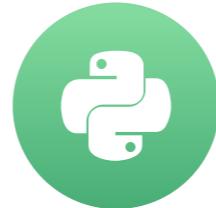


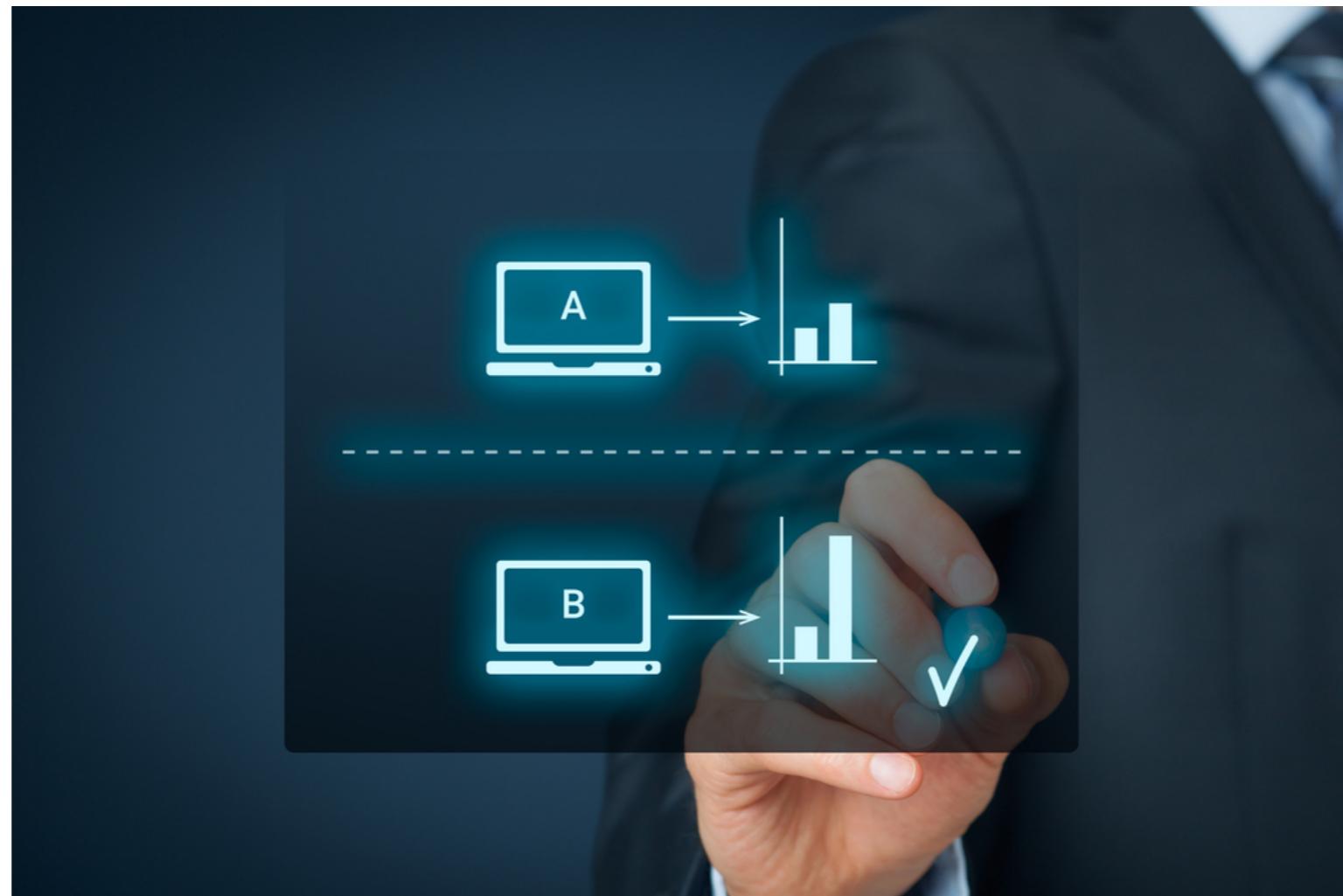
Introduction to A/B testing

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON



Ryan Grossman
Data Scientist, EDO

Overview

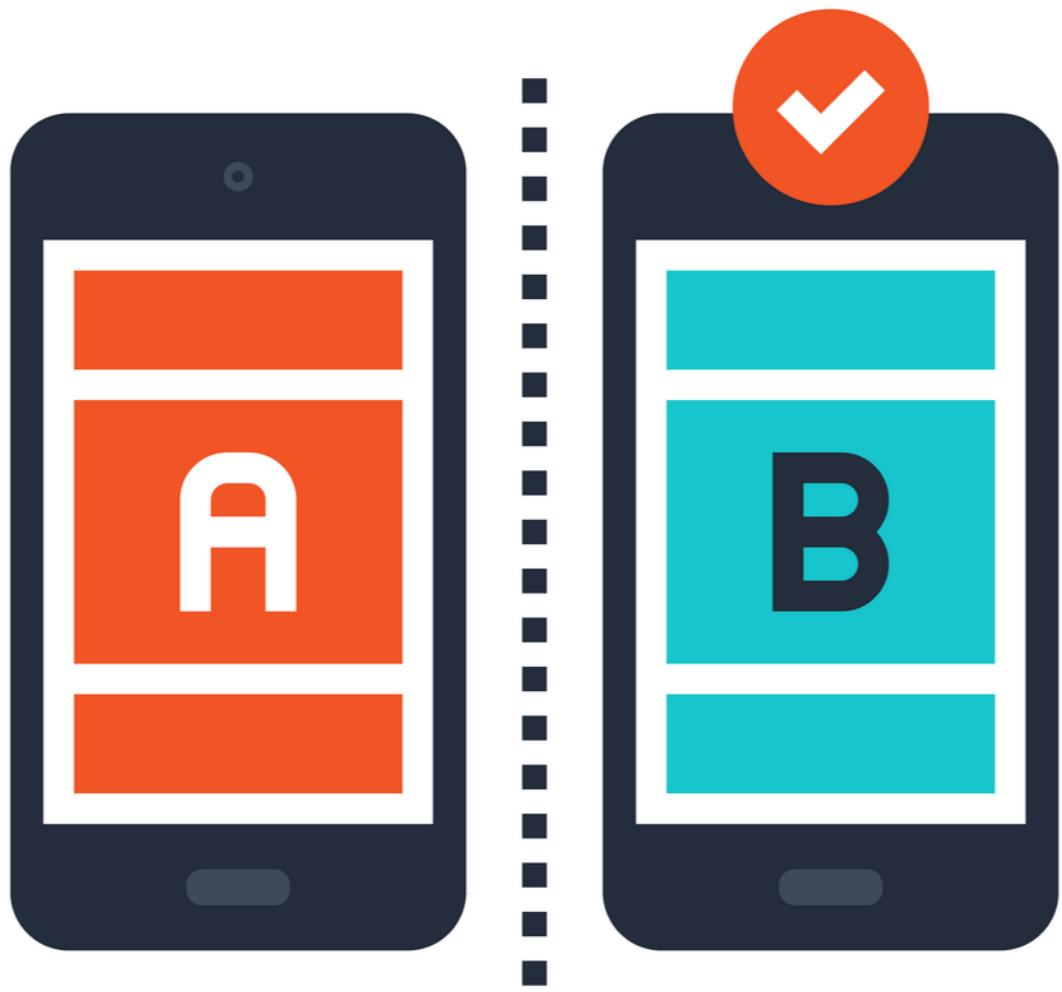




MANAGEMENT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.





A/B TESTING

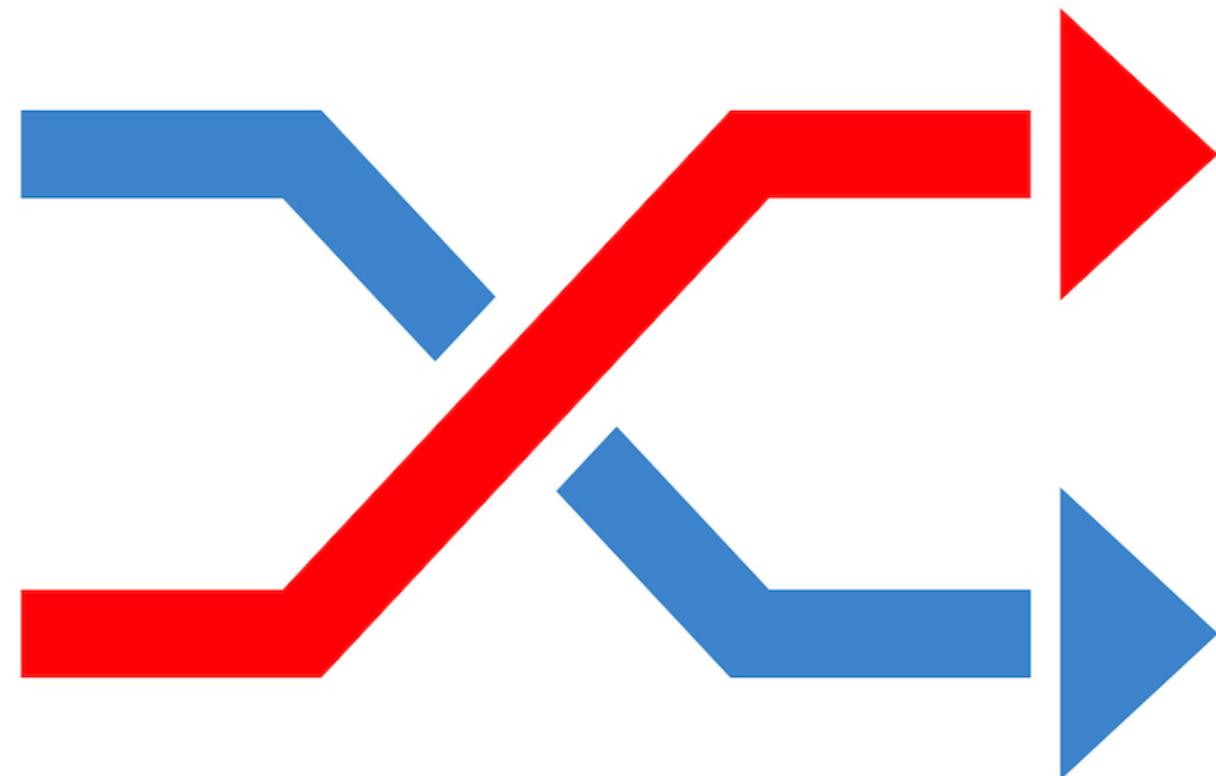
A/B Testing Example

MANAGEMENT

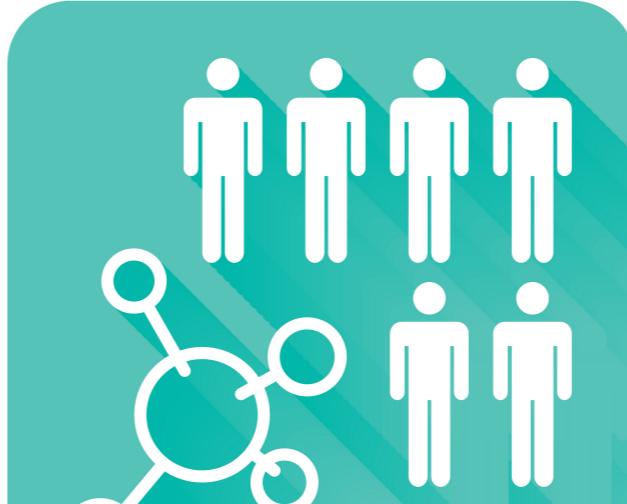
Lore ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit.



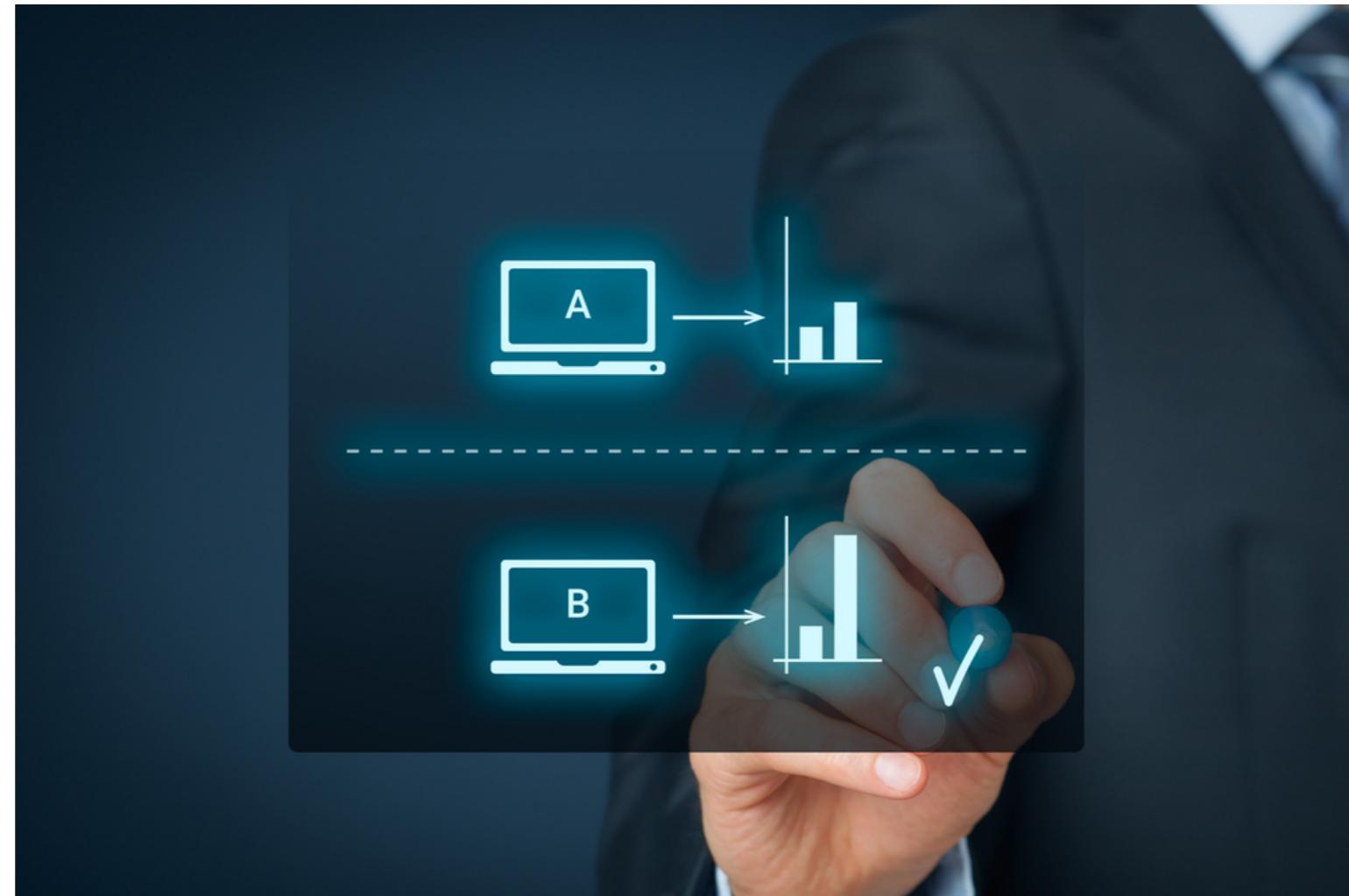
The Importance of Randomness



A/B Testing Flexibility



Good and Bad Cases for A/B Testing



Good and Bad Cases for A/B Testing



Good and Bad Cases for A/B Testing

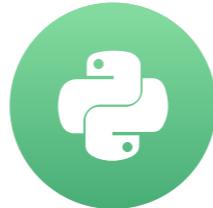


Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Initial A/B test design

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON



Ryan Grossman
Data Scientist, EDO

Increasing Revenue through A/B Testing



Generalizability of A/B Testing



Paywall Views & Demographics Data

```
demographics_data = pd.read_csv('user_demographics.csv')
demographics_data.head(n=4)
```

```
uid          reg_date   device gender country  age
0    52774929  2018-03-07    and      F    FRA    27
1    84341593  2017-09-22    iOS      F    TUR    22
2    41201055  2017-11-24    and      F    USA    20
3    68477880  2016-12-08    and      F    BRA    18
```

```
paywall_views = pd.read_csv('paywall_views.csv')
paywall_views.head(n=4)
```

```
uid          date purchase sku price
0    52774929 2018-03-11     0   NaN   NaN
1    52774929 2018-03-13     0   NaN   NaN
2    52774929 2018-03-14     0   NaN   NaN
3    68477880 2017-09-05     0   NaN   NaN
```

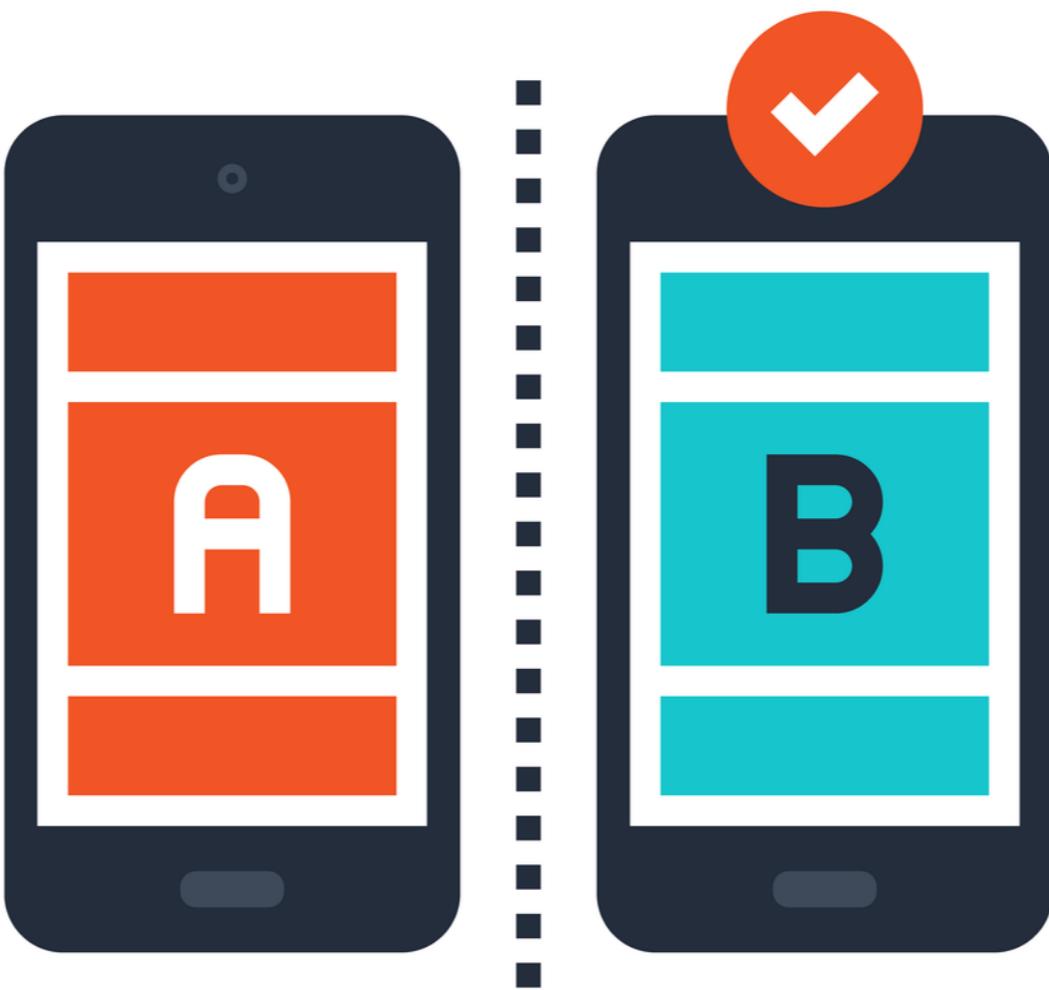
A/B Testing Terminology



Response Variable



Factors & Variants



Experimental Unit



Calculating Experimental Units



Calculating Experimental Units

```
purchase_data = demographics_data.merge(paywall_views,  
                                         how='left', on=['uid'])  
  
purchase_data_agg = purchase_data.groupby(by=['uid'],  
                                         as_index=False)  
  
total_purchases = purchase_data_agg.purchase.sum()  
  
total_purchases.purchase = np.where(  
                                         np.isnan(total_purchases.purchase),  
                                         0, total_purchases.purchase)  
  
total_purchases.purchase.mean()
```

Calculating Experimental Units

```
total_purchases.purchase.mean()
```

```
3.15
```

Calculating Experimental Units

```
min(total_purchases.purchase)
```

```
0.0
```

```
max(total_purchases.purchase)
```

```
17.0
```

Calculating Experimental Units



Calculating Experimental Units

```
purchase_data_agg.date = paywall_views.date.dt.floor('d')
purchase_data_agg = purchase_data.groupby(by=['uid', 'date'], as_index=False)
total_purchases = purchase_data_agg.purchase.sum()
total_purchases.purchase = np.where(np.isnan(total_purchases.purchase), 0, total_purchases.purchase)
total_purchases.purchase.mean()
```

0.0346

```
min(total_purchases.purchase)
```

0.0

```
max(total_purchases.purchase)
```

3.0

Randomness of Experimental Units



Designing Your A/B Test

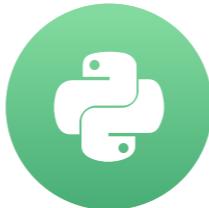


Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

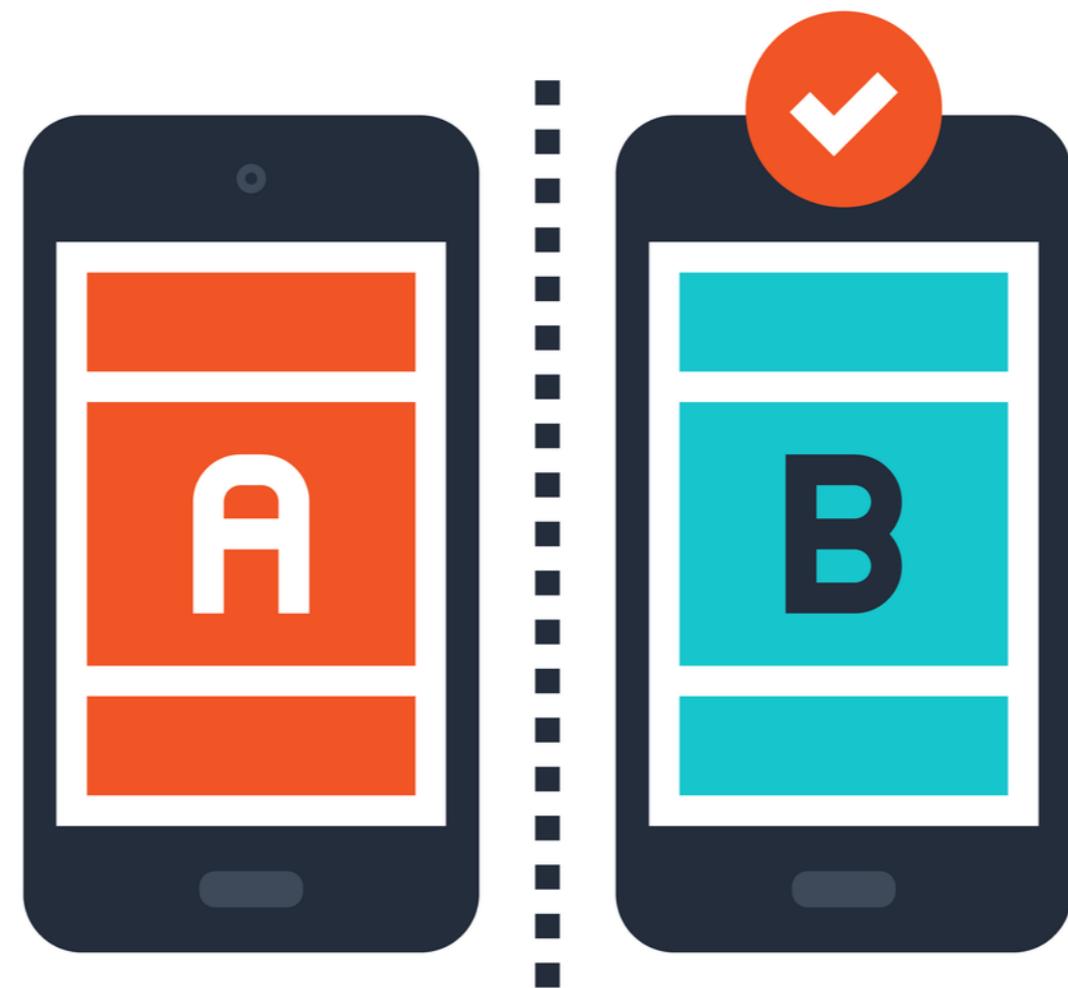
Preparing to run an A/B test

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON



Ryan Grossman
Data Scientist, EDO

Paywall A/B Test Variants



Main Concerns in Designing a Test



Test Sensitivity

%

Evaluating Different Sensitivities



Finding Revenue Per User

```
purchase_data = demographics_data.merge(paywall_views,  
                                         how='left', on=['uid'])  
  
purchase_data_agg = purchase_data.groupby(by=['uid'],  
                                         as_index=False)  
  
total_revenue = purchase_data_agg.price.sum()  
  
total_revenue.price = np.where(np.isnan(total_revenue.price),  
                               0, total_revenue.price)  
  
avg_revenue = total_revenue.price.mean()  
  
avg_revenue
```

16.161

Evaluating Our Sensitivities

```
one_pct_lift = avg_revenue * 1.01  
one_pct_lift
```

```
16.322839545454478
```

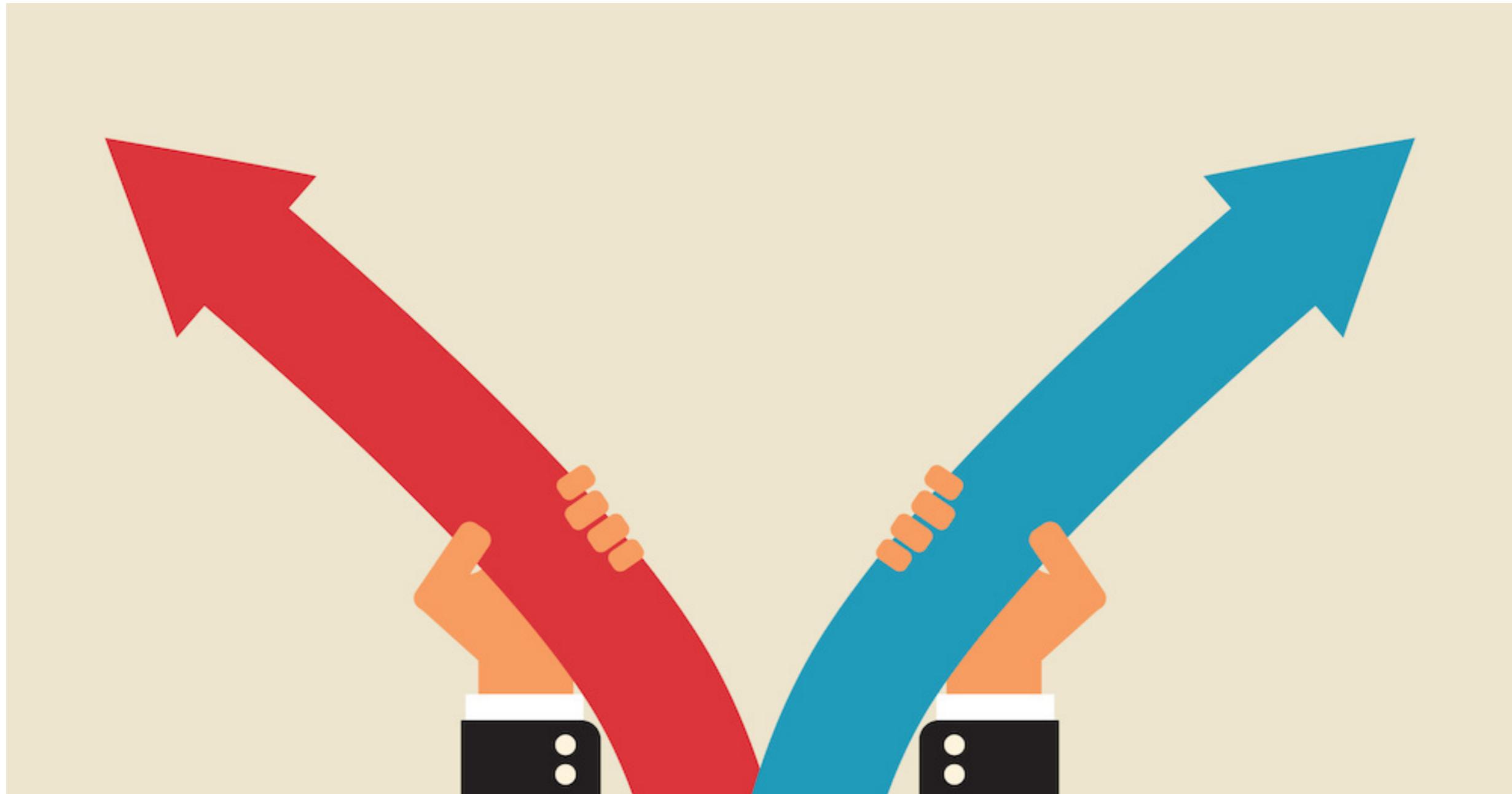
```
ten_pct_lift = avg_revenue * 1.1  
ten_pct_lift
```

```
17.77
```

```
twenty_pct_lift = avg_revenue * 1.2  
twenty_pct_lift
```

```
19.393
```

Understanding Variability in Our Data



Standard Deviation

```
revenue_variation = total_revenue.price.std()  
revenue_variation
```

```
17.520
```

Variability of Revenue Per User

```
revenue_variation / avg_revenue
```

```
1.084
```

Variability of Purchases Per User

```
total_purchases = purchase_data_agg.purchase.sum()  
total_purchases.purchase = np.where(np.isnan(total_purchases.purchase),  
                                     0, total_purchases.purchase)  
  
avg_purchases = total_purchases.purchase.mean()  
avg_purchases
```

3.15

```
purchase_variation = total_purchases.purchase.std()  
purchase_variation
```

2.68

```
purchase_variation / avg_purchases
```

0.850

Choosing our Experimental Unit & Response Variable



Choosing Experimental Unit & Response Variable

```
purchase_data = demographics_data.merge(paywall_views,  
                                         how='inner', on=['uid'])  
  
purchase_data_agg = purchase_data.groupby(by=['uid'],  
                                         as_index=False)  
  
conversion_rate = (sum(purchase_data.purchase) /  
                    purchase_data.purchase.count())  
  
conversion_rate
```

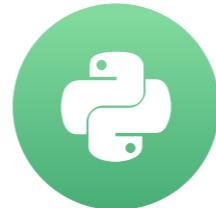
0.347

Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

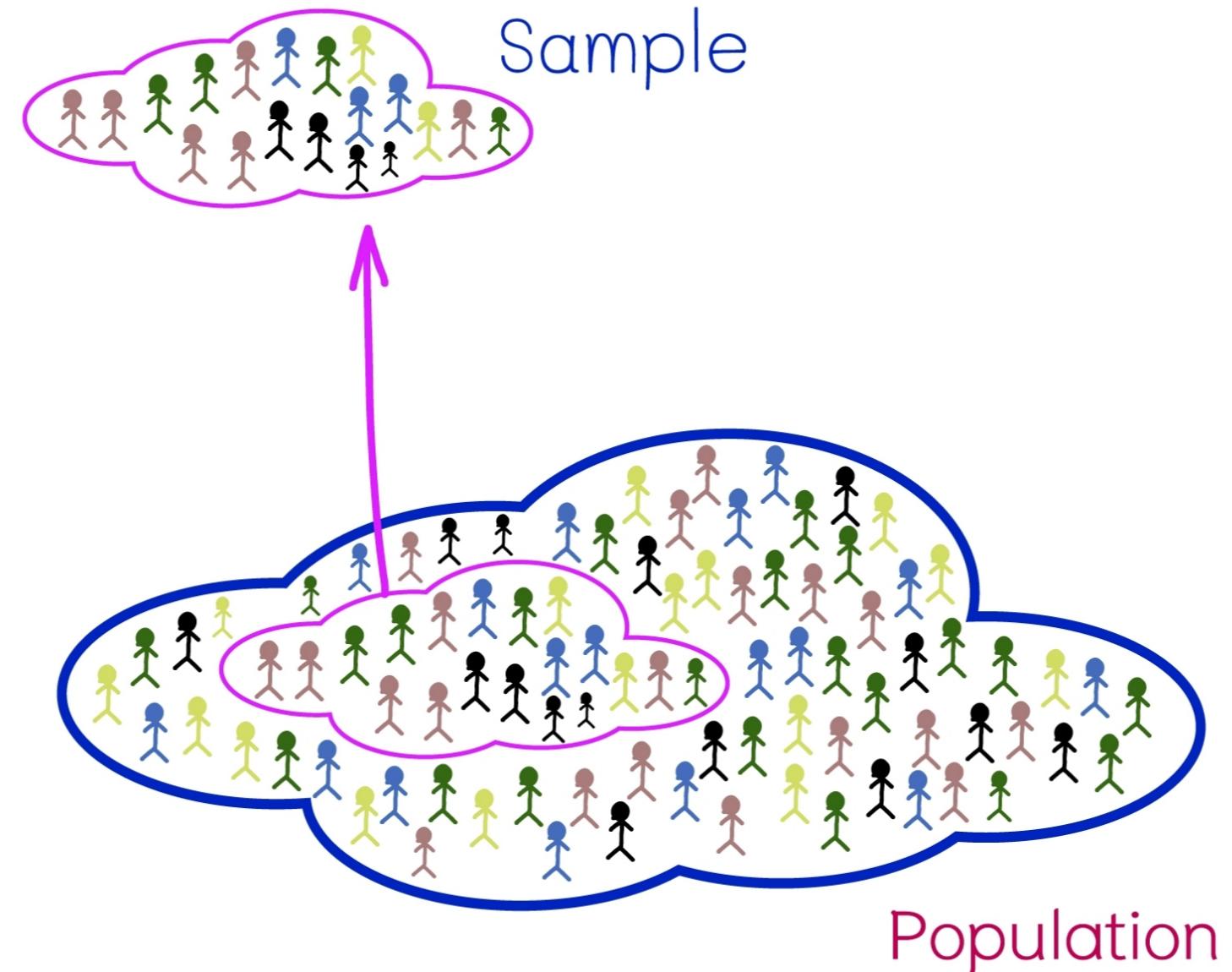
Calculating Sample Sizes

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON



Ryan Grossman
Data Scientist, EDO

Calculating Sample Sizes



Null Hypothesis



Types of Error

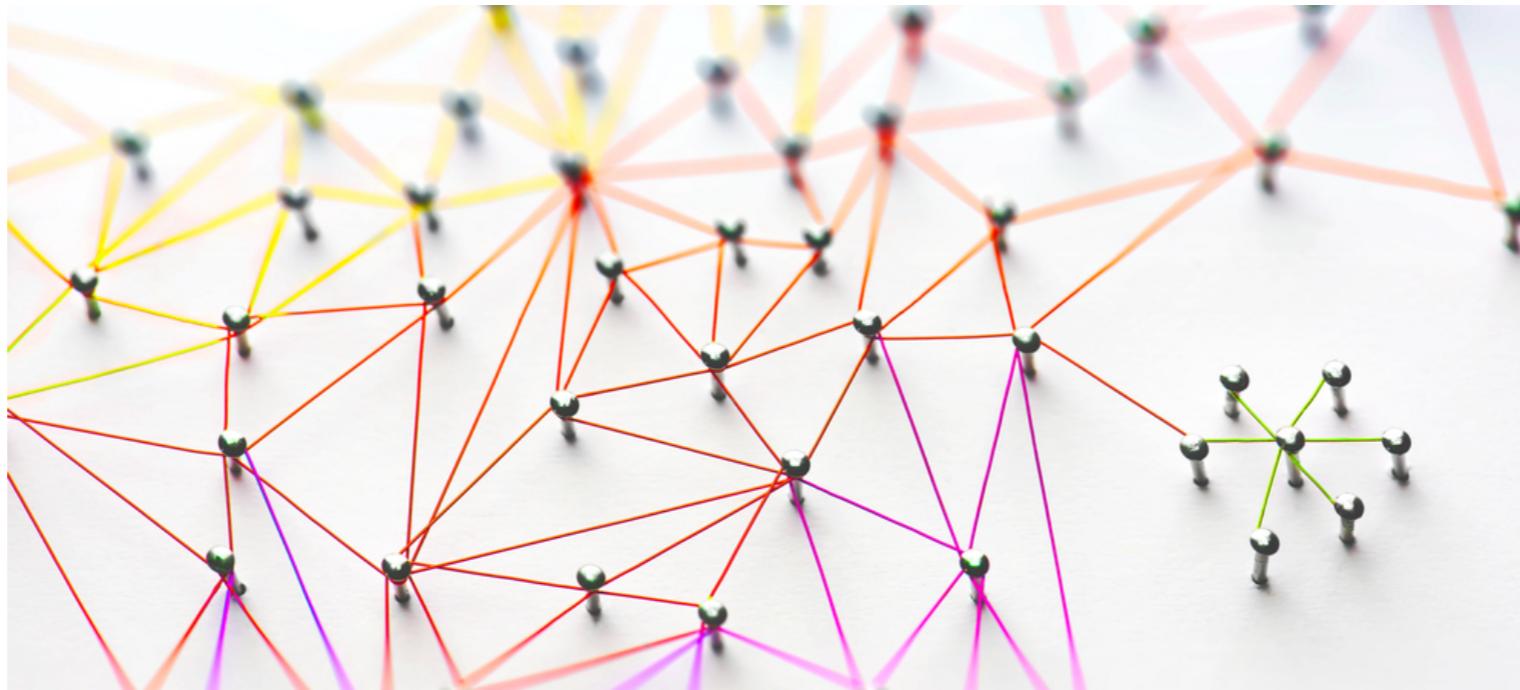
		Null Hypothesis	
		TRUE	FALSE
Null Hypothesis	Accept	Correct	Type II Error
	Reject	Type I Error	Correct

Confidence Level

Statistical Power



Connecting the Different Components



Power Formula

$\alpha = 1 - \text{confidence level}$

$p_1 = \text{Base Rate}, p_2 = \text{Base Rate} + \text{Sensitivity Lift}$

$$qu = \Phi^{-1} \left(1 - \frac{\alpha}{2} \right)$$

$$diff = |p_1 - p_2|, \quad \bar{p} = \frac{(p_1 + p_2)}{2}$$

$$v_1 = p_1 \times (1 - p_1), v_2 = p_2 \times (1 - p_2), \quad \bar{v} = \bar{p} \times (1 - \bar{p})$$

$$Power = \Phi \left(\frac{\sqrt{n} \times diff - qu \times \sqrt{2\bar{v}}}{\sqrt{v_1 + v_2}} \right) + 1 - \Phi \left(\frac{\sqrt{n} \times diff + qu \times \sqrt{2\bar{v}}}{\sqrt{v_1 + v_2}} \right)$$

Sample Size Function

```
def get_power(n, p1, p2, cl):
    alpha = 1 - cl
    qu = stats.norm.ppf(1 - alpha/2)
    diff = abs(p2 - p1)
    bp = (p1 + p2) / 2
    v1 = p1 * (1 - p1)
    v2 = p2 * (1 - p2)
    bv = bp * (1 - bp)
    power_part_one = stats.norm.cdf((n**0.5 * diff - qu * (2 * bv)**0.5) /
                                    (v1 + v2)**0.5)
    power_part_two = 1 - stats.norm.cdf((n**0.5 * diff + qu * (2 * bv)**0.5) /
                                    (v1 + v2)**0.5)
    power = power_part_one + power_part_two
    return(power)
```

```
def get_sample_size(power, p1, p2, cl, max_n = 1000000):
    n = 1
    while n <= max_n:
        tmp_power = get_power(n, p1, p2, cl)
        if tmp_power >= power:
            return n
        else:
            n = n + 1
```

Calculating our Needed Sample Size

```
sample_size_per_group = get_sample_size(0.8,  
                                       conversion_rate, conversion_rate * 1.1, 0.95)  
  
sample_size_per_group
```

45788

Generalness of this Function



Decreasing the Sample Size



Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON