Дипломная работа по теме:

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

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

Автор: Морозова Юлия Андреевна

Оглавление:

1. Введение 1
2. Обоснование выбора темы 1
3. Определение цели и задач исследования 1
4. Основные понятия и определения 2-3
5. Обзор библиотек: Matplotlib, Seaborn, Plotly 3-4
6. Методы и подходы к визуализации данных 4-5
7. Методы разработки визуализаций 5-6
8. Преимущества и недостатки библиотек 6-9
9. Анализ и интерпретация результатов 10-13
10. Заключение 13
11. **Введение:**

Визуализация данных - это процесс представления информации в графической или визуальной форме для облегчения понимания, анализа и интерпретации данных. В современном мире, где объём данных постоянно растёт, эффективная визуализация становится критически важной для извлечения полезной информации и принятия обоснованных решений.

**2.Обоснование выбора темы:**

В основном из-за потребностей рынка. Объём данных стремительно увеличивается, умение эффективно визуализировать информацию становится важным навыком для специалистов в различных областях, включая науку, бизнес, маркетинг и многие другие. Визуализация данных позволяет не только представлять информацию в удобной и понятной форме, но и выявлять закономерности, тренды и аномалии, которые могут быть неочевидны при анализе «сырых данных». В этой связи выбор темы, связанной с визуализацией данных с использованием популярных библиотек Python, таких как Matplotlib, Seaborn, Plotly, имеет несколько обоснований:

1. Растущая значимость данных:

Организации сталкиваются с необходимостью извлечения полезной информации из огромных объёмов данных, то есть визуализация данных становится ключевым инструментов для анализа и интерпретации информации.

1. Разнообразие инструментов:

Каждая из библиотек предлагает уникальные возможности, но Matplotlib, Seaborn, Plotly являются тремя наиболее популярными библиотеками в экосистеме Python. Понимание их особенностей и уместности использования в различных сценариях поможет пользователям выбрать наиболее подходящий инструмент для своих задач.

1. Разные подходы к визуализации:

Matplotlib предоставляет низкоуровневый интерфейс с высокой гибкостью, Seaborn упрощает создание статистических графиков с улучшенным стилем, а Plotly позволяет создавать интерактивные визуализации. Это разнообразие позволяет выбрать подходящий инструмент в зависимости от требований проекта и целевой аудитории.

**3.Определение цели и задач исследования:**

Цель исследования: Создать набор визуализаций с использованием Matplotlib, Seaborn, Plotly, сравнить их функциональность и удобство. Исследование направлено на выявление особенностей, преимуществ и недостатков каждой из библиотек, а также на определение их уместности для различных типов визуализаций и задач. В конечном итоге, цель состоит в том, чтобы помочь пользователям выбрать наиболее подходящий инструмент для визуализации данных в зависимости от их конкретных потребностей и контекста применения.

Задачи исследования:

Для достижения поставленной цели необходимо решить следующие задачи:

****4.Основные понятия и определения****

Существует множество библиотек для визуализации данных в Python, каждая из которых предлагает уникальные особенности и инструменты. В этом контексте обозначу три наиболее популярных библиотеки - Matplotlib, Seaborn, Plotly. Каждая из них имеет свои преимущества и недостатки, что делает их подходящими для различных задач и типов визуализации.

Pandas

• DataFrame: Двумерная структура данных, аналогичная таблице в реляционных базах данных, которая позволяет хранить данные в виде строк и столбцов. DataFrame является основным объектом в Pandas.

• Series: Одномерный массив, который может хранить данные любого типа. Series представляет собой один столбец DataFrame.

• Индексация: Метод доступа к данным в DataFrame или Series с использованием меток или числовых индексов.

• Группировка (groupby): Метод, который позволяет агрегировать данные по определенному критерию, например, по категориям.

• Объединение (merge): Процесс объединения нескольких DataFrame на основе общих столбцов.

▎Matplotlib

• Фигура (Figure): Основной контейнер для всех графиков и визуализаций. В Matplotlib создается одна фигура, которая может содержать несколько осей.

• Оси (Axes): Область в фигуре, где отображаются данные. Каждая ось может содержать один или несколько графиков.

• Субплоты (subplots): Метод для создания нескольких графиков в одной фигуре, организованных в сетку.

• Легенда (legend): Элемент графика, который помогает идентифицировать различные линии или точки на графике.

• Стили (styles): Наборы параметров, которые определяют внешний вид графиков, включая цвета, шрифты и размеры.

▎Seaborn

• Тепловая карта (heatmap): График, который отображает данные в виде цветовой матрицы, часто используется для визуализации корреляций между переменными.

• Категориальные графики: Seaborn предоставляет удобные функции для создания графиков, таких как boxplot, violinplot, barplot, которые позволяют визуализировать распределение и сравнение категориальных данных.

• Парные графики (pairplot): Позволяет визуализировать взаимосвязи между всеми парами переменных в DataFrame.

▎Plotly

• Интерактивность: Plotly позволяет создавать интерактивные графики, которые можно масштабировать, перемещать и наводить курсор для получения дополнительной информации в режиме реального времени.

• Графики (Figures): Основной объект в Plotly, который может содержать один или несколько графиков (traces).

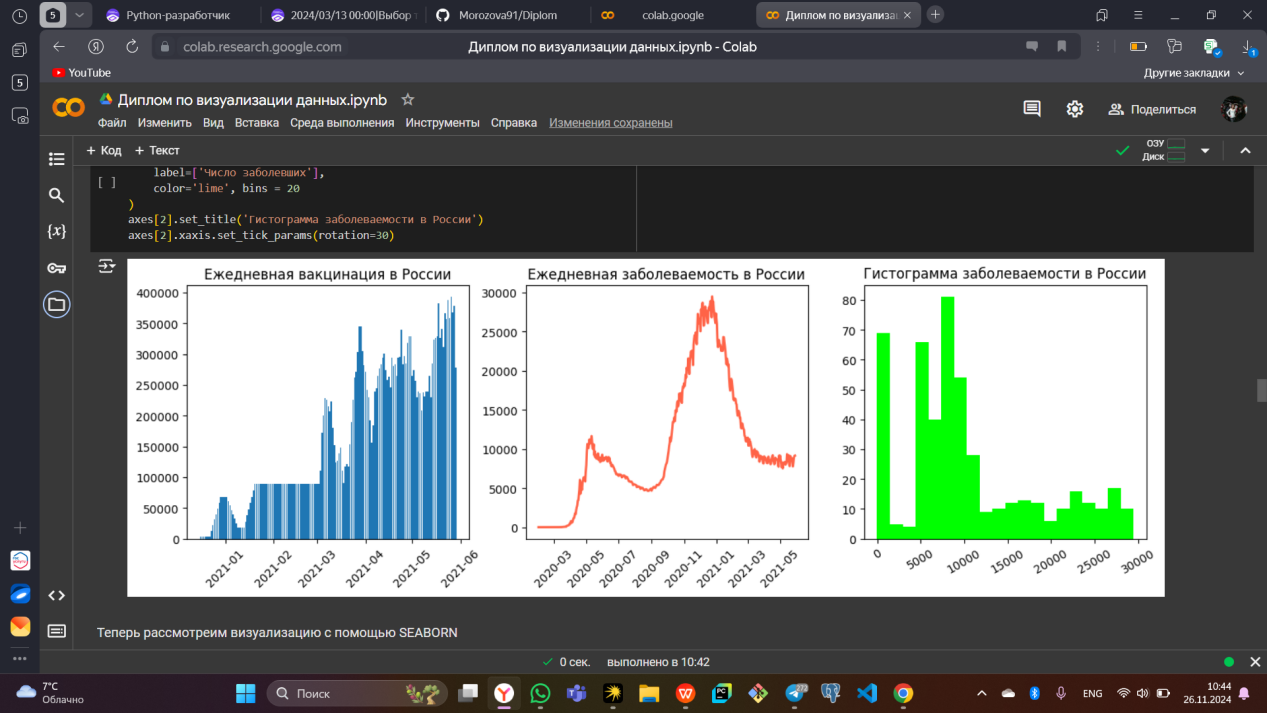
• Трассы (Traces): Элементы графика, представляющие собой данные, которые отображаются, например, линии, точки, столбцы и т. д.

• Разметка (Layout): Параметры, определяющие внешний вид графика, такие как заголовки, оси, легенды и другие элементы.

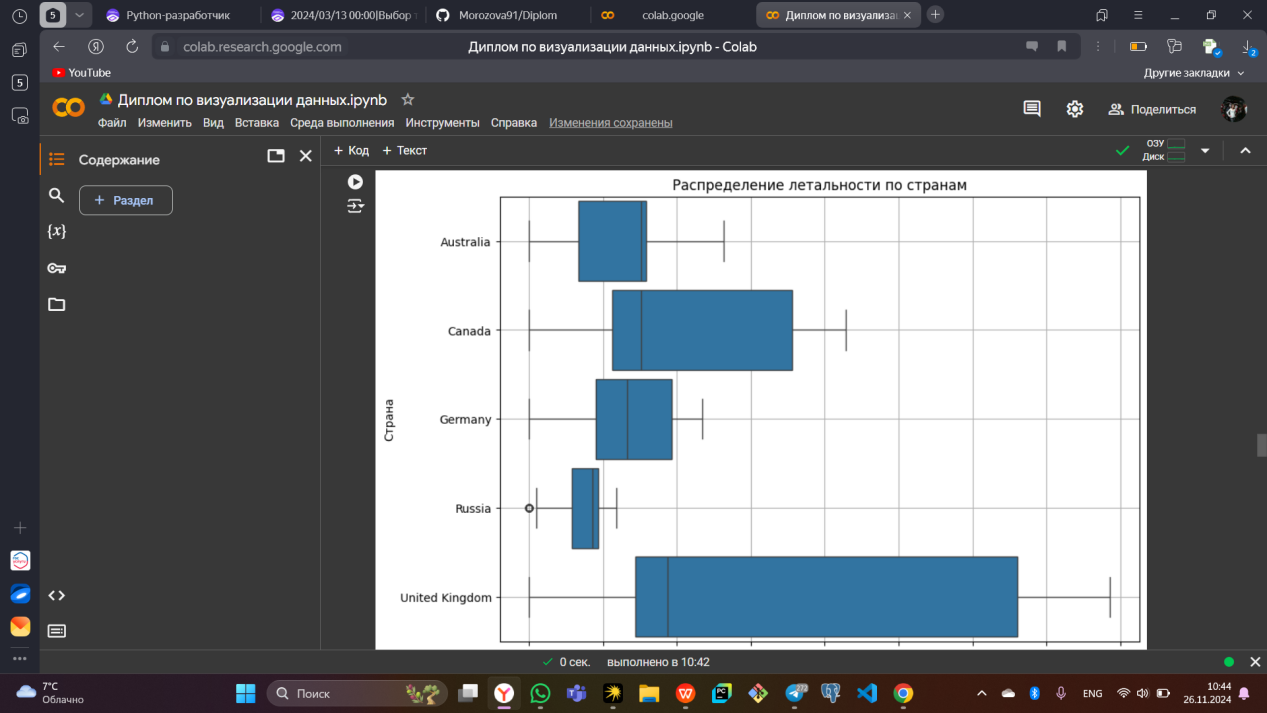
• Dash: Фреймворк для создания веб-приложений на основе Plotly, который позволяет строить интерактивные визуализации и панели управления.

**5. Обзор библиотек:**

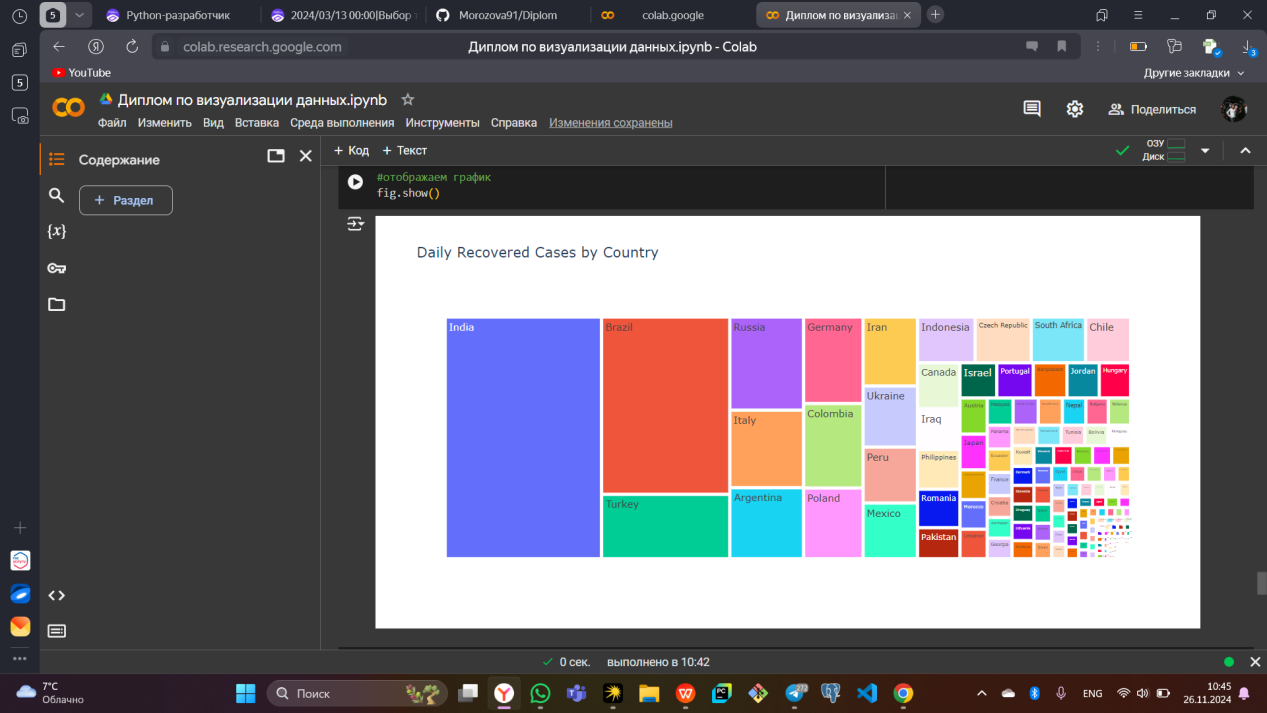
• Matplotlib: Библиотека для создания статических, анимационных и интерактивных визуализаций. Она предоставляет широкий набор инструментов для настройки графиков и является основой для многих других библиотек.



• Seaborn: Библиотека, построенная на основе Matplotlib, которая упрощает создание статистических графиков и улучшает их стиль. Seaborn предлагает высокоуровневый интерфейс для визуализации сложных данных.



• Plotly: Библиотека для создания интерактивных графиков. Она позволяет пользователям взаимодействовать с визуализациями и идеально подходит для веб-приложений и отчетов.



**6. Методы и подходы к визуализации данных.**

В этой секции мы рассмотрим основные методы и подходы, используемые при разработке визуализаций данных с помощью библиотек Matplotlib, Seaborn и Plotly. Эти методы и подходы помогут создать эффективные и информативные графики, которые будут способствовать лучшему пониманию данных.

5.1. Подходы к визуализации данных

1. Определение цели визуализации:

• Прежде чем приступить к разработке визуализации, важно чётко определить её цель. Это может быть исследование данных, представление результатов анализа, создание отчётов или интерактивных панелей управления. Понимание цели поможет выбрать правильные типы графиков и подходы к их созданию.

1. Выбор типа визуализации:

• Столбчатые диаграммы: Хороши для сравнения категорий

• Линейные графики: Подходят для отображения изменений во времени.

• Круговые диаграммы: Используются для представления частей целого, но их следует использовать с осторожностью.

• Точечные диаграммы: Подходят для отображения взаимосвязей между двумя переменными.

• Тепловые карты: Полезны для отображения плотности данных или корреляций.

• Гистограммы: Используются для отображения распределения данных.

3. Аудитория:

• Учитывайте целевую аудиторию визуализации. Разные группы пользователей могут иметь разные уровни подготовки и ожидания. Например, визуализации, предназначенные для научной аудитории, могут требовать более сложных графиков и детального анализа, в то время как для широкой публики лучше использовать более простые и понятные представления.

**7. Методы разработки визуализаций**

1. Подготовка данных:

• Очистка данных: Убедитесь, что данные чистые, без пропусков и аномалий. Это критически важно для точности визуализации.

• Преобразование данных: Возможно, вам потребуется агрегировать данные, изменить формат или создать новые переменные для более информативной визуализации.

1. Проектирование визуализации:

• Элементы дизайна: цветовая палитра, шрифты, размеры и другие элементы дизайна. Визуализация должна быть не только информативной, но и эстетически приятной.

• Упрощение: не должно быть перегруженности графиков лишними элементами. Нужны ключевые данные.

• Подписи и легенды

**8.Преимущества и недостатки Matplotlib, Seaborn и Plotly**

В этой секции мы проведём обзор трёх наиболее популярных библиотек для визуализации данных в Python: Matplotlib, Seaborn и Plotly. Каждая из этих библиотек имеет свои особенности, преимущества и недостатки, которые делают их подходящими для различных задач и типов визуализаций.

8.1. Matplotlib

Описание:

Matplotlib — это одна из самых старых и широко используемых библиотек для визуализации данных в Python. Она предоставляет мощные инструменты для создания статических, анимационных и интерактивных графиков.

Преимущества:

• Гибкость: Matplotlib позволяет создавать широкий спектр графиков, от простых до сложных.

• Настройка: Библиотека предлагает множество параметров для настройки графиков, включая цвета, шрифты, размеры и стили.

• Совместимость: Matplotlib хорошо работает с другими библиотеками, такими как NumPy и Pandas, что делает его удобным для анализа данных.

Недостатки:

• Сложность: Для создания более сложных визуализаций может потребоваться написание большого количества кода.

• Эстетика: По умолчанию графики могут выглядеть менее привлекательно по сравнению с визуализациями, созданными с использованием Seaborn или Plotly.

8.2. Seaborn

Описание:

Seaborn — это библиотека, построенная на основе Matplotlib, которая упрощает создание статистических графиков и улучшает их стиль. Она предлагает высокоуровневый интерфейс для визуализации сложных данных.

Преимущества:

• Удобство: Seaborn позволяет легко создавать красивые и информативные графики с минимальным количеством кода.

• Статистические функции: Библиотека включает множество функций для создания статистических графиков, таких как тепловые карты, диаграммы рассеяния с регрессионными линиями и т.д.

• Эстетика: Seaborn предлагает более привлекательные и современные стили по умолчанию по сравнению с Matplotlib.

Недостатки:

• Гибкость: Хотя Seaborn удобен для создания многих типов графиков, он может быть менее гибким для более сложных визуализаций, требующих детальной настройки.

• Зависимость от Matplotlib: Seaborn использует Matplotlib в качестве основы, поэтому для более сложных настроек может потребоваться знание Matplotlib.

8.3. Plotly

Описание:

Plotly — это библиотека для создания интерактивных графиков. Она позволяет пользователям взаимодействовать с визуализациями, что делает её особенно полезной для веб-приложений и отчетов.

Преимущества:

• Интерактивность: Plotly позволяет создавать интерактивные графики, которые поддерживают наведение курсора, масштабирование и другие элементы взаимодействия.

• Совместимость с веб-технологиями: Plotly хорошо интегрируется с веб-приложениями и может быть использован для создания интерактивных панелей управления.

• Кросс-платформенность: Графики Plotly могут быть легко экспортированы в различные форматы, включая HTML, что делает их удобными для использования в вебе.

Недостатки:

Сложность: Для создания графиков с высокой степенью интерактивности может потребоваться больше времени и усилий.

• Зависимость от интернет-соединения: Некоторые функции Plotly могут требовать подключения к интернету для работы с облачными сервисами.

|  |  |  |  |
| --- | --- | --- | --- |
| Библиотека | Преимущества | Недостатки | Сайт с документацией |
| Matplotlib | -Широкий спектр типов графиков;  -Легко интегрируется с другими библиотеками Python ,такими как NumPy и Pandas;  -Обширная документация и большое сообщество разработчиков. | -Может быть сложной для изучения и использования, особенно для начинающих аналитиков и разработчиков;  -Не поддерживает интерактивность по умолчанию, но может быть расширена с помощью дополнительных библиотек;  - Для создания более сложных визуализаций может потребоваться написание большого количества кода;  -Эстетика: По умолчанию графики могут выглядеть менее привлекательно по сравнению с визуализациями, созданными с использованием Seaborn или Plotly. | <https://matplotlib.org/stable/index.html> |
| Seaborn | Специализируется на создании статистических графиков;  -Более проста в использовании, чем Matplotlib, но все еще требует некоторых знаний в области визуализации данных;  -Более привлекательные и современные стили по умолчанию по сравнению с Matplotlib. | -Не поддерживает интерактивность, но может использоваться для создания статистических графиков;  -Менее гибкий для более сложных визуализаций, требующих детальной настройки;  -Seaborn использует Matplotlib в качестве основы, поэтому для более сложных настроек может потребоваться знание Matplotlib. | <https://seaborn.pydata.org/> |
| Plotly | -Поддерживает интерактивные графики;  -Одна из самых простых библиотек для создания интерактивных графиков;  -Отличная документация и активное сообщество разработчиков;  -Может интегрироваться с другими библиотеками через API (может быть использован для создания интерактивных панелей управления.). | -Для создания графиков с высокой степенью интерактивности может потребоваться больше времени и усилий.  -Зависимость от интернет-соединения: Некоторые функции Plotly могут требовать подключения к интернету для работы с облачными сервисами. | <https://plotly.com/python-api-reference/> |

**9.Анализ и интерпретация результатов.**

**В данном контексте рассмотрим графики и код трёх библиотек Matplotlib, Seaborn, Plotly.**

**Matplotlib**

**import matplotlib.pyplot as plt**

**# Пример данных**

**x = [1, 2, 3, 4, 5]**

**y = [2, 3, 5, 7, 11]**

**# Создание графика**

**plt.plot(x, y)**

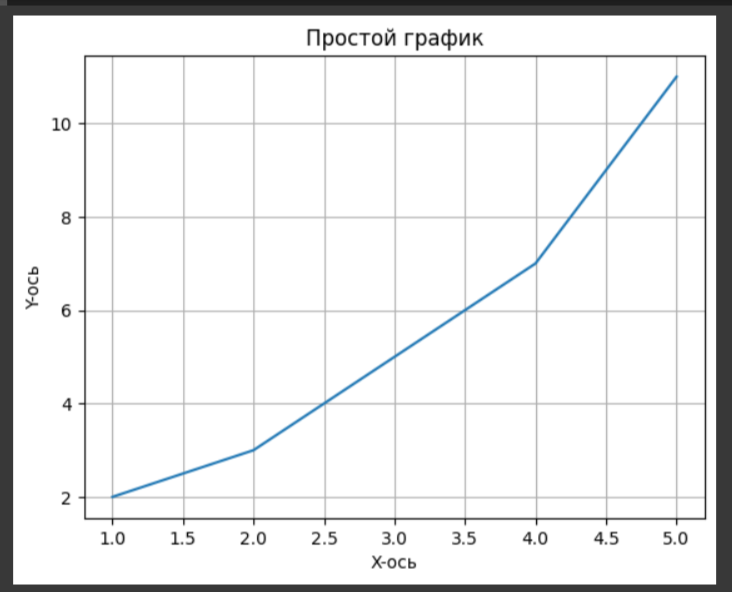
**plt.title('Простой график')**

**plt.xlabel('X-ось')**

**plt.ylabel('Y-ось')**

**plt.grid()**

**plt.show()**

****

**Seaborn**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**# Пример данных**

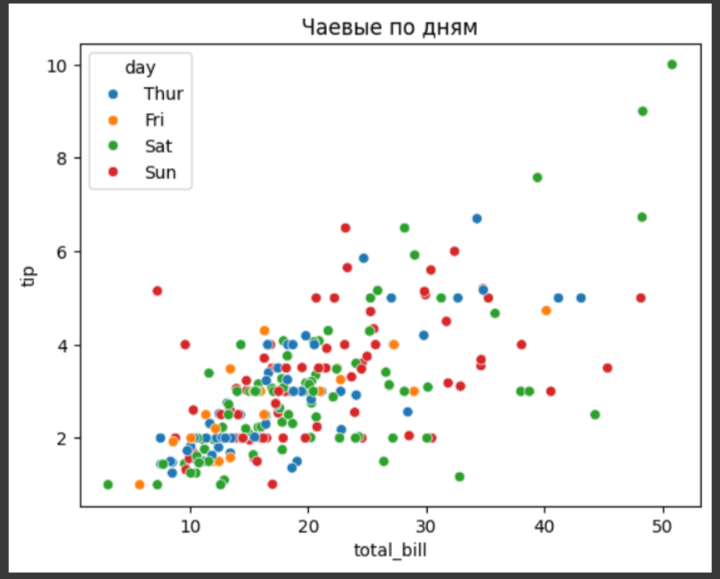
**tips = sns.load\_dataset("tips") --> загружаем встроенный набор данных tips, который содержит информацию о чаевых, оставляемых в ресторане, включая такие параметры, как общая сумма счета (total\_bill), сумма чаевых (tip), день недели (day), количество людей за столом и т.д.**

**# Создание график рассеяния**

**sns.scatterplot(data=tips, x="total\_bill", y="tip", hue="day") --> по оси X отложена общая сумма счета (total\_bill), а по оси Y сумма чаевых (tip). Параметр hue="day" отвечает за цветовую кодировку точек**

**plt.title('Чаевые по дням')**

**plt.show()**

****

**Результат**

**В результате выполнения этого кода получаем график рассеяния, который показывает взаимосвязь между общей суммой счета и суммой чаевых, с различными цветами для каждого дня недели. Это позволяет быстро визуализировать, как чаевые варьируются в зависимости от дня.**

**Plotly**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**import plotly.express as px**

**# Создание DataFrame**

**data = {**

**'Категория': ['A', 'B', 'C', 'A', 'B', 'C'],**

**'Значение': [10, 20, 15, 25, 30, 35]**

**}**

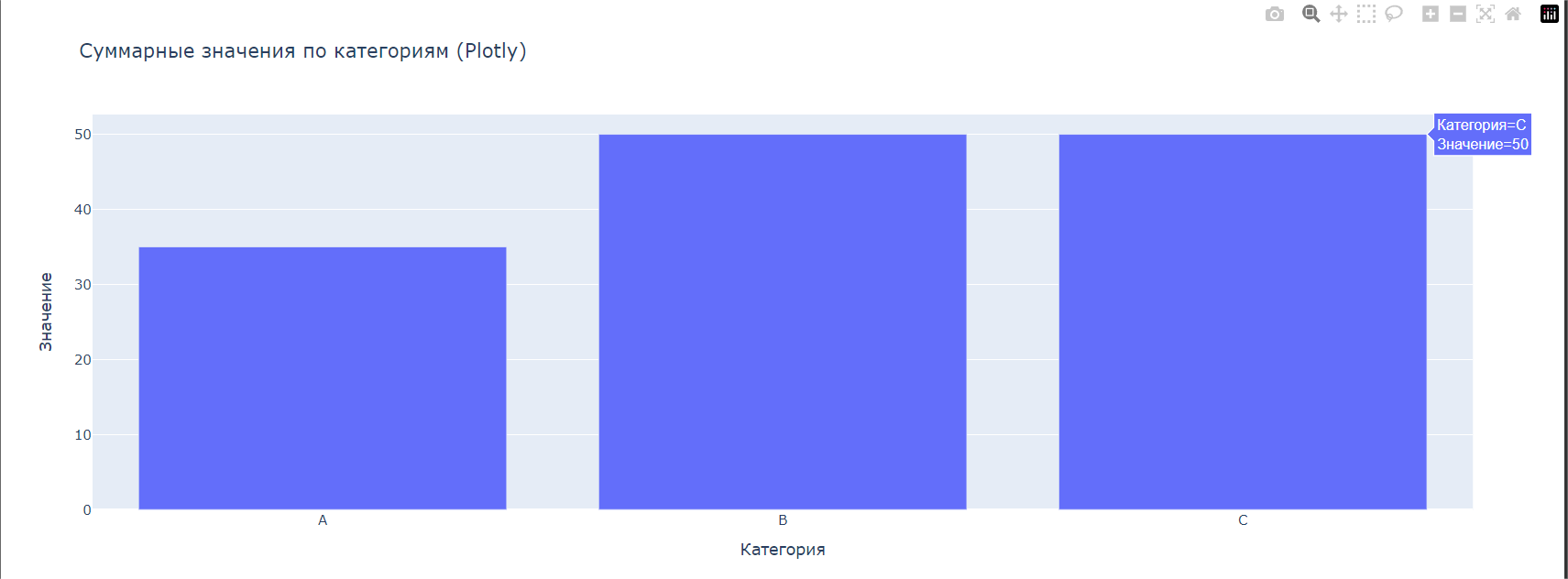
**df = pd.DataFrame(data)**

**# Анализ данных с помощью Pandas**

**summary = df.groupby('Категория').sum()**

**# Создание графика**

**fig = px.bar(summary, x=summary.index, y=”Значение”, title = “Суммарные значения по категориям(Plotly)”) --> Создаем столбчатую диаграмму (bar chart), где по оси X отложены категории, а по оси Y — суммы значений.**

**fig.show()**

**Результат**

**В результате выполнения этого кода вы получите столбчатую диаграмму, которая показывает суммарные значения для каждой категории (A, B и C). Это позволяет визуально оценить, какая категория имеет наибольшее значение.**

**Были протестированы три библиотеки для визуализации данных, как можно увидеть, по коду они довольно схожи, но у каждой свои особенности, такие как, например, интерактивность (Plotly), автоматический подбор цветовой гаммы (Seaborn), в Matplotlib нужно больше кода. Считаю, что выбор библиотеки для визуализации данных зависит от ваших конкретных задач, уровня подготовки и предпочтений. Каждая из перечисленных библиотек имеет свои сильные и слабые стороны, и часто полезно комбинировать их в зависимости от требований проекта.**

**10.Заключение.**

**Визуализация данных является мощным инструментом, который помогает преобразовать сырые данные в полезную информацию. Она облегчает анализ, улучшает коммуникацию и поддерживает принятие решений, делая данные более доступными и понятными для широкой аудитории. В условиях современного мира, где объемы данных постоянно растут, умение эффективно визуализировать информацию становится важным навыком для специалистов в различных областях.**

**Дальнейшие планы.**

**Создать сайт на основе фреймворка Django, с внедрением Pandas, Matplotlib, Seaborn, Plotly, чтобы пользователь мог сам анализировать, к примеру, свои траты. Предложить несколько вариантов построения графиков, данные внесённые пользователем сохраняются и подставляются в графики. Тем самым пользователь сможет распланировать дальнейшие траты, может выделить для себя более важные категории.**