2024 год

ДИПЛОМНАЯ РАБОТА

Сравнение различных библиотек для визуализации данных:

**Matplotlib, Seaborn, Plotly**

Автор:

Барковский Владислав Игоревич

Содержание:

1. Введение. ............................................................................................................................ 2
2. Обоснование выбора темы. ............................................................................................. 2
3. История создания библиотек. .......................................................................................... 4
4. Определение целей и задач исследования. .................................................................. 5
5. Основные характеристики библиотек. ........................................................................... 6
6. Сравнение преимуществ и недостатков. ....................................................................... 9
7. Ключевые различия. ....................................................................................................... 10
8. Сравнительный анализ графических библиотек. ....................................................... 13
9. Рекомендации к выбору графической библиотеки. .................................................. 27
10. Итог. ....................................................................................................................................27

Послесловие. .................................................................................................................... 28

**1. Введение.**

Визуализация данных – это процесс представления информации в виде графиков и диаграмм. Она позволяет преобразовать большие объемы данных в компактные и наглядные изображения, что значительно упрощает их анализ и прогнозирование. Этот метод является ключевым инструментом в науке о данных, так как помогает сделать сложные данные более понятными и доступными для восприятия.

Библиотеки Matplotlib, Seaborn и Plotly являются основными инструментами для создания визуализаций данных при помощи языка программирования Python.

**2. Обоснование выбора темы.**

Тема, которую я выбрал, имеет большое практическое значение для всех, кто работает с данными, будь то их сбор, группировка, анализ, сортировка или выборка. Особенно она важна для тех, кто принимает решения на основании этих данных. Работая в области анализа, важно понимать ценность времени — своего и коллег, а грамотная работа с данными может предоставить максимум полезной информации за минимальное время, тем самым уменьшить время на принятие решения и увеличить вероятность того, что решение будет верным.

Помимо этого, на мой выбор оказали влияние следующие аспекты:

1. **Актуальность темы**:
   * В условиях современного мира количество данных стремительно увеличивается, и возникает острая потребность в их визуализации для проведения анализа и принятия обоснованных решений.
   * Сравнительный анализ популярных библиотек для визуализации данных сегодня особенно востребован.
2. **Практическая применимость**:
   * Работа над этой темой позволит мне глубже изучить возможности и ограничения популярных инструментов для обработки данных.
   * Полученные знания помогут мне выбрать наиболее подходящий инструмент для конкретных задач в будущем.
3. **Глубокое исследование**:
   * Каждая библиотека обладает своими уникальными особенностями и преимуществами, что открывает возможность для детального анализа.
   * Сравнение библиотек выявит сильные и слабые стороны каждой из них.
4. **Применение результатов**:
   * Результаты исследования можно использовать при выборе инструментов для будущих проектов.
   * Эта информация может быть полезна для других исследователей и разработчиков.

Таким образом, выбранная тема не только обогатит меня теоретически, но и предоставит ценные практические навыки по работе с популярными инструментами для визуализации данных.

**3. История создания библиотек Matplotlib, Seaborn, Plotly.**

Чтобы лучше понять назначение этих библиотек, стоит начать с краткого экскурса в историю их создания. Это поможет оценить цели разработки, этапы эволюции и текущее состояние библиотек, что весьма важно при сравнении их возможностей.

**Matplotlib**

Созданная в 2003 году нейробиологом Джоном Д. Хантером (John D. Hunter), эта библиотека изначально задумывалась как аналог MATLAB, но для языка Python. Основная цель Джона заключалась в том, чтобы сделать работу с Matplotlib простой, а сложные задачи — выполнимыми.

Со временем Matplotlib выросла в полноценную библиотеку для визуализации данных. Постепенно добавлялись новые функции для построения графиков и диаграмм, совершенствовались интерфейсы и документация, повышалась производительность при обработке больших объёмов данных. С 2012 года библиотека активно поддерживается сообществом и остаётся основным инструментом для создания статистических графиков.

**Seaborn**

Разработанная в 2010 году Майклом Бетанкуртом (Michael W. T. Betancourt), Seaborn создавалась как дополнение и улучшение возможностей Matplotlib для визуализации данных. Её основная задача — упростить создание привлекательных статистических графиков, используя мощности Matplotlib.

Сегодня Seaborn функционирует как оболочка для Matplotlib и широко применяется в сообществе Python для визуализации данных. Часто её используют совместно с другими библиотеками из экосистем NumPy и SciPy. По умолчанию Seaborn создаёт красивые графики и предоставляет множество функций для анализа статистических данных, что помогает находить связи и закономерности в наборах данных.

**Plotly**

Компания Plotly была основана в 2011 году Джеком Честером (Jack Chester), Крисом Уилкоксом (Chris Wilcox), Мэттью Шарпом (Matthew Sharp) и Алексом Дэвисом (Alex Davis). В тот же год началась разработка одноимённой библиотеки для визуализации данных на языке Python — Plotly.py. Проект возглавил поэт и программист Фил Эйверс, и его целью стало создание интерактивных графиков в Python.

Первая версия разрабатывалась три года, и в 2014 году Plotly.py стала одной из самых загружаемых библиотек для визуализации данных. Однако разработчики не остановились на достигнутом, и в 2016 году вышло крупное обновление — Plotly Dash 1.0.0. В 2017 году Plotly была выпущена как свободная библиотека с открытым исходным кодом.

После этого компания продолжила развивать функциональность библиотеки, выпустив такие продукты, как:

* Plotly.express — для упрощённого создания графиков;
* Dash Enterprise — платная версия Dash с дополнительными возможностями;
* Plotly.js 1.55 — с улучшенной поддержкой.

На сегодняшний день Plotly продолжает активно развиваться, предлагая всё больше возможностей для визуализации данных, включая создание интерактивных графиков, оставаясь одним из лидеров среди аналогичных библиотек для Python.

**4. Определение целей и задач исследования:**

Цель данного исследования заключается в том, чтобы помочь пользователям выбрать подходящую библиотеку среди трёх популярных вариантов — Matplotlib, Seaborn и Plotly — в зависимости от стоящих перед ними задач.

Для достижения этой цели необходимо:

* Определить ключевые характеристики каждой из библиотек,

выделить их особенности, преимущества и недостатки.

* Провести сравнительный анализ функциональных возможностей

всех трёх библиотек на примерах наиболее распространённых видов графиков.

* Исследовать уровень интерактивности и возможности интеграции библиотек в веб-приложения.
* Разработать рекомендации по выбору подходящей библиотеки

в зависимости от специфики решаемой задачи.

**5. Основные характеристики библиотек.**

Не имеет смысла перечислять абсолютно все особенности библиотек, поэтому мы сосредоточимся лишь на главных аспектах, которые позволят лучше понять уникальные черты каждой из рассматриваемых библиотек.

**Ключевые особенности Matplotlib:**

1. **Гибкость и универсальность.**

Matplotlib отличается высоким уровнем гибкости и универсальности:

* + - Предоставляет обширный набор инструментов для создания разнообразных типов графиков и диаграмм.
    - Позволяет легко настраивать и адаптировать визуальные элементы под конкретные нужды.
    - Интегрируется с другими популярными библиотеками Python, такими как NumPy и Pandas.

1. **Простота использования.**

Matplotlib славится своей простотой в использовании:

* + - Предлагает два подхода к созданию графиков: быстрый (через модуль pyplot) и объектно-ориентированный.
    - Быстрый способ подходит для простых графиков, тогда как объектно-ориентированный — для более сложных визуализаций.
    - Имеет интерфейс, похожий на MATLAB, что удобно для тех, кто ранее работал с этим программным обеспечением.

1. **Качество публикаций и широкая применимость.**

Matplotlib широко используется в научной и деловой среде благодаря следующим особенностям:

* + - Позволяет создавать графики высокого качества, подходящие для научных публикаций и презентаций.
    - Применяется во множестве областей, включая научные исследования, бизнес-аналитику и образование.
    - Обладает богатой документацией и активным сообществом разработчиков.

**Ключевые особенности Seaborn.**

1. **Улучшенный интерфейс для статистических графиков.**

Seaborn предоставляет более удобный и визуально привлекательный интерфейс для создания статистических графиков по сравнению с Matplotlib:

* + - Включает готовые функции для создания различных типов графиков, например, countplot, barplot, boxplot и др.
    - Поддерживает интеграцию стилей и цветовых схем, что делает графики более профессиональными.
    - Легко настраивается для изменения параметров визуализации.

1. **Гибкость в работе с категориальными переменными.**

Seaborn особенно полезен при работе с категориальными данными:

* + - Быстро создает графики, отображающие зависимость целевого показателя от нескольких категориальных признаков.
    - Поддерживает различные виды столбчатых диаграмм (grouped, stacked), что удобно для сравнения частот между категориями.
    - Легко добавляет дополнительные признаки для создания вложенных графиков (faceted plots).

1. **Интеграция с Pandas и Matplotlib.**

Seaborn тесно связан с другими популярными библиотеками:

* + - Работает на основе данных из объектов Pandas DataFrame.
    - Использует Matplotlib для рендеринга графиков, обеспечивая высокую производительность и широкие возможности настройки.
    - Упрощает управление стилями и параметрами через rcParams.

Эти особенности делают Seaborn мощным инструментом для создания качественных статистических визуализаций в Python, особенно когда речь идет о работе с категориальными данными.

**Ключевые особенности Plotly.**

1. **Интерактивность и высококачественные графики.**

Plotly известен своей способностью создавать интерактивные и высококачественные графики:

* + - Позволяет строить разнообразные типы графиков, включая линейные диаграммы, точечные графики, столбчатые диаграммы, круговые диаграммы и многие другие.
    - Графики легко настраиваются и адаптируются для достижения нужного внешнего вида.
    - Поддерживает создание сложных визуализаций, включая 3D-графики и интерактивные элементы.

1. **Удобство использования.**

Plotly предлагает удобный и интуитивно понятный интерфейс:

* + - Библиотека включает высокоуровневый API (Plotly Express), позволяющий быстро создавать графики с минимальным количеством кода.
    - Поддерживает различные форматы входных данных, включая объекты Pandas DataFrame, массивы NumPy и xarray.
    - Легко добавляет легенды, метки и другие элементы интерактивности к графикам.

1. **Гибкость и расширяемость.**

Plotly предоставляет широкие возможности для настройки и расширения функционала:

* + - Позволяет создавать сложные графики с большим числом элементов и взаимодействий.
    - Поддерживает создание анимаций и интерактивных элементов, таких как кнопки и ползунки.
    - Имеет встроенную поддержку для создания веб-приложений с использованием Dash, что позволяет разрабатывать полноценные аналитические приложения.

Эти особенности делают Plotly отличным выбором для создания интерактивных и динамичных визуализаций, особенно если требуется высокая степень взаимодействия с пользователем.

**6. Сравнение преимуществ и недостатков.**

Преимущества **Matplotlib**:

* Высокая гибкость и универсальность.
* Широкий диапазон настроек и возможностей кастомизации.
* Поддержка множества типов графиков.

Недостатки **Matplotlib**:

* Сложность освоения из-за непростого синтаксиса.
* Требуется писать больше кода для реализации определённых типов визуализаций.

Преимущества **Seaborn**:

* Улучшенный интерфейс для создания статистических графиков.
* Более привлекательная визуальная составляющая.
* Проще в использовании по сравнению с Matplotlib.

Недостатки **Seaborn**:

* Меньшая гибкость по сравнению с Matplotlib.
* Ограниченный ассортимент типов графиков.

Преимущества **Plotly**:

* Интерактивность графиков.
* Возможность создания сложных визуализаций.
* Хорошая поддержка 3D-графиков и географических данных.

Недостатки **Plotly**:

* Низкая производительность при работе с большими объёмами данных.
* Зависимость от веб-технологий может стать препятствием в некоторых средах.
* Увеличение размера файлов для интерактивных графиков.

**7. Ключевые различия.**

Я выделил пять ключевых различий между библиотеками **Matplotlib**, **Seaborn** и **Plotly**, которые помогут вам выбрать необходимую библиотеку:

1. **Интерактивность**: Plotly > Seaborn > Matplotlib
2. **Простота использования**: Plotly > Seaborn > Matplotlib
3. **Эстетика**: Plotly > Seaborn > Matplotlib
4. **Производительность**: Matplotlib > Plotly > Seaborn
5. **Настраиваемость**: Matplotlib > Plotly > Seaborn

Подробности:

* **Интерактивность**: Plotly предлагает самые интерактивные графики, Seaborn и Matplotlib создают статичные изображения.
* **Простота использования**: Plotly предоставляет высокоуровневый API для быстрой работы, Seaborn легче в использовании, чем Matplotlib, но менее гибкий, чем Plotly.
* **Эстетика**: Plotly имеет современный и привлекательный дизайн по умолчанию, Seaborn улучшает визуальное оформление по сравнению с Matplotlib, но уступает Plotly.
* **Производительность**: Matplotlib быстрее обрабатывает большие наборы данных, Seaborn близок к нему, а Plotly медленнее из-за интерактивности.
* **Настраиваемость**: Matplotlib предоставляет наибольшую гибкость в настройке, Plotly также предлагает хорошие возможности, а Seaborn имеет меньше возможностей для настройки.

Для визуализации этих критериев я создал горизонтальный бар-график с помощью библиотеки Plotly, который позволяет быстро сравнить библиотеки по каждому критерию.

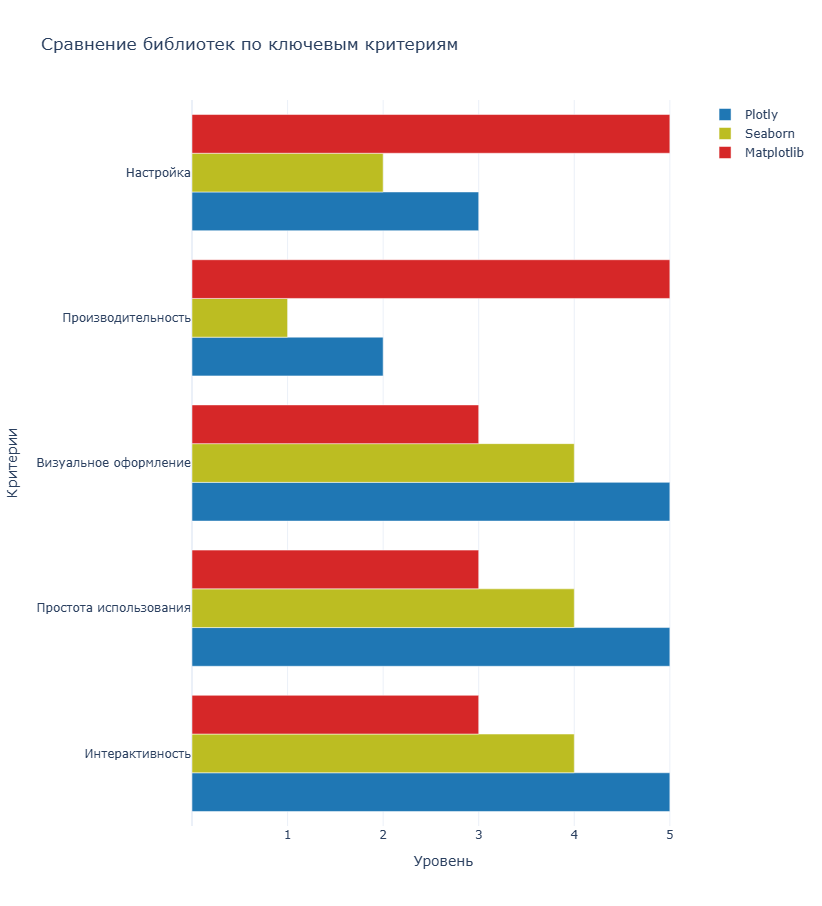
Для его создания я оценил по 5-бальной шкале каждую из библиотек по всем 5 критериям:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Matplotlib | Seaborn | Protly |
| Интерактивность | 5 | 4 | 3 |
| Простота использования | 5 | 4 | 3 |
| Эстетика | 5 | 4 | 3 |
| Производительность | 2 | 1 | 5 |
| Настраиваемость | 2 | 2 | 5 |

Итак, будем считать, что:

Ось X: Уровень (например, от 1 до 5), где 1 — самый низкий, а 5 — самый высокий уровень по каждой из категорий.

* + Ось Y: Категории: интерактивность, простота использования, эстетика, производительность, персонализация.
  + Каждая категория будет иметь три столбца (по одному для каждой библиотеки) с разными цветами для каждой библиотеки (синим для Plotly, зелёным для Seaborn и красным для Matplotlib).



Например, если для нас приоритетным является качество визуального оформления, целесообразно рассмотреть использование библиотеки **Plotly**, тогда как для начинающих пользователей, которым требуется минимизировать сложность настройки, может подойти библиотека **Matplotlib**.

Ниже представлен код для написания данного графика:



Важность визуализации данных невозможно переоценить. На протяжении нескольких страниц были подробно рассмотрены ключевые отличия различных библиотек, что позволило предоставить читателю возможность осознанного выбора наиболее подходящего инструмента.

Несмотря на значительные усилия, затраченные на изучение представленного материала, порой решение принимается интуитивно при первом же взгляде на простой график. Это подтверждает значимость визуальных средств представления информации в процессе анализа данных. Данный пример иллюстрирует мысль, высказанную ранее: визуализация играет ключевую роль в современном мире, являясь мощным инструментом для восприятия и интерпретации сложных данных.

**8. Сравнительный анализ графических библиотек.**

Мы уже рассмотрели описания библиотек на официальных ресурсах, проанализировали различные источники информации и провели сравнительный анализ ключевых характеристик. Однако остался важный аспект, который требует дополнительного внимания — внешний вид создаваемых графиков.

На мой взгляд, этот критерий также заслуживает включения в общий процесс оценки, поскольку конечный результат визуализации часто определяет удобство использования библиотеки для пользователя. Если графики, созданные разными библиотеками, выглядят идентично, а объемы необходимого кода сопоставимы, разработчик может руководствоваться личными предпочтениями, выбирая тот инструмент, с которым ему удобнее работать для быстрой реализации и презентации результатов.

Важно помнить, что разработка кода для построения графика составляет лишь часть задачи. Основная работа заключается в сборе, обработке и подготовке данных (включая группировку, агрегацию, очистку и другие операции), которая занимает значительную долю общего времени выполнения проекта. Соответственно, этап создания самого графика должен занимать минимальное количество времени.

Для объективной оценки предлагаю провести тестирование трёх рассматриваемых библиотек на примере построения одинаковых типов графиков:

* Линейные графики;
* Столбчатые диаграммы;
* Круговые диаграммы.
* Особенные графики.

Несмотря на схожесть итоговых изображений, каждая библиотека обладает уникальными особенностями в области стилистики и функционала, которые могут оказать влияние на выбор конкретного инструмента для решения задач визуализации.

Для проведения тестов я воспользовался готовыми наборами данных с сайта **Kaggle.com**.

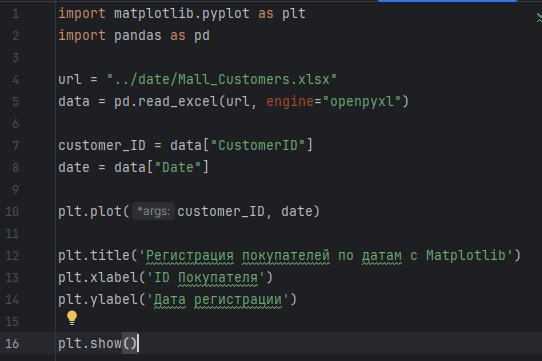
Во избежание казусов во время тестирования кода, я отформатировал скачанные данные в файл формата **Excel**. Файлы с данными так же сохранены в папке проекта на **GitHub**.

1. **Линейный график.**

Линейные графики применяются для визуализации данных, изменяющихся во времени, демонстрируя зависимость одной переменной от другой. Они особенно эффективны для анализа временных рядов, таких как: динамика цен, температурные колебания, изменения объемов продаж. Эти графики позволяют выявлять цикличность, сезонность и прочие тенденции.

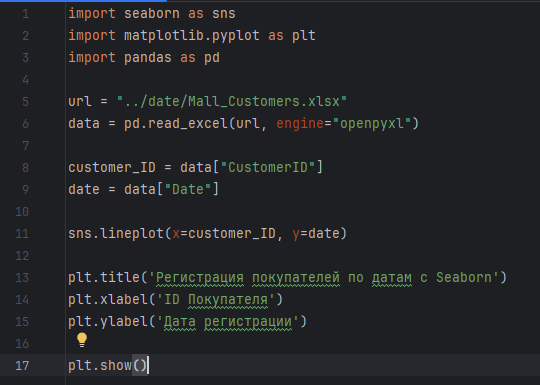
Для проведения исследования была использована база данных, содержащая информацию об идентификаторах покупателей и датах их регистрации. Целью анализа являлось определение дней с наибольшим количеством регистраций, а также периодов с наименьшей активностью пользователей.

**Matplotlib.**



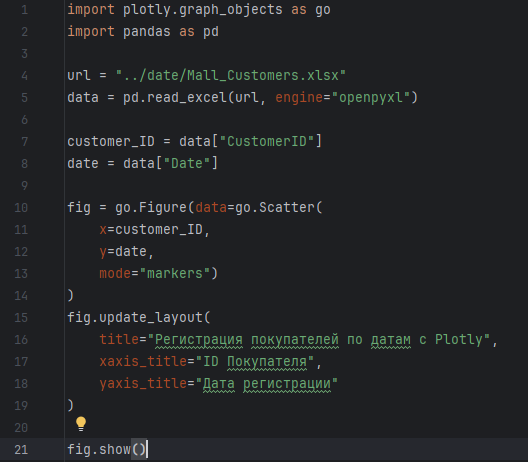
****

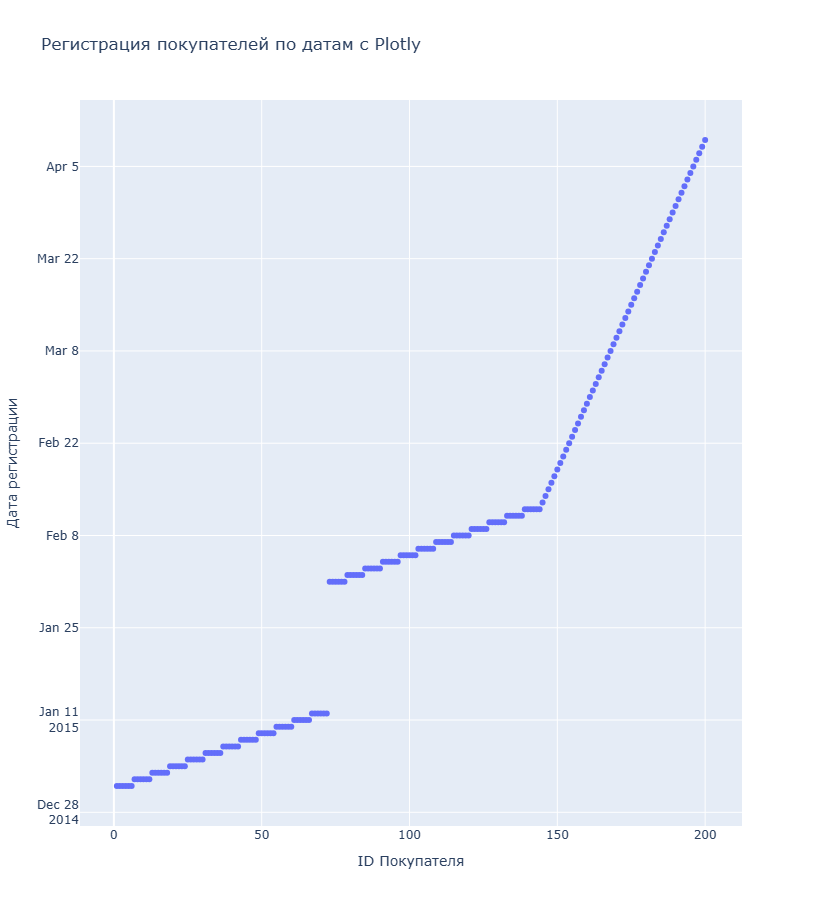
**Seaborn.**

****



**Plotly.**

****

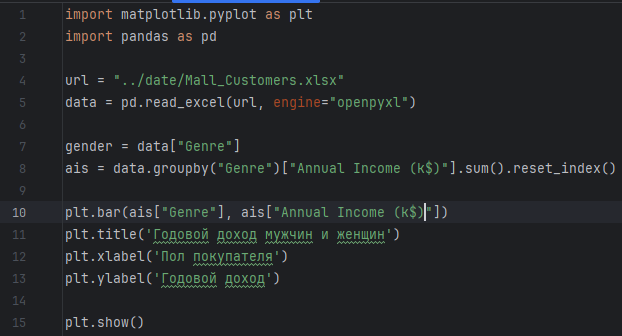
****

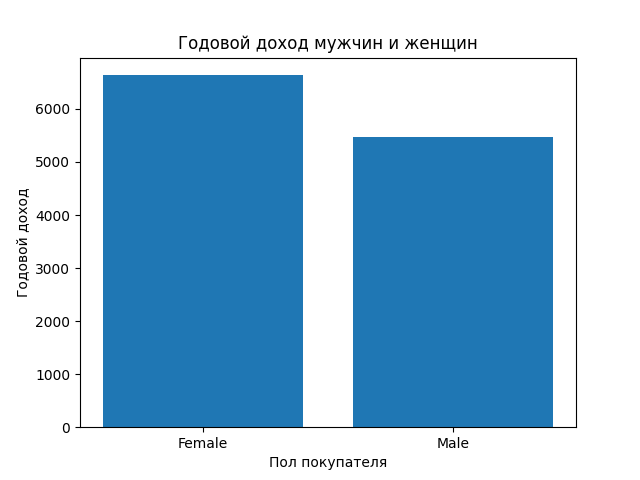
1. **Столбчатая диаграмма.**

Столбчатые диаграммы являются эффективным средством для сравнения количественных показателей или величины различных категорий данных. Они особенно полезны в случаях, когда необходимо провести сравнительный анализ данных, относящихся к различным категориям, например, суммы продаж по месяцам или сравнение успеваемости учащихся. Кроме того, столбчатые диаграммы широко применяются для визуализации абсолютных значений, таких как количество произведённых товаров на разных заводах, а также для отображения долей отдельных частей в общем объёме, например, доли отдела в совокупной прибыли компании.

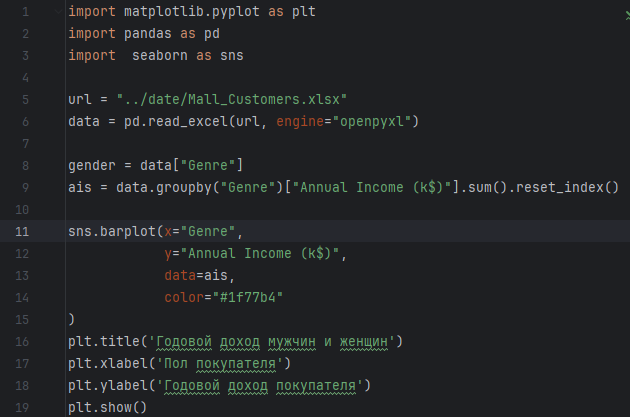
Для проведения исследования была использована база данных, содержащая информацию о поле покупателей и их годовом доходе. Целью анализа являлось определение связи пола с годовым доходом покупателей.

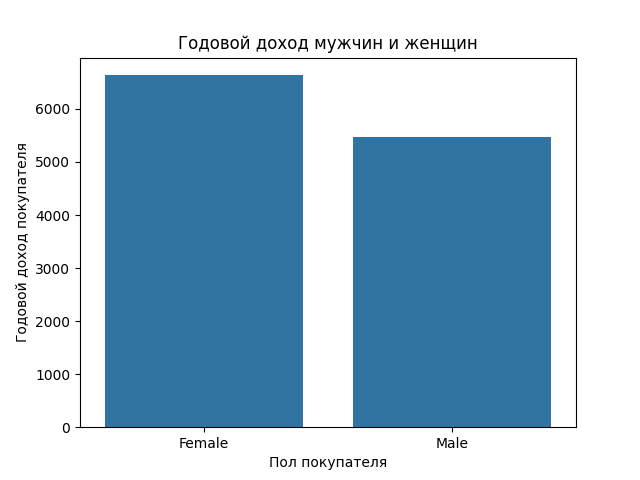
**Matplotlib.**



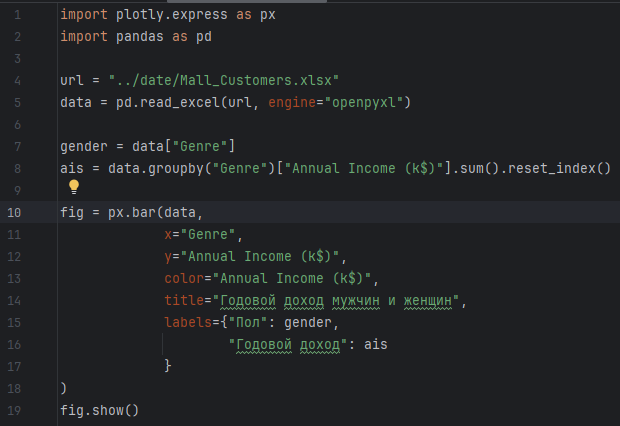


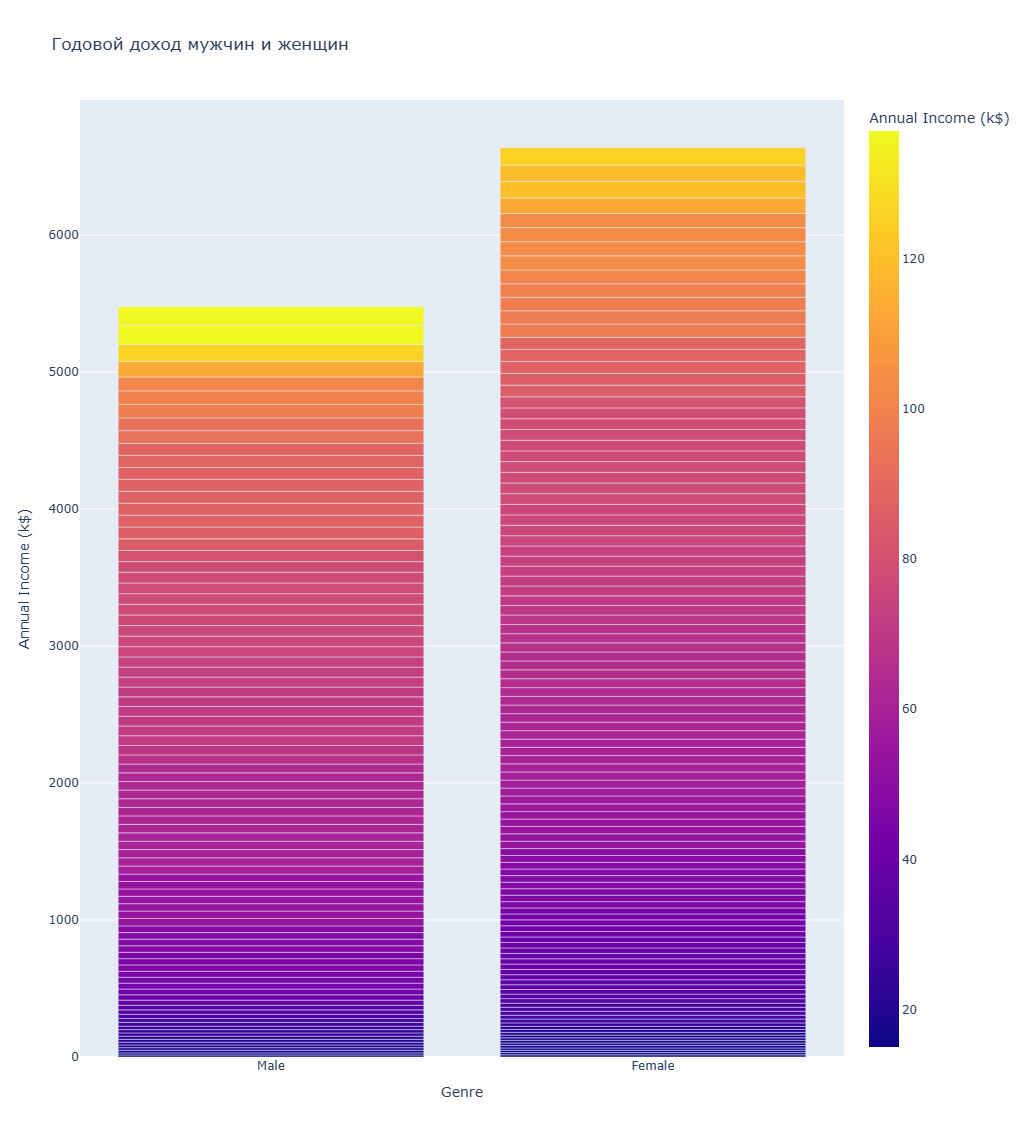
**Seaborn**.





**Plotly**.

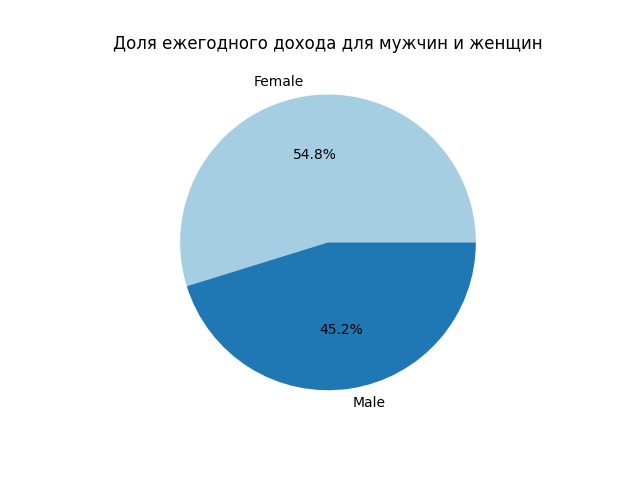
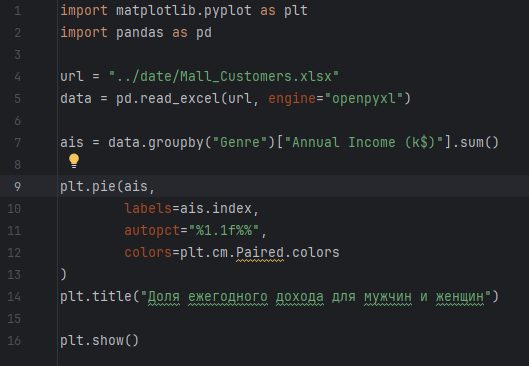




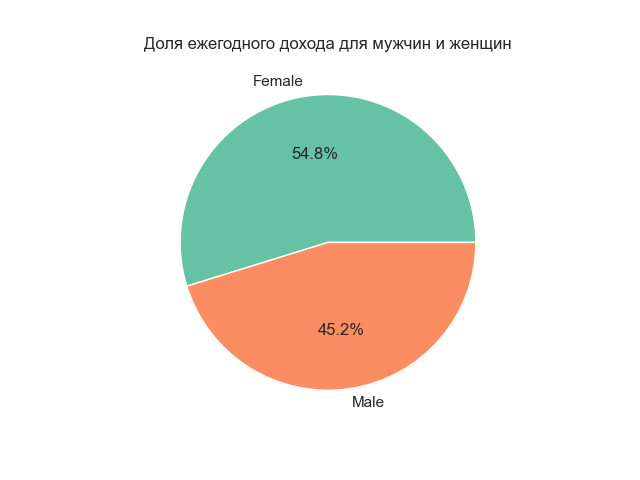
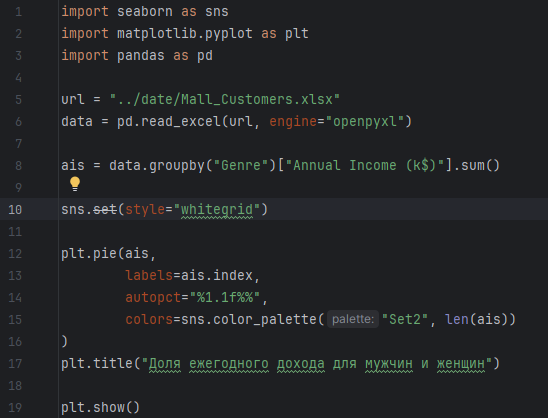
1. **Круговая диаграмма.**

Круговые диаграммы преимущественно применяются для визуализации долей или процентов, составляющих часть от целого. Этот тип диаграмм позволяет наглядно представить соотношение каждой категории (сегмента) друг с другом. Круговые диаграммы хорошо подходят для анализа таких данных, как распределение продаж по регионам, рыночные доли компаний в определённой отрасли, процентное распределение клиентов по возрастным группам или категориям товаров.

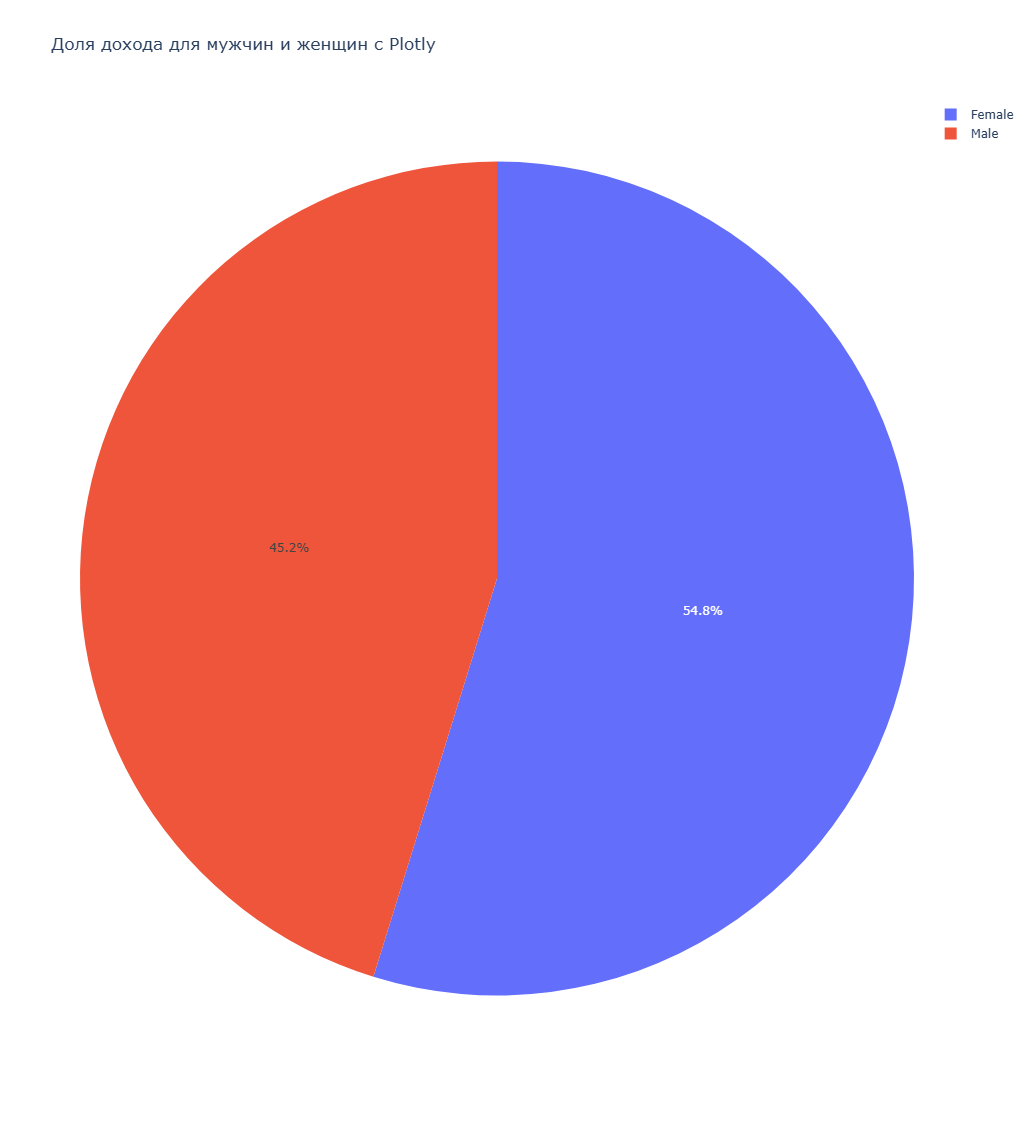
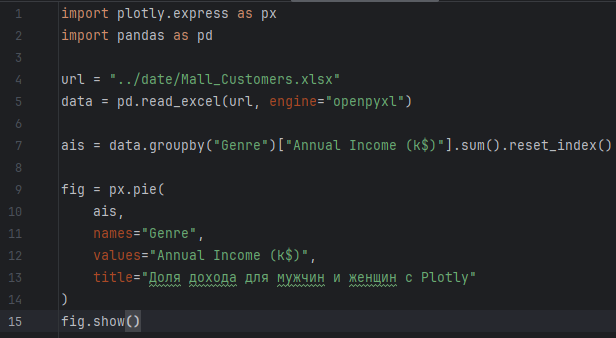
**Matplotlib.**



**Seaborn.**



**Plotly.**

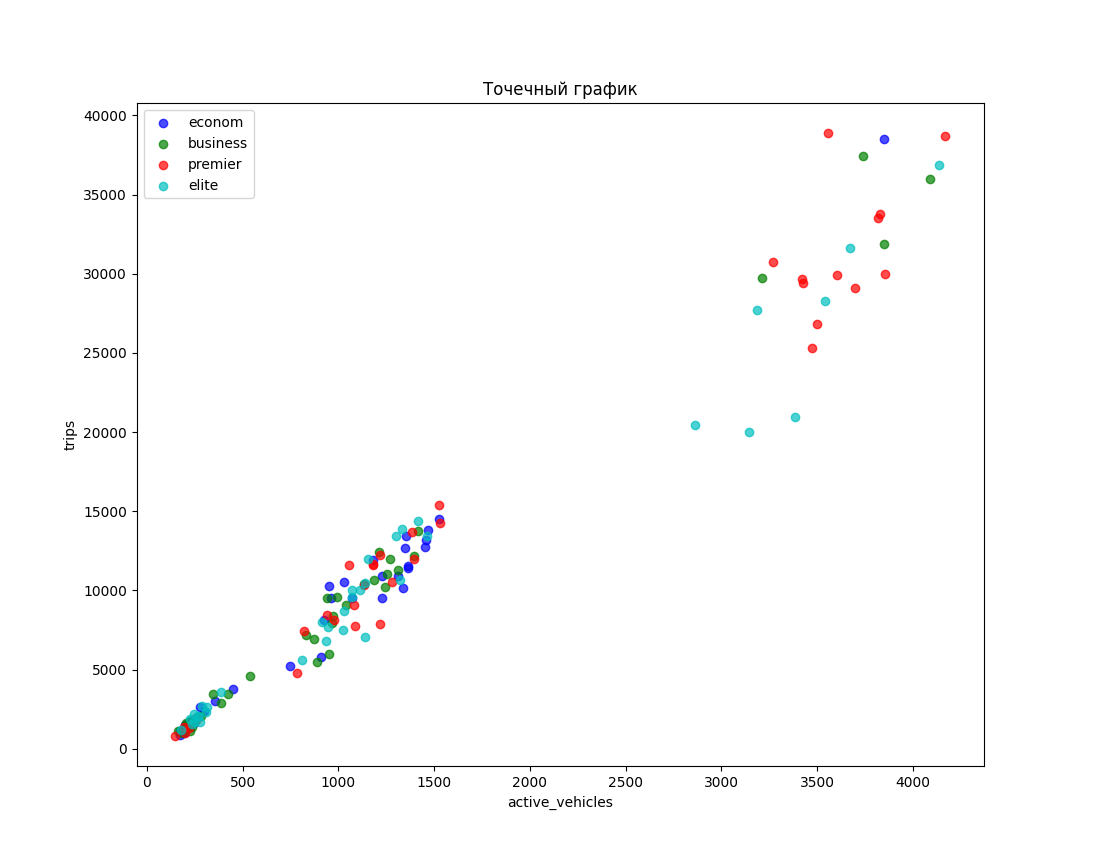
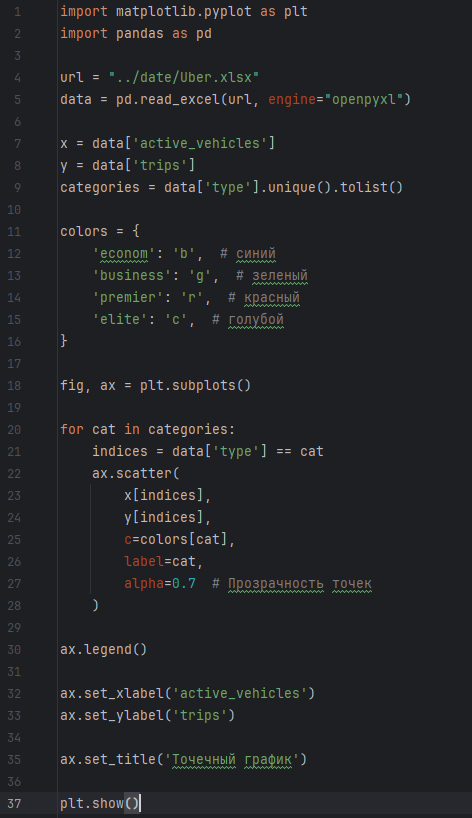


1. **Самые красивые графики (Субъективно).**

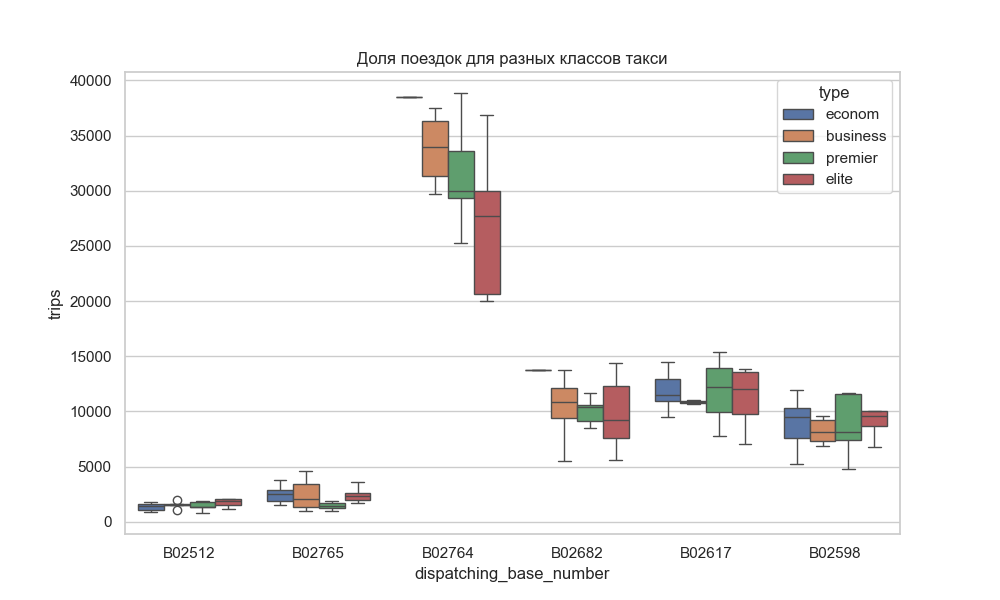
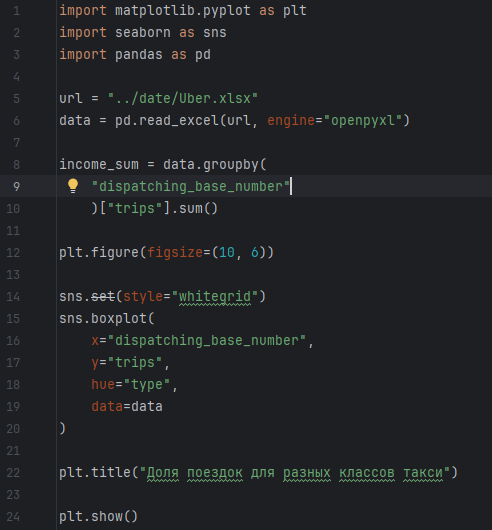
Поскольку основной акцент сделан на библиотеки для визуализации, эстетическая привлекательность создаваемых графиков и диаграмм приобретает особое значение. В ходе исследования возник закономерный вопрос: какими возможностями обладают рассматриваемые библиотеки в создании визуально привлекательных графиков и диаграмм? Каковы их максимальные возможности в этом аспекте?

Предлагается провести заключительный этап анализа, в рамках которого будет оценена способность каждой из библиотек создавать самые эффектные и красивые визуализации.

**Matplotlib. Точечный график.**



**Seaborn. Ящик с усами.**



**Plotly. Матрица рассеяния.**



**9. Рекомендации к выбору графической библиотеки.**

При выборе между **Matplotlib**, **Seaborn** и **Plotly** для визуализации данных, следует учитывать цели проекта, уровень опыта и требуемую производительность.

1. **Цель проекта**:
   * Научные исследования: **Matplotlib** — для точных и настраиваемых графиков.
   * Бизнес-анализ: **Plotly** — для интерактивных и презентационных визуализаций.
   * Экспресс-анализ: **Seaborn** — для быстрого создания красивых графиков.
2. **Уровень опыта**:
   * Новички: **Seaborn** — простое использование готовых стилей.
   * Средний уровень: **Matplotlib** — полная настройка каждого элемента.
   * Продвинутые пользователи: **Plotly** — интерактивность и интеграция с веб-приложениями.
3. **Производительность**:
   * Большие объемы данных: **Matplotlib** — высокая производительность.
   * Реальное время: **Plotly** — поддержка интерактивных элементов.
   * Минимальная задержка: **Seaborn** — быстрое создание простых графиков.
4. **Эстетика и дизайн**:
   * Красивые графики "из коробки": **Seaborn**.
   * Полный контроль над стилем: **Matplotlib**.
   * Современный и интерактивный дизайн: **Plotly**.
5. **Совместимость с другими инструментами**:
   * Работа в Jupyter Notebook: все три библиотеки.
   * Веб-приложения: **Plotly**.
   * Совместимость с Pandas и NumPy: все три библиотеки.

**10. Итог**

Выбор между **Matplotlib**, **Seaborn** и **Plotly** зависит от конкретных задач проекта. **Matplotlib** подходит для точной научной визуализации, **Seaborn** — для быстрого анализа, а **Plotly** — для интерактивных и современных веб-визуализаций.

**Послесловие.**

Завершая наш сравнительный анализ графических библиотек **Matplotlib**, **Seaborn** и **Plotly**, хочется подчеркнуть важность выбора подходящего инструмента для работы с визуализациями. Каждая из рассмотренных библиотек имеет свои сильные стороны и особенности, которые делают её идеальной для определенных типов задач. Важно учитывать не только функциональные возможности, но и удобство использования, производительность и, конечно, эстетическую составляющую.

Стоит отметить, что хотя результаты визуализаторов могут выглядеть похожими, нюансы реализации и удобства работы с ними играют ключевую роль. Например, **Plotly** предоставляет мощные интерактивные инструменты, тогда как **Seaborn** предлагает лаконичные и элегантные стили для быстрого создания красивых графиков. **Matplotlib** же остается классическим выбором благодаря своей гибкости и широкому спектру возможностей настройки.

Однако важно понимать, что ни одна из библиотек не является универсальной панацеей. Выбор той или иной зависит от конкретных потребностей проекта, уровня подготовки команды разработчиков и даже личных предпочтений. Иногда успех визуализации определяется не столько техническими характеристиками библиотеки, сколько умением дизайнера или аналитика интерпретировать и представлять данные.

Поэтому, подводя итоги нашего исследования, можно сказать, что каждая из рассмотренных библиотек находит своё место в арсенале современного специалиста по работе с данными. И, несмотря на субъективность некоторых критериев оценки, всегда стоит экспериментировать и искать наиболее подходящий инструмент для конкретной задачи.