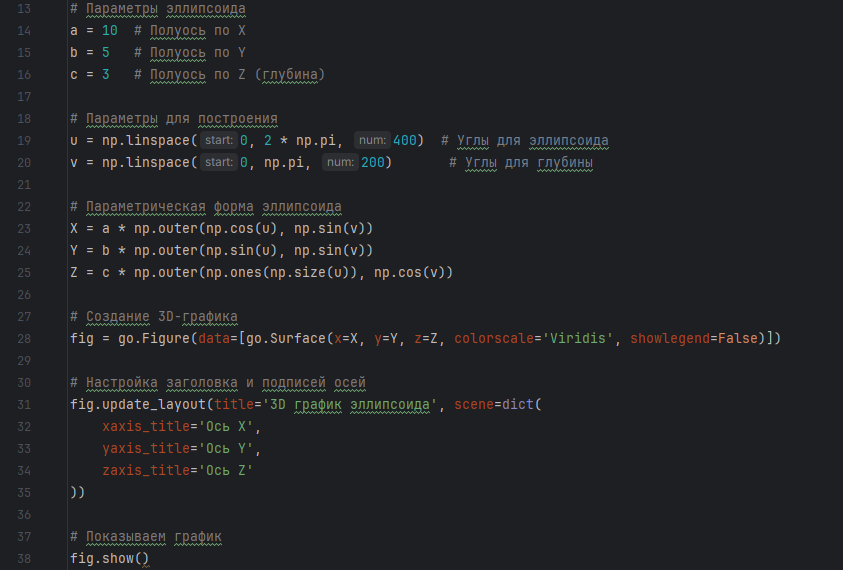


Рисунок 35. Код создания коробочной графика

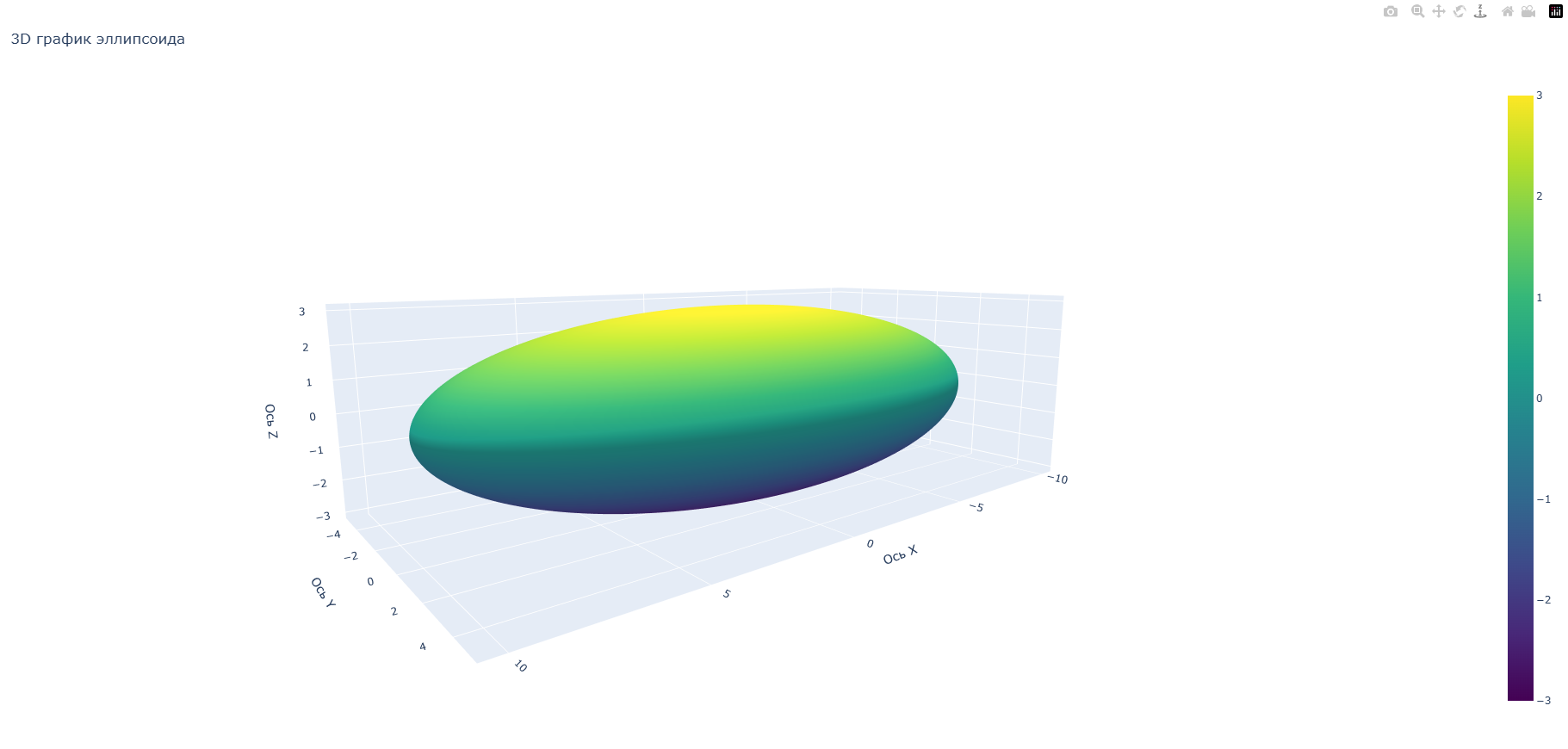


Рисунок 36. Коробочный график