CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Introduction to A/B testing

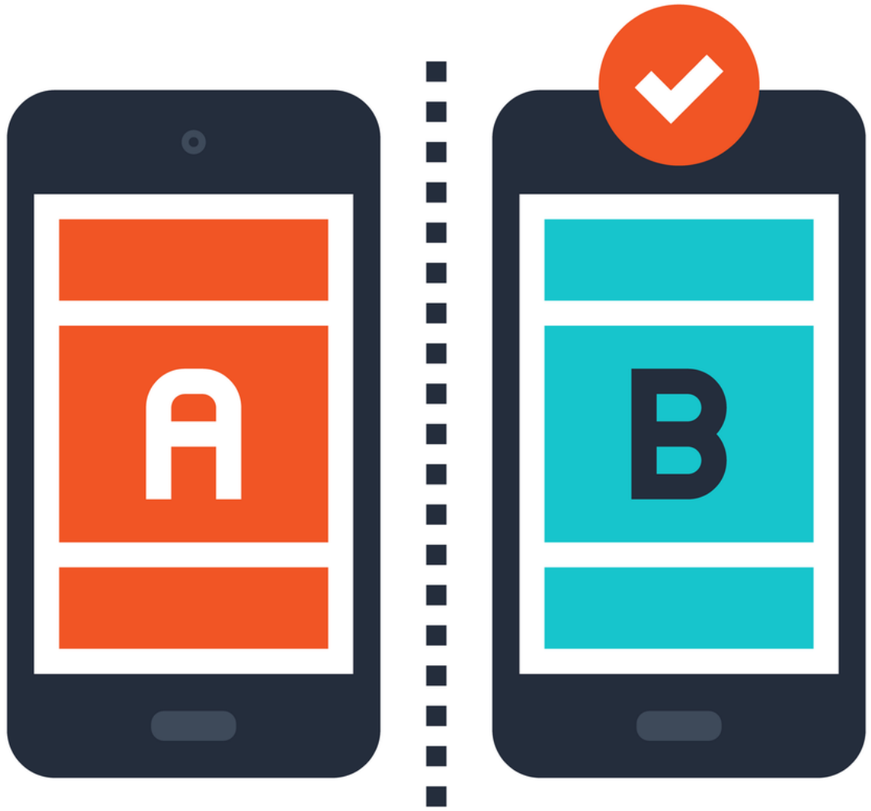
Ryan Grossman
Data Scientist, EDO

Overview


A
B

MANAGEMENT

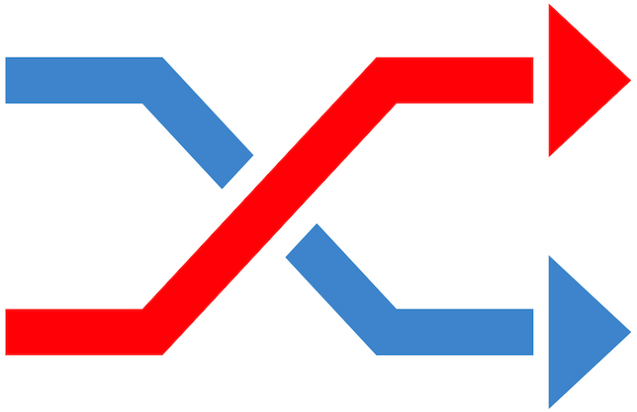
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.



A/B TESTING

A/B Testing Example

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

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Initial A/B test design

Ryan Grossman
Data Scientist, EDO

Increasing Revenue through A/B Testing

Generalizabiliy of A/B Testing

Paywall Views & Demographics Data

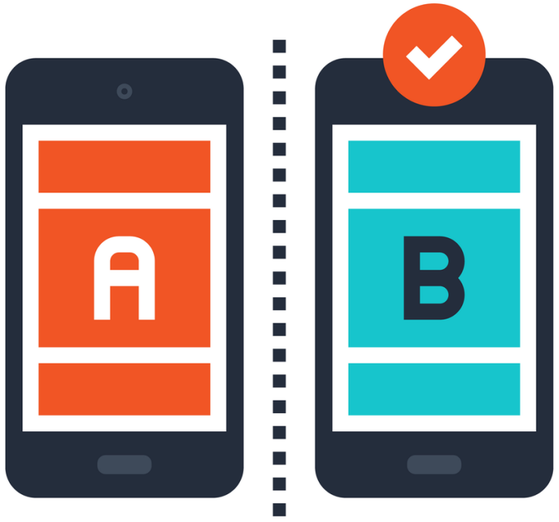
```
In [1]: demographics_data = pd.read_csv('user_demographics.csv')

In [2]: demographics_data.head(n=4)

Out[2]:
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
```

```
In [3]: paywall_views = pd.read_csv('paywall_views.csv')

In [4]: paywall_views.head(n=4)

Out [4]:
        uid        date                    purchase        sku        price
0    52774929    2018-03-11 04:11:01        0            NaN        NaN
1    52774929    2018-03-13 21:28:54        0            NaN        NaN
2    52774929    2018-03-14 12:22:17        0            NaN        NaN
3    84341593    2017-09-25 06:13:14        0            NaN        NaN
```

A/B Testing Terminology

Response Variable

Factors & Variants



Experimental Unit

Calculating Experimental Units

Calculating Experimental Units

```
In [5]: purchase_data = demographics_data.merge(paywall_views,
                                               how='left', on=['uid'])

In [6]: purchase_data_agg = purchase_data.groupby(by=['uid'],
                                                  as_index=False)

In [7]: total_purchases = purchase_data_agg.purchase.sum()

In [8]: total_purchases.purchase = np.where(
                                        np.isnan(total_purchases.purchase),
                                        0, total_purchases.purchase)

In [9]: total_purchases.purchase.mean()
```

Calculating Experimental Units

```
In [9]: total_purchases.purchase.mean()

Out[9]: 3.15
```

Calculating Experimental Units

```
In [10]: min(total_purchases.purchase)

Out[10]: 0.0

In [11]: max(total_purchases.purchase)

Out [12]: 17.0
```

Calculating Experimental Units

Calculating Experimental Units

```
In [13]: purchase_data_agg.date = paywall_views.date.dt.floor('d')

In [14]: purchase_data_agg = purchase_data.groupby(
                                        by=['uid', 'date'],
                                        as_index=False)

In [15]: total_purchases = purchase_data_agg.purchase.sum()

In [16]: total_purchases.purchase = np.where(
                                    np.isnan(total_purchases.purchase),
                                    0, total_purchases.purchase)

In [17]:total_purchases.purchase.mean()

Out[17]: 0.0346

In [18]: min(total_purchases.purchase)

Out[18]: 0.0

In [19]: max(total_purchases.purchase)

Out[19]: 3.0
```

Randomness of Experimental Units

Designing Your A/B Test

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

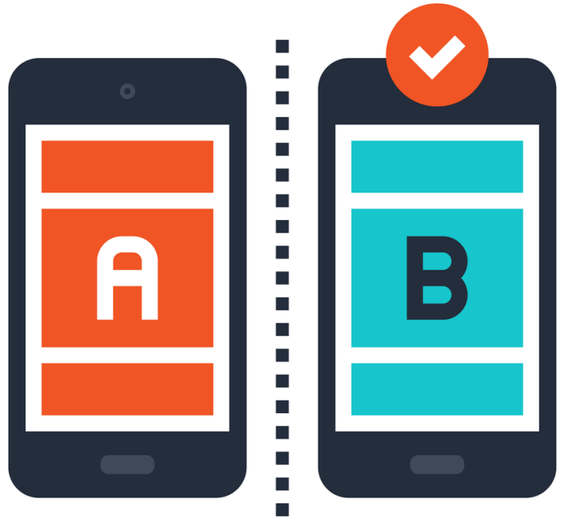
Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

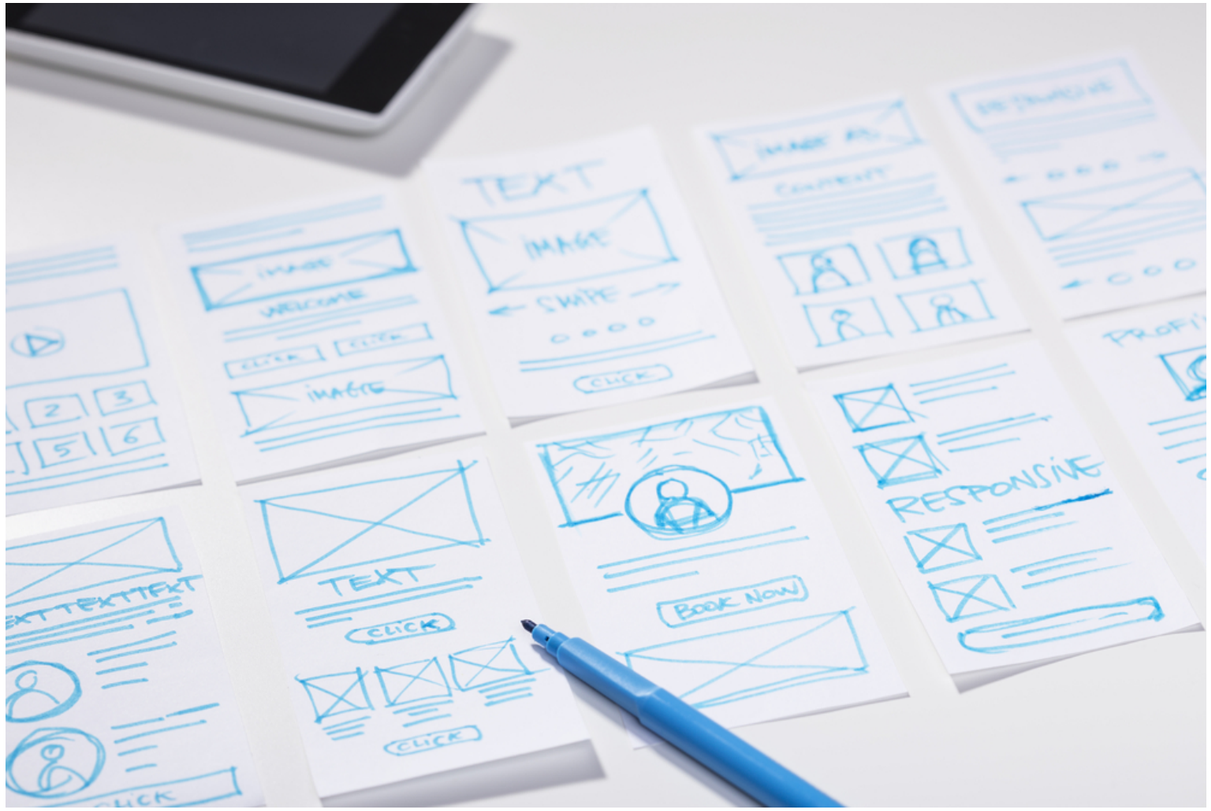
Preparing to run an A/B test

Ryan Grossman
Data Scientist, EDO

Paywall A/B Test Variants



Main Concerns in Designing a Test



Test Sensitivity

Evaluating Different Sensitivities

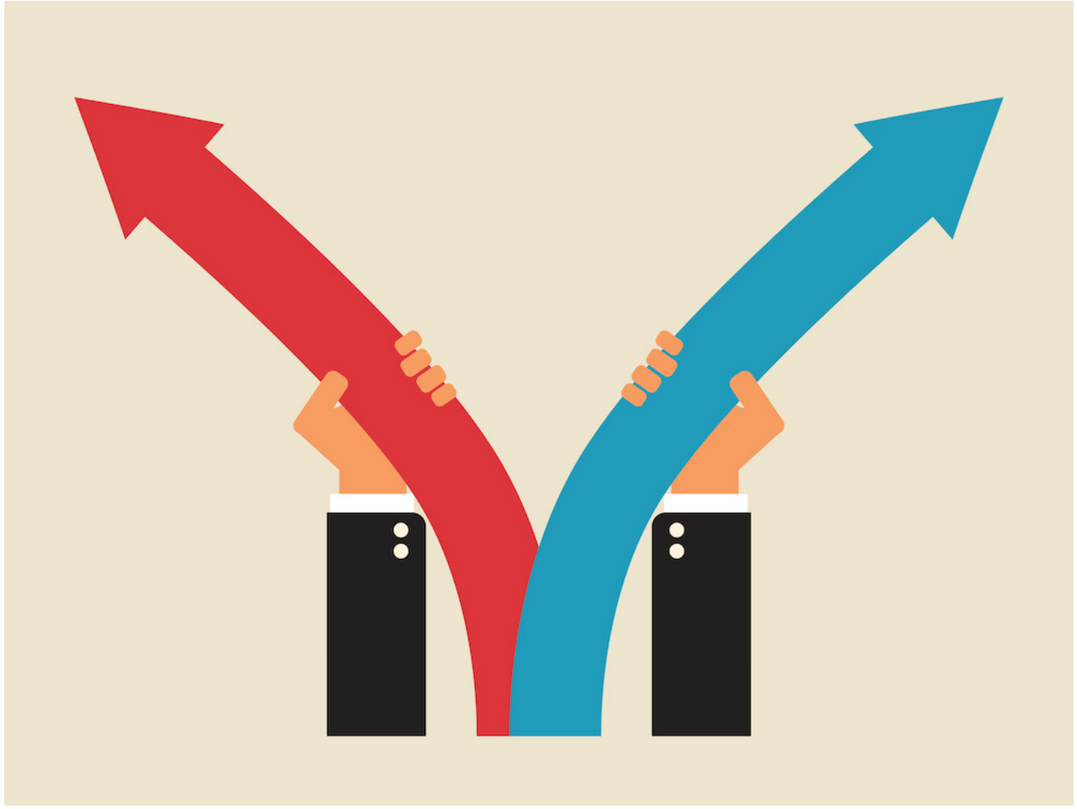
Finding Revenue Per User

```
In [1]: purchase_data = demographics_data.merge(paywall_views,
                                                how='left', on=['uid'])

In [2]: purchase_data_agg = purchase_data.groupby(by=['uid'],
                                                  as_index=False)

In [3]: total_revenue = purchase_data_agg.price.sum()

In [4]: total_revenue.price = np.where(np.isnan(total_revenue.price),
                                       0, total_revenue.price)

In [5]: avg_revenue = total_revenue.price.mean()

In [6]: avg_revenue

Out[6]: 16.161
```

Evaluating Our Sensitivities

```
In [7]: one_pct_lift = avg_revenue * 1.01

In [8]: one_pct_lift

Out[8]: 16.322839545454478

In [9]: ten_pct_lift = avg_revenue * 1.1

In [10]: ten_pct_lift

Out[10]: 17.77

In [11]: twenty_pct_lift = avg_revenue * 1.2

In [12]: twenty_pct_lift

Out [12]: 19.393
```

Understanding Variability in Our Data



Standard Deviation

```
In [13]: revenue_variation = total_revenue.price.std()

In [14]: revenue_variation

Out[14]: 17.520
```

Variability of Revenue Per User

```
In [15]: revenue_variation / avg_revenue

Out[15]: 1.084
```

Variability of Purchases Per User

```
In [16]: total_purchases = purchase_data_agg.purchase.sum()

In [17]: total_purchases.purchase = np.where(
                                        np.isnan(total_purchases.purchase),
                                        0, total_purchases.purchase)

In [18]: avg_purchases = total_purchases.purchase.mean()

In [19]: avg_purchases

Out[19]: 3.15

In [20]: purchase_variation = total_purchases.purchase.std()

In [21]: purchase_variation

Out[21]: 2.68

In [22]: purchase_variation / avg_purchases

Out [22]: 0.850
```

Choosing our Experimental Unit & Response Variable

Choosing our Experimental Unit & Response Variable

```
In [23]: purchase_data = demographics_data.merge(paywall_views,
                                how='inner', on=['uid'])

In [24]: purchase_data_agg = purchase_data.groupby(by=['uid'],
                                                   as_index=False)

In [25]: conversion_rate = (sum(purchase_data.purchase) /
            purchase_data.purchase.count())

In [26]: conversion_rate

Out[26]: 0.347
```

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

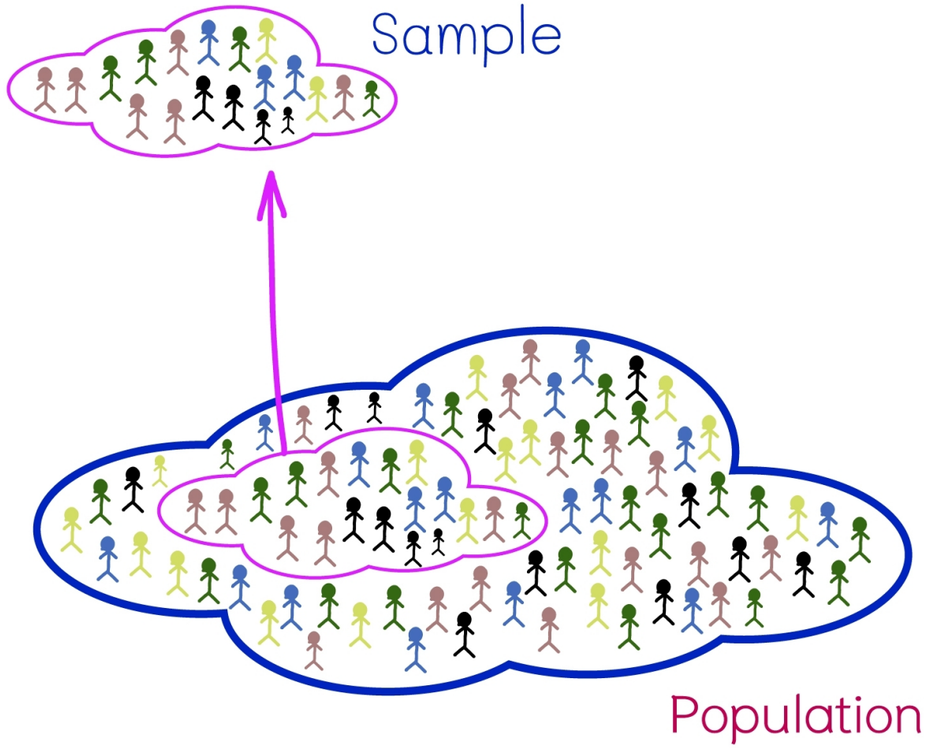
Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Calculating Sample Sizes

Ryan Grossman
Data Scientist, EDO

Calculating Sample Sizes



Null Hypothesis

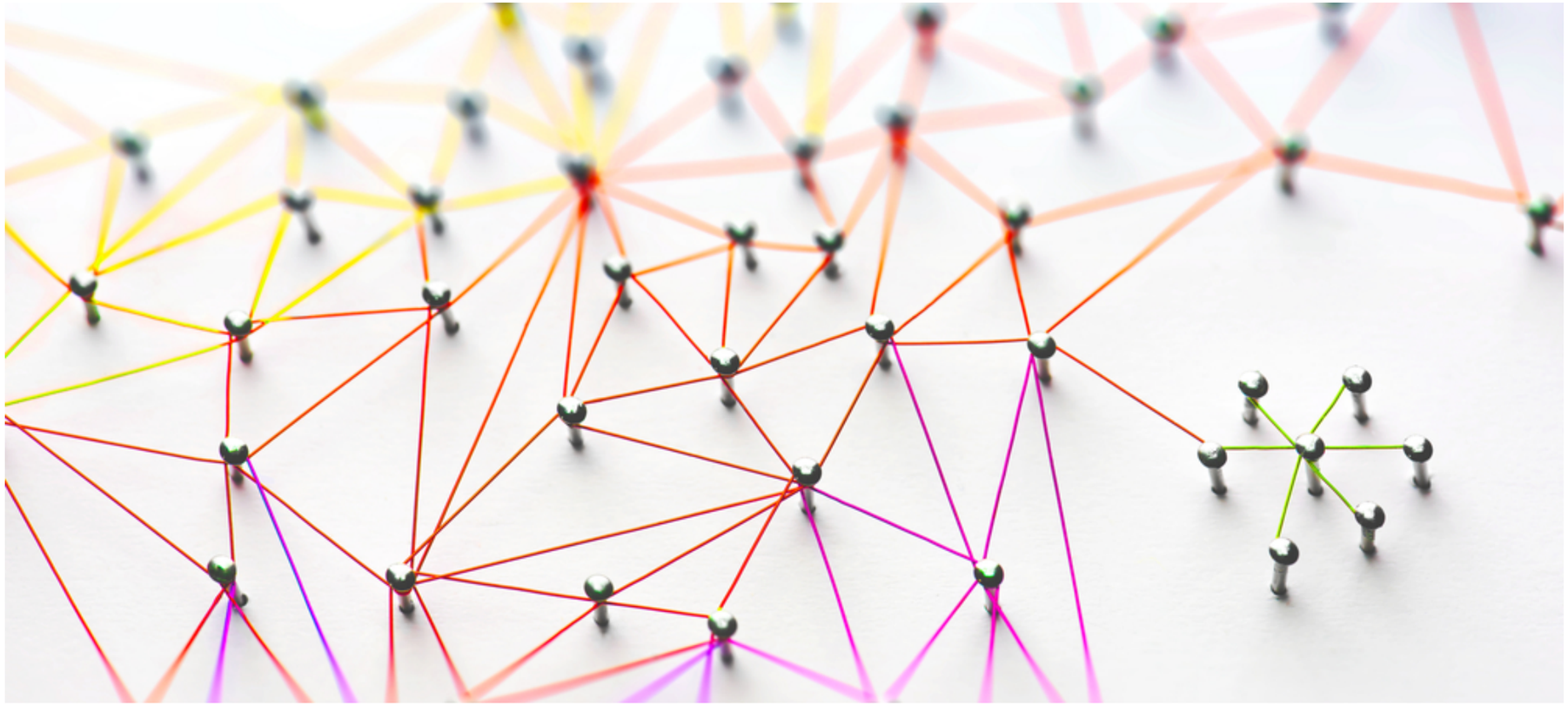
Types of Error

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

Confidence Level

Statistical Power

Connecting the Different Components



Power Formula

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

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

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

$$diff = |p_1 - p_2|$$

$$\bar{p} = \frac{(p_1 + p_2)}{2}$$

$$v_1 = p_1 \times (1 - p_1),\ v_2 = p_2 \times (1 - p_2)$$

$$\bar{v} = \bar{p} \times (1 - \bar{p})$$

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

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

```
In [3]: sample_size_per_group =  get_sample_size(0.8,
            conversion_rate, conversion_rate * 1.1, 0.95)

In [4]: sample_size_per_group

Out[4]: 45788
```

Generalness of this Function

Decreasing the Sample Size

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!