

# Course introduction and overview

CUSTOMER ANALYTICS AND A/B TESTING IN PYTHON



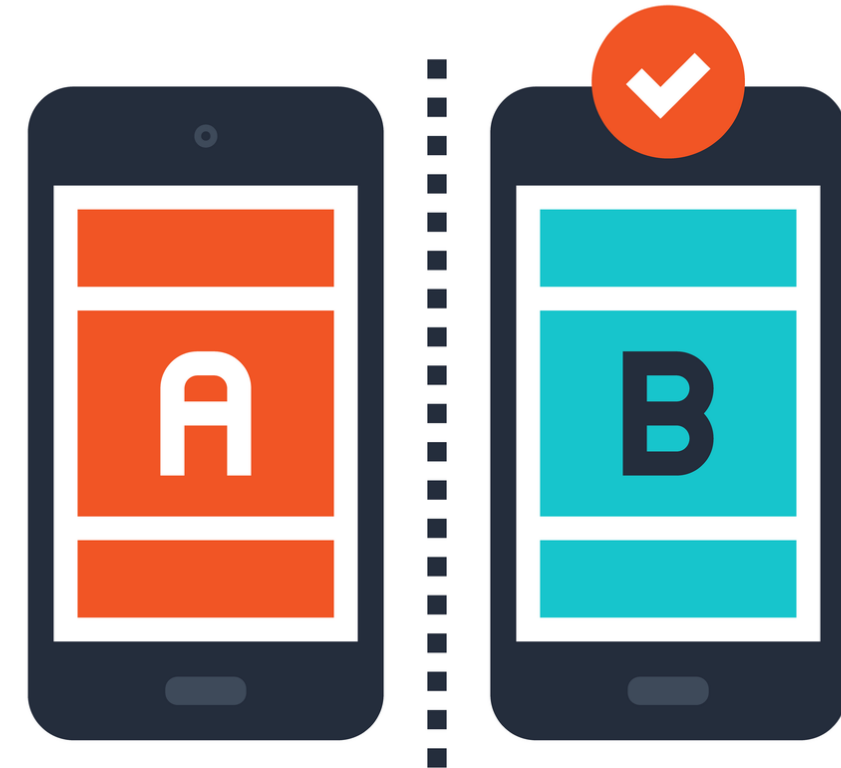
**Ryan Grossman**  
Data Scientist, EDO

# What is A/B testing?

- **A/B Testing:** Test different ideas against each other in the real world
- Choose the one that statistically performs better

# Why is A/B testing important?

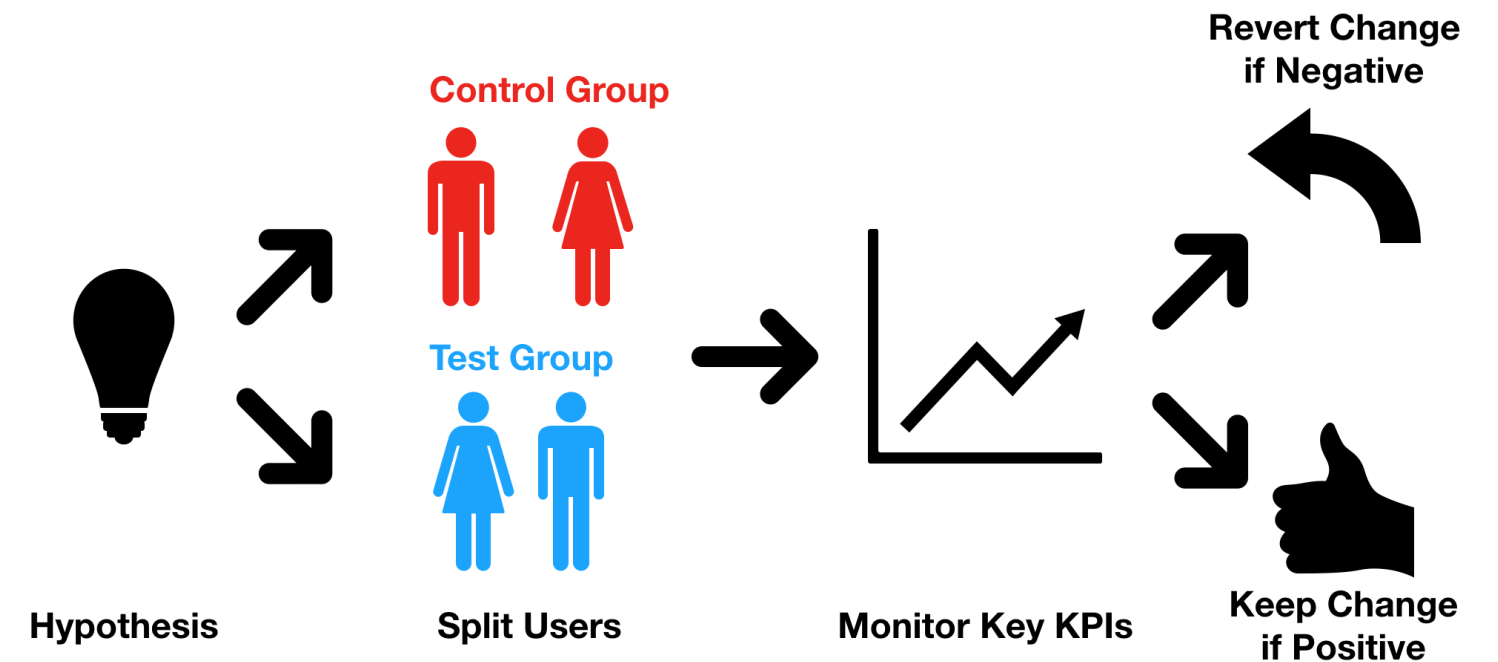
- No guessing
- Provides accurate answers - quickly
- Allows to rapidly iterate on ideas
- ...and establish causal relationships



**A/B TESTING**

# A/B test process

1. Develop a hypothesis about your product or business
2. **Randomly** assign users to two different groups
3. Expose:
  - Group 1 to the the current product rules
  - Group 2 to a product that tests the hypothesis
4. Pick whichever performs better according to a set of KPIs



# 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:** Measure impact of changes on KPIs
- **KPIs** — metrics important to an organization
  - likelihood of a side-effect
  - revenue
  - conversion rate
  - ...



# How to identify KPIs

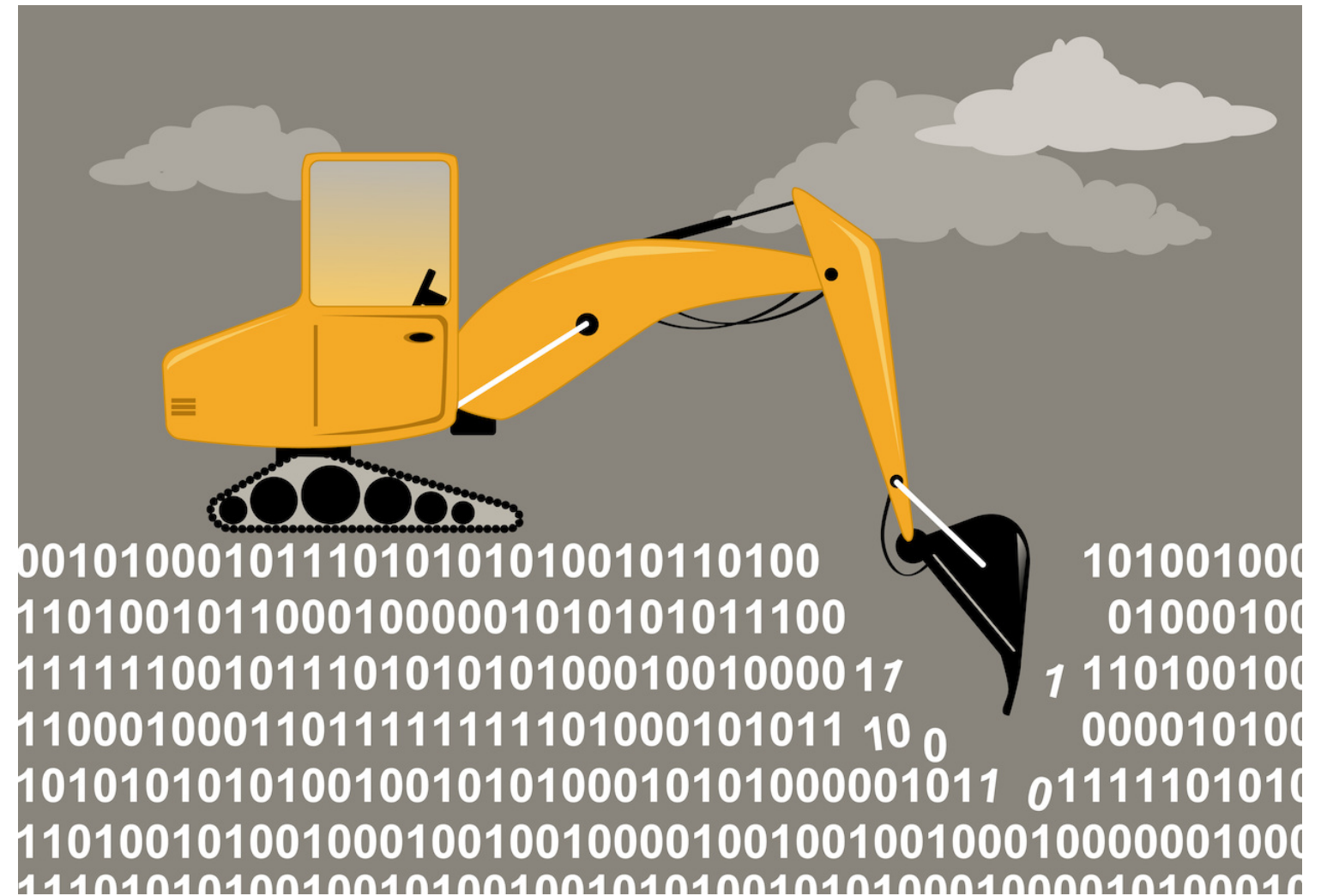
Experience + Domain knowledge + Exploratory data analysis

- **Experience & Knowledge** - *What is important to a business*
- **Exploratory Analysis** - *What metrics and relationships impact these KPIs*



# Next Up...

- Exploratory Data Analysis (EDA)
- Identify KPIs and areas for further analysis

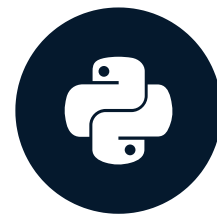


# Let's practice!

CUSTOMER ANALYTICS AND A/B TESTING IN PYTHON

# Identifying and understanding KPIs

CUSTOMER ANALYTICS AND A/B TESTING IN PYTHON



**Ryan Grossman**  
Data Scientist, EDO

# Example: meditation app

## Services

- Paid subscription
- In-app purchases

## Goals/KPIs

- Maintain high free → paid 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
99231338	2017-03-26	iOS	M	FRA	59

# 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

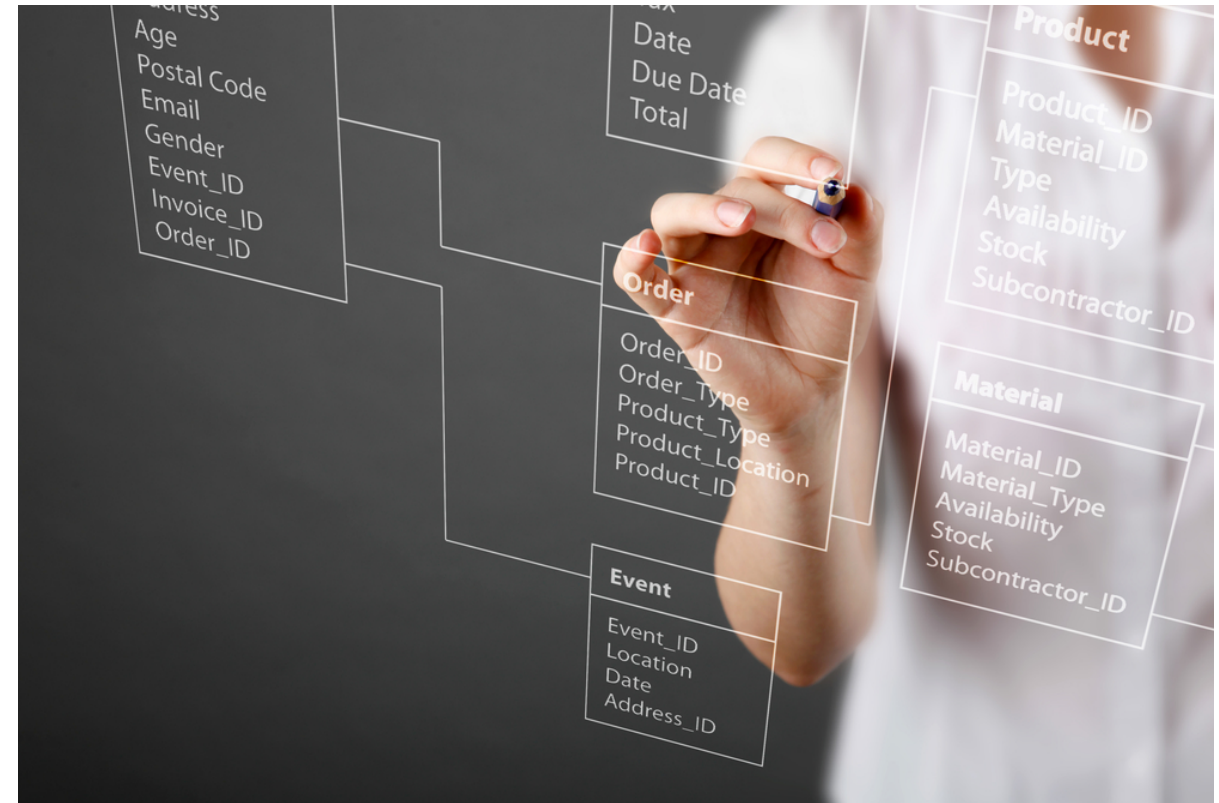
- **Conversion Rate:** Percentage of users who subscribe after the free trial
  - Of users who convert within one week?  
One month?...
  - Across all users or just a subset?
  - ...

## Choosing a KPI

- Stability over time
- Importance across different user groups
- Correlation with other business factors

# Joining the demographic and subscription data

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





# Merging mechanics

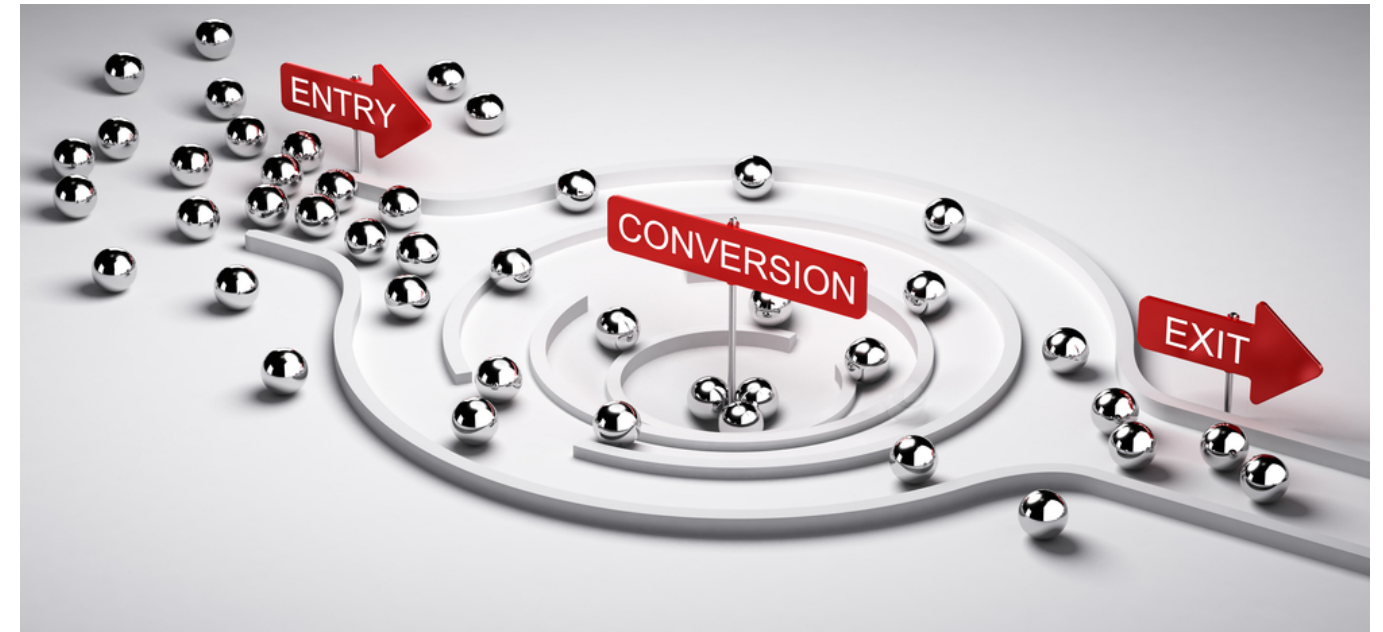
```
# merge customer_demographics (left) and customer_subscriptions (right)
sub_data_demo = customer_demographics.merge(
    # right dataframe
    customer_subscriptions,
    # join type
    how='inner',
    # columns to match
    on=['uid'])

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 AND A/B TESTING IN PYTHON

# Exploratory analysis of KPIs

CUSTOMER ANALYTICS AND A/B TESTING IN PYTHON



**Ryan Grossman**  
Data Scientist, EDO

# KPIs

- **Reminder:** conversion rate is just one KPI
- Most companies will have many KPIs
- Each serves a different purpose

# 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)
```

# Grouping Data: .groupby()

- **by** : fields to group by
- **axis** : **axis=0** will group by columns, **axis=1** will group by rows
- **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>
```

# Aggregating data - mean price paid per group

```
# Mean price paid for each country/device  
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()` :

```
# Find the mean price paid with 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:

```
# Mean and median price paid for each country/device  
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

# .agg(): multiple functions, multiple columns

Pass a dictionary of column names and aggregation functions

```
# Calculate multiple metrics across different groups
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

# .agg(): custom functions

```
def truncated_mean(data):  
    """Compute the mean excluding outliers"""  
    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)  
  
# Find the truncated mean age by group  
sub_data_grp.agg({'age': [truncated_mean]})
```

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

# Let's practice!

CUSTOMER ANALYTICS AND A/B TESTING IN PYTHON

# Calculating KPIs - a practical example

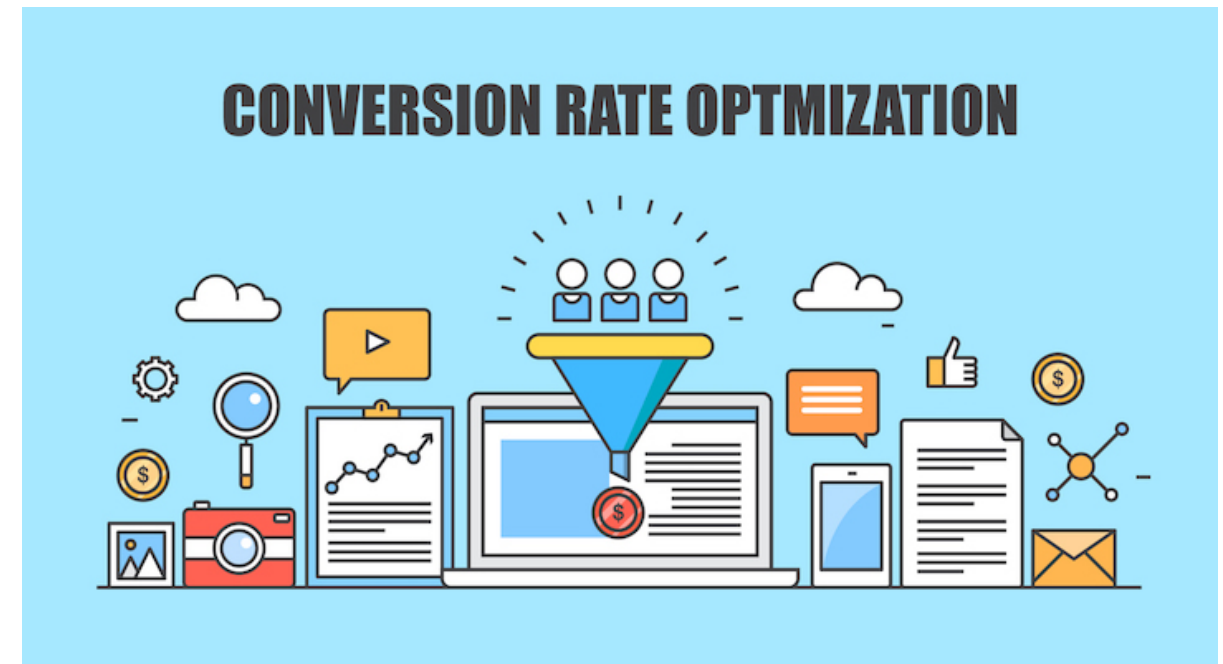
CUSTOMER ANALYTICS AND A/B TESTING IN PYTHON



**Ryan Grossman**  
Data Scientist, EDO

# Goal - comparing our KPIs

- **Goal:** Examine the KPI "user conversion rate" after the free trial
- **Week One Conversion Rate:** Limit to users who convert in their first week after the trial ends



# Conversion rate : maximum lapse date

```
import pandas as pd
from datetime import datetime, timedelta

current_date = pd.to_datetime('2018-03-17')
```

- **Lapse Date:** Date the trial ends for a given user

```
# What is the maximum lapse date in our data
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 remaining in our data
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 remaining in our data
total_subs_count = total_subs.price.count()
print(total_subs_count)
```

648

# KPI calculation: find the conversion rate

- **Conversion Rate:** *Total Subscribers / Potential Subscribers*

```
# calculate the conversion rate with our previous values
conversion_rate = total_subs_count / total_users_count
print(conversion_rate)
```

```
0.23250807319698599
```

# Cohort conversion rate

```
# Create a copy of our dataframe
conv_sub_data = conv_sub_data.copy()

# keep users who lapsed prior to the last 14 days (2 weeks)
max_lapse_date = current_date - timedelta(days=14)
conv_sub_data = sub_data_demo[
    (sub_data_demo.lapse_date <= max_lapse_date)
]
```

# Cohort conversion rate

- **Sub Time:** How long it took a user to subscribe

```
# Find the days between lapse and subscription if they
# subscribed ... and pd.NaT otherwise
sub_time = np.where(
    # if: a subscription date exists
    conv_sub_data.subscription_date.notnull(),
    # then: find how many days since their lapse
    (conv_sub_data.subscription_date - conv_sub_data.lapse_date).dt.days,
    # else: set the value to pd.NaT
    pd.NaT)
# create a new column 'sub_time'
conv_sub_data['sub_time'] = sub_time
```

# Cohort conversion rate

- `gcr7()` , `gcr14()` : calculate the 7 and 14 day conversion rates

```
# group by the relevant cohorts
```

```
purchase_cohorts = conv_sub_data.groupby(by=['gender', 'device'], as_index=False)
```

```
# find the conversion rate 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?

- Infinitely many potential KPIs
- How long does it take to determine
  - Monthly Conversion Rate = *1 Month Wait time*
- Leverage Exploratory Data Analysis
  - *Reveals relationships between metrics and key results*
- **Keep In Mind** How do these KPIs and my Business goals relate

# Why is conversion rate important?

- Strong measure of growth
- Potential early warning sign of problems
  - Sensitive to changes in the overall ecosystem



# Next chapter: continue exploring conversion rates

- How does this KPI evolve over time?
- See how changes can impact different groups differently

# Let's practice!

CUSTOMER ANALYTICS AND A/B TESTING IN PYTHON