URBAN UNIVERSITY

Агарков Алексей Евгеньевич

Сравнение различных библиотек для визуализации данных:

Matplotlib, Seaborn и Plotly

ДИПЛОМНАЯ РАБОТА ПО PYTHON РАЗРАБОТКЕ

Выполнил:

Студент потока № 71

**Дипломная работа по теме:**

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

Содержание

* Введение
* Цели и задачи работы
* Основные понятия и определения в области визуализации данных
* Обзор библиотек Matplotlib, Seaborn и Plotly
* Сравнение библиотек Matplotlib, Seaborn и Plotly на примере датасет набора ежемесячных данных о продажах.
* Рекомендации по выбору библиотек
* Примеры применения визуализации в различных областях
* Результат сравнения библиотек
* Заключение

**Введение**

Визуализация данных становится все более ценным навыком, востребованным во многих организациях. Она помогает вам разобраться в данных и донести свои выводы до менее подготовленной аудитории.

Визуализация данных – это мощный способ получения информации на основе данных и передачи ее другим пользователям.

Основной задачей анализа данных является понимание взаимосвязей внутри набора данных и их актуальности для проработки вариантов использования. Хорошая визуализация часто быстрее приводит к хорошим идеям, чем часы сбора и перебора данных обычными способами и более понятна нетехнической аудитории. По этим причинам визуализация данных является центральным видом деятельности в любой организации, которая хочет принимать сложные решения на основе данных.

Существует множество ситуаций, когда визуализация данных даст максимальный результат, например, при проведении презентации по продажам, проведении маркетинговых исследований или настройке панели мониторинга ключевых показателей эффективности.

Кроме того, визуализация данных позволяет отслеживать эффективность принятых решений. Сравнение визуализированных данных до и после внедрения изменений дает возможность оценить их влияние и, при необходимости, скорректировать стратегию. Это создает цикл постоянного улучшения, где визуализация данных становится неотъемлемой частью процесса анализа и принятия решений.

В мире анализа данных существует множество инструментов и библиотек, которые позволяют создавать наглядные и информативные графики. Среди них наиболее популярными являются Matplotlib, Seaborn и Plotly. Каждая из этих библиотек имеет свои особенности и преимущества, что делает их подходящими для различных задач визуализации.

**Цели и задачи работы**

Основная цель работы: Провести сравнительный анализ библиотек визуализации данных Matplotlib, Seaborn и Plotly, чтобы определить их сильные и слабые стороны, а также выбрать наиболее подходящую библиотеку для различных задач визуализации.

Задачи работы:

1. Изучить функциональность каждой из библиотек, их возможности и особенности.
2. Создать визуализации с использованием каждой библиотеки на одном и том же наборе данных для сопоставимости.
3. Провести сравнительный анализ созданных визуализаций по критериям эстетики, информативности и удобства использования.
4. Подготовить рекомендации по выбору библиотеки в зависимости от конкретных потребностей пользователя.
5. Привести примеры из различных областей.

**Основные понятия и определения в области визуализации данных:**

* Визуализация данных (Data Visualization): Процесс представления данных в графической форме для облегчения анализа и интерпретации информации.
* Библиотека визуализации (Visualization Library): Набор инструментов и функций, предназначенных для создания графиков и диаграмм. В данном проекте рассматриваются Matplotlib, Seaborn и Plotly.
* Matplotlib: Библиотека для создания статических, а также интерактивных графиков. Она обеспечивает гибкость и контроль над визуализациями.
* Seaborn: Библиотека, основанная на Matplotlib, которая упрощает создание статистических графиков и предлагает более эстетичные визуализации с готовыми стилями.
* Plotly: Библиотека для создания интерактивных графиков, позволяющая пользователям взаимодействовать с визуализациями, что особенно полезно в веб-приложениях.
* Типы графиков (Types of Plots): Различные формы представления данных, такие как линейные графики, столбчатые диаграммы, диаграммы рассеяния и тепловые карты.
* Интерактивность (Interactivity): Способность визуализации реагировать на действия пользователя, такие как наведение курсора и щелчки.
* Эстетика графиков (Plot Aesthetics): Оформление графиков, включая шрифты, размеры, цветовые схемы и оформление осей.
* Метрики визуализации (Visualization Metrics): Параметры, используемые для оценки качества визуализаций, такие как читаемость и информативность.
* График (Plot): Визуальное представление данных, которое может принимать различные формы, такие как линии, точки, столбцы и т.д.
* Оси (Axes): Линии, которые определяют границы графика (обычно ось X и ось Y), на которых отображаются данные.
* Подписи (Labels): Текстовые метки, которые описывают оси, заголовок графика и отдельные элементы графика.
* Легенда (Legend): Область графика, которая объясняет, что означают различные элементы графика (например, цвета или маркеры для разных категорий данных).
* Стили (Styles): Наборы параметров, которые определяют внешний вид графиков, включая цвет, шрифт и толщину линий.
* Данные (Data): Информация, которую мы хотим визуализировать, представленная в различных форматах, таких как списки, массивы или DataFrame (в библиотеке pandas).
* Подграфики (Subplots): Возможность размещения нескольких графиков в одном окне или области для сравнения различных наборов данных.
* Форматирование (Formatting): Настройка внешнего вида графиков, включая цвет, шрифт и стиль линий, что позволяет улучшить визуальную привлекательность.
* Экспорт (Export): Процесс сохранения графиков в различных форматах (например, PNG, JPEG, PDF) для использования вне среды программирования.
* Анимация (Animation): Процесс создания движущихся графиков, который помогает визуализировать изменения данных во времени.
* Тепловая карта (Heat map): Визуализация данных, где значения представлены цветами, что позволяет быстро идентифицировать паттерны и аномалии.
* Сетчатая система (Grid System): Система координат, которая помогает организовать расположение графиков и элементов на графике.
* Кастомизация (Customization): Процесс изменения стандартных параметров графиков для достижения желаемого внешнего вида и функциональности.
* Точки данных (Data Points): Индивидуальные элементы данных, которые отображаются на графике.
* Трендовая линия (Trend Line): Линия, которая показывает общее направление данных на графике, обычно используется для выявления тенденций.
* Диапазон (Range): Разница между максимальным и минимальным значением данных на оси графика.

**Обзор библиотек Matplotlib, Seaborn и Plotly**

Мы можем охарактеризовать библиотеки визуализации данных по следующим факторам:

* **Интерактивность**: предполагает ли библиотека интерактивные элементы
* **Синтаксис**: какой уровень контроля предлагает библиотека и следует ли она определенной парадигме
* **Основные преимущества и варианты использования**: к какой ситуации библиотека является лучшим выбором.

В таблице приведены лучшие библиотеки визуализации **Python** в соответствии с этими факторами:

|  |  |  |  |
| --- | --- | --- | --- |
| **Библиотека** | **Интерактивность** | **Синтаксис** | **Основные преимущества и варианты использования** |
| Matplotlib | Ограничено | Низкоуровневый | Индивидуальные диаграммы |
| Seaborn | Ограничено | Высокоуровневый | Быстрые и презентабельные отчеты |
| Plotly | Да | Высокоуровневый и низкоуровневый | Коммерческие приложения и информационные панели |

Теперь разберем каждую библиотеку подробнее.

**Matplotlib**

[**Matplotlib**](https://matplotlib.org/) – наиболее широко используемая библиотека визуализации. Она появилась в 2003 году как бесплатная замена **MATLAB**, пакета прикладных программ для решения задач технических вычислений.

Из-за её раннего запуска и популярности вокруг **Matplotlib** существует огромное комьюнити. Вы можете с легкостью найти обучающие материалы и форумы с обсуждениями, а также множество дополнительных инструментов, которые расширяют использование библиотеки (например, в географических данных или 3D визуализации). Кроме того многие библиотеки **Python** (например, **pandas**) используют её в своих функциях визуализации.

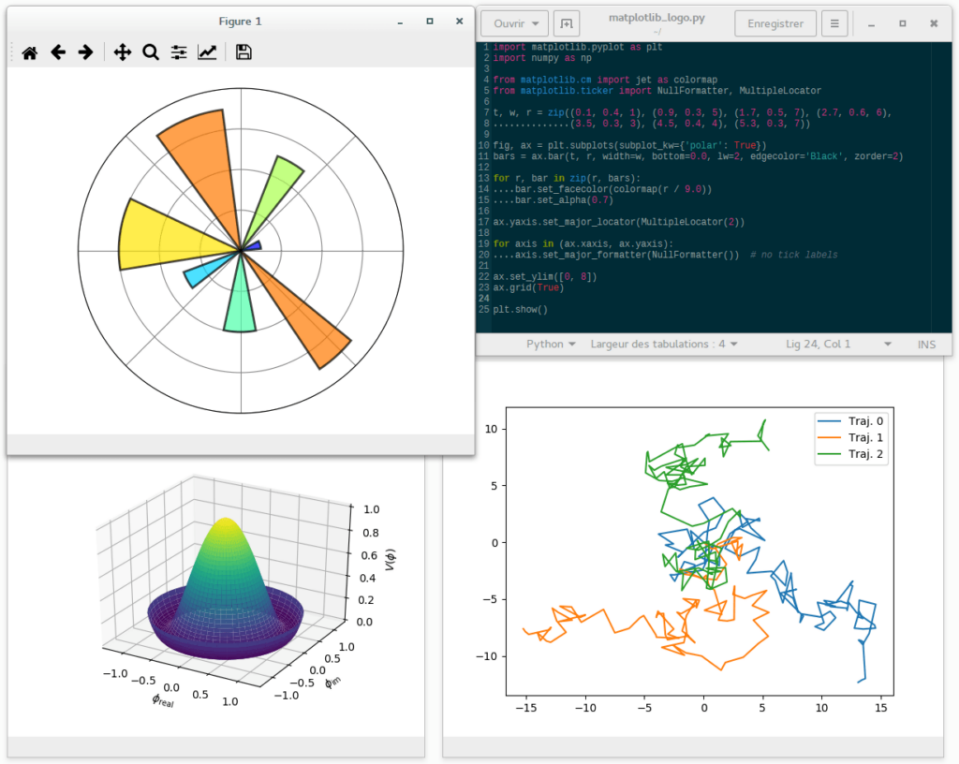
**Matplotlib** обеспечивает детальное управление графиками, что делает его универсальным решением с широкой номенклатурой графиков и опций конфигурации. Однако его многочисленные возможности настройки усложняют его использование и могут отпугнуть. Стандартная тема **Matplotlib** не соответствует современным рекомендациям по визуализации. Также для полноценного использования основных функций библиотеки могут понадобиться дополнительные библиотеки (например, по обработке временных данных).

**Достоинства:**

1. Позволяет создавать практически любые виды графиков и диаграмм благодаря низкоуровневому интерфейсу.
2. Поддерживает множество типов графиков, включая линейные графики, гистограммы, 3D графики, контурные графики, круговые диаграммы и др.
3. Обширные возможности настройки всех элементов графиков, таких как оси, легенды, цвета, линии, маркеры и шрифты.
4. Возможность создания анимаций и динамических визуализаций.
5. Поддержка сохранения графиков в различных форматах (PNG, PDF, SVG и др.).

**Недостатки:**

* Может быть сложным для новичков из-за низкоуровневого интерфейса и большого количества настроек.
* По умолчанию графики могут выглядеть не очень привлекательно, что требует дополнительных усилий для улучшения дизайна.



**Seaborn**

[**Seaborn**](https://seaborn.pydata.org/examples/index.html) – это библиотека визуализации, делающая графики **Matplotlib** более удобными. Она уходит от сложности **Matplotlib** и предлагает интуитивно понятный синтаксис и презентабельные результаты без каких-либо дополнительных действий.

Библиотека **seaborn** поддерживает создание статистических графиков. Она хорошо взаимодействует с датафреймами **pandas**, обеспечивает отображение данных на визуализации и может преобразовывать данные как часть графика.

Также у него есть выразительная стандартная тема с различными цветовыми палитрами, определенными с учетом лучших мировых практик.

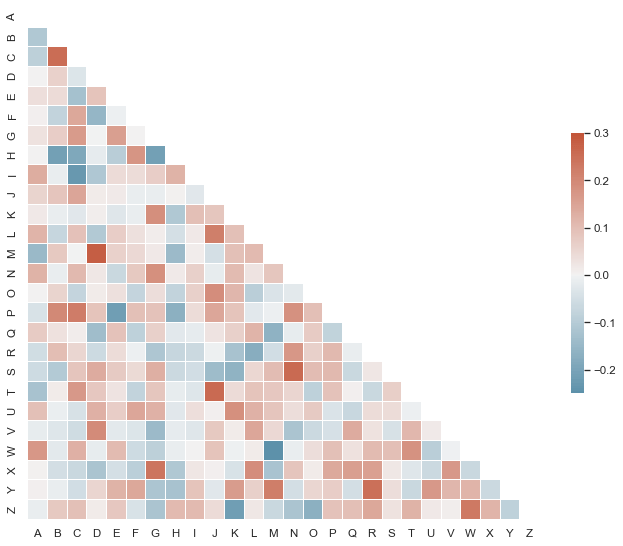
Поскольку **seaborn** является оболочкой для **Matplotlib** вы можете настроить свои графики, обратившись к базовым объектам **Matplotlib**.

**Достоинства**:

1. Делает процесс создания графиков проще и быстрее.
2. Хорошо подходит для создания статистических графиков, таких как коробчатые диаграммы и тепловые карты.
3. Имеет предустановленные стили, которые делают графики более привлекательными без лишних усилий.
4. Легко работает с данными из Pandas, что упрощает анализ.
5. Комбинируют информацию о распределении данных и их плотности, что позволяет лучше понять распределение.

Недостатки:

* Ограничен в настройках по сравнению с Matplotlib, что может быть недостатком для сложных визуализаций. Поскольку Seaborn построен на Matplotlib, пользователи могут столкнуться с ограничениями, если захотят использовать функционал Matplotlib, который не поддерживается в Seaborn.



**Plotly**

[**Plotly**](https://plotly.com/python/) – это библиотека визуализации данных с открытым исходным кодом и часть экосистемы, разработанной Plotly, Inc. Компания также разрабатывает **Dash**, библиотеку информационных панелей на **Python**, и предлагает услуги приложений визуализации данных для корпоративных клиентов. По этой причине **Plotly** – отличный инструмент для создания интерактивных визуализаций и информационных панелей, ориентированных на бизнес.

**Plotly** предлагает высокоуровневый интерфейс для быстрой разработки и низкоуровневый для большего контроля. Он также отображает графики из простых словарей и имеет широкий спектр предопределенных типов графиков.

**Достоинства**:

1. Позволяет создавать интерактивные графики, которые можно масштабировать, перемещать и наводить курсор для получения дополнительной информации.
2. Поддерживает создание графиков, обновляющихся в реальном времени, что полезно для мониторинга данных.
3. Обширный набор графиков, включая 3D графики, карты, графики с областями, пузырьковые графики и специализированные визуализации.
4. Возможность сохранять и делиться графиками через облачные сервисы, такие как Plotly Chart Studio.

**Недостатки:**

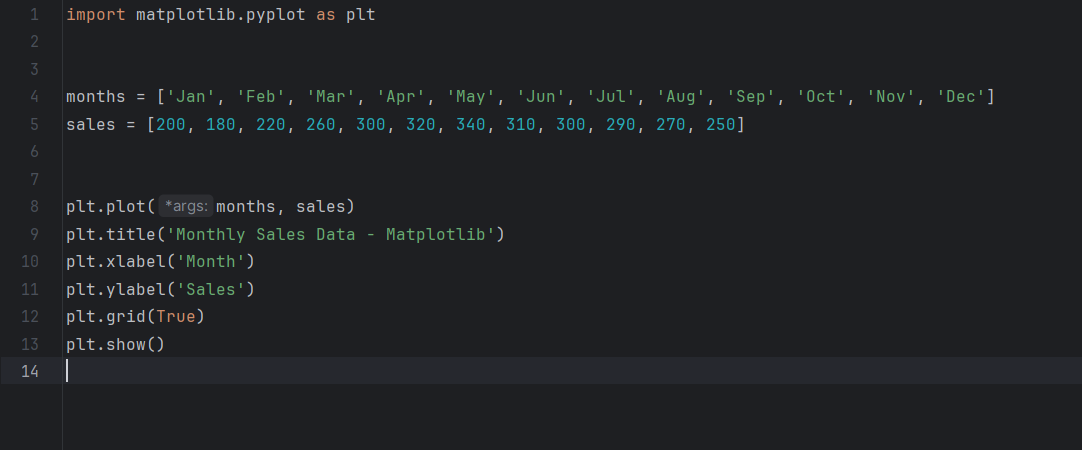
* Интерактивные графики могут требовать больше ресурсов, что может быть проблемой для больших наборов данных.
* Для достижения более сложных визуализаций может потребоваться больше времени на изучение и настройку.
* Некоторые функции могут требовать подписки на платные версии Plotly, что может быть ограничивающим фактором для некоторых пользователей.



**Сравнение библиотек Matplotlib, Seaborn и Plotly на примере набора ежемесячных данных о продажах.**

Создадим линейные диаграммы для отображения набора ежемесячных данных о продажах. С помощью этого постараемся выявить уникальные особенности и преимущества каждой библиотеки: от простоты Matplotlib и улучшенной эстетики Seaborn до интерактивных возможностей Plotly.

**Линейный график с использованием Matplotlib**



Пояснение к коду

1. *import matplotlib.pyplot as plt:* эта строка импортирует модуль matplotlib.pyplot и переименовывает его в plt для удобства использования. Matplotlib — популярная библиотека Python для создания диаграмм и графиков.

2. *plt.plot(months, sales):* эта функция создаёт линейную диаграмму. Она принимает на вход два списка: months и sales. months — это список названий месяцев (например, «январь», «февраль» и т. д.), а sales — это список соответствующих значений продаж за каждый месяц. Линейная диаграмма строится с месяцами по оси X и продажами по оси Y.

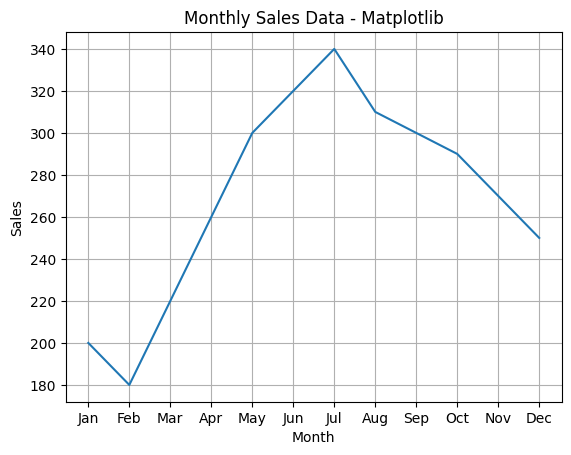
3. *plt.title(‘Monthly Sales Data — Matplotlib’):* добавляет заголовок к диаграмме. В данном случае заголовок — 'Ежемесячные данные о продажах — Matplotlib'.

4. *plt.xlabel('Month')* и *plt.ylabel('Sales'):* эти строки обозначают ось X как 'Месяц', а ось Y — как 'Продажи'. Это помогает понять, что представляет собой каждая ось на графике.

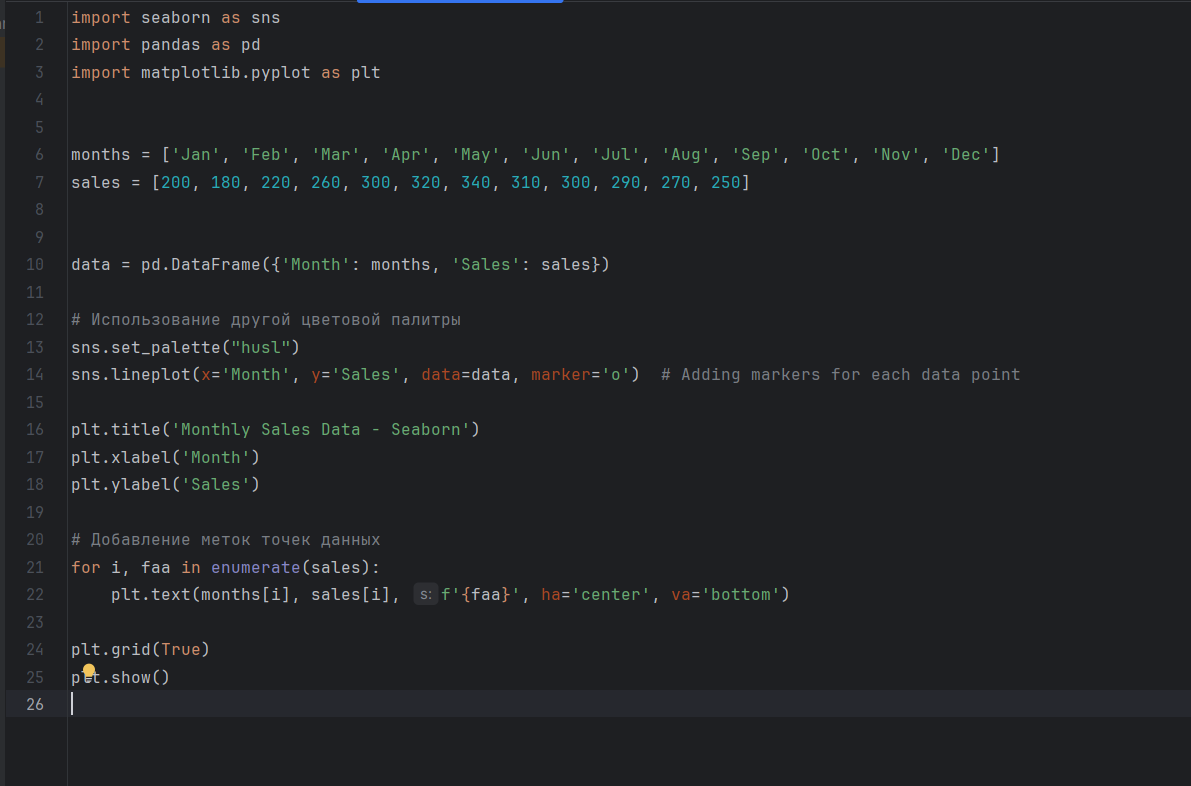
5. *plt.grid(True):* Эта строка добавляет на график сетку, облегчающую чтение и понимание значений. Линии сетки проходят как по вертикали (по месяцам), так и по горизонтали (по показателям продаж).

6. plt.show(): наконец, эта строка отображает график. Без этой команды график создавался бы в фоновом режиме, но не отображался бы на экране.

Результат выполнения кода



**Линейный график с использованием Seaborn**



Пояснение к коду

1. Импорт библиотек

*import seaborn as sns:* подключает библиотеку Seaborn, которая используется для создания привлекательной и информативной статистической графики.

*import pandas as pd:* импортирует библиотеку Pandas, мощный инструмент для обработки и анализа данных.

2. Подготовка данных

*data = pd.DataFrame({'Month': months, 'Sales': sales}):* Здесь с помощью Pandas создаётся фрейм данных (своего рода таблица). В нём есть два столбца: 'Месяц' и 'Продажи', заполненные значениями из списков месяцев и продаж.

3. Настройка Seaborn

*sns.set\_palette("husl"):* изменяет цветовую палитру, используемую Seaborn, на "husl", которая представляет собой набор приятных глазу цветов. Это повлияет на внешний вид линейчатой диаграммы.

4. Создание линейной диаграммы

*sns.lineplot(x=' Month', y=' Sales', data= data, marker='o'):* эта функция создает линейную диаграмму с помощью Seaborn. Она отображает 'Продажи' в зависимости от 'Месяца', а 'маркер='o' добавляет маленькие кружочки в каждой точке данных на линии.

5. Добавление заголовков и меток

Функции *plt.title, plt.xlabel* и *plt.ylabel* используются для добавления заголовка к диаграмме и обозначения оси X как «Месяц», а оси Y как «Продажи».

6. Добавление меток точек данных

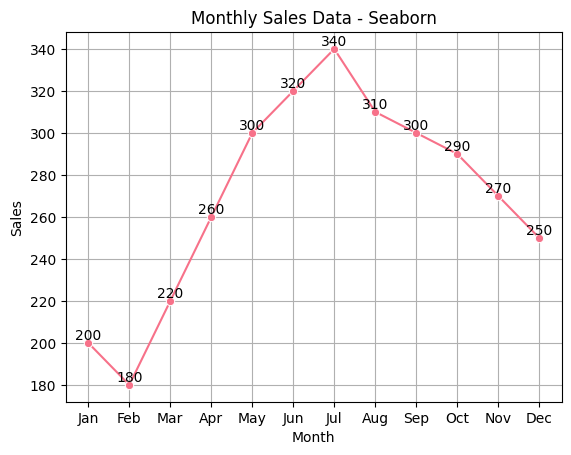
Цикл *for* с *plt.text* используется для размещения метки для каждой точки данных на графике *enumerate(sales),* перебирает список продаж, и для каждого значения *(faa)* добавляется текстовая метка в соответствующем месяце. Части *ha='center'*, *va='bottom'* отвечают за выравнивание, благодаря чему каждая метка располагается по центру по горизонтали и в нижней части по вертикали.

7. Последние штрихи

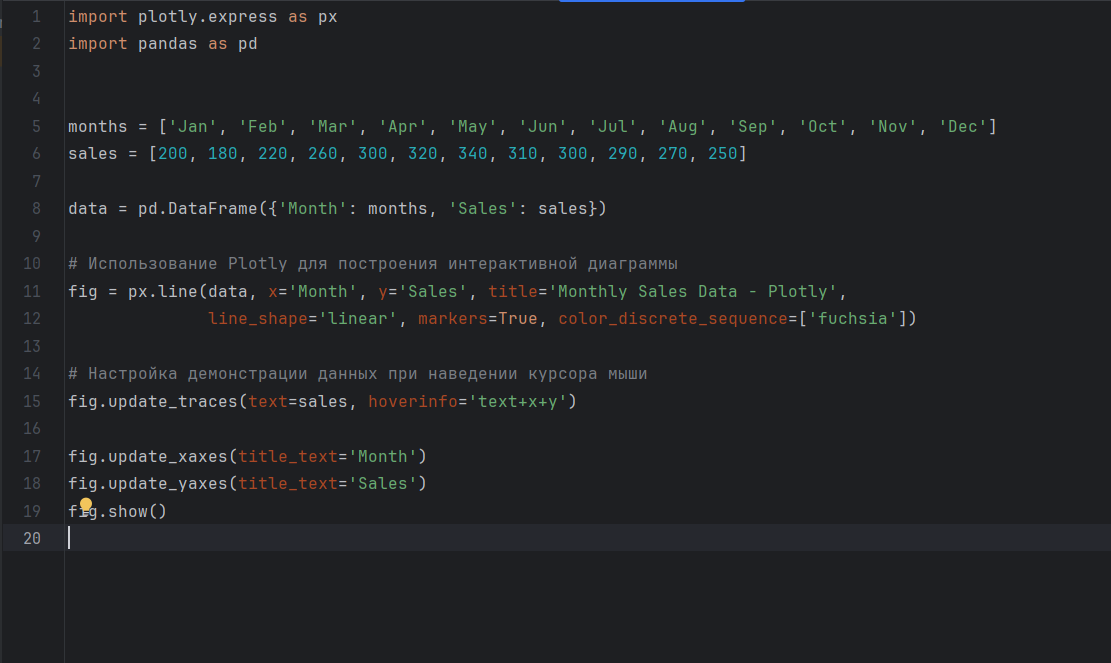
*plt.grid(True):* добавляет на график сетку, облегчающую его чтение.

*plt.show():* На экране отображается итоговый график.

Результат выполнения кода



Линейный график с использованием Plotly



Пояснение к коду

1. Импорт Plotly Express

*import plotly.express as px:* Эта строка импортирует библиотеку Plotly Express, которая отлично подходит для создания интерактивных графиков и диаграмм.

2. Настройка данных

*data = pd.DataFrame({'Month': months, 'Sales': sales}):* Здесь с помощью Pandas (что подразумевается под pd, которое обычно используется для Pandas) создаётся фрейм данных. Этот фрейм данных похож на таблицу с двумя столбцами: «Месяц» и «Продажи», заполненными значениями из списков месяцев и продаж.

3. Создание интерактивной линейной диаграммы

*fig = px.line(...):* В этой части создается линейная диаграмма. На диаграмме «Продажи» представлены в зависимости от «Месяца».

*title='Monthly Sales Data — Plotly':* добавляет заголовок к диаграмме.

*line\_shape='linear':* гарантирует, что линия на графике будет прямой.

*markers=True:* добавляет маркеры (маленькие точки) к каждой точке данных на линии для лучшей видимости.

*color\_discrete\_sequence=['fuchsia']:* задает цвет линии как фуксию — яркий пурпурно-красный цвет.

4. Настройка информации о Наведении Курсора мыши

*fig.update\_traces(text=sales, hoverinfo='text+x+y'):* строка кода описывает, что происходит, когда вы наводите курсор мыши на диаграмму. Она показывает данные о продажах (а также месяц и значения продаж), когда вы наводите курсор на любую точку на линии.

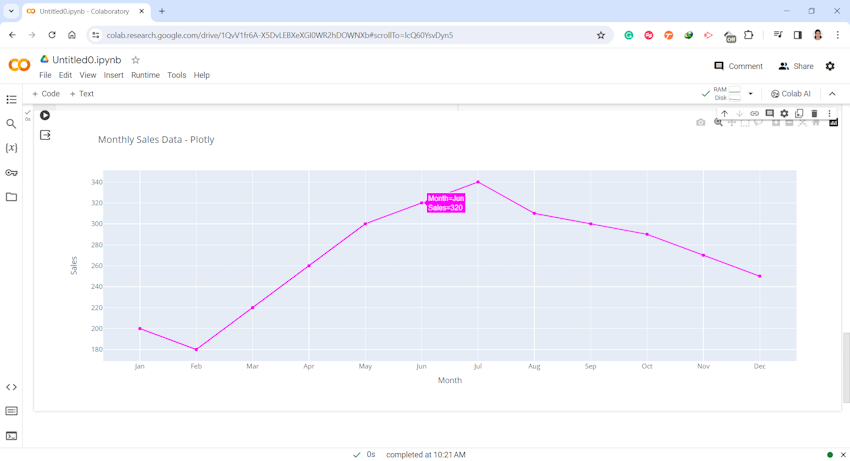
5. Обновление осей диаграммы

*fig.update\_xaxes(title\_text='Month')* и *fig.update\_yaxes(title\_text='Sales'):* эти строки обозначают ось X как 'Месяц', а ось Y — как 'Продажи'. Это сделано для того, чтобы диаграмму было легче читать.

6. Отображение диаграммы

*fig.show():* Эта последняя строка фактически отображает диаграмму на вашем экране.

Результат выполнения кода



В Matplotlib и Seaborn мы добавляем метки для каждой точки данных с помощью цикла. Это позволяет отображать количество продаж за каждый месяц. Цветовая палитра Seaborn изменена на более яркую. Мы также добавляем маркеры к каждой точке данных для наглядности. В Plotly мы используем интерактивную функцию наведения курсора для отображения данных. Цвет линии установлен на 'фуксию', а для лучшей видимости добавлены маркеры. Эти настройки улучшают читаемость и эстетическую привлекательность диаграмм, позволяя с первого взгляда получить более чёткое представление о данных.

**Рекомендации по выбору библиотек**

**Matplotlib:**

• Если нужны высококачественные статические графики для научных публикаций, отчетов или презентаций.

• Настройка каждого аспекта графики, включая цвета, шрифты, размеры осей, линии и маркеры.

• При Обработке больших наборов данных и визуализации без значительных затрат на производительность.

• Если необходимо сделать сложные графики, например, многоуровневые графики, графики с несколькими осями или графики с дополнительными пояснениями.

• Если вы работаете с библиотеками для анализа данных, такими как NumPy и Pandas.

• Поддерживает создание анимаций, что позволяет визуализировать изменения данных во времени и делать графики более динамичными и информативными.

**Seaborn:**

• Удобные функции для создания статистических графиков, таких как распределения, корреляционные матрицы и категориальные графики. Он автоматически обрабатывает статистические параметры и предоставляет красивые стили по умолчанию.

• Хорошо интегрируется с Pandas DataFrame, что упрощает визуализацию данных. Pandas позволяет эффективно обрабатывать и манипулировать большими наборами данных.

• Предлагает множество встроенных тем и палитр, которые упрощают настройку графиков и делают их более привлекательными без необходимости вручную настраивать каждый элемент.

• Предоставляет мощные инструменты для сравнения групп, такие как facetgrid, который позволяет разбивать данные на подгруппы и визуализировать их в одном графике.

• Имеет простой и интуитивно понятный синтаксис, что делает его доступным для пользователей, которые только начинают работать с визуализацией данных.

**Plotly:**

• Предоставляет мощные инструменты для создания интерактивных графиков, позволяя пользователям взаимодействовать с визуализациями, масштабировать, наводить курсор для получения дополнительной информации и переключаться между различными представлениями данных.

• Эффективно обрабатывает большие наборы данных и позволяет визуализировать их без значительного снижения производительности. Это особенно полезно для анализа больших объемов данных, таких как временные ряды или многомерные наборы данных.

• Поддерживает множество типов графиков, включая 3D-графики, графики с несколькими осями и комбинированные графики. Это позволяет создавать сложные визуализации, которые могут быть полезны для глубокого анализа данных.

• Хорошо интегрируется с веб-приложениями и позволяет легко встраивать графики в HTML-страницы, что делает его отличным выбором для веб-визуализации и создания интерактивных дашбордов.

• Предоставляет удобные инструменты для создания графиков, которые позволяют легко сравнивать разные наборы данных, например, через наложение графиков или использование под графиков.

• Предлагает мощные инструменты для создания карт и визуализации геоданных, что делает его отличным выбором для проектов, связанных с географическим анализом.

**Примеры применения визуализации в различных областях**

**Образование:**

Учебные заведения активно используют визуализацию данных для анализа успеваемости студентов и оценки эффективности образовательных программ. Например, с помощью библиотек, таких как Matplotlib и Seaborn, можно создавать графики, показывающие распределение оценок по предметам. Это помогает выявить слабые места в учебных планах и определить, какие темы требуют дополнительного внимания. Кроме того, дашборды могут включать визуализации, отображающие динамику успеваемости студентов по семестрам, что позволяет преподавателям и администраторам принимать обоснованные решения о необходимости изменения подходов к обучению.

**Финансовые услуги:**

В финансовом секторе визуализация данных играет ключевую роль в анализе рыночных тенденций и оценке инвестиционных портфелей. Финансовые аналитики используют интерактивные графики, созданные с помощью Plotly, для отображения изменений цен акций, объемов торгов и других финансовых показателей в реальном времени. Например, можно создать многослойный график, который показывает, как различные факторы, такие как экономические отчеты или изменения в политике, влияют на цены акций. Это помогает инвесторам принимать более информированные решения и своевременно реагировать на изменения на рынке.

**Торговля и ритейл:**

В ритейле визуализация данных используется для анализа покупательского поведения и оптимизации товарных запасов. Ритейлеры применяют тепловые карты и диаграммы для отображения данных о продажах по регионам и времени. Например, визуализация может показывать, какие товары наиболее популярны в разных магазинах или в определенные сезоны, что позволяет компаниям адаптировать свои стратегии маркетинга и управления запасами. Также можно использовать графики для анализа эффективности рекламных акций, сравнивая продажи до и после кампании.

**Спорт и фитнес:**

В спортивной аналитике визуализация данных помогает тренерам и спортсменам анализировать производительность и прогресс тренировок. С помощью библиотек, таких как Matplotlib и Plotly, можно создавать графики, отображающие результаты тестов на выносливость, силу и другие показатели. Например, тренеры могут визуализировать изменения в результатах спортсменов на протяжении сезона, что позволяет выявить тренды и корректировать тренировочные программы. Также, анализируя данные о физических нагрузках и восстановлении, можно оптимизировать режим тренировок для достижения максимальных результатов.

**Государственное управление:**

Государственные учреждения используют визуализацию данных для анализа социальных и экономических показателей, таких как уровень безработицы, доступ к образованию и здравоохранению. С помощью интерактивных карт и графиков можно визуализировать распределение ресурсов и демографические изменения в разных регионах. Например, карты могут показывать уровень безработицы по округам, что помогает в планировании программ поддержки и улучшения экономической ситуации. Визуализация данных также может быть использована для мониторинга эффективности государственных программ и оценки их воздействия на общество.

**Результат сравнения библиотек**

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристика** | **Matplotlib** | **Seaborn** | **Plotly** |
| Тип графиков | Статические и интерактивные | Статистические | Интерактивные, 3D, карты |
| Легкость использования | Средняя (может быть сложной для новичков) | Высокая (интуитивно понятный интерфейс) | Высокая (простота для создания интерактивных графиков) |
| Эстетика | Низкая по умолчанию, требует настройки | Высокая (предустановленные стили и палитры) | Высокая (интерактивные и визуально привлекательные) |
| Кастомизация | Высокая (гибкость в настройках) | Средняя (ограничена по сравнению с Matplotlib) | Средняя (достаточно для большинства задач) |
| Интерактивность | Ограниченная (в основном статические графики) | Ограниченная (статические графики) | Высокая (возможность взаимодействия с графиками) |
| Поддержка 3D | Да (но требует дополнительных усилий) | Нет | Да |
| Работа с данными | Хорошая интеграция с NumPy и Pandas | Отличная интеграция с Pandas | Хорошая интеграция с Pandas и другими источниками |
| Анимация | Да | Нет | Да |
| Экспорт | Поддержка различных форматов (PNG, PDF, SVG) | Поддержка форматов через Matplotlib | Поддержка экспорта в HTML и облачные сервисы |
| Области применения | Научные исследования, инженерия, отчеты | Научные исследования, анализ данных | Бизнес-аналитика, веб-разработка, интерактивные отчеты |
| Преимущества | Гибкость, мощные инструменты для визуализации | Упрощение создания красивых статистических графиков | Интерактивность, возможность работы с большими данными |
| Недостатки | Сложность для новичков, требует настройки | Ограниченные возможности кастомизации | Требует больше ресурсов, некоторые функции платные |

**Заключение**

Визуализация данных помогает человеческому мозгу воспринимать и извлекать информацию. Основная цель визуализации данных  —  облегчить поиск закономерностей, тенденций и выбросов в огромных массивах данных.

В годы развития больших данных визуализация данных приобретает все большее значение. Она помогает заинтересованным сторонам оперативно визуализировать бизнес-показатели и принимать решения, основанные на данных. Вот почему выбор подходящего инструмента визуализации и его интеграция в сайт становятся все более важными задачами.