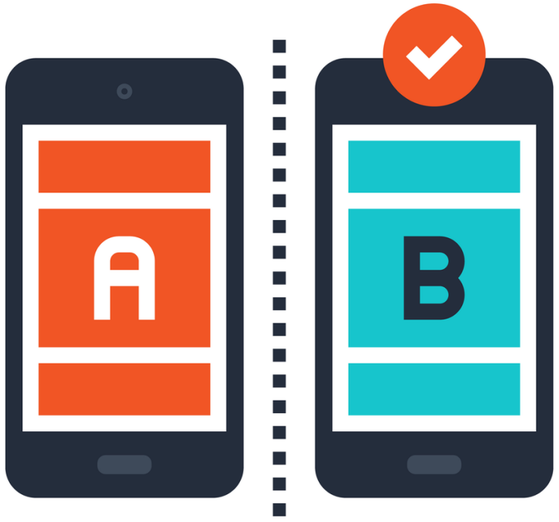
CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Analyzing the A/B test results

Ryan Grossman
Data Scientist, EDO

Analyzing A/B Test Results

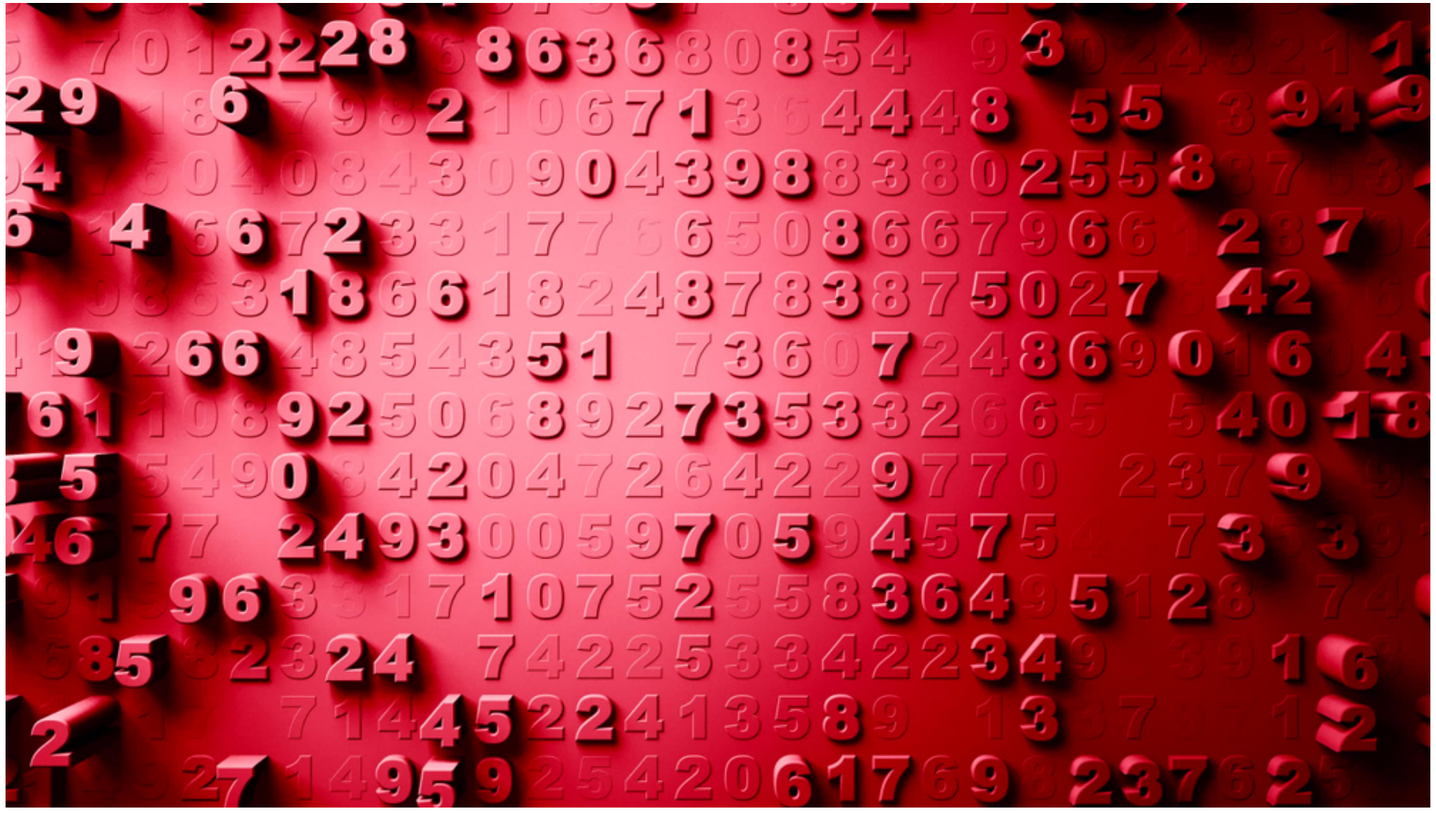
Evaluating our Test



Test Results

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

In [2]: test_results = pd.read_csv('ab_test_results.csv')

In [3]: test_results.date = pd.to_datetime(test_results.date)

In [4]: test_results.head(n=5)

Out[4]:
uid                 date    purchase    sku    price    group
90554036.0    2018-02-27 14:22:12    0    NaN    NaN    C
90554036.0    2018-02-28 08:58:13    0    NaN    NaN    C
90554036.0    2018-03-01 09:21:18    0    NaN    NaN    C
90554036.0    2018-03-02 10:14:30    0    NaN    NaN    C
90554036.0    2018-03-03 13:29:45    0    NaN    NaN    C
```

Confirming Test Results



Confirming Test Results

```
In [5]: test_results_grpd = test_results.groupby(by=['group'],
                                                 as_index=False)

In [6]: test_results_grpd.uid.count()

Out[6]:
    group    uid
0     C      48236
1     V      49867
```

Confirming Test Results

```
In [7]: test_results_demo = test_results.merge(test_demo,
                                  how='inner', on='uid')

In [8]: test_results_grpd = test_results_demo.groupby(by=
                                  ['country','gender', 'device', 'group' ],
                                  as_index=False)

In [9]: test_results_grpd.uid.count()

Out[9]:
country    gender    device    group    uid
BRA        F         and       C        5070
BRA        F         and       V        4136
BRA        F         iOS       C        3359
BRA        F         iOS       V        2817
BRA        M         and       C        3562
BRA        M         and       V        3673
BRA        M         iOS       C        2940
BRA        M         iOS       V        3109
CAN        F         and       C        747
CAN        F         and       V        806
CAN        F         iOS       C        447
```

Finding The Test & Control Group Conversion Rates

```
In [10]: test_results_grpd = test_results_demo.groupby(
                                  by=['group'], as_index=False)

In [11]: test_results_summary = test_results_grpd.agg(
                                      {'purchase': ['count', 'sum']})

In [12]: test_results_summary

Out[12]:
    group    purchase
count    sum
0    C     48236    1657
1    V     49867    2094

In [13]: test_results_summary['conv'] = (
                test_results_summary.purchase['sum'] /
                test_results_summary.purchase['count'])

In [14]: test_results_summary

Out[14]:
    group    purchase    conv
count    sum
0    C     48236    1657    0.034351
1    V     49867   2094     0.041984
```


STATISTICAL SIGNIFICANCE

p-Values

p-value:

- Probability under the Null Hypothesis of obtaining a result as or more extreme than the one observed.
- Represents a measure of the evidence against retaining the Null Hypothesis.

Interpreting a p-Value

p-value	Conclusion
< 0.01	very strong evidence against the Null Hypothesis
0.01 - 0.05	strong evidence against the Null Hypothesis
0.05 - 0.10	very weak evidence against the Null Hypothesis
> 0.1	small to no evidence against the Null Hypothesis

Next Steps

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!

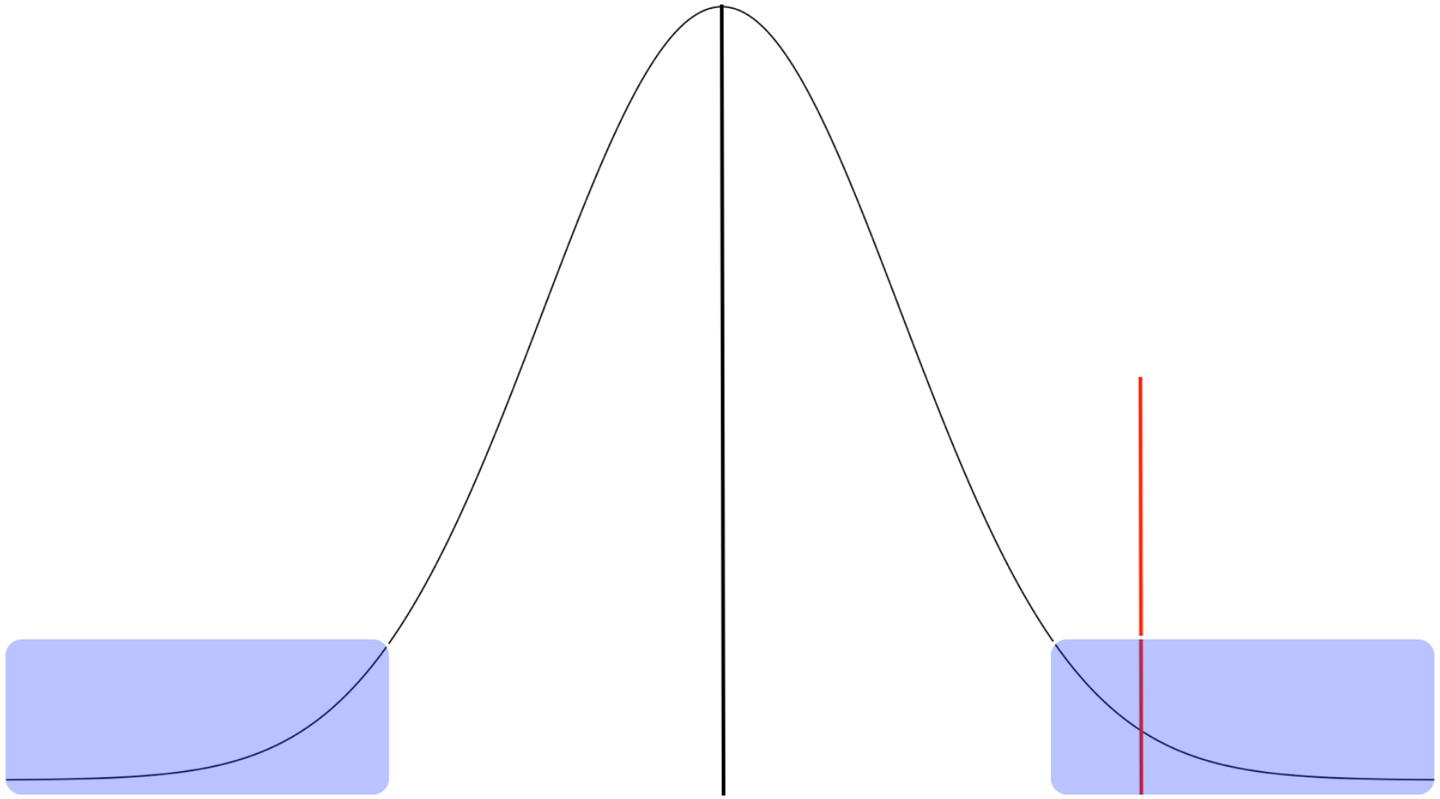
CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Understanding statistical significance

Ryan Grossman
Data Scientist, EDO

Next Steps In Our Analysis

Revisiting Statistical Significance



p-value Function

```
def get_pvalue(con_conv, test_conv,con_size,  test_size,):
    lift =  - abs(test_conv - con_conv)

    scale_one = con_conv * (1 - con_conv) * (1 / con_size)
    scale_two = test_conv * (1 - test_conv) * (1 / test_size)
    scale_val = (scale_one + scale_two)**0.5

    p_value = 2 * stats.norm.cdf(lift, loc = 0, scale = scale_val )

    return p_value
```

Calculating our p-value

```
In [1]: con_conv = 0.034351
In [2]: test_conv = 0.041984
In [3]: con_size = 48236
In [4]: test_size = 49867

In [5]: p_value = get_pvalue(con_conv, test_conv, con_size, test_size)

In [6]: p_value

Out[6]: 4.2572974855869089e-10
```

Finding the Test Power

```
In [7]: get_power(test_size, con_conv, test_conv, 0.95)

Out[7]: 0.999992594137228 19
```


CONFIDENCE
INTERVAL

Confidence Intervals

Confidence Interval

- Provides contextualizaton of the estimation process.
- The conversion rate is a fixed quantity, the estimation is what is variable.

Confidence Intervals

Two Sided Confidence Interval

- $\mu \pm \Phi \left(\alpha + \frac{1-\alpha}{2}\right) * \sigma$
- μ: Estimated Mean
- σ: Estimated Standard Deviation
- α: Desired Confidence Interval Width

Calculating Confidence Intervals

```
def get_ci(lift, alpha, sd):
  val = abs(stats.norm.ppf((1 - alpha)/2))

  lwr_bnd = lift - val * sd
  upr_bnd = lift + val * sd

  return_val = (lwr_bnd, upr_bnd)
  return(return_val)
```

Calculating Confidence Intervals

```
In [8]: lift = test_conv - con_conv

In [9]: sd = ((test_conv * (1 - test_conv)) / test_size +
              (con_conv * (1 - con_conv)) / con_size)**0.5

In [10]: get_ci(lift, 0.95, sd)

Out[10]: (0.00523714629485782 72, 0.0100288537051421 75)
```

Next Steps

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

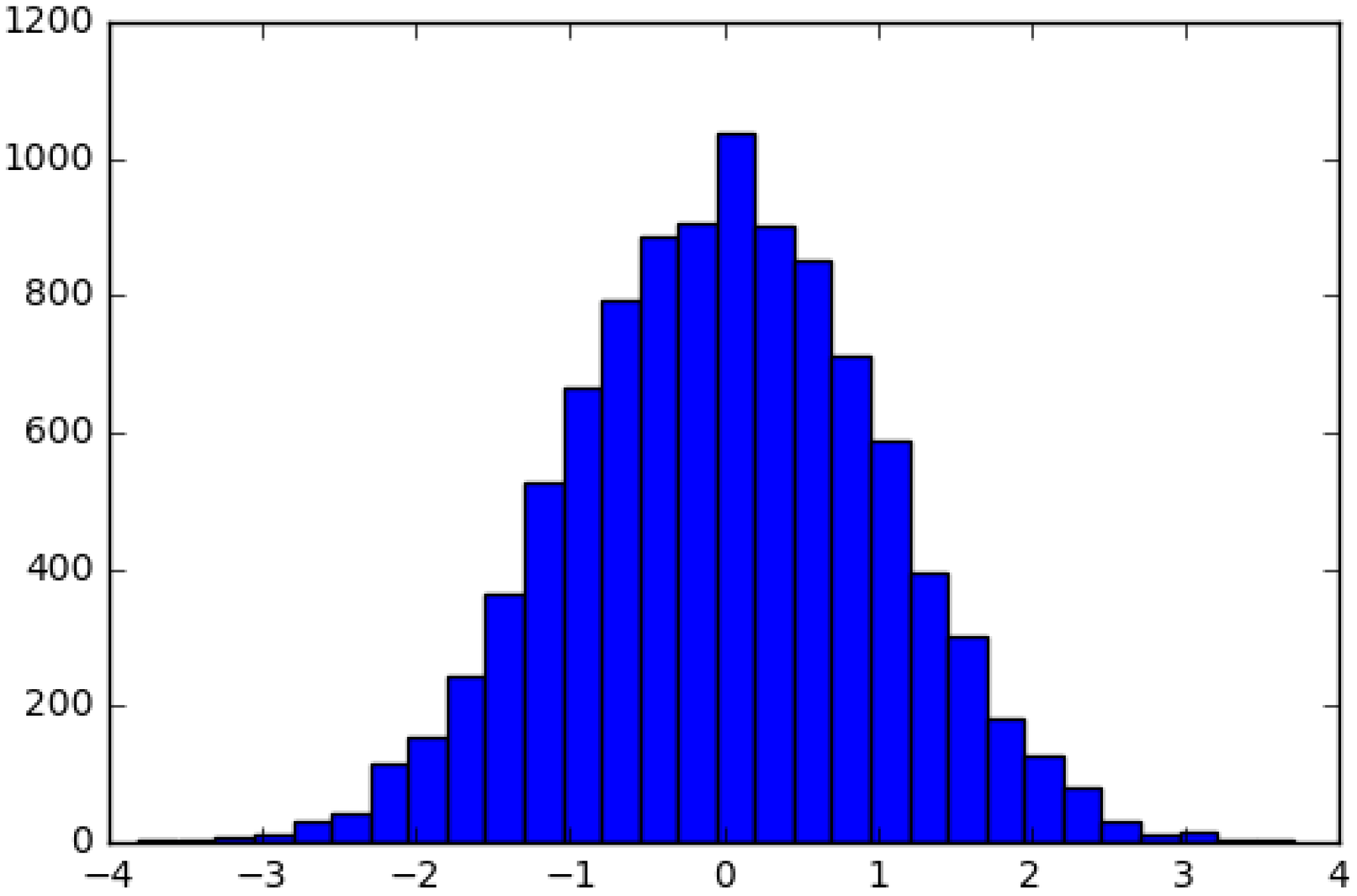
Interpreting your test results

Ryan Grossman
Data Scientist, EDO

Communicating Your Test Results

	Test Group	Control Group
Sample Size	7030	6970
Run Time	2 Weeks	2 Weeks
Mean	3.12	2.69
Variance	3.20	2.64
Estimated Lift: 0.56 *		
Confidence Intervel 0.56 ± 0.4		

**** Significant at the 0.05 Level***

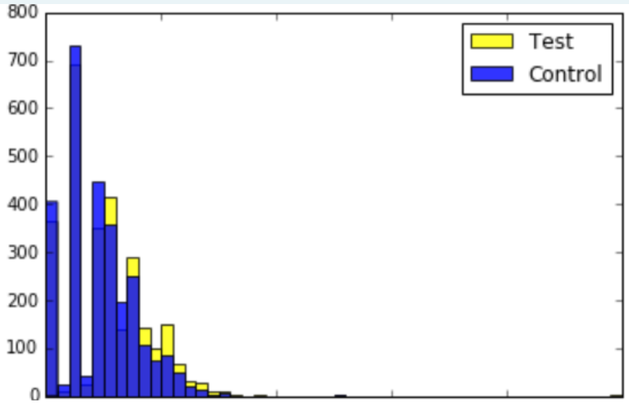

1200
1000
800
600
400
200
0
−4
−3
−2
−1
0
1
2
3
4

Generating Histograms - Data

```
In [1]: test_results_rollup.head(n=10)

Out[1]:
    uid      group    purchase
11128497.0    V        0.000000
11145206.0    V        0.050000
11163353.0    C        0.150000
11215368.0    C        0.000000
11248473.0    C        0.157895
11258429.0    V        0.086957
11271484.0    C        0.071429
11298958.0    V        0.157895
11325422.0    C        0.045455
11340821.0    C        0.040000
```

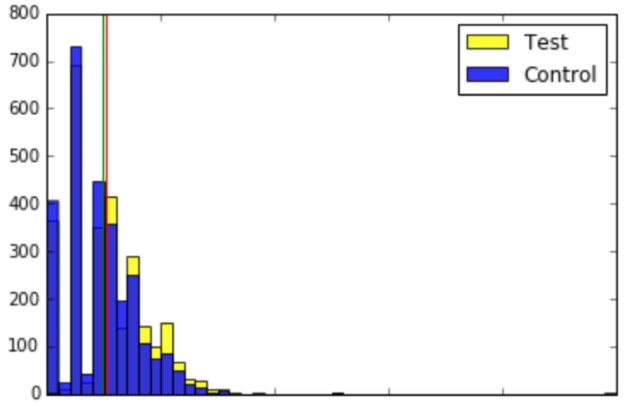
Generating Histograms - Code

```
In [2]: variant_results_rollup = test_results_rollup[
            test_results_rollup.group == 'V']

In [3]: control_results_rollup = test_results_rollup[
            test_results_rollup.group == 'C']

In [4]: plt.hist(variant_results_rollup['purchase'],
            color = 'yellow', alpha = 0.8, bins =50, label = 'Test')

In [5]: plt.hist(control_results_rollup['purchase'],
            color = 'blue', alpha = 0.8, bins = 50, label = 'Control')

In [6]: plt.legend(loc='upper right')

In [7]: plt.show()
```



Adding Lines & Annotations

```
In [8]: plt.axvline(x = np.mean(variant_results_rollup.purchase), color = 'red')

In [9]: plt.axvline(x= np.mean(test_results_rollup.purchase), color = 'green')

In [10]: plt.show()
```

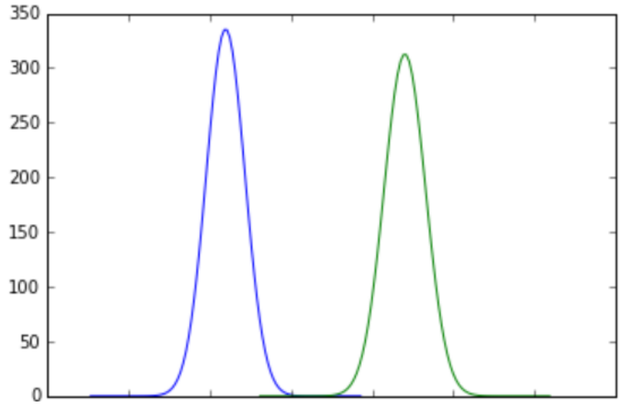


Plotting the Distribution

```
In [11]: mean_control = 0.090965
In [12]: mean_test = 0.102005
In [13]: var_control = (mean_control * (1 - mean_control)) / 58583
In [14]: var_test = (mean_test * (1 - mean_test)) / 56350

In [15]: control_line = np.linspace(-3 * var_control**0.5 +
                mean_control, 3 * var_control**0.5 +
                mean_control, 100)

In [16]: test_line = np.linspace(-3 * var_test**0.5 +
              mean_test, 3 * var_test**0.5
              + mean_test, 100)
```

Plotting the Distribution

```
In [17]: plt.plot(control_line,mlab.normpdf(control_line,
                  mean_control, var_control**0.5))

In [18]: plt.plot(test_line,mlab.normpdf(test_line,
                  mean_test, var_test**0.5))

In [19] plt.show()
```

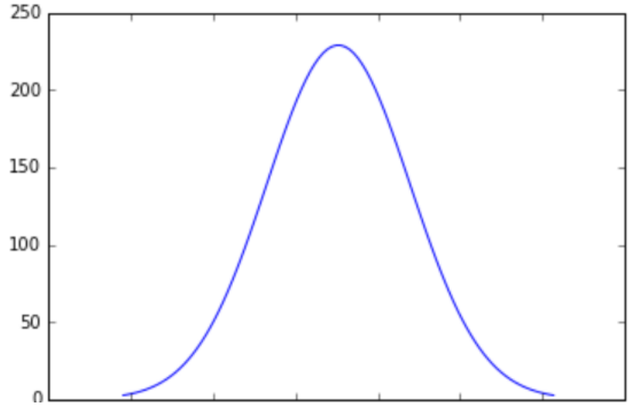


Plotting the Difference of Distributions

```
In [20]: lift = mean_test - mean_control

In [21]: var = var_test + var_control
```

Plotting the Difference of Distributions

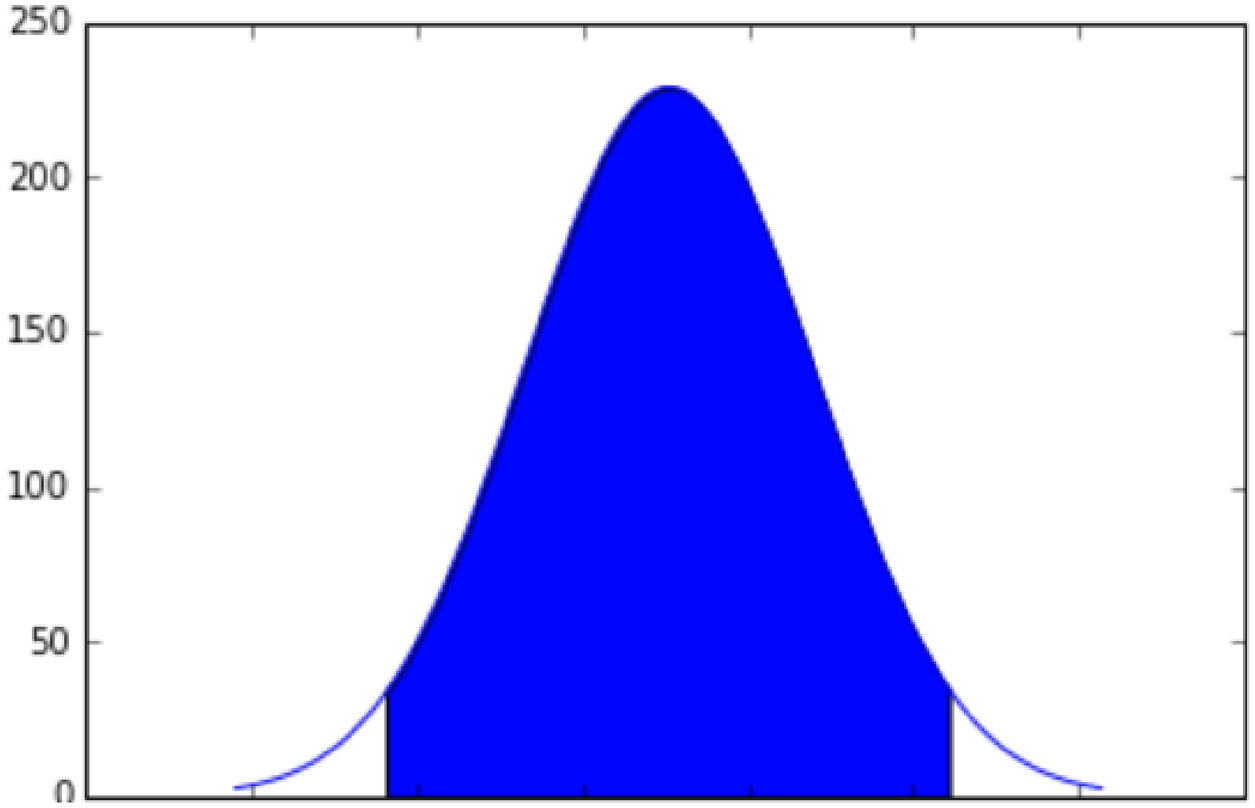
```
In [22]: diff_line = np.linspace(-3 * var**0.5 +
             lift, 3 * var**0.5
             + lift, 100)

In [23]: plt.plot(diff_line,mlab.normpdf(diff_line,
                 lift, var**0.5))

In [24] plt.show()
```



Plotting the Confidence Interval

```
In [25]: section = np.arange(0.007624, 0.01445 , 1/10000)

In [26]: plt.fill_between(section,
                          mlab.normpdf(section,
                          lift, var**0.5))

In [27]: plt.plot(diff_line,mlab.normpdf(diff_line,
                         lift, var**0.5))

In [28]: plt.show()
```

Plotting the Confidence Interval



CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Finale

Ryan Grossman
Data Scientist, EDO

CUSTOMER ANALYTICS & A/B TESTING IN PYTHON

Let's practice!