**Тестовое задание Python**

Тестовое задание необходимо выполнить в Google Colab.

1. Подгрузить датафрейм из предоставленного файла data.feather.
2. Провести базовый EDA - посчитать среднее, медиану, стандартное отклонение, квартили для столбцов цен (\*\_price\_1) и объёмов (\*\_vol\_1) - отдельно для каждого столбца. Визуализировать боксплотами. Установить, были ли выбросы по ценам (например, через межквартильный размах).
3. Визуализировать распределения цен и объёмов любым способом.
4. Рассчитать среднюю цену инструмента (bid\_price+ask\_price)/2 для всех точек и по ней рассчитать за для всех точек взвешенную по времени экспоненциальную скользящую среднюю по нижеописанному принципу. Замерить время выполнения (можно просто через %%timeit), попробовать достичь времени выполнения менее 1 с.
   1. Инициализируемся средней ценой для первой строки датафрейма
   2. Если разница по времени со следующей строчкой превышает 1 секунду, то условно считаем её равной одной секунде (параметр в формуле ниже)
   3. Очередное значение EMA = прошлое значение EMA + (очередная средняя цена - прошлое значение EMA)\*(разница по времени до предыдущего значения)/(10000 секунд):
5. Ввести столбец position, инициализировать нулями
6. Рассчитать значения в этом столбце по следующей логике:
   1. первые 10000 секунд - 0
   2. далее, если средняя цена в очередной строке больше EMA более чем на 50 и значение в предыдущей строке pos меньше 10 -> +1
   3. Если текущая средняя цена < EMA более чем на 50 и предыдущее значение pos > -10 -> -1
   4. Иначе значение не меняется

Постараться уложиться в 20 секунд.

1. Визуализировать результаты.

**Тестовое задание SQL**

Предоставленная база SQLITE содержит 2 таблицы - данные по депозитам на условных торговых счетах и таблицу соответствия счетов торговым роботам, с периодами активности этих роботов.

В таблице депозитов есть поле timestamp - кратное минуте, записи в рамках одной минуты для разных счетов имеют одинаковый timestamp.

У каждого робота 2 счёта, и они могут меняться (в этом случае период активности по счёту завершается, и в таблице открывается новый период с новым счётом).

Необходимо написать SQL запросы:

1. Вывести сумму депозитов по счетам для каждого робота за его периоды активности.
2. Рассчитать для каждого робота доход или убыток (в валюте) с начала записанных в таблице депозитов данных до каждого timestamp’a
3. Для каждого робота вывести таблицу, в которой отражались бы в разных столбцах депозиты по его счетам (для всех таймстемпов), а также столбец с относительной разницей депозитов (т.е. |депозит1-депозит2|/(депозит1 + депозит2)) и флаг, если такая разница больше 10%.
4. Для каждого таймстемпа вывести название робота, на котором больше всего денег.