**Финальная работа по курсу:**

**"Аналитик данных с нуля 2.0"**

**Автор:**

**Сашина Юлия Борисовна**

**Город Тюмень, 2025 г.**

Содержание

[**Раздел 1. Цель проекта** 2](#_Toc201421878)

[Какую задачу необходимо решить? 2](#_Toc201421879)

[Как подходить к её решению? 3](#_Toc201421880)

[**Раздел 2. Анализ источников** 3](#_Toc201421881)

[Какой вариант работы с данными выбран: БД (SQL) / Python. Описание, почему был выбран используемый вариант 3](#_Toc201421882)

[Исходники кодов по загрузке данных 3](#_Toc201421883)

[**Раздел 3. Очистка данных** 4](#_Toc201421884)

[Описание, как вычислять читеров. Каким способом пользоваться, чтобы убрать данные по тем читерам, которые не были обнаружены на момент теста 4](#_Toc201421885)

[**Раздел 4. Использование статистических методов** 6](#_Toc201421886)

[Процесс построения доверительных интервалов для каждой метрики 6](#_Toc201421887)

[Исходники кода 7](#_Toc201421888)

[**Раздел 5. Формирование отчёта** 8](#_Toc201421889)

[Сравнение метрик ARPU (средняя прибыль на игрока), ARPPU (средняя прибыль на платящего игрока) и траты внутриигровой валюты между различными группами игроков 9](#_Toc201421890)

[Графики сравнения метрик по дням (Power BI) 9](#_Toc201421891)

[Исходники программы Python для построения графиков и таблиц 10](#_Toc201421892)

[Сводная таблица в Excel с ARPU по группам и платформам 10](#_Toc201421893)

[**Раздел 6. Выводы** 11](#_Toc201421894)

# **Раздел 1. Цель проекта**

## Какую задачу необходимо решить?

В игре (бесплатный командный онлайн-шутер) есть внутриигровая валюта, которую игроки могут выигрывать, побеждая в матчах, а могут покупать за настоящие деньги.

Необходимо выяснить, стоит ли проводить акцию по премиальной броне по скидке в дальнейшем через анализ влияния на ARPU (средняя прибыль на игрока), ARPPU (средняя прибыль на платящего игрока) и траты внутриигровой валюты. Если игроки, участвовавшие в акции, принесли больше денег, чем игроки, у которых акции не было, то стоит повторять акцию и при этом уже на всех игроках.

## Как подходить к её решению?

Среди игроков есть читеры — игроки, которые с помощью взлома игры начисляют себе большие объёмы внутриигровой валюты. Есть список известных читеров, но есть и ещё не пойманные читеры, чьи результаты могут повлиять на выводы. Попробуем найти их.

Чтобы сравнить результаты тестовой и контрольной групп, нужно сравнить средние по группам, а также построить доверительные интервалы от средних значений с точностью 95%. Если доверительные интервалы пересекаются, то это означает, что результаты случайны и акция не принесла результатов.

# **Раздел 2. Анализ источников**

Какой вариант работы с данными выбран: БД (SQL) / Python. Описание, почему был выбран используемый вариант.

Python и SQL являются двумя самыми популярными средствами для анализа данных и управления данными. SQL и Python используются для разных целей: SQL используется для управления данными, а Python — для разработки, машинного обучения, анализа данных и так далее. Поскольку задачей проекта является анализ исходных данных в формате csv, Python отлично подойдет для данной работы.

Python — это универсальный язык, который можно использовать для решения широкого спектра задач, что делает его подходящим для различных проектов и отраслей. Используя обширные библиотеки и фреймворки, мы можем анализировать данные и манипулировать ими, а также получать мощные возможности с помощью таких библиотек, как Pandas, Numpy. Python — простой в освоении и читаемый язык. У него понятный синтаксис, который помогает легче писать код.

Исходники кодов по загрузке данных

Обработка данных проводилась в облачной среде Google Colab на языке Python. Код для загрузки библиотек:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy.stats import kurtosis

from scipy.stats import skew

import scipy.stats as st

from scipy import stats

Код для загрузки файлов csv с локального компьютера в Google Colab:

from google.colab import files

uploaded = files.upload()

Выгрузка списка имен столбцов каждой таблицы с помощью кода:

df = pd.read\_csv('ABgroup.csv')#название каждой из 5 таблиц

header = list(df.columns)

print("Список имён столбцов:", header)

Исходные данные содержатся в 5 таблицах в формате csv со следующими именами столбцов:

- ABgroup ['user\_id', 'group']

- Cash ['user\_id', 'date', 'cash']

- Cheaters ['user\_id', 'cheaters']

- Money ['user\_id', 'date', 'money']

- Platforms ['user\_id', 'platform']

Код для загрузки данных в DataFrame:

df\_cash = pd.read\_csv('Cash.csv') #название таблицы

# **Раздел 3. Очистка данных**

Описание, как вычислять читеров. Каким способом пользоваться, чтобы убрать данные по тем читерам, которые не были обнаружены на момент теста

Код для создания DataFrame cash и money и подготовки данных к работе над задачей:

df\_cash = pd.read\_csv('Cash.csv')

del df\_cash['date'] #удаляем столбец с датой

df\_cash = df\_cash.groupby(['user\_id'])['cash'].sum().reset\_index() #объединяем одинаковые id, суммируем cash, возвращаем в таблицу

df\_cash.head()

df\_money = pd.read\_csv('Money.csv')

del df\_money['date']

df\_money = df\_money.groupby(['user\_id'])['money'].sum().reset\_index()

df\_money.head()

Объединим cash и money в общий DataFrame:

df\_merge = df\_cash.merge(df\_money, on='user\_id')

df\_unique = df\_merge.drop\_duplicates()

df\_unique.head()

df\_unique.shape[0] #число юзеров в датафрейме 1 080 000

Нужно очистить данные – удалить читеров из файла cheaters.csv, используя данные из столбца cheaters (1 – читер, 0 – не читер). Таким образом будут удалены выявленные читеры.

Код для создания DataFrame cheaters и удаления выявленных читеров:

df\_chiters = pd.read\_csv('Cheaters.csv')

df\_chiters = df\_chiters.loc[(df\_chiters['cheaters'] == 1)] #фильтруем по 1

list\_of\_chiters = df\_chiters['user\_id'].tolist() #создадим список с id

print(len(set(list\_of\_chiters))) #в списке 353 выявленных читера

Код для удаления читеров из списка читеров из объединенного DataFrame cash и money:

df\_unique = df\_unique[~df\_unique['user\_id'].isin(list\_of\_chiters)]

df\_unique.head()

df\_unique.shape[0] #количество оставшихся юзеров 1 079 647

Невыявленных читеров будем искать путем 99 квантиля. Создадим DataFrame с выбором по условиям: (cash для читеров больше 99 квантиля и юзеры с money = 0):

quantile = df\_unique['cash'].quantile(0.99)

quantile #получилось 9350

df\_ch = df\_unique.loc[(df\_unique['cash'] > 9350) & (df\_unique['money'] == 0)]

df\_ch.head()

df\_ch.shape[0] #получилось 344 невыявленных читера

Удалим найденных на последнем шаге читеров из списка читеров:

chiter\_list = df\_ch['user\_id'].tolist() #из столбца df список читеров создаем список с читерами

df\_unique = df\_unique[~df\_unique['user\_id'].isin(chiter\_list)] #удаляем, если противоположно {id юзера есть в списке}, т.е. не в списке

df\_unique.shape[0] #количество оставшихся юзеров - не читеров 1 079 303

Загрузим данные файла Platforms.csv в DataFrame и добавим в таблицу с юзерами столбец с названием платформы:

df\_platform = pd.read\_csv('Platforms.csv')

df\_unique = df\_unique.merge(df\_platform, on='user\_id', how='left')

df\_unique = df\_unique.drop\_duplicates()

df\_unique.head()

Загрузим данные файла ABgroup.csv в DataFrame и добавим в таблицу столбец c типом группы (тестовая или контрольная):

df\_group = pd.read\_csv('ABgroup.csv')

df\_unique = df\_unique.merge(df\_group, on='user\_id', how='left')

df\_unique = df\_unique.drop\_duplicates()

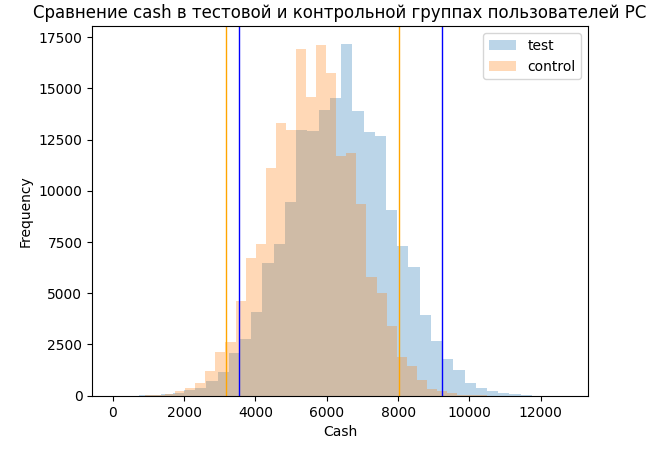
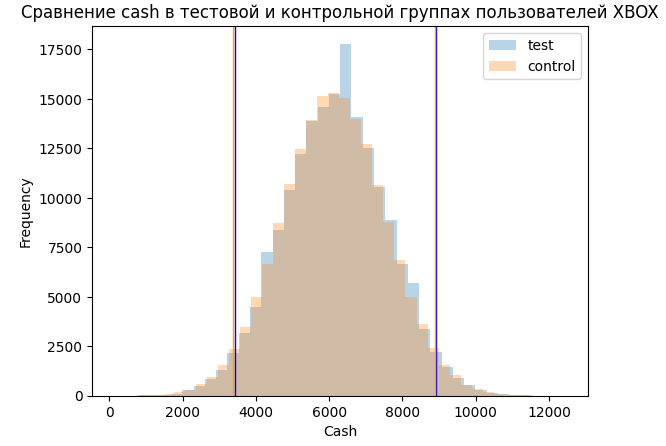
df\_unique.head()

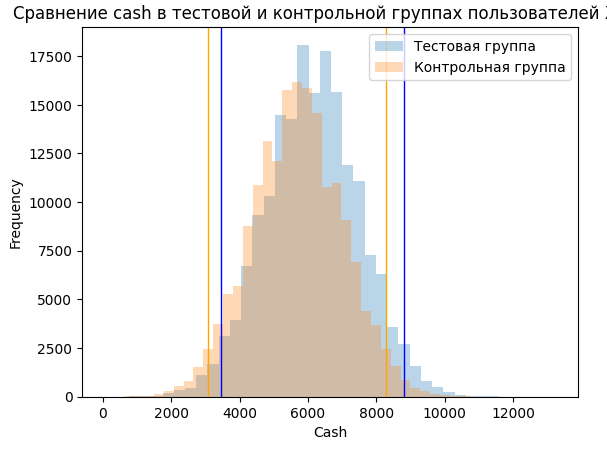
# **Раздел 4. Использование статистических методов**

Процесс построения доверительных интервалов для каждой метрики

Разделим данные на 3 группы платформ: PC, XBox и PS4, а затем по каждой платформе на тестовую и контрольную группы.

Посчитаем средние cash (траты внутриигровой валюты) для каждой платформы для тестовой группы и для контрольной группы, сравним средние по группам, вычислим доверительный интервал, построим графики и сравним их.

**

**

## Исходники кода

Разделим юзеров по 3 платформам путем создания DataFrame для каждой платформы:

df\_pc = df\_unique[df\_unique['platform'] == 'PC']

df\_xbox = df\_unique[df\_unique['platform'] == 'XBox']

df\_ps4 = df\_unique[df\_unique['platform'] == 'PS4']

Разделим юзеров каждой платформы на тестовую и контрольную группу (на примере платформы PC):

df\_pc\_test = df\_pc[df\_pc['group'] == 'test']

df\_pc\_control = df\_pc[df\_pc['group'] == 'control']

Посчитаем средние cash (траты внутриигровой валюты) для каждой платформы для тестовой группы и для контрольной группы и вычислим доверительный интервал (на примере платформы PC):

df\_pc\_test\_mean = df\_pc\_test['cash'].mean()

df\_pc\_test\_std = df\_pc\_test['cash'].std()

ci\_test = stats.norm.interval(0.95, loc=df\_pc\_test\_mean, scale=df\_pc\_test\_std)

print(df\_pc\_test\_mean)

print(df\_pc\_test\_std)

print(f'95% доверительный интервал: {ci\_test}')

df\_pc\_control\_mean = df\_pc\_control['cash'].mean()

df\_pc\_control\_std = df\_pc\_control['cash'].std()

ci\_control = stats.norm.interval(0.95, loc=df\_pc\_control\_mean, scale=df\_pc\_control\_std)

print(df\_pc\_control\_mean)

print(df\_pc\_control\_std)

print(f'95% доверительный интервал: {ci\_control}')

Построим графики (на примере платформы PC):

plt.hist(df\_pc\_test['cash'], bins=40, alpha=0.3, label='test')

plt.hist(df\_pc\_control['cash'], bins=40, alpha=0.3, label='control')

plt.axvline(ci\_test[0], color='blue', linewidth=1)

plt.axvline(ci\_test[1], color='blue', linewidth=1)

plt.axvline(ci\_control[0], color='orange', linewidth=1)

plt.axvline(ci\_control[1], color='orange', linewidth=1)

plt.title('Сравнение cash в тестовой и контрольной группах пользователей PC')

plt.ylabel('Frequency')

plt.xlabel('Cash')

plt.legend(loc='upper right')

plt.savefig('pc.png')

plt.show()

**Раздел 5. Формирование отчёта**

## Сравнение метрик ARPU (средняя прибыль на игрока), ARPPU (средняя прибыль на платящего игрока) и траты внутриигровой валюты между различными группами игроков

Сравнение ARPU (средняя прибыль на игрока):

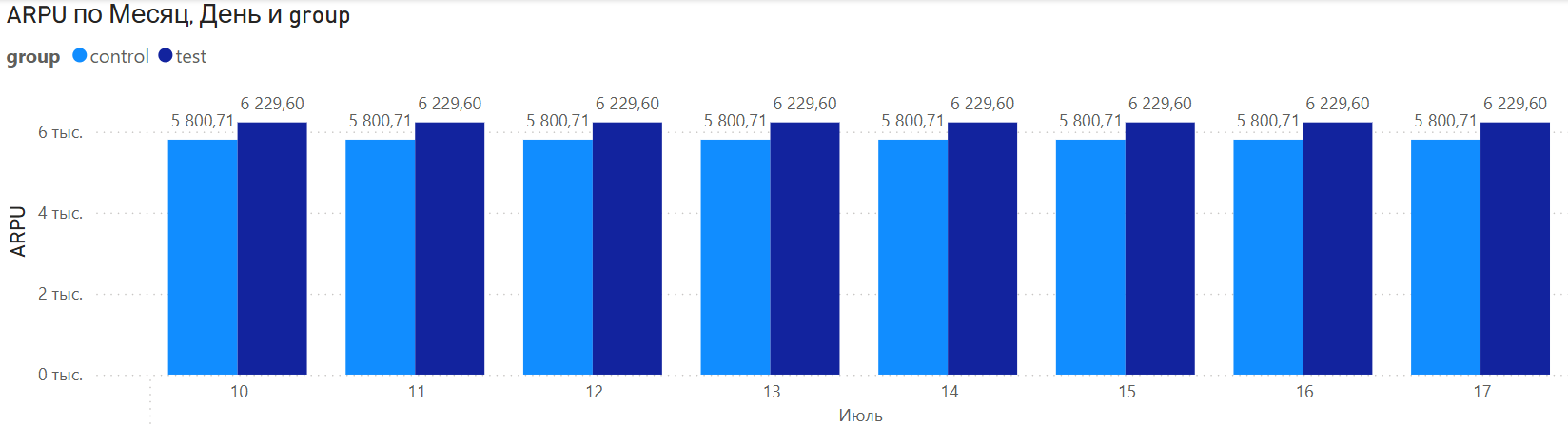
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PC | | XBox | | PS4 | |
| test | control | test | control | test | control |
| 6 382 | 5 587 | 6 176 | 6 132 | 6 130 | 5 682 |

Сравнение ARPPU (средняя прибыль на платящего игрока):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PC | | XBox | | PS4 | |
| test | control | test | control | test | control |
| 6,27 | 5,65 | 6,13 | 6,10 | 6,08 | 5,74 |

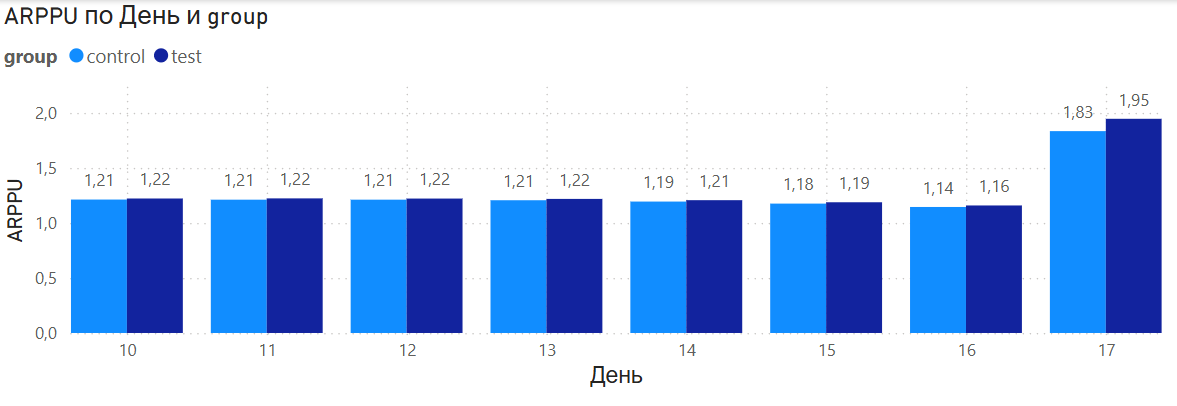
Графики сравнения метрик по дням (Power BI)

ARPU по дням (по группам и платформам):



# 

ARPPU по дням (по группам и платформам):



# 

## Исходники программы Python для построения графиков и таблиц

Считаем ARPU и ARPPU для тестовой и контрольной группы по каждой платформе (на примере тестовой группы платформы PC):

df\_pc\_test\_mean

df\_pc\_test\_money\_mean = df\_pc\_test['money'.mean()

df\_pc\_test\_money\_mean

Сохраняем DataFrame таблицы money, сash, unique в файлы формата csv, предварительно отфильтровав юзеров от читеров (на примере cash):

df\_cash = pd.read\_csv('Cash.csv')

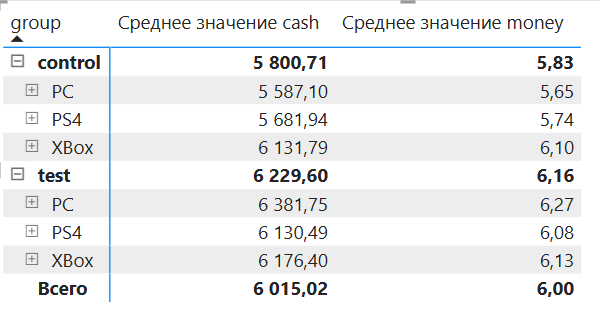
df\_cash = df\_cash[~df\_cash['user\_id'].isin(list\_of\_chiters)] #удаляем выявленных читеров

df\_cash = df\_cash[~df\_cash['user\_id'].isin(chiter\_list)] #удаляем невыявленных читеров

df\_cash.to\_csv('Cash\_withoutCheaters.csv', sep=';')

## Сводная таблица в Excel с ARPU по группам и платформам

В MS Excel сохранить файл df\_unique с объединенными данными ARPU по группам и платформам не получается в связи с очень большим объемом данных. Её можно заменить матрицей MS Power BI:



# **Раздел 6. Выводы**

Анализ по платформам:

1. Xbox

Графики практически слились: доверительные интервалы для тестовой и контрольной групп практически пересекаются. Разница между группами статистически незначима: результаты следует считать случайными. Проводить акцию повторно не нужно.

2. PS4

График тестовой группы смещен чуть правее: средние траты немного выше в тестовой группе, чем в контрольной. Доверительные интервалы имеют небольшое расхождение. Вероятно, есть незначительный положительный эффект от акции. Лучше не распространять акцию на всех игроков, а провести повторную акцию на тестовой группе с новой выборкой.

3. PC

График тестовой группы смещен значительно правее, чем в тестах 2 предыдущих платформ, хотя доверительные интервалы по-прежнему значительно пересекаются. Акция оказала положительный эффект, пусть и не особо значительный. Можно провести повторную акцию на тестовой группе, например, с новой расширенной выборкой, или же рекомендовать повторить акцию.