Дипломная работа по теме: Сравнение различных библиотек для визуализации данных

Автор: Белебезьев Анатолий Дмитриевич

Содержание

[**1. Введение** 3](#_Toc183076868)

[**Обоснование выбора темы** 3](#_Toc183076869)

[**Определение цели и задач исследования** 4](#_Toc183076870)

[**2. Краткое сравнение библиотек** 6](#_Toc183076871)

[**3. Подход к разработке** 7](#_Toc183076872)

[Matplotlib 7](#_Toc183076873)

[Seaborn 10](#_Toc183076874)

[Plotly 13](#_Toc183076875)

[**4. Анализ и интерпретация результатов** 17](#_Toc183076876)

[Сравнение библиотек 17](#_Toc183076877)

[Интерпретация результатов и рекомендации к использованию 18](#_Toc183076878)

[**5. Структура программы** 19](#_Toc183076879)

[**6. Заключение** 20](#_Toc183076880)

[Обзор выполненной работы 20](#_Toc183076881)

[Дальнейшие планы 20](#_Toc183076882)

[**7. Список литературы** 21](#_Toc183076883)

# **1. Введение**

## **Обоснование выбора темы**

1. **Актуальность визуализации данных**

* Современный мир переполнен данными, которые важно не только собирать, но и эффективно интерпретировать. Визуализация играет ключевую роль в анализе данных, помогая исследователям, аналитикам и разработчикам быстро находить закономерности, выявлять аномалии и доносить информацию до аудитории. Тема сравнения библиотек актуальна, так как выбор подходящего инструмента для визуализации может значительно упростить процесс анализа.

1. **Широкий выбор инструментов**

* В Python существует множество библиотек для визуализации, таких как Matplotlib, Seaborn, Plotly и другие. Каждая из них обладает уникальными возможностями, преимуществами и ограничениями, что создает необходимость их сравнения для определения лучшего решения под конкретные задачи.

1. **Универсальность применения**

* Визуализация данных используется в различных областях: науке, бизнесе, маркетинге, образовании и других. Выбор подходящего инструмента может существенно повлиять на качество анализа и принятие решений. Поэтому важно понять, в каких случаях предпочтительнее использовать ту или иную библиотеку.

1. **Образовательная ценность**

* Сравнение библиотек — полезная тема для студентов, аналитиков и разработчиков, которые только начинают работать с данными. Это позволяет быстро ознакомиться с ключевыми инструментами и выбрать наиболее подходящий для текущих или будущих проектов.

1. **Развитие навыков**

* Тема способствует не только изучению основ визуализации данных, но и позволяет освоить особенности работы с различными типами графиков, настройки параметров, а также познакомиться с интерактивными и статичными подходами к построению графиков.

1. **Помощь в оптимизации процессов**

* Правильный выбор библиотеки может ускорить процессы анализа данных, сделать графики более читаемыми и улучшить взаимодействие с командой или клиентами. Поэтому важно понять, какой инструмент лучше всего справляется с конкретной задачей.

## **Определение цели и задач исследования**

Цель исследования: В данной работе будут рассмотрены три популярные библиотеки для визуализации данных в Python: **Matplotlib**, **Seaborn** и **Plotly**. Сравним их по ключевым параметрам, включая простоту использования, возможности настройки, типы поддерживаемых графиков и интерактивность, чтобы помочь определить, какой из них лучше подходит для конкретных задач.

Задачи исследования:

1. Краткий обзор библиотек **Matplotlib**, **Seaborn** и **Plotly.** Описание их преимуществ и недостатков.
2. Создание кода с использованием этих библиотек с различными настройками, данными и типами графиков.
3. Реализация возможности последовательного запуска примеров с этими библиотеками.
4. Написание дипломной работы: составить дипломную работу, включающую в себя введение, обзор материалов, методологию и результаты исследования, анализ результатов, выводы и рекомендации.

Цели и задачи исследования направлены на получение практических результатов, которые позволят показать преимущества и более подходящие места использования обозначенных выше библиотек.

# **2. Краткое сравнение библиотек**

1. **Matplotlib**

* **Описание**: это базовая библиотека для визуализации данных в Python, которая предоставляет гибкость и полный контроль над графиками. Она служит основой для многих других библиотек, включая Seaborn.

1. **Seaborn**

* **Описание**: Библиотека, построенная на основе Matplotlib, предназначенная для более удобного и эстетичного создания визуализаций. Она отлично подходит для работы с данными в формате DataFrame (например, из pandas).

1. **Plotly**

* **Описание**: Библиотека для интерактивной визуализации данных, которая поддерживает как веб-приложения, так и локальные графики. Отлично подходит для сложных и динамических визуализаций.

# **3. Подход к разработке**

Для сравнения способов визуализации, создадим простое приложение. Там будет выбор библиотеки для проверки. После выбора программа будет показывать базовые графики для показа внешнего вида этого метода. После будут рисоваться уже более сложные графики, объединённые вместе для показа простоты или сложности для восприятия и для показа комплексного изображения графиков.

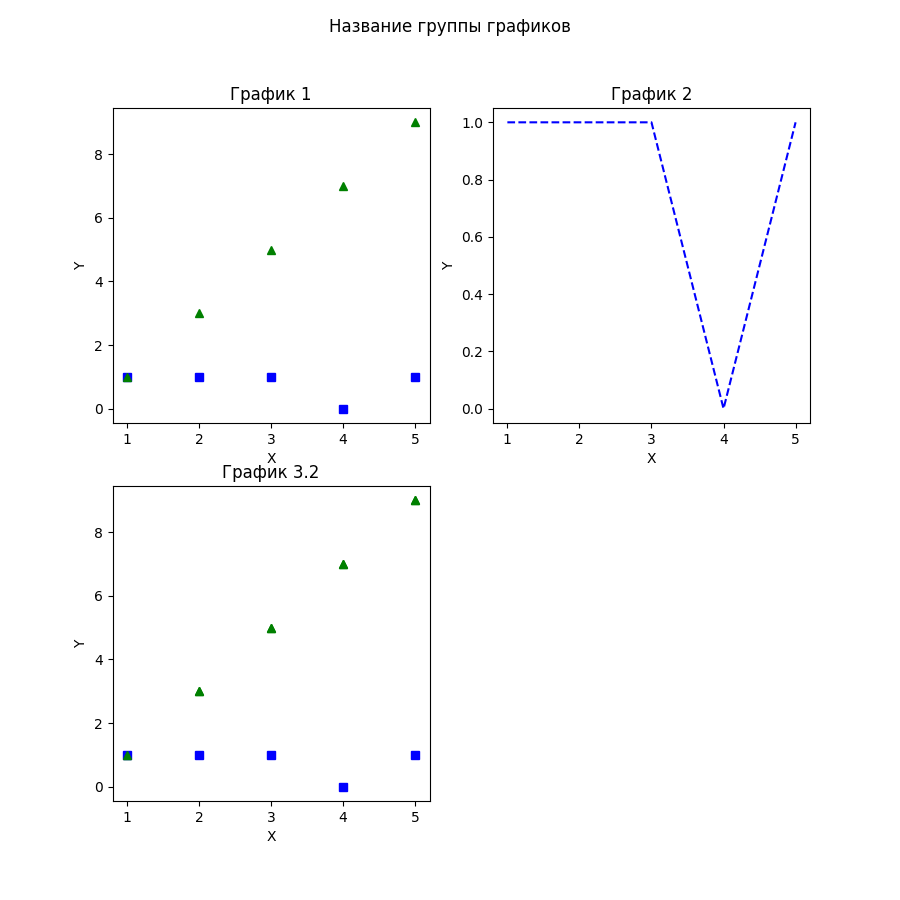
Создадим три отдельных файла для независимой проверки всех трёх библиотек.

Для работы и проверки качества и простоты связи с базами данных, создадим файл для работы с базой данных и саму базу данных.

Для демонстрации визуализации, будут использоваться заранее созданные данные как из базы данных, так и уже вписанные в программу.

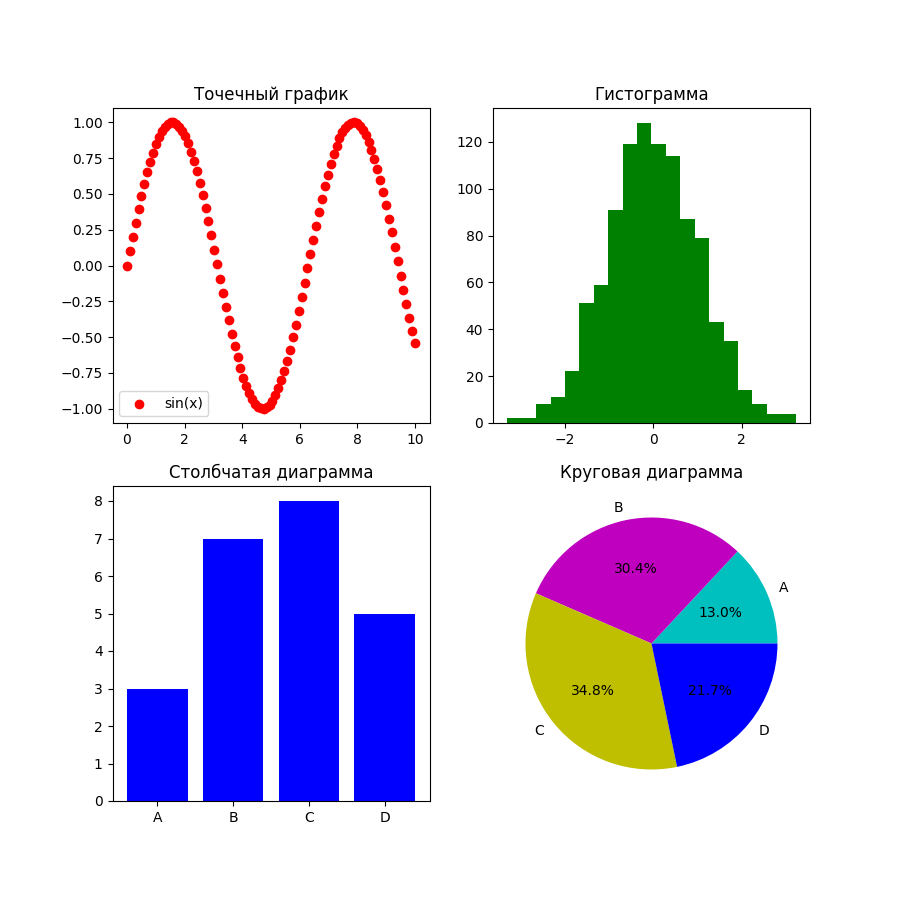
## Matplotlib

Создаём файл связанный с Matplotlib. Первая часть задачи показать графики из этой библиотеки для последующего сравнения.   
Согласно плану, делим эту визуализацию на части:

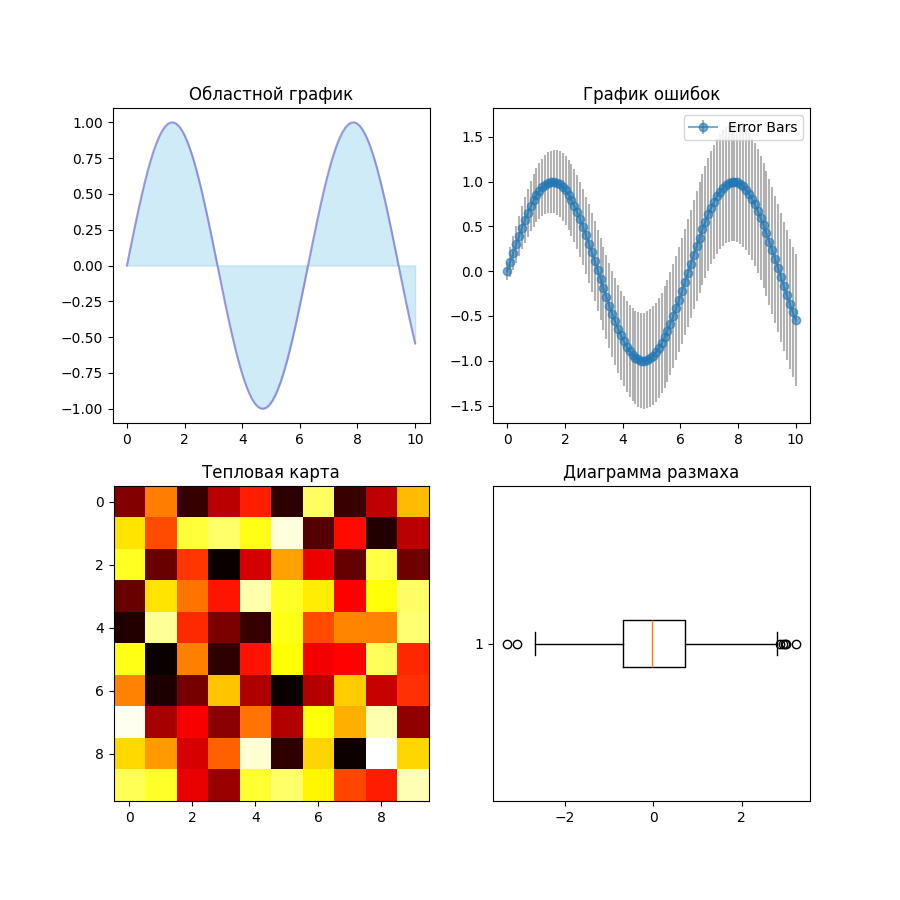
Первая функция строит для нас линейные графики. В созданном окне можно изучить возможности масштабирования графиков и остальные возможности (по типу сохранения графика) рис.1

Рисунок

Последующие функции рисуют более сложные графики с разными настройками и визуальными эффектами. Их визуальную составляющую и будем сравнивать с другими библиотеками. Рис2-3



Рисунок



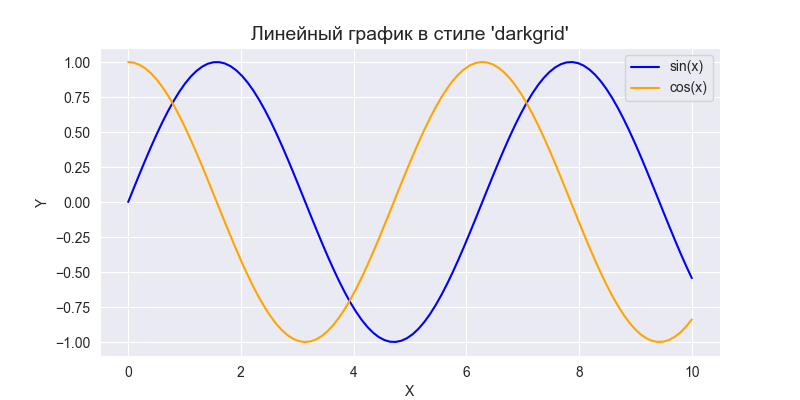
Рисунок

Вторая часть задания – сравнение простоты использования. В коде для реализации этих методов приходится использовать большое количество дополнительных параметров, а часть данных переводить в Numpy Array для удобства.

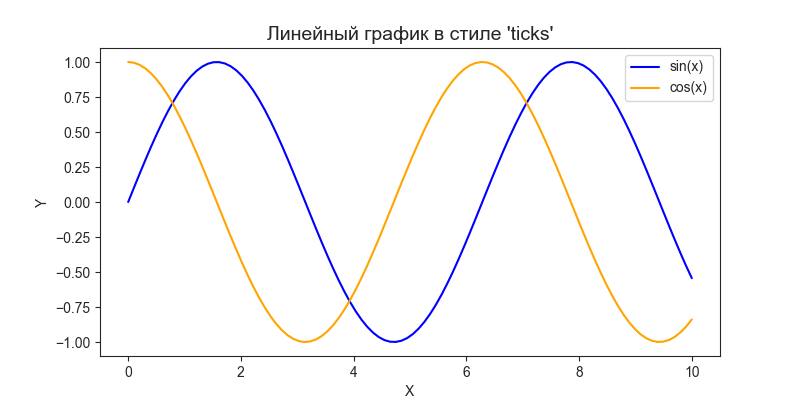
## Seaborn

Создаём файл связанный с Seaborn. Первая часть задачи показать графики из этой библиотеки для последующего сравнения.   
Согласно плану, делим эту визуализацию на части:

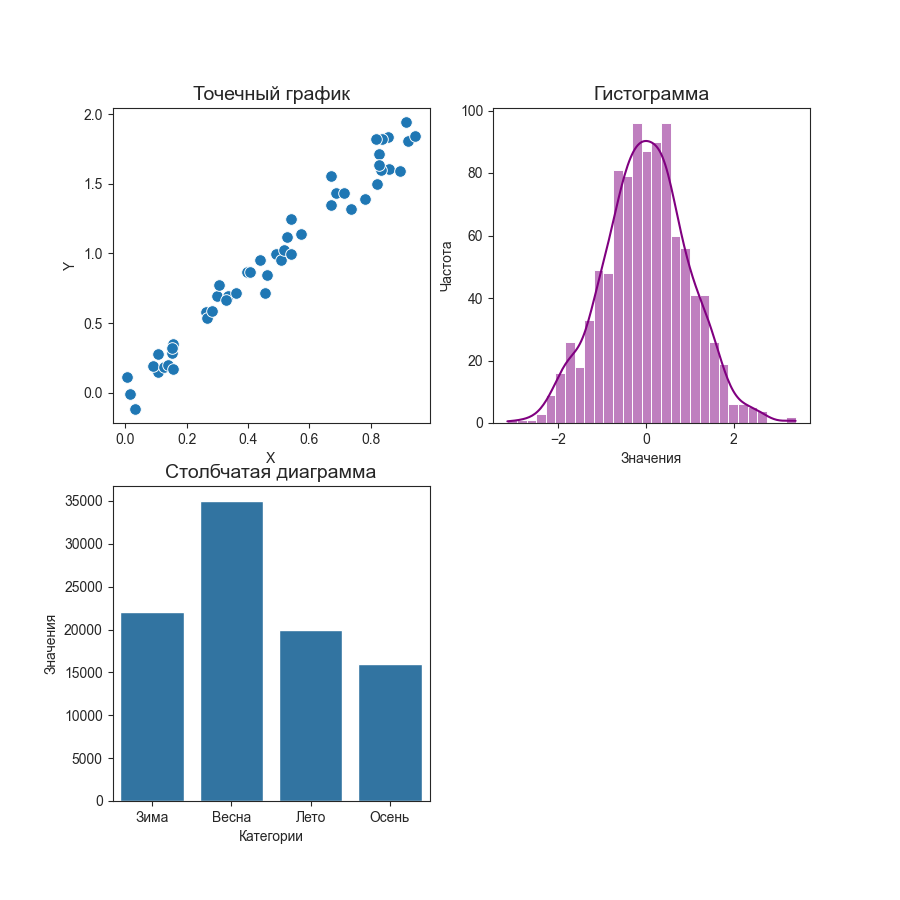
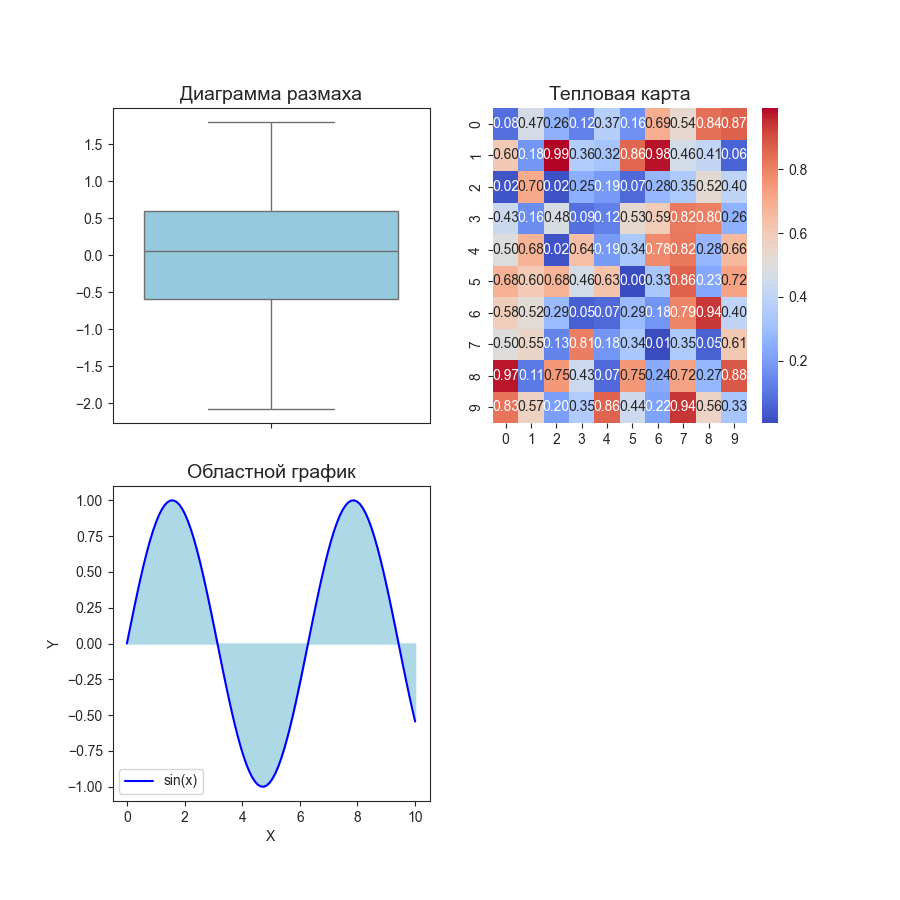
первая функция строит для нас линейные графики. В созданном окне можно изучить возможности масштабирования графиков и остальные возможности (по типу сохранения графика) рис.4-5



Рисунок



Рисунок

Последующие функции рисуют более сложные графики с разными настройками и визуальными эффектами. Их визуальную составляющую и будем сравнивать с другими библиотеками. Рис6-7Рисунок 

Рисунок

Из получившихся результатов, мы видим, что круговой график не имеет способа создания в seaborn. Но создание всех остальных графиков проще с точки зрения визуала и для части данных не нужно преобразовывать данные из баз данных.

## Plotly

Создаём файл связанный с Plotly. Первая часть задачи показать графики из этой библиотеки для последующего сравнения.   
Согласно плану, делим эту визуализацию на части:

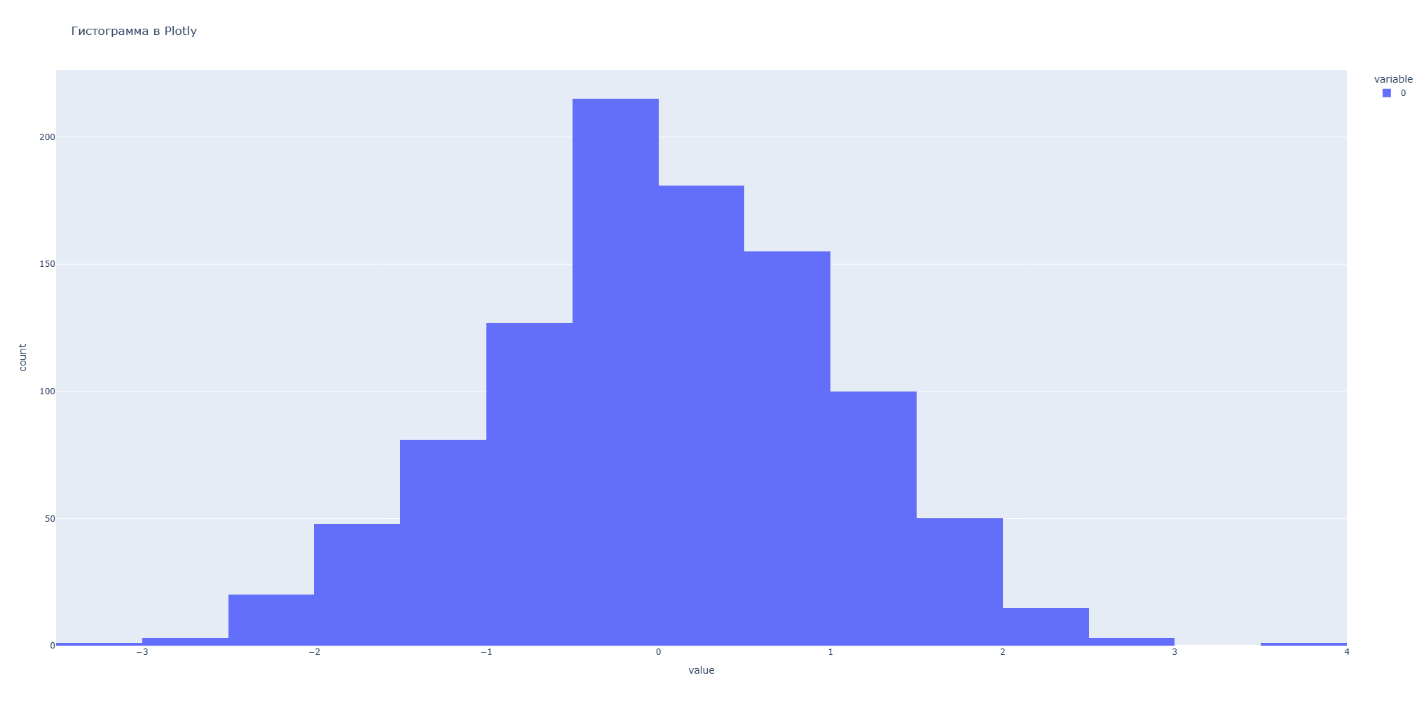
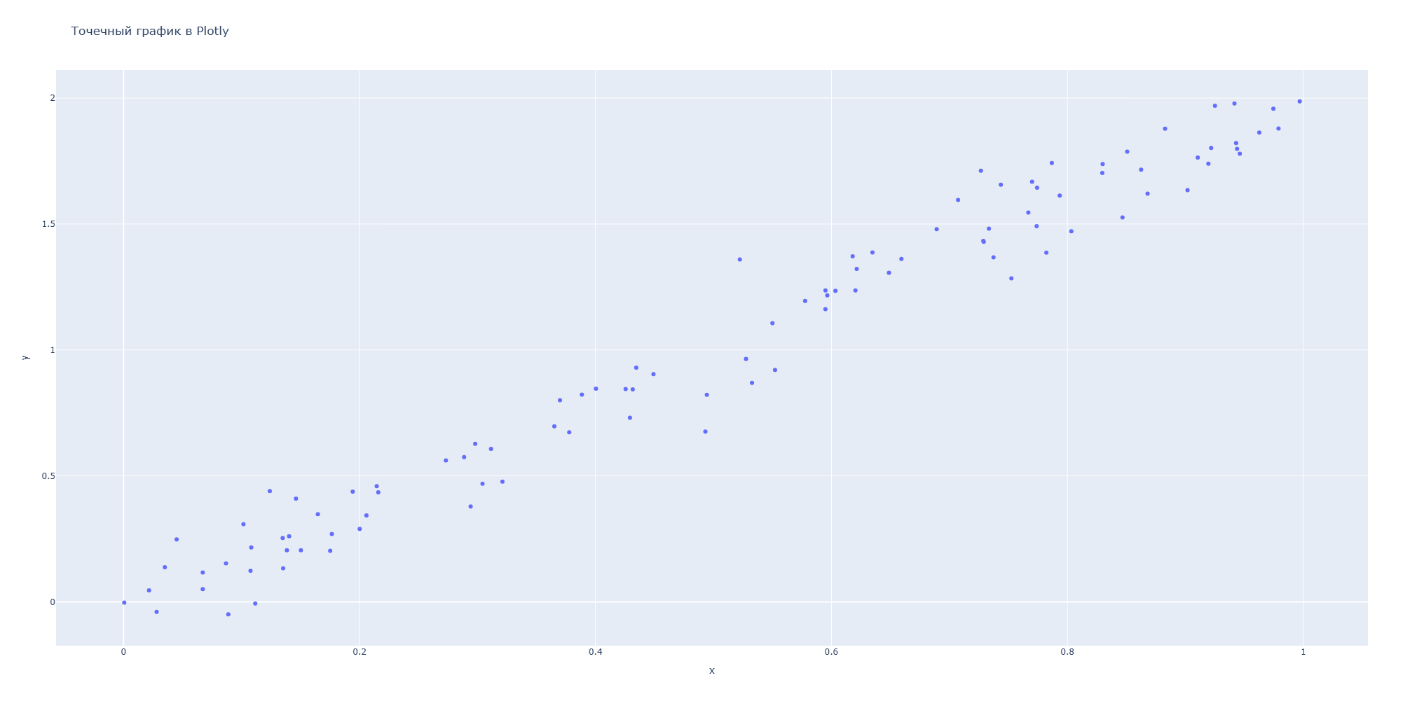
Здесь немного другая логика построения графиков. В первом случае, создаём графики через plotly.express. В созданном окне можно изучить возможности масштабирования графиков и остальные возможности, включая интерактивное включение и отключение элементов. Рис8-12

Рисунок 

Рисунок

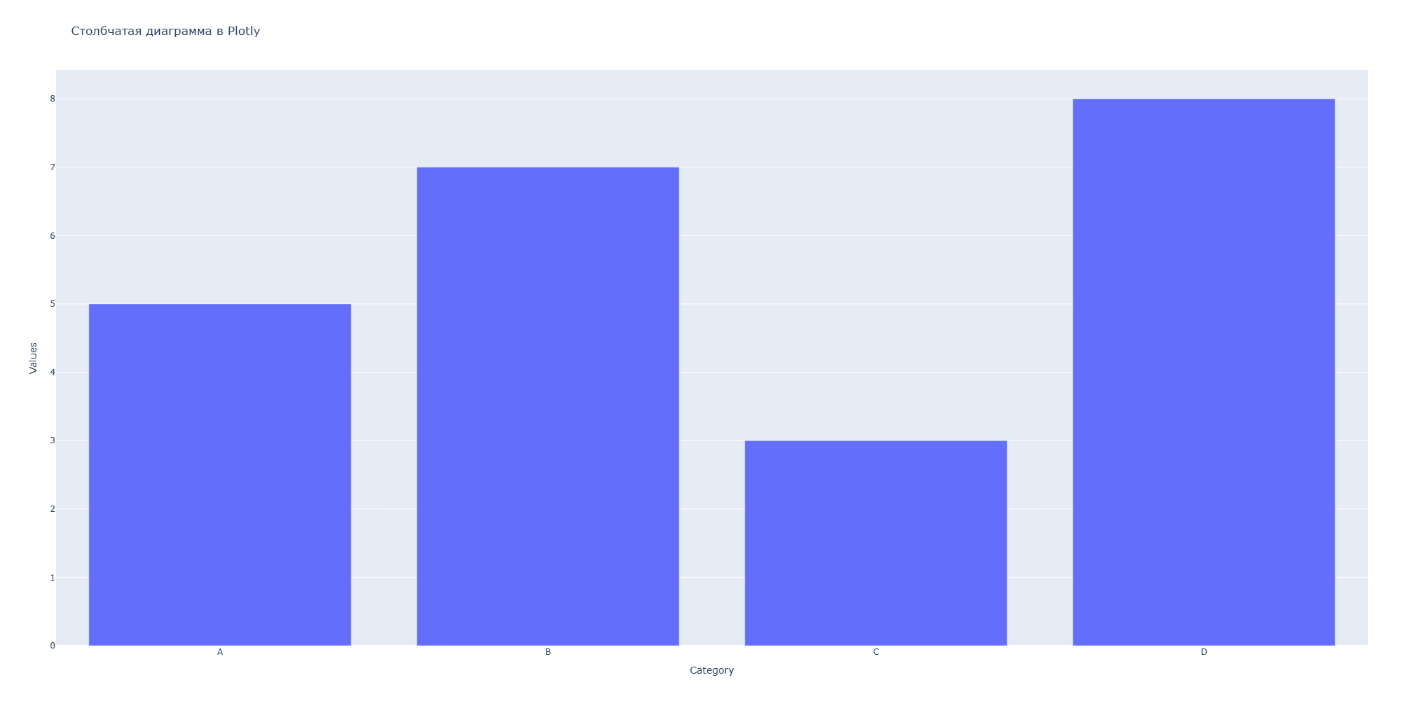
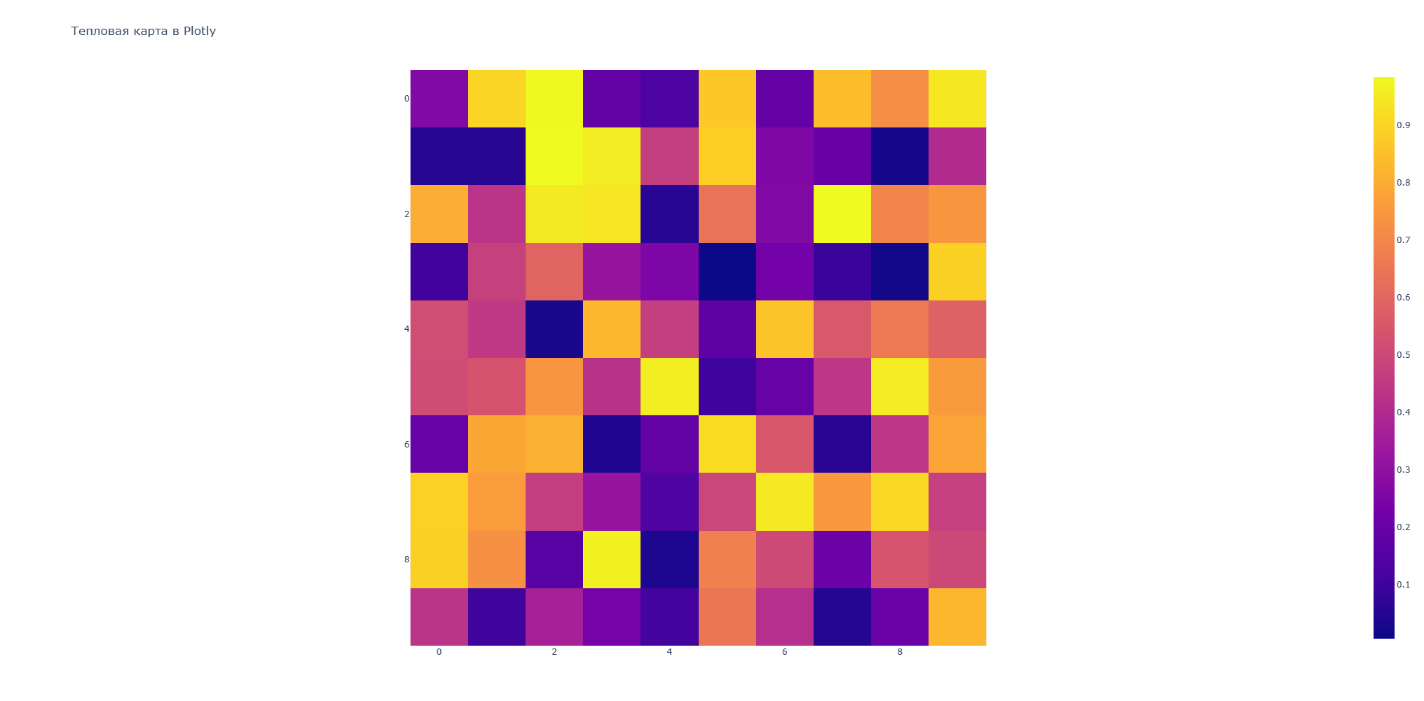
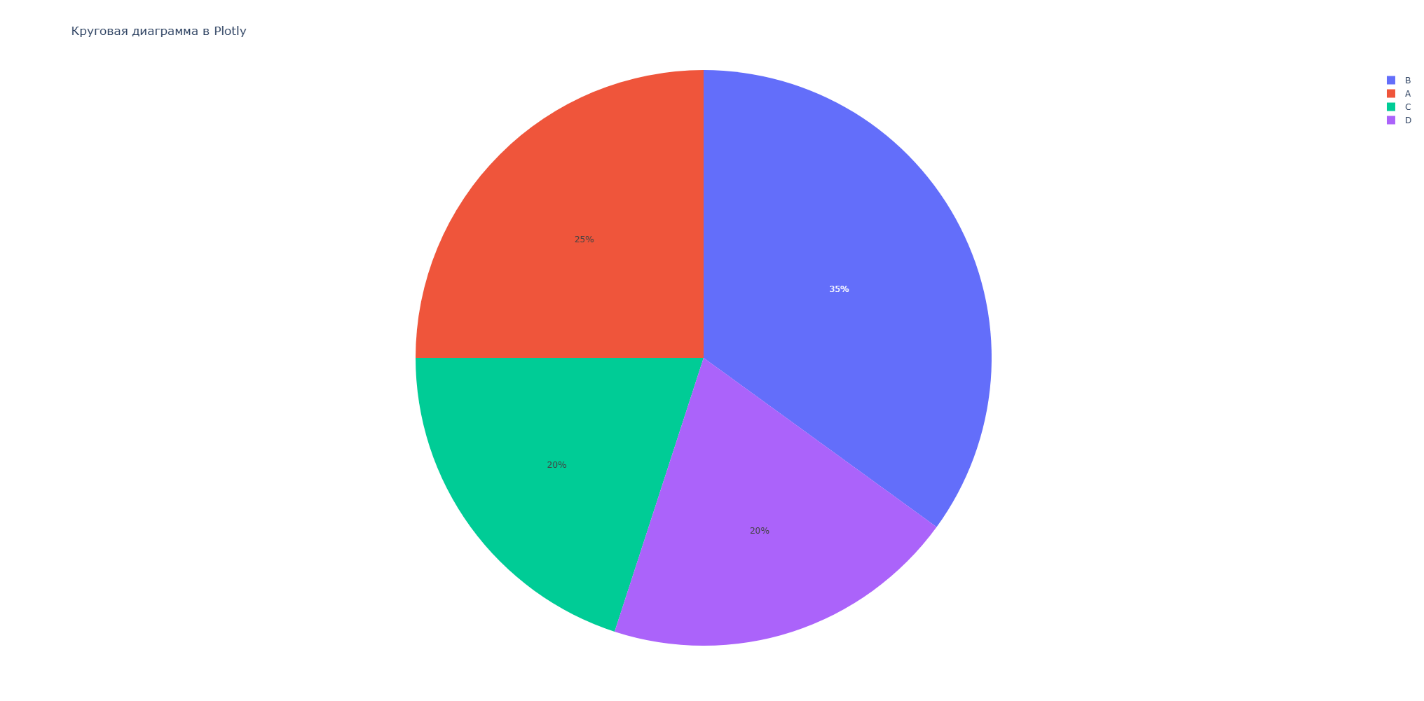


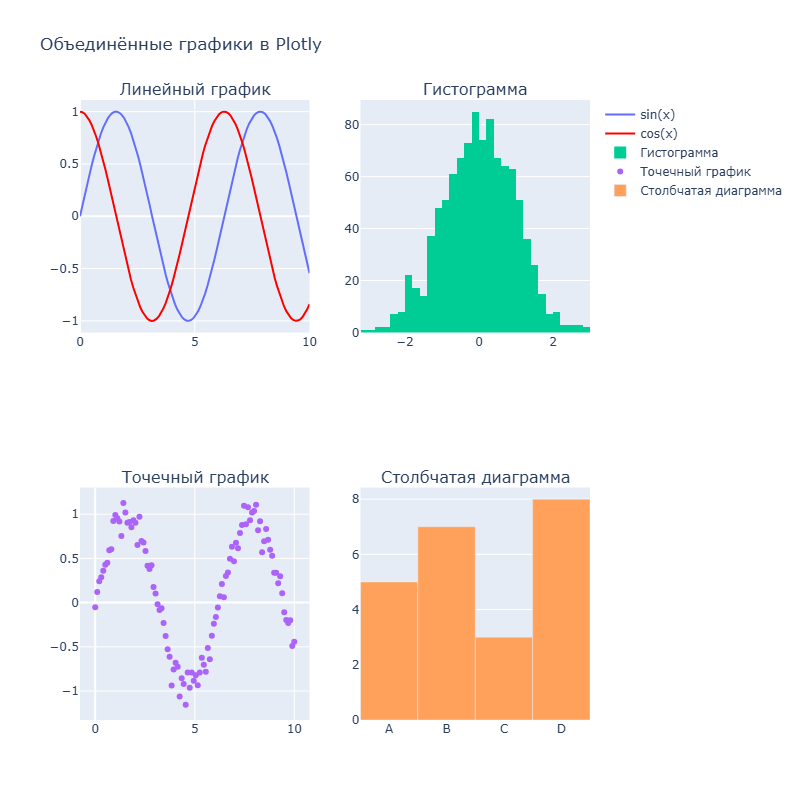
Рисунок 

Рисунок



Рисунок

Во второй части, создаём графики с помощью plotly.graph\_objects. И для показа возможности добавляем графики на один экран. Рис13



Рисунок

Данная библиотека обладает гораздо более сложным синтаксисом, но большей функциональностью и встроенной поддержкой веба.

# **4. Анализ и интерпретация результатов**

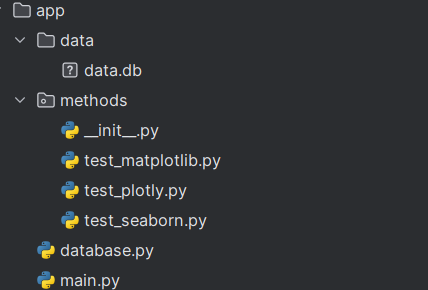
## Сравнение библиотек

В ходе работы, были продемонстрированы основные инструменты из библиотек Matplotlib, Seaborn, Plotly. По мере разбора всех трёх библиотек были показаны преимущества и недостатки, а также возможные сферы применения.

## Интерпретация результатов и рекомендации к использованию

1. **Matplotlib**  
   Функциональность: Основной инструмент для создания визуализаций в Python с гибкой настройкой. Поддерживает 2D- и 3D-графики, такие, как гистограммы, графики рассеяния и контурные диаграммы.  
   Удобство использования: требует больше кода, особенно для сложных визуализаций, и имеет простой, но менее эстетичный дизайн по умолчанию.  
   Лучшее применение: подходит для научных и исследовательских проектов, где требуется строгий контроль над деталями.
2. **Seaborn**  
   Функциональность: Построен поверх Matplotlib, предлагая высокоуровневые API для быстрого создания стильных графиков и работы с данными. Включает встроенные статистические графики (например, boxplots, violinplots, heatmaps).  
   Удобство использования: упрощает создание графиков и легко интегрируется с Pandas. Меньше необходимости в настройке стилей, так как Seaborn использует красивые, готовые по умолчанию стили.  
   Лучшее применение: Идеален для исследовательского анализа данных (EDA), особенно когда нужны быстро и легко создаваемые статистические графики.
3. **Plotly**  
   Функциональность: поддерживает интерактивные графики, 3D-диаграммы, карты и другие сложные визуализации. Можно встраивать в веб-приложения и взаимодействовать с графиками в режиме реального времени.  
   Удобство использования: Графики не так просты в создании, но поддерживают интерактивность. Библиотека может быть сложнее для адаптации по сравнению с Seaborn или Matlibplot.  
   Лучшее применение: прекрасно подходит для дашбордов и аналитики, особенно если визуализации должны быть интерактивными или отображаться в веб-приложениях.

# **5. Структура программы**



# **6. Заключение**

## Обзор выполненной работы

Подводя итоги, все три реализации библиотек работают и позволяют изучить внешний вид и внутреннее устройство визуализации данных. Каждая библиотека имеет свои преимущества в зависимости от целей, но часто используется комбинация из них: Matplotlib для базовых графиков, Seaborn для исследовательской аналитики и Plotly для интерактивных приложений.

## Дальнейшие планы

Возможны улучшения программы: увеличение списка исследуемых графиков, добавление выбора отдельных графиков и возможность их произвольно объединять, реализация добавления пользовательских данных, большая поддержка баз данных и веб-интерфейсов.

# **7. Список литературы**

<https://matplotlib.org/stable/users/index.html> - документация Matplotlib

<https://seaborn.pydata.org/tutorial.html> - документация Seaborn

<https://plotly.com/python/> - документация Plotly