

# Итоговый проект по курсу

А/В-тестирование:  
практическое руководство



# Анализ исходных данных

- Тестовый датасет был загружен в Jupyter Notebook и обработан с помощью библиотеки pandas
- Инсайты
  - Больше половины всех заказов были сделаны из центра города.
  - у Android конверсия хуже, чем на iOS
  - В сессиях, где `price_seen = 0` - нулевая конверсия в успешный заказ
  - Чем выше `order_class`, тем длиннее дистанция поездок
  - Больше половины всех отмен заказов приходится на пользователей с ОС Android.



# Формирование гипотез

№	Гипотеза	Источник	Почему не стоит проверять	Ваш личный рейтинг 1-10	Почему поставили такую оценку?
1	Если предлагать скидку 3 процента для поездки с перифирии в центр, то конверсия в количество заказов вырастет	Больше половины всех заказов были сделаны из центра города.	-	5	Вероятно, дело в количестве людей проживающих в центре и на перифирии. (в центре людей живет меньше, на перифирии больше)
2	Если сделать приложение на Android таким же как на iOS, то конверсия в количестве заказов увеличится	у Android конверсия хуже, чем на iOS	-	5	Возможно дело не в приложении, а в платежоспособности пользователей (телефоны на iOS стоят дороже, чем телефоны на Android)
3	Если всегда заранее показывать стоимость поездки клиенту, то конверсия в количестве заказов будет больше 0	В сессиях, где price_seen = 0 - нулевая конверсия в успешный заказ	-	10	В сессиях с показанной ценой высокая конверсия, с непоказанной - никакая
4	Если снизить стоимость поездок в период surge, то конверсия увеличится	Низкая конверсия в период surge	-	7	Цены не будут отталкивать пользователей
5	Если ввести небольшой штраф за отмену поездки на Android, то количество отмен уменьшится.	Больше половины всех отмен заказов приходится на пользователей с ОС Android.	-	8	Людам будет сложнее сделать отмену заказа, если они будут понимать, что потеряют какую-то часть своих денег.





# Подготовка к А/Б-тестированию

## 1. Выбрали особенность -

Для устройств Android наблюдаем более высокие юзерские отмены

Выбираем особенность - Для устройств Android наблюдаем более высокие юзерские отмены

```
In [5]: df_cancelled = df[df['user_cancelled']==1]
```

```
In [6]: df_cancelled.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11269 entries, 22 to 101493
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Unnamed: 0             11269 non-null  int64   
1   date                   11269 non-null  object  
2   user_id                11269 non-null  int64   
3   hour                   11269 non-null  int64   
4   app_opened             11269 non-null  int64   
5   price_seen             11269 non-null  int64   
6   order_made             11269 non-null  int64   
7   surge                   11269 non-null  object  
8   ride_completed         11269 non-null  int64   
9   user_cancelled         11269 non-null  int64   
10  age                     11269 non-null  int64   
11  os                      11269 non-null  object  
12  city_center_order      11269 non-null  int64   
13  order_class             11269 non-null  object  
14  distance                11269 non-null  float64 
15  rfm                     11269 non-null  object  
dtypes: float64(1), int64(10), object(5)
memory usage: 1.5+ MB
```

## 2. Разбили пользователей на группы.

```
In [11]: df_android_cancel['rand'] = 0.0

for i in range(len(df_android_cancel)):
    t = random.random()
    df_android_cancel['rand'].values[i] = t

df_android_cancel['group'] = df_android_cancel['rand'].apply(lambda x: '1' if x<0.5 else '2')
df_android_cancel
```

```
Out[11]:
```

	user_cancelled	rand	group
user_id			
100093	0.000000	0.925815	2
100118	0.500000	0.928886	2
100134	0.250000	0.864864	2
100166	0.250000	0.546697	2
100255	0.250000	0.911154	2
...	...	...	...
999870	0.500000	0.775410	2
999878	0.333333	0.085504	1
999945	0.000000	0.090418	1
999962	0.000000	0.699203	2



# Проведение А/Б-теста

3. Проверили группы 1 и 2 на однородность (Критерий Манна-Уитни).

## Проверка групп на однородность

$H_0$ : группы 1 и 2 однородны

$H_1$ : группы 1 и 2 не однородны

```
In [27]: stat, p = stats.mannwhitneyu(group_3['user_cancelled'], group_4['user_cancelled'])
print('Statistics=%.3f, p=%.3f' % (stat, p))

alpha = 0.05
if p > alpha:
    print('Однородны')
else:
    print('Не однородны')
```

Statistics=70456852.500, p=0.449  
Однородны

Таким образом не отклоняем  $H_0$



# Анализ результатов А/Б-теста

- Если юзер Android нажимает “отменить”, предлагать промокод в случае НЕ отмены

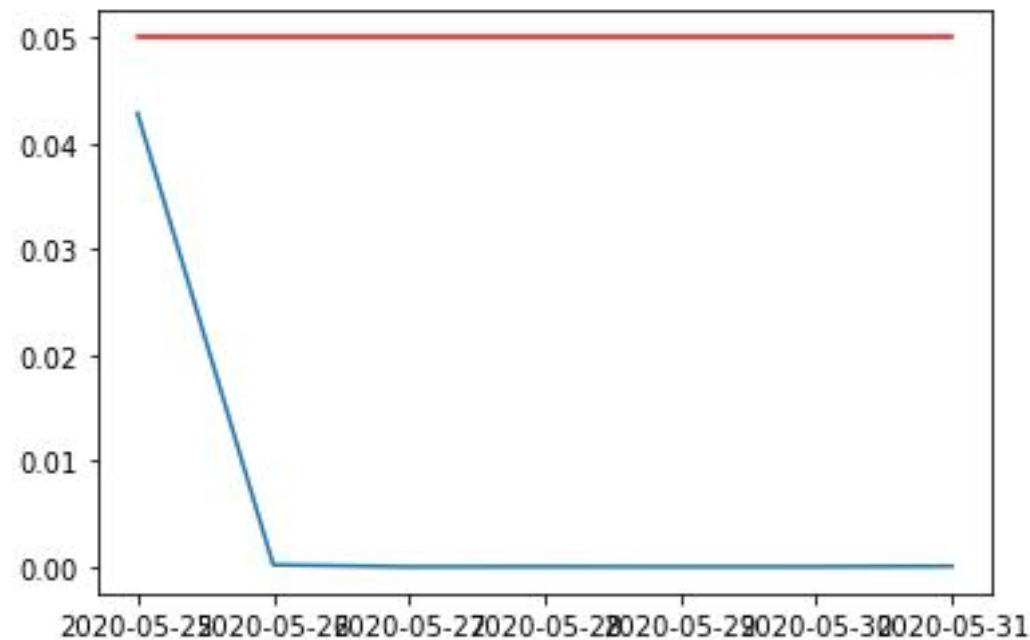
Ожидаем: ниже отмены на Android устройствах

- Отфильтровали датасет по своей гипотезе, используя метрику, которая описана в блоке “Ожидаем”
- Построили периодический p-value по выбранной метрике.

```
In [20]: alpha=0.05

fig, ax = plt.subplots()
ax.plot(sorted(android['date'].unique()), Pvalue)
ax.plot(sorted(android['date'].unique()), [alpha]*7, color='red')
```

```
Out[20]: [<matplotlib.lines.Line2D at 0x1f950ce28e0>]
```



P-value ниже alpha => **предложение промокода юзерам Android при нажатии кнопки отмены заказа приводит к сокращению отмен. Фича работает.**



# Инсайты курса

*Раньше я думал, что A/B-тестирование это очень сложно,  
а теперь разобрался в теме.*

