Проект по A/B-тестированию

Постановка задачи

Ваша задача — провести оценку результатов A/B-теста. В вашем распоряжении есть датасет с действиями пользователей, техническое задание и несколько вспомогательных датасетов.

- Оцените корректность проведения теста
- Проанализируйте результаты теста

Чтобы оценить корректность проведения теста, проверьте:

- пересечение тестовой аудитории с конкурирующим тестом,
- совпадение теста и маркетинговых событий, другие проблемы временных границ теста.

Техническое задание

- Название теста: recommender_system_test;
- Группы: A (контрольная), B (новая платёжная воронка);
- Дата запуска: 2020-12-07;
- Дата остановки набора новых пользователей: 2020-12-21;
- Дата остановки: 2021-01-04;
- Аудитория: 15% новых пользователей из региона EU;
- Назначение теста: тестирование изменений, связанных с внедрением улучшенной рекомендательной системы;
- Ожидаемое количество участников теста: 6000.
- Ожидаемый эффект: за 14 дней с момента регистрации в системе пользователи покажут улучшение каждой метрики не менее, чем на 10%:
  - конверсии в просмотр карточек товаров — событие product_page
  - просмотры корзины — product_card
  - покупки — purchase.

Загрузите данные теста, проверьте корректность его проведения и проанализируйте полученные результаты.

Данные

ab_project_marketing_events.csv final_ab_new_users.csv final_ab_events.csv final_ab_participants.csv

/datasets/ab_project_marketing_events.csv — календарь маркетинговых событий на 2020 год;

Структура файла:

- name — название маркетингового события;
- regions — регионы, в которых будет проводиться рекламная кампания;
- start_dt — дата начала кампании;
- finish_dt — дата завершения кампании.

/datasets/final_ab_new_users.csv — все пользователи, зарегистрировавшиеся в интернет-магазине в период с 7 по 21 декабря 2020 года;

Структура файла:

- user_id — идентификатор пользователя;
- first_date — дата регистрации;
- region — регион пользователя;
- device — устройство, с которого происходила регистрация.

/datasets/final_ab_events.csv — все события новых пользователей в период с 7 декабря 2020 по 4 января 2021 года;

Структура файла:

- user_id — идентификатор пользователя;
- event_dt — дата и время события;
- event_name — тип события;
- details — дополнительные данные о событии. Например, для покупок, purchase, в этом поле хранится стоимость покупки в долларах.

/datasets/final_ab_participants.csv — таблица участников тестов.

Структура файла:

- user_id — идентификатор пользователя;

- ab_test — название теста;
- group — группа пользователя.

Как сделать задание?

- Опишите цели исследования
- Исследуйте данные
  - Требуется ли преобразование типов?
  - Присутствуют ли пропущенные значения и дубликаты? Если да, то какова их природа?
- Проведите исследовательский анализ данных
  - Исследуйте конверсию в воронке на разных этапах?
  - Обладают ли выборки одинаковыми распределениями количества событий на пользователя?
  - Присутствуют ли в выборках одни и те же пользователи?
  - Как число событий распределено по дням?
  - Подумайте, есть ли какие-то нюансы данных, которые нужно учесть, прежде чем приступать к A/B-тестированию?
- Проведите оценку результатов A/B-тестирования
  - Что можно сказать про результаты A/B-тестирования?
  - Проверьте статистическую разницу долей z-критерием
- Опишите выводы по этапу исследовательского анализа данных и по проведённой оценке результатов A/B-тестирования

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt

import seaborn as sns
sns.set(rc={'figure.figsize':(10, 8)})

import scipy.stats as stats
from scipy import stats as st

import math as mth

import numpy as np

import pandas as pdm
from datetime import datetime,timedelta

from pathlib import Path
```

```
import matplotlib.dates as mdates

import math
import cmath

import plotly.graph_objects as go
import plotly.express as px

import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
```

Исследуйте данные

In [2]:
```
ab_project_marketing_events = pd.read_csv('/datasets/ab_project_marketing_events.csv', sep=',')
final_ab_new_users = pd.read_csv('/datasets/final_ab_new_users.csv', sep=',')
final_ab_events = pd.read_csv('/datasets/final_ab_events.csv', sep=',')
final_ab_participants = pd.read_csv('/datasets/final_ab_participants.csv', sep=',')
```

In [3]:
```
ab_project_marketing_events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   name       14 non-null     object
 1   regions    14 non-null     object
 2   start_dt   14 non-null     object
 3   finish_dt  14 non-null     object
dtypes: object(4)
memory usage: 576.0+ bytes
```

In [4]:
```
final_ab_new_users.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61733 entries, 0 to 61732
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   user_id     61733 non-null  object
```

```
 1   first_date  61733 non-null  object
 2   region      61733 non-null  object
 3   device      61733 non-null  object
dtypes: object(4)
memory usage: 1.9+ MB
```

In [5]:
```
final_ab_events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440317 entries, 0 to 440316
Data columns (total 4 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   user_id     440317 non-null  object
 1   event_dt    440317 non-null  object
 2   event_name  440317 non-null  object
 3   details     62740 non-null   float64
dtypes: float64(1), object(3)
memory usage: 13.4+ MB
```

In [6]:
```
final_ab_events['details'].value_counts()
```

Out[6]:
```
4.99      46362
9.99       9530
99.99      5631
499.99     1217
Name: details, dtype: int64
```

In [7]:
```
final_ab_events['details'] = final_ab_events['details'].fillna(0)
```

In [8]:
```
final_ab_events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440317 entries, 0 to 440316
Data columns (total 4 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   user_id     440317 non-null  object
 1   event_dt    440317 non-null  object
 2   event_name  440317 non-null  object
 3   details     440317 non-null  float64
```

```
dtypes: float64(1), object(3)
memory usage: 13.4+ MB
```

In [9]:
```
final_ab_participants.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18268 entries, 0 to 18267
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   user_id  18268 non-null  object
 1   group    18268 non-null  object
 2   ab_test  18268 non-null  object
dtypes: object(3)
memory usage: 428.3+ KB
```

In [10]:
```
ab_project_marketing_events.duplicated().sum()
```

Out[10]: 0

In [11]:
```
final_ab_new_users.duplicated().sum()
```

Out[11]: 0

In [12]:
```
final_ab_events.duplicated().sum()
```

Out[12]: 0

In [13]:
```
final_ab_participants.duplicated().sum()
```

Out[13]: 0

In [14]:
```
final_ab_events['event_dt'] = final_ab_events['event_dt'].astype('datetime64')
```

In [15]:
```
final_ab_new_users['first_date'] = final_ab_new_users['first_date'].astype('datetime64')
```

In [16]:
```
ab_project_marketing_events['start_dt'] = ab_project_marketing_events['start_dt'].astype('datetime64')
ab_project_marketing_events['finish_dt'] = ab_project_marketing_events['finish_dt'].astype('datetime64')
```

In [17]:
```
final_ab_events['event_name'].unique()
```

Out[17]:
```
array(['purchase', 'product_cart', 'product_page', 'login'], dtype=object)
```

пропуски говорят о том, что событие было бесплатным

Проведите исследовательский анализ данных

Исследуйте конверсию в воронке на разных этапах?

In [18]:
```
final_ab_new_users = pd.merge(final_ab_new_users, final_ab_participants, how = 'left')
final_ab_new_users = final_ab_new_users.dropna()
final_ab_new_users
```

Out[18]:

	user_id	first_date	region	device	group	ab_test
0	D72A72121175D8BE	2020-12-07	EU	PC	A	recommender_system_test
2	2E1BF1D4C37EA01F	2020-12-07	EU	PC	A	interface_eu_test
3	50734A22C0C63768	2020-12-07	EU	iPhone	B	interface_eu_test
13	E6DE857AFBDC6102	2020-12-07	EU	PC	B	recommender_system_test
14	E6DE857AFBDC6102	2020-12-07	EU	PC	B	interface_eu_test
...	...	...	...	...	...	...
63315	27007FC1A9B62FC5	2020-12-20	EU	iPhone	B	interface_eu_test
63317	89CB0BFBC3F35126	2020-12-20	EU	PC	B	recommender_system_test
63322	75F25D4DADA37ABB	2020-12-20	EU	Android	B	interface_eu_test
63323	2C29721DDDA76B2A	2020-12-20	EU	iPhone	B	interface_eu_test
63333	1C7D23927835213F	2020-12-20	EU	iPhone	B	interface_eu_test

18268 rows × 6 columns

In [19]:
```
final_ab_events = pd.merge(final_ab_events, final_ab_new_users, how = 'left')
final_ab_events = final_ab_events.dropna()
final_ab_events
```

Out[19]:

	user_id	event_dt	event_name	details	first_date	region	device	group	ab_test
3	96F27A054B191457	2020-12-07 04:02:40	purchase	4.99	2020-12-07	EU	iPhone	B	interface_eu_test
5	831887FE7F2D6CBA	2020-12-07 06:50:29	purchase	4.99	2020-12-07	EU	Android	A	recommender_system_test
9	A92195E3CFB83DBD	2020-12-07 00:32:07	purchase	4.99	2020-12-07	EU	Android	A	interface_eu_test
11	354D653172FF2A2D	2020-12-07 15:45:11	purchase	4.99	2020-12-07	EU	Mac	A	interface_eu_test
12	7FCD34F47C13A9AC	2020-12-07 22:06:13	purchase	9.99	2020-12-07	EU	PC	B	interface_eu_test
...	...	...	...	...	...	...	...	...	...
446183	75845C83258FBF73	2020-12-30 06:42:52	login	0.00	2020-12-07	EU	Android	B	recommender_system_test
446185	4584E51B99DE51AE	2020-12-30 07:39:32	login	0.00	2020-12-07	EU	Mac	A	interface_eu_test
446186	9DF4F595A906A0BA	2020-12-30 18:28:55	login	0.00	2020-12-07	EU	Android	B	interface_eu_test
446205	6181F3835EBE66BF	2020-12-30 12:00:00	login	0.00	2020-12-07	EU	Android	A	interface_eu_test
446209	F80C9BDDEA02E53C	2020-12-30 09:53:39	login	0.00	2020-12-07	EU	iPhone	A	interface_eu_test

106625 rows × 9 columns

In [20]:
```
final_ab_new_users_grouped = final_ab_new_users.groupby(['ab_test','group'], as_index=False).agg({'user_id':'count'}).sort_values(
final_ab_new_users_grouped
```

Out[20]:

	ab_test	group	user_id
0	interface_eu_test	A	5831
1	interface_eu_test	B	5736
2	recommender_system_test	A	3824

	ab_test	group	user_id
3	recommender_system_test	B	2877

- Название теста: recommender_system_test;
- Группы: A (контрольная), B (новая платёжная воронка);
- Дата запуска: 2020-12-07;
- Дата остановки набора новых пользователей: 2020-12-21;
- Дата остановки: 2021-01-04;
- Аудитория: 15% новых пользователей из региона EU;

In [21]:
```
final_ab_events_copy = final_ab_events.copy()
```

In [22]:
```
filtered_users = final_ab_events.query("ab_test != 'recommender_system_test'")['user_id']
```

In [23]:
```
final_ab_events = final_ab_events.query("region == 'EU' and event_dt > '2020-12-07' and event_dt < '2020-12-21'")
final_ab_events = final_ab_events.query('user_id not in @filtered_users')
```

In [24]:
```
other_test_users = final_ab_events_copy.query("ab_test != 'recommender_system_test'")['user_id']
final_ab_events.query('user_id in @other_test_users')
```

Out[24]:

user_id	event_dt	event_name	details	first_date	region	device	group	ab_test

In [25]:
```
final_ab_events.head()
```

Out[25]:

	user_id	event_dt	event_name	details	first_date	region	device	group	ab_test
5	831887FE7F2D6CBA	2020-12-07 06:50:29	purchase	4.99	2020-12-07	EU	Android	A	recommender_system_test
17	3C5DD0288AC4FE23	2020-12-07 19:42:40	purchase	4.99	2020-12-07	EU	PC	A	recommender_system_test
58	49EA242586C87836	2020-12-07 06:31:24	purchase	99.99	2020-12-07	EU	iPhone	B	recommender_system_test
75	A640F31CAC7823A6	2020-12-07 18:48:26	purchase	4.99	2020-12-07	EU	PC	B	recommender_system_test

	user_id	event_dt	event_name	details	first_date	region	device	group	ab_test
121	A9908F62C41613A8	2020-12-07 11:26:47	purchase	9.99	2020-12-07	EU	PC	B	recommender_system_test

In [26]:
```
final_ab_events['session_week'] = final_ab_events['event_dt'].dt.week
final_ab_events['session_date'] = final_ab_events['event_dt'].dt.date
```

```
/tmp/ipykernel_48/1494218649.py:1: FutureWarning: Series.dt.weekofyear and Series.dt.week have been deprecated.  Please use Serie
s.dt.isocalendar().week instead.
  final_ab_events['session_week'] = final_ab_events['event_dt'].dt.week
```

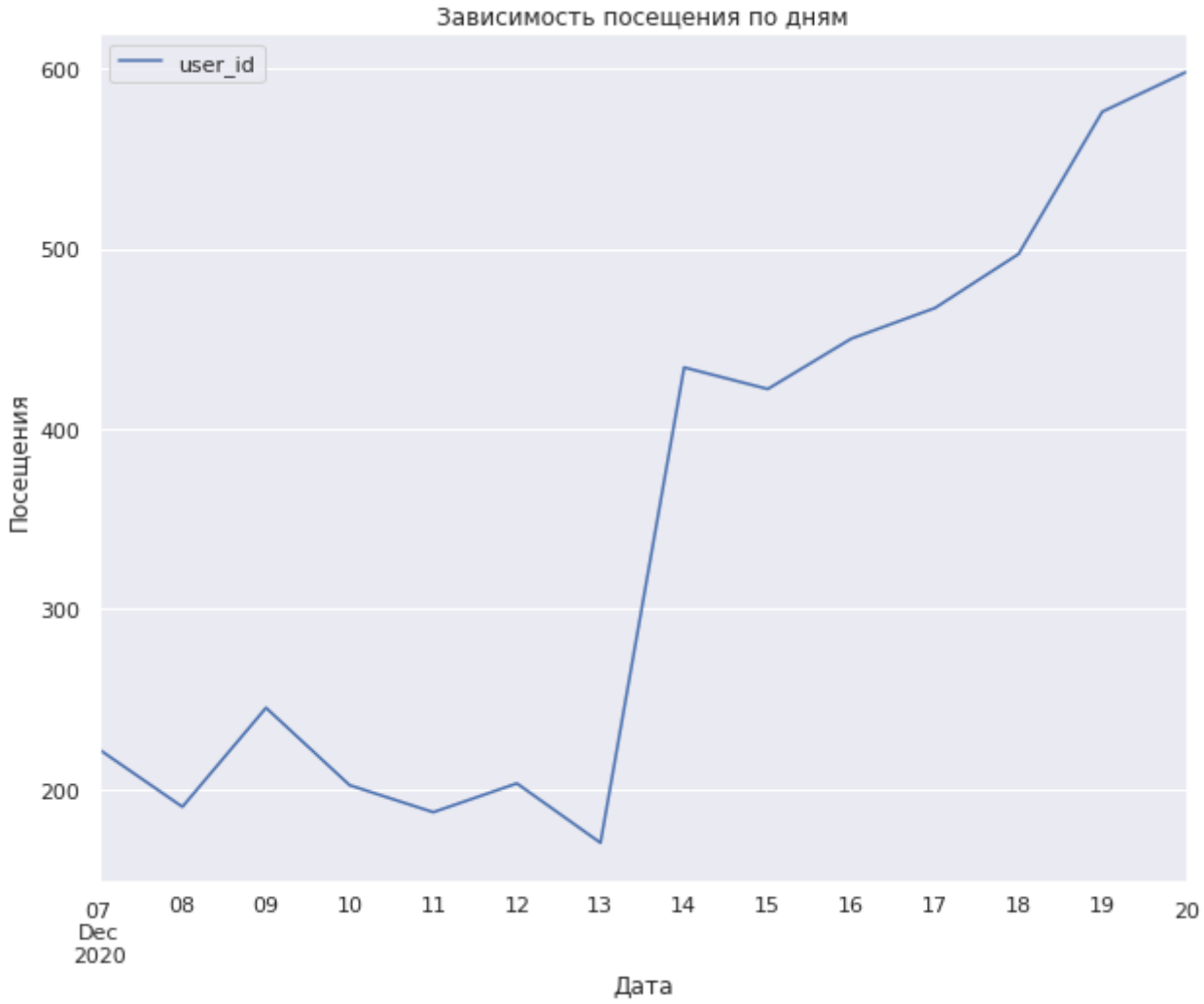
In [27]:
```
final_ab_events['session_date'] = final_ab_events['session_date'].astype('datetime64')
```

In [28]:
```
dau_total = final_ab_events.groupby('session_date').agg({'user_id': 'nunique'}).mean()
wau_total = final_ab_events.groupby(['session_week']).agg({'user_id': 'nunique'}).mean()
```

In [29]:
```
dau_total_gr = final_ab_events.groupby('session_date').agg({'user_id': 'nunique'})
wau_total_gr = final_ab_events.groupby(['session_week']).agg({'user_id': 'nunique'})
```

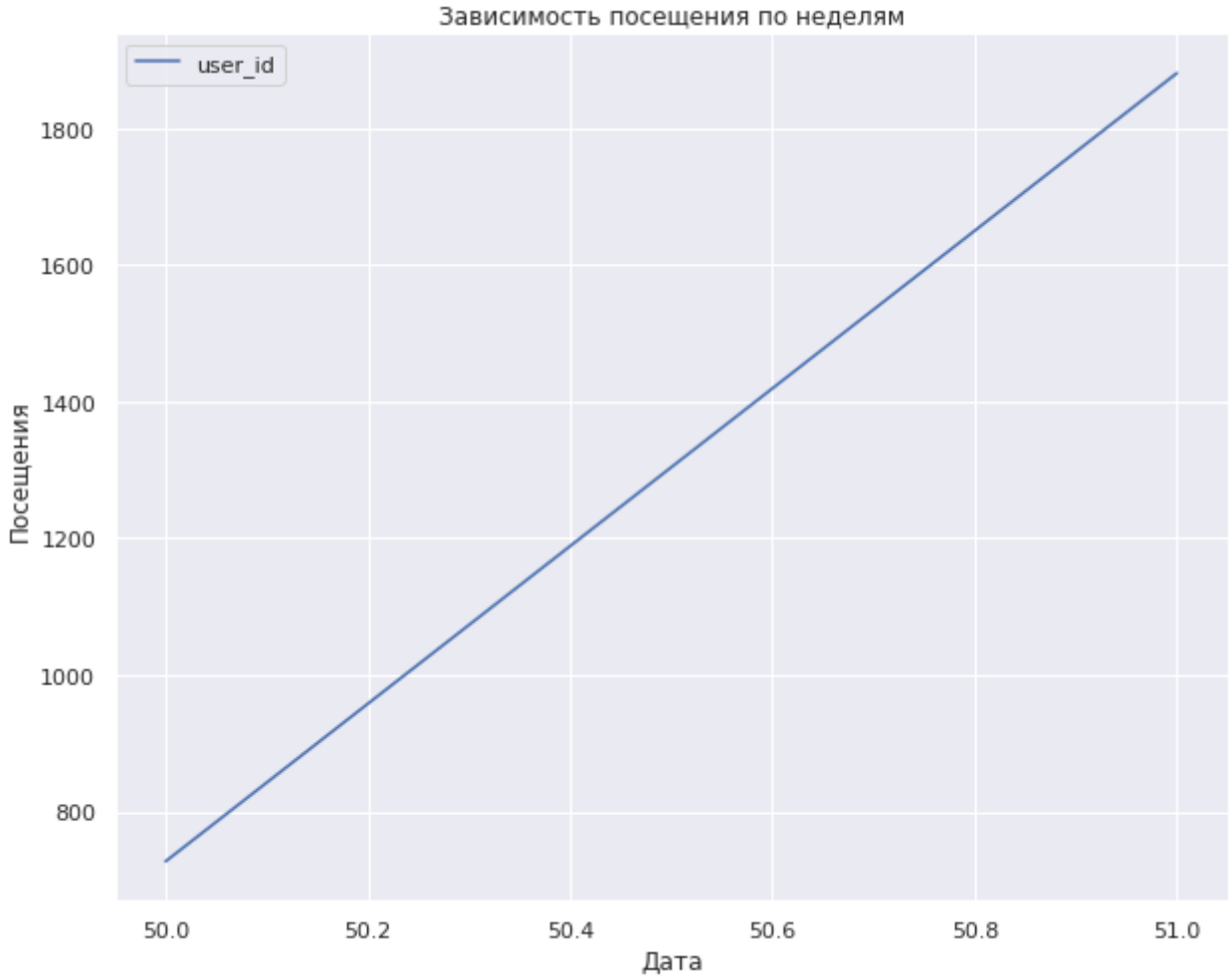
In [30]:
```
ax_dau = dau_total_gr.plot()
ax_dau.set_title('Зависимость посещения по дням')
ax_dau.set_xlabel('Дата')
ax_dau.set_ylabel('Посещения')
```

Out[30]:
```
Text(0, 0.5, 'Посещения')
```



In [31]:
```
ax_wau = wau_total_gr.plot()
ax_wau.set_title('Зависимость посещения по неделям')
ax_wau.set_xlabel('Дата')
ax_wau.set_ylabel('Посещения')
```

Out[31]:
```
Text(0, 0.5, 'Посещения')
```



In [32]:
```
final_ab_events.nunique()
```

Out[32]:
```
user_id       2244
event_dt      7204
event_name       4
details          5
first_date      14
region           1
device           4
group            2
ab_test          1
```

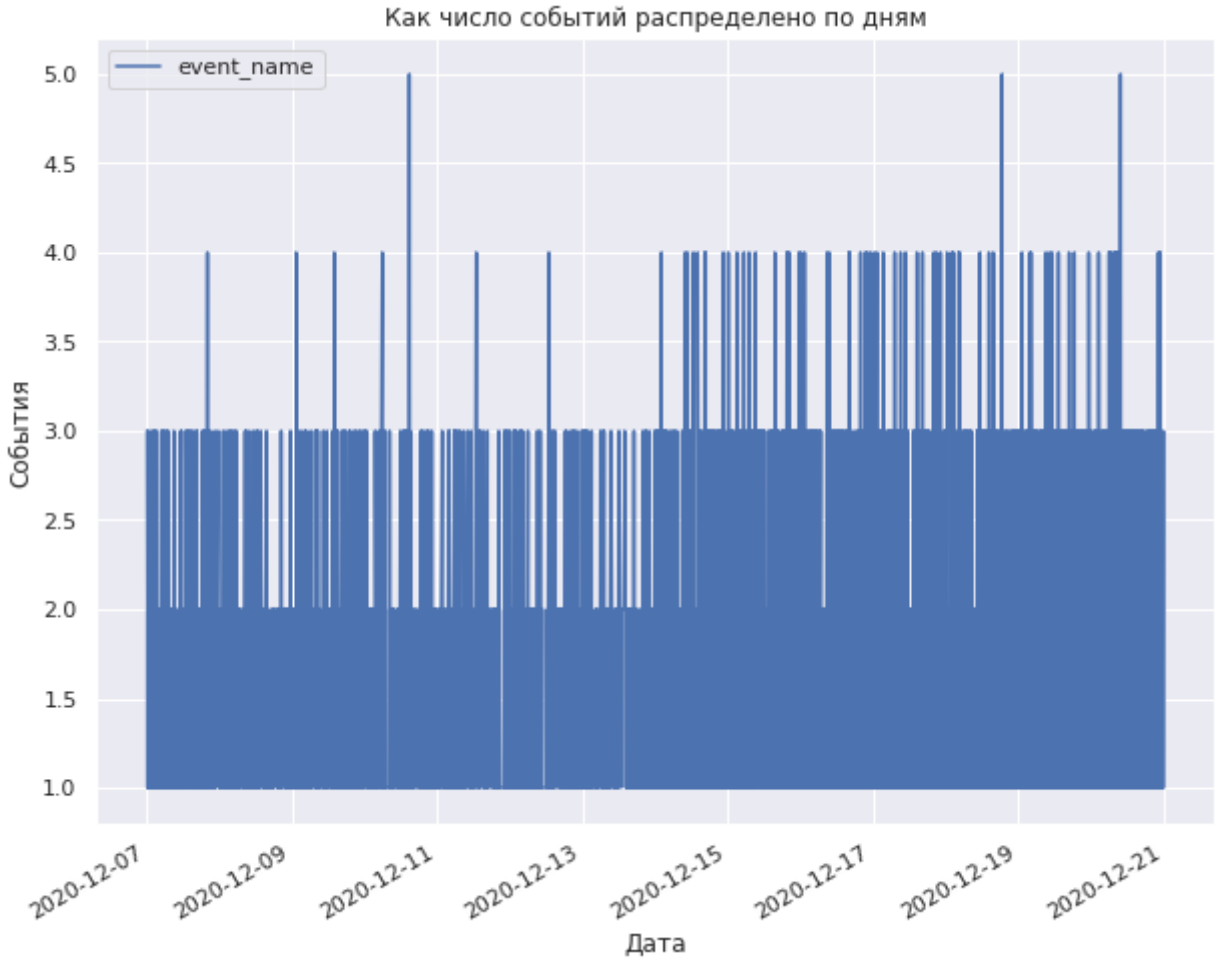
```
session_week       2
session_date      14
dtype: int64
```

повторяющиеся значения по user_id есть

In [33]:
```
event = final_ab_events.groupby('event_dt').agg({'event_name':'count'})
```

In [34]:
```
ax = event.plot()
ax.set_title('Как число событий распределено по дням')
ax.set_xlabel('Дата')
ax.set_ylabel('События')
```

Out[34]:
```
Text(0, 0.5, 'События')
```



In [35]:
```
final_ab_events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10740 entries, 5 to 371225
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   user_id       10740 non-null  object
 1   event_dt      10740 non-null  datetime64[ns]
 2   event_name    10740 non-null  object
 3   details       10740 non-null  float64
 4   first_date    10740 non-null  datetime64[ns]
```

```
 5   region        10740 non-null  object
 6   device        10740 non-null  object
 7   group         10740 non-null  object
 8   ab_test       10740 non-null  object
 9   session_week  10740 non-null  int64
 10  session_date  10740 non-null  datetime64[ns]
dtypes: datetime64[ns](3), float64(1), int64(1), object(6)
memory usage: 1006.9+ KB
```

In [36]:
```
final_ab_events_count = (final_ab_events
                .groupby(['event_name'])['user_id'].nunique()
                .reset_index()
                .rename(columns={'user_id':'counts'})
                        )
final_ab_events_count
```

Out[36]:

	event_name	counts
0	login	2243
1	product_cart	684
2	product_page	1402
3	purchase	684

In [37]:
```
df = pd.merge(final_ab_new_users, final_ab_events,on='user_id', how = 'left')
df['details'].unique()
```

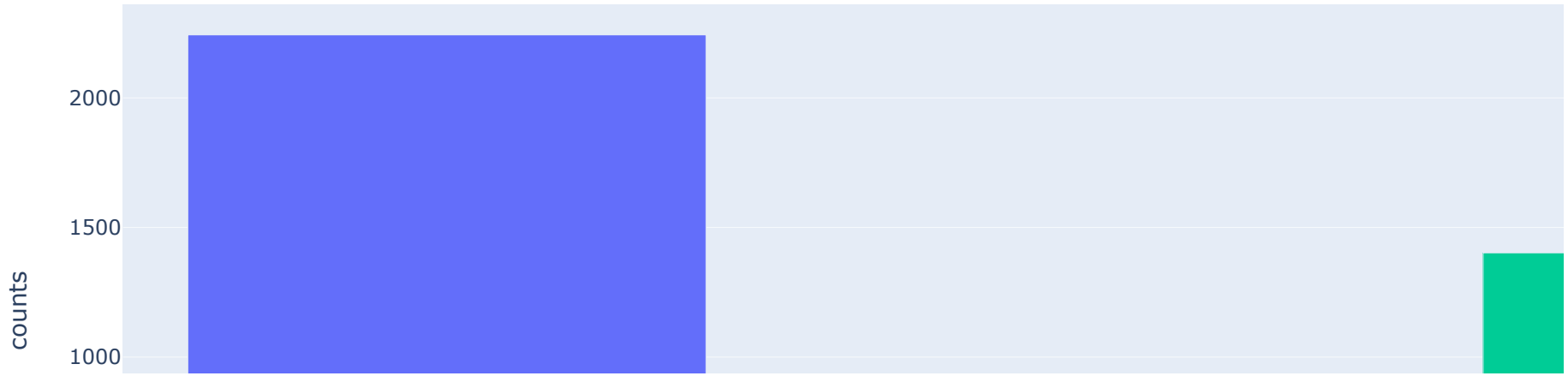
Out[37]:
```
array([  0.  ,    nan,   4.99,  99.99,   9.99, 499.99])
```

In [38]:
```
final_ab_events_sum = (final_ab_events
                .groupby(['region'])['details'].sum()
                .reset_index()
                .rename(columns={'details':'sum_details'})
                        )
final_ab_events_sum
```

Out[38]:

	region	sum_details
0	EU	29391.07

In [39]:
```
fig = px.bar(final_ab_events_count, x='event_name', y='counts', color='event_name')
fig.update_xaxes(tickangle=45)
fig.show()
```



In [40]:
```
fig = px.bar(final_ab_events_sum, x='region', y='sum_details', color='region')
fig.update_xaxes(tickangle=45)
```

```
fig.show()
```

In [41]:

```
event = final_ab_events.groupby('user_id').agg({'details':'count'})
event
```

Out[41]:

user_id	details
0010A1C096941592	6
003DF44D7589BBD4	9
00505E15A9D81546	4
006E3E4E232CE760	4
00A52DCF85F1BE03	1
...	...
FF1AB7A774128512	4
FF40F25452E70E3D	4
FF4456FBA59116E3	6
FF825C1D791989B5	2
FFAE9489C76F352B	6

2244 rows × 1 columns

In [42]:

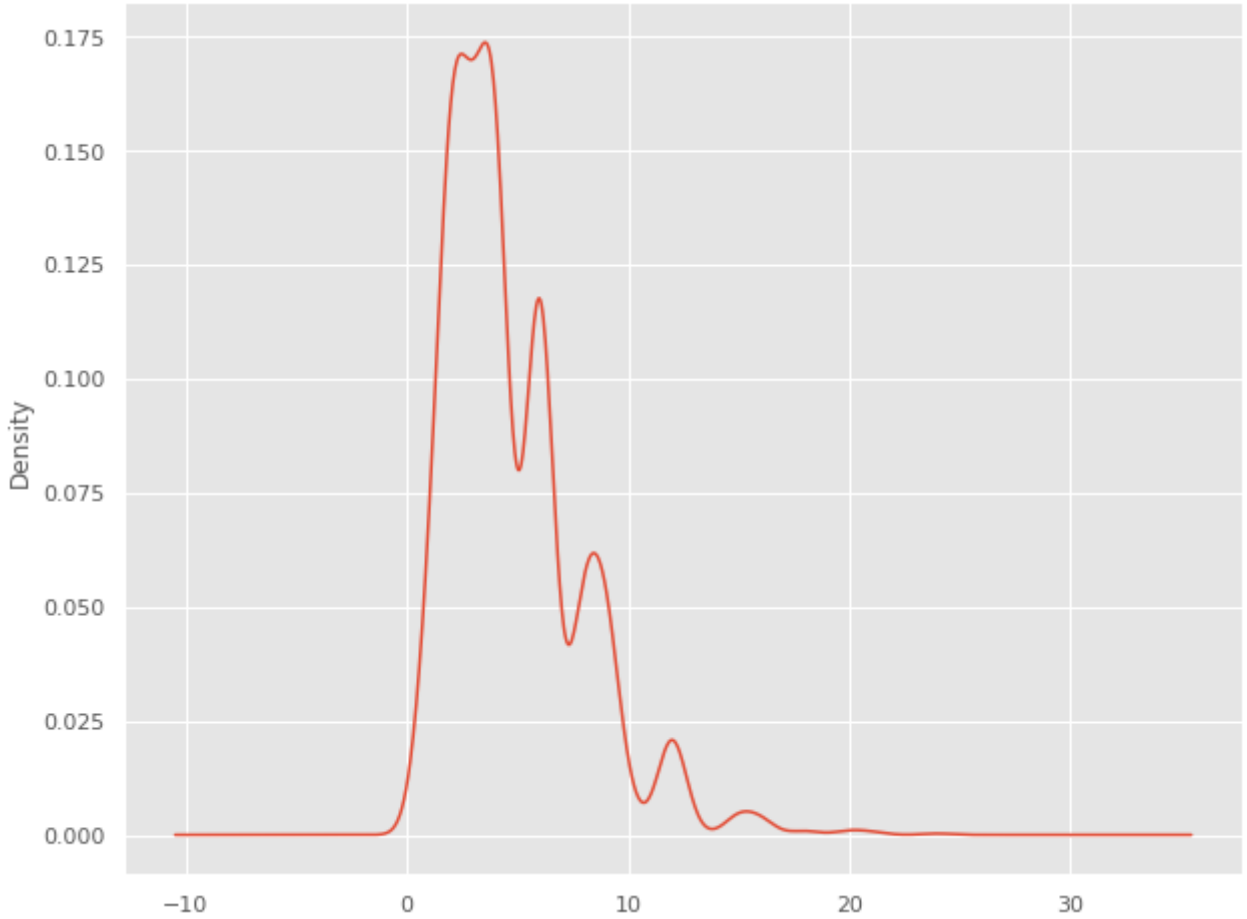
```
matplotlib.style.use('ggplot')

s = pd.Series(event['details'])

s.plot.kde()
```

Out[42]:

```
<AxesSubplot:ylabel='Density'>
```



In [43]:
```
final_ab_events
```

Out[43]:

	user_id	event_dt	event_name	details	first_date	region	device	group	ab_test	session_week	session_date
5	831887FE7F2D6CBA	2020-12-07 06:50:29	purchase	4.99	2020-12-07	EU	Android	A	recommender_system_test	50	2020-12-07
17	3C5DD0288AC4FE23	2020-12-07 19:42:40	purchase	4.99	2020-12-07	EU	PC	A	recommender_system_test	50	2020-12-07
58	49EA242586C87836	2020-12-07 06:31:24	purchase	99.99	2020-12-07	EU	iPhone	B	recommender_system_test	50	2020-12-07

	user_id	event_dt	event_name	details	first_date	region	device	group	ab_test	session_week	session_date
75	A640F31CAC7823A6	2020-12-07 18:48:26	purchase	4.99	2020-12-07	EU	PC	B	recommender_system_test	50	2020-12-07
121	A9908F62C41613A8	2020-12-07 11:26:47	purchase	9.99	2020-12-07	EU	PC	B	recommender_system_test	50	2020-12-07
...	...	...	...	...	...	...	...	...	...	...	...
371185	87C4963DF01E3B3C	2020-12-20 13:18:15	login	0.00	2020-12-20	EU	Android	B	recommender_system_test	51	2020-12-20
371195	3CA972F86411CF13	2020-12-20 06:46:36	login	0.00	2020-12-20	EU	PC	A	recommender_system_test	51	2020-12-20
371208	0F7D49FC184EDCDE	2020-12-20 04:04:27	login	0.00	2020-12-20	EU	PC	A	recommender_system_test	51	2020-12-20
371224	574ACBC674BC385D	2020-12-20 04:15:43	login	0.00	2020-12-20	EU	Mac	A	recommender_system_test	51	2020-12-20
371225	0416B34D35C8C8B8	2020-12-20 20:58:25	login	0.00	2020-12-20	EU	Android	A	recommender_system_test	51	2020-12-20

10740 rows × 11 columns

покупок больше чем переходов на product_card

In [44]:
```
ab_project_marketing_events = ab_project_marketing_events.rename(columns={'regions': 'region'})
```

In [45]:
```
df = pd.merge(final_ab_new_users, ab_project_marketing_events,on='region', how = 'left')
```

In [46]:
```
final_ab_new_users
```

Out[46]:

	user_id	first_date	region	device	group	ab_test
0	D72A72121175D8BE	2020-12-07	EU	PC	A	recommender_system_test
2	2E1BF1D4C37EA01F	2020-12-07	EU	PC	A	interface_eu_test

	user_id	first_date	region	device	group	ab_test
3	50734A22C0C63768	2020-12-07	EU	iPhone	B	interface_eu_test
13	E6DE857AFBDC6102	2020-12-07	EU	PC	B	recommender_system_test
14	E6DE857AFBDC6102	2020-12-07	EU	PC	B	interface_eu_test
...	...	...	...	...	...	...
63315	27007FC1A9B62FC5	2020-12-20	EU	iPhone	B	interface_eu_test
63317	89CB0BFBC3F35126	2020-12-20	EU	PC	B	recommender_system_test
63322	75F25D4DADA37ABB	2020-12-20	EU	Android	B	interface_eu_test
63323	2C29721DDDA76B2A	2020-12-20	EU	iPhone	B	interface_eu_test
63333	1C7D23927835213F	2020-12-20	EU	iPhone	B	interface_eu_test

18268 rows × 6 columns

In [47]:
```
df = pd.merge(final_ab_new_users, ab_project_marketing_events,on='region', how = 'left')
df
```

Out[47]:

	user_id	first_date	region	device	group	ab_test	name	start_dt	finish_dt
0	D72A72121175D8BE	2020-12-07	EU	PC	A	recommender_system_test	NaN	NaT	NaT
1	2E1BF1D4C37EA01F	2020-12-07	EU	PC	A	interface_eu_test	NaN	NaT	NaT
2	50734A22C0C63768	2020-12-07	EU	iPhone	B	interface_eu_test	NaN	NaT	NaT
3	E6DE857AFBDC6102	2020-12-07	EU	PC	B	recommender_system_test	NaN	NaT	NaT
4	E6DE857AFBDC6102	2020-12-07	EU	PC	B	interface_eu_test	NaN	NaT	NaT
...	...	...	...	...	...	...	...	...	...
18534	27007FC1A9B62FC5	2020-12-20	EU	iPhone	B	interface_eu_test	NaN	NaT	NaT
18535	89CB0BFBC3F35126	2020-12-20	EU	PC	B	recommender_system_test	NaN	NaT	NaT
18536	75F25D4DADA37ABB	2020-12-20	EU	Android	B	interface_eu_test	NaN	NaT	NaT
18537	2C29721DDDA76B2A	2020-12-20	EU	iPhone	B	interface_eu_test	NaN	NaT	NaT

	user_id	first_date	region	device	group	ab_test	name	start_dt	finish_dt
18538	1C7D23927835213F	2020-12-20	EU	iPhone	B	interface_eu_test	NaN	NaT	NaT

18539 rows × 9 columns

In [48]:
```
df = pd.merge(df,final_ab_events,on=['user_id'], how = 'left')
df
```

Out[48]:

	user_id	first_date_x	region_x	device_x	group_x	ab_test_x	name	start_dt	finish_dt	event_dt	event_name	details	f
0	D72A72121175D8BE	2020-12-07	EU	PC	A	recommender_system_test	NaN	NaT	NaT	2020-12-07 21:52:10	product_page	0.0	
1	D72A72121175D8BE	2020-12-07	EU	PC	A	recommender_system_test	NaN	NaT	NaT	2020-12-07 21:52:07	login	0.0	
2	2E1BF1D4C37EA01F	2020-12-07	EU	PC	A	interface_eu_test	NaN	NaT	NaT	NaT	NaN	NaN	
3	50734A22C0C63768	2020-12-07	EU	iPhone	B	interface_eu_test	NaN	NaT	NaT	NaT	NaN	NaN	
4	E6DE857AFBDC6102	2020-12-07	EU	PC	B	recommender_system_test	NaN	NaT	NaT	NaT	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	
27030	27007FC1A9B62FC5	2020-12-20	EU	iPhone	B	interface_eu_test	NaN	NaT	NaT	NaT	NaN	NaN	
27031	89CB0BFBC3F35126	2020-12-20	EU	PC	B	recommender_system_test	NaN	NaT	NaT	NaT	NaN	NaN	
27032	75F25D4DADA37ABB	2020-12-20	EU	Android	B	interface_eu_test	NaN	NaT	NaT	NaT	NaN	NaN	
27033	2C29721DDDA76B2A	2020-12-20	EU	iPhone	B	interface_eu_test	NaN	NaT	NaT	NaT	NaN	NaN	
27034	1C7D23927835213F	2020-12-20	EU	iPhone	B	interface_eu_test	NaN	NaT	NaT	NaT	NaN	NaN	

27035 rows × 19 columns

Не все маркетинговые события подходят к нашей выборке

- Christmas&New Year Promo, CIS New Year Gift Lottery проходят в период теста

- Christmas&New Year Promo точно повлиял на результаты, что нельзя сказать о New Year Gift Lottery так как мы не располагаем данными по значительному времени данной акции

Проведите оценку результатов A/B-тестирования

- H0 - группы, которые используют Mac и PC в европе различаются
- H1 - группы статистически одинаковы

In [49]:
```
final_ab_events['group'] = [1 if x == 'A' else 2 for x in final_ab_events['group']]
```

In [50]:
```
final_ab_events_group = final_ab_events.groupby(['ab_test','group'], as_index=False).agg({'user_id':'count'}).sort_values(['ab_tes
final_ab_events_group
```

Out[50]:

	ab_test	group	user_id
0	recommender_system_test	1	8027
1	recommender_system_test	2	2713

In [51]:
```
test = final_ab_events.pivot_table(index='event_name', columns='group',values='user_id',aggfunc='nunique')
test
```

Out[51]:

group	1	2
event_name		
login	1651	592
product_cart	514	170
product_page	1072	330
purchase	517	167

In [52]:
```
final_ab_events_copy = final_ab_events_copy.query("event_dt > '2020-12-21' and event_dt < '2020-01-04'")
```

In [53]:
```
test1 = final_ab_events_copy.pivot_table(index='event_name', columns='group',values='user_id',aggfunc='nunique')
test1
```

Out[53]:

event_name

In [54]:
```
test.sum()
```

Out[54]:
```
group
1    3754
2    1259
dtype: int64
```

In [55]:
```
people = final_ab_events.groupby('group')['user_id'].nunique()
users = people.to_frame().reset_index()
users = users.set_index(users.columns[0])
users
```

Out[55]:

	user_id
group	
1	1651
2	593

- H0 - между группами А и Б нет различимой разницы
- H1 - выборки отличаются между собой

In [56]:
```
def z_test(groupA, groupB, event, alpha):
    p1_ev = test.loc[event, groupA]
    p2_ev = test.loc[event, groupB]
    p1_us = users.loc[groupA, 'user_id']
    p2_us = users.loc[groupB, 'user_id']
    p1 = p1_ev / p1_us
    p2 = p2_ev / p2_us
    difference = p1 - p2
    p_combined = (p1_ev + p2_ev) / (p1_us + p2_us)
```

```
    z_value = difference / mth.sqrt(p_combined * (1 - p_combined) * (1 / p1_us + 1 / p2_us))
    distr = st.norm(0, 1)
    p_value = ((1 - distr.cdf(abs(z_value))) * 2)
    print('Проверка для  {} и {}, событие: {}, p-значение: {p_value:.2f}'.format(groupA, groupB, event, p_value=p_value))
    if (p_value < alpha):
        print("Отвергаем нулевую гипотезу")
    else:
        print("Не получилось отвергнуть нулевую гипотезу")
```

In [57]:

```
for event in test.index:
    z_test(1, 2, event, 0.0125)
    print()
```

```
Проверка для  1 и 2, событие: login, p-значение: 0.10
Не получилось отвергнуть нулевую гипотезу

Проверка для  1 и 2, событие: product_cart, p-значение: 0.26
Не получилось отвергнуть нулевую гипотезу

Проверка для  1 и 2, событие: product_page, p-значение: 0.00
Отвергаем нулевую гипотезу

Проверка для  1 и 2, событие: purchase, p-значение: 0.15
Не получилось отвергнуть нулевую гипотезу
```

alpha/4 = 0.0125

Использовали поправку Бонферонни для множественных тестов

Выводы

- Самые большие прибыли в EU регионе, второй в рейтинге - N. America
- униркальных покупателей 19,5к - это на 200 человек больше чем предыдущее число в воронке, связано скорее всего с количеством товара или покупка в составе набора
- 2 пика по датам: если пик 23ого числа можно связать с Рождеством, то по пику 14ого числа у меня пока нет идей
- к концу месяца посещение падает - пользователи готовяться к Новому Году
- Гипотезы подтвердились

- группа A выглядит лучше

Вывод:

- пик по датам с 13 по 14 число - связано скорее с тем, что в этот период наступает рождество в Европе, далее прирост можно объяснить наступлением нового года
- отфильтровали данные:
    - по дате запуска: 2020-12-07 и дата остановки набора новых пользователей: 2020-12-21
    - пользователей с даты остановки набора новых пользователей: 2020-12-21 и дате остановки: 2021-01-04 в тесте нет
    - по региону EU
    - по рекомендуемому тесту recommender_system_test
- попадают 2 маркетинговых события, одно из которых влияет на наше исследование: Christmas&New Year Promo и CIS New Year Gift Lottery
    - CIS New Year Gift Lottery: не влияет, т.к. как отмечено выше, данных в тесте нет по датам данного события
- Мы проверили гипотезы:
    - Проверка для 1 и 2, событие: login, p-значение: 0.10 тест показал что пользователи, которые залогинились отличаются по группам
    - Проверка для 1 и 2, событие: product_cart, p-значение: 0.26 тест показал что просмотры карточек товаров отличаются по группам
    - Проверка для 1 и 2, событие: product_page, p-значение: 0.00 тест показал что просмотры корзины одинаковые по группам
    - Проверка для 1 и 2, событие: purchase, p-значение: 0.15 тест показал что покупки отличаются по группам

Рекомендации:

- Определить различие между тестируемыми группами А и Б
- Оставить приоритет за группой А, т.к. она "выигрывает" по приоритетным параматрам
- так как маркетинговые события сильно влияют на исследования в будущем надо проводить тесты по возможности исключая влияние маркетинговых событий