

# Консультация Спринт 2. Проект глазами ревьюера 🙄🙄

Горленко Екатерина,  
Наставник, ревьюер факультета DA, Яндекс.Практикум

Яндекс Практикум

**Все заряжены  
и готовы? :)**



# Наши договорённости

## Организованность

- Будь вовремя
- Понятное имя в Zoom (и везде)
- Камера включена
- Правило одного включенного микрофона
- Вопросы в чате

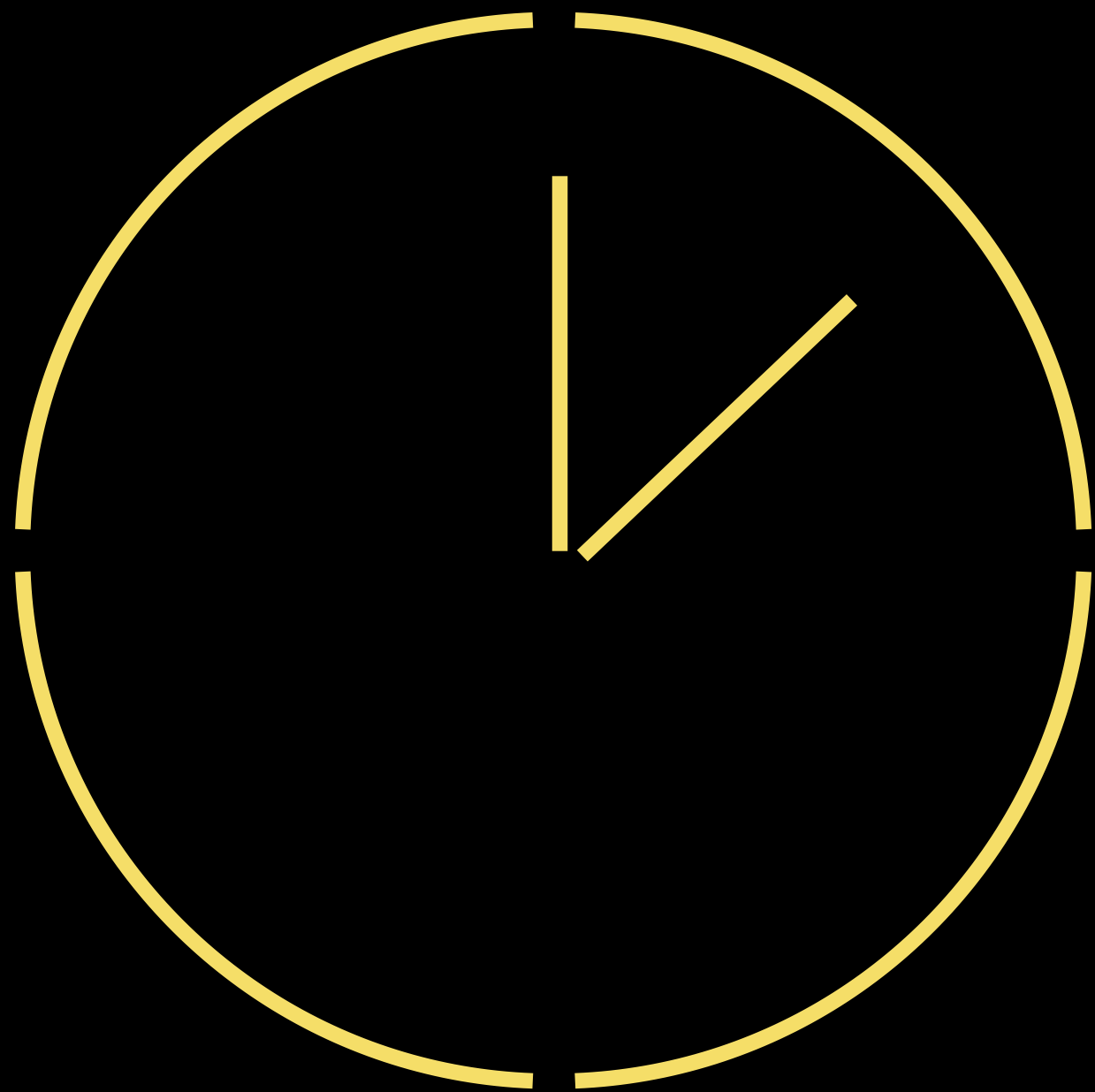
## Комфортная коммуникация

- Общаемся на «ты»
- Вовлеченность и проактивность
- Обучение – это ошибки
- Уважительное отношение
- Критикуешь – предлагай
- Береги общее время
- Запись только для потока

## Цель – научиться

- Мы – разные
- Учимся самостоятельно принимать решения
- Самостоятельность – это поиск и общение, а не одиночество
- Взаимопомощь
- Нет спойлерам

# План встречи



- Знакомство (5 мин)
- Статус спринта (5 мин)
- Основная часть (30 - 40 мин):
  - Что такое проект?
  - Структура проекта
  - Оформление
  - Частые ошибки

Вопросы? (10 мин)

# Знакомство

Статус спринта

[menti.com](https://menti.com)

код 56 27 08 9

# Основная часть

# Проект vs реальная задача

- Что такое проект?
- Структура проекта
- Оформление
- Частые ошибки

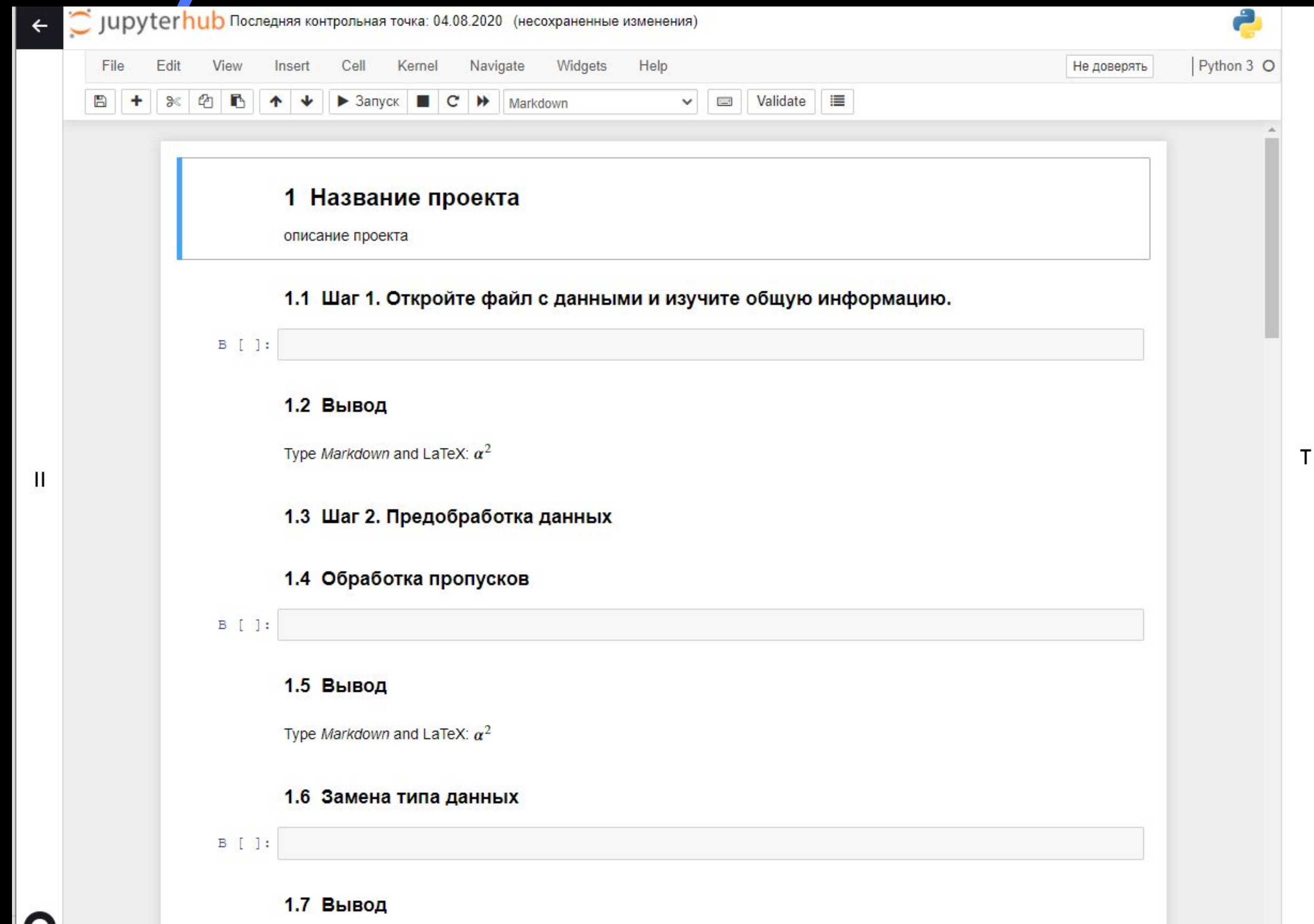
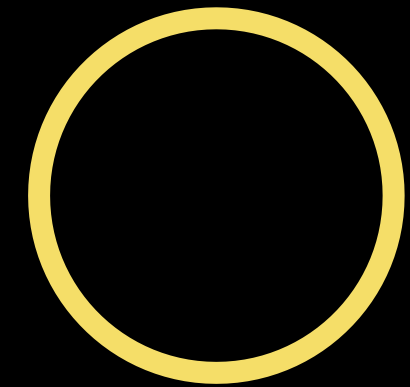


Яндекс Практикум	Твоя работа
Проект	Боевая задача
Ревьюер	Коллега / лид / топ



# Jupyter Notebook

- Что такое проект?
- Структура проекта
- Оформление
- Частые ошибки



# Jupyter Notebook


- Что такое проект?
- Структура проекта
- Оформление
- Частые ошибки

## 1.20 Чек-лист готовности проекта

Поставьте 'x' в выполненных пунктах. Далее нажмите Shift+Enter.

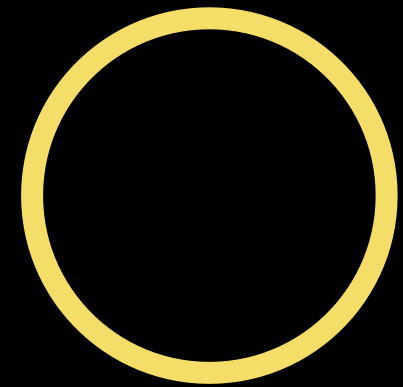
- ☒ открыт файл;
- ☐ файл изучен;
- ☐ определены пропущенные значения;
- ☐ заполнены пропущенные значения;
- ☐ есть пояснение, какие пропущенные значения обнаружены;
- ☐ описаны возможные причины появления пропусков в данных;
- ☐ объяснено, по какому принципу заполнены пропуски;
- ☐ заменен вещественный тип данных на целочисленный;
- ☐ есть пояснение, какой метод используется для изменения типа данных и почему;
- ☐ удалены дубликаты;
- ☐ есть пояснение, какой метод используется для поиска и удаления дубликатов;
- ☐ описаны возможные причины появления дубликатов в данных;
- ☐ выделены леммы в значениях столбца с целями получения кредита;
- ☐ описан процесс лемматизации;
- ☐ данные категоризированы;
- ☐ есть объяснение принципа категоризации данных;
- ☐ есть ответ на вопрос: "Есть ли зависимость между наличием детей и возвратом кредита в срок?";
- ☐ есть ответ на вопрос: "Есть ли зависимость между семейным положением и возвратом кредита в срок?";
- ☐ есть ответ на вопрос: "Есть ли зависимость между уровнем дохода и возвратом кредита в срок?";
- ☐ есть ответ на вопрос: "Как разные цели кредита влияют на его возврат в срок?";
- ☐ в каждом этапе есть выводы;
- ☐ есть общий вывод.

# Основные разделы

- 
- Что такое проект?
  - Структура проекта
  - Оформление
  - Частые ошибки

1. Название проекта
2. Цель проекта (!)
3. Шаги / задачи / описание
4. Шаг 1  
Решение с комментариями  
Вывод по шагу 1 (!)
5. Шаг 2 ... n
6. Общий вывод (!)

# Основные разделы



- Что такое проект?
- Структура проекта
- Оформление
- Частые ошибки

1. Название проекта

## 2. Цель проекта (!)

3. Шаги / задачи / описание

4. Шаг 1

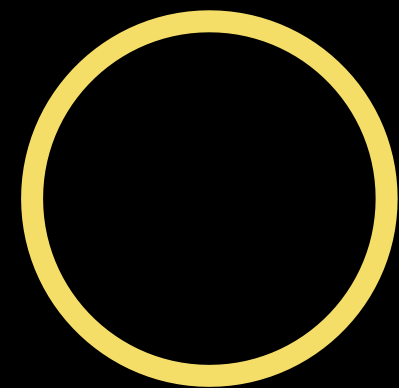
Решение с комментариями  
Вывод по шагу 1 (!)

5. Шаг 2 ... n

6. ОБЩИЙ ВЫВОД (!)



# Цель проекта



- Что такое проект?
- Структура проекта
- Оформление
- Частые ошибки

☐ Скопировать кусок текста из описания «Практикума»

## Описание проекта

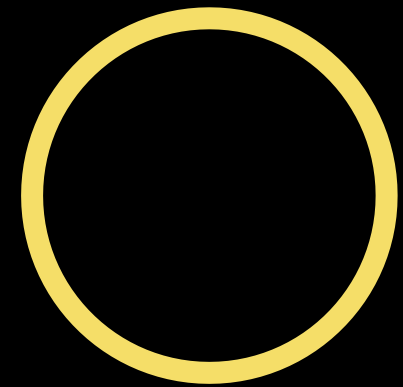
Заказчик — кредитный отдел банка. Нужно разобраться, влияет ли семейное положение и количество детей клиента на факт погашения кредита в срок.  
Входные данные от банка — статистика о платёжеспособности клиентов.

☐ «... научиться что-то делать ...»

☐ Забыть про цель

☒ Сформулировать практическую цель исследования

# Основные разделы



- Что такое проект?
- Структура проекта
- Оформление
- Частые ошибки

1. Название проекта
2. Цель проекта (!)
3. Шаги / задачи / описание
4. Шаг 1

## Решение с комментариями

Вывод по шагу 1 (!)

5. Шаг 2 ... n
6. Общий вывод (!)

## Комментарий к расчетам

Как видно из предыдущего результата, ранее установленное разделение на категории в целом верно. Однако появились одна, довольно многочисленная, категория - 'операция'. Если посмотреть контекст, в котором это слово употреблялось, становится ясно, что оно относится на операции с недвижимостью, а поскольку недвижимость самое часто встречающееся слово, то и довольно большое количество повторений слова 'операция' оправданно. Теперь проведем лемматизацию для столбца 'purpose'.

```
В [27]: #лемматизация для столбца 'purpose'
all_cat = ['автомобиль', 'жилье', 'недвижимость', 'образование', 'свадьба']


#функция которая возвращает категорию, к которой относится принятое значение из столбца 'purpose'
#или None, в случае, если принятое значение не относится к категориям которые мы назначили
def purp_cat(purpose):
    purp_lemm = m.lemmatize(purpose)
    for cat in all_cat:
        if cat in purp_lemm:
            return cat
    return None

customers['purpose'] = customers['purpose'].apply(purp_cat)

customers.info()
```

## Комментарий к коду

# Основные разделы

- 
- Что такое проект?
  - Структура проекта
  - Оформление
  - Частые ошибки


1. Название проекта
2. Цель проекта (!)
3. Шаги / задачи / описание
4. Шаг 1  
Решение с комментариями  
**Вывод по шагу 1 (!)**
5. Шаг 2 ... n
6. Общий вывод (!)





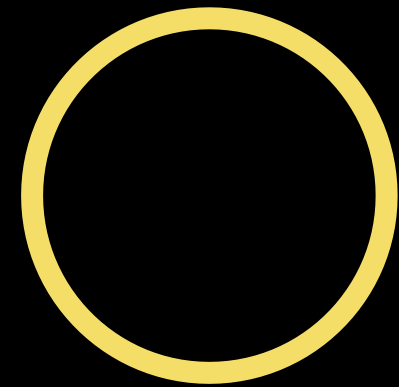
ШОК! Не все умеют читать  
код!

# Основные разделы

- 
- Что такое проект?
  - Структура проекта
  - Оформление
  - Частые ошибки

1. Название проекта
2. Цель проекта (!)
3. Шаги / задачи / описание
4. Шаг 1  
Решение с комментариями  
Вывод по шагу 1 (!)
5. Шаг 2 ... n
6. **Общий вывод (!)**

# Общий вывод

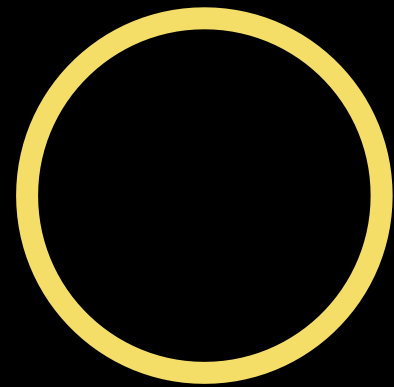


- Что такое проект?
- Структура проекта
- Оформление
- Частые ошибки

- ✓ Собрать все найденные метрики
- ✓ Подчеркнуть важные инсайты
- ✓ Обобщить наблюдения и выводы в рамках поставленной цели
- ✓ Дать рекомендации по дальнейшему использованию исследования

- ☐ Скопировать все предыдущие блоки с выводами

# Оформление



- Что такое проект?
- Структура проекта
- Оформление
- Частые ошибки

- Используй маркдаун для всех текстовых блоков
- Заголовки разного уровня
- Нумерация
- Списки
- Оглавление
- Форматирование

[https://paulradzkov.com/2014/markdown\\_cheatsheet/](https://paulradzkov.com/2014/markdown_cheatsheet/)  
Шпаргалка по Маркдаун

<https://www.notion.so/08a7e9d6ada746fbb9562171e29dacf8>  
Советы от Практикума  
Яндекс.Практикум



# Оформление

- Что такое проект?
- Структура проекта
- **Оформление**
- Частые ошибки

[https://paulradzkov.com/2014/markdown\\_cheatsheet/](https://paulradzkov.com/2014/markdown_cheatsheet/)  
Шпаргалка по Маркдаун

<https://www.notion.so/08a7e9d6ada746fbb9562171e29dacf8>  
Советы от Практикума  
Яндекс.Практикум

## Работа с проектом

- 📄 Скачать проект и датасет
- 📁 Как называть учебные (и не только) файлы
- 🔗 Как встроить сторонний файл с данными в проект
- 📶 Загрузить проект в тренажер / JupyterHub, не отправляя на проверку

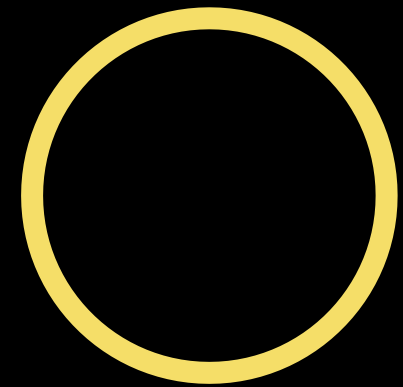
## Оформление проекта

- 📄 Как оформлять проект
- 👤 Правила размещения проектов
- 🐶 Размещение проектов на репозитории
- 🔧 Работа с локальным репозиторием с помощью консольного git
- 🌍 Гитхаб в Крыму. Инструкция работы

## Установка и настройка Jupyter Notebook

- 📦 Как установить Jupyter Notebook на свой компьютер и работать локально
- 📖 Установка библиотек
- 🌱 Установка окружения — как избежать конфликта версий библиотек

# Частые ошибки



- Что такое проект?
- Структура проекта
- Оформление
- Частые ошибки

- Код **не отрабатывает** (Kernel > Restart & Run All)
- Проект сделан **на локальной машине** и:
  - Путь к данным неверный
  - Библиотеки несовместимы
- Нет комментариев и **ВЫВОДОВ**
- **Не все шаги** выполнены
- **Порядок шагов** нарушен
- Весь код написан **в одной** или нескольких ячейках
- Комментарии и выводы написаны **как комментарии** к коду
- **Нет подписей** к графикам, осям и легенды
- Оставлен и закомментирован старый, **ненужный код**

## 1. Подготовка данных к

Прочитаем исходные файлы, сохраним в переменные

```
In [1719]: import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
import math

visits = pd.read_csv('/datasets/visits_lo
```



```

In [2]: #data.head(10)

In [3]: #data.tail(10)

In [4]: #data.info()

In [5]: # 0
##data.sort_values(by = 'children')['children'].value_counts()

In [6]: # 0.1
##print(data[data['children'] == 20])

In [7]: # 0.2
##print(data[data['children'] == -1])

In [8]: # 1
##print('Минимальное значение: {}'.format(data['days_employed'].min()))
##print('Максимальное значение: {}'.format(data['days_employed'].max()))

In [9]: # 2
##print('Минимальное значение: {}'.format(data['dob_years'].min()))
##print('Минимальное ненулевое значение: {}'.format(data[data['dob_years'] != 0]['dob_years'].min()))
##print('Максимальное значение: {}'.format(data['dob_years'].max()))

In [10]: # 3
##data.sort_values(by = 'education')['education'].value_counts()

In [11]: # 4
##data.sort_values(by = 'education_id')['education_id'].value_counts()

In [12]: # 5
##data.sort_values(by = 'family_status')['family_status'].value_counts()

In [13]: # 6
##data.sort_values(by = 'family_status_id')['family_status_id'].value_counts()

In [14]: # 7
##data.sort_values(by = 'gender')['gender'].value_counts()

In [15]: # 7.8
#data[data['gender'] == 'XNA']

In [16]: # 8
##data.sort_values(by = 'Income_type')['income_type'].value_counts()

In [17]: # 8.8
##data[data['income_type'] == 'предприниматель']

```

## 1.1.2 Функция первичной обработки

Данная функция определяет тип данных в столбце, убирает пропуски, выдает основную статистическую информацию и строит графики.

```

In [2]: def get_information_before_manipulation(data, name_column):
    type_value = data[name_column].dtype.name

    if type_value == 'float64' or type_value == 'int64':
        print_graphs_before_manipulation2(data, name_column, type_value)
        return print_information_before_manipulation(data, name_column, type_value)

    if type_value == 'object':
        return print_information_before_manipulation(data, name_column, type_value)

    if type_value == 'bool':
        print_graphs_before_manipulation2(data, name_column, type_value)
        return print_information_before_manipulation(data, name_column, type_value)

    if type_value == 'datetime64[ns]':
        print_graphs_before_manipulation2(data, name_column, type_value)
        return print_information_before_manipulation(data, name_column, type_value)

In [3]: def print_information_before_manipulation(data, name_column, type_value):
    list_params = [['Всего строк в выборке', len(data)],
                    ['Уникальные значения', len(data[name_column].unique())],
                    ['Пропущенные значения', int(len(data)-data[name_column].describe()[1]['count'])],
                    ['Пропущенные значения в %', '{:.2f}'.format(1-data[name_column].describe()[1]['count']/len(data))],
                    ['Тип данных в столбце', type_value]]

    if type_value == 'float64' or type_value == 'int64':
        list_params_float64 = [['Количество уникальных значений', len(data[name_column].unique())],
                                ['Минимальное значение', data[name_column].min()],
                                ['Максимальное значение', data[name_column].max()],
                                ['Среднее арифметическое', round(data[name_column].mean(), 2)],
                                ['Медиана', data[name_column].median()],
                                ['Стандартное отклонение (σ)', data[name_column].describe()[1]['std']],
                                ['Дисперсия (σ**2)', data[name_column].var()]]

        list_params = list_params + list_params_float64
        df = pd.DataFrame(list_params)
        df.columns = ['Характеристика', name_column]
        df = df.set_index('Характеристика')
        return df

    elif type_value == 'object':
        list_params_object = [['Количество уникальных значений', len(data[name_column].unique())]]
        list_params = list_params + list_params_object
        df = pd.DataFrame(list_params)
        df.columns = ['Характеристика', name_column]
        df = df.set_index('Характеристика')
        return df

    elif type_value == 'bool':
        df = pd.DataFrame(list_params)
        df.columns = ['Характеристика', name_column]
        df = df.set_index('Характеристика')
        return df

    elif type_value == 'datetime64[ns]':
        list_params_datetime64 = [['Количество уникальных значений', len(data[name_column].unique())],
                                    ['Минимальное значение', data[name_column].min()],
                                    ['Максимальное значение', data[name_column].max()]]

        list_params = list_params + list_params_datetime64
        df = pd.DataFrame(list_params)
        df.columns = ['Характеристика', name_column]
        df = df.set_index('Характеристика')
        return df

In [4]: def print_graphs_before_manipulation2(data, name_column, type_value):
    if type_value == 'float64' or type_value == 'int64':
        data = data[pd.isnull(data[name_column]) == False]
        bins = len(data[name_column].unique())
        labels = data[name_column].index
        values = data[name_column].values

        sns.set(style='white', palette='muted', color_codes=True)
        f, axes = plt.subplots(1, 2, figsize=(16, 4))

        sns.boxplot(values, orient='g', color='r', ax=axes[0])
        sns.distplot(values, kde_kws={'shade': True}, color='g', ax=axes[1])

        f.suptitle('Визуализация по столбцу {}'.format(name_column),
                  fontsize=14,
                  fontweight='bold')

        axes[0].title.set_text('Boxplot, срезы {}, датасет {}'.format(name_column))
        axes[1].title.set_text('Гистограмма, срезы {}, датасет {}'.format(name_column))

In [5]: def get_information_after_manipulation(start_data, finish_data, name_column):
    type_value_finish = finish_data[name_column].dtype.name
    type_value_start = start_data[name_column].dtype.name

    list_params = [
        ['Всего строк в выборке', len(start_data), len(finish_data)],
        ['Уникальные значения', len(start_data[name_column].unique()), len(finish_data[name_column].unique())],
        ['Пропущенные значения', int(start_data[name_column].describe()[1]['count']), int(finish_data[name_column].describe()[1]['count'])],
        ['Пропущенные значения в %', '{:.2f}'.format(1-start_data[name_column].describe()[1]['count']/len(start_data)), '{:.2f}'.format(1-finish_data[name_column].describe()[1]['count']/len(finish_data))],
        ['Тип данных в столбце', type_value_start, type_value_finish]]

    if type_value_finish == 'float64' or type_value_finish == 'int64':
        list_params_float64 = [['Количество уникальных значений', len(start_data[name_column].unique()), len(finish_data[name_column].unique())],
                                ['Минимальное значение', start_data[name_column].min(), finish_data[name_column].min()],
                                ['Максимальное значение', start_data[name_column].max(), finish_data[name_column].max()],
                                ['Среднее арифметическое', round(start_data[name_column].mean(), 2), round(finish_data[name_column].mean(), 2)],
                                ['Медиана', start_data[name_column].median(), finish_data[name_column].median()],
                                ['Стандартное отклонение (σ)', start_data[name_column].describe()[1]['std'], finish_data[name_column].describe()[1]['std']],
                                ['Дисперсия (σ**2)', start_data[name_column].var(), finish_data[name_column].var()]]

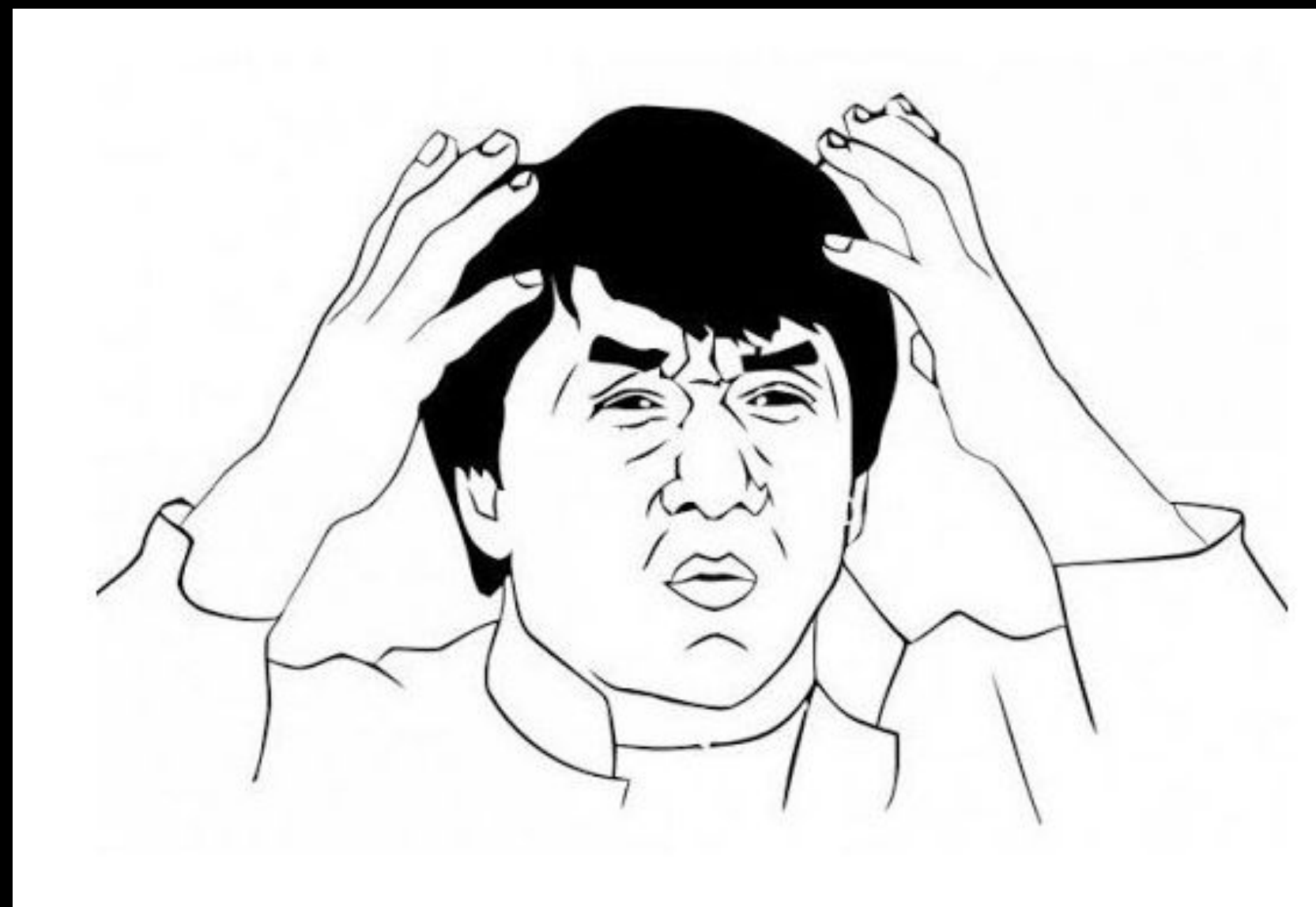
        list_params = list_params + list_params_float64
        df = pd.DataFrame(list_params)
        df.columns = ['Характеристика', 'Dataframe 1', 'Dataframe 2']
        df = df.set_index('Характеристика')
        return df

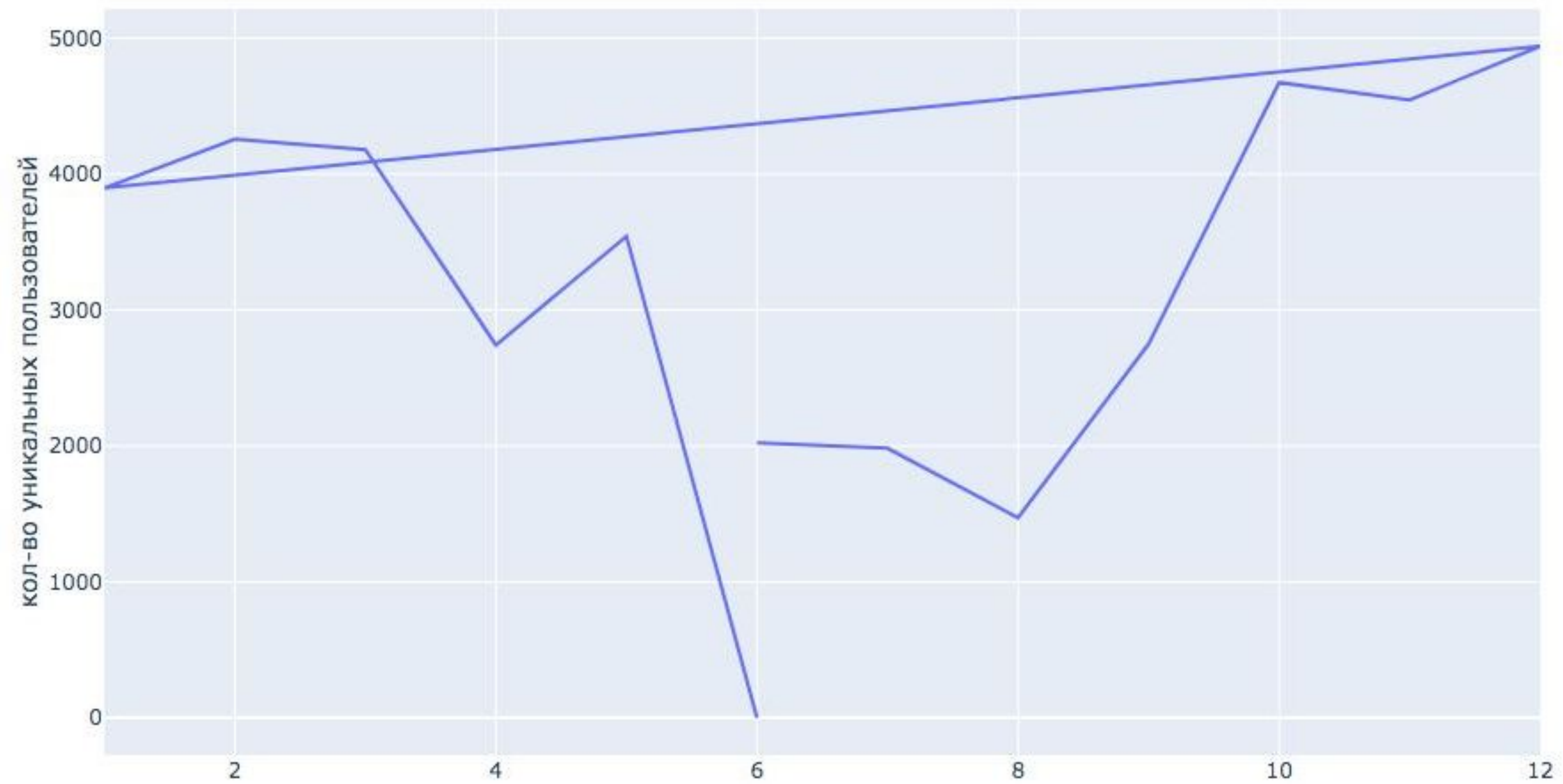
```

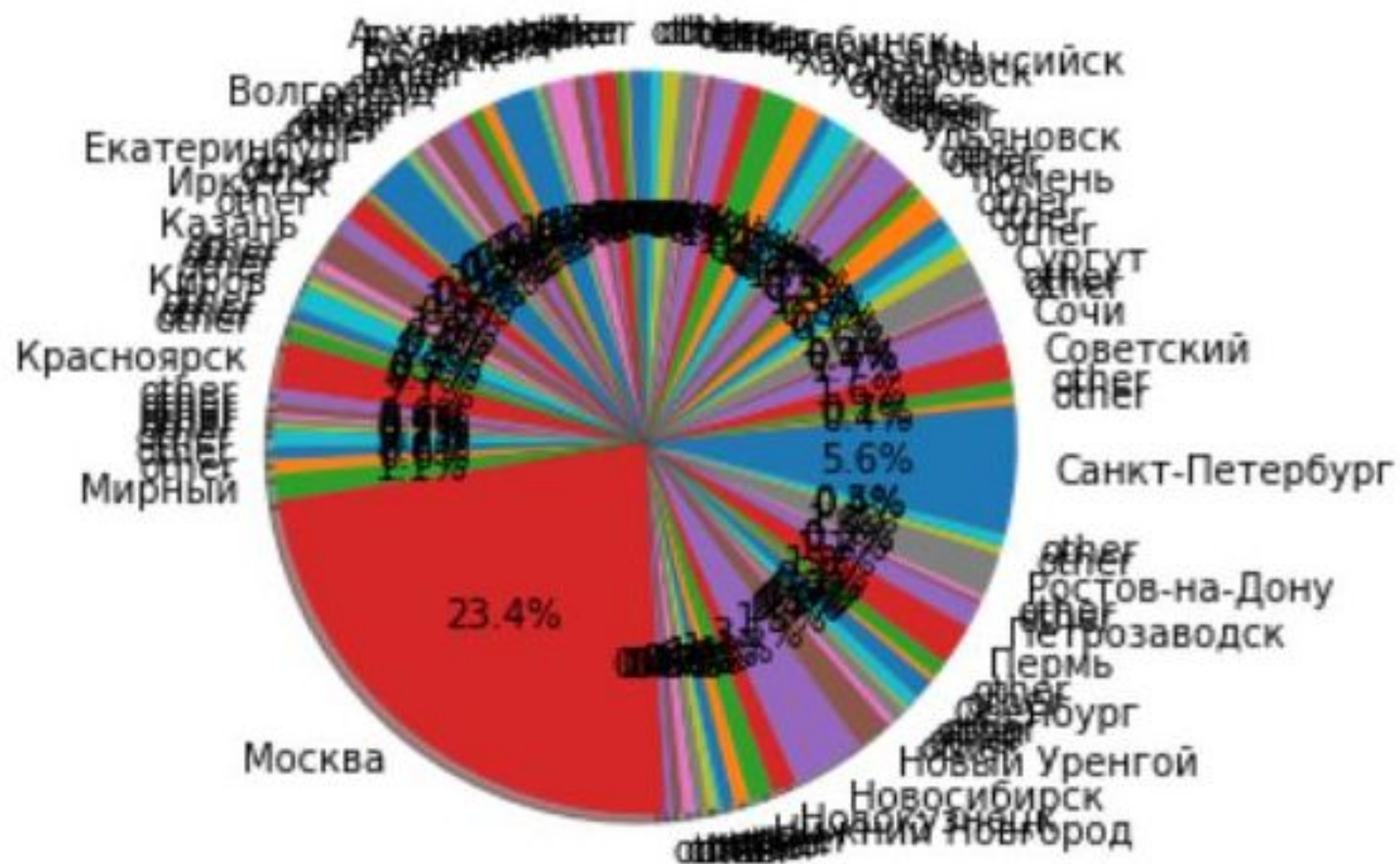


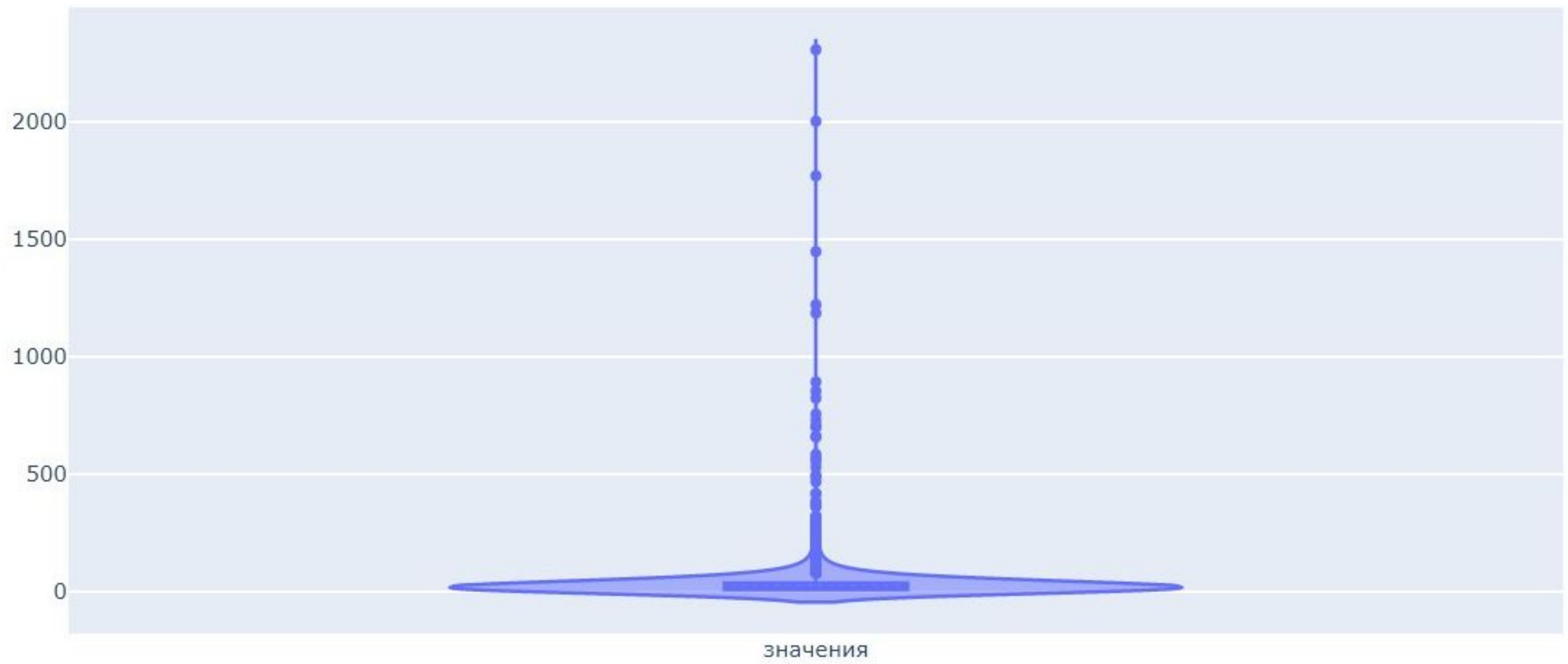
#### **Шаг 4. Общий вывод**

Ну в общем все понятно по графикам!



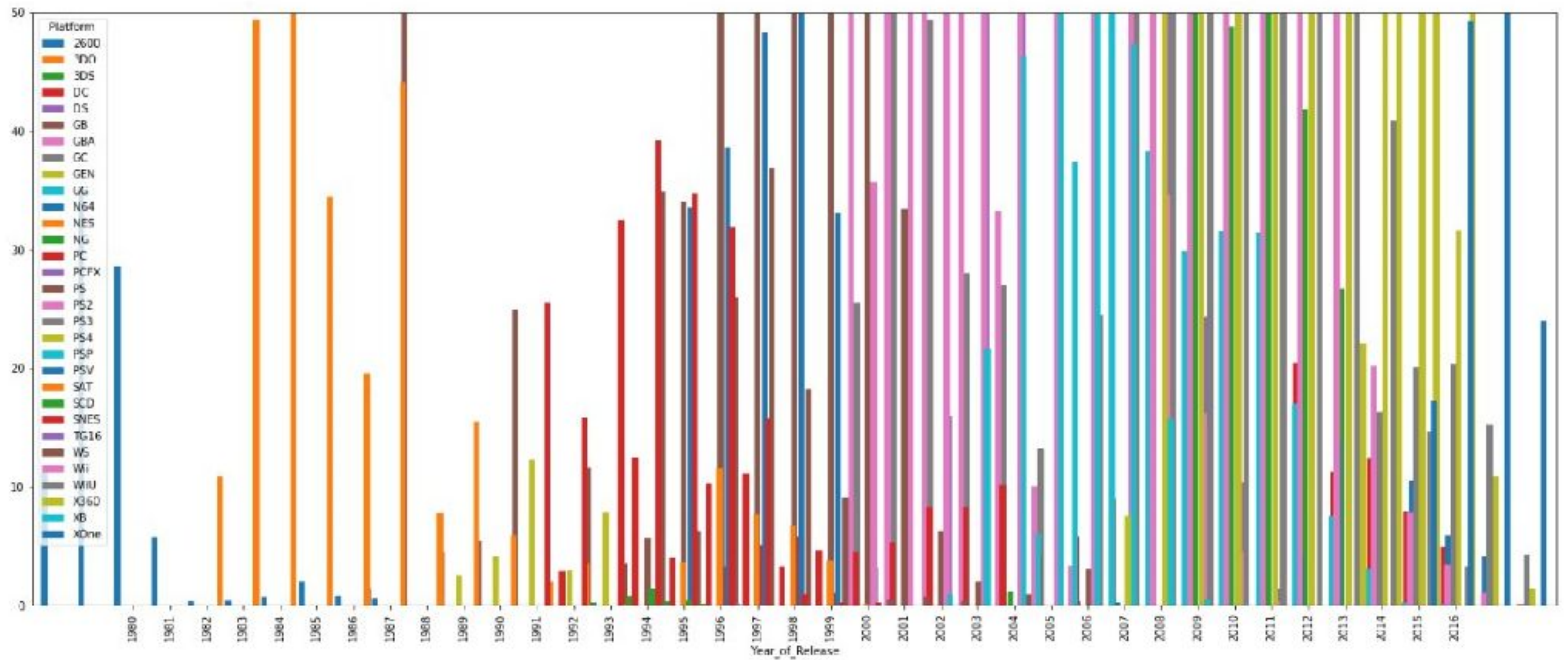






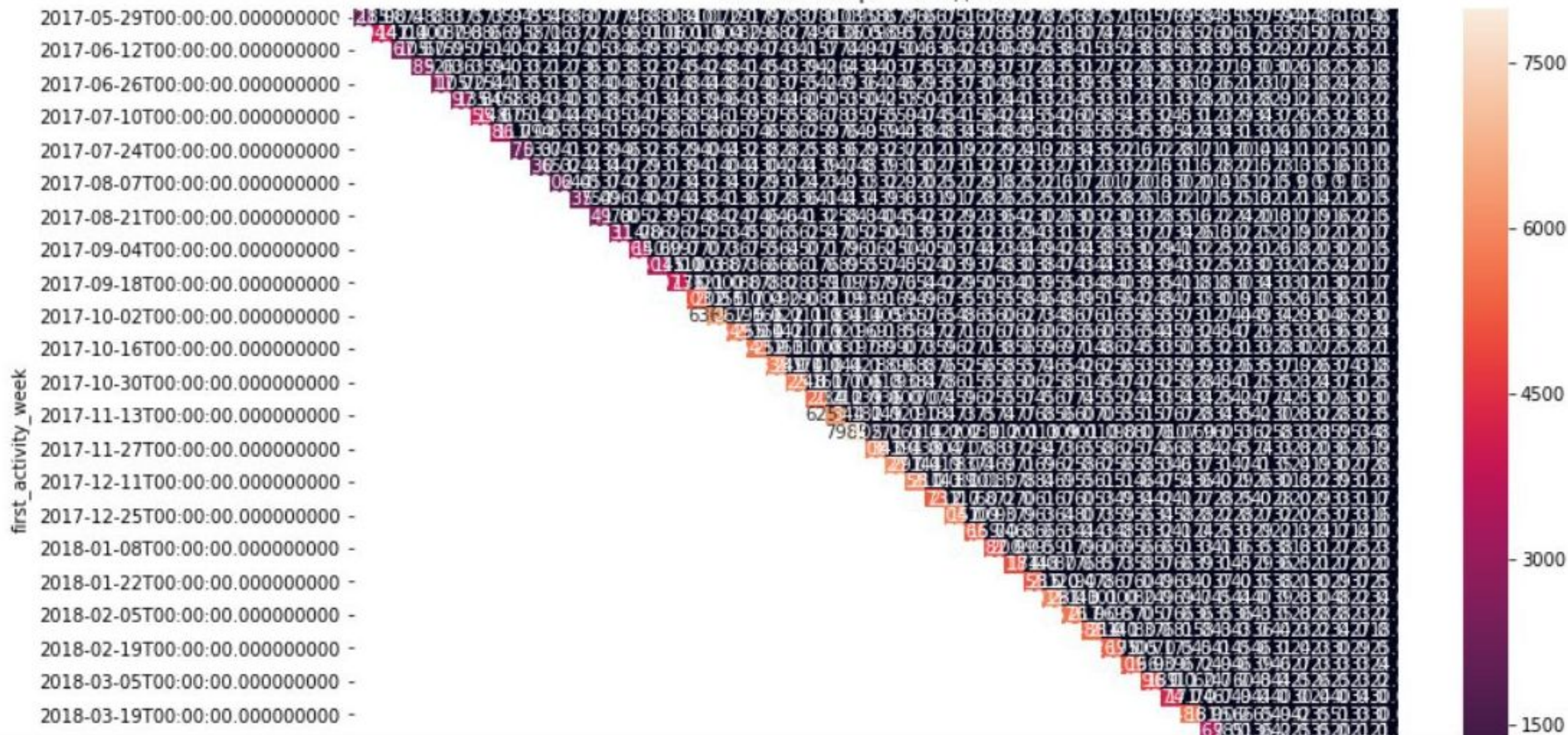


```
Out[608]: <matplotlib.axes._subplots.AxesSubplot at 0x7f34bc6197d0>
```



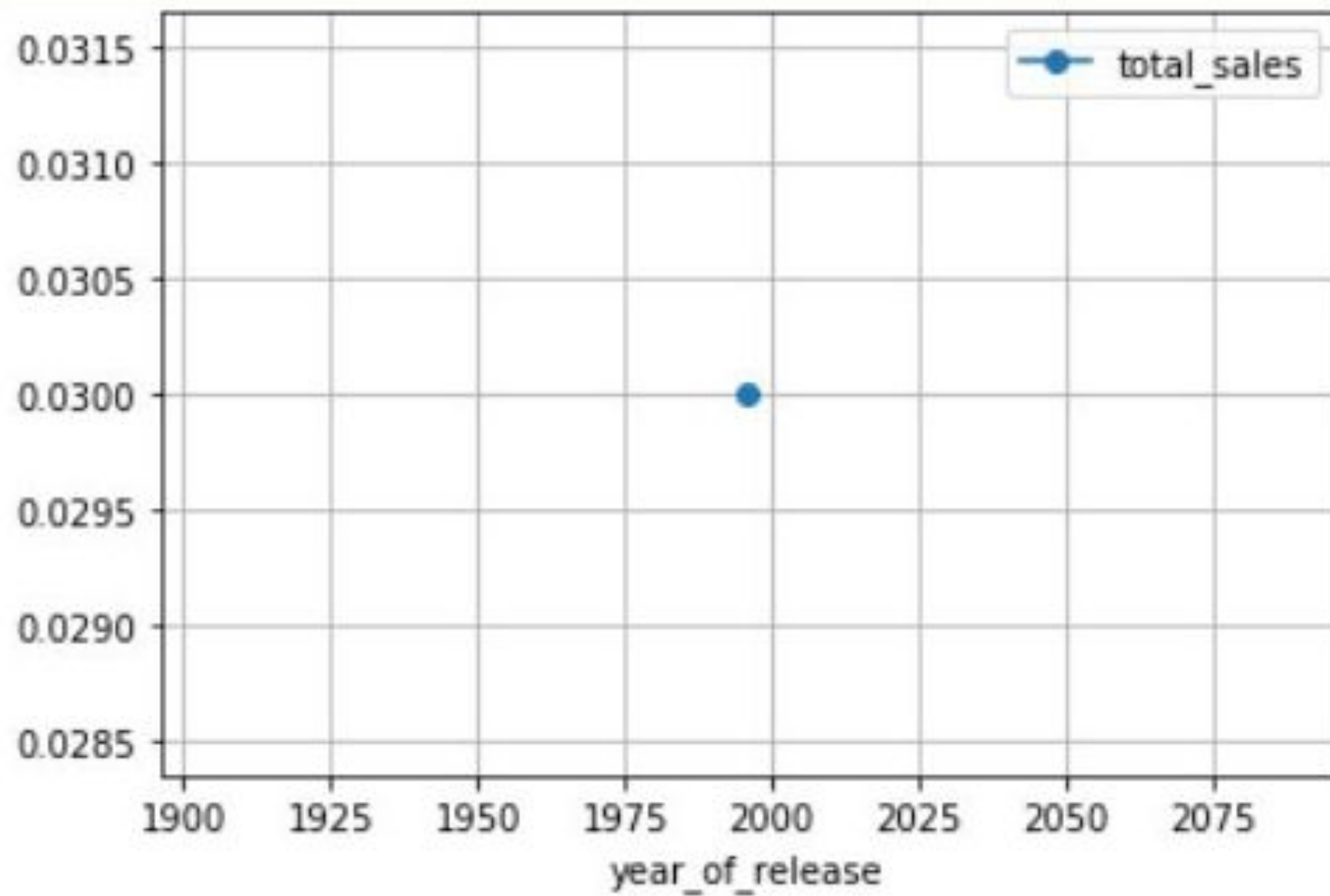


# Жизнь когорт по неделям

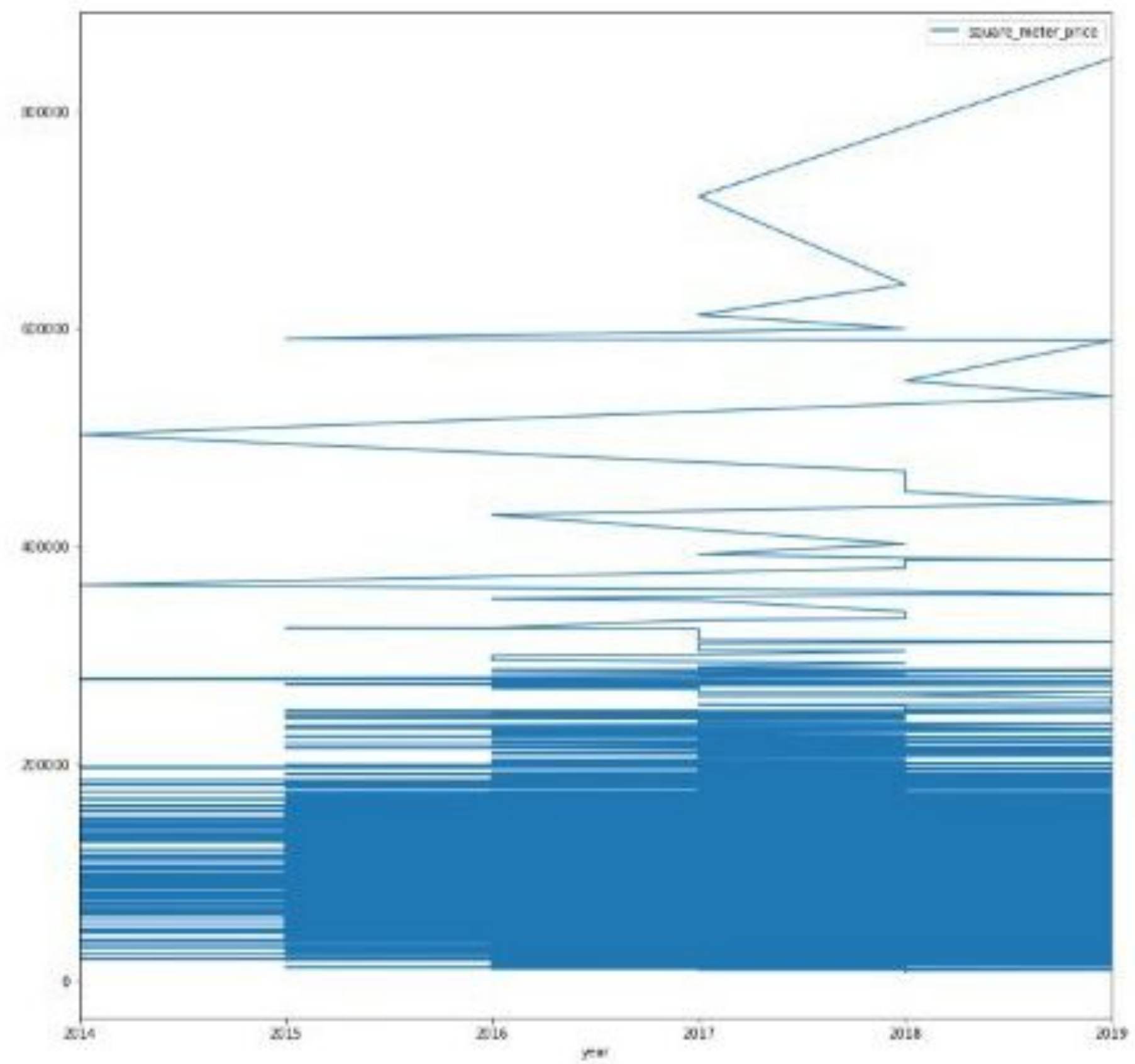












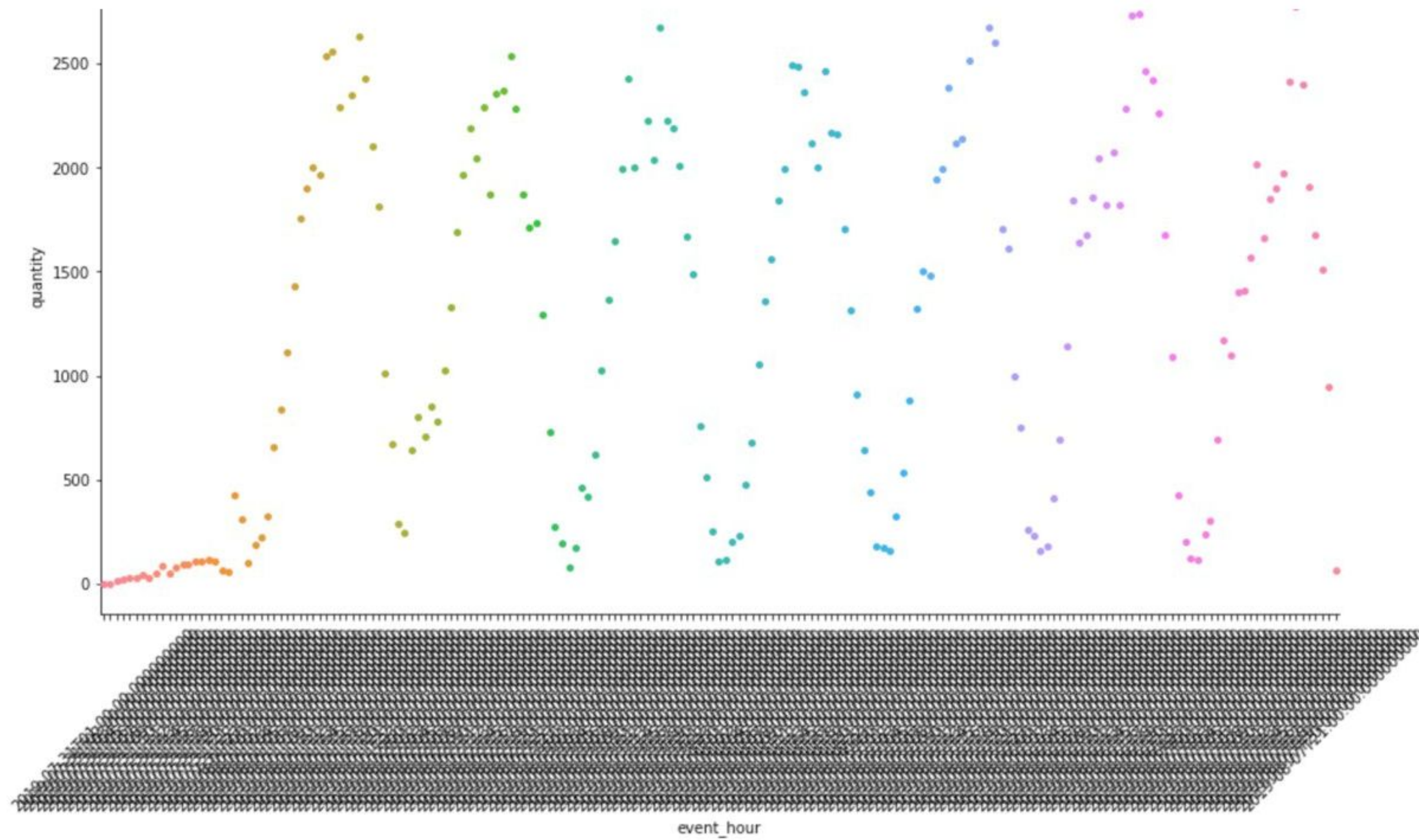
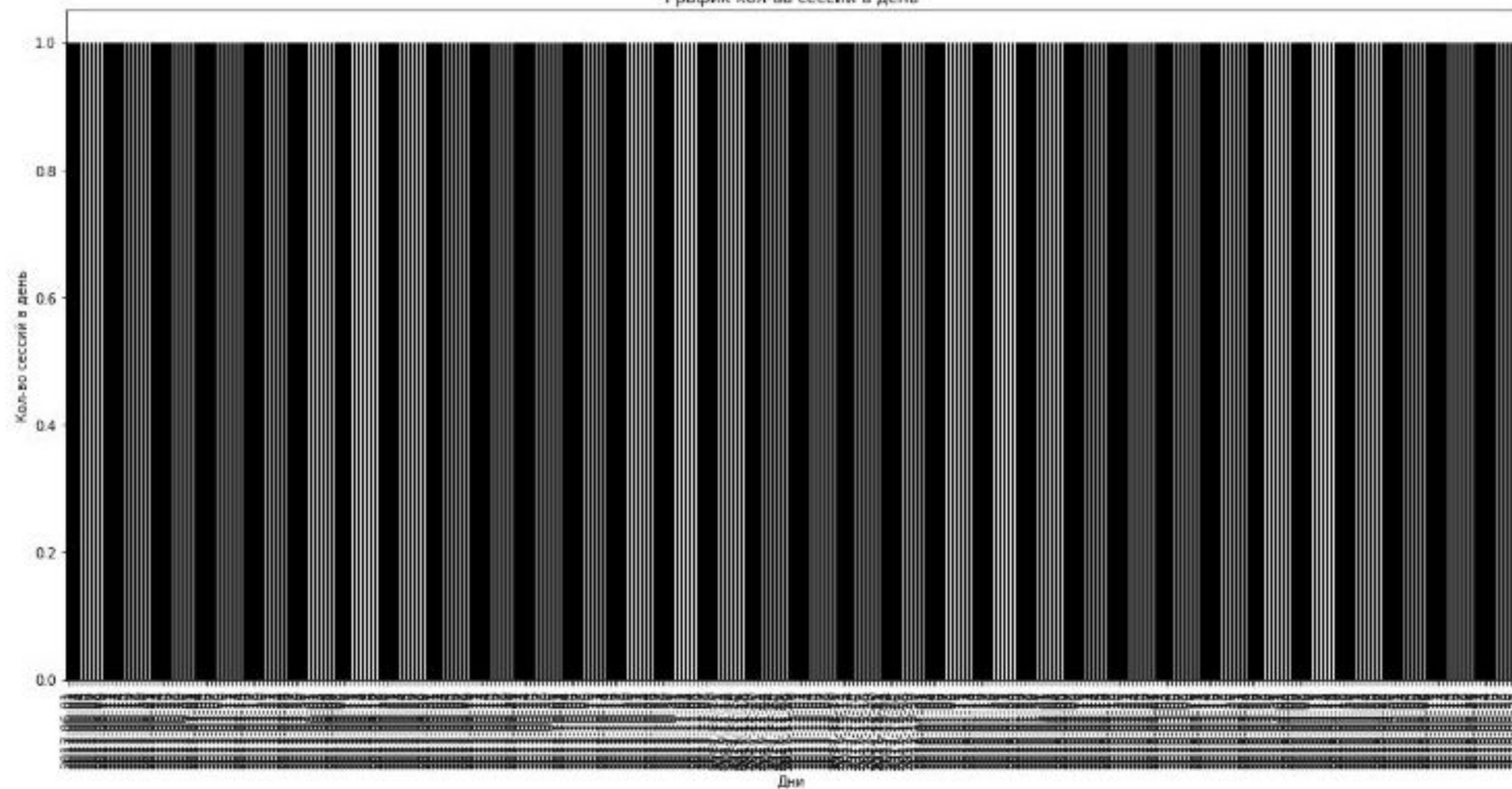
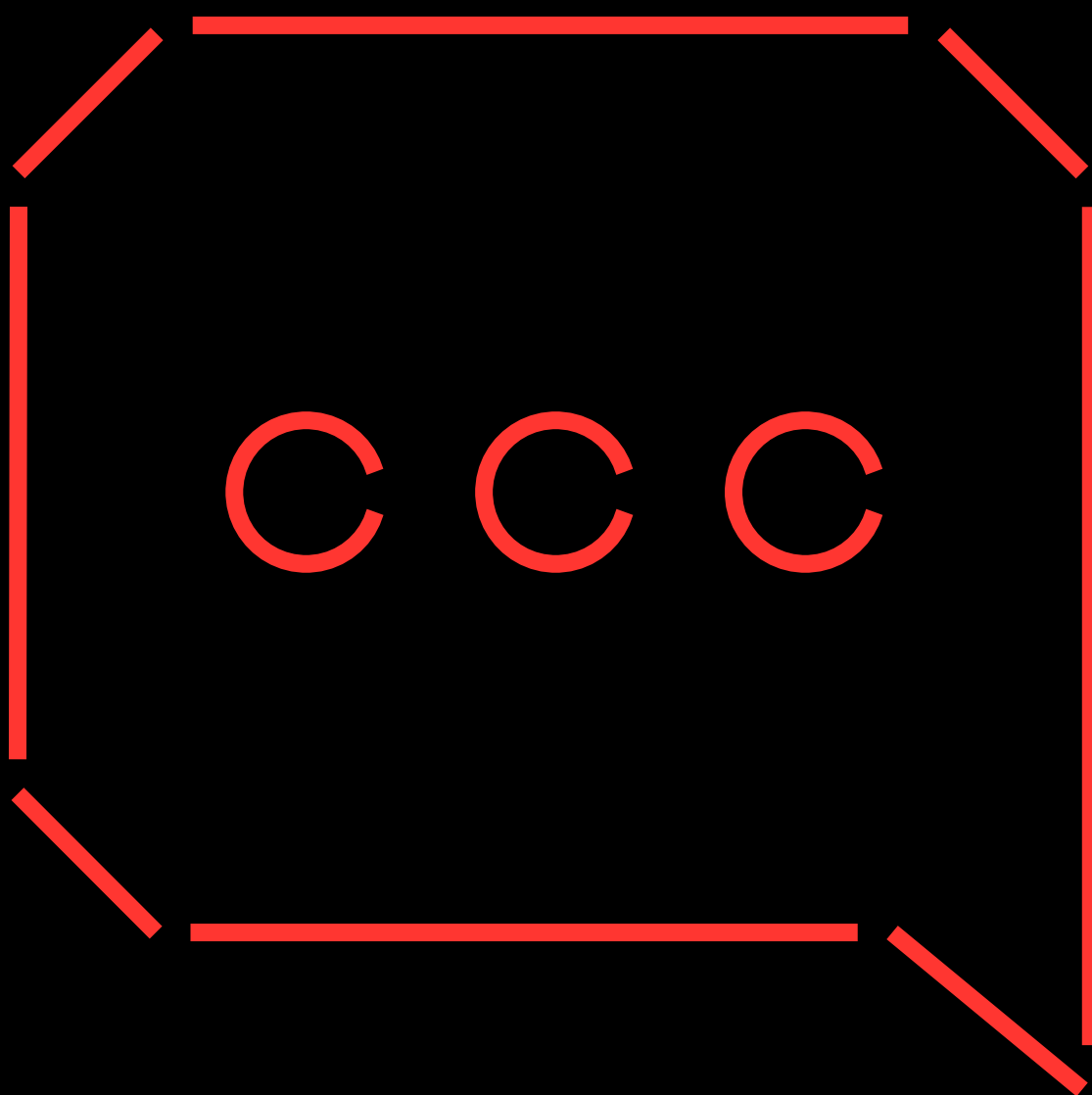


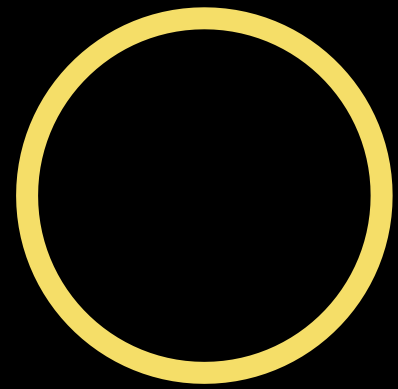
График кол-ва сессий в день





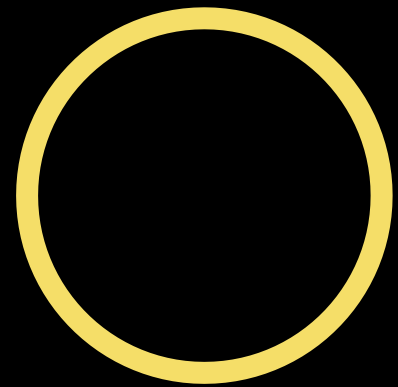
Вопросы?

# Вопросы



- Наталья Кока Подскажите пожалуйста, может быть существует шаблон для общего вывода, или насколько детальным он должен быть?

# Вопросы



- Дана Муратбек Когда будем свои проекты добавлять в репозиторий (github) или делиться им с помощью ссылки, важно ли, чтобы он был на английском - выводы и описание внутри тетрадки? Или это не важно? В дальнейшем реальные ревьюеры или рекрутеры будут смотреть только на код?

# Обратная связь

Напишите в чат впечатления от консультации