Проект на Python

В этом проекте вы можете использовать только следующие библиотеки:

* pandas
* matplotlib
* numpy
* seaborn
* requests
* plotly
* python-dotenv

Описание API

У API есть два маршрута со следующими параметрами:

* Запрос на данные по посещениям от даты START до даты END:
* /visits?begin=START&end=END
* Запрос на данные по регистрациям c даты START по дату END:
* /registrations?begin=START&end=END

Формат даты должен быть вида YYYY-MM-DD, например 2023-04-23

Запрос по пути /visits?begin=START&end=END вернет данные в формате JSON с массивом записей следующего вида:

* visit\_id - id посетителя
* platform - платформа с которой был заход на сайт: web, ios, android, bot
* user\_agent - User-Agent посещения
* datetime - дата посещения сайта

Расчет метрик

В этом шаге расчитаем конверсию визитов в регистрации. На выходе у вас должен получиться датафрейм со следующими полями:

* date\_group — дата
* platform — платформа: web, ios, android
* visits — визиты в дату
* registrations — регистрации в дату
* conversion — конверсия

Среди визитов есть боты — это поисковые и SEO-боты, которые регулярно сканируют страницы. Определить их можно по слову bot в User-Agent.

Сагрегируем данные по дате и платформам. Также сохраним полученный датафрейм в формате JSON.

Требования

* Данные должны быть отсортированы по дате, от ранних к более поздним
* Визиты ботов не должны влиять на расчет конверсии.

Задачи

Выполните Jupyter Notebook следующие задачи:

* Сгруппируйте данные визитов по датам и платформам
* Сгруппируйте также данные регистраций по датам и платформам
* Объедините датайфреймы, сделайте итоговый датафрейм с расчетом конверсии
* Сохраните датафрейм в формате JSON *conversion.json*
* Поля датафрейма:
  + date\_group - сагрегированный столбец дат
  + platform - платформа (android,web,ios)
  + visits - визиты за период date\_group
  + registrations - регистрации за период date\_group
  + conversion - конверсия по платформе

Добавляем рекламы

В этом шаге добавим данные по рекламным кампаниям. На выходе получим датафрейм со следующими полями:

* date\_group — дата
* visits — визиты в дату
* registrations — регистрации в дату
* cost — затраты на рекламу, 0 если не было затрат
* utm\_campaign — название рекламной кампании, none если не было в этот период рекламы

Сагрегируем данные по дате и сохраним их в JSON

Описание CSV

CSV-таблица содержит записи следующего вида:

* date — дата проведения кампании
* utm\_source — utm-источник: yandex, vk, google, youtube, tg
* utm\_medium — utm-медиум: cpc или social
* utm\_campaign — название кампании
* cost — затраты на рекламу

Например:

2023-03-01T09:16:57,google,cpc,virtual\_reality\_workshop,238

2023-03-02T12:48:25,google,cpc,virtual\_reality\_workshop,164

2023-04-25T11:30:16,google,cpc,full\_stack\_dev\_challenge,266

Требования

* Данные должны быть отсортированы по дате, от ранних к более поздним

Визуализация

В этом шаге визуализируем наши расчеты. Построим следующие графики в формате PNG:

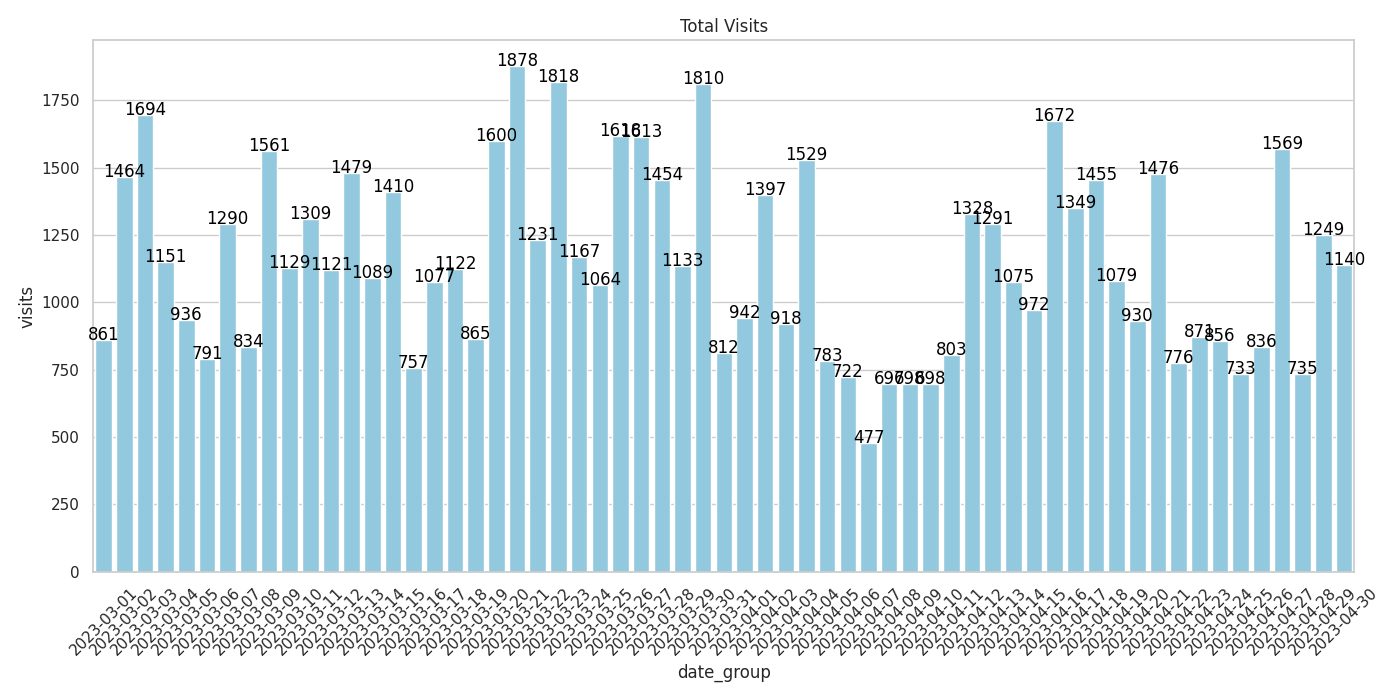
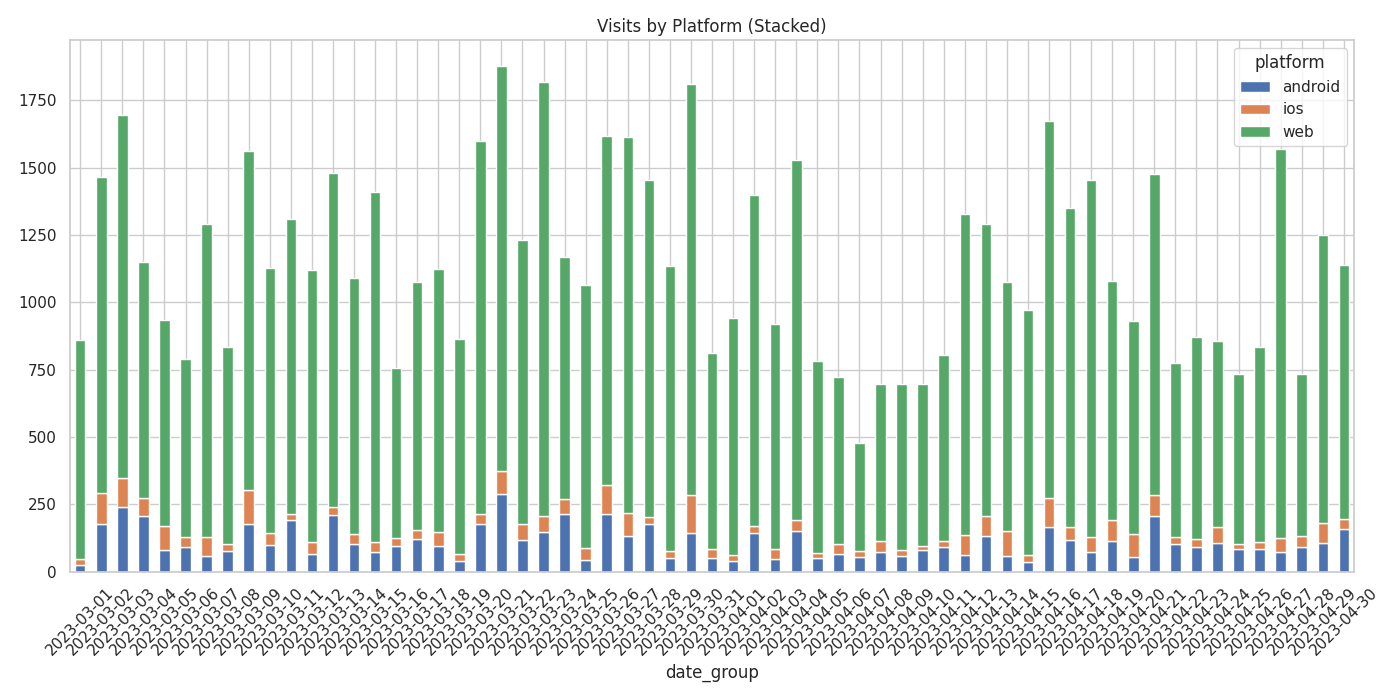
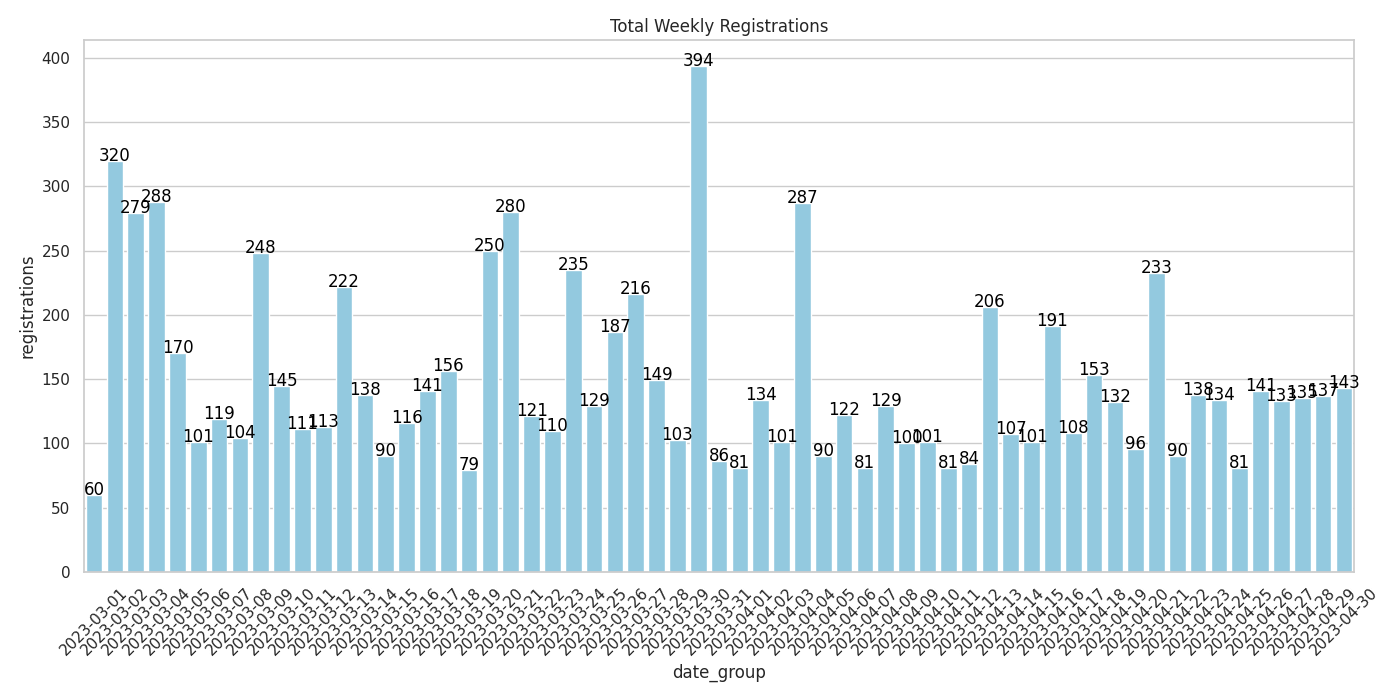
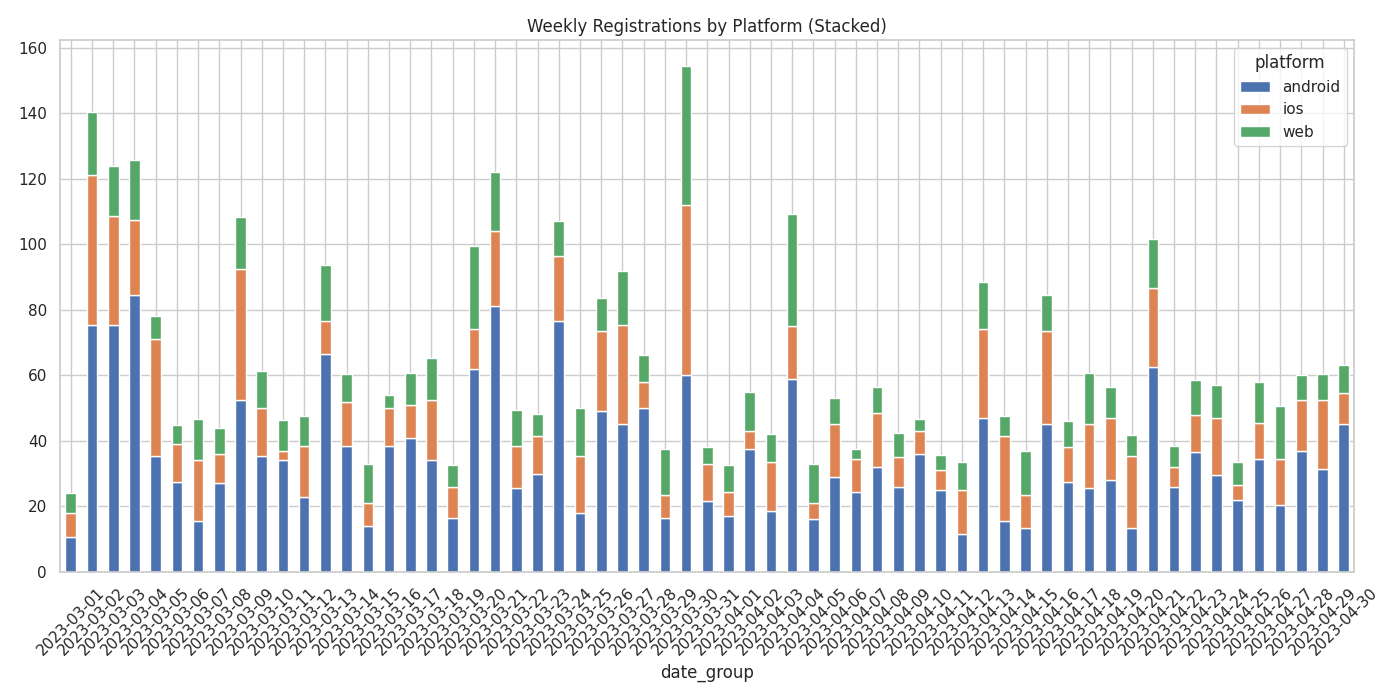
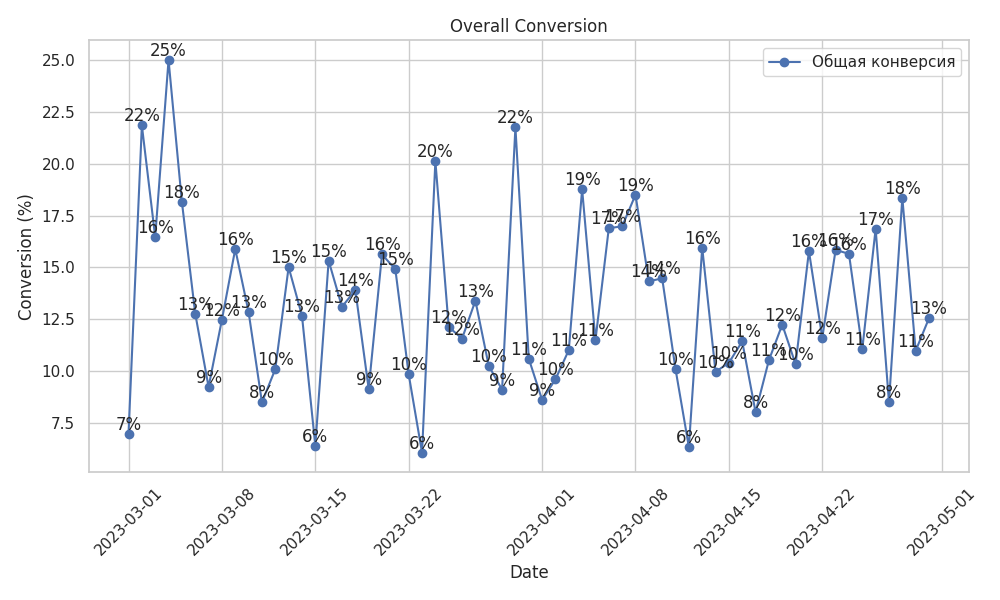
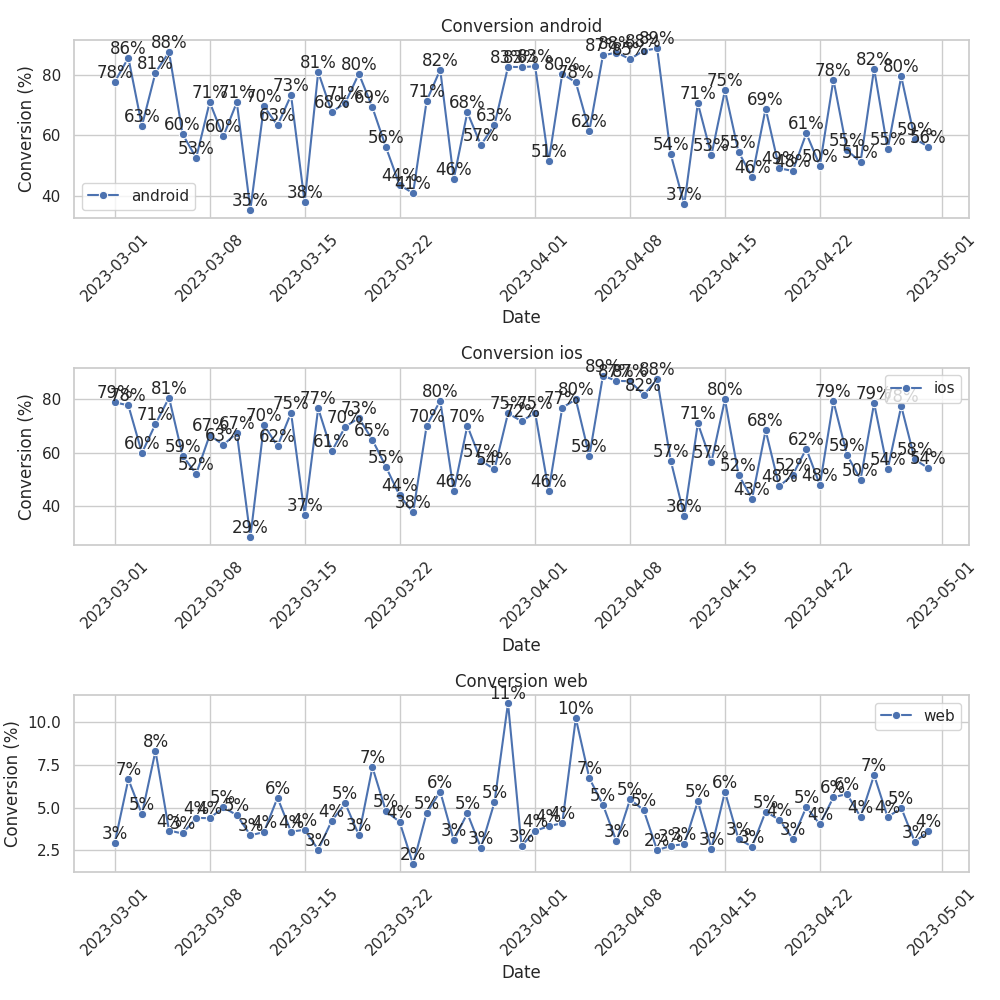
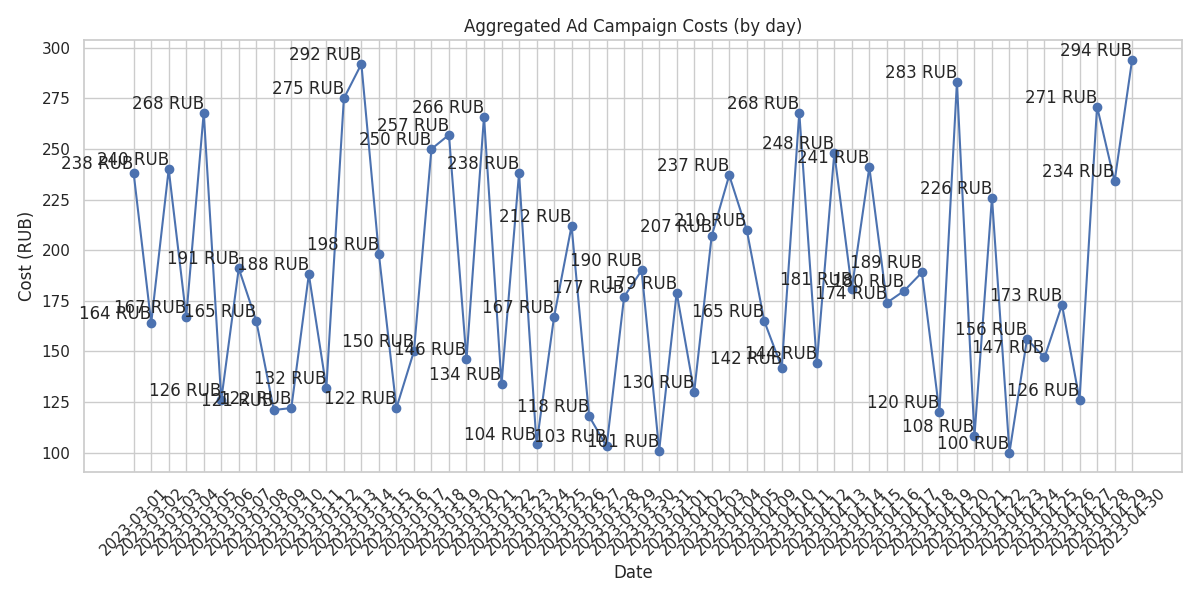
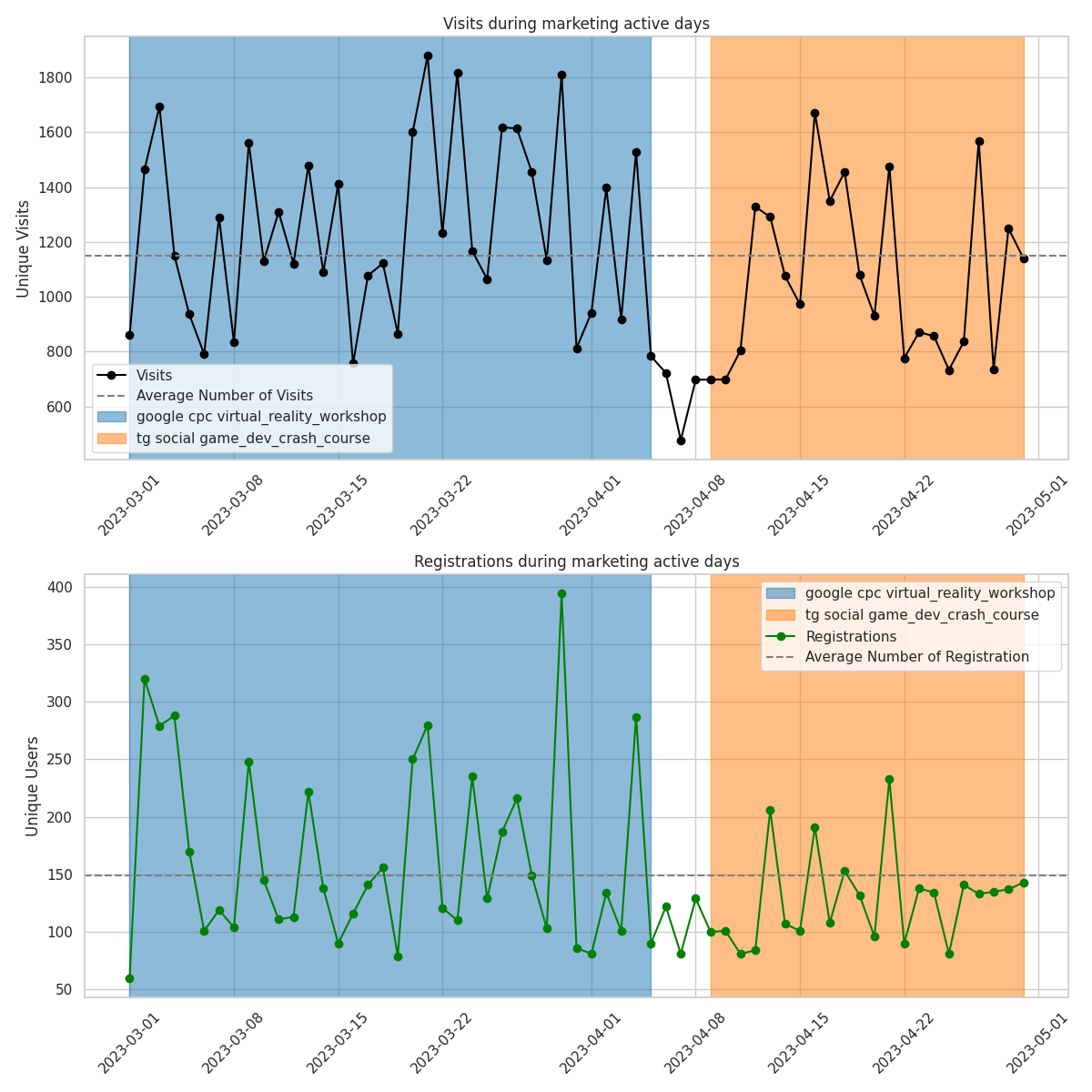
* Итоговые визиты
* Итоговые визиты с разбивкой по платформам: web, android, ios
* Итоговые регистрации
* Итоговые регистрации с разбивкой по платформе: web, android, ios
* Конверсия по каждой платформе
* Средняя конверсия
* Стоимости реклам
* Визиты за весь период с цветовым выделением рекламной кампании
* Регистрации за весь период с цветовым выделением рекламной кампании

Задачи

Выполните Jupyter Notebook следующие задачи:

* Установите библиотеки для визуализации
* Постройте требуемые графики
* Сохраните их в PNG в директорию *./charts*, например как plt.savefig('./charts/registrations\_by\_platform.png')

Примеры графиков

* Итоговые визиты 
* Итоговые визиты с разбивкой по платформам 
* Итоговые регистрации 
* Итоговые регистрации с разбивкой по платформе 
* Итоговые конверсии 
* Конверсия по каждой платформе 
* Стоимости реклам 
* Визиты и регистрации с выделением рекламных кампаний 

Требования к проекту

Проверьте свой код по пунктам указанным ниже. Если необходимо, внесите исправления в проект:

1. Репозиторий содержит Jupyter Notebook charts\_project.ipynb со всеми расчетами
2. Notebook работает с переменными окружения для выполнения запросов
3. Notebook читает файл ads.csv из директории проекта
4. Notebook сохраняет в текущую директорию JSON файлы с именами conversion.json и ads.json
5. Notebook сохраняет в директорию charts графики (имена могут быть любыми)
6. Презентация заполнена и лежит в формате PDF
7. Графики в презентации содержат числовые подписи для удобного анализа
8. Презентация содержит выводы аналитика

Презентация и выводы

Создадим письменную презентацию с анализом и рекомендациями на основе полученных результатов. Пример вопросов:

1. Увеличиваются ли заходы и регистрации с запуском рекламы
2. Есть ли периоды, где случались просадки в заходах?
3. Есть ли период, где случались просадки в регистрациях?
4. Связаны ли просадки с увеличением/уменьшением рекламы / рекламного бюджета / сменой креатива?
5. Если просели регистрации, связано ли это с рекламным трафиком, или у нас сломался какой-то из способов регистраций?

Задачи

* Проверьте, что Jupyter Notebook содержит необходимые скрипты, визуализацию
* Проверьте, что Python скрипт работает корректно
* Подготовьте презентацию с выводами и графиками