



# Mobile App A/B Test

## Introduction

There is a mobile app with two variations for an enrollment button. A says 'Secure Free Trial', and B says 'Enroll Now'. The goal is to see if changing to B will result in more clicks and boost the company's sales. In this experiment a one-tailed z-test for comparing two proportions will be used.

```
In [1]: ▶ import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
```

$$H_0 : \mu_B \leq \mu_A$$

$$H_1 : \mu_B > \mu_A$$



## Data

```
In [2]: ▶ ▾ def measure_click(ctr):
            return 1 if np.random.uniform(0,1) < ctr else 0

        ▾ def measure_a():
            return measure_click(ctr=0.005)

        ▾ def measure_b():
            return measure_click(ctr=0.007)
```



## Pilot Study

---

- This pilot study is set up for a power of 80% and a false positive rate of 5%. The practical significance determined by the company is 0.1%. So in addition to statistical significance, B needs to be at least 0.1% better than A. A sample of 1,000 is being used to determine the number of measurements needed.

```
In [3]: ▶ def design_ab_test():
        ▶     def pilot_study(num_pilot_measurements):
        ▶         clicked_pre_a = np.array([measure_a() for _ in range(num_pilot_measurements)])
        ▶         clicked_pre_b = np.array([measure_b() for _ in range(num_pilot_measurements)])
        ▶         sd_1 = np.sqrt(clicked_pre_a.std()**2 + clicked_pre_b.std()**2)
        ▶         return sd_1
        ▶     sd_1 = pilot_study(1000)
        ▶     prac_sig = 0.001
        ▶     num_ind = (2.48*sd_1/prac_sig)**2
        ▶     return int(num_ind)
```

```
In [4]: ▶ np.random.seed(17)
        ▶ num_ind = design_ab_test()
        ▶ num_ind
```

Out[4]: 91561

- 
- So 91,561 individual measurements are needed to confidently detect a meaningful difference between the two variants.
-

## ▼ Run A/B Test

```
In [5]: ▶ def run_ab_test(num_ind):  
        clicked_a = []  
        clicked_b = []  
        for n in range(num_ind // 2):  
            clicked_a.append(measure_a())  
            clicked_b.append(measure_b())  
  
        return np.array(clicked_a), np.array(clicked_b)
```

```
In [6]: ▶ np.random.seed(17)  
        clicked_a, clicked_b = run_ab_test(num_ind)
```

## ▼ Comparing CTRs

```
In [7]: ▶ df_a = pd.DataFrame({'click': clicked_a, 'group': 'A'})  
        df_b = pd.DataFrame({'click': clicked_b, 'group': 'B'})  
        df = pd.concat([df_a, df_b], ignore_index=True)
```

```
In [8]: ▶ df['group'].value_counts()
```

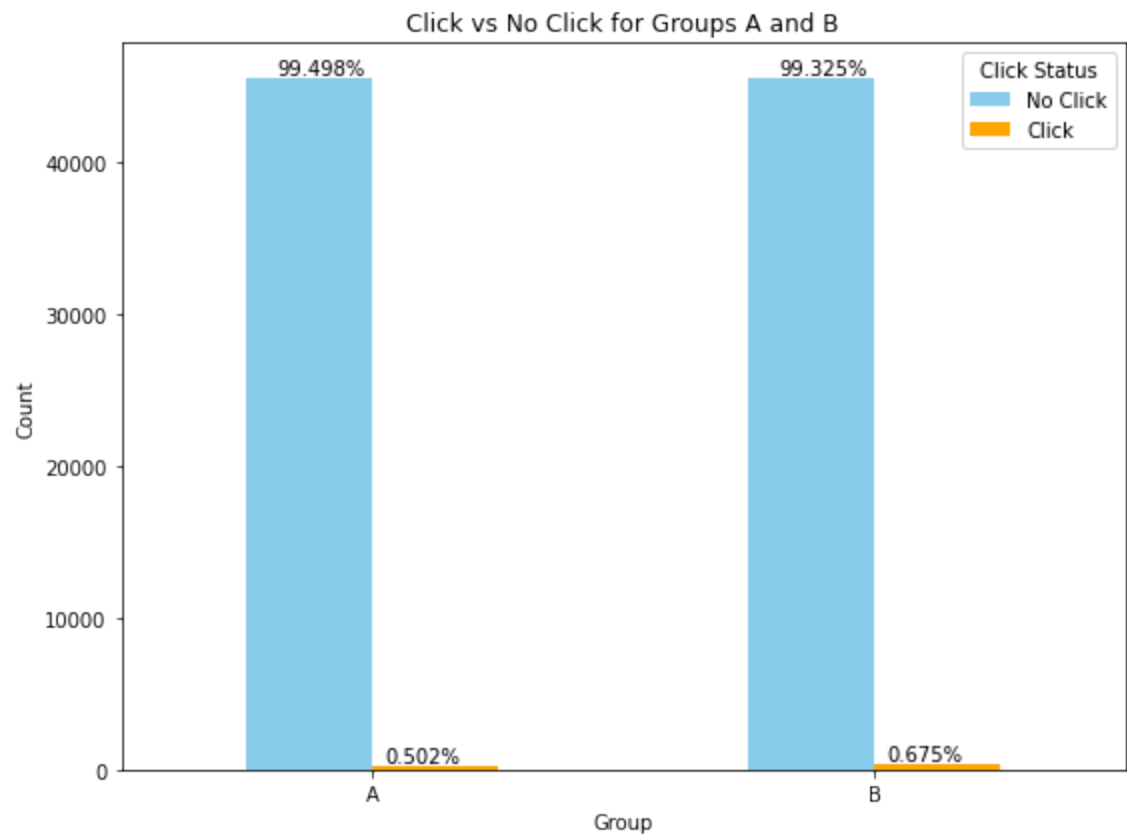
```
Out[8]: group  
A      45780  
B      45780  
Name: count, dtype: int64
```

- 
- By design, the samples sizes for A and B were balanced to provide more accurate results with reduced variability.
-

```
In [9]: ▶ print(f"A was clicked {sum(df[df.group=='A'].click)} times.")
        print(f"B was clicked {sum(df[df.group=='B'].click)} times.")
```

A was clicked 230 times.  
B was clicked 309 times.

```
In [10]: ▶ #↔
```



## Analyzing Results

```
In [11]: ▶ def analyze_a_b_test(clicked_a, clicked_b):  
    mean_a = clicked_a.mean()  
    mean_b = clicked_b.mean()  
    std_a = clicked_a.std()  
    std_b = clicked_b.std()  
    m = mean_b - mean_a  
    se = np.sqrt((std_a**2+std_b**2)/num_ind)  
    z = m/se  
  
    return z, mean_a, mean_b, std_a, std_b
```

```
In [12]: ▶ z, mean_a, mean_b, std_a, std_b = analyze_a_b_test(clicked_a, clicked_b)
```

```
In [13]: ▶ np.random.seed(17)  
    clicked_a, clicked_b = run_ab_test(num_ind)  
    print(z.round(2))
```

4.83

- The value of the test statistic being greater than the critical of 1.64 indicates strong evidence against the null hypothesis.

```
In [14]: ▶ alpha=0.05  
    p_value = 1 - norm.cdf(z)  
    ▼ if p_value < alpha:  
        print('Reject the null hypothesis')
```

Reject the null hypothesis

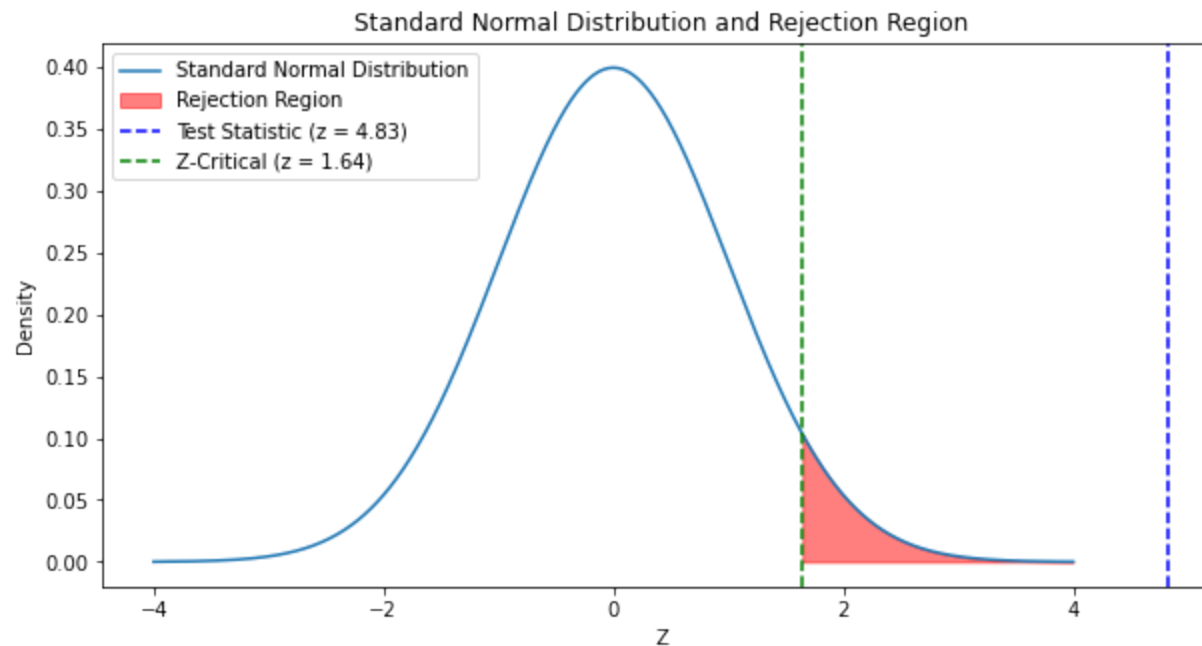
- Because the p value is less than our chosen alpha of 0.05 we reject the null hypothesis and conclude the mean ctr of B is significantly higher than the mean ctr of A.

## Standard Normal Distribution with Rejection Region

```
In [15]: x = np.linspace(-4, 4, 1000)
y = norm.pdf(x)

z_critical = 1.64

plt.figure(figsize=(10, 5))
plt.plot(x, y, label="Standard Normal Distribution")
plt.fill_between(x, y, where=(x >= z_critical), color="red", alpha=0.5, label="Rejection Region")
plt.axvline(z, color="blue", linestyle="--", label=f"Test Statistic (z = {z:.2f})")
plt.axvline(z_critical, color="green", linestyle="--", label=f"Z-Critical (z = {z_critical})")
plt.title("Standard Normal Distribution and Rejection Region")
plt.xlabel("Z")
plt.ylabel("Density")
plt.legend()
plt.show()
```



- 
- This graph shows our test statistic,  $z$ , being well within the rejection region.
- 

## ▼ 95% Confidence Interval

```
In [16]: ▶ mean_diff = mean_b - mean_a
std_err = np.sqrt((std_a**2 + std_b**2) / num_ind)

lower_bound = mean_diff - 1.64 * std_err
upper_bound = mean_diff + 1.64 * std_err

print(f"95% Confidence Interval: ({lower_bound:.6f}, {upper_bound:.6f})")
```

95% Confidence Interval: (0.001139, 0.002312)

- The lower bound of the CI is higher than our practical significance of 0.001.

## ▼ Conclusion

- In conclusion, the A/B test above indicates variant B performs significantly better than variant A both statistically and practically. Therefore the mean CTR of B is greater than A, and that difference is large enough for motivation to change the product.