

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Ассистент

должность, уч. степень, звание

подпись, дата

В.В. Боженко

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Проведение А/В тестирования

по курсу: Введение в анализ данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4916

подпись, дата

Е.А. Ясиновский

инициалы, фамилия

Санкт-Петербург 2022

Цель работы: изучить алгоритмы для проведения А/В тестирования для принятия решений.

Ход работы:

В начале были загружены (рисунок 1) и проверены на корректность(рисунок 2) все данные. После этого был произведен поиск пересекающихся пользователей и последующее их удаление из датафрейма (Рисунок 3).

```
orders = pd.read_csv("orders.csv")
visitors = pd.read_csv("visitors.csv")
```

✓ 0.3s

Рисунок 1 — Загрузка данных

Очистка данных от некорректных значений

```
print("Пропуски")
print(orders.isna().sum())
print("\n")
print(visitors.isna().sum())
print("\n")
print("Дубликаты")
print(orders.duplicated().sum())
print("\n")
print(visitors.duplicated().sum())
print("\n")
```

Преобразование типов данных

```
orders.info()

orders["date"] = orders["date"].astype(np.datetime64)
visitors.info()

visitors["date"] = visitors["date"].astype(np.datetime64)
```

Рисунок 2 - Проверка корректности данных

```

AGroup = orders[orders["group"] == "A"]
BGroup = orders[orders["group"] == "B"]
for aIter, aRow in AGroup.iterrows():
    for bIter, bRow in BGroup.iterrows():
        if aRow["visitorId"] == bRow["visitorId"]:
            orders = orders.drop(orders[orders.visitorId == aRow["visitorId"]].index)

```

Рисунок 3 — Проверка пересекающихся пользователей

Далее была произведена агрегация данных, в итоговой таблице мы объединили кол-во пользователей, заказов и суммы заказов по конкретной дате в конкретной группе (рисунок 4).

```

dateGroupPairs = orders[["date", "group"]].drop_duplicates().reset_index()
dateGroupPairs = dateGroupPairs.drop("index", axis=1)

### Агрегация записей о заказах по дате и группе
ordersAgg = dateGroupPairs.apply(
    lambda pair:
        orders[np.logical_and(
            orders["date"] == pair["date"],
            orders["group"] == pair["group"]
        )].agg({
            "date": "max",
            "group": "max",
            "transactionId": "nunique",
            "revenue": "sum",
            "visitorId": "nunique",
        }),
    axis=1
).sort_values(by=["date", "group"]).reset_index().drop("index", axis=1)

### Агрегация данных о посетителях по дате и группе
visitorsAgg = dateGroupPairs.apply(
    lambda pair:
        visitors[np.logical_and(
            visitors["date"] == pair["date"],
            visitors["group"] == pair["group"]
        )].agg({
            "date": "max",
            "group": "max",
            "visitors": "sum",
        }),
    axis=1
).sort_values(by=["date", "group"]).reset_index().drop("index", axis=1)

ordersAgg

```

Создаю общую таблицу для данных агрегированных по дате и группе для заказов и посетителей

```

commonData = ordersAgg.drop(["date", "group"], axis=1).join(visitors, sort=False)
commonData.rename(columns={"transactionId": "ordersCount", "visitorId": "buyersCount"}, inplace=True)
commonData = commonData[["date", "group", "ordersCount", "buyersCount", "revenue", "visitors"]]
commonData

```

Рисунок 4 — Агрегация данных

Далее выполнил задание по своему варианту (22), а именно: построил график разброса (рисунок 5) и смотря на него узнал количество пользователей и число их заказов (рисунок 6).

```
ordersByUser = orders.groupby(["visitorId"]).agg({"transactionId": "nunique").rename(columns={"transactionId": "ordersCount"}).sort_values(by=["ordersCount"], ascending=False)
ordersByUser

fig, ax = plt.subplots()
x = pd.Series(range(0, len(ordersByUser)))
y = ordersByUser["ordersCount"]
plt.scatter(x, y, c= [[0,0,1,0,1]], s=20)
ax.set_title("Число заказов по пользователям")
fig.set_figwidth(10)
fig.set_figheight(10)
ax.set_xlabel("Количество пользователей")
ax.set_ylabel("Количество заказов")
plt.show()
```

Рисунок 5 — Построение графика

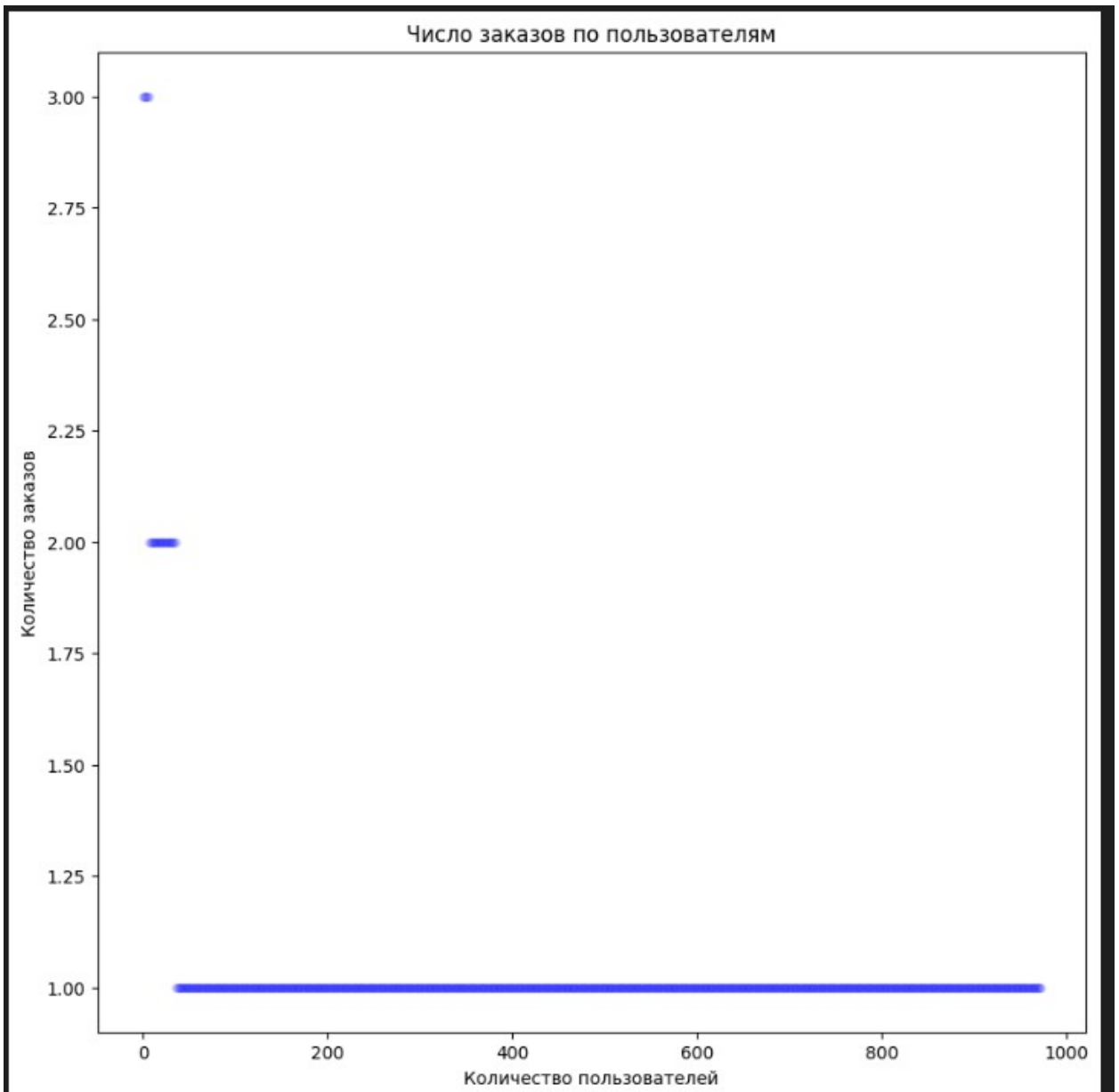


Рисунок 6 — График

Исходя из представленного выше графика видно, что большинство пользователей совершали заказ только один раз, небольшая часть сделала 2 заказа, и только единицы сделали 3 заказа. Больше трех заказов не сделал ни один пользователь.

Далее выполнил вторую часть задания. Для этого я загрузил и очистил данные логов о событиях на сайте (рисунок 7), посчитал сколько раз вызывалось каждое событие (рисунок 8), а также посчитал сколько пользователей совершали каждое из событий (рисунок 9).

```
logs_exp = pd.read_csv("logs_exp.csv", sep="\t")
print(logs_exp.isna().sum())
print(logs_exp.duplicated().sum())
logs_exp = logs_exp.drop_duplicates().reset_index().drop(columns=["index"], axis=1)
logs_exp
```

Рисунок 7 — Загрузка и очистка данных логов

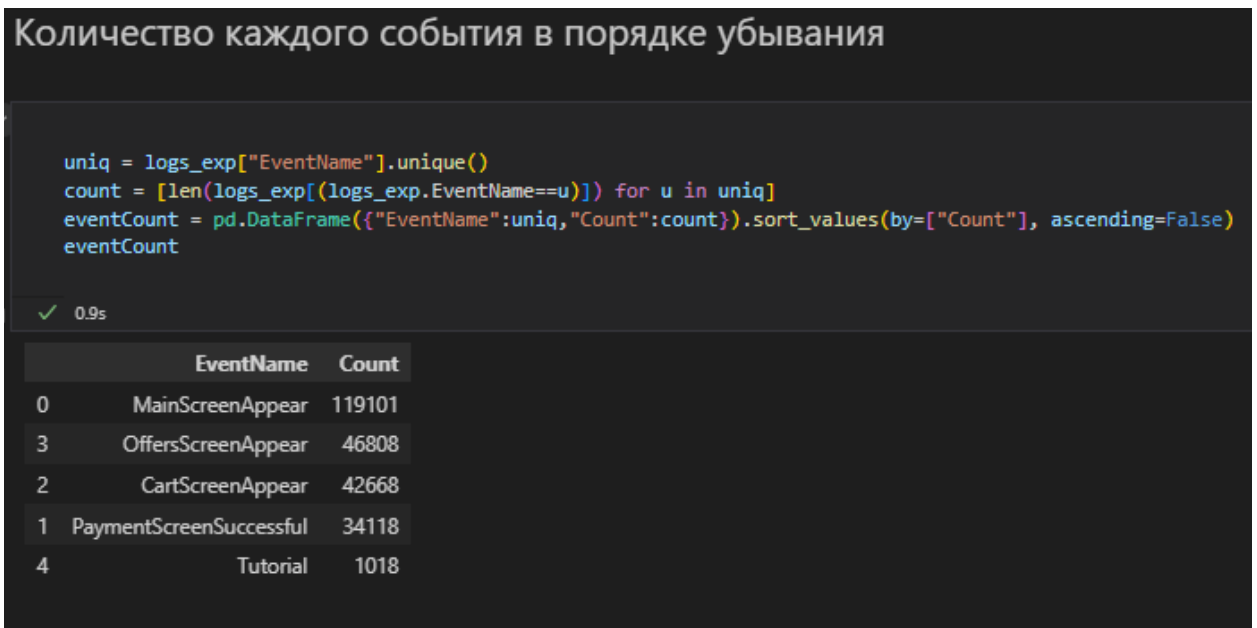
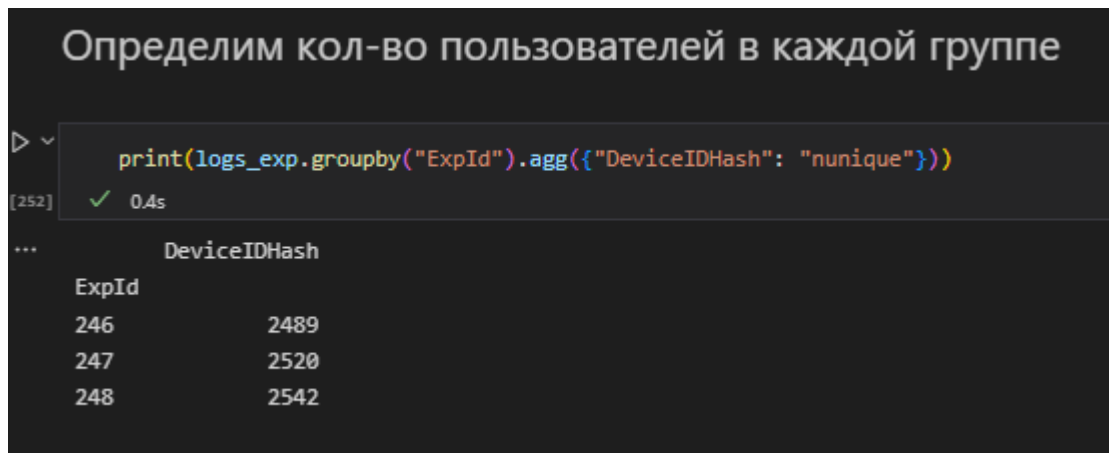


Рисунок 8 — Количество вызова каждого события



Рисунок 9 - Сколько пользователей совершали каждое из событий

Далее я определил кол-во пользователей в каждой группе (рис. 10), и также проверил нет ли пользователей, который одновременно состоят в нескольких группах (рис. 11).



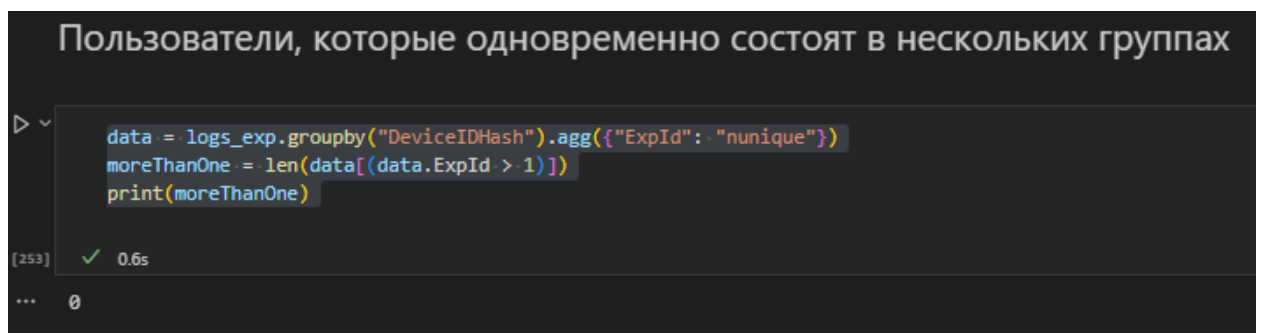
Определим кол-во пользователей в каждой группе

```
print(logs_exp.groupby("ExpId").agg({"DeviceIDHash": "nunique"}))
```

[252] ✓ 0.4s

ExpId	DeviceIDHash
246	2489
247	2520
248	2542

Рисунок 10 — Количество пользователей в каждой группе



Пользователи, которые одновременно состоят в нескольких группах

```
data = logs_exp.groupby("DeviceIDHash").agg({"ExpId": "nunique"})  
moreThanOne = len(data[(data.ExpId > 1)])  
print(moreThanOne)
```

[253] ✓ 0.6s

0

Рисунок 11 — Количество пользователей состоящих одновременно в нескольких группах

В ходе поиска не было найдено ни одного пользователя, который бы находился одновременно в двух группах, а также мы еще раз убедились в балансе наших групп, каждая содержит примерно одинаковое количество пользователей.

Далее я произвел А/А (рис. 13) и А/В (рис. 14) тесты, с помощью функции тестирования представленной на рисунке 12.

```

import scipy
import math
def TestGroup(groupA,groupB,event,alpha):
    group_users = np.array([logs_exp.query('EventName == @event and ExpId == @groupA').DeviceIDHash.nunique(),
                             logs_exp.query('EventName == @event and ExpId == @groupB').DeviceIDHash.nunique()])
    #получим количество пользователей в группах
    total_users = np.array([logs_exp.query('ExpId == @groupA').DeviceIDHash.nunique(),
                             logs_exp.query('ExpId == @groupB').DeviceIDHash.nunique()])
    # пропорция успехов в первой группе:
    p1 = group_users[0]/total_users[0]
    # пропорция успехов во второй группе:
    p2 = group_users[1]/total_users[1]
    # пропорция успехов в комбинированном датасете:
    p_combined = (group_users[0] + group_users[1]) / (total_users[0] + total_users[1])
    # разница пропорций в датасетах
    difference = p1 - p2
    # считаем статистику в ст.отклонениях стандартного нормального распределения
    z_value = difference / math.sqrt(p_combined * (1 - p_combined) * (1 / total_users[0] + 1 / total_users[1]))
    # задаем стандартное нормальное распределение (среднее 0, ст.отклонение 1)
    distr = scipy.stats.norm(0, 1)
    p_value = (1 - distr.cdf(abs(z_value))) * 2
    print('p-значение: ', p_value)
    if p_value < alpha:
        print('Отвергаем нулевую гипотезу: есть значимая разница между долями')
    else:
        print('Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными')

```

Рисунок 12 — Метод для проведения тестов

А/А тест. Нулевая гипотеза: Группа 246 и 247 не имеет различий

Альтернативная гипотеза: Группа 246 и 247 отличаются.

```

TestGroup(246,247,"CartScreenAppear",0.1)
TestGroup(246,247,"MainScreenAppear",0.1)
TestGroup(246,247,"OffersScreenAppear",0.1)
TestGroup(246,247,"PaymentScreenSuccessful",0.1)
TestGroup(246,247,"Tutorial",0.1)

```

✓ 0.1s

```

p-значение: 0.19817540651836918
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными
p-значение: 0.5856639791209877
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными
p-значение: 0.3233225048227193
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными
p-значение: 0.10902546202332886
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными
p-значение: 0.8756574258016974
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

```

Рисунок 13 — Результаты А/А теста

А/А тест не отверг нулевую гипотезу для всех событий.

После этого был произведен А/В тест. Нулевая гипотеза: Нет различий между группой А (246) и группой В (248). Альтернативная: Есть различия между контрольной и экспериментальной группами.

```
TestGroup(246,248,"CartScreenAppear",0.1)
TestGroup(246,248,"MainScreenAppear",0.1)
TestGroup(246,248,"OffersScreenAppear",0.1)
TestGroup(246,248,"PaymentScreenSuccessful",0.1)
TestGroup(246,248,"Tutorial",0.1)
✓ 0.1s

p-значение: 0.10534150642722295
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными
p-значение: 0.39774960182882513
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными
p-значение: 0.2531656758489338
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными
p-значение: 0.23403912852637143
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными
p-значение: 0.8962914725257976
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными
```

Рисунок 14 — Результаты A/B теста

A/B тест не отвергнул нулевую гипотезу ни для одного события.

Ссылка на Jupyter Notebook - <https://github.com/EgorYasinovskiy/Data-Analys.git>

Вывод:

В процессе выполнения лабораторной работы были изучены алгоритмы для проведения А/В тестирования для принятия решений.

В первой части лабораторной работы была проведена предварительная работа с данными Orders и Visitors. Создали массив уникальных пар значений дат и групп теста, после чего получили агрегированные кумулятивные данные по дням о заказах и по дням о посетителях.

Построили график выручки по группам. На графике можно заметить, что до 17 августа 2019 года выручка у обеих групп была приблизительно равна, однако после этого дня у группы В поднялась примерно на 2 единицы и дальше обе группы поднимались равномерно.

Во второй части работы провели предварительную работы с данными Logs_exp. Вывели на экран количество каждого события в порядке убывания частоты и нашли событие, встречающееся чаще всего: событие MainScreenAppear с 119.101 записей. Подсчитали сколько пользователей совершали каждое из событий. Определили количество пользователей в каждой группе и проверили пересекающихся пользователей (т.е. тех пользователей, которые присутствуют в обеих группах). Пересекающихся пользователей не было обнаружено.

Провели А/А и А/В тесты.

Для А/А теста:

- Нулевая гипотеза: Группа 246 и 247 не имеет различий
- Альтернативная гипотеза: Группа 246 и 247 отличаются.

Для всех событий тест А/А не отверг нулевую гипотезу. Это хорошо, поскольку тестирование проводится внутри одной группы.

Для А/В теста:

- Нулевая гипотеза: Нет различий между группой А (246) и группой В (248).
- Альтернативная: Есть различия между контрольной и экспериментальной группами.

Для всех событий тест A/B не показал статистически значимую разницу между группами. Иначе говоря, также не отверг нулевую гипотезу.

Подводя итог, можно сказать, что цель, поставленная в начале выполнения работы, успешно достигнута.

Список используемой литературы:

1. Методические указания по выполнению лабораторной работы №1;
2. Методические указания по выполнению лабораторной работы №5.