Перед вами данные взаимодействий с рекламными объявлениями на некоторой площадке за 6 дней. И таблица с характеристиками рекламных клиентов (тех, кто разместил эти объявления).

Задачи

1. Разберемся с распределением количества показов и кликов. Посчитайте среднее количество показов и среднее количество кликов на объявления за весь период (округлите до целых).
2. Нарисуйте график распределения показов на объявление за весь период.
3. Давайте посчитаем скользящее среднее показов с окном 2. Какое значение скользящего среднего получим за 6 апреля 2019 года (ответ округлите до целых)?
4. Скользящее среднее часто используется для поиска аномалий в данных. Давайте попробуем нанести на один график значения арифметического среднего по дням и скользящего среднего количества показов. В какой день наблюдается наибольшая разница по модулю между арифметическим средним и скользящим средним? Дни, в которых скользящее среднее равно NaN, не учитываем.
5. \*Напишите функцию, которая найдет проблемное объявление (с наибольшим/наименьшим количеством показов) в день, в который была замечена самая большая по модулю аномалия.
6. Теперь подгрузим данные по рекламным клиентам и найдем среднее количество дней от даты создания рекламного клиента и первым запуском рекламного объявления этим клиентом.
7. Вычислите конверсию из создания рекламного клиента в запуск первой рекламы в течение не более 365 дней. Ответ дайте в процентах и округлите до сотых. (Фильтровать нужно по значению в формате pd.Timedelta(365, unit='d'))
8. Давайте разобъем наших клиентов по промежуткам от создания до запуска рекламного объявления, равным 30. Определите, сколько уникальных клиентов запустили свое первое объявление в первый месяц своего существования (от 0 до 30 дней). Список промежутков для метода pd.cut – [0, 30, 90, 180, 365]
9. А теперь – выведем на интерактивный график эти категории с количеством уникальных клиентов в них.

Описание данных

ads\_data.csv – данные взаимодействий с рекламными объявлениями

* date – дата
* time – время
* event – действие (просмотр/клик)
* platform – платформа
* ad\_id – id объявления
* client\_union\_id – id рекламного кабинета
* campaign\_union\_id – id рекламной кампании
* ad\_cost\_type – тип оплаты
* ad\_cost – цена
* has\_video – есть ли видео
* target\_audience\_count – размер аудитории

ads\_clients\_data.csv – характеристики рекламных клиентов

* date – дата
* client\_union\_id – id рекламного кабинета
* community\_id – id сообщества
* create\_date – дата создания рекламного клиента

Данные находятся на JupyterHub, можно скачать [архив](https://stepik.org/media/attachments/course/72204/ads.zip)

Для начала посчитаем среднее количество разных рекламных событий (показы и клики) по объявлениям. Тонкий момент: помните, что некоторые показываемые объявления не кликались, и при группировке в таких случаях не будет строк с кликами, а для подсчёта среднего по объявлениям их нужно учитывать. Округлите ответы до целых чисел.

Могут понадобиться:

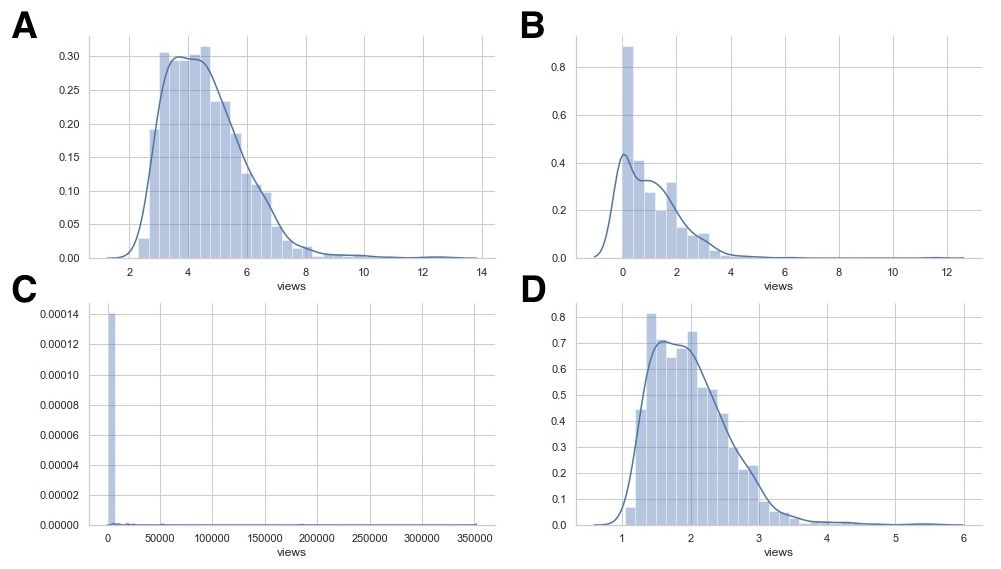
* pivot()
* fillna()

[Подсказки](https://www.notion.so/6-1-PYTHON-30032ddef90549ef90a78ee1cf74254a)

Давайте посмотрим на распределение показов объявлений!

* По оси x – число показов
* По оси y – доля объявлений с такими показами

Прологарифмируйте число показов, а затем постройте такой график, чтобы можно было понять форму распределения. Выберите соответствующий график:



Для такого же отображения используйте эти настройки

sns.set(

font\_scale=2,

style="whitegrid",

rc={'figure.figsize':(20,7)}

)

Могут пригодиться:

* [sns.distplot](https://lab.karpov.courses/learning/212/module/2218/lesson/20494/59055/277425/)
* [np.log](https://numpy.org/doc/1.18/reference/generated/numpy.log.html)

[Подсказки](https://www.notion.so/6-2-PYTHON-380d82369d8c4d8cafe320ba93d9fffb)

Теперь нужно найти среднее количество показов на 1 объявление (ad\_id) по дням, не учитывая объявления, у которых не было показов (обычное среднее). После этого посчитайте скользящее среднее по полученному результату с окном в 2 элемента.

**Вопрос:** Какое значение скользящего среднего получим за 6 апреля 2019 года? Округлите ответ до целого числа.

Могут пригодиться:

* [pivot()](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pivot_table.html#pandas.DataFrame.pivot_table)
* [скользящее среднее](https://lab.karpov.courses/learning/212/module/2218/lesson/20494/59055/277416/)

[Подсказки](https://www.notion.so/6-3-PYTHON-32926db7a2dc4bcba7e805c6588c0963)

Скользящее среднее часто используется для поиска аномалий в данных. Давайте попробуем нанести на один график значения просто среднего количества показов по дням и скользящего среднего на основе этих данных (которое мы получили в прошлом задании).

Посчитайте, в какой день наблюдается наибольшая разница по модулю между арифметическим средним и скользящим средним? Дни, в которых скользящее среднее равно NaN, не учитываем. Запишите ответ в формате yyyy-MM-dd

Пример ответа:

2020-12-31

Могут пригодиться:

* sns.lineplot()
* [скользящее среднее](https://lab.karpov.courses/learning/212/module/2218/lesson/20494/59055/277416/)
* [значение по модулю](https://docs.python.org/3/library/functions.html#abs)

[Подсказки](https://www.notion.so/6-4-PYTHON-966eb9ddfa4a4b56ad42d0c92469d55c)

Объедините данные рекламы с данными о рекламных клиентах и найдите среднее количество дней от даты создания рекламного клиента (create\_date) и первым запуском рекламного объявления этим клиентом (date).

В качестве ответа укажите среднее число дней.

[Подсказки](https://www.notion.so/6-6-PYTHON-3661d02d16de447cb95ec573ba8b589a)

💀 Сложное задание! 💀

Давайте вычислим конверсию из создания рекламного кабинета в запуск первой рекламы в течение не более 365 дней.

Конверсия – отношение числа клиентов, которые совершили целевое действие (создали рекламу), к общему числу клиентов. Например, 1000 клиентов создали кабинет, но лишь 50 из них запустили рекламу. В таком случае конверсия будет равна

100050​∗100=5%

Чтобы отобрать данные за нужный промежуток, можно воспользоваться сравнением времени до создания первой рекламы с pd.Timedelta(365, unit='day')

Ответ дайте в процентах и округлите до сотых. Пример ответа:

34.69

**NB!** У нас могут быть люди, которые не сделали рекламу вообще. Не забудьте их учесть :)

[Подсказки](https://www.notion.so/6-7-PYTHON-270fffcae91e4983980afa57ccdfec3a)

Давайте разобъем наших клиентов по промежуткам от создания рекламного кабинета до запуска первого рекламного объявления.

Определите, сколько уникальных клиентов запустили свое первое объявление в первый месяц своего существования (от 0 до 30 дней). Список промежутков дней для метода pd.cut() – [0, 30, 90, 180, 365]. Создать временные промежутки можно с помощью [pd.to\_timedelta()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.to_timedelta.html" \t "_blank).

Могут пригодиться:

* [pd.cut()](https://lab.karpov.courses/learning/212/module/2218/lesson/20494/59055/277424/)
* value\_counts()

[Подсказки](https://www.notion.so/6-8-PYTHON-1534f42a6cdf4151901ee30b0166e190)

И последнее задание: постройте интерактивный барплот, на котором будут показаны категории с количеством уникальных клиентов в них (то, что у вас получилось на прошлом степе, по запуску первого рекламного объявления). Расположите их так, чтобы внизу была самая малочисленная, а наверху – самая большая.

[Подсказки](https://www.notion.so/6-9-PYTHON-5ee9f13855534578ab6c26f7c14e8fd0)

## Дополнительный проект

В данном проекте вы поработаете с довольно нетипичными для аналитика данными: gff файлом, который описывает геном мухи (описание формата можно найти [здесь](http://gmod.org/wiki/GFF3)). При его выполнении вам может понадобиться часть конспекта про работу со строковыми данными, а именно – про строковые методы серий.

### **Задачи**

1. Выделите из колонки attributes значение атрибута Parent. Т.е. если там записано Parent=x, то нам нужен x
2. Выясните, какое из этих значений является самым частым
3. Постройте распределение встречаемости значений в столбце type

Данные можно достать [отсюда](https://stepik.org/media/attachments/course/72204/gff.tsv)

Какое значение атрибута Parent самое частое?

*Note:* Если вы хотите решить это задание с использование регулярных выражений, то с теорией можно познакомиться [тут](https://www.youtube.com/watch?v=oFaf3NcMudA) и [тут](https://www.youtube.com/watch?v=QBaU6CMeUUE).

[Подсказки](https://www.notion.so/6-1-PYTHON-343520c537c64750a59a16e4fa8e4393%22)

Какое самое частое значение в колонке type?

[Подсказки](https://www.notion.so/6-2-PYTHON-701037dd393749d1bfc5cbfc89a096fd)

exon

CDS

five\_prime\_UTR

mRNA

protein

three\_prime\_UTR

Небольшой пример использования re:

<https://youtu.be/LrzYD3Bin68>

## Дополнительные материалы по regexp

* [Документация по re](https://docs.python.org/3/library/re.html)
* [howto по re](https://docs.python.org/3/howto/regex.html)
* [Звезда и плюс Клини](https://en.wikipedia.org/wiki/Kleene_star)
* [Статья о re на Habr](https://habr.com/ru/post/349860/)
* [О внутренностях re](https://habr.com/ru/post/166777/)
* [Ещё раз об использовании re в неподходящих задачах](https://stackoverflow.com/questions/1732348/regex-match-open-tags-except-xhtml-self-contained-tags?noredirect=1&lq=1)
* [Сайт для проверки паттернов](https://regexr.com/)
* [Другой сайт для проверки паттернов](https://regex101.com/)
* [Крупный источник информации о regexp](https://www.regular-expressions.info/)
* [Ещё один источник](http://www.rexegg.com/)
* [Демонстрация regexp от хейтеров regexp](https://ihateregex.io/)
* [Генерация паттерна из текста с примерами](http://regex.inginf.units.it/)