**Финальная работа по курсу: «Аналитик данных с нуля 2.0»**

Выполнила: Манякова Елизавета

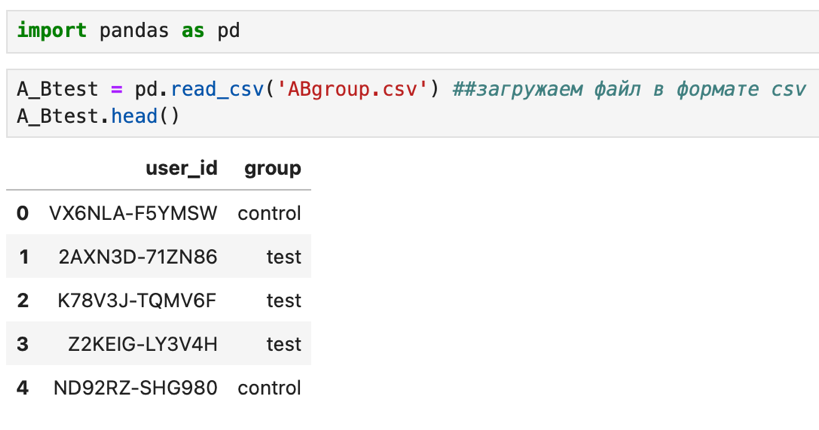
Цель работы: определить, как проведение акции повлияло на прибыль компании: если игроки, участвовавшие в акции, принесли больше денег, чем игроки, у которых акции не было, то стоит повторять акцию уже на всех игроках.

Для решения поставленной цели стоит проанализировать имеющиеся файлы с информацией о распределении игроков по группам теста, о тратах внутриигровой валюты, об обнаруженных читерах, о платежах и о игровых платформах. Нам известно, что присутствуют читеры, то есть те игроки, которые с помощью взлома игры начисляют себе большие объемы внутриигровой валюты. Явные читеры уже представлены, поэтому для обнаружения скрытых, еще не известных, читеров, необходимо проанализировать действия известных. Данные о читерах стоит удалить из общего списка игроков, так как их результаты могут повлиять на выводы.

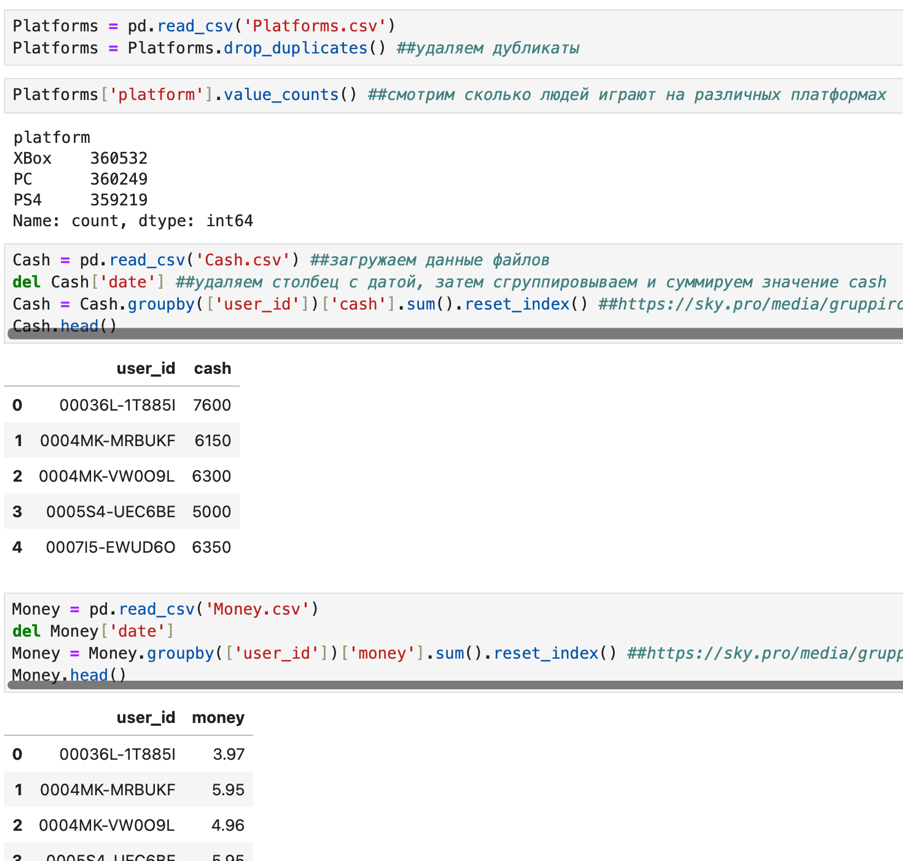
Источником анализа выбран Python, так как данная программа имеет большое количество открытых библиотек, позволяющих обработать и визуализировать полученные результаты.

Перед началом работы необходимо загрузить исходные таблицы с данными с помощью библиотеки pandas [1]:

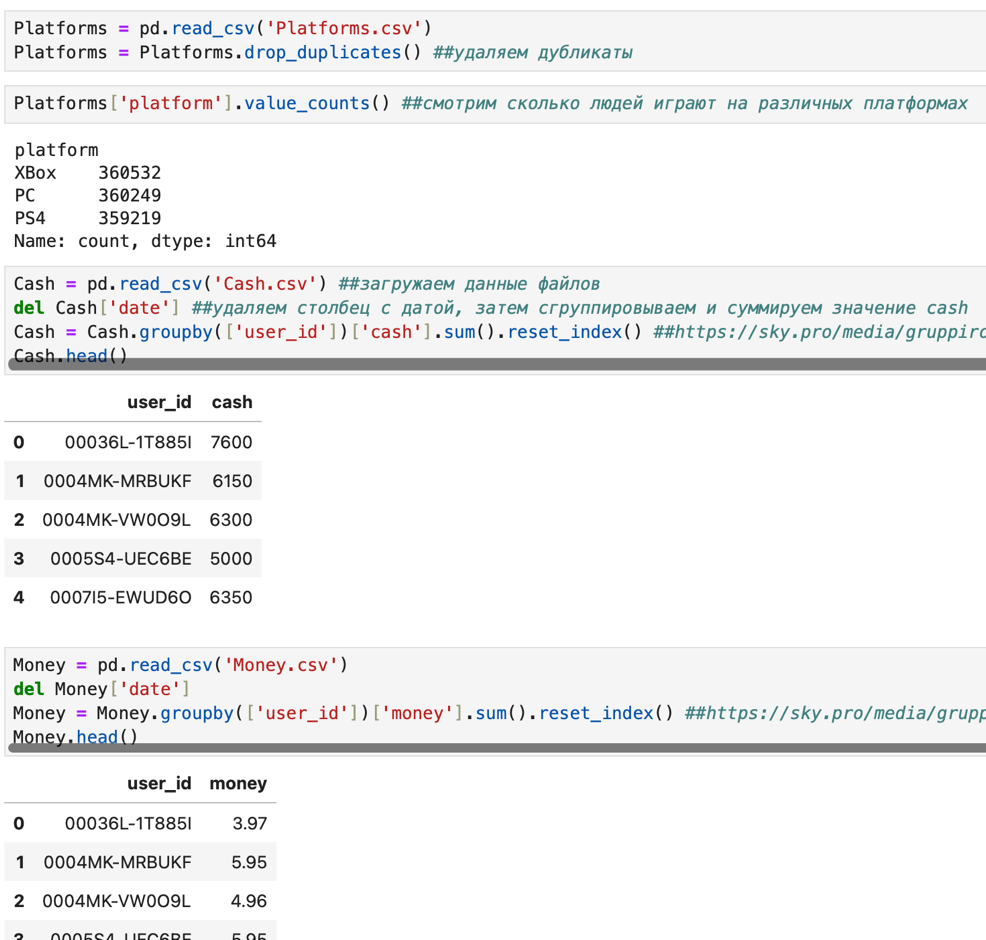
1. Загружаем данные из файла ‘ABgroup.csv’ формата csv, просматриваем струкртуру файла.



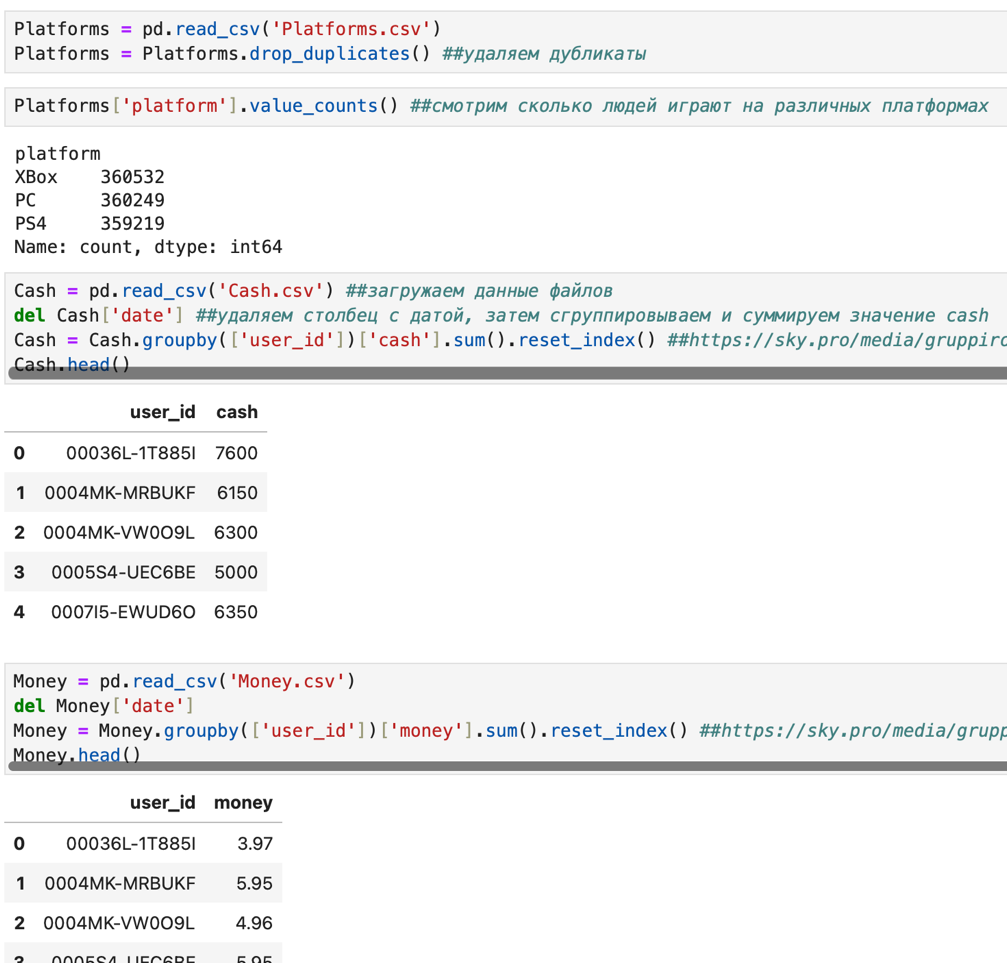
1. Загружаем данные из файла ‘Platforms.csv’ и сразу удаляем дубликаты, как так сбор информации происходит за 8 дней (тоже самое происходит и во всех других файлах).



1. Загружаем данные из файла ‘Cash.csv’, удаляем столбец date, группируем по user\_id [2] и ссумируем все значения cash за 8 дней сбора информации.



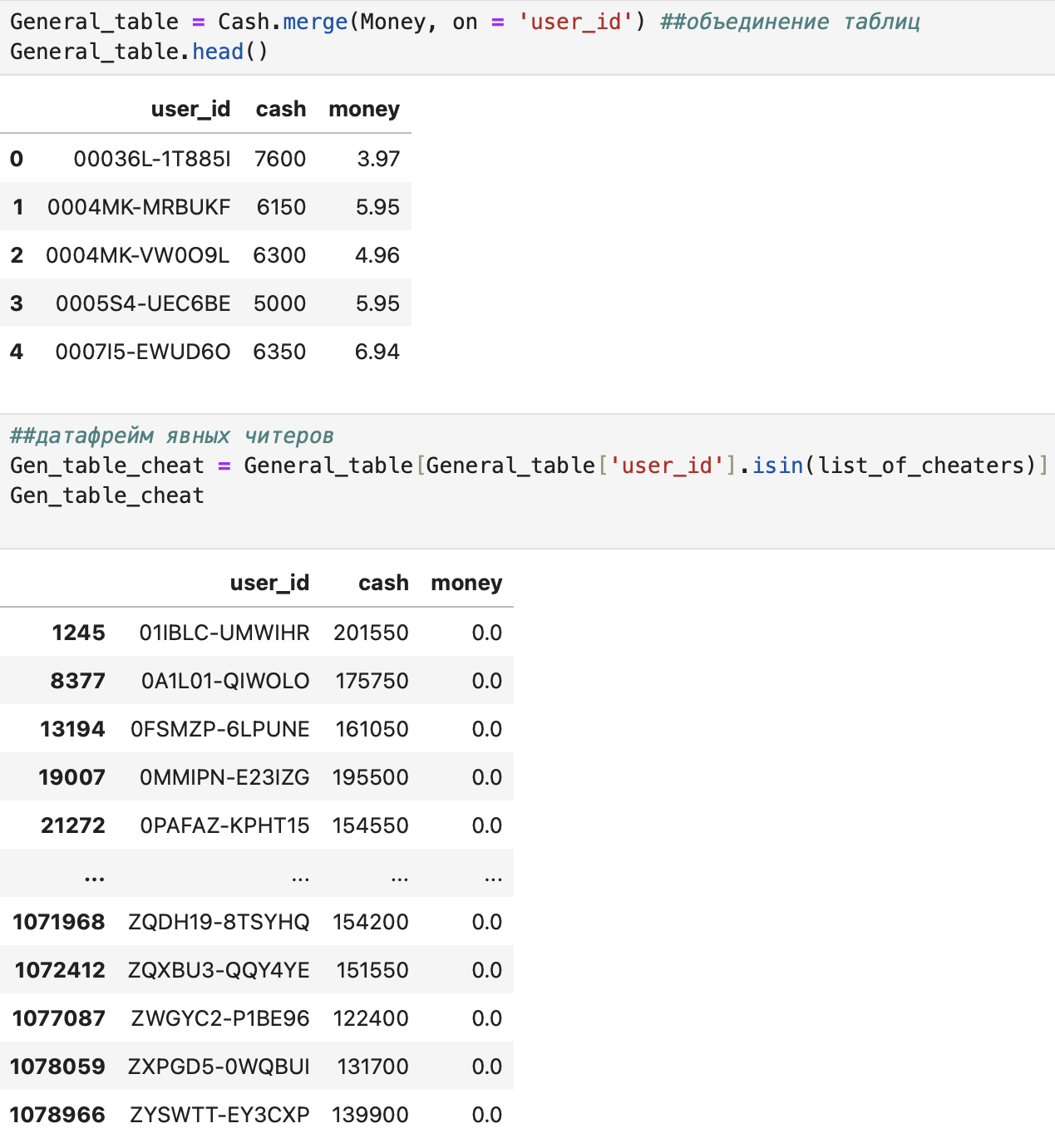
1. Загружаем данные из файла ‘Money.csv’, проделываем все тоже самое что и с файлом ‘Cash.csv’.



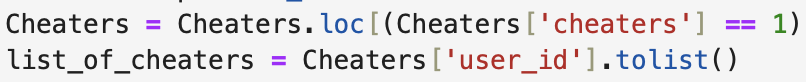
1. Загружаем данные из файла ‘Cheaters.csv’ с известными читерами. В колонке cheaters указано одно из двух значений: 0 и 1 (явный читер)

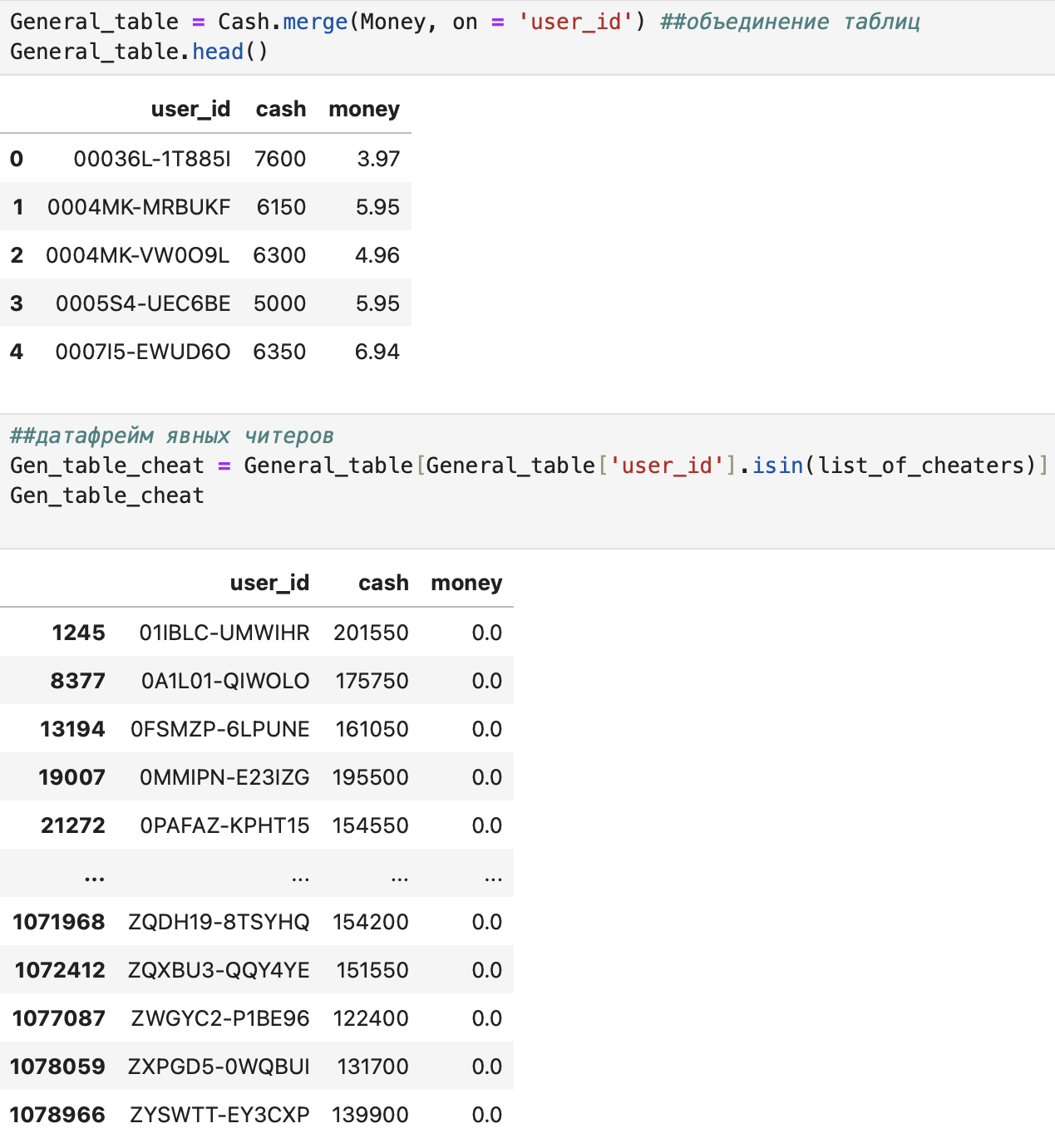


Объединим таблицы Cash и Money по параметру user\_id:



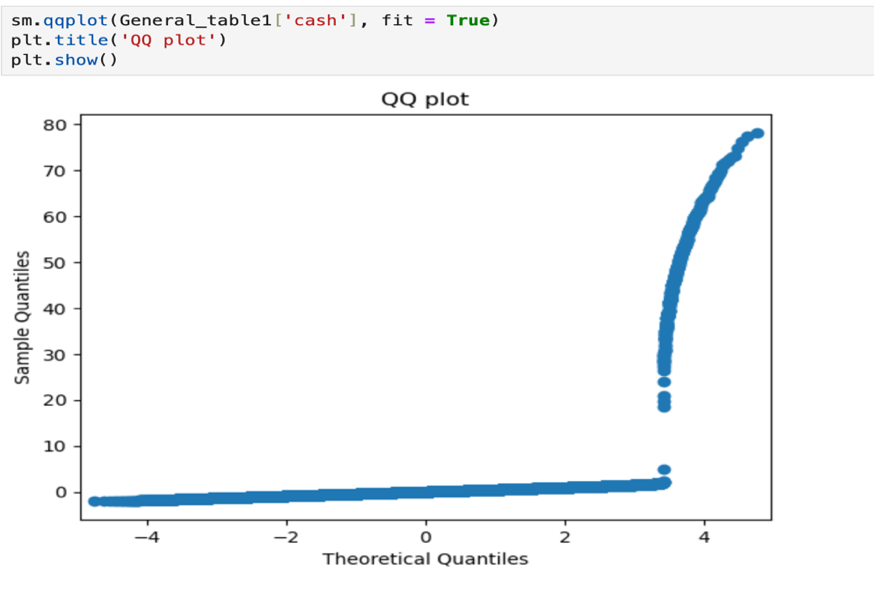
Для выявления скрытых читеров, необходимо проанализировать поведение известных. Создадим отдельный Датафрейм с явными читерами по параметру Cheaters[‘cheaters’] == 1, через список user\_id известных читеров, после чего из объединенной таблицы (Cash и Money) выбрать только читеров для дальнейшего анализа их поведения [3]. Выявлено 353 явных читеров.



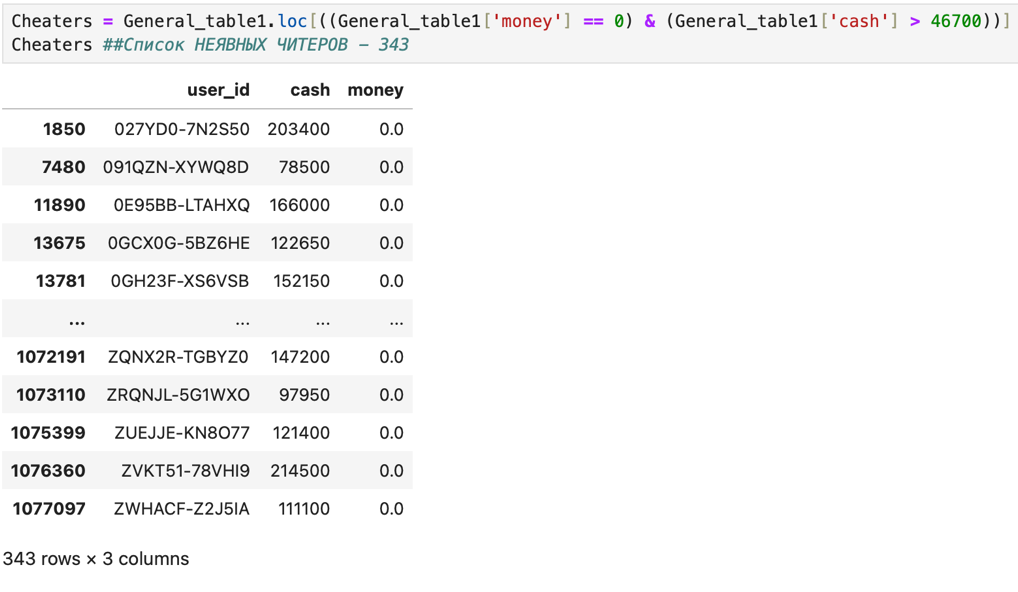


Замечена следующая закономерность: в ходе акции читеры в графе money имеют значение 0 (money = 0), то есть они не покупали внутриигровую валюту, но в тоже время, тратили cash (внутриигровую валюты), значение которого намного выше среднего (min = 46 700, max = 253 150, среднее = 146 807, 932).

Из графика QQ plot [4] после удаления только Явных читеров замечено, что здесь не нормальное распределение. Это можно заметить по тому факту, что прямая не проходит под углом 45 градусов. Здесь видим резкий скачок, это говорит про то, что в данной выборке присутствуют ЧИТЕРЫ.

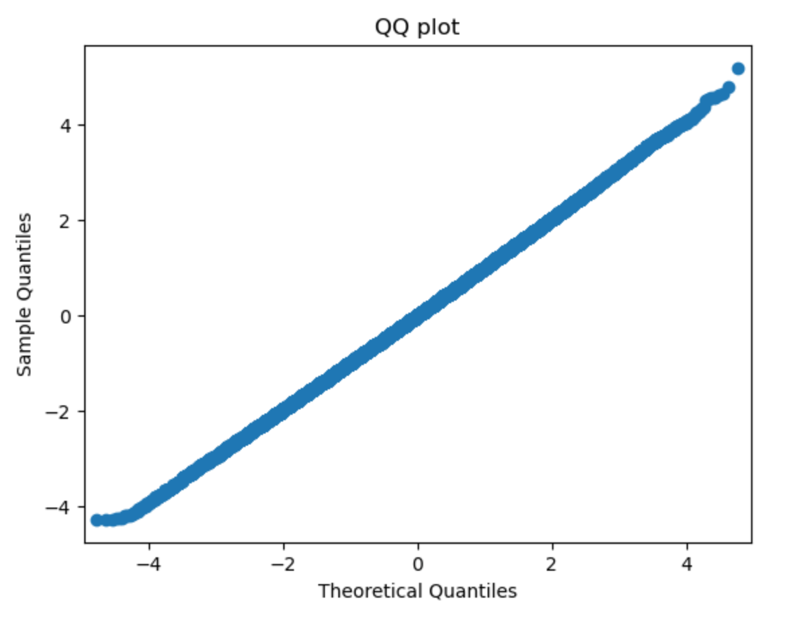


Проанализировав поведение явных читеров, пришли к выводу, что читеры – это те игроки, которые не покупают внутриигровую валюты (money = 0) и тратят больше остальных (cash > 46700). Записав это условие, получили 343 читера:

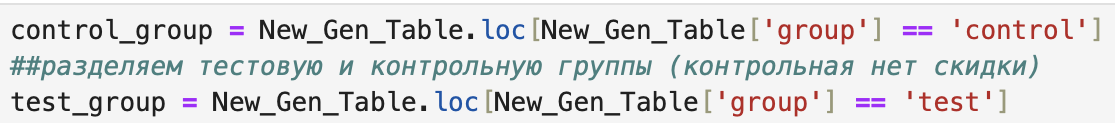


Необходимо удалить неявных читеров из общего DataFrame, в котором уже удалены явные читеры, и сохранить в новый: new\_Gen\_Table (получаем 1079304 users).

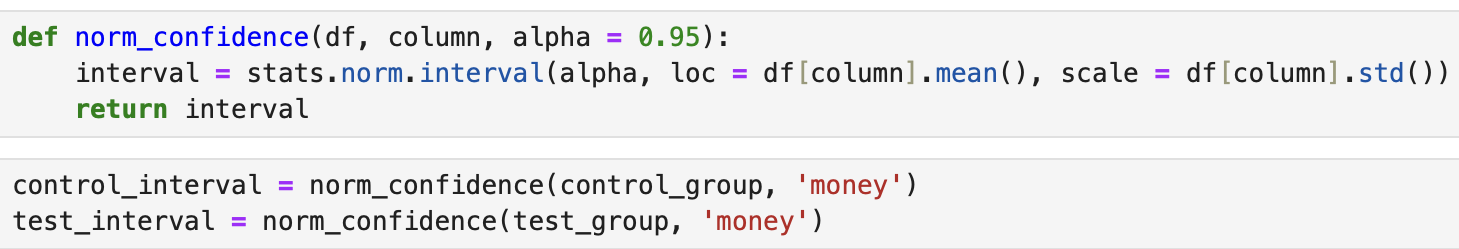
Проверим наши значения на QQ Plot [4]. Получаем прямую, проходящую под углом в 45 градусов, что говорит о том, что теперь можно анализировать результаты акции:



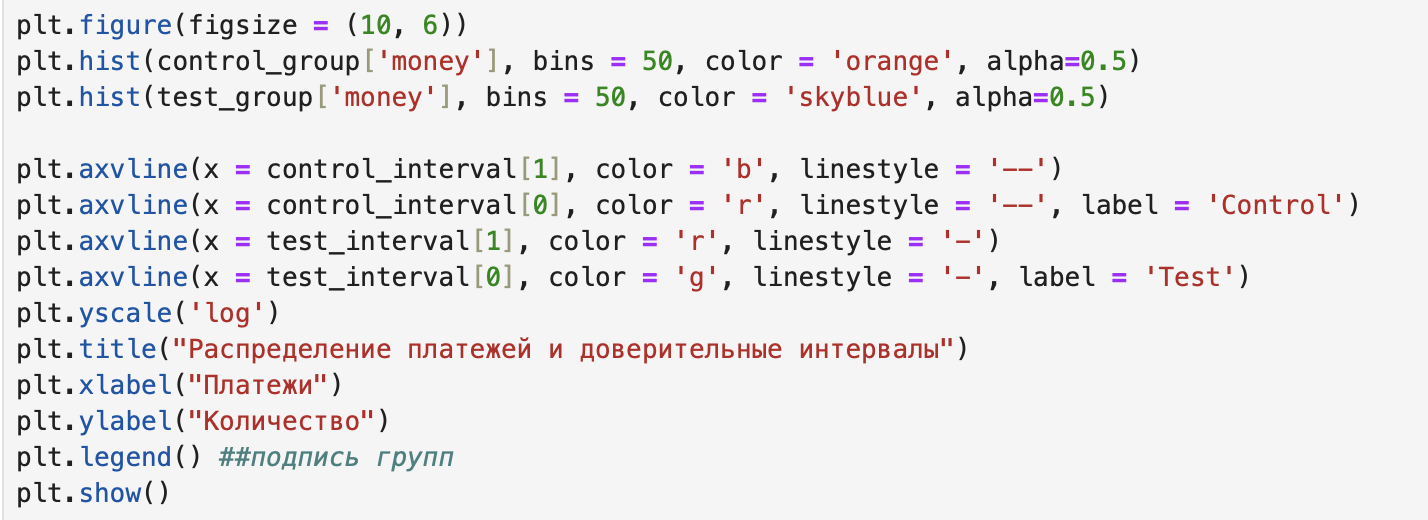
После удаления всех читеров из общей таблицы, метрики по столбцу money распределены нормально. Разделим общую таблицу на несколько DataFram-ов: на тестовую и контрольную группы.



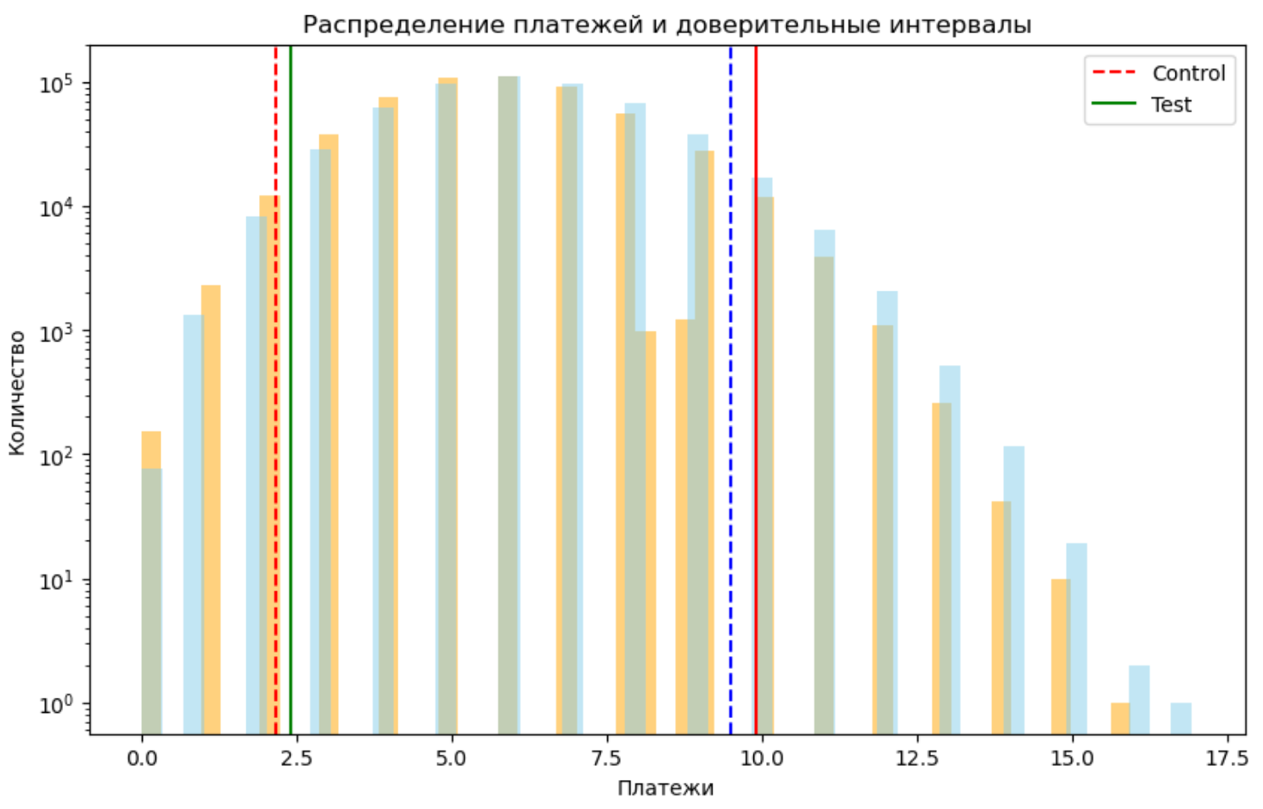
Воспользуемся функцией scipy.stats.norm [5] для нахождения доверительных интервалов для обеих групп (тестовой и контрольной). Определим доверительные интервалы:



Для анализа полученных результатов, построим графики распределения платежей - hist [6], а также укажем доверительные интервалы [7] для групп:



Получаем график:

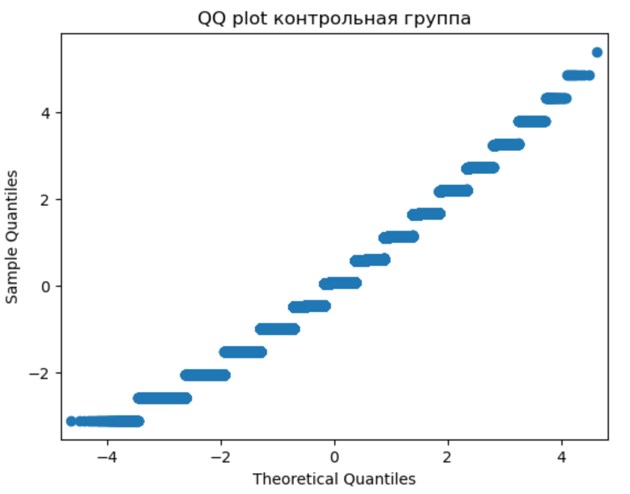
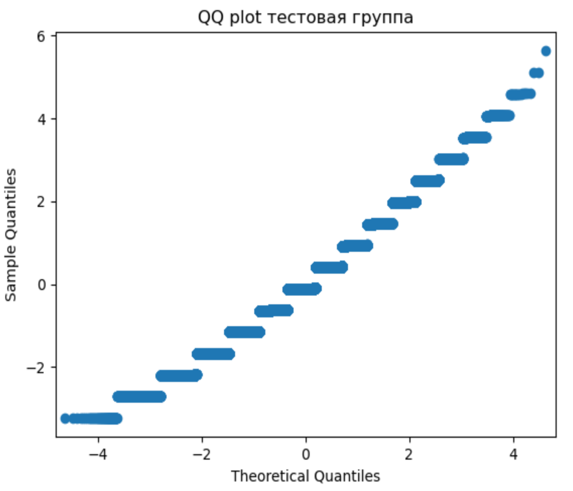


(оранжевая гистограмма – контрольная группа, голубая – тестовая группа)

Интервалы с доверительной вероятность 95% означает, что при проведении большого числа независимых экспериментов в 95% случаев истинное значение параметра будет лежать внутри этого интервала.

Таким образом, из этого графика мы можем заметить, что доверительные интервалы для тестовой и контрольной групп почти не отличаются, это значит, что проведение акции не привело к значительным результатам, то есть акция не удалась.

Если мы будем сравнивать QQ – Plot [4] графики для контрольной и тестовой групп, то мы не увидим разницы, что значит – акция не удалась.

**ОТЧЕТ**

Необходимо рассчитать ARPU (средняя прибыль на игрока) и ARPPU (средняя прибыль на платящего игрока):

ARPU = сумма money / на всех игроков :

ARPU = 6 471 254.9 7000001 / 1 080 000 = 5.9 19275

ARPPU = сумма money / на всех игроков у кого money > 0:

ARPPU = 6 471 254.970000001 / 1 079 303 = 5.995772243753608

Видно, что данные метрики почти не отличаются.

Траты внутриигровой валюты по различным группам:

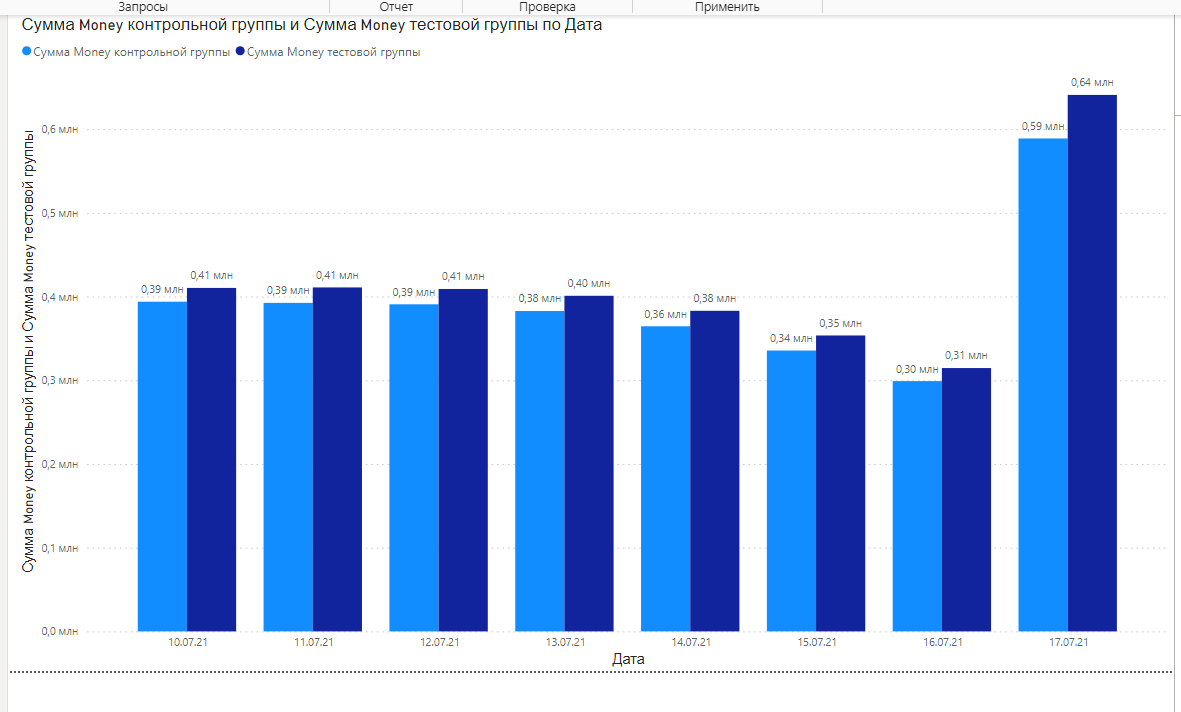
1. Контрольная группа: SUM\_cash = 3 132 349 400
2. Тестовая группа: SUM\_cash = 3 359 696 050

Разница между этими показателями составляет 227 346 650 (тестовая

группа тратит больше).

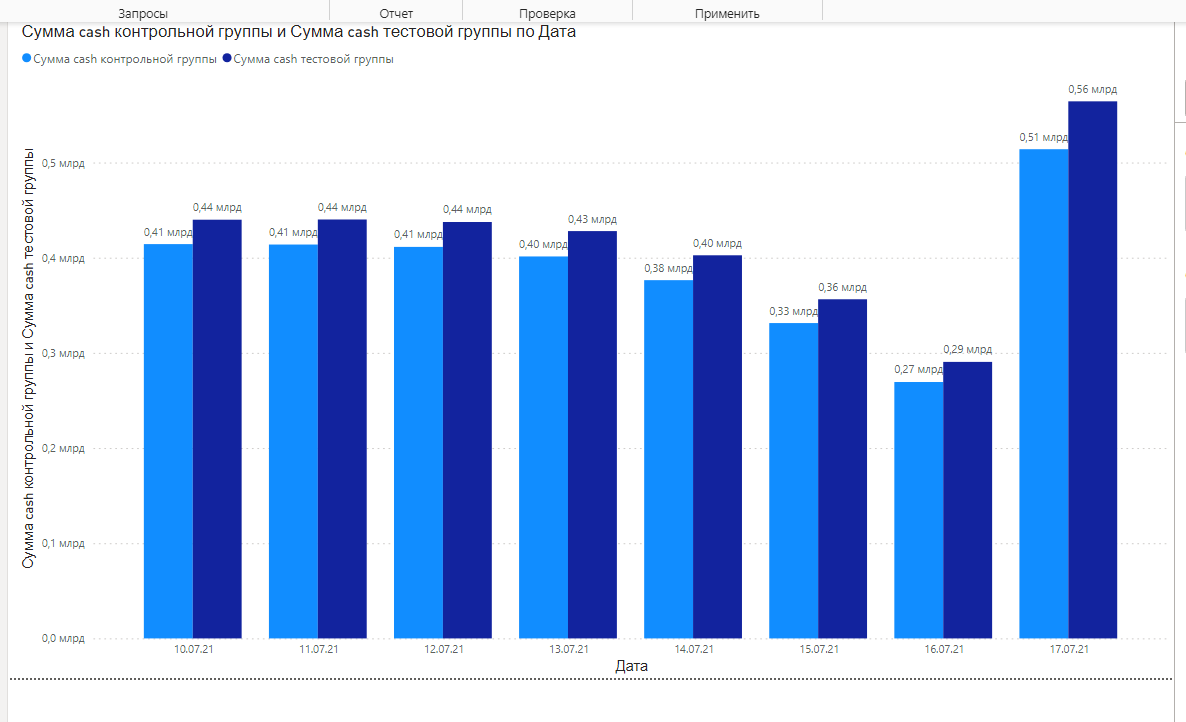
Представим общие данные в виде графиков, выполненных в Power BI:

*График 1. Сумма платежей контрольной и тестовой групп*



*График 2. Сумма трат внутриигровой валюты контрольной и*

*тестовой групп*



Из приведенных выше графиков сравнения метрик, можем сделать вывод, что структура трат пользователей имеет сезонный (подневной) характер: пик в один день и далее постепенное снижение. Можем также заметить, что никто из игроков не копит внутриигровую валюту, и они тратят ее также как и тратят реальные деньги (покупают валюту).

**Итог**: акция не удалась, ее не стоит проводить еще раз.

**Внешние ресурсы**

1. Руководство по библиотеке Pandas, [Электронный ресурс] - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html>;
2. Группировка строк в DataFrame и преобразование в список в Pandas, [Электронный ресурс] - <https://sky.pro/media/gruppirovka-strok-v-dataframe-i-preobrazovanie-v-spisok-v-pandas/>;
3. Выбор строк из DataFrame pandas по списку значений, [Электронный ресурс] - <https://sky.pro/media/vybor-strok-iz-dataframe-pandas-po-spisku-znachenij/#:~:text=Для%20этого%20можно%20использовать%20функцию%20isin()%20из%20библиотеки%20pandas.&text=Таким%20образом%2C%20с%20помощью%20функции,DataFrame%20pandas%20по%20списку%20значений>;
4. QQ – Plots, [Электронный ресурс] - <https://sky.pro/media/vybor-strok-iz-dataframe-pandas-po-spisku-znachenij/#:~:text=Для%20этого%20можно%20использовать%20функцию%20isin()%20из%20библиотеки%20pandas.&text=Таким%20образом%2C%20с%20помощью%20функции,DataFrame%20pandas%20по%20списку%20значений>;
5. Руководство по scipy.stats.norm, [Электронный ресурс] - https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.norm.html;
6. Документация по hist, [Электронный ресурс] - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.hist.html>;
7. Доверительные интервалы, [Электронный ресурс] - <https://lit-review.ru/biostatistika/doveritelnyjj-interval/>;