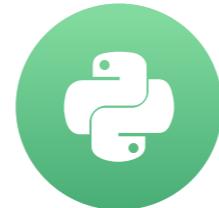


Course introduction and overview

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON



Ryan Grossman
Data Scientist, EDO

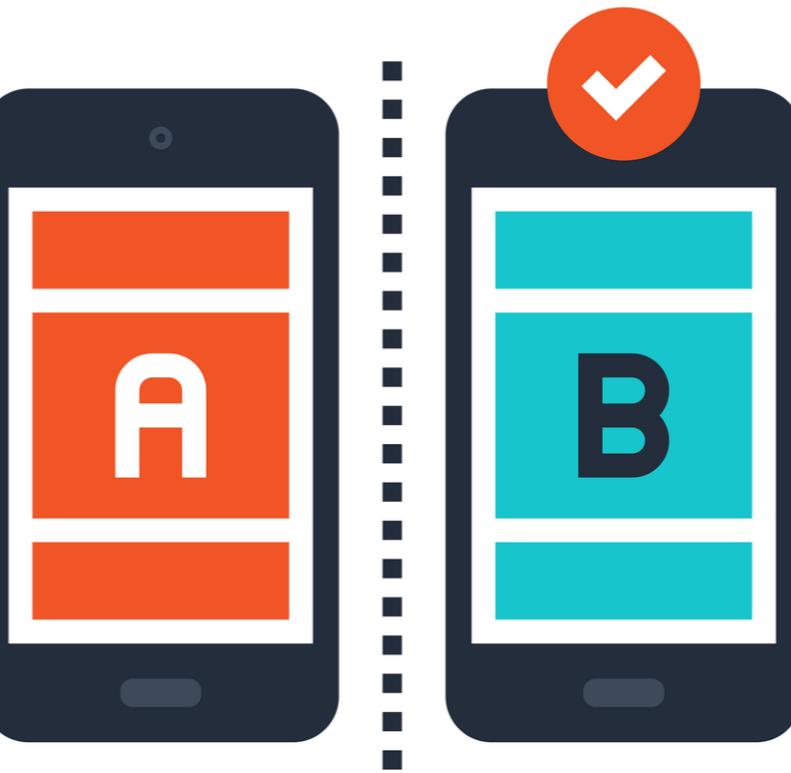
What is A/B testing?



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

Why is A/B testing important?

- No guessing
- Provides accurate answers
- Allows to rapidly iterate on ideas
- ...and establish causal relationships



A/B TESTING

How does A/B testing work?



Where can A/B testing be used?

Users + ideas → A/B test

- testing impact of drugs
- incentivizing spending
- driving user growth
- ...and many more!



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 (KPIs)

- A/B tests are run to improve KPIs
- KPIs – metrics important to the organization
 - likelihood of a side-effect
 - revenue
 - conversion rate
 - ...

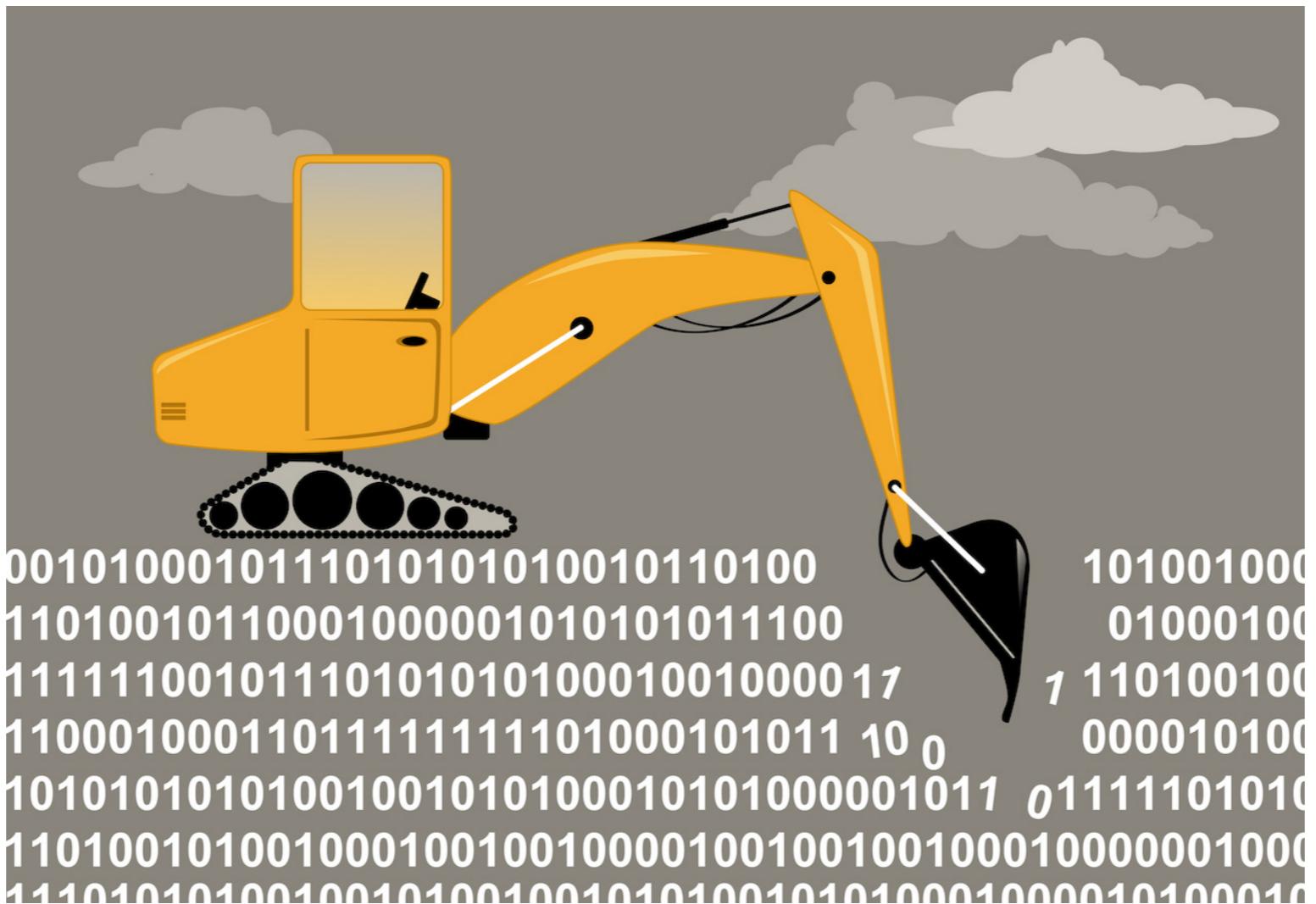


Identifying KPIs

Experience + Domain knowledge + Exploratory data analysis



Next up: exploring data & KPIs

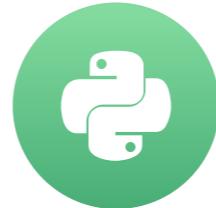


Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Identifying and understanding KPIs

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON



Ryan Grossman
Data Scientist, EDO

Example: meditation app

Services

- Paid subscription
- One-off in-app purchases



Goals/KPIs

- Maintain high conversion rate

Dataset 1: User demographics

```
import pandas as pd  
  
# load customer_demographics  
customer_demographics = pd.read_csv('customer_demographics.csv')  
  
# print the head of customer_demographics  
print(customer_demographics.head())
```

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

Dataset 2: User actions

```
# load customer_subscriptions  
customer_subscriptions = pd.read_csv('customer_subscriptions.csv')  
  
# print the head of customer_subscriptions  
print(customer_subscriptions.head())
```

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



Choosing a KPI

- Stable, generalizable KPIs are better than custom KPIs
- Correlation with business factors

Match demographic to subscription data

- Merging -- equivalent of SQL `JOIN`
- In `pandas` :
 - `pd.merge(df1, df2)`
 - `df1.merge(df2)`



Merging mechanics

```
# merge customer_demographics (left) and customer_subscriptions (right)
sub_data_demo = customer_demographics.merge(
    # right dataframe
    customer_subscriptions,
    # join type
    how='inner',
    # columns to match
    on=[ 'uid' ])
sub_data_demo.head()
```

```
uid          reg_date        device  ... price
54030729  2017-06-29       and      ... 499
```

Next steps

- Aggregate combined dataset
- Calculate the potential KPIs

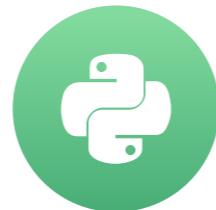


Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Exploratory analysis of KPIs

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON



Ryan Grossman
Data Scientist, EDO

Reminder: conversion rate is just one KPI

- Most companies will have many KPIs
- Each serving a different purpose
- Our case: conversion rate

Methods for calculating KPIs

Group: `pandas.DataFrame.groupby()`

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

Aggregate: `pandas.DataFrame.agg()`

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

Group data: .groupby()

- **by** : fields to group by
- **axis** : `axis=0` will group by columns
- **as_index** : `as_index=True` will use group labels as index

```
# sub_data_demo - combined demographics and purchase data
sub_data_grp = sub_data_demo.groupby(by=['country', 'device'],
                                      axis=0,
                                      as_index=False)

sub_data_grp
```

```
<pandas.core.groupby.DataFrameGroupBy object at 0x10ec29080>
```

Aggregate data

Mean price paid per group

```
sub_data_grp.price.mean()
```

	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

Aggregate data: .agg()

Pass the name of an aggregation function to `agg()` :

```
sub_data_grp.price.agg('mean')
```

	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

.agg(): multiple functions

Pass a list of names of aggregation functions:

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

		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

.agg(): multiple functions, multiple columns

Pass a dictionary of column names and aggregation functions

```
sub_data_grp.agg({'price': ['mean', 'min', 'max'],
                  'age': ['mean', 'min', 'max']})
```

	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

.agg(): custom functions

```
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)
```

Pass a custom function:

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

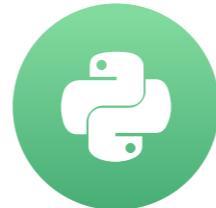
```
country      device      age
                           truncated_mean
0      BRA        and      22.636364
...
...
```

Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Calculating KPIs - a practical example

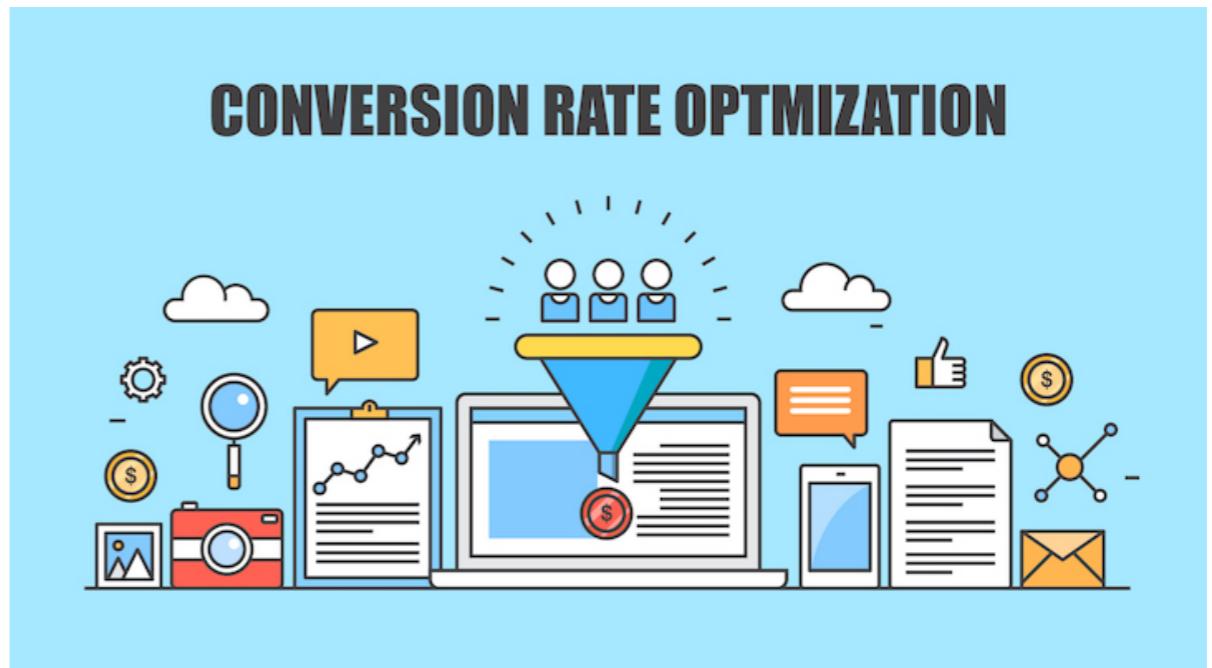
CUSTOMER ANALYTICS & A/B TESTING IN PYTHON



Ryan Grossman
Data Scientist, EDO

Goal - comparing our KPIs

- Goal: examine KPI "user conversion rate" after free trial ends
- Look at **first week after the trial ends**



Conversion rate : check max lapse date

```
import numpy as np  
from datetime import datetime, timedelta  
current_date = pd.to_datetime('2018-03-17')
```

- "Lapse date" = Date that the trial ended

```
print(sub_data_demo.lapse_date.max())
```

```
'2018-03-17'
```

KPI calculation : restrict users by lapse date

```
# latest lapse date: a week before today  
max_lapse_date = current_date - timedelta(days=7)  
  
# restrict to users lapsed before max_lapse_date  
conv_sub_data = sub_data_demo[(sub_data_demo.lapse_date < max_lapse_date)]  
  
# count the users  
total_users_count = conv_sub_data.price.count()  
print(total_users_count)
```

2787

KPI calculation: restrict subscription date

```
# latest subscription date: within 7 days of lapsing  
max_sub_date = conv_sub_data.lapse_date + timedelta(days=7)  
  
# filter the users with non-zero subscription price  
# who subscribed before max_sub_date  
total_subs = conv_sub_data[  
    (conv_sub_data.price > 0) &  
    (conv_sub_data.subscription_date <= max_sub_date)]  
  
# count the users  
total_subs_count = total_subs.price.count()  
print(total_subs_count)
```

648

KPI calculation: find the conversion rate

```
conversion_rate = total_subs_count / total_users_count  
print(conversion_rate)
```

```
0.23250807319698599
```

Cohort conversion rate

```
# keep users lapsed in the last 14 days  
max_lapse_date = current_date - timedelta(days=14)  
conv_sub_data = conv_sub_data.copy()  
conv_sub_data = sub_data_demo[ (sub_data_demo.lapse_date <= max_lapse_date) ]
```

Cohort conversion rate

```
# find number of days between subscription and lapse date if subscribed  
# ... and pd.NaT otherwise  
sub_time = np.where(  
    # if  
        conv_sub_data.subscription_date.notnull(),  
    # then  
        (conv_sub_data.subscription_date - conv_sub_data.lapse_date).dt.days,  
    # else  
        pd.NaT)  
  
# create a new column 'sub_time'  
conv_sub_data['sub_time'] = sub_time
```

Cohort conversion rate

- `gcr7()` , `gcr14()` : conversion rates over 7 and 14 days

```
# form cohorts
purchase_cohorts = conv_sub_data.groupby(by=[ 'gender', 'device'], as_index=False)
# find group conversion rated for each cohort using gcr7,gcr14
purchase_cohorts.agg({sub_time: [gcr7,gcr14]})
```

	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 to choose KPI metrics?

- Consider how long it takes

Monthly conversion rate: impractical to wait for a month

- Use exploratory data analysis

Reveal relationships between metrics and key results

- Explore ties to business metrics

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?
- Changes can impact groups in drastically different ways

Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON