**ДИПЛОМНАЯ РАБОТА**

на тему

«Сравнение различных библиотек для визуализации данных:

Matplotlib, Seaborn и Plotly»

**Автор**

**Горбунов Артем Станиславович**

**Самара. 2024**

**Содержание**

[1 ВВЕДЕНИЕ 4](#_Toc186318826)

[1.1 Обоснование выбора темы 4](#_Toc186318827)

[2 ОПРЕДЕЛЕНИЕ ЦЕЛИ И ЗАДАЧ ИССЛЕДОВАНИЯ 5](#_Toc186318828)

[3 ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ 6](#_Toc186318829)

[4 МЕТОДЫ И ПОДХОДЫ К РАЗРАБОТКЕ 8](#_Toc186318830)

[4.1 Выбор исходных данных для анализа и подбор инструментария 8](#_Toc186318831)

[4.1.1 Выбор исходных данных для анализа 8](#_Toc186318832)

[4.1.2 Подбор инструментария 8](#_Toc186318833)

[4.2 Описание архитектуры проекта 8](#_Toc186318834)

[4.2.1 Структура проекта 8](#_Toc186318835)

[4.2.2 Создание визуализаций 8](#_Toc186318836)

[4.3 Сравнение функциональности и удобства использования 8](#_Toc186318837)

[4.3.1 Анализ функциональности 8](#_Toc186318838)

[4.3.2 Оценка удобства использования 8](#_Toc186318839)

[4.4 Обеспечение документальности и отчетности 8](#_Toc186318840)

[4.4.1 Документация 8](#_Toc186318841)

[4.4.2 Отчет о сравнении 9](#_Toc186318842)

[5 ОБЗОР ПОПУЛЯРНЫХ ИНСТРУМЕНТОВ 10](#_Toc186318843)

[5.1 Matplotlib 10](#_Toc186318844)

[5.2 Seaborn 10](#_Toc186318845)

[5.3 Plotly 11](#_Toc186318846)

[6 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ 12](#_Toc186318847)

[6.1 Планирование и анализ требований 12](#_Toc186318848)

[6.2 Основные требования 12](#_Toc186318849)

[6.3 Технические требования 12](#_Toc186318850)

[7 РАЗРАБОТКА В СООТВЕТСТВИИ С СОЗДАННОЙ ДОКУМЕНТАЦИЕЙ 13](#_Toc186318851)

[7.1 Планирование разработки 13](#_Toc186318852)

[7.2 Разработка 13](#_Toc186318853)

[8 АНАЛИЗ И ИНТЕРПРЕТАЦИЯ РЕЗУЛЬТАТОВ 17](#_Toc186318854)

[8.1 Сравнение библиотек 17](#_Toc186318855)

[8.2 Интерпретация результатов 17](#_Toc186318856)

[8.3 Рекомендации по выбору библиотеки в зависимости от проекта 26](#_Toc186318857)

[8.3.1 Научные исследования и публикации 26](#_Toc186318858)

[8.3.2 Анализ данных и отчетность 26](#_Toc186318859)

[8.3.3 Веб-приложения и интерактивные дашборды 26](#_Toc186318860)

[8.3.4 Презентации и визуализация данных для бизнеса 27](#_Toc186318861)

[8.3.5 Образовательные проекты и курсовые работы 27](#_Toc186318862)

[8.3.6 Проекты с акцентом на визуализацию больших данных 27](#_Toc186318863)

[9 ЗАКЛЮЧЕНИЕ 28](#_Toc186318864)

[9.1 Обзор выполненной работы 28](#_Toc186318865)

[9.2 Дальнейшие планы 28](#_Toc186318866)

[10 ПРИЛОЖЕНИЕ 29](#_Toc186318867)

# ВВЕДЕНИЕ

## Обоснование выбора темы

Современный мир – это мир технологий и информации. С каждым годом разрабатывается все больше инструментов, как аппаратных, так и программных, которые позволяют нам сокращать время производства, повышать надежность или же просто помогать нам в ежедневных задачах.

Объем и сложность данных растут, что делает их анализ, фильтрацию, систематизацию, верификацию и визуализацию непростой задачей. В целом, эффективная визуализация данных является важнейшим аспектом в различных областях науки, технологий и бизнеса, и помогает не только лучше понять информацию, но и выявить скрытые закономерности, тенденции и аномалии. В связи с этим, выбор соответствующего инструмента для визуализации данных становится критически важным для аналитиков, исследователей и бизнесменов.

С точки зрения практического применения визуализация данных позволит эффективно представлять сложные объемы информации в наглядной и понятной форме, что облегчает анализ и восприятие данных. Качественная визуализация может сыграть ключевую роль в анализе научных и технических исследований, анализе тенденций и трендов на рынке, а также анализе статистики коммерческой и иных деятельностей.

Изучение темы визуализации данных позволит мне в дальнейшем использовать соответствующие инструменты для оценки продаж оборудования различных модификаций, а также оценки эффективности и объема продаж в различных условиях. Также визуализация данных позволит мне в дальнейшем отображать данные о расходах и погрешности прибора при подборе оборудования.

# ОПРЕДЕЛЕНИЕ ЦЕЛИ И ЗАДАЧ ИССЛЕДОВАНИЯ

Основной целью данной дипломной работы является сравнение трех библиотек для визуализации данных с точки зрения функциональности, производительности, простоты использования и качества визуализации. В ходе исследования будут рассмотрены основные методы, принципы и подходы каждой библиотеки, приведены примеры их применения, а также проведен анализ случаев, в которых использование одной библиотеки может быть предпочтительнее другой.

# ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

В ходе исследования и разработки мы будем оперировать понятиями и определениями, указанными далее:

* ***Визуализация данных*** – процесс представления данных в графической или визуальной форме для облегчения их анализа и интерпретации;
* ***График*** – графическое представление данных, которое может принимать различные формы: линейные графики, столбчатые диаграммы, круговые диаграммы и т.п.;
* ***Диаграмма*** - графическое представление данных, которое показывает взаимосвязи между переменными, например, диаграмма рассеяния или линейная диаграмма;
* ***Легенда*** - элемент графика, который объясняет, что означают различные цвета, символы или линии, используемые для представления данных;
* ***Интерактивность*** – возможность взаимодействия пользователя с визуализацией, что позволяет динамически изменять данные или параметры графика;
* ***Статическая*** ***визуализация*** – графическое представление данных, которое не изменяется в ответ на действия пользователя; обычно это изображения, созданные заранее;
* ***Цветовая*** ***палитра*** – набор цветов, используемых для отображения данных в визуализации, что может помочь в восприятии информации и выделении ключевых элементов;
* ***Подборка*** ***данных*** – процесс выбора и фильтрации данных, которые будут использованы для визуализации, что может включать агрегацию, группировку и трансформацию данных;
* ***Метки осей*** – текстовые обозначения, которые помогают идентифицировать, что представляют оси графика, включая единицы измерения и категории;
* ***Оси*** - линии на графике, которые представляют собой шкалы для измерения данных; обычно есть две оси: горизонтальная (X) и вертикальная (Y);
* ***Тренд*** - общее направление, в котором движутся данные, что может быть выявлено через линейные графики или другие визуализации;
* ***График плотности*** - визуализация, показывающая распределение данных по непрерывной переменной, часто используется для выявления аномалий и тенденций;
* ***Аномалия*** – отклонение от ожидаемого поведения в данных, которое может быть выявлено с помощью визуализации и требует дальнейшего анализа;
* ***Субплоты*** - отдельные графики, размещенные на одной фигуре, что позволяет сравнивать несколько наборов данных или визуализаций одновременно;
* ***Сетчатый график (Facet Plot)*** - способ разделения данных на несколько графиков по категориям, что позволяет сравнивать различные группы в одном представлении
* ***Уровень абстракции*** - степень сложности интерфейса библиотеки, которая определяет, насколько легко пользователю создавать визуализации, начиная от низкоуровневых настроек до высокоуровневых команд;
* ***Сериализация*** - процесс преобразования структуры данных или объекта в формат, который может быть легко сохранен или передан и затем восстановлен;
* ***Агрегация*** - процесс объединения данных в группы для упрощения анализа, часто включает в себя подсчет, суммирование или усреднение значений;
* ***Кросс-табуляция*** - метод представления данных, который показывает взаимосвязь между двумя или более переменными в виде таблицы;
* ***Фильтрация данных*** - процесс выбора подмножества данных на основе определенных критериев, что позволяет сосредоточиться на конкретных аспектах информации.

# МЕТОДЫ И ПОДХОДЫ К РАЗРАБОТКЕ

## Выбор исходных данных для анализа и подбор инструментария

### Выбор исходных данных для анализа

Начнем с определения задач, которые необходимо решить с помощью визуализации данных, и подберем соответствующие наборы данных. Это могут быть открытые датасеты из различных областей, например, финансовые данные, данные о здоровье, данные о климате и т.д.

### Подбор инструментария

На основе выбранных данных определим, какие библиотеки визуализации будут наиболее подходящими.

## Описание архитектуры проекта

### Структура проекта

Определение структуры проекта, включая файлы с исполняющим кодом, кодом взаимодействия с базами данных, базы данных.

### Создание визуализаций

Разработка скриптов для создания различных типов графиков с использованием каждой библиотеки.

## Сравнение функциональности и удобства использования

### Анализ функциональности

По каждой из задач проводится анализ функциональности, глубина настройки, степень соответствия полученного результата к ожидаемому результату.

### Оценка удобства использования

По каждой из задач проводится анализ объема кода, удобство его редактирования и дополнения, и читаемость результатов.

## Обеспечение документальности и отчетности

### Документация

Создать подробную документацию, в которой будут описаны методы визуализации, примеры кода и интерпретация полученных графиков.

### Отчет о сравнении

Подготовка отчета, в котором будет представлено сравнение функциональности и удобства использования Matplotlib, Seaborn и Plotly, а также рекомендации по выбору библиотеки в зависимости от конкретных задач визуализации данных.

# ОБЗОР ПОПУЛЯРНЫХ ИНСТРУМЕНТОВ

## Matplotlib

Matplotlib основана на концепции объектно-ориентированного программирования, что позволяет пользователям создавать графики, используя объекты и методы. Основные принципы Matplotlib включают объектно-ориентированный подход, систему координат и кастомизацию.

В Matplotlib графики создаются как объекты, обладающие различными атрибутами, что дает возможность к кастомизации, а система координат, используемая для размещения графиков на плоскости, позволяет создавать сложные визуализации с подграфиками.

Библиотека содержит в себе широкий набор инструментов для создания статических, анимационных и интерактивных графиков.

Типы графиков, реализованных в библиотеке:

* Линейные
* Диаграммы рассеяния
* Гистограммы
* Круговые диаграммы
* Тепловые карты
* 3D графики

Matplotlib хорошо интегрируется с другими библиотеками Python, такими как:

* NumPy: для работы с массивами чисел и выполнения математических операций;
* Pandas: для анализа данных и работы с табличными данными;
* Seaborn: для более сложной визуализации и статических графиков, построенных на основе Matplotlib.

## Seaborn

Данная библиотека для визуализации построена на основе Matplotlib и представляет из себя высокоуровневый интерфейс. Основные цели библиотеки Seaborn – упрощение создания сложных визуализаций и улучшение эстетического оформления графиков.

Seaborn основана на некоторых важных принципах:

* Статистическая визуализация: библиотека ориентирована на визуализацию статистических данных, что позволяет пользователям легко создавать графики;
* Интуитивно понятный API: библиотека использует высокоуровневые функции, которые позволяют пользователям создавать сложные графики с минимальными усилиями.
* Эстетика: библиотека автоматически применяет стиль и цветовые палитры, что делает графики более привлекательными, однако позволяет настроить стиль и цветовую схему самостоятельно;
* Интеграция с Pandas: библиотека хорошо работает с DataFrame из библиотеки Pandas, что упрощает процесс визуализации данных, хранящихся таблиной форме.

Seaborn хорошо интегрируется с другими библиотеками Python, такими как:

* Matplotlib: построен на указанной библиотеке и может использовать его функции для более детальной настройки;
* Pandas: для анализа данных и работы с табличными данными;
* NumPy: для работы с массивами чисел и выполнения математических операций.

## Plotly

Библиотека Plotly – это библиотека для интерактивной визуализации данных, которая позволяет создавать графики и дашборды с высокой степенью интерактивности. Она поддерживает множество типов графиков и хорошо интегрируется с различными языками программирования, такими как Python, R и JavaScript.

Plotly ориентирован на создание интерактивных визуализаций, что позволяет пользователям взаимодействовать с графиками. Основные принципы работы библиотеки включают:

* Интерактивность: графики, созданные с помощью Plotly, позволяют пользователям взаимодействовать с данными, например, масштабировать, перемещать и наводить курсор для получения дополнительной информации;
* Удобство использования: библиотека предлагает интуитивно понятный API, который позволяет создавать сложные графики с минимальными усилиями;
* Кроссплатформенность: библиотека поддерживает различные языки программирования и может использоваться в веб-приложениях, что делает его универсальным инструментом для визуализации данных.

Plotly хорошо интегрируется с другими библиотеками Python, такими как:

* Pandas: библиотека хорошо работает с DataFrame из библиотеки Pandas, что упрощает процесс визуализации данных, хранящихся таблиной форме;
* Dash: предлагает фреймворк Dash для создания веб-приложений с интерактивными графиками, что позволяет пользователям разрабатывать собственные дашборды и приложения для визуализации данных;
* Jupyter Notebook: библиотека легко интегрируется с Jupyter Notebook, позволяя создавать интерактивные графики прямо в своих интерактивных блокнотах.

# ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

## Планирование и анализ требований

Для начала необходимо выбрать инструменты, представленные в описанных библиотеках, для реализации цели сравнения возможностей визуализации данных.

Разработка схемы архитектуры, включающей пользовательский интерфейс, бэкенд для обработки данных и интеграцию с библиотеками визуализации (Matplotlib, Seaborn, Plotly).

Поиск датасета для использования в качестве исходных данных для построения и анализа графиков.

Создание базовой версии приложения с минимально необходимым функционалом, позволяющим загружать данные и визуализировать их с использованием выбранных библиотек.

Подключение и настройка Matplotlib, Seaborn и Plotly для создания графиков на основе загруженных данных.

Разработка диалоговых окон для загрузки файлов, выбора фильтрации и типа графика.

Реализация логики обработки событий, таких как нажатия кнопок и выбор параметров, для динамического обновления графиков.

## Основные требования

Пользователь должен иметь возможность загружать данные в формате CSV или Excel для анализа.

Приложение должно позволять пользователю выбирать между Matplotlib, Seaborn и Plotly для создания графиков.

Пользователь должен иметь возможность настраивать фильтрацию и тип графика.

Приложение должно отображать созданные графики в интерактивном режиме и предоставлять возможность их экспорта в PNG или JPEG.

## Технические требования

Фронтенд: Использовать библиотеки для создания графического интерфейса, такие как Tkinter, PyQt или wxPython.

Бэкенд: Python для обработки данных и логики приложения.

Интеграция с библиотеками визуализации: Подключение Matplotlib, Seaborn и Plotly для обработки загруженных данных и генерации графиков на основе пользовательских настроек.

Экспорт графиков: Реализация функционала для экспорта графиков в различные форматы (PNG, PDF, HTML).

# РАЗРАБОТКА В СООТВЕТСТВИИ С СОЗДАННОЙ ДОКУМЕНТАЦИЕЙ

## Планирование разработки

Разработка была разделена на несколько этапов: поиск датасета для построения графиков, проектирование интерфейса, реализация логики обработки данных, создание минимального функционального кода для вывода графиков, дополнение функционального кода для отображения графиков до визуально понятного состояния.

## Разработка

Общая структура приложения изображена на рисунке 1

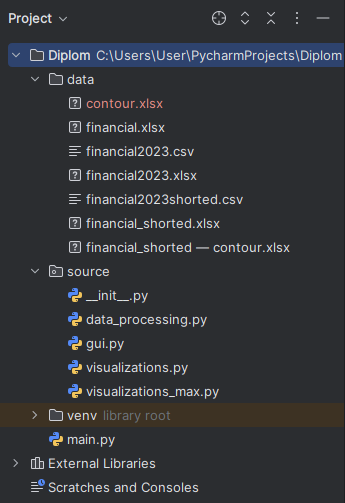


Рисунок 1 – Структура проекта

Фронтэнд разработка.

Был создан пользовательский интерфейс с использованием PyQt5 в файле gui, который содерит в себе несколько функциональных частей.

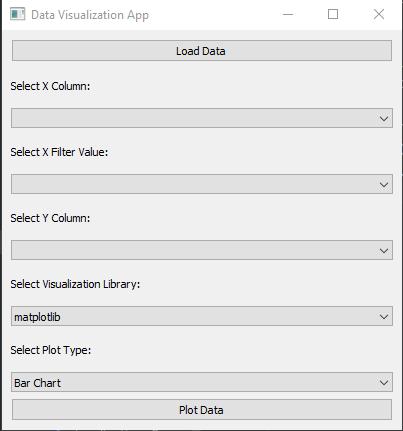


Рисунок 2 – Пользовательский интерфейс

На рисунке 2 изображен пользовательский интерфейс, на котором выполнены некоторые кнопки, запускающие определенные процесса. Кнопка «Load Data» вызывает проводник с целью выбора файла. Во избежание загрузки неподдерживаемого формата выбор ограничен типами файлов «.csv» и «.xlsx». Выбор файла отображен на рисунке 3.

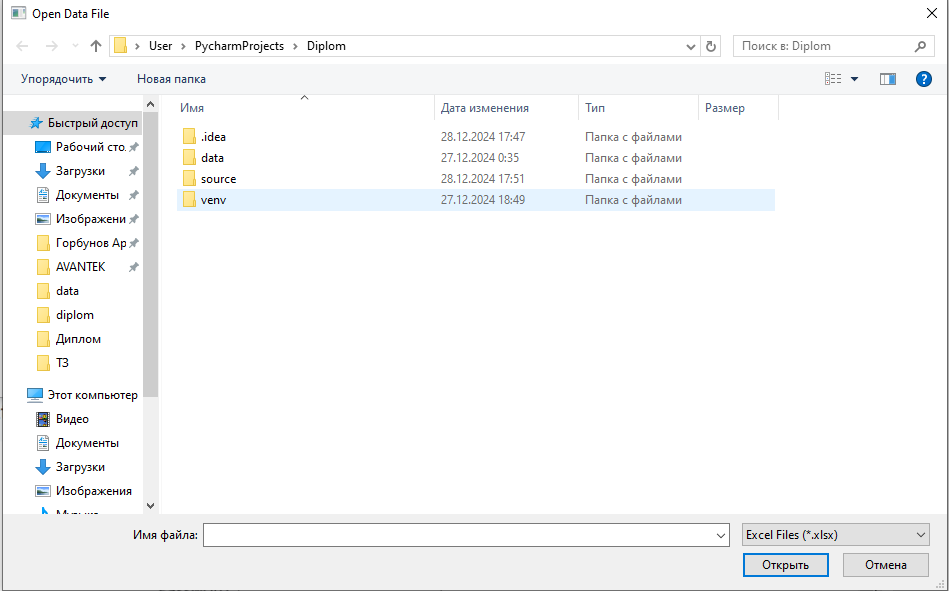


Рисунок 3 – Выбор файла для загрузки

После выбора файла производится его обработка и передача данных в поля Select X column, Select X Filter value, Select Y column, Select Visualization Library, Select Plot Type и Select Z column. Select Z column возникает только при выборе Plot Type равного Contour, так как этот тип графика требует 3 осей. Список выводится из данных

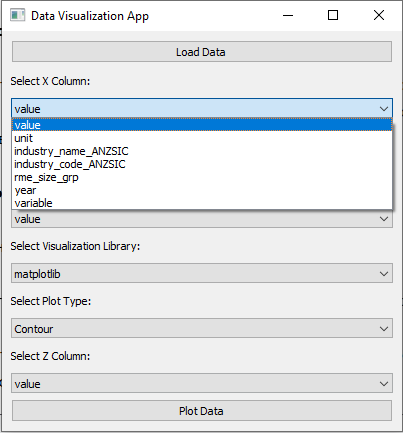


Рисунок 4 – Пользовательский интерфейс после загрузки данных

После выбора всех полей нажатие кнопки Plot Data происходит формирование графика с учетом выбора. На рисунке 5 график сформированный без фильтрации по Х.

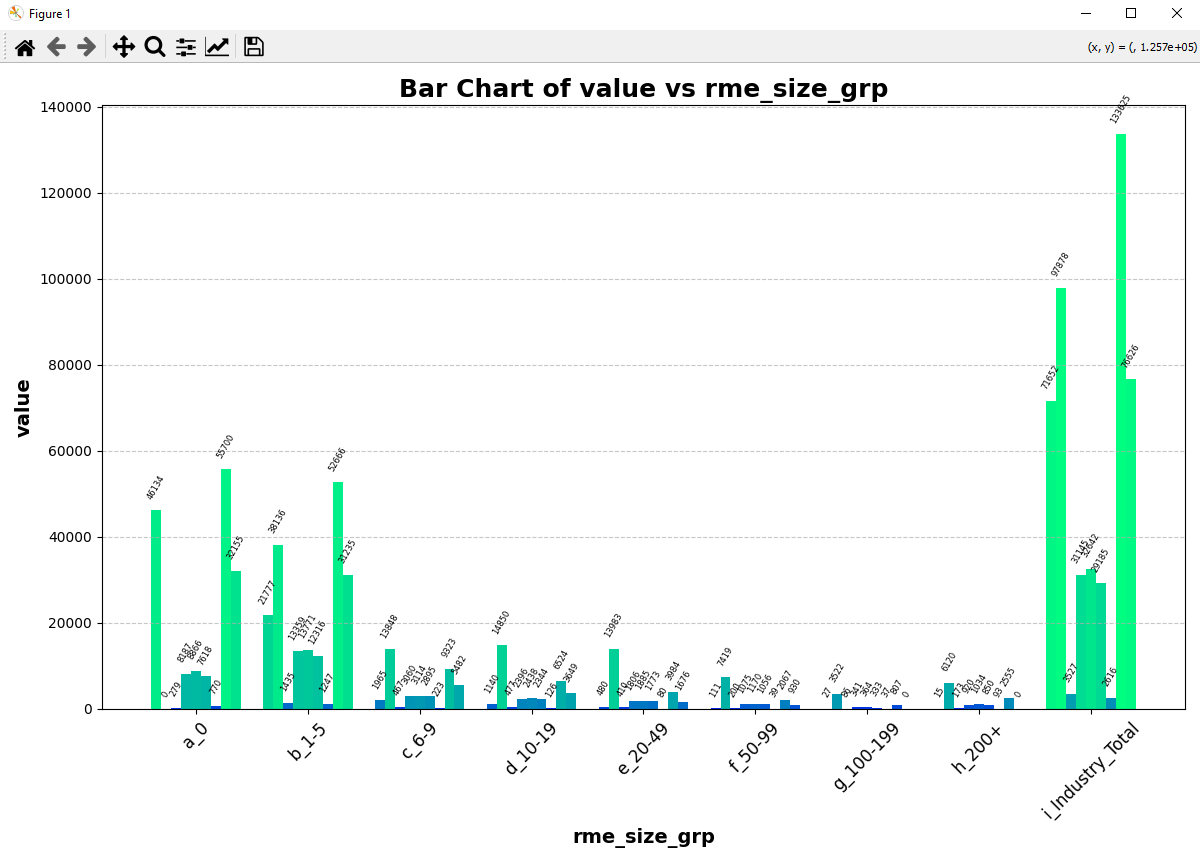


Рисунок 5 – График без фильтрации

На рисунке 6 график сформированный с фильтрацией по Х.

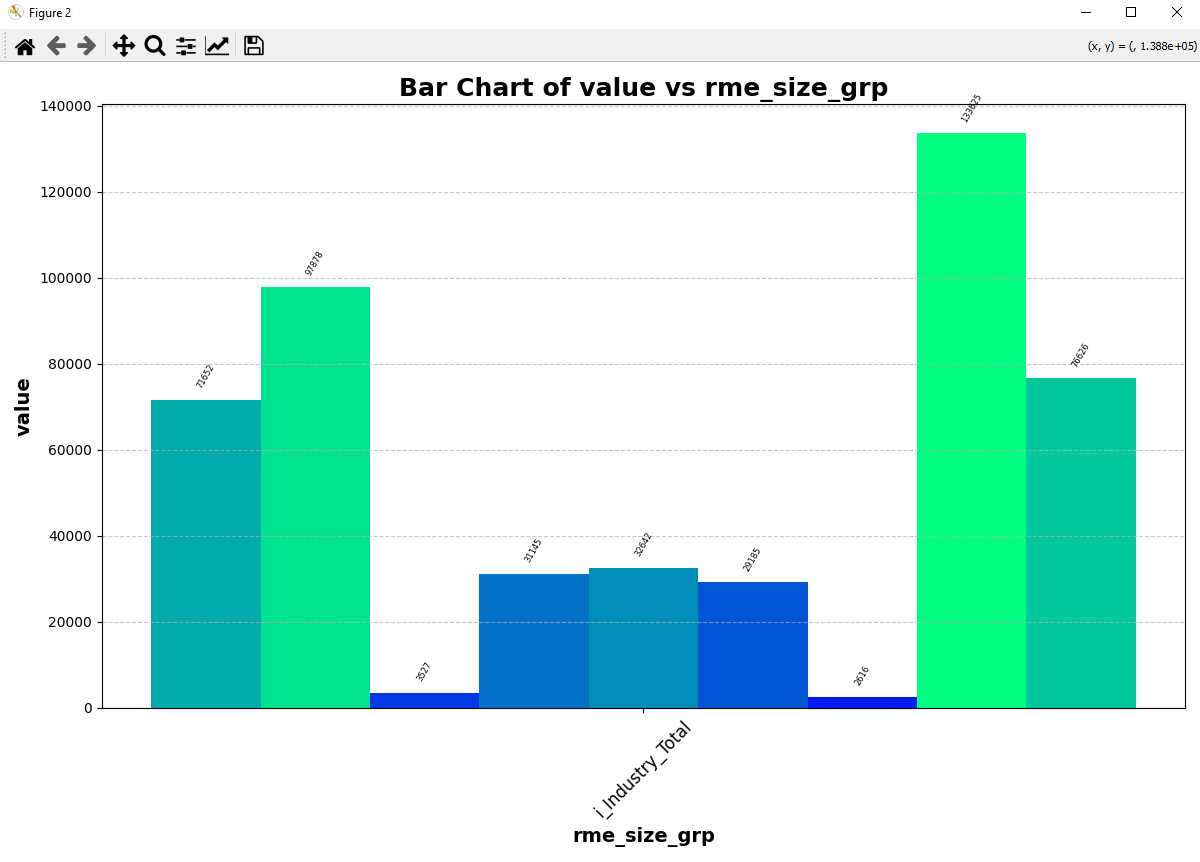


Рисунок 6 – График с фильтрацией

Бэкенд разработка

В файле data\_processing созданы методы обработки данных для их использования при построении графиков. Использованы функции чтения .csv .xlsx форматов из библиотеки Pandas. Написана логика обработки отдельной колонки или всего датасета с переводом данных в числовые значения.

В файле visualizations описаны функции, которые позволяют построить графики с применением различных библиотек, а точнее функций из данных библиотек, однако, код позволяет лишь построить график по полученным данным и не содержит настроек.

В файле visualizations\_max описаны функции, которые позволяют построить графики с применением различных библиотек, а точнее функций из данных библиотек, и выдают понятные, визуально хорошо воспринимаемые графики даже с учетом сложности исходных данных.

В папке data находятся исходные данные и дообработанные файлы для построения графиков.

Для построения графиков bar, line, histogram используются financial\_shorted. В Х выбирается rme\_size\_grp, в Y - value

Для построения графиков contour используется файл financial\_shorted – contour. В Х выбирается x\_value, в Y – y\_value, в Z – height. Допускается иной выбор данных, но в таком случае график будет отличаться от моего.

# АНАЛИЗ И ИНТЕРПРЕТАЦИЯ РЕЗУЛЬТАТОВ

## Сравнение библиотек

Для минимального построения достаточно просто использовать данные, которые мы считали и использовать функции построения нужного графика из библиотеки, а после вызвать функцию show() от используемой библиотеки для вывода графика на экран.

Минимальный код для построения графика в каждой из библиотек состоит из 2 строчек для bar, line и histogram. Построение contour в matplotlib займет 3 строчки кода, в seaborn – 3-4 строчки, в plotly – 7.

Рассмотрим разницу в подходе и организации кода по каждой из библиотек на примере наиболее настроенных и визуально информативных графиков.

Matplotlib предлагает высокую гибкость и возможность настройки, но требует больше кода для создания графиков. Требует ручной обработки данных. Цветовая палитра применяется через Pandas и создание цветовой карты (colormap). Благодаря детальной настройке легче получать требуемый результат.

Seaborn упрощает процесс визуализации данных благодаря удобному синтаксису и набору стилей по умолчанию. Применение встроенной цветовой схемы происходит по указанию значения для параметра «palette». Библиотека отлично работает с Pandas DataFrame. При работе со сложными данными и необходимостью группировки данных код осложнен и менее гибок к настройке.

Plotly позволяет создавать интерактивные графики. Библиотека содержит много цветовых схем, а добавление аннотаций и подписей осей происходит очень удобно через параметры внутри функций.

## Интерпретация результатов

Для начала рассмотрим графики полученные минимальным кодом.

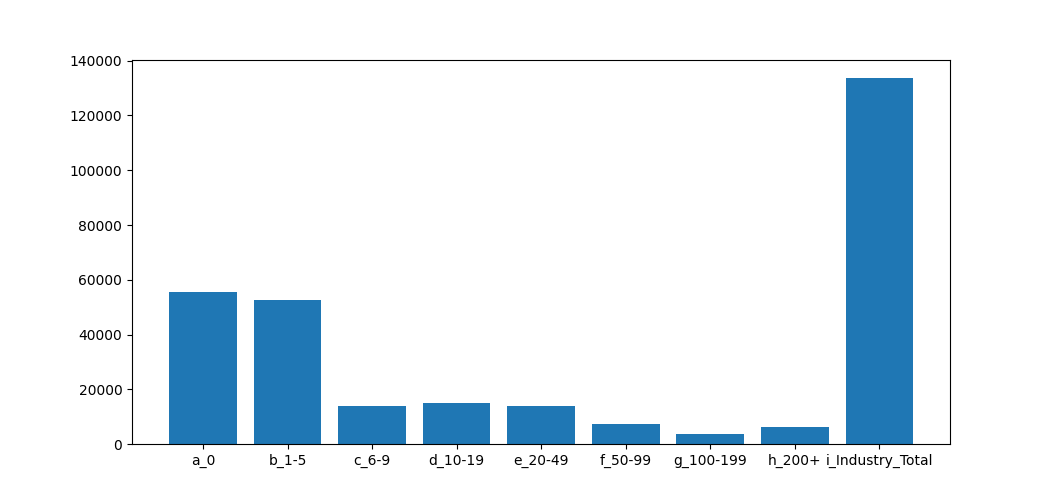


Рисунок 7 – Matplotlib Bar

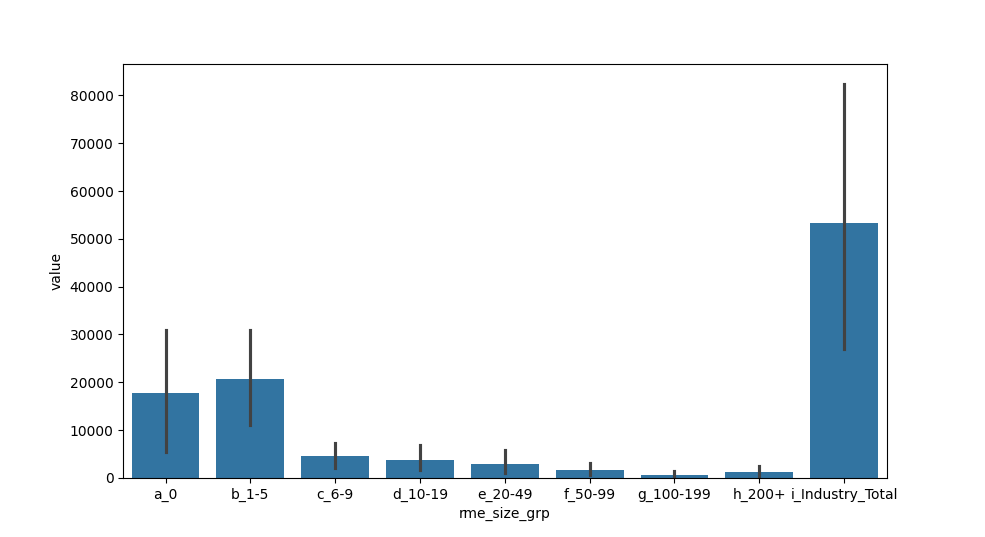


Рисунок 8 – Seaborn Bar

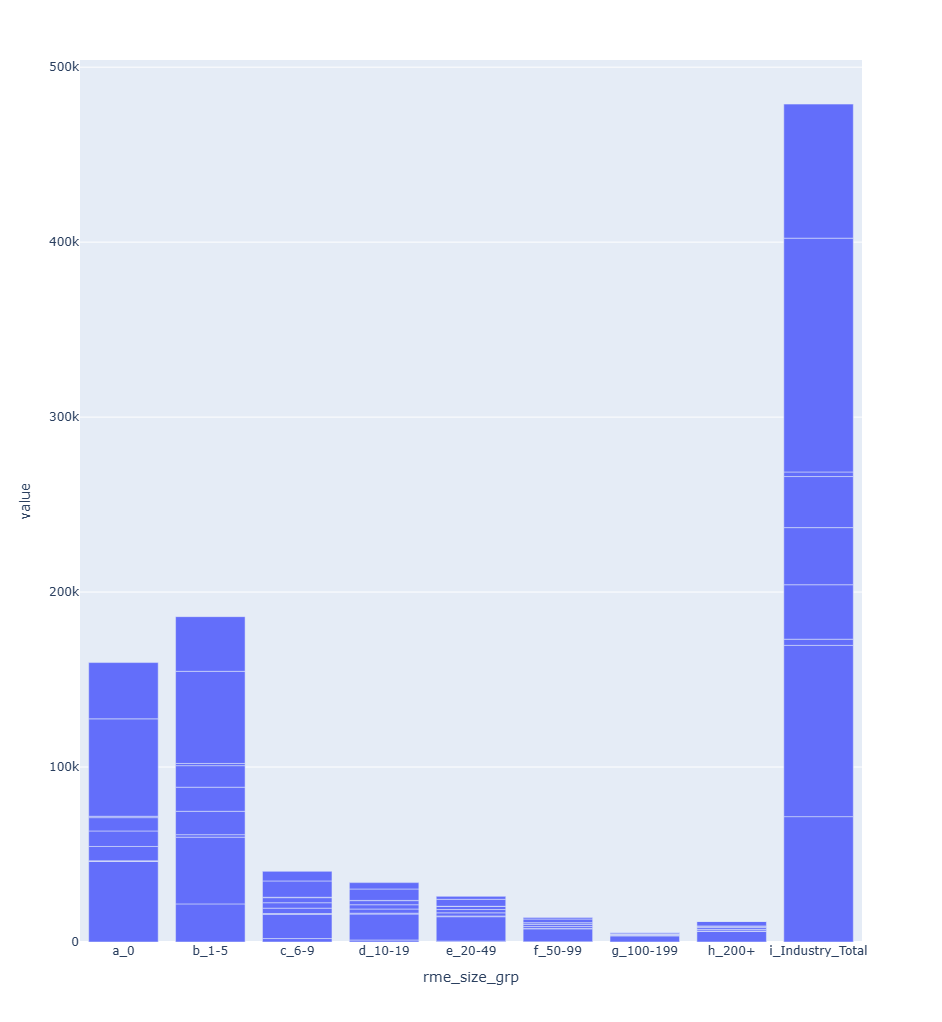


Рисунок 9 – Plotly Bar

Как видно на рисунках 7, 8 и 9 библиотека matplotlib практически не дает понимания о данных и в ней не указаны заголовки. В библиотеке plotly видно, что значениям по оси Х соответствуют множественные данные, а также присутствуют заголовки и минимальная сетка. В библиотеке seaborn не только присутствуют заголовки, но и автоматически строится так называемая погрешность, которая показывает минимальное и максимальное значение внутри одного Х, а также сам столбец является аппроксимацией значений.

Рассмотрим графики трехмерных данных.

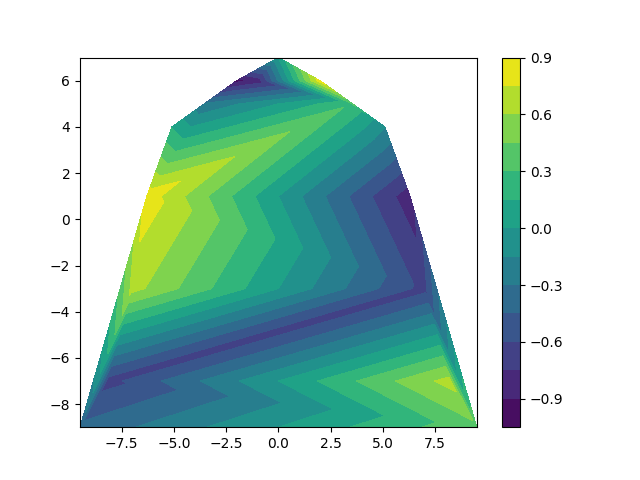


Рисунок 10 – Matplotlib Contour

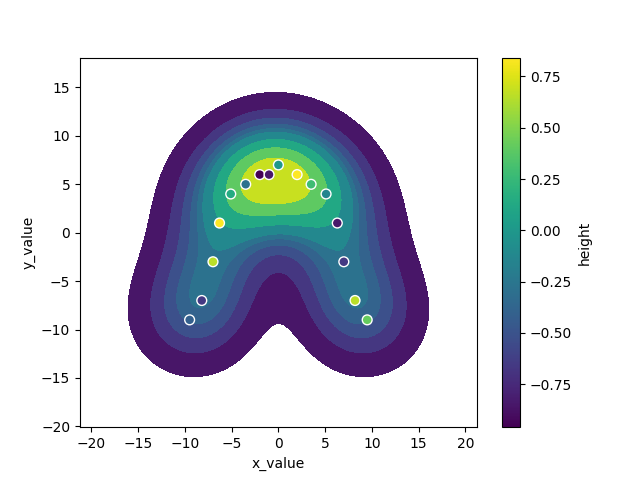


Рисунок 11 – Seaborn Contour

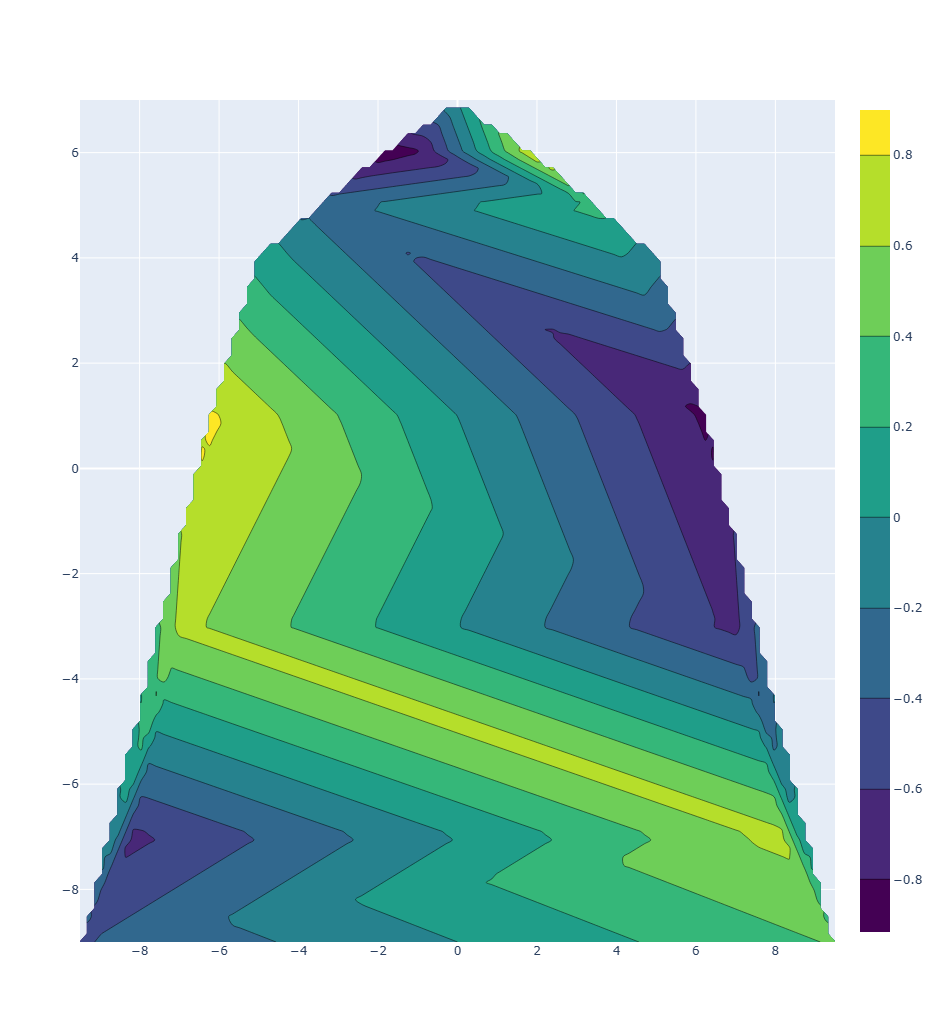


Рисунок 12 – Plotly Contour

Как видно на рисунках 10, 11 и 12 seaborn вновь дает более визуально понятное представление, а также автоматически расставляет точки на графике, по которым понятно какая высота в соответствии с цветом.

Также дополню рисунками по графикам Line. Ситуация аналогичная. Matplotlib и Plotly не поняли как обрабатывать данные. Seaborn же построил график с погрешностями.

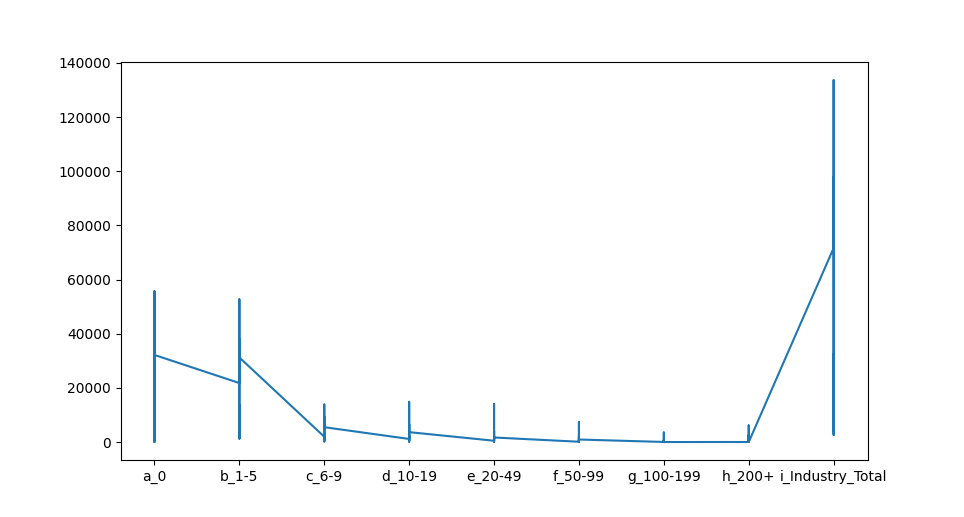


Рисунок 13 – Matplotlib Line

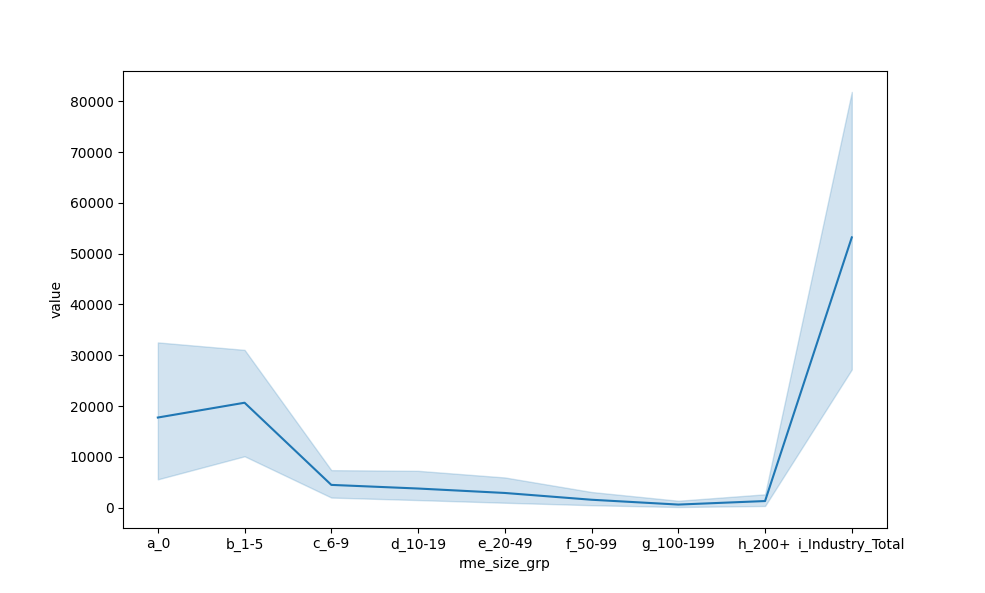


Рисунок 14 – Seaborn Line

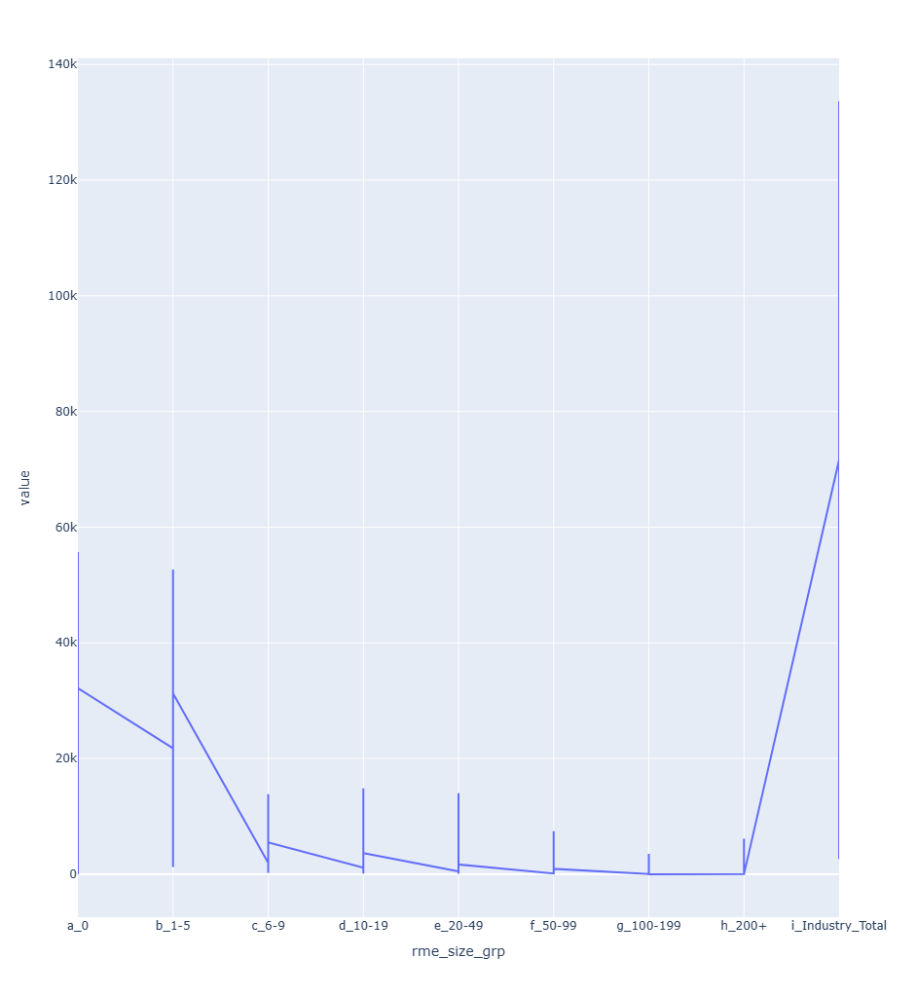


Рисунок 15 – Plotly Line

И графики по histogram. На которых аналогичная ситуация. Seaborn дополнительно сам построил тренд.

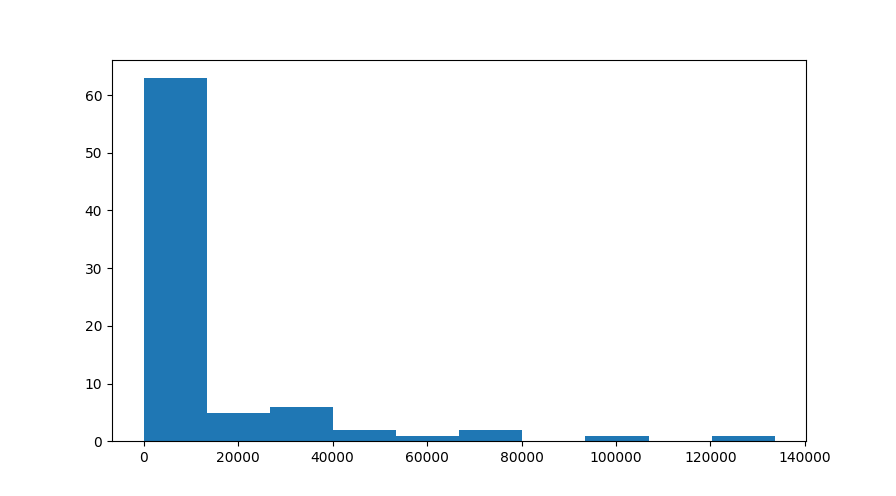


Рисунок 16 – Matplotlib Histogram

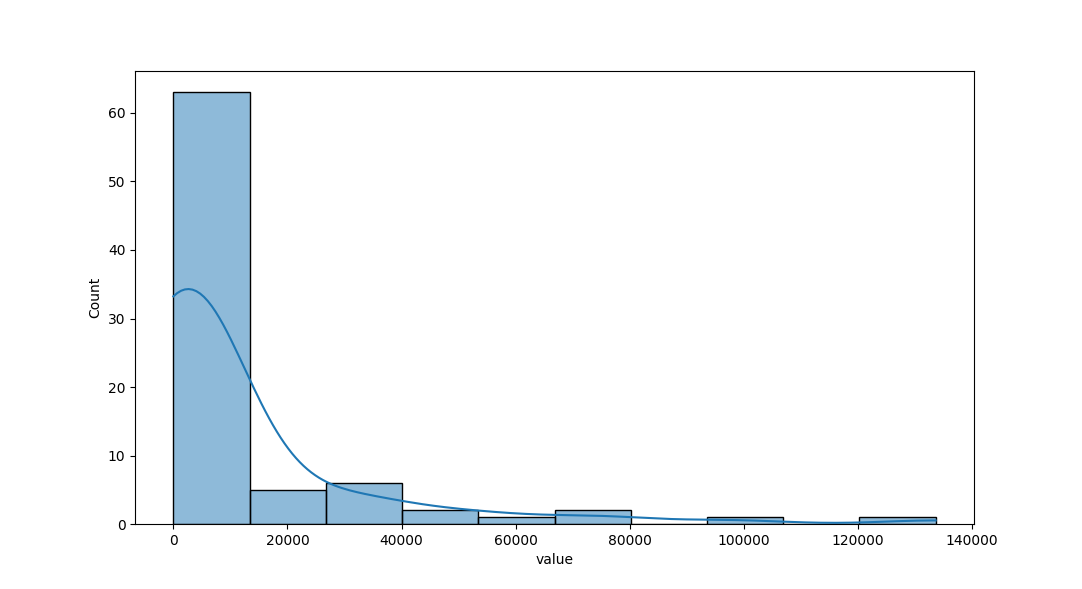


Рисунок 17 – Seaborn Histogram

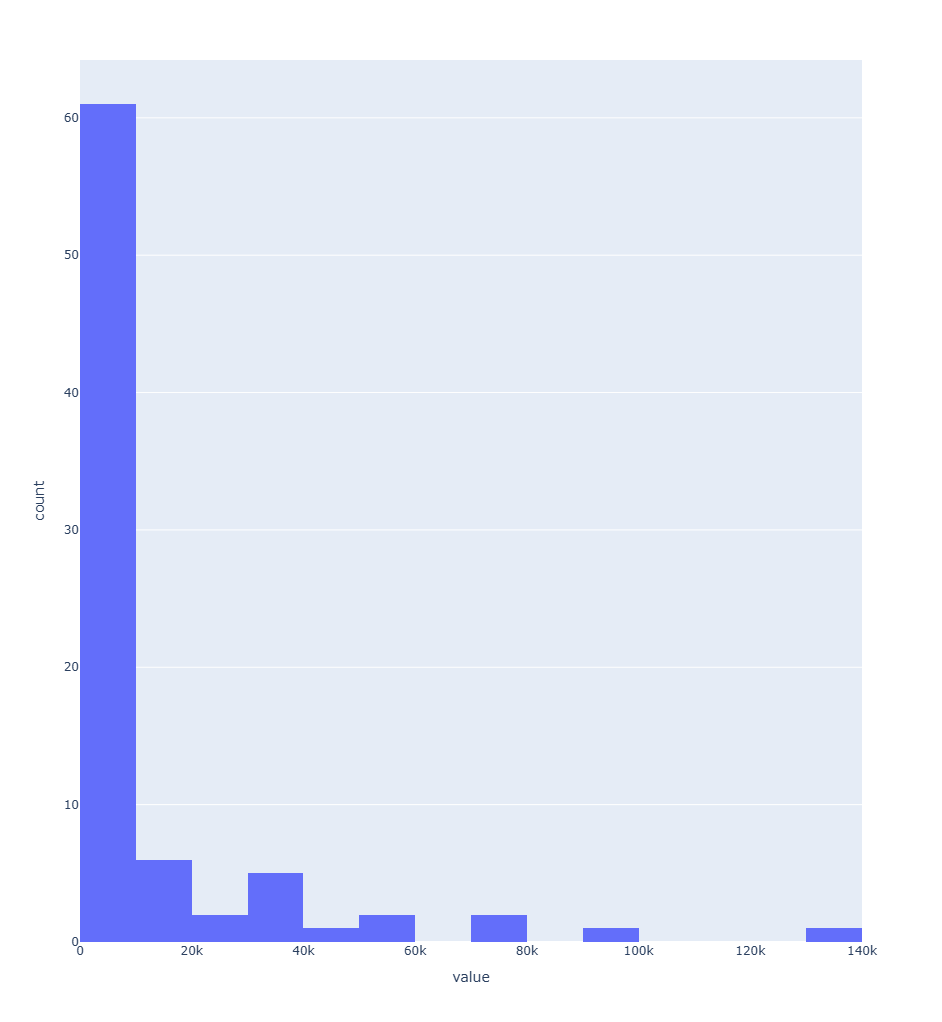


Рисунок 18 – Plotly Histogram

Далее мы рассмотрим графики с настройками отображения.

В каждой из библиотек мы получили результат, который обеспечен цветовой картой, надписями, заголовками, группировкой. Хорошим преимуществом для Plotly является интерактивная фильтрация, которая позволяет исключать данные из визуализации. В seaborn и matplotlib эта функция была в некотором смысле реализована с помощью поля фильтрации Х, которая позволяла фильтровать данные по одной группе.

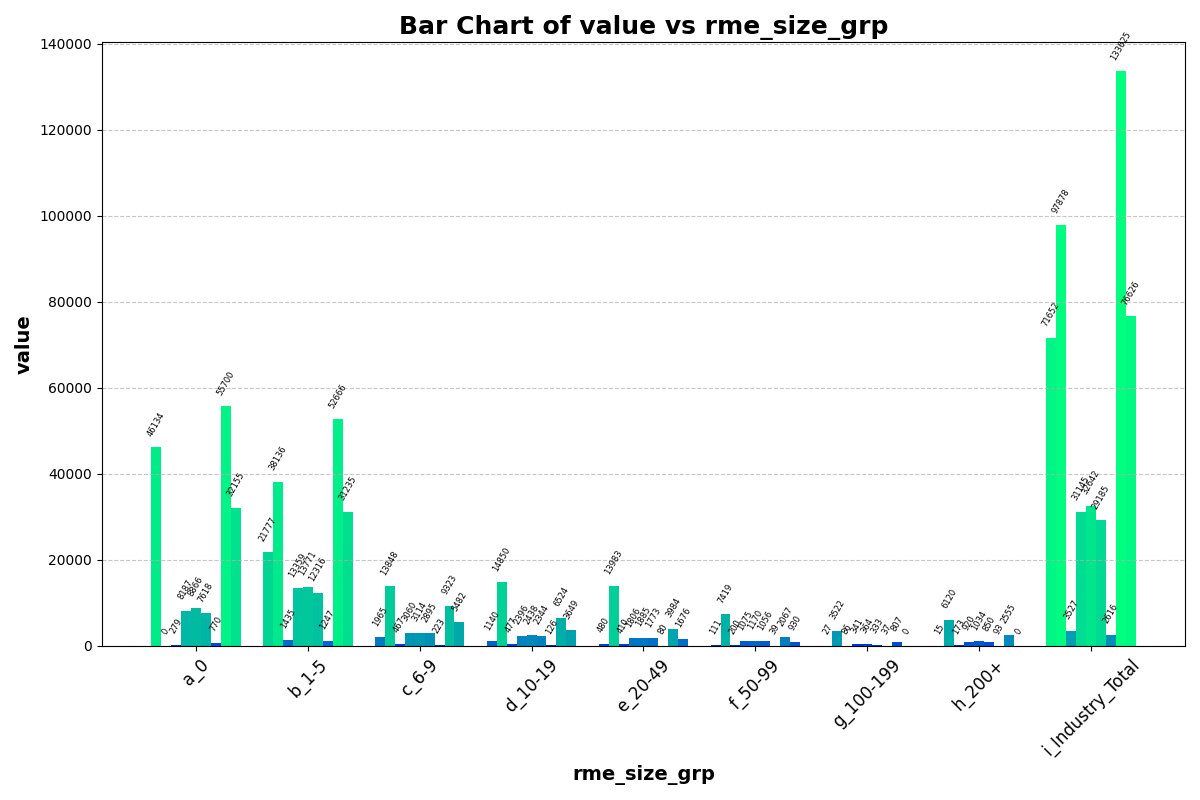


Рисунок 19 – Matplotlib Bar final

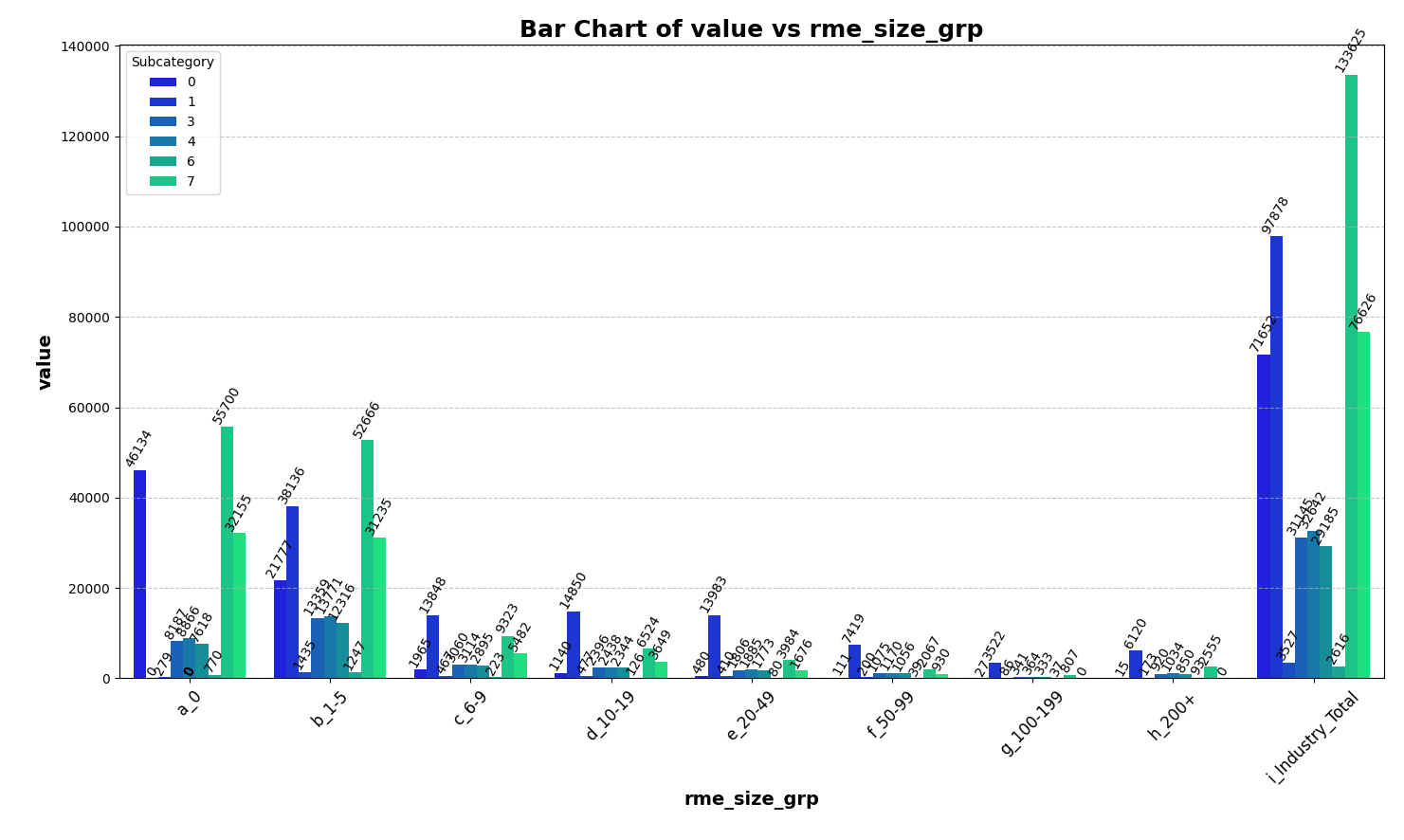


Рисунок 20 – Seaborn Bar final

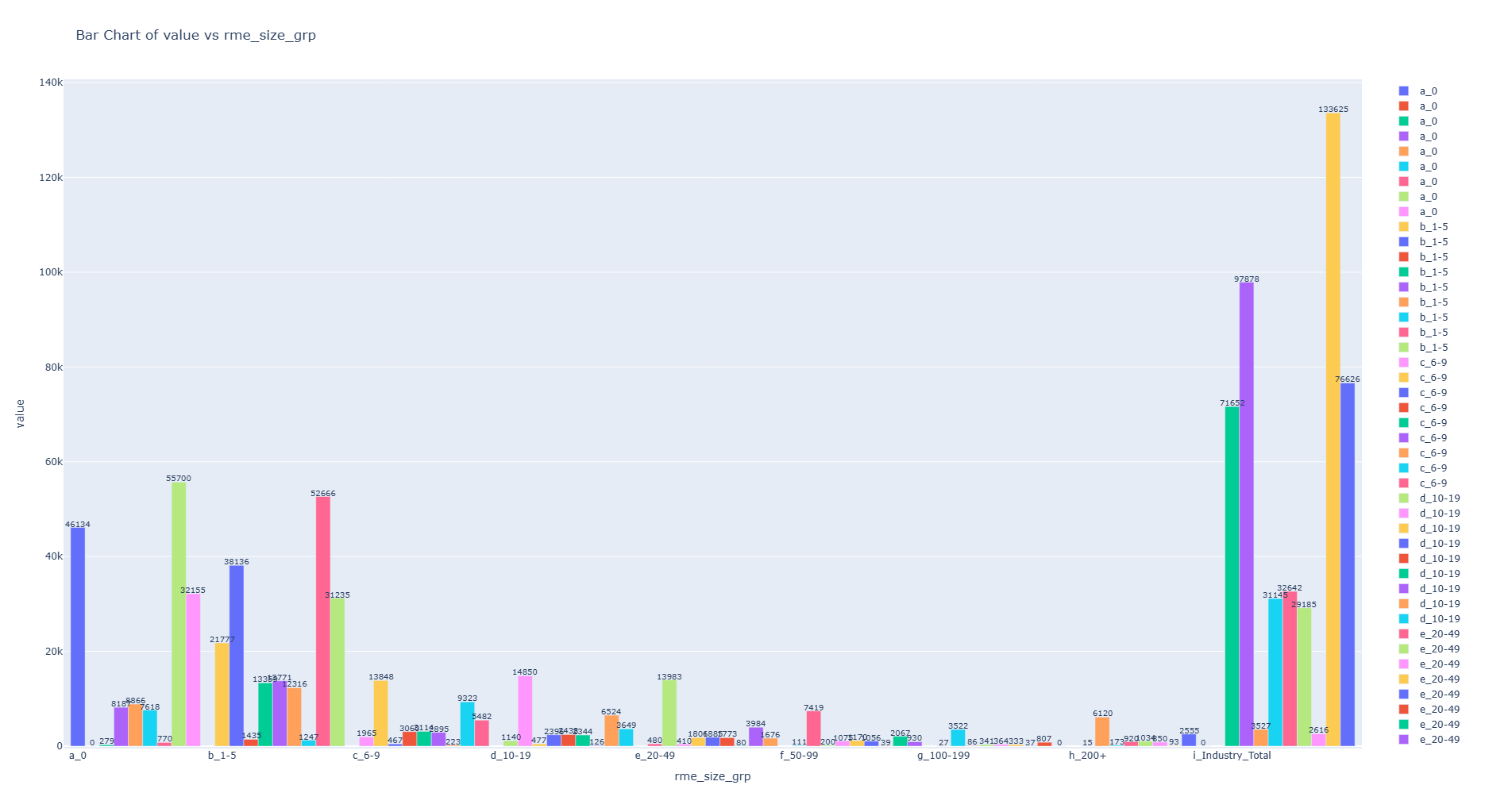


Рисунок 21 – Plotly Bar final

Остальные графики размещены в приложении.

## Рекомендации по выбору библиотеки в зависимости от проекта

### Научные исследования и публикации

Рекомендуемая библиотека: Matplotlib

Причины: Matplotlib предоставляет высокую гибкость и возможность детальной настройки графиков, что важно для точного представления данных в научных работах. Вы сможете адаптировать графики под требования журналов и конференций.

### Анализ данных и отчетность

Рекомендуемая библиотека: Seaborn

Причины: Seaborn идеально подходит для быстрого анализа данных и создания стильных визуализаций. Она хорошо интегрируется с Pandas, что позволяет легко обрабатывать данные и визуализировать их в привлекательном виде. Это особенно полезно для создания отчетов и презентаций.

### Веб-приложения и интерактивные дашборды

Рекомендуемая библиотека: Plotly

Причины: Plotly позволяет создавать интерактивные графики, что делает её отличным выбором для веб-приложений и дашбордов. Пользователи смогут взаимодействовать с данными, что повышает вовлеченность и улучшает восприятие информации.

### Презентации и визуализация данных для бизнеса

Рекомендуемая библиотека: Plotly или Seaborn

Причины: Для бизнес-презентаций важна как визуальная привлекательность, так и возможность взаимодействия с графиками. Plotly предоставляет интерактивные элементы, в то время как Seaborn позволяет быстро создавать стильные и информативные графики.

### Образовательные проекты и курсовые работы

Рекомендуемая библиотека: Seaborn или Matplotlib

Причины: Seaborn будет хорошим выбором для быстрого создания визуализаций с минимальными усилиями, что полезно для образовательных проектов. Matplotlib также подойдет, если требуется глубокая настройка графиков.

### Проекты с акцентом на визуализацию больших данных

Рекомендуемая библиотека: Plotly

Причины: Для работы с большими объемами данных и их визуализации Plotly предоставляет мощные инструменты для создания интерактивных графиков, которые могут легко обрабатывать и отображать большие наборы данных.

# ЗАКЛЮЧЕНИЕ

## Обзор выполненной работы

В рамках данной работы была проведена визуализация данных с использованием трех популярных библиотек: Matplotlib, Seaborn и Plotly. Основная цель заключалась в создании базовых графиков (столбчатых, линейных, гистограмм и контурных) и их доработке для достижения максимальной визуальной понятности и информативности.

### ****Создание базовых графиков****

**Столбчатые графики (Bar Charts)**:

* **Matplotlib**: Используя Matplotlib, был создан базовый столбчатый график для представления категориальных данных. График был простым, но требовал доработки для улучшения читабельности;
* **Seaborn**: С Seaborn был создан более стильный столбчатый график с использованием встроенных цветовых палитр, что сделало его более привлекательным и информативным
* **Plotly**: В Plotly был разработан интерактивный столбчатый график, позволяющий пользователю взаимодействовать с данными, что добавило ценности для анализа

**Линейные графики (Line Charts):**

**Гистограммы (Histograms):**

* **Matplotlib**: Базовая гистограмма была создана для визуализации распределения данных, но требовала улучшений;
* **Seaborn**: С помощью Seaborn была создана гистограмма с наложением KDE (Kernel Density Estimate), что позволило лучше визуализировать распределение;
* **Plotly**: В Plotly была реализована интерактивная гистограмма, что дало возможность пользователю настраивать диапазоны и исследовать данные

**Контурные графики (Contour Plots)**:

* **Matplotlib**: Контурный график был создан для визуализации двумерных данных, однако имел ограниченные визуальные возможности;
* **Seaborn**: С Seaborn был создан более эстетичный контурный график с улучшенной цветовой схемой и четкими уровнями;
* **Plotly**: Контурный график в Plotly был интерактивным, позволяя пользователю исследовать уровни и значения в реальном времени

### ****Доработка графиков для визуального оформления и информативности****

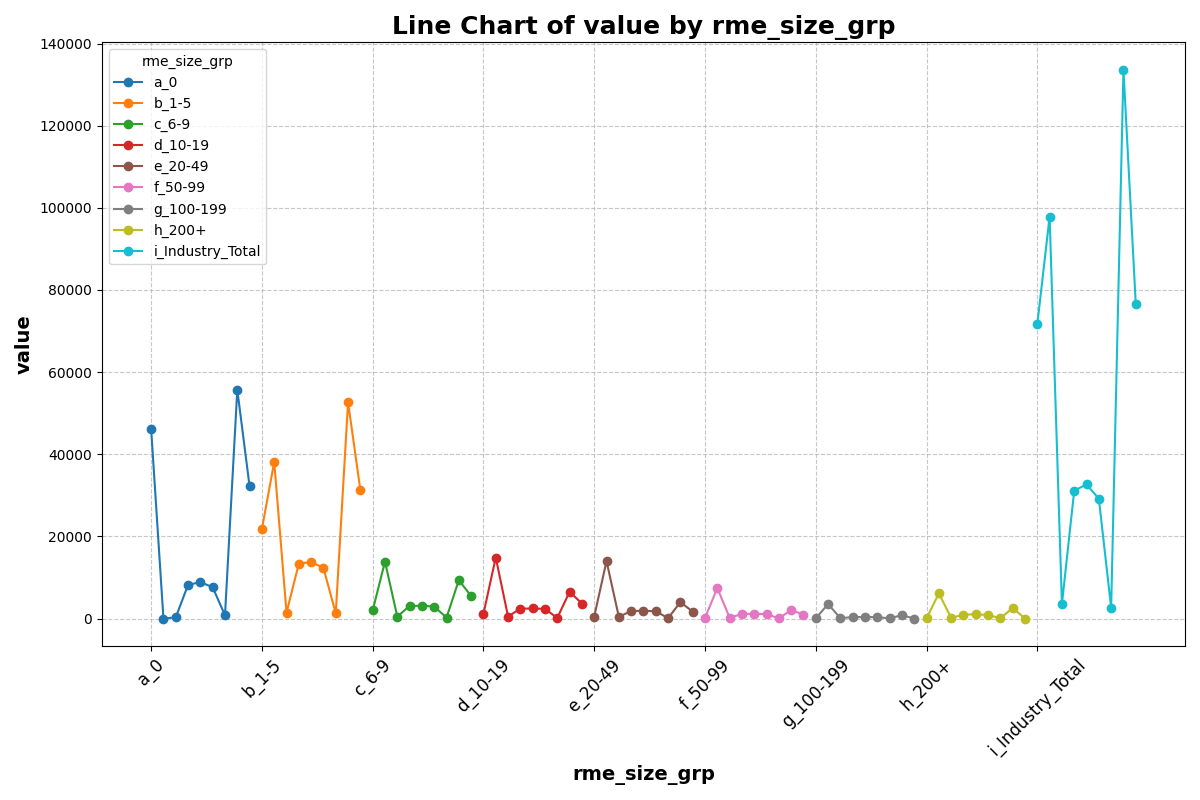
После создания базовых графиков, была проведена работа по их доработке:

* Улучшение цветовой схемы: Все графики были доработаны с использованием цветовых палитр, которые обеспечивают лучшую контрастность и восприятие информации;
* Добавление подписей и заголовков: Каждому графику были добавлены четкие заголовки, подписи осей и легенды, что значительно улучшило их информативность;
* Группировка данных: К каждому графику была применена группировка данных по оси Х для улучшения отображения и исключения визуального шума.

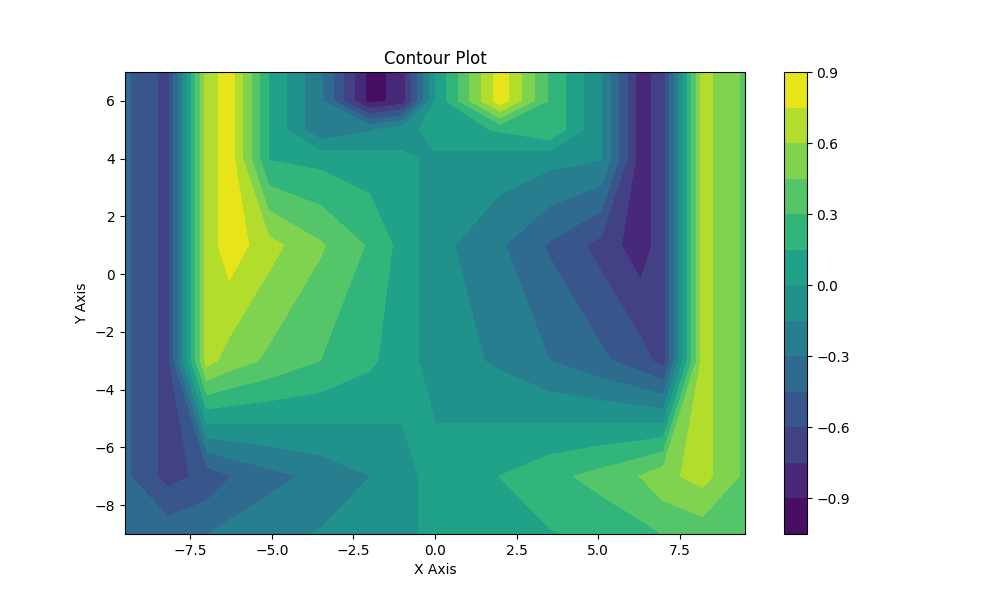
## Дальнейшие планы

Доработка приложения для улучшения производительности и стабильности. Дополнение функциональными кнопками для индивидуальной настройки цветовой схемы, подписей и прочих настроек. Дополнение новыми графиками.

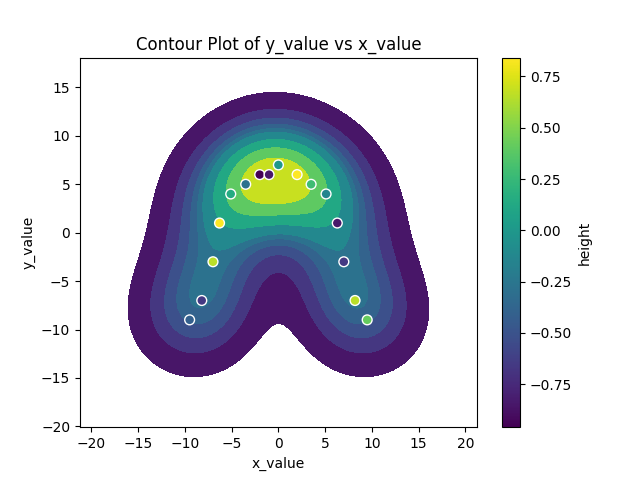
# ПРИЛОЖЕНИЕ



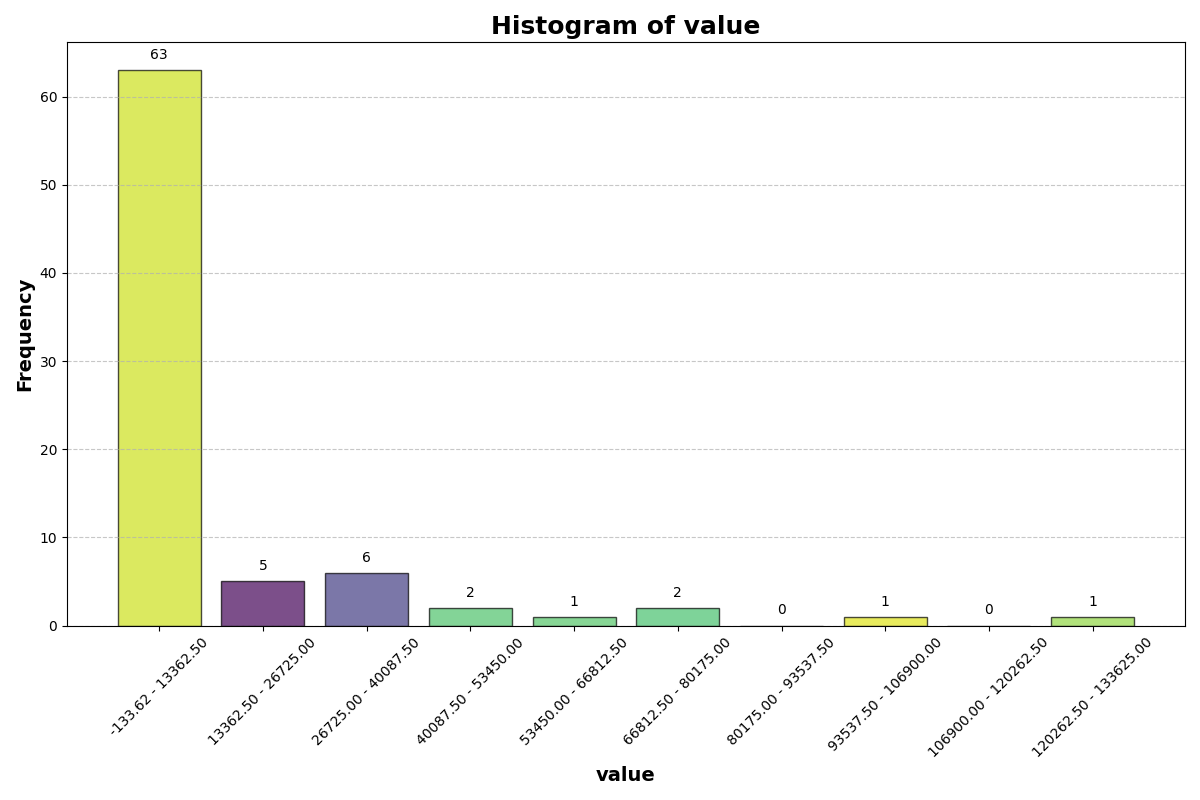
Matplotlib Line



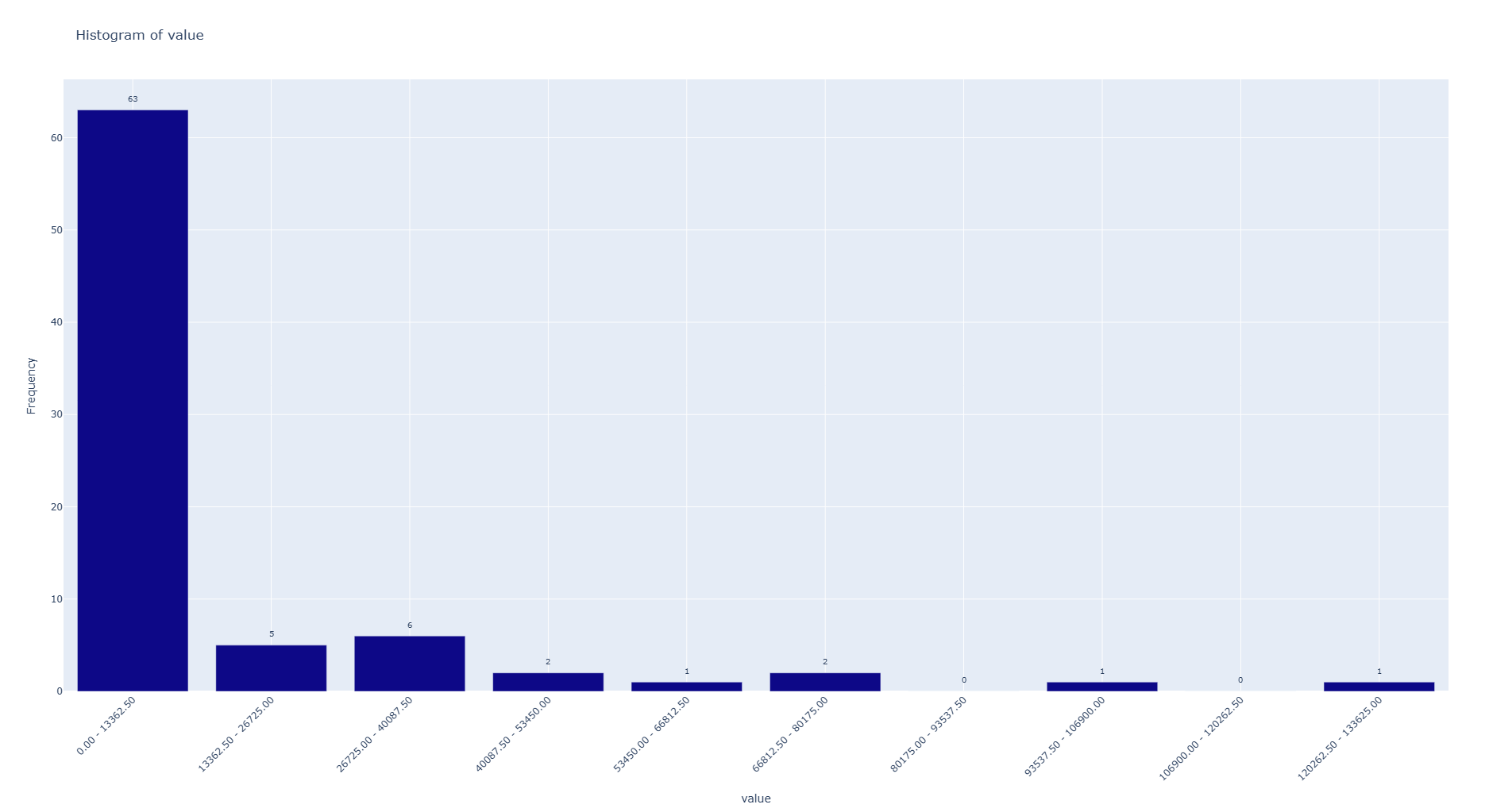
Matplotlib Contour



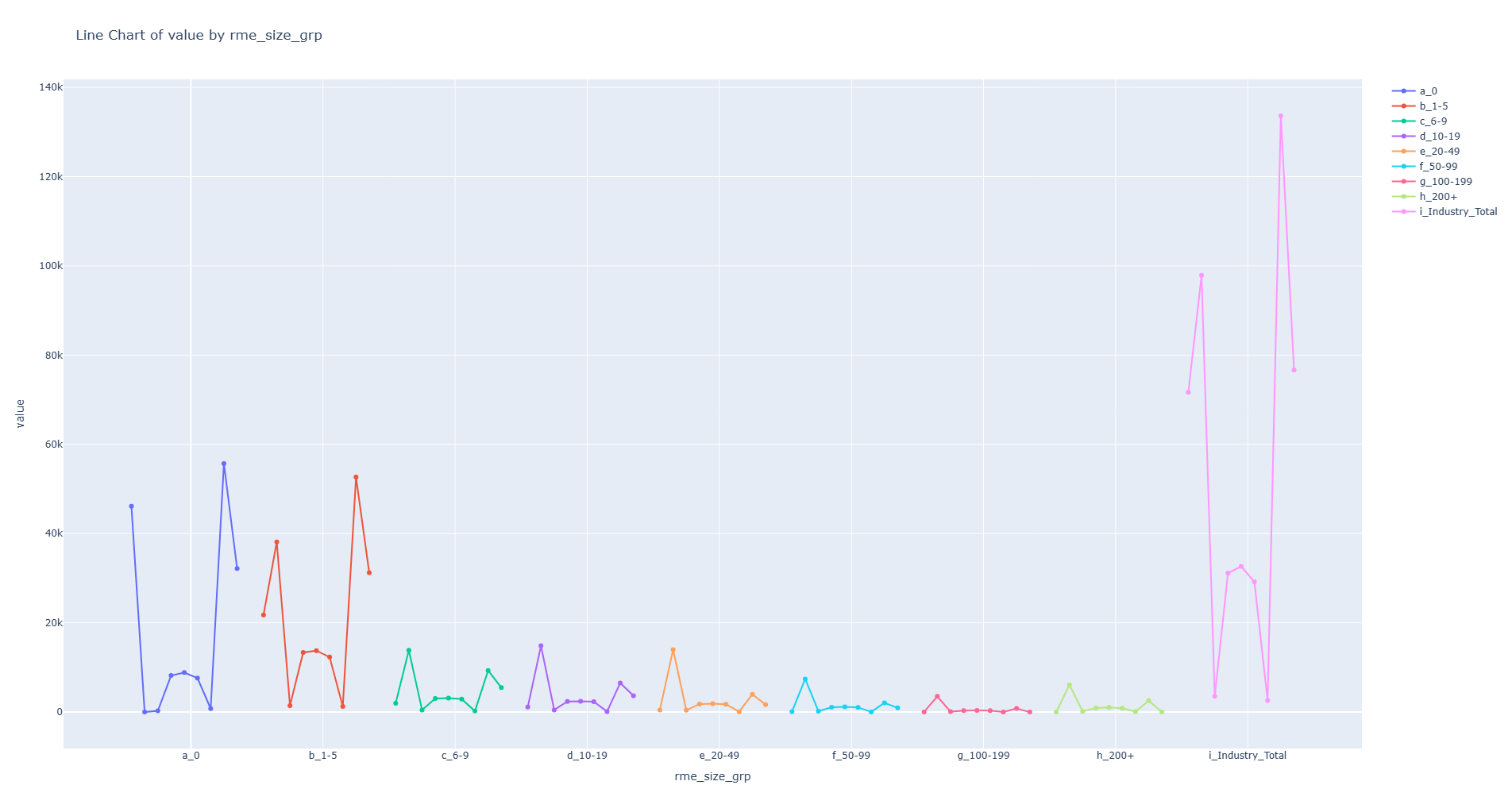
Seaborn Contour



Matplotlib Histogram



Plotly Histogram



Plotly Line