Дипломная работа

на тему:

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

Мирошник Д.Г.

Содержание

[Введение 4](#_Toc184755112)

[1. Теоретические основы визуализации данных 7](#_Toc184755113)

[1.1. Понятие и цели визуализации данных 7](#_Toc184755114)

[1.2. История и эволюция методов визуализации 7](#_Toc184755115)

[1.3. Важность визуализации данных в современном мире 8](#_Toc184755116)

[1.4. Популярные инструменты визуализации данных 8](#_Toc184755117)

[1.5. Краткий обзор библиотек Matplotlib, Seaborn и Plotly 9](#_Toc184755118)

[1.5.1. Matplotlib 9](#_Toc184755119)

[1.5.2. Seaborn 10](#_Toc184755120)

[1.5.3. Plotly 10](#_Toc184755121)

[2. Методология исследования 12](#_Toc184755122)

[2.1. Постановка задачи 12](#_Toc184755123)

[2.2. Описание подхода к тестированию 12](#_Toc184755124)

[2.3. Набор тестов для визуализации данных 13](#_Toc184755125)

[2.4. Данные для исследования 14](#_Toc184755126)

[2.5. Тестирование 14](#_Toc184755127)

[3. Практическое сравнение Matplotlib, Seaborn и Plotly 15](#_Toc184755128)

[3.1. Данные для тестирования 15](#_Toc184755129)

[3.2. Построение базовых графиков 15](#_Toc184755130)

[3.2.1. Линейный график 15](#_Toc184755131)

[3.2.2. Гистограмма распределения данных 17](#_Toc184755132)

[3.3. Построение 3D-поверхности 19](#_Toc184755133)

[4. Обсуждение результатов 22](#_Toc184755134)

[4.1. Изучение и тестирование библиотек визуализации 22](#_Toc184755135)

[4.2. Реализация программы «GraphPythonMag» 22](#_Toc184755136)

[4.3. Описание работы программы 23](#_Toc184755137)

[4.4. Предложения по улучшению программы 24](#_Toc184755138)

[Заключение 26](#_Toc184755139)

[Список литературы 28](#_Toc184755140)

# Введение

Современные методы обработки и анализа данных становятся ключевыми инструментами для решения широкого спектра задач в различных отраслях: науке, бизнесе, медицине, промышленности и других. Однако сами по себе данные, особенно в больших объемах, зачастую сложно интерпретировать без использования средств визуализации. Визуализация данных позволяет не только представлять информацию в удобной и наглядной форме, но и находить закономерности, выявлять аномалии и принимать обоснованные решения.

Python, как один из наиболее популярных языков программирования в области анализа данных, предоставляет богатый набор библиотек для визуализации. Среди них Matplotlib, Seaborn и Plotly занимают особое место. Каждая из этих библиотек имеет свои уникальные возможности и ограничения, что делает их использование подходящим для различных задач и аудиторий. Тем не менее, выбор подходящей библиотеки зачастую вызывает затруднения у аналитиков, особенно начинающих.

Актуальность работы также связана с потребностями профессионального геофизического сообщества. В профессиональных обрабатывающих комплексах для геофизических данных функционал построения графиков зачастую ограничен, а их разрешение оставляет желать лучшего. Библиотеки Python позволяют преодолевать эти ограничения, создавая графики типографического качества, интерактивные визуализации и 3D-поверхности.

Цель исследования – провести сравнительный анализ библиотек Matplotlib, Seaborn и Plotly для определения их преимуществ, недостатков и областей применения, с акцентом на работу с геофизическими данными.

Для достижения этой цели были поставлены следующие задачи:

– Изучить теоретические основы визуализации данных и роль Python-библиотек в этой области;

– Определить критерии, по которым будет проводиться сравнение библиотек;

– Провести тестирование библиотек на реальных геофизических данных;

– Оценить функциональные возможности, производительность и удобство использования каждой библиотеки при обработке и визуализации геофизической информации;

– Разработать программу с графическим интерфейсом для визуализации данных;

– Сделать выводы и предоставить рекомендации по выбору библиотеки в зависимости от задач.

Объектом исследования являются библиотеки для визуализации данных. Предметом исследования выступают функциональные возможности, производительность и удобство использования библиотек Matplotlib, Seaborn и Plotly при обработке и визуализации геофизических данных.

Для проведения исследования использовались следующие методы:

– Изучение официальной документации библиотек, научных публикаций и обучающих материалов;

– Разработка и сравнение одинаковых графиков с использованием каждой библиотеки;

– Оценка библиотек по определенным критериям, включая функциональность, производительность, интерактивность, визуальную эстетику и простоту интеграции;

– Создание программы с графическим интерфейсом и прототипа веб-приложения.

Работа содержит комплексный подход к сравнению трех популярных библиотек визуализации, включая анализ их возможностей в контексте работы с реальными геофизическими данными. Впервые проводится сравнительный анализ с акцентом на практическую применимость в профессиональных геофизических системах.

Практическая значимость исследования заключается в следующем:

Разработанные рекомендации помогут специалистам эффективно выбирать инструменты визуализации для задач, связанных с обработкой и анализом геофизических данных.

Библиотеки Matplotlib, Seaborn и Plotly позволяют преодолевать ограничения профессиональных геофизических комплексов, создавая графики типографического качества, интерактивные визуализации и 3D-поверхности.

В рамках исследования была реализована программа с графическим интерфейсом, облегчающая работу с данными.

Таким образом, данная работа является актуальной и практически значимой, предлагая структурированный подход к сравнению наиболее востребованных библиотек Python для визуализации данных.

# 1. Теоретические основы визуализации данных

## 1.1. Понятие и цели визуализации данных

Визуализация данных представляет собой процесс представления данных в графическом или визуальном формате, который позволяет пользователю интерпретировать и анализировать информацию более эффективно. Это одна из ключевых стадий работы с данными, которая используется для:

– Поддержки принятия решений;

– Поиска закономерностей, аномалий и трендов;

– Облегчения понимания сложных взаимосвязей;

– Представления результатов анализа в удобной для восприятия форме.

Научные исследования, управление бизнес-процессами, медицина и инженерия – это лишь некоторые области, где визуализация данных играет критически важную роль. Например, в геофизике визуализация используется для интерпретации данных измерений, таких как сейсмические сигналы, изменения гравитационных или электромагнитных полей.

## 1.2. История и эволюция методов визуализации

История визуализации данных берет свое начало с XVII века, когда Уильям Плейфер разработал первые графики – линейный график, столбчатую диаграмму и круговую диаграмму. [4] Позднее, в XIX веке, появились более сложные методы, такие как тепловые карты и контурные диаграммы, которые начали применяться для географических и научных целей.

С развитием вычислительной техники визуализация данных стала ключевым инструментом в анализе больших массивов информации. Появление персональных компьютеров в 1980-х годах и внедрение специализированного программного обеспечения, такого как Tableau и MATLAB, сделало возможным быстрое создание сложных визуализаций. Сегодня широкое распространение языков программирования, таких как Python и R, позволяет пользователям гибко обрабатывать и визуализировать данные с минимальными затратами.

## 1.3. Важность визуализации данных в современном мире

Современный мир переживает взрыв данных. Объем информации, создаваемой ежедневно, измеряется эксабайтами, и без визуализации такой массив данных практически невозможно анализировать. Основные преимущества визуализации включают:

– Ускорение анализа данных, графики и диаграммы позволяют быстро интерпретировать результаты;

– Простота коммуникации, визуальные формы данных легче воспринимаются аудиторией, особенно в междисциплинарных командах;

– Демонстрация сложных связей, интерактивные графики помогают выявить корреляции и зависимости.

В профессиональных сферах, таких как геофизика, визуализация данных играет особую роль. Геофизические данные часто являются многомерными и сложными для интерпретации. Визуализация помогает специалистам представить результаты измерений в виде, удобном для анализа и принятия решений.

## 1.4. Популярные инструменты визуализации данных

На сегодняшний день существует множество инструментов визуализации данных, каждый из которых обладает уникальными преимуществами и ограничениями. Наиболее популярные из них приведены в таблице 1 [5-7].

Таблица 1. Сравнительная таблица популярных инструментов визуализации данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Инструмент | Тип графиков | Интерактивность | Легкость освоения | Область применения | Стоимость |
| Excel | 2D | Низкая | Высокая | Базовые визуализации | Лицензия (MS Office) |
| Tableau | 2D, интерактивные | Высокая | Средняя | Бизнес-аналитика | Высокая |
| Power BI | 2D, интерактивные | Высокая | Средняя | Бизнес-аналитика | Средняя |
| Matplotlib | 2D, 3D (ограниченно) | Низкая | Низкая | Наука, публикационные графики | Бесплатно |
| Seaborn | 2D | Низкая | Высокая | Статистика, исследовательский анализ | Бесплатно |
| Plotly | 2D, 3D, интерактивные | Высокая | Средняя | Веб-приложения, интерактивные отчеты | Бесплатно |
| ggplot2 | 2D | Низкая | Средняя | Наука, статистика | Бесплатно |
| D3.js | Любые | Высокая | Низкая | Веб-приложения, сложные графики | Бесплатно |

## 1.5. Краткий обзор библиотек Matplotlib, Seaborn и Plotly

Библиотеки Matplotlib, Seaborn и Plotly являются ключевыми инструментами визуализации данных в Python. Они предлагают пользователям широкий спектр возможностей, начиная от создания простых статических графиков до сложных интерактивных 3D-визуализаций. В этом разделе подробно рассматриваются основные особенности, преимущества и недостатки каждой из этих библиотек.

### 1.5.1. Matplotlib

Matplotlib – это одна из наиболее популярных библиотек Python для построения графиков. Она была разработана в 2003 году Джоном Хантером с целью предоставить исследователям инструмент для создания графиков, похожий на возможности MATLAB. Matplotlib считается фундаментальной библиотекой для визуализации данных в Python и лежит в основе многих других библиотек, таких как Seaborn.

Основные возможности Matplotlib

* Поддержка различных типов графиков: линейные графики, гистограммы, диаграммы рассеяния (scatter plot), контурные графики, 3D-графики.

- Высокая кастомизация: Пользователь может настроить практически любой элемент графика: оси, метки, шрифты, цвета, легенды и т.д. Поддерживает использование собственного стиля оформления.

- Интеграция с другими библиотеками: легко используется с Pandas, NumPy и SciPy. Совместима с библиотеками для научных расчетов и обработки данных.

- Масштабируемость: подходит для построения графиков как небольших, так и больших наборов данных.

### 1.5.2. Seaborn

Seaborn — это библиотека Python для визуализации данных, построенная на основе Matplotlib. Она была создана Майклом Уаском (Michael Waskom) и впервые выпущена в 2012 году. Seaborn упрощает процесс создания графиков и добавляет новые возможности для работы со статистическими данными. Благодаря интуитивному API и встроенным стилям, Seaborn является популярным инструментом среди специалистов по анализу данных и исследователей.

Основные возможности Seaborn:

* Статистические графики: графики распределения (гистограммы, KDE-графики). Диаграммы размаха (boxplot) и виолончельные диаграммы (violin plot). Парные графики (pairplot) для исследования взаимосвязей. Тепловые карты (heatmap) для визуализации корреляций и матриц.

- Упрощение работы с данными: прямое взаимодействие с объектами DataFrame из Pandas. Автоматическое управление цветами и стилями для категориальных данных.

- Стилизация графиков: Встроенные стили оформления графиков, такие как darkgrid, whitegrid, dark, white, и ticks. Автоматическая обработка осей и легенд для улучшения читаемости.

### 1.5.3. Plotly

Plotly — это мощная библиотека Python для создания интерактивных графиков, диаграмм и дашбордов. Она была разработана компанией Plotly Inc. и впервые выпущена в 2013 году. Основной акцент библиотеки сделан на интерактивность и возможность интеграции графиков в веб-приложения, что делает её идеальным выбором для бизнес-аналитики, научных исследований и презентаций.

Основные возможности Plotly:

* Интерактивные графики: Поддержка масштабирования, навигации и всплывающих подсказок. Динамическое обновление графиков в реальном времени.
* Широкий набор типов графиков: Линейные графики, гистограммы, scatter plot. Box plot, violin plot, тепловые карты (heatmap). 3D-графики (поверхности, scatter, линии). Карты (географические и контурные).
* Интеграция с веб-приложениями: Экспорт графиков в HTML для публикации в вебе. Интеграция с Dash — фреймворком для создания интерактивных дашбордов.
* Поддержка работы с большими данными: Графики остаются интерактивными даже при больших объемах данных.
* Кастомизация графиков: Гибкая настройка цветов, шрифтов, размеров и других элементов.

# 2. Методология исследования

## 2.1. Постановка задачи

Визуализация данных играет ключевую роль в анализе и интерпретации информации, особенно в таких сложных областях, как геофизика. Целью данной работы является сравнение трех популярных библиотек визуализации данных: Matplotlib, Seaborn и Plotly. Для достижения этой цели необходимо разработать методику исследования, включающую анализ возможностей каждой библиотеки, их производительности и удобства использования.

Основные задачи, решаемые в рамках методологии:

* Определить критерии для сравнения библиотек.
* Разработать набор тестов для визуализации различных типов данных.
* Провести экспериментальное тестирование с использованием реальных геофизических данных.
* Проанализировать полученные результаты и сформулировать выводы.

## 2.2. Описание подхода к тестированию

Для объективного анализа библиотек были выбраны следующие критерии:

* Функциональность

Наличие базовых типов графиков (линейные, гистограммы, scatter plot).

Поддержка сложных графиков (heatmap, boxplot, pairplot, 3D-графики).

Дополнительные возможности, такие как анимации и интерактивность.

* Удобство использования

Простота синтаксиса и удобство API.

Возможность интеграции с другими библиотеками Python (например, Pandas, NumPy, SciPy).

Качество документации и доступность обучающих материалов.

* Визуальная эстетика

Качество графиков.

Возможности настройки стилей и оформления.

* Интерактивность

Возможность взаимодействия с графиками: масштабирование, фильтрация, выделение областей.

Поддержка экспорта интерактивных графиков в веб-формат (HTML).

* Производительность

Время рендеринга графиков.

Эффективность работы с большими наборами данных.

* Простота интеграции

Возможность использования в локальных приложениях (GUI) и веб-приложениях.

Поддержка экспорта графиков в различные форматы (PNG, SVG, HTML).

## 2.3. Набор тестов для визуализации данных

Для объективного сравнения возможностей библиотек разработан набор тестов, включающий следующие задачи:

* Базовые графики

Линейный график временных рядов.

Гистограмма распределения данных.

* 3D-визуализация

Построение 3D-поверхности на основе геофизических данных (данные МЛЭ для построения рельефа морского дна).

* Интеграция в приложения

Отображение графиков в графическом интерфейсе пользователя (GUI).

## 2.4. Данные для исследования

Для тестирования библиотек были использованы реальные данные магнитометрических исследований и данных многолучевого эхолота (МЛЭ).

Файл «forExample.csv» использовался для построения графиков и гистограм (для тестирования использовались, из всего набора, 3 канала (колонки): Fore\_Field (Y), Aft\_Field (Y), Dist (X)).

Файл «Sea\_Depth.csv» использовался для построения 3D поверхности (для тестирования использовались все 3 канала (колонки): (X, Y, Sea\_Depth(Z)).

Исходные данные хранятся в каталоге проекта «Data Example».

## 2.5. Тестирование

Каждая библиотека тестировалась в одинаковых условиях на следующих этапах:

* Подготовка данных

Импорт данных в формате CSV или Excel.

Очистка данных от пропусков и выбросов, приведение к единому формату.

* Создание графиков и гистограмм

Построение одинаковых типов графиков с использованием каждой библиотеки.

Настройка стилей, цветов и параметров графиков.

* Сравнение по критериям

Измерение времени выполнения кода для построения графиков.

Оценка сложности кода (количество строк, читаемость).

Визуальная оценка качества графиков.

* Документирование результатов

Сохранение графиков и метрик производительности.

# ****3. Практическое сравнение Matplotlib, Seaborn и Plotly****

В данной главе приводится практическое сравнение библиотек Matplotlib, Seaborn и Plotly на основе задач визуализации данных. Для тестирования использовались реальные данные, представленные в формате CSV, а также набор тестов, описанный в главе 2. Целью анализа является оценка функциональности, удобства использования, производительности и визуального качества графиков.

## ****3.1. Данные для тестирования****

Для проведения эксперимента использовались два набора данных:

Файл «forExample.csv» использовался для построения графиков и гистограм (для тестирования использовались, из всего набора, 3 канала (колонки): Fore\_Field (Y), Aft\_Field (Y), Dist (X)).

Файл «Sea\_Depth.csv» использовался для построения 3D поверхности (для тестирования использовались все 3 канала (колонки): (X, Y, Sea\_Depth(Z)).

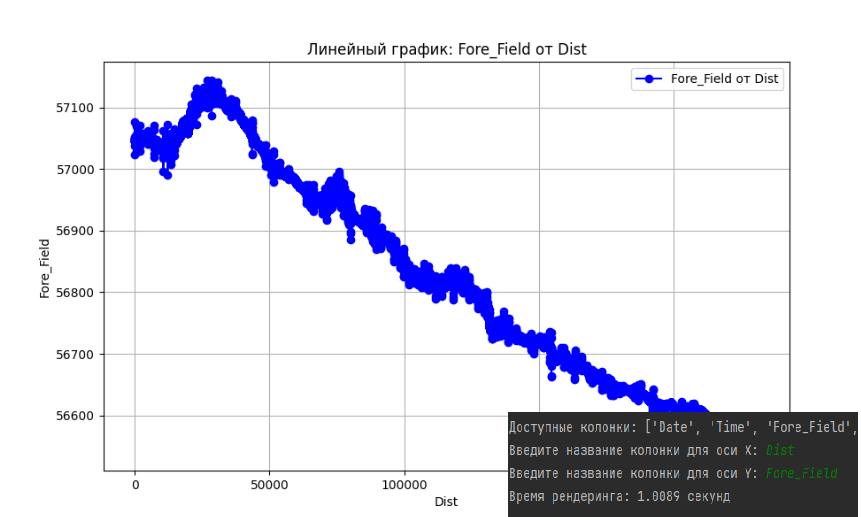
## ****3.2. Построение базовых графиков****

### **3.2.1. Линейный график**

**Matplotlib**: Гибкость настройки позволяет точно контролировать стиль графика, но требуется больше строк кода.

Запуск теста: test\_ Matplotlib.py. Файл данных: forExample.csv.

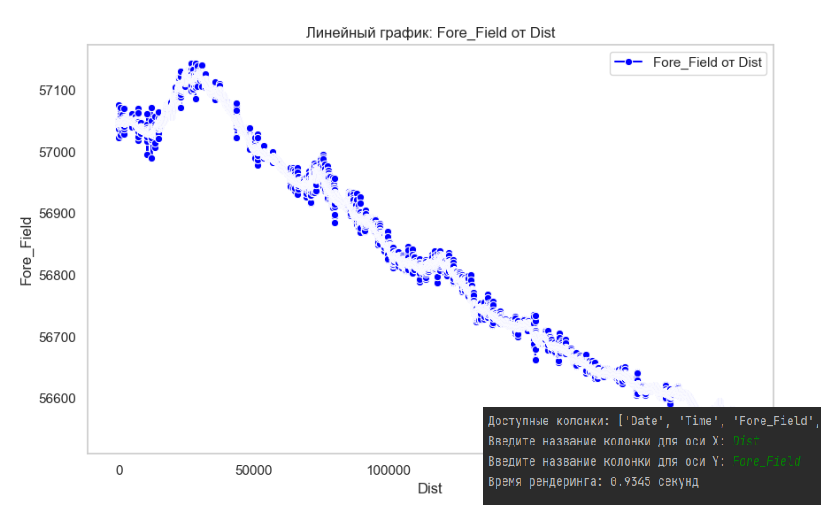
Время рендеринга: 1.0089



**Seaborn**: Простота и автоматическая стилизация делают код лаконичным, но эстетически график смотрится менее привлекательным.

Запуск теста: test\_Seaborn.py. Файл данных: forExample.csv.

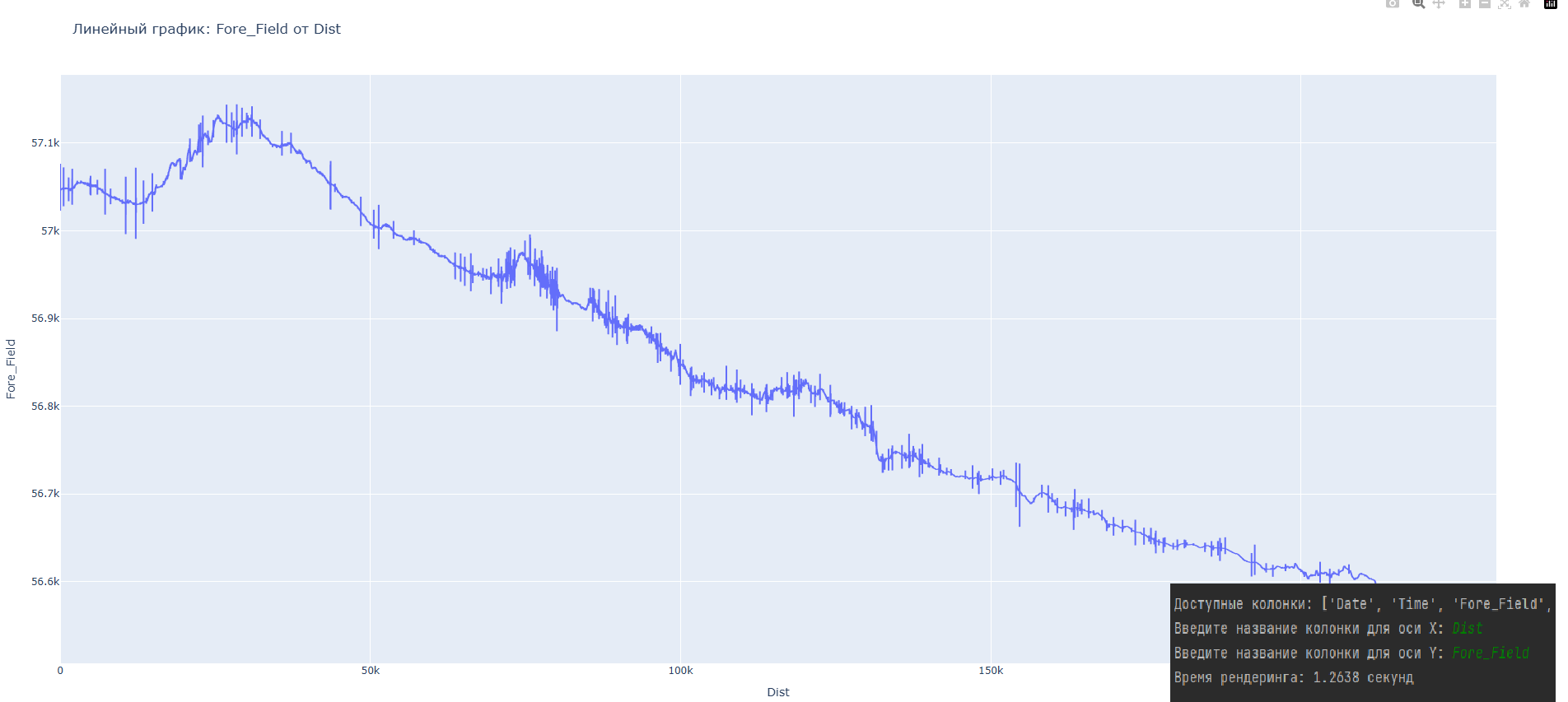
Время рендеринга: 0.9345



**Plotly**: Интерактивность позволяет увеличивать и уменьшать масштаб, что полезно при анализе.

Запуск теста: test\_Plotly.py. Файл данных: forExample.csv.

Время рендеринга: 1.2638

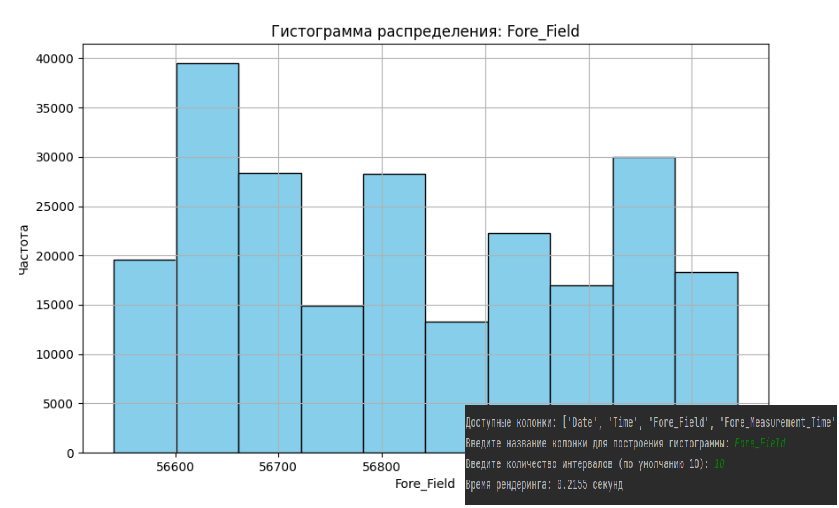


### **3.2.2. Гистограмма распределения данных**

**Matplotlib**: График требует дополнительных усилий для настройки.

Запуск теста: test\_ Matplotlib.py. Файл данных: forExample.csv.

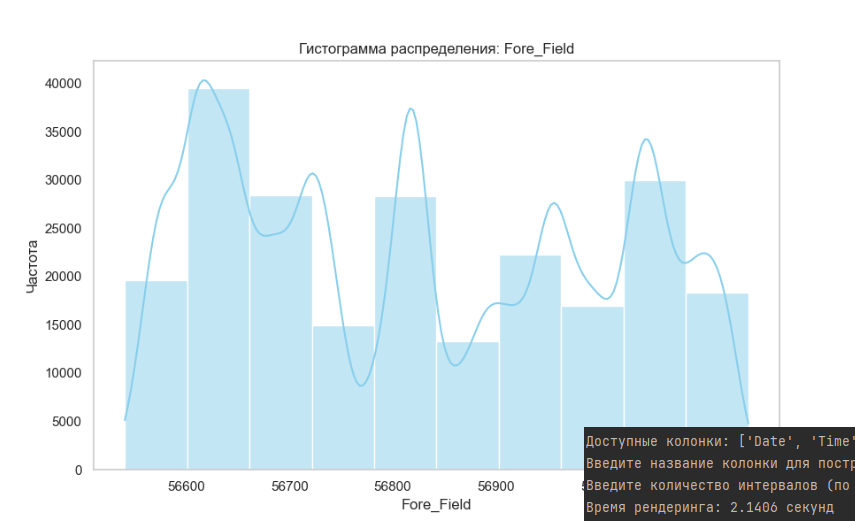
Время рендеринга: 2.2155



**Seaborn**: Встроенная поддержка KDE (гладкая кривая плотности) делает график информативным.

Запуск теста: test\_Seaborn.py. Файл данных: forExample.csv.

Время рендеринга: 2.1406



**Plotly**: Интерактивность позволяет исследовать отдельные столбцы распределения.

Запуск теста: test\_Plotly.py. Файл данных: forExample.csv.

Время рендеринга: 1.2638

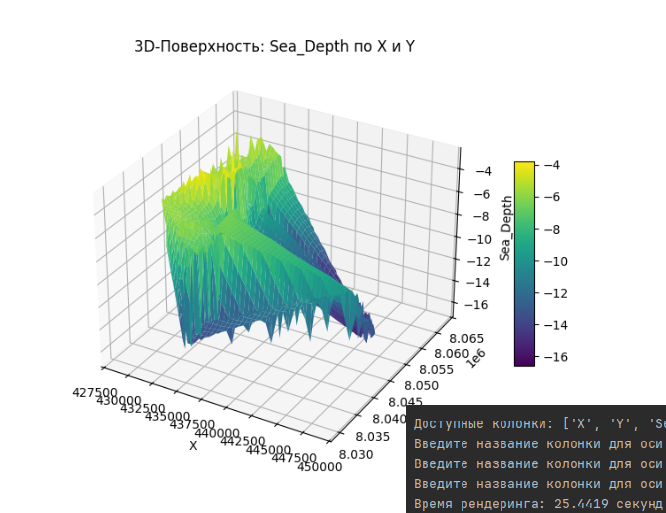


## ****3.3. Построение 3D-поверхности****

**Matplotlib**: Требует больше усилий для настройки 3d-поверхности.

Запуск теста: 3Dtest\_Matplotlib.py. Файл данных: Sea\_Depth.csv.

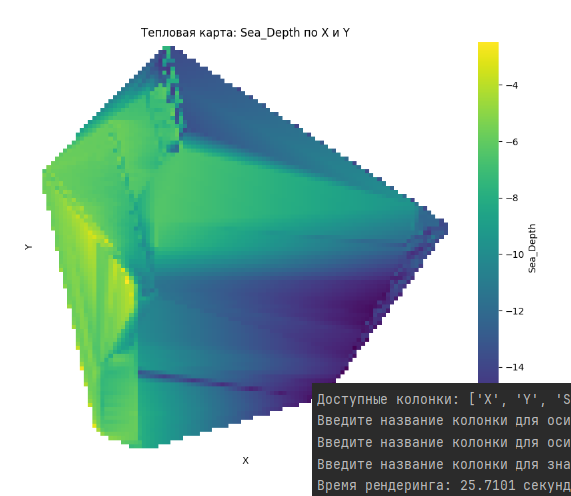
Время рендеринга: 25.4419



**Seaborn**. Библиотека **Seaborn** не поддерживает построение 3D-графиков, так как она сосредоточена на 2D-визуализациях. Однако, в качестве эксперимента по производительности, можно использовать аналог 3D-поверхности в виде тепловой карты (**heatmap**) с использованием Seaborn. Эта визуализация будет отображать изменения значений на двумерной сетке.

Запуск теста: 3Dtest\_Seaborn.py. Файл данных: Sea\_Depth.csv.

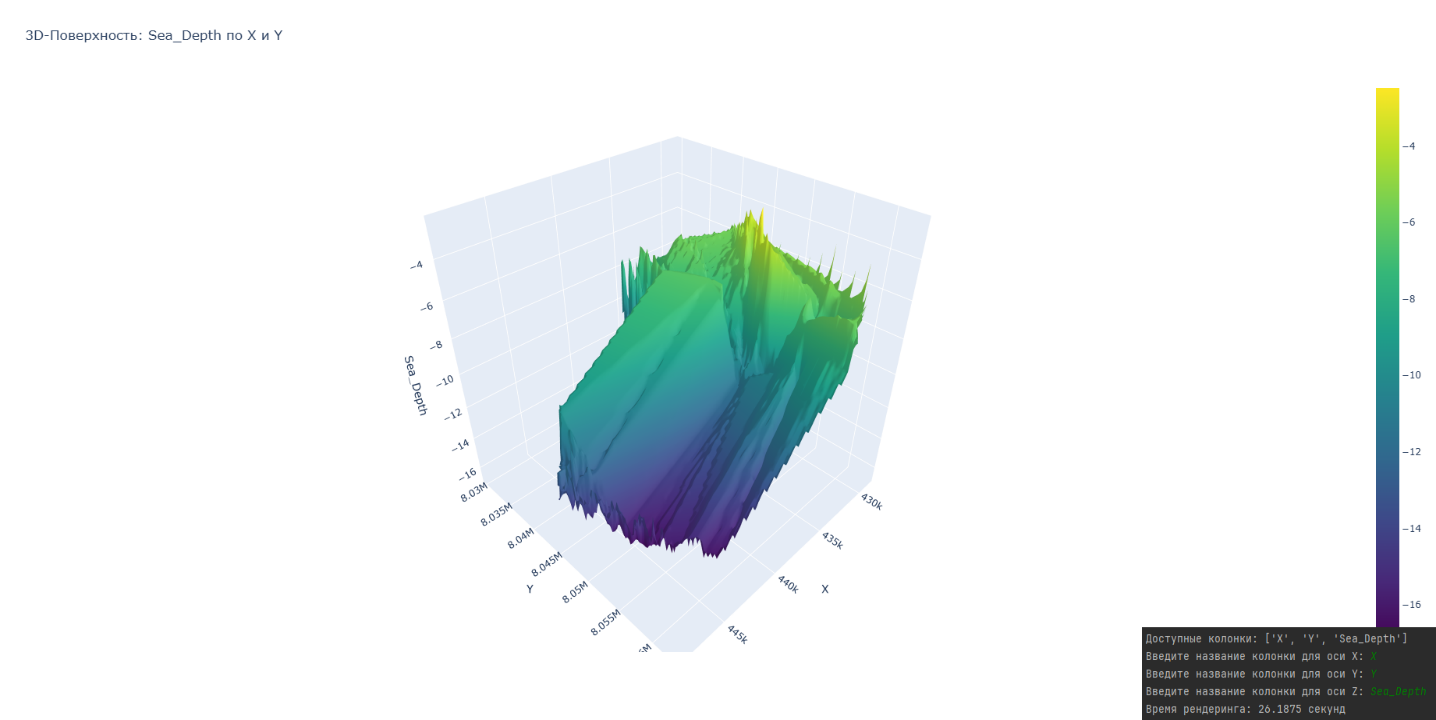
Время рендеринга: 25.7101



**Plotly.** Вне конкуренции по эстетическому виду, возможности интерактивной работы и простоты синтаксиса.

Запуск теста: 3Dtest\_Plotly.py. Файл данных: Sea\_Depth.csv.

Время рендеринга: 26.1875



# ****4. Обсуждение результатов****

## 4.1. Изучение и тестирование библиотек визуализации

В процессе работы над дипломной работой были изучены и протестированы три популярные библиотеки визуализации данных: Matplotlib, Seaborn и Plotly. Каждая из библиотек была оценена по следующим критериям:

Функциональность – поддержка различных типов графиков.

Производительность – время рендеринга графиков при обработке небольших и больших объемов данных.

Эстетичность – внешний вид графиков по умолчанию и возможности настройки.

Результаты тестирования показали, что каждая из библиотек имеет свои преимущества и ограничения:

Matplotlib продемонстрировал высокую гибкость в настройке графиков и стабильную производительность, однако его использование требует большего объема кода для сложных графиков.

Seaborn выделился интуитивным интерфейсом и эстетически приятными графиками "из коробки", что делает его идеальным для быстрых исследований.

Plotly показал свои сильные стороны в интерактивной визуализации, особенно при работе с 3D-графиками, однако его производительность ниже при рендеринге больших объемов данных.

Эти характеристики стали основой для выбора подходящих библиотек при разработке программы.

## 4.2. Реализация программы «GraphPythonMag»

В результате анализа возможностей библиотек была реализована программа с графическим интерфейсом, которая помогает специалистам в обработке и визуализации геофизических данных. Программа была разработана как дополнительный инструмент к основным рабочим программам, используемым для обработки данных. Она предоставляет функции построения различных типов графиков и позволяет визуализировать результаты геофизических исследований.

Для реализации функционала программы были выбраны следующие библиотеки:

Matplotlib – для построения простых графиков, таких как линейные графики и графики временных рядов. Matplotlib предоставляет полный контроль над графиками, что позволяет настраивать каждый элемент визуализации. Это особенно полезно при построении графиков, предназначенных для публикаций или отчетов.

Seaborn – для построения гистограмм. Seaborn упрощает создание статистических графиков и автоматически применяет стили, которые делают гистограммы более наглядными.

Plotly – для построения 3D-поверхностей. Plotly идеально подходит для создания интерактивных 3D-графиков, которые позволяют пользователям исследовать данные, масштабировать и поворачивать график для детального анализа.

## 4.3. Описание работы программы

Программа имеет следующие функции.

Загрузка данных. Пользователь может загрузить файл в формате CSV с геофизическими данными.

Программа автоматически определяет доступные столбцы и предлагает их для выбора при построении графиков.

Построение графиков.

Линейные графики (Matplotlib): позволяют визуализировать магнитного поля в зависимости от дистанции.

Гистограммы (Seaborn). используются для анализа распределения данных, например, частоты встречаемости определенных значений параметров.

3D-поверхности (Plotly): отображают пространственное распределение параметров, таких как глубина моря или плотность пород в зависимости от координат.

Графический интерфейс.

Программа имеет интуитивно понятный интерфейс с кнопками, выпадающими списками и радиокнопками для выбора параметров визуализации.

Поддерживается многоязычный интерфейс, что делает программу удобной для международной аудитории.

Экспорт графиков.

Интерактивные графики Plotly могут быть экспортированы в формате HTML для последующего использования в веб-приложениях или презентациях.

## 4.4. Предложения по улучшению программы

Несмотря на достигнутые результаты, программа имеет потенциал для дальнейшего развития.

Добавление функционала:

Поддержка новых типов графиков, таких как тепловые карты (heatmaps) для анализа корреляций.

Оптимизация производительности:

Реализация механизмов предварительной обработки данных (например, фильтрации и агрегации), чтобы сократить время рендеринга графиков при работе с большими наборами данных.

Использование многопоточности для ускорения выполнения операций.

Интеграция с основными рабочими программами:

Возможность экспорта данных из программы в формате, совместимом с основными геофизическими инструментами.

Интерактивность интерфейса:

Реализация функции drag-and-drop для загрузки файлов.

Включение инструментов для кастомизации графиков через интерфейс (изменение цвета, стиля линий и т. д.).

Публикация графиков в веб-приложениях:

Интеграция с фреймворками, такими как Dash или Streamlit, для создания веб-версии программы, доступной для удаленного использования.

# Заключение

Визуализация данных является неотъемлемой частью анализа, интерпретации и представления информации в современных исследованиях и прикладных задачах. В рамках данной работы была проведена разработка и анализ приложения для визуализации данных с использованием библиотек Python: Matplotlib, Seaborn и Plotly. Приложение создано с целью упрощения работы специалистов, работающих с большими объемами данных, включая геофизические измерения.

Были изучены ключевые аспекты визуализации данных, её роль в различных отраслях и эволюция инструментов визуализации от ранних графических методов до современных интерактивных решений.

Проведен детальный обзор Matplotlib, Seaborn и Plotly, включая их функциональные возможности, преимущества и ограничения.

Было написано 6 тестов для тестирования возможности библиотек. Проведено сравнение библиотек на основе производительности, функциональности, удобства использования и визуальной эстетики. Для каждой библиотеки выявлены области применения и даны рекомендации по их использованию.

Создано графическое приложение, которое позволяет загружать данные, выбирать параметры визуализации и строить графики. Приложение поддерживает работу с несколькими библиотеками, включая построение 3D-графиков с помощью Plotly.

Реализована поддержка многозадачности и многоязычного интерфейса, что делает приложение доступным для широкой аудитории.

Разработанное приложение и результаты исследования имеют прикладное значение. Приложение может быть использовано геофизиками для обработки данных измерений, создания графиков и анализа результатов.

Возможность создания интерактивных графиков упрощает представление данных и их интерпретацию.

В ходе написания работы выявлены направления для дальнейших исследований и улучшений.

Выполненная работа продемонстрировала потенциал библиотек Matplotlib, Seaborn и Plotly для решения задач визуализации данных. Реализованное приложение подтвердило свою практическую значимость, предоставив пользователям инструмент для анализа и представления информации. Предложенные направления для улучшения открывают возможности для дальнейшего развития проекта, делая его универсальным и востребованным в различных областях.

# Список литературы

1. https://matplotlib.org/
2. https://seaborn.pydata.org/
3. https://plotly.com/python/
4. <https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B5%D0%B9%D1%84%D1%8D%D1%80,_%D0%A3%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC>
5. <https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%83%D1%87%D0%BD%D0%B0%D1%8F_%D0%B2%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F#%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5>
6. <https://proglib.io/p/top-7-bibliotek-vizualizacii-dannyh-v-2024-godu-obzor-i-sravnenie-2024-08-15>
7. <https://visme.co/blog/ru/vizualizaciya-dannyh/>