



CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Course introduction and overview

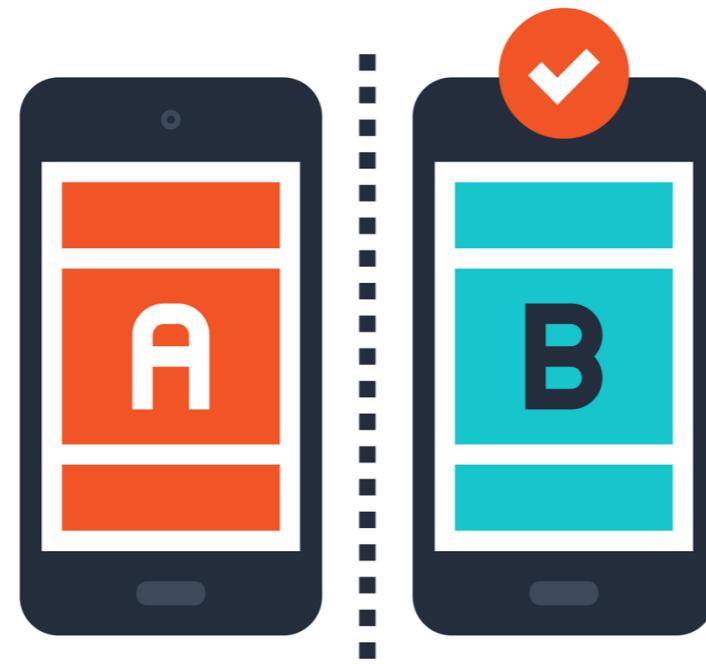
Ryan Grossman
Data Scientist, EDO

What Is A/B Testing?



- Test two or more different ideas against each other
- See which one empirically performs better

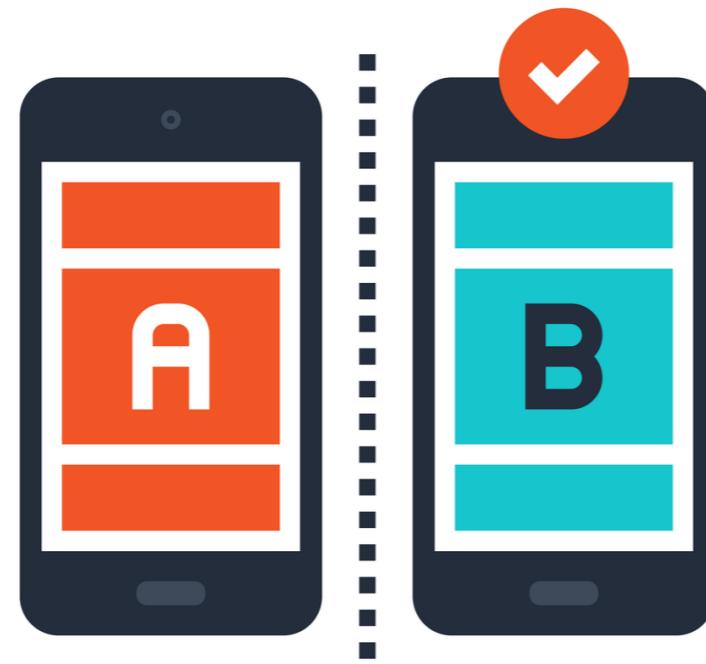
Why is A/B Testing Important?



A/B TESTING

- No guessing

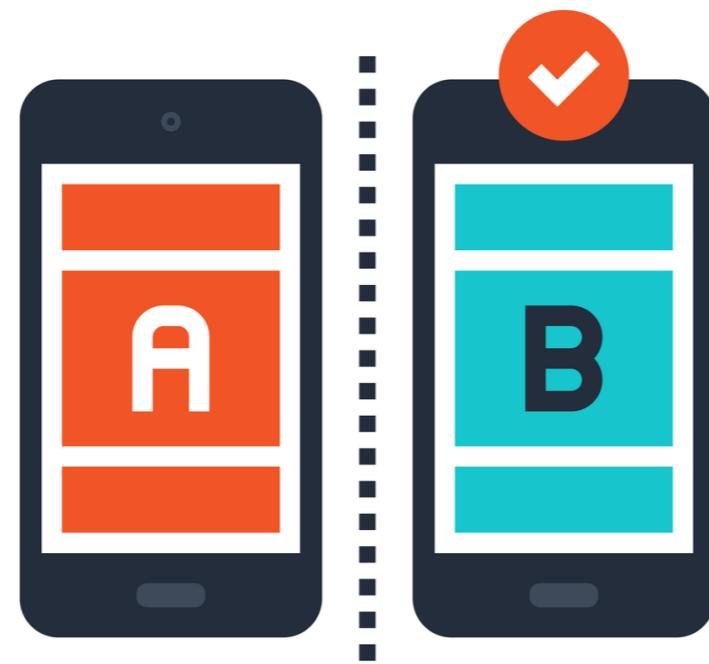
Why is A/B Testing Important?



A/B TESTING

- Accurate answers

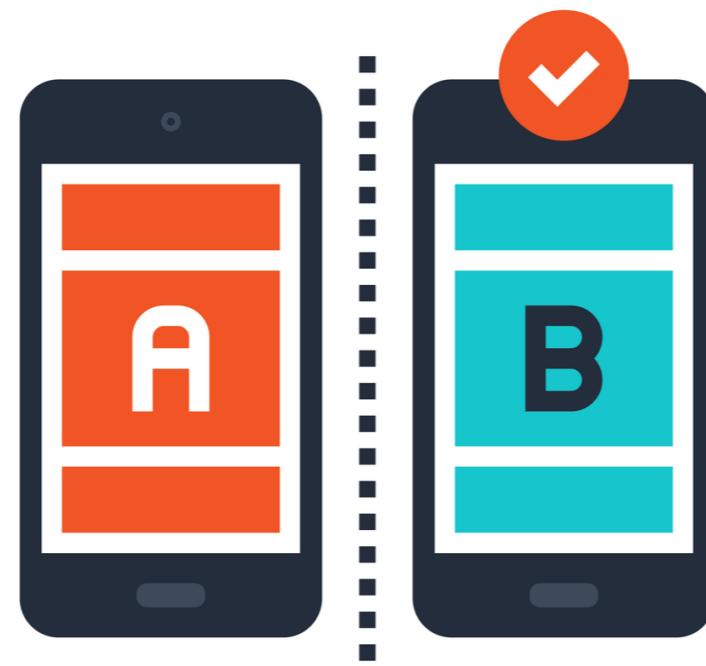
Why is A/B Testing Important?



A/B TESTING

- Rapidly iterate on ideas

Why is A/B Testing Important?



A/B TESTING

- Establish causal relationships

How Does A/B Testing Work?



Where Can A/B Testing be Used?



Course Progression

1. Understanding Users - *Key Performance Indicators*
2. Identifying Trends - *Exploratory Data Analysis*
3. Optimizing Performance - *Design of A/B Tests*
4. Data Driven Decisions - *Analyzing A/B Test Results*

Key Performance Indicators



Uncovering KPIs



Experience

Uncovering KPIs



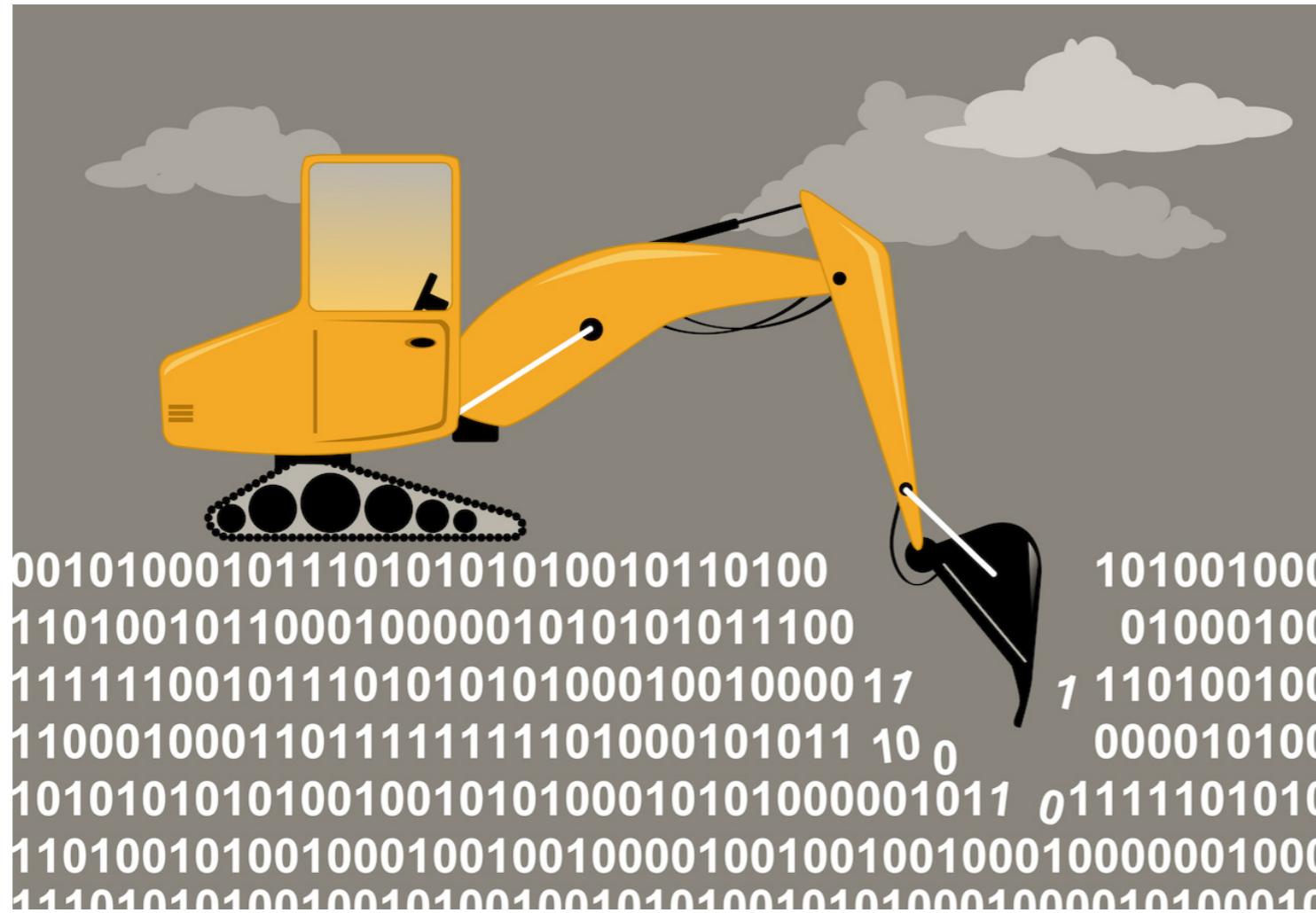
Domain Knowledge

Uncovering KPIs



Exploratory Data Analysis

Exploring Data



Customer dataset



CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!



CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Identifying and understanding KPIs

Ryan Grossman
Data Scientist, EDO

Mobile app that offers meditation services



- Paid subscription
- One-off in-app purchases

Generalizability of A/B Testing



- The same techniques can be generalized

Dataset 1: User Demographics

```
In [1]: import pandas as pd  
  
In [2]: customer_demographics = pd.read_csv('customer_demographics.csv')  
  
In [3]: customer_demographics.head()  
  
Out[3]:  
uid      reg_date    device  gender  country  age  
54030035  2017-06-29    and      M     USA     19  
72574201  2018-03-05    iOS      F     TUR     22  
64187558  2016-02-07    iOS      M     USA     16  
92513925  2017-05-25    and      M     BRA     41  
99231338  2017-03-26    iOS      M     FRA     59
```

Dataset 2: User Actions

```
In [4]: customer_subscriptions = pd.read_csv('customer_subscriptions.csv')
```

```
In [5]: customer_subscriptions.head()
```

```
Out[5]:
```

uid	lapse_date	subscription_date	price
59435065	2017-07-06	2017-07-08	499
26485969	2018-03-12	None	0
64187658	2016-02-14	2016-02-14	499
99231339	2017-04-02	None	0
64229717	2017-05-24	2017-05-25	499

KPI: Conversion Rate



Defining Our KPI



- Stable, generalizable KPIs are better than custom KPIs

Merging the datasets



- `pd.merge(df1, df2)`
- `df1.merge(df2)`

Merging Mechanics

Merge Components

```
In [6]: sub_data_demo = customer_demographics.merge(...)  
In [6]: sub_data_demo = customer_demographics.merge(  
                  customer_subscriptions, ...)  
In [6]: sub_data_demo = customer_demographics.merge(  
                  customer_subscriptions,  
                  how='inner', ....)  
In [6]: sub_data_demo = customer_demographics.merge(  
                  customer_subscriptions,  
                  how='inner', on=['uid'])
```

```
In [7]: sub_data_demo.head()  
Out[7]:  
uid          reg_date      device  ... price  
54030729    2017-06-29    and     ... 499
```

Next Steps



- Aggregate combined dataset
- Calculate the potential KPIs



CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!



CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Tools for the exploratory analysis of KPIs

Ryan Grossman
Data Scientist, EDO

Reminder: Conversion rate is just one KPI

- Most companies will have many KPIs
- Each serving a different purpose
- We are working through one of these cases: Conversion rate

Methods for Calculating KPIs

pandas.DataFrame.groupby

```
DataFrame.groupby(by=None, axis=0, level=None,  
                 as_index=True, sort=True,  
                 group_keys=True, squeeze=False, **kwargs)
```

pandas.DataFrame.agg

```
DataFrame.agg(func, axis=0, *args, **kwargs)
```

pandas .groupby()

```
In [1] sub_data_grp = sub_data_demo.groupby(...)
```

by Argument

```
In [1] sub_data_grp = sub_data_demo.groupby(by=['country', 'device'], ...)
```

axis Argument

```
In [1] sub_data_grp = sub_data_demo.groupby(by=['country', 'device'],  
                                         axis=0, ...)
```

as index Argument

Aggregating Our Data

DataFrameGroupBy Object

```
In [2]: sub_data_grp  
Out[2]: <pandas.core.groupby.DataFrameGroupBy object at 0x10ec29080>
```

Mean Price Paid

```
In [3]: sub_data_grp.price.mean()  
Out[3]:  
    country   device   price  
0      BRA      and  312.163551  
1      BRA      iOS  247.884615  
2      CAN      and  431.448718  
3      CAN      iOS  505.659574  
4      DEU      and  398.848837  
5      DEU      iOS  313.128000  
6      FRA      and  320.391304  
7      FRA      iOS  324.786408
```

The "Agg" Method

Simple Aggregation

```
In [4]: sub_data_grp.price.agg('mean')
```

```
Out[4]:
```

	country	device	price
0	BRA	and	312.163551
1	BRA	iOS	247.884615
2	CAN	and	431.448718
3	CAN	iOS	505.659574
4	DEU	and	398.848837
5	DEU	iOS	313.128000
6	FRA	and	320.391304
7	FRA	iOS	324.786408

The "Agg" Method

Expanded Aggregation

```
In [5]: sub_data_grp.price.agg(['mean', 'median'])
```

```
Out[5]:
```

		mean	median
country	device		
BRA	and	312.163551	0
	iOS	247.884615	0
CAN	and	431.448718	699
	iOS	505.659574	699
DEU	and	398.848837	499
	iOS	313.128000	0
FRA	and	320.391304	0
	iOS	324.786408	0
TUR	and	216.622951	0
	iOS	249.638462	0
USA	and	420.124650	499
	iOS	380.463265	499

The "Agg" Method

Full Aggregation

Out [6]

	country	device	price			age		
			mean	min	max	mean	min	max
0	BRA	and	312.163551	0	999	24.303738	15	67
1	BRA	iOS	247.884615	0	999	24.024476	15	79
2	CAN	and	431.448718	0	999	23.269231	15	58
3	CAN	iOS	505.659574	0	999	22.234043	15	38
4	DEU	and	398.848837	0	999	23.848837	15	67
5	DEU	iOS	313.128000	0	999	24.208000	15	54
6	FRA	and	320.391304	0	999	24.808696	15	55
7	FRA	iOS	324.786408	0	999	25.475728	15	62
8	TUR	and	216.622951	0	999	24.704918	15	59
9	TUR	iOS	249.638462	0	999	23.623077	15	65
10	USA	and	420.124650	0	999	25.000000	15	81
11	USA	iOS	380.463265	0	999	25.146939	15	84

Custom Aggregation

Truncating Function

```
In [7]: def truncated_mean(data):
    top_val = data.quantile(.9)
    bot_val = data.quantile(.1)

    trunc_data = data[(data <= top_val) & (data >= bot_val)]
    mean = trunc_data.mean()

    return mean
```

Custom Function Aggregation

```
In [8]: sub_data_grp.agg({'age': [truncated_mean] })
```

```
Out[8]:
      country     device     age
                           truncated_mean
0        BRA        and          22.636364
...
...
```



CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!

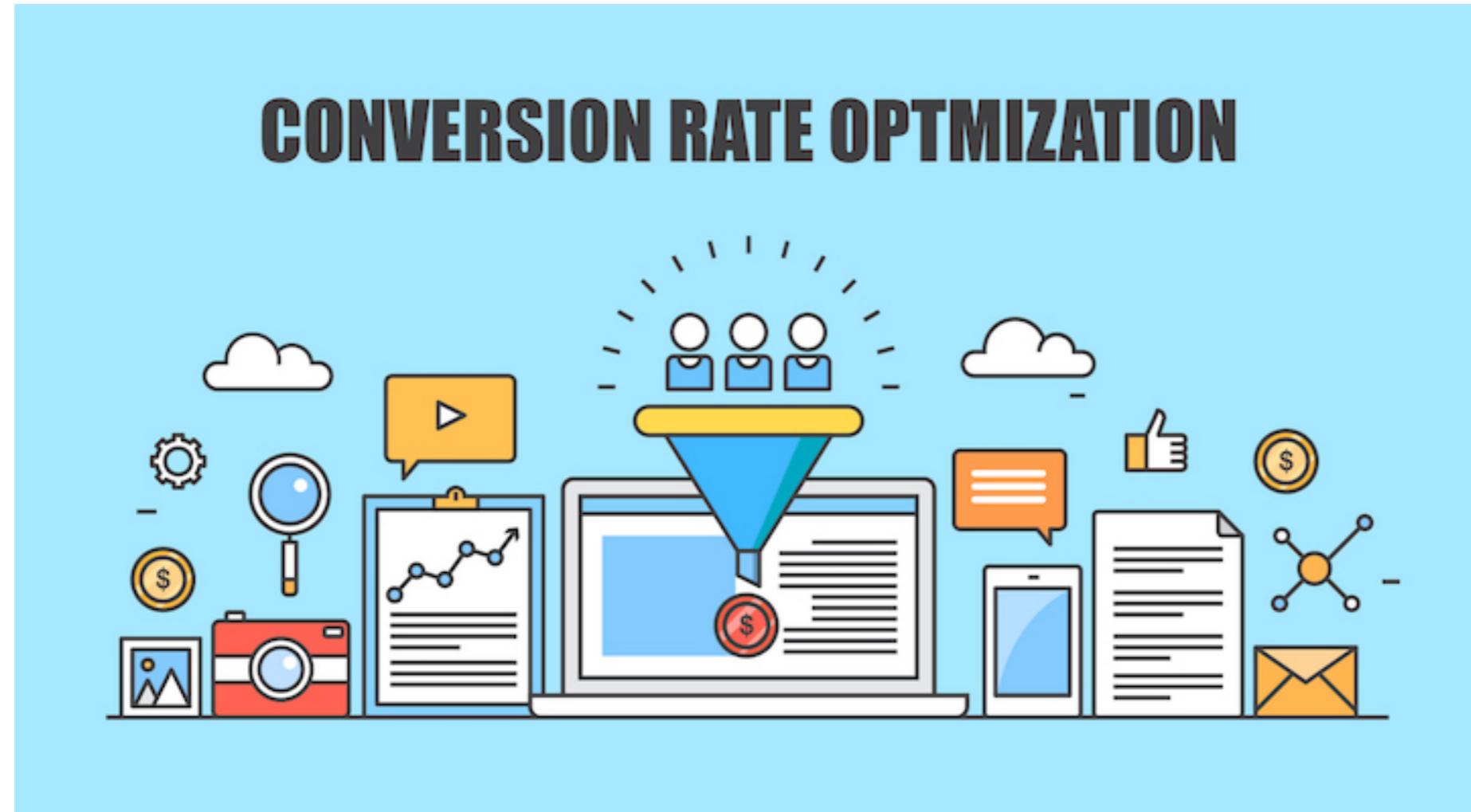


CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Calculating KPIs - A practical example

Ryan Grossman
Data Scientist, EDO

Goal - Comparing Our KPIs



- Goal: Examine KPI of user conversion rate after free trial ends
- In this case: Look at first week after the trial ends

KPI Calculation

```
In [1]: import numpy as np  
In [2]: from datetime import datetime, timedelta  
In [3]: sub_data_demo.lapse_date.max()  
Out[3]: '2018-03-17'
```

- "Lapse date" = Date that the trial ended

KPI Calculation

- Remove users who lapsed today or any of the prior 7 days
 - Ensures everyone has a full 7 days to subscribe

KPI Calculation

```
In [7]: total_users = conv_sub_data.price.count()
```

```
Out[7]: 2787
```

```
In [8]: sub_days = conv_sub_data.lapse_date + timedelta(days=7)
```

```
In [9]: total_subs = conv_sub_data[  
        (conv_sub_data.price > 0) &  
        (conv_sub_data.subscription_date <= sub_days)]
```

```
In [10]: total_subs = total_subs.price.count()
```

```
In [11]: total_subs
```

```
Out[11]: 648
```

KPI Calculation

```
In [12]: conversion_rate = total_subs / total_users
```

```
In [13]: conversion_rate
```

```
Out[13]: 0.23250807319698599
```

Cohort Conversion Rate

```
In [14]: max_lapse_date = current_date - timedelta(days=14)  
In [15]: conv_sub_data = conv_sub_data.copy()  
In [16]: conv_sub_data = sub_data_demo[ (sub_data_demo.lapse_date  
          <= max_lapse_date) ]
```

Cohort Conversion Rate

```
In [17]: sub_time = np.where(  
    conv_sub_data.subscription_date.notnull(),  
    (conv_sub_data.subscription_date -  
     conv_sub_data.lapse_date).dt.days,  
    pd.NaT)
```

```
In [18]: conv_sub_data['sub_time'] = sub_time
```

Cohort Conversion Rate

```
In [20]: purchase_cohorts = conv_sub_data.groupby(  
        by=['gender', 'device'], as_index=False)  
  
In [21]: purchase_cohorts.agg({sub_time: [gcr7,gcr14] })  
  
Out[21]:  
      gender    device    sub_time  
              gcr7      gcr14  
0       F      and  0.221963  0.230140  
1       F      iOS   0.229310  0.237931  
2       M      and  0.252349  0.257718  
3       M      iOS   0.218045  0.225564
```



How long does it take to gain insight into a metric?

- Monthly conversion rate
 - Would need to wait a month from the lapse date
 - Impractical

Exploratory data analysis

- Can reveal relationships between metrics and key results
- Can be tied to business metrics in important ways

Why is conversion rate important?

- Could potentially serve as a warning of potential problems later on

Next chapter: Continue exploring conversion rates

- How does it evolve over time?

Measuring KPIs across groups is crucial

- Changes can impact groups in drastically different ways



CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!