

Проект по A/B-тестированию

Постановка задачи

Цель проекта:

Провести оценку результатов A/B-теста.

Задачи проекта:

- 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: А (контрольная), В (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_cart`
 - покупки — `purchase`.
- Загрузите данные теста, проверьте корректность его проведения и проанализируйте полученные результаты.

План проектной работы

Этап 1: Выгрузка предобработка данных

Выгрузить все датасеты. Изучить общую информацию о датасетах, какого типа данные представлены и в каких количествах.

- Преобразовать типы данных, если это требуется?
- Выяснить присутствуют ли пропущенные значения и дубликаты? Если да, то какова их природа?

Этап 2: Оценка корректности проведения теста.

- Проверить соответствие данных требованиям технического задания. Проверьте корректность всех пунктов технического задания.
- Проанализировать время проведения теста. Убедитесь, что оно не совпадает с маркетинговыми и другими активностями.
- Изучить аудиторию теста. Удостоверьтесь, что нет пересечений с конкурирующим тестом и нет пользователей, участвующих в двух группах теста одновременно. Проверьте равномерность распределения пользователей по тестовым группам и правильность их формирования.

Этап 3: Исследовательский анализ данных

- Количество событий на пользователя одинаково распределены в выборках?
- Как число событий в выборках распределено по дням?
- Как меняется конверсия в воронке в выборках на разных этапах?
- Какие особенности данных нужно учесть, прежде чем приступить к A/B-тестированию?

Этап 4: Оценка результатов A/B-теста.

- Что можно сказать про результаты A/B-тестирования?
- Проверьте статистическую разницу долей z-критерием

Этап 5: Вывод

- Опишите выводы по этапу исследовательского анализа данных и по проведённой оценке результатов A/B-тестирования.
- Сделайте общее заключение о корректности проведения теста.

Проектная работа

Выгрузка предобработка данных

```
In [10]: import pandas as pd
from scipy import stats as st
import datetime as dt
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from math import sqrt
import plotly.express as px
from plotly import graph_objects as go
import math as mth
from statsmodels.stats.proportion import proportions_ztest
```

```
In [11]: marketing_events = pd.read_csv('ab_project_marketing_events.csv', parse_dates = ['start_dt', 'finish_dt'])
users = pd.read_csv('final_ab_new_users.csv', parse_dates = ['first_date'])
events = pd.read_csv('final_ab_events.csv', parse_dates = ['event_dt'])
participants = pd.read_csv('final_ab_participants.csv')
```

```
In [12]: marketing_events.info()
marketing_events.head()
```

```
<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    datetime64[ns]
3   finish_dt    14 non-null    datetime64[ns]
dtypes: datetime64[ns](2), object(2)
memory usage: 576.0+ bytes
```

```
Out[12]:
```

	name	regions	start_dt	finish_dt
0	Christmas&New Year Promo	EU, N.America	2020-12-25	2021-01-03
1	St. Valentine's Day Giveaway	EU, CIS, APAC, N.America	2020-02-14	2020-02-16
2	St. Patric's Day Promo	EU, N.America	2020-03-17	2020-03-19
3	Easter Promo	EU, CIS, APAC, N.America	2020-04-12	2020-04-19
4	4th of July Promo	N.America	2020-07-04	2020-07-11

Перед нами датасет marketing_events, состоящий из 4 столбцов и 14 строк. Пропусков не обнаружено. Данные содержат информацию о маркетинговых событиях на 2020 год.

```
In [13]: users.info()
users.head()
```

```
<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  datetime64[ns]
2   region      61733 non-null  object
3   device      61733 non-null  object
dtypes: datetime64[ns](1), object(3)
memory usage: 1.9+ MB
```

```
Out[13]:
```

	user_id	first_date	region	device
0	D72A72121175D8BE	2020-12-07	EU	PC
1	F1C668619DFE6E65	2020-12-07	N.America	Android
2	2E1BF1D4C37EA01F	2020-12-07	EU	PC
3	50734A22C0C63768	2020-12-07	EU	iPhone
4	E1BDDCE0DAFA2679	2020-12-07	N.America	iPhone

Перед нами датасет users, состоящий из 4 столбцов и 61733 строк. Пропусков не обнаружено. Данные содержат информацию о всех пользователях, зарегистрировавшихся в интернет-магазине в период с 7 по 21 декабря 2020 года.

```
In [14]: events.info()
events.head()

<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  datetime64[ns]
2   event_name  440317 non-null  object
3   details     62740 non-null   float64
dtypes: datetime64[ns](1), float64(1), object(2)
memory usage: 13.4+ MB
```

```
Out[14]:
```

	user_id	event_dt	event_name	details
0	E1BDDCE0DAFA2679	2020-12-07 20:22:03	purchase	99.99
1	7B6452F081F49504	2020-12-07 09:22:53	purchase	9.99
2	9CD9F34546DF254C	2020-12-07 12:59:29	purchase	4.99
3	96F27A054B191457	2020-12-07 04:02:40	purchase	4.99
4	1FD7660FDF94CA1F	2020-12-07 10:15:09	purchase	4.99

Перед нами датасет events, состоящий из 4 столбцов и 440317 строк. В столбце details есть пропуски. Данные содержат информацию о всех событиях новых пользователей в период с 7 декабря 2020 по 4 января 2021 года.

```
In [15]: participants.info()
participants.head()

<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
```

```
Out[15]:
```

	user_id	group	ab_test
0	D1ABA3E2887B6A73	A	recommender_system_test
1	A7A3664BD6242119	A	recommender_system_test
2	DABC14FDDFADD29E	A	recommender_system_test
3	04988C5DF189632E	A	recommender_system_test
4	482F14783456D21B	B	recommender_system_test

Перед нами датасет participants, состоящий из 3 столбцов и 18268 строк. Пропусков не обнаружено. Данные содержат информацию о участниках тестов.

Мы выгрузили 4 датасета, преобразование типов данных не требуется, названия колонок имеют стандартную форму. Удалим явные дубликаты и рассмотрим колонку details в events. Поскольку эта колонка содержит дополнительные данные о событии, то оставляем пропуски, так как не каждое событие содержит дополнительные данные.

```
In [16]: marketing_events = marketing_events.drop_duplicates()
users = users.drop_duplicates()
events = events.drop_duplicates()
participants = participants.drop_duplicates()
```

Оценка корректности проведения теста.

Проверить соответствие данных требованиям технического задания. Проверьте корректность всех пунктов технического задания.

Техническое задание

- Название теста: `recommender_system_test`;
- группы: А — контрольная, В — 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 пользователи покажут улучшение каждой метрики не менее, чем на 10%:
 - конверсии в просмотр карточек товаров — событие `product_page`,
 - просмотры корзины — `product_cart`,
 - покупки — `purchase`.

```
In [17]: #оставляем лишь участников теста и фильтруем сроки их регистрации
users = users[users['first_date'] <= '2020-12-21']

participants_rec = participants.query('ab_test == "recommender_system_test"')
participants_rec_upd = participants_rec['user_id']

users_rec = users.query('user_id in @participants_rec_upd')
users_upd = users.query('user_id in @participants_rec')
users_upd = users[users['first_date'] <= '2020-12-21']

users_min = users['first_date'].min()
users_max = users['first_date'].max()
display(f'Дата первой регистрации: {users_min}, Дата последней регистрации: {users_max}')
```

'Дата первой регистрации: 2020-12-07 00:00:00, Дата последней регистрации: 2020-12-21 00:00:00'

```
In [18]: #оставляем лишь участников теста
events = events[events['event_dt'] <= '2021-01-04']

events_min = events['event_dt'].min()
events_max = events['event_dt'].max()
display(f'Дата первого события: {events_min}, Дата последнего события: {events_max}')
```

'Дата первого события: 2020-12-07 00:00:33, Дата последнего события: 2020-12-30 23:36:33'

```
In [19]: #проверка на 15% новых пользователей из EU
users_upd_grpb = users_rec.groupby('region')['user_id'].count().reset_index()
display(users_upd_grpb)

regions_grp = users_upd.groupby('region')['user_id'].count().reset_index()

display('Процент новых пользователей из EU:', users_upd_grpb.query('region == "EU"')['user_id'] / regions_grp.qu
```

	region	user_id
0	APAC	72
1	CIS	55
2	EU	6351
3	N.America	223

'Процент новых пользователей из EU:'
2 15.0
Name: user_id, dtype: float64

Последним пунктом ТЗ мы имеем несостыковку. Планируемая дата остановки - 2021-01-04, в то время как в реальности последнее событие датируется 2020-12-30. Из-за этого мы не получаем ожидаемого времени со дня регистрации - 14 дней. Для пользователей, зарегистрировавшихся в последний день по ТЗ 2020-12-21 имеем лайфтайм 9 дней. Отфильтруем события, совершенные не позже 14 дней.

```
In [20]: reg = users_rec[['user_id', 'first_date']]

events_upd = events.merge(reg, on = 'user_id')
```

```
events_upd['days'] = (events_upd['event_dt'] - events_upd['first_date']).dt.days
events_upd.query('days > 14')
```

Out[20]:

		user_id	event_dt	event_name	details	first_date	days
28	2B06EB547B7AAD08	2020-12-22 17:27:15	purchase	4.99	2020-12-07	15	
31	2B06EB547B7AAD08	2020-12-22 17:27:17	product_cart	NaN	2020-12-07	15	
34	2B06EB547B7AAD08	2020-12-22 17:27:15	product_page	NaN	2020-12-07	15	
37	2B06EB547B7AAD08	2020-12-22 17:27:15	login	NaN	2020-12-07	15	
55	538F954F6B3AECE4	2020-12-27 07:09:20	purchase	4.99	2020-12-07	20	
...	
23169	016F758EB5C5A5DA	2020-12-29 07:51:00	login	NaN	2020-12-13	16	
23211	E2F981AE3D6A2CE8	2020-12-29 15:02:21	login	NaN	2020-12-14	15	
23232	9E537FA089BAECFE	2020-12-29 14:52:33	login	NaN	2020-12-14	15	
23372	D0D1426BAFDD1FE5	2020-12-29 09:53:34	login	NaN	2020-12-14	15	
23405	9745CE1D5B111CB1	2020-12-29 19:10:57	login	NaN	2020-12-14	15	

628 rows × 6 columns

In [21]:

```
amount_of_events = events_upd.groupby('days')['user_id'].count().reset_index()
amount_of_events['%'] = round(amount_of_events['user_id'] / amount_of_events['user_id'].sum() * 100, 2)
amount_of_events
```

Out[21]:

	days	user_id	%
0	0	8131	32.92
1	1	3754	15.20
2	2	2578	10.44
3	3	1805	7.31
4	4	1494	6.05
5	5	1208	4.89
6	6	1070	4.33
7	7	986	3.99
8	8	762	3.09
9	9	606	2.45
10	10	517	2.09
11	11	347	1.40
12	12	347	1.40
13	13	251	1.02
14	14	214	0.87
15	15	171	0.69
16	16	94	0.38
17	17	86	0.35
18	18	80	0.32
19	19	60	0.24
20	20	69	0.28
21	21	33	0.13
22	22	31	0.13
23	23	4	0.02

Как мы видим в данных есть 628 событий, совершенных позже 14 дней со дня регистрации. Также мы наблюдаем, что практически 60% всех событий происходят в первые 3 дня с момента регистрации. Учитывая это можем предположить, что горизонт событий в 14 дней не настолько принципиален, как это просится в ТЗ. Отфильтруем данные по этим значениям.

In [22]:

```
#фильтруем данные по дате регистрации
events_upd = events_upd.query('days <= 14')

#разделяем пользователей в зависимости от группы
participants_upd_a = participants_rec.query('group == "A"')['user_id']
participants_upd_b = participants_rec.query('group == "B"')['user_id']
```

In [23]:

```
#разделяем пользователей на группы А и В
events_upd_a = events_upd.query('user_id in @participants_upd_a')
```

```
events_upd_b = events_upd.query('user_id in @participants_upd_b')
```

Время проведения теста. Убедитесь, что оно не совпадает с маркетинговыми и другими активностями.

```
In [24]: marketing_events.sort_values(by = 'start_dt')
```

	name	regions	start_dt	finish_dt
6	Chinese New Year Promo	APAC	2020-01-25	2020-02-07
1	St. Valentine's Day Giveaway	EU, CIS, APAC, N.America	2020-02-14	2020-02-16
8	International Women's Day Promo	EU, CIS, APAC	2020-03-08	2020-03-10
2	St. Patric's Day Promo	EU, N.America	2020-03-17	2020-03-19
3	Easter Promo	EU, CIS, APAC, N.America	2020-04-12	2020-04-19
7	Labor day (May 1st) Ads Campaign	EU, CIS, APAC	2020-05-01	2020-05-03
9	Victory Day CIS (May 9th) Event	CIS	2020-05-09	2020-05-11
11	Dragon Boat Festival Giveaway	APAC	2020-06-25	2020-07-01
4	4th of July Promo	N.America	2020-07-04	2020-07-11
13	Chinese Moon Festival	APAC	2020-10-01	2020-10-07
12	Single's Day Gift Promo	APAC	2020-11-11	2020-11-12
5	Black Friday Ads Campaign	EU, CIS, APAC, N.America	2020-11-26	2020-12-01
0	Christmas&New Year Promo	EU, N.America	2020-12-25	2021-01-03
10	CIS New Year Gift Lottery	CIS	2020-12-30	2021-01-07

Как мы наблюдаем время проведения теста совпадает с Christmas&New Year Promo. Последние 5 дней теста с 2020-12-25 по 2020-12-30 проводились параллельно с новогодним промо.

```
In [25]: proc = events_upd[events_upd['event_dt'] >= '2020-12-25']['event_name'].count() / events_upd['user_id'].count()  
display(f'Процент событий во время промо от всех событий: {proc:.1%}')
```

'Процент событий во время промо от всех событий: 11.8%'

Аудитория теста. Удостоверьтесь, что нет пересечений с конкурирующим тестом и нет пользователей, участвующих в двух группах теста одновременно. Проверьте равномерность распределения по тестовым группам и правильность их формирования.

```
In [26]: users_eu = users_rec.query('region == "EU"')  
table = users_eu.merge(events, on = 'user_id')  
table = table.merge(participants_rec, on = 'user_id')  
table
```

	user_id	first_date	region	device	event_dt	event_name	details	group	ab_test
0	D72A72121175D8BE	2020-12-07	EU	PC	2020-12-07 21:52:10	product_page	NaN	A	recommender_system_test
1	D72A72121175D8BE	2020-12-07	EU	PC	2020-12-07 21:52:07	login	NaN	A	recommender_system_test
2	DD4352CDCF8C3D57	2020-12-07	EU	Android	2020-12-07 15:32:54	product_page	NaN	B	recommender_system_test
3	DD4352CDCF8C3D57	2020-12-07	EU	Android	2020-12-08 08:29:31	product_page	NaN	B	recommender_system_test
4	DD4352CDCF8C3D57	2020-12-07	EU	Android	2020-12-10 18:18:27	product_page	NaN	B	recommender_system_test
...
23415	0416B34D35C8C8B8	2020-12-20	EU	Android	2020-12-21 22:28:29	product_page	NaN	A	recommender_system_test
23416	0416B34D35C8C8B8	2020-12-20	EU	Android	2020-12-24 09:12:51	product_page	NaN	A	recommender_system_test
23417	0416B34D35C8C8B8	2020-12-20	EU	Android	2020-12-20 20:58:25	login	NaN	A	recommender_system_test
23418	0416B34D35C8C8B8	2020-12-20	EU	Android	2020-12-21 22:28:29	login	NaN	A	recommender_system_test
23419	0416B34D35C8C8B8	2020-12-20	EU	Android	2020-12-24 09:12:49	login	NaN	A	recommender_system_test

23420 rows × 9 columns

```
In [27]: #расчет пользователей, входящих в оба теста  
participants_two_tests = participants.groupby('user_id')['ab_test'].count().reset_index()  
participants_two_tests = participants_two_tests.query('ab_test > 1')['user_id']  
count_of_two_tests = table.query('user_id in @participants_two_tests')['user_id'].nunique()  
display(f'Количество пользователей, которые участвовали в обоих тестах: {count_of_two_tests}')
```

```
#расчет пользователей, входящих в обе группы  
participants_two_grps = participants_rec.groupby('user_id')['group'].count().reset_index()  
participants_two_grps = participants_two_grps.query('group > 1')['user_id']  
count_of_two_grps = table.query('user_id in @participants_two_grps')['user_id'].nunique()  
display(f'Количество пользователей, которые были в обеих группах: {count_of_two_grps}')
```

'Количество пользователей, которые участвовали в обоих тестах: 887'

```
In [28]: table.groupby('group')['user_id'].nunique()
```

```
Out[28]: group
A      2604
B       877
Name: user_id, dtype: int64
```

Как мы видим в группу А входят 2604 человек, а в группу В 887. Разница достаточно большая и мы можем предположить, что размер выборки В будет недостаточен для получения статистически значимых результатов. Более того в данных есть 887 участников, которые были в обеих группах теста. Это тоже неблагоприятное условие для теста, так как одна группа может исказить результаты другой группы. Оценим в какие группы другого теста попали пользователи.

```
In [29]: another_test_grps = table.query('user_id in @participants_two_tests')
         another_test_grps.groupby('group')['user_id'].nunique()
```

```
Out[29]: group
A      665
B      222
Name: user_id, dtype: int64
```

В группе А 665 человек из 2604 принимали участие в обоих тестах, в группе В принимали участие 222 из 877. Получается, что 25% пользователей из каждой группы принимали участие в двух тестах. Это достаточно большая часть выборки, однако удалять этих пользователей из теста не стоит, так как это еще сильнее уменьшит мощность теста. Учтем, что количество таких людей распределено каждой группе равномерно по 25%.

Исследовательский анализ данных

Количество событий на пользователя одинаково распределены в выборках?

```
In [30]: #медианное количество событий для участников группы A
table.query('group == "A"]').groupby('user_id')['event name'].count().median()
```

```
Out[30]: 6.0
```

```
In [31]: #медианное количество событий для участников группы В
table.query('group == "B"').groupby('user_id')['event name'].count().median()
```

```
Out[31]: 6.0
```

Как мы наблюдаем, несмотря на большое различие в размерах выборок медианное количество событий для каждого участника одинаково - 6 событий.

Как число событий в выборках распределено по дням?

```
In [32]: #медианное количество событий в день для участников группы A
table.query('group == "A"]').groupby('first date')['event name'].count().median()
```

```
Out[32]: 1180.0
```

```
In [33]: #медианное количество событий в день для участников группы В
table.query('group == "B").groupby('first date')['event name'].count().median()
```

Out[33]: 242.0

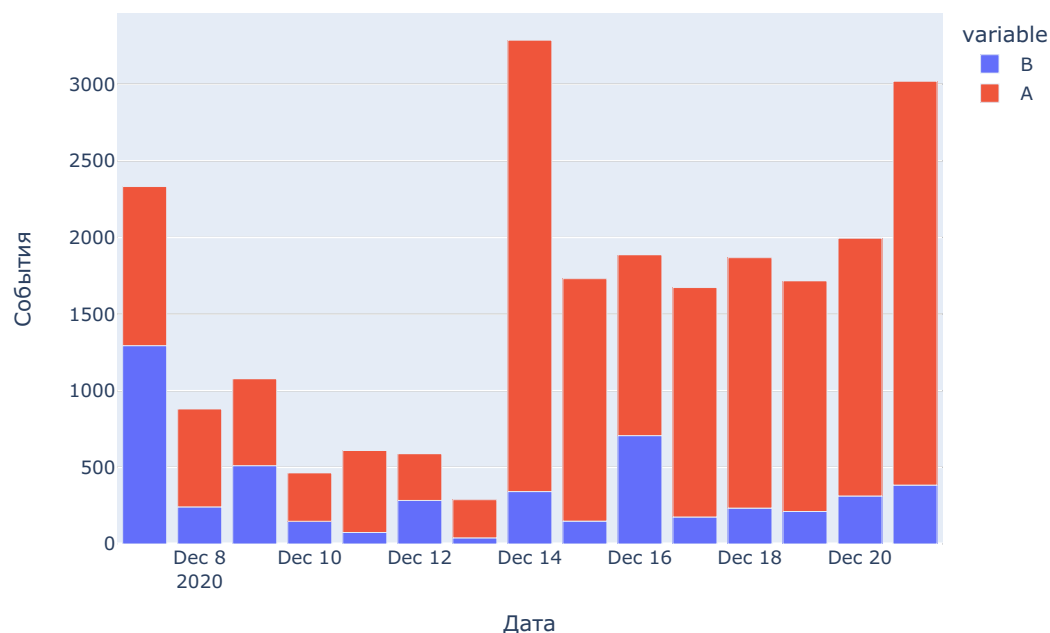
```
In [34]: #таблица событий по дням среди групп
events_grp = table.pivot_table(index = 'first_date', columns = 'group', values = 'event_name', aggfunc = 'count',
display(events_grp)

fig = px.bar(events_grp, y = ['B','A'], x = 'first_date', title='События по дням среди групп')
fig.update_layout(xaxis_title="Дата",
yaxis_title="События",)
fig.show()
```

group	first_date	A	B
0	2020-12-07	1038	1294
1	2020-12-08	639	242
2	2020-12-09	567	511
3	2020-12-10	316	148
4	2020-12-11	535	75
5	2020-12-12	305	284
6	2020-12-13	249	40
7	2020-12-14	2945	342
8	2020-12-15	1582	149
9	2020-12-16	1180	707
10	2020-12-17	1497	176
11	2020-12-18	1634	234
12	2020-12-19	1505	212
13	2020-12-20	1682	313
14	2020-12-21	2635	384



События по дням среди групп



Как мы видим по таблице и графику событий в группе A в разы больше, чем событий в группе B.

Как меняется конверсия в воронке в выборках на разных этапах?

```
In [35]: test = table.pivot_table(index='event_name', columns='group', values='user_id', aggfunc='nunique').reset_index(
display(test)

fig = go.Figure()

fig.add_trace(go.Funnel(
    name = 'A',
    y = test['event_name'],
    x = test['A'],
    textinfo = "value+percent initial"))
fig.add_trace(go.Funnel(
    name = 'B',
    orientation = "h",
    y = test['event_name'],
    x = test['B'],
    textposition = "inside",
    textinfo = "value+percent initial"))
fig.update_layout(title = 'Конверсия событий для групп', xaxis_title="Количество событий", yaxis_title="Действия")
fig.show()
```


group	event_name	A	B
0	login	2604	877
1	product_cart	782	244
2	product_page	1685	493
3	purchase	833	249

Как мы видим конверсия в воронке убывает для тестируемой группы. В изначальной группе А показатели выше по каждому этапу. На 9% выше product_page, на 4% выше purchase и на 2% выше product_cart.

Какие особенности данных нужно учесть, прежде чем приступить к A/B-тестированию?

- Корректность данных относительно ТЗ.
- Время проведения теста 14 дней.
- Трехкратное различие в размерах выборок. Группа А больше тестируемой группы.
- Время проведения теста в последние 5 дней совпадает с началом события Christmas&New Year Promo.
- Среди пользователей есть участники обеих групп(А и В).
- Исходя из получившихся воронок группа А показывает конверсию больше, чем группа В.

Оценка результатов A/B-теста.

Проверьте статистическую разницу долей z-критерием

```
In [36]: users_groups = table.groupby('group')['user_id'].unique().reset_index()

test['users_A'] = users_groups.query('group == "A"')['user_id'].max()
test['users_B'] = users_groups.query('group == "B"')['user_id'].max()

#составляем функцию для проведения z-теста и проверки статистической значимости между группами
def proportion_test(successes, leads):
    # пропорция переходов в первой группе:
    p1 = successes[0]/leads[0]
    # пропорция переходов во второй группе:
    p2 = successes[1]/leads[1]
    # пропорция успехов в комбинированном датасете:
    p_combined = (successes[0] + successes[1]) / (leads[0] + leads[1])
    # разница пропорций в датасетах
    difference = p1 - p2
    z_value = difference / mth.sqrt(p_combined * (1 - p_combined) * (1/leads[0] + 1/leads[1]))
```

```

distr = st.norm(0, 1)
p_value = (1 - distr.cdf(abs(z_value))) * 2
print(successes)
print('p-value: ', p_value)
if p_value < alpha:
    print('Отвергаем нулевую гипотезу: между долями есть значимая разница\n')
else:
    print(
        'Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными\n')
alpha = 0.05 / 4

print('H0 - доли уникальных посетителей, побывавших на этапе воронки, одинаковы.')
print('H1 - между долями уникальных посетителей, побывавших на этапе воронки, есть значимая разница. \n')
for idx, event in enumerate(test['event_name']):
    successes = np.array([test['A'][idx], test['B'][idx]])
    leads = np.array([test['users_A'][idx], test['users_B'][idx]])
    print(event)
    print(' A      B')
    proportion_test(successes, leads)

```

H0 - доли уникальных посетителей, побывавших на этапе воронки, одинаковы.

H1 - между долями уникальных посетителей, побывавших на этапе воронки, есть значимая разница.

```

login
  A      B
[2604  877]
p-value: nan
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

```

```

product_cart
  A      B
[782 244]
p-value: 0.21469192029582396
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

```

```

product_page
  A      B
[1685  493]
p-value: 6.942739359416805e-06
Отвергаем нулевую гипотезу: между долями есть значимая разница

```

```

purchase
  A      B
[833 249]
p-value: 0.04652482738393027
Не получилось отвергнуть нулевую гипотезу, нет оснований считать доли разными

```

```

C:\Users\Admin\AppData\Local\Temp\ipykernel_9404\156454465.py:17: RuntimeWarning:
invalid value encountered in double_scalars

```

Что можно сказать про результаты A/B-тестирования?

Было проведено 4 проверки гипотез, где сравнивались группы А и В. Вся эта проверка образовала множественный тест. Нами был установлен уровень значимости - 5%. При таком уровне значимости шанс ошибиться и получить ложнопозитивный результат составляет примерно 18.54%. На основе z-testa мы оставляем все гипотезы, кроме гипотезы на событии purchase. Значит статистически значимое различие между группами все же есть.

Вывод

Нами была составлена воронка продаж. Был проведен A/B-тест с проверкой 4 гипотез и 2 групп. На основе результатов исследования мы можем сделать вывод, что статистически значимое различие между группами все же присутствует, а значит тестирование было проведено при некорректных данных. Стоит увеличить выборки и подвести их примерно к одному размеру. Важно провести тест в свободное от других маркетинговых событий время. Воронка продаж показала, что группа А превосходит группу В, поэтому рекомендую обдумать преобразование тестируемой группы и сделать ее более отличающейся от группы А.