

Анализ дизайна эксперимента

N п/п	Ошибка	Предложение
1	Не определена продолжительность эксперимента	Стоит определять продолжительность до начала эксперимента, а также убедиться, что в период проведения не будет событий, которые бы могли повлиять на результаты эксперимента (например, объявления локдауна или изменений в сервисе доставки)
2	Не определен размер тестовой выборки	Для получения статистически значимого результата стоит определять выборку заранее, либо следить за результатами по ходу эксперимента и принимать решение о его продолжении или завершении
3	Не определены критерии по которым пользователи будут отбираться для контрольной и тестовой выборки	Для лучшей интерпретации результатов лучше определить сегменты пользователей. Так пользователи, которые регулярно пользуются услугой не будут сталкиваться с “кривыми” сценариями и на них изменения не повлияют.
4	Желаемые изменения в гипотезе описаны как “избегание “кривых” сценариев”. При этом в качестве метрики используется конверсия на добавление в корзину и показатель отказов.	Так как гипотеза не содержит предполагаемых изменений этих метрик, нельзя будет сделать выводы об эффективности изменений. Лучше в гипотезе указать эти изменения, например, выбор адреса до этапа выбора магазина повысит конверсию на 5-10%. Значение метрики также поможет в определении размера выборки.
5	В качестве метрики выбрана конверсия на добавление в корзину, но сервис зарабатывает не на этом, а на продаже товаров	Лучше добавить конверсию на оплату заказов в качестве основной метрики, а конверсию на добавление в корзину использовать в качестве дополнительной
6	Не определен уровень статистической значимости	Уровень связан с размером выборки и изменением метрик в гипотезе. В дизайне эксперимента лучше определять эти три параметра.

7	В дизайне эксперимента для тестовой группы предлагается открывать модальку для ввода адреса при вводе адреса	Это может быть неудобно пользователям и сбивать их. Лучше вместо поля для ввода использовать кнопку, при нажатии на которую открывать модальку для ввода (и именно так и сделано на проде).
---	--	---

Анализ данных эксперимента

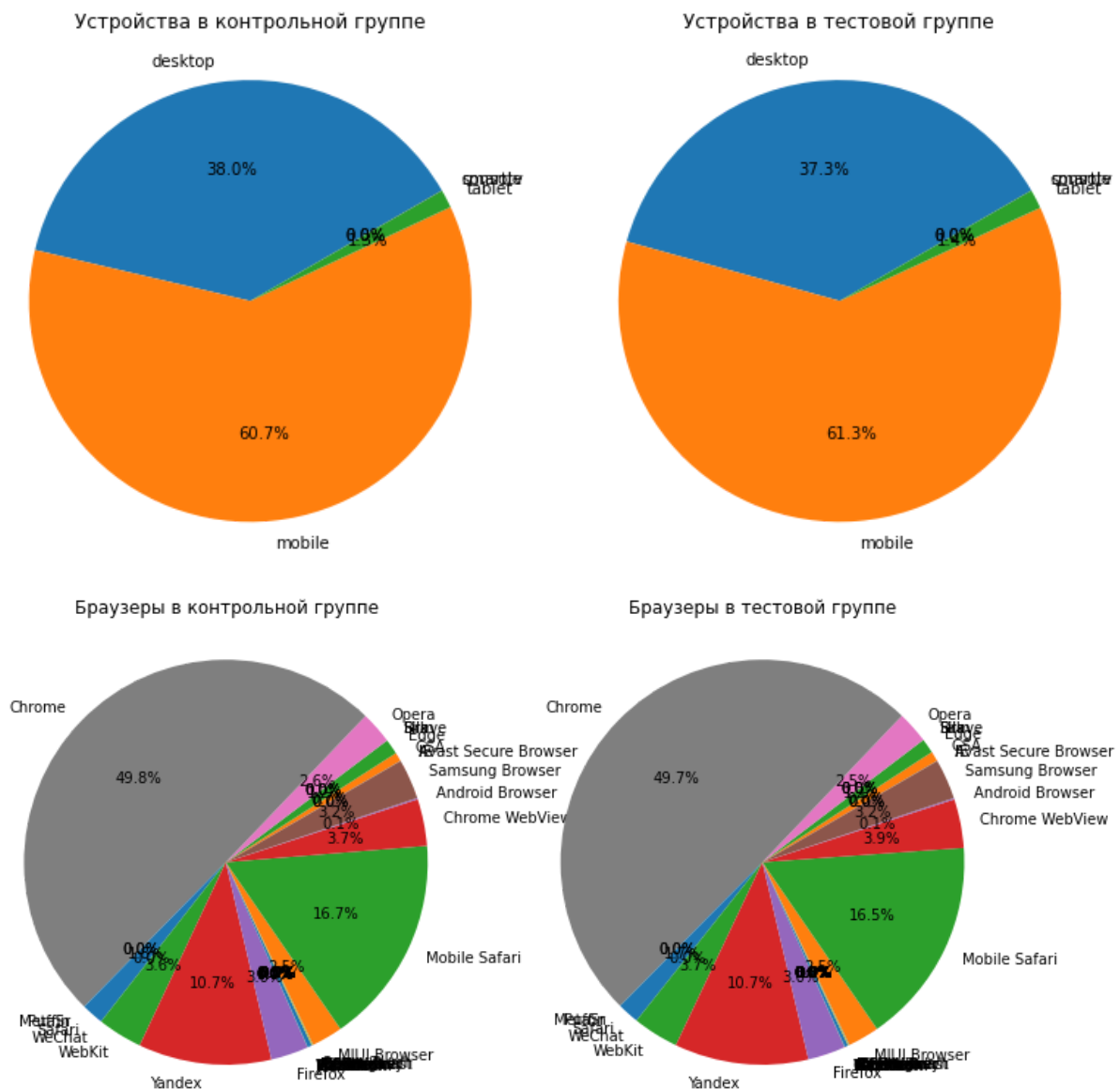
Последовательность анализа

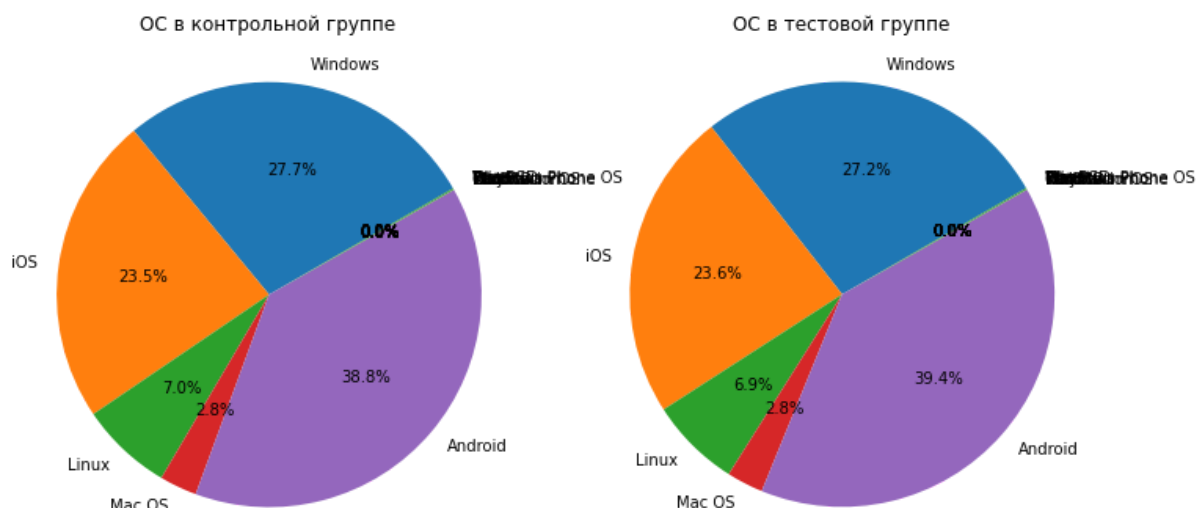
Шаг 1. Загрузим данные, зададим для полей с timestamp тип данных dateTime.

Шаг 2. Проверим, что в отдельных датасетах нет дубликатов. Если есть удалим их.

Шаг 3. Проверим, что распределение пользователей на контрольную и тестовую выборку равномерно.

Отношение количества пользователей в контрольной и тестовой выборке





Шаг 4. Проверим, что разделение на контрольную и тестовую группу проведено корректно, и одни и те же пользователи не входят в обе группы. Оказалось, что это не так и 528 пользователей попали в обе группы. Исключим записи о действиях этих пользователей из всех датасетов. Тогда в контрольной группе остается 434727 пользователей, в тестовой - 54820.

Шаг 5. Объединим датасеты в один, в котором каждая запись будет представлять собой действие пользователя с указанием времени и типа этого действия.

Шаг 6. Посчитаем метрики:

Метрика	Метод расчета	Значение для тестовой группы	Значение для контрольной группы
Конверсия на добавление в корзину	Посчитаем количество уникальных пользователей, которые посмотрели лендинг. Затем посчитаем количество пользователей из числа посмотревших лендинг, которые добавляли товар в корзину или нажимали на кнопку "Добавить в корзину".	0.1738	0.1804
Конверсия на оформление заказа	Посчитаем количество уникальных пользователей, которые посмотрели лендинг. Затем посчитаем количество пользователей из числа посмотревших лендинг, которые оформили заказ.	0.0313	0.0292

Показатель отказов	Посчитаем количество уникальных пользователей, для которых есть только события с типом “Добавление в группу” и “Просмотр лендинга”. Разделим полученное число на количество уникальных пользователей просмотревших лендинг.	0.5209	0.5321
--------------------	---	--------	--------

Шаг 7. Рассчитаем p-value для каждой метрики, а также минимальный размер выборки, на которой может быть получен статистически значимый результат.

Метрика	P-value	Минимальный размер выборки
Конверсия на добавление в корзину	0.013	78141
Конверсия на оформление заказа	0.074	155330
Показатель отказов	0.001	46420

Выводы

1. Изменение лендинга снижает конверсию на добавление в корзину на 0.62%, но повышает конверсию на оформление заказа 0.21%.
2. При уровне значимости 0.01 (с вероятностью 1% изменения лендинга не приведут к изменению метрик) изменения обеих конверсий не являются статистически значимыми.
3. Изменения в лендинге приводят к статистически не значимому увеличению показателя отказа.
4. Таким образом, эксперимент следует считать неуспешным.

Аномалии в данных эксперимента

Аномалия	Дополнительно	Возможная причина
528 пользователя попали в обе группы	Добавление одного пользователя в обе группы	Распределенное добавление в группы?

	происходило в одно время	
56 пользователей добавлялись в группы несколько раз	Добавление одного пользователя в группы происходило в разные дни. Устройства/браузер/ОС не менялись	Нет проверки на добавление пользователя в группе?
Некоторые пользователи были добавлены в группы, но не совершали других действий	192802 - уникальных пользователей 195305 - записей о добавлении	
2576 пользователей оформили заказ, но не посмотрели лендинг	После добавления пользователи сразу просматривали страницу ритейлера	
20830 не указали адрес, но добавили товар в корзину		Записи об указании адреса могли потеряться
156 пользователей добавили товары в корзину без нажатия кнопки "Добавить в корзину"		Записи об указании адреса могли потеряться
26831 пользователей нажали на "Добавить в корзину", но товар не был добавлен	Из этих пользователей 11 оформили заказ	Записи о добавлении товара в корзину могли потеряться
134 пользователя оформили заказ без добавления товара в корзину (в том числе без нажатия кнопки "Добавить в корзину")		Записи о добавлении товара в корзину или нажатии кнопки "Добавить в корзину" могли потеряться

Потенциальные ошибки

1. Не проверить вхождение одних и тех же пользователей в разные группы.
2. Рассчитывает конверсию от числа пользователей в группах.
3. Рассчитывать конверсию на добавление в корзину от количества пользователей добавивших товар в корзину или нажавших на кнопку "Добавить в корзину" (часть записей об этих действиях скорее всего утеряна, нужно учитывать оба).

Код

Для анализа использовался Python 3.7 и Jupyter Notebook.

Загрузка данных на примере датасета с группами пользователей

```
# Группы пользователей
users_origin = pd.read_csv('./Career Factory/AB Test Hit.csv')
users_origin['hit_at'] = users_origin['hit_at'].astype('datetime64[ns]')
```

Удаление дублей на примере датасета с записями о просмотре лендинга

```
landing_view = landing_view.drop_duplicates(ignore_index = True)
```

Поиск одинаковых пользователей в обеих группах

```
# Проверим, что в контрольной и тестовой группах нет одинаковых пользователей
intersecting_users = users[users['group'] == 'default'].merge(users[users['group'] ==
'address_first'], on='anonymous_id', suffixes = ('_control', '_test'))

# Удалим пользователей, которые есть в контрольной и тестовой группе
duplicated_users =
users[users['anonymous_id'].isin(intersecting_users['anonymous_id'])]['anonymous_id']

users = users[~users['anonymous_id'].isin(intersecting_users)]
landing_view = landing_view[~landing_view['anonymous_id'].isin(duplicated_users)]
...
```

Создание датасета с событиями

```
events = pd.DataFrame(columns=['anonymous_id', 'group', 'event_type', 'timestamp'])

group_dtype = pd.api.types.CategoricalDtype(categories=['default', 'address_first'],
ordered=True)
even_type_dtype = pd.api.types.CategoricalDtype(categories=['user_added',
'landing_viewed',
'retailer_page_viewed',
'address_change_initiated',
'address_not_in_delivery_zone',
'shop_selection_started',
'shop_selected',
'add_to_cart_clicked',
'product_added',
```

```

                                'order_completed'
                                ], ordered=True)

events.astype({"anonymous_id": object,
               "group": group_dtype,
               "event_type": even_type_dtype,
               "timestamp": 'datetime64[ns]'})

# Добавление пользователей в группы
users_ = users[['anonymous_id', 'group', 'hit_at']]
users_['event_type'] = 'user_added'
users_ = users_.rename(columns={"hit_at": "timestamp"})
events = events.append(users_)
...

```

Расчет конверсий

```

# Количество пользователей, которые посмотрели лендинг

landing_viewed_ = events[events['event_type'].isin(['landing_viewed'])]
landing_viewed_test = landing_viewed[landing_viewed['group']
== 'address_first']['anonymous_id'].unique()
landing_viewed_control = landing_viewed[landing_viewed['group']
== 'default']['anonymous_id'].unique()

# Количество пользователей, которые посмотрели лендинг, а затем добавили товар в корзину
product_added = events[events['event_type'].isin(['product_added', 'add_to_cart_clicked'])]

product_added_test =
product_added[product_added['anonymous_id'].isin(landing_viewed_test)]['anonymous_id'].
unique()

product_added_control =
product_added[product_added['anonymous_id'].isin(landing_viewed_control)]['anonymous_i
d'].unique()

print('Конверсия на добавление в корзину в тестовой группе',
product_added_test.shape[0]/landing_viewed_test.shape[0])
print('Конверсия на добавление в корзину в контрольной группе',
product_added_control.shape[0]/landing_viewed_control.shape[0])

# Количество пользователей, которые посмотрели лендинг, а затем оформили заказ
order_completed = events[events['event_type'].isin(['order_completed'])]
order_completed_test =
order_completed[order_completed['anonymous_id'].isin(landing_viewed_test)]['anonymous_i
d'].unique()

```

```
order_completed_control =  
order_completed[order_completed['anonymous_id'].isin(landing_viewed_control)]['anonymous_id'].unique()
```

```
print('Конверсия на оформление заказа в тестовой группе',  
order_completed_test.shape[0]/landing_viewed_test.shape[0])  
print('Конверсия на оформление заказа в контрольной группе',  
order_completed_control.shape[0]/landing_viewed_control.shape[0])
```

Расчет показателя отказов

```
active_users = events[events['event_type'].isin(['retailer_page_viewed',  
        'address_change_initiated', 'address_not_in_delivery_zone',  
        'shop_selection_started', 'add_to_cart_clicked', 'product_added',  
        'order_completed'])]['anonymous_id'].unique()
```

```
inactive_users_events = events[~events['anonymous_id'].isin(active_users)]
```

Тестовая группа

```
users_events_control = events[(events['event_type'] == 'landing_viewed') & (events['group']  
== 'default')]
```

```
inactive_users_events_control = inactive_users_events[(inactive_users_events['event_type']  
== 'landing_viewed') & (inactive_users_events['group'] == 'default')]
```

```
inactive_users_events_control['anonymous_id'].unique().shape[0]/users_events_control['anonymous_id'].unique().shape[0]
```

Контрольная группа

```
users_events_control = events[(events['event_type'] == 'landing_viewed') & (events['group']  
== 'default')]
```

```
inactive_users_events_control = inactive_users_events[(inactive_users_events['event_type']  
== 'landing_viewed') & (inactive_users_events['group'] == 'default')]
```

```
inactive_users_events_control['anonymous_id'].unique().shape[0]/users_events_control['anonymous_id'].unique().shape[0]
```

Расчет p-value

```
p1 = successes_A / trials_A
```

```
p2 = successes_B / trials_B
```

```
p_combined = (successes_A + successes_B) / (trials_A + trials_B)
```

```

difference = p1 - p2
z_value = difference / math.sqrt(abs(p_combined * (1 - p_combined) * (1/trials_A +
1/trials_B)))
distr = st.norm(0, 1)
p_value = (1 - distr.cdf(abs(z_value))) * 2

```

Расчет размера выборки

effect_size = sms.proportion_effectsize(x, y) # x и y - значения метрики для тестовой и контрольной группы

```

required_n = sms.NormalIndPower().solve_power(
    effect_size,
    power=0.8,
    alpha=0.01,
    ratio=1
)

```

```
required_n = ceil(required_n)
```

Поиск аномалий на примере поиска пользователей, добавленных в группы, но не выполнивших других действий

Пользователи, которые были добавлены в группы, но не сделали никаких действий

```

us1 = events[events['event_type'] == 'user_added']['anonymous_id']
us2 = events[events['event_type'].isin(['landing_viewed', 'retailer_page_viewed',
    'address_change_initiated', 'address_not_in_delivery_zone',
    'shop_selection_started', 'add_to_cart_clicked', 'product_added',
    'order_completed'])]['anonymous_id']
us1[~us1.isin(us2)].unique().shape

```

