The Math Behind A/B Testing with Example Python Code

Nguyen Ngo
Aug 3, 2018 · 14 min read

Animation by Gal Shir

While taking the A/B testing course by Google on Udacity, I had some questions about some of the mathematical steps that were not clearly covered by the course. This is understandable because the course was intended to be a compressed and concise overview. To resolve my questions, I turned to other sources on the web and decided to summarize what I learned in this article.

Outline for A/B Tests

1. Set up the experiment.
2. Run the test and record the success rate for each group.
3. Plot the distribution of the difference between the two samples.
4. Calculate the statistical power.
5. Evaluate how sample size affects A/B tests.

1. Set Up The Experiment

We will run an A/B test for a hypothetical company that is trying to increase the amount of users that sign up for a premium account. The goal of running an A/B test is to evaluate if a change in a website will lead to improved performance in a specific metric. You may decide to test very simple alternatives such as changing the look of a single button on a webpage or testing different layouts and headlines. You could also run an A/B test on multi-step processes which may have many differences. Examples of this include the steps required in signing up a new user or processing the sale on an online marketplace. A/B testing is a huge subject and there are many techniques and rules on setting up an experiment. In addition to the Udacity course, here are a few other useful resources below:

- Optimizely's Glossary for A/B Testing
- Evan Miller's A/B Testing Articles
- Facebook's Help Webpage on Ads Measurement

For this article, I will keep it simple so that we can just focus on the math.

Baseline Conversion Rate and Lift

Before running the test, we will know the **baseline conversion rate** and the **desired lift** or increase in signups that we would like to test. The baseline conversion rate is the current rate at which we sign up new users under the existing design. For our example, we want to use our test to confirm that the changes we make to our signup process will result in at least a 2% increase in our sign up rate. We currently sign up 10 out of 100 users who are offered a premium account.

```
# code examples presented in Python
bcr = 0.10  # baseline conversion rate
d_hat = 0.02  # difference between the groups
```

Control Group (A) and Test Group (B)

Typically, the total number of users participating in the A/B test make up a small percentage of the total amount of users. Users are randomly selected and assigned to either a control group or a test group. The sample size that you decide on will determine how long you might have to wait until you have collected enough. For example, websites with large audiences may be able to collect enough data very quickly, while other websites may have to wait a number of weeks. There are some events that happen rarely even for high-traffic websites, so determining the necessary sample size will inform how soon you can assess your experiment and move on to improving other metrics.

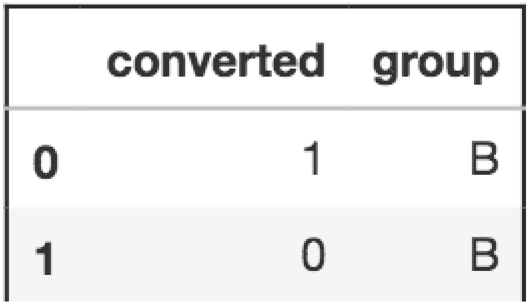
Initially, we will collect 1000 users for each group and serve the current signup page to the control group and a new signup page to the test group.

```
# A is control; B is test
N_A = 1000
N_B = 1000
```

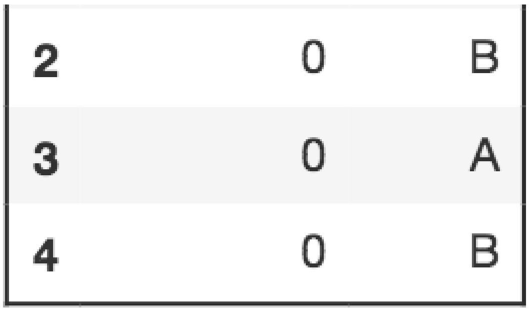
2. Run the Test

Because this is a hypothetical example, we will need "fake" data to work on. I wrote a function that will generate data for our simulation. The script can be found at my Github repo here.

```
ab_data = generate_data(N_A, N_B, bcr, d_hat)
```



	converted	group
0	1	B
1	0	B

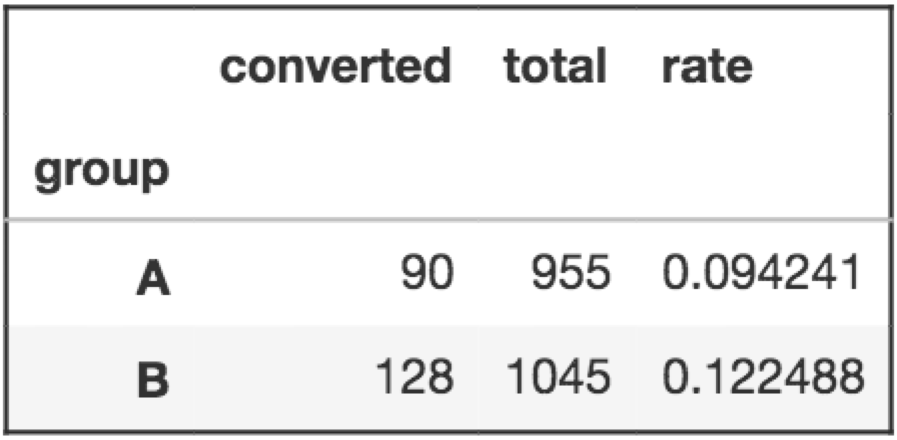


2	0	B
3	0	A
4	0	B

The `generate_data` function returned the table on the left. **Only the first five rows are shown.** The `converted` column indicates whether a user signed up for the premium service or not with a 1 or 0, respectively. The `A` group will be used for our control group and the `B` group will be our test group.

Let’s look at a summary of the results using the pivot table function in Pandas.

```
ab_summary = ab_data.pivot_table(values='converted', index='group',
aggfunc=np.sum)

# add additional columns to the pivot table
ab_summary['total'] = ab_data.pivot_table(values='converted',
index='group', aggfunc=lambda x: len(x))
ab_summary['rate'] = ab_data.pivot_table(values='converted',
index='group')
```



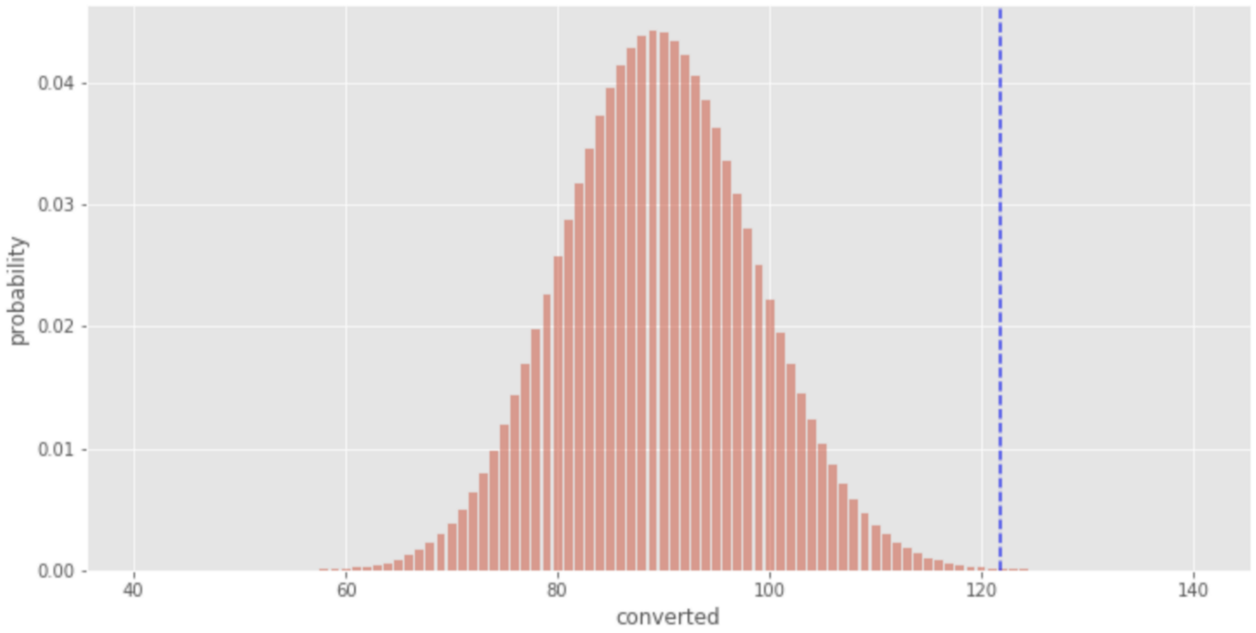
	converted	**total**	**rate**
group			
A	90	955	0.094241
B	128	1045	0.122488

It looks like the difference in conversion rates between the two groups is 0.028 which is greater than the lift we initially wanted of 0.02. **This is a good sign but this is not enough evidence for us to confidently go with the new design.** At this point we have not measured how confident we are in this result. This can be mitigated by looking at the distributions of the two groups.

3. Compare the Two Groups

We can compare the two groups by plotting the distribution of the control group and calculating the probability of getting the result from our test group. We can assume that the distribution for our control group is binomial because the data is a series of Bernoulli trials, where each trial only has two possible outcomes (similar to a coin flip).

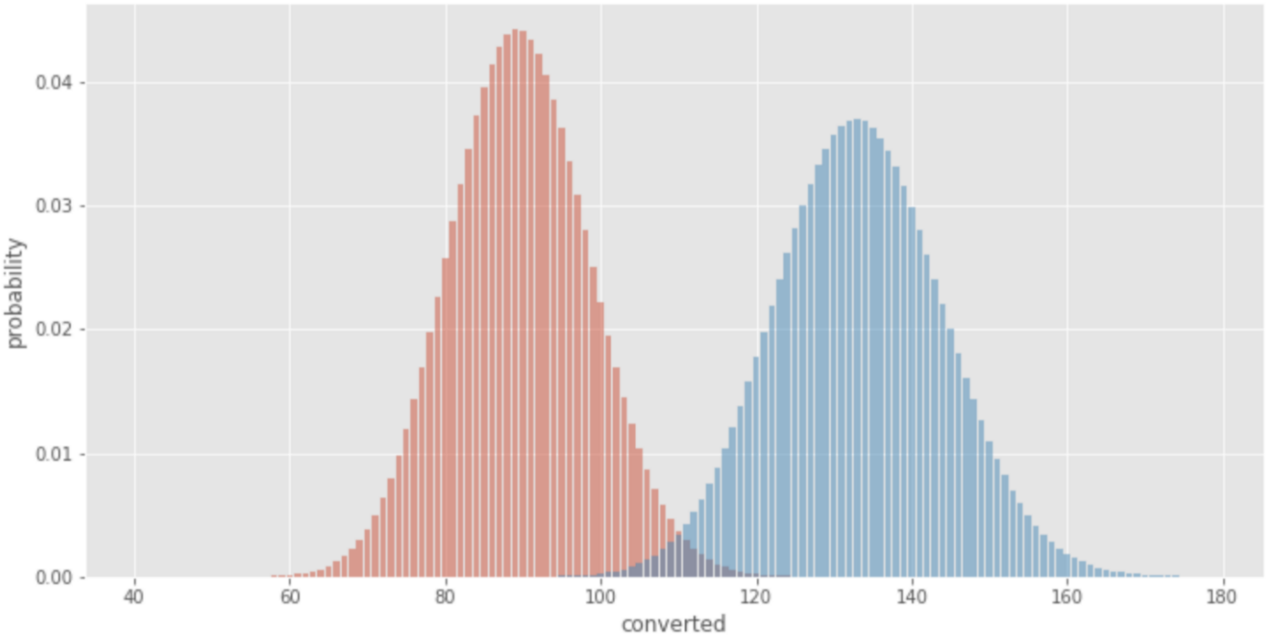
```
fig, ax = plt.subplots(figsize=(12,6))
x = np.linspace(A_converted-49, A_converted+50, 100)
y = scs.binom(A_total, A_cr).pmf(x)
ax.bar(x, y, alpha=0.5)
ax.axvline(x=B_cr * A_total, c='blue', alpha=0.75, linestyle='--')
plt.xlabel('converted')
plt.ylabel('probability')
```



The distribution for the control group is shown in red and the result from the test group is indicated by the blue dashed line. We can see that the probability of getting the result from the test group was very low. **However, the probability does not convey the confidence level of the results.** It does not take the sample size of our test group into consideration. Intuitively, we would feel more confident in our results as our sample sizes grow larger. Let's continue and plot the test group results as a binomial distribution and compare the distributions against each other.

Binomial Distribution

```
fig, ax = plt.subplots(figsize=(12,6))
xA = np.linspace(A_converted-49, A_converted+50, 100)
yA = scs.binom(A_total, p_A).pmf(xA)
ax.bar(xA, yA, alpha=0.5)
xB = np.linspace(B_converted-49, B_converted+50, 100)
yB = scs.binom(B_total, p_B).pmf(xB)
ax.bar(xB, yB, alpha=0.5)
plt.xlabel('converted')
plt.ylabel('probability')
```



Binomial distributions for the control (red) and test (blue) groups

We can see that the test group converted more users than the control group. We can also see that the peak of the test group results is lower than the control group. How do we interpret the difference in peak probability? We should focus instead on the conversion rate so that we have an apples-to-apples comparison. In order to calculate this, we need to standardize the data and compare the probability of successes, ***p***, for each group.

Bernoulli Distribution and the Central Limit Theorem

To do this, first, consider the Bernoulli distribution for the control group.

$$X \sim Bernoulli(p)$$

where p is the conversion probability of the control group.

According to the properties of the Bernoulli distribution, the mean and variance are as follows:

$$E(X) = p$$

$$Var(X) = p(1-p)$$

According to the central limit theorem, by calculating many sample means we can approximate the true mean of the population, ***μ***, from which the data for the control group was taken. The distribution of the sample means, ***p***, will be **normally distributed** around the true mean with a standard deviation equal to the standard error of the mean. The equation for this is given as:

$$\sigma_{\bar{x}} = \frac{s}{\sqrt{n}} = \frac{\sqrt{p(1-p)}}{\sqrt{n}}$$

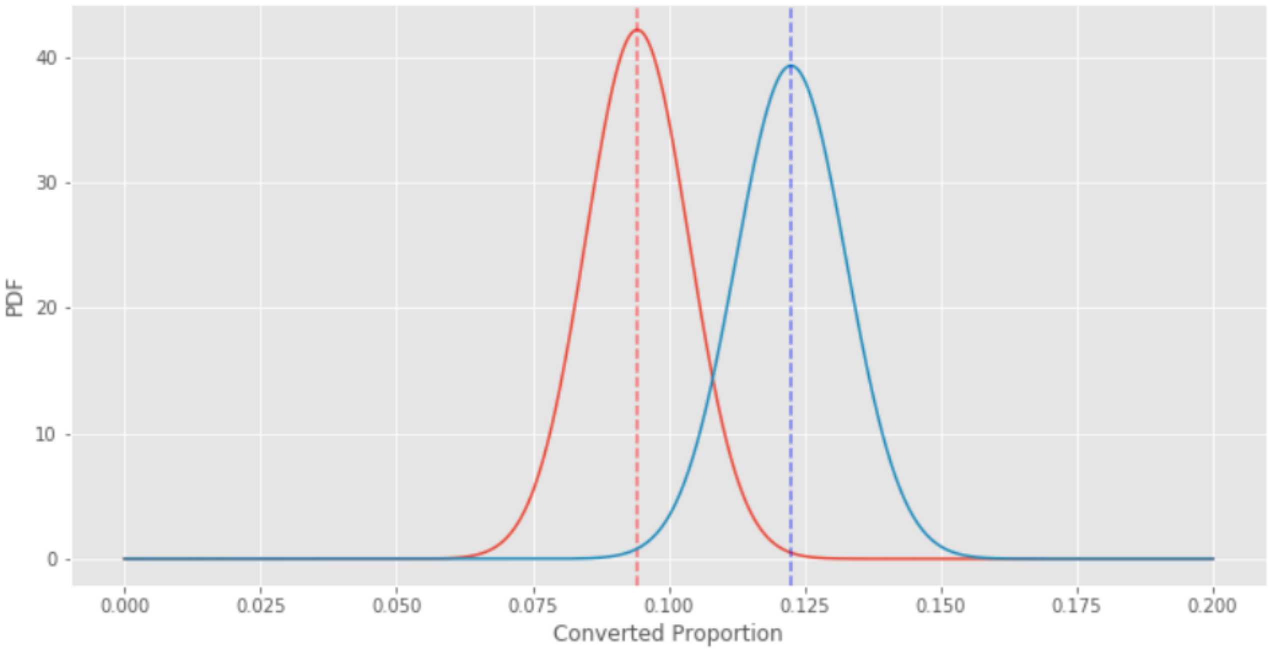
Therefore, we can represent both groups as a normal distribution with the following properties:

$$\hat{p} \sim Normal\left(\mu = p,\ \sigma = \frac{\sqrt{p(1-p)}}{\sqrt{n}}\right)$$

The same can be done for the test group. So, we will have two normal distributions for `p_A` and `p_B`.

```
# standard error of the mean for both groups
SE_A = np.sqrt(p_A * (1-p_A)) / np.sqrt(A_total)
SE_B = np.sqrt(p_B * (1-p_B)) / np.sqrt(B_total)

# plot the null and alternative hypothesis
fig, ax = plt.subplots(figsize=(12,6))

x = np.linspace(0, .2, 1000)

yA = scs.norm(p_A, SE_A).pdf(x)
ax.plot(xA, yA)
ax.axvline(x=p_A, c='red', alpha=0.5, linestyle='--')

yB = scs.norm(p_B, SE_B).pdf(x)
ax.plot(xB, yB)
ax.axvline(x=p_B, c='blue', alpha=0.5, linestyle='--')
```

```
    plt.xlabel('Converted Proportion')
    plt.ylabel('PDF')
```



Control (red) and test (blue) groups as normal distributions for the proportion of successes

The dashed lines represent the mean conversion rate for each group. The distance between the red dashed line and the blue dashed line is equal to mean difference between the control and test group. `d_hat` is the distribution of the difference between random variables from the two groups.

$$\hat{d} = \hat{p}_B - \hat{p}_A$$

Variance of the Sum

Recall that the null hypothesis states that the **difference in probability** between the two groups is zero. Therefore, the mean for this normal distribution will be at zero. The other property we will need for the normal distribution is the standard deviation or the variance. (Note: The variance is the standard deviation squared.) The variance of the difference will be dependent on the variances of the probability for both groups.

A basic property of variance is that the variance of the sum of two random independent variables is the sum of the variances.

$$Var(X+Y) = Var(X) + Var(Y)$$

$$Var(X-Y) = Var(X) + Var(Y)$$

This means that the null hypothesis and alternative hypothesis will have the same variance which will be the sum of the variances for the control group and the test group.

$$Var(\hat{d}) = Var(\hat{p}_B - \hat{p}_A) = Var(\hat{p}_A) + Var(\hat{p}_B) = \frac{p_A(1-p_A)}{n_A} + \frac{p_B(1-p_B)}{n_B}$$

The standard deviation can then be calculated as:

$$\sigma = \sqrt{Var(\hat{d})} = \sqrt{\frac{p_A(1-p_A)}{n_A} + \frac{p_B(1-p_B)}{n_B}}$$

If we put this equation in terms of standard deviation for the Bernoulli distribution, ***s***:

$$\sigma = \sqrt{Var(\hat{d})} = \sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}$$

and we get the Satterthwaite approximation for pooled standard error. If we calculate the pooled probability and use the pooled probability to calculate the standard deviation for both groups, we get:

$$\sigma = \sqrt{Var(\hat{d})} = \sqrt{\frac{s_p^2}{n_A} + \frac{s_p^2}{n_B}} = \sqrt{s_p^2\left(\frac{1}{n_A} + \frac{1}{n_B}\right)} = \sqrt{\hat{p}_p(1-\hat{p}_p)\left(\frac{1}{n_A} + \frac{1}{n_B}\right)}$$

where:

$$\hat{p}_p = \frac{p_A N_A + p_B N_B}{N_A + N_B}$$

This is the same equation used in the Udacity course. Both equations for pooled standard error will give you very similar results.

With that, we now have enough information to construct the distributions for the null hypothesis and the alternative hypothesis.

Compare the Null Hypothesis vs. the Alternative Hypothesis

Let's start off by defining the null hypothesis and the alternative hypothesis.

- The null hypothesis is the position that the change in the design made for the test group **would result in no change** in the conversion rate.
- The alternative hypothesis is the opposing position that the change in the design for the test group **would result in an improvement (or reduction)** in the conversion rate.

According to the Udacity course, the null hypothesis will be a normal distribution with a mean of zero and a standard deviation equal to the pooled standard error.

$$H_0 : d = 0$$

The null hypothesis

$$\hat{d}_0 \sim Normal(0,\ SE_{pool})$$

The alternative hypothesis has the same standard deviation as the null hypothesis, but the mean will be located at the difference in the conversion rate, `d_hat`. This makes sense because we can calculate the difference in the conversion rates directly from the data, but the normal distribution represents the possible values our experiment could have given us.

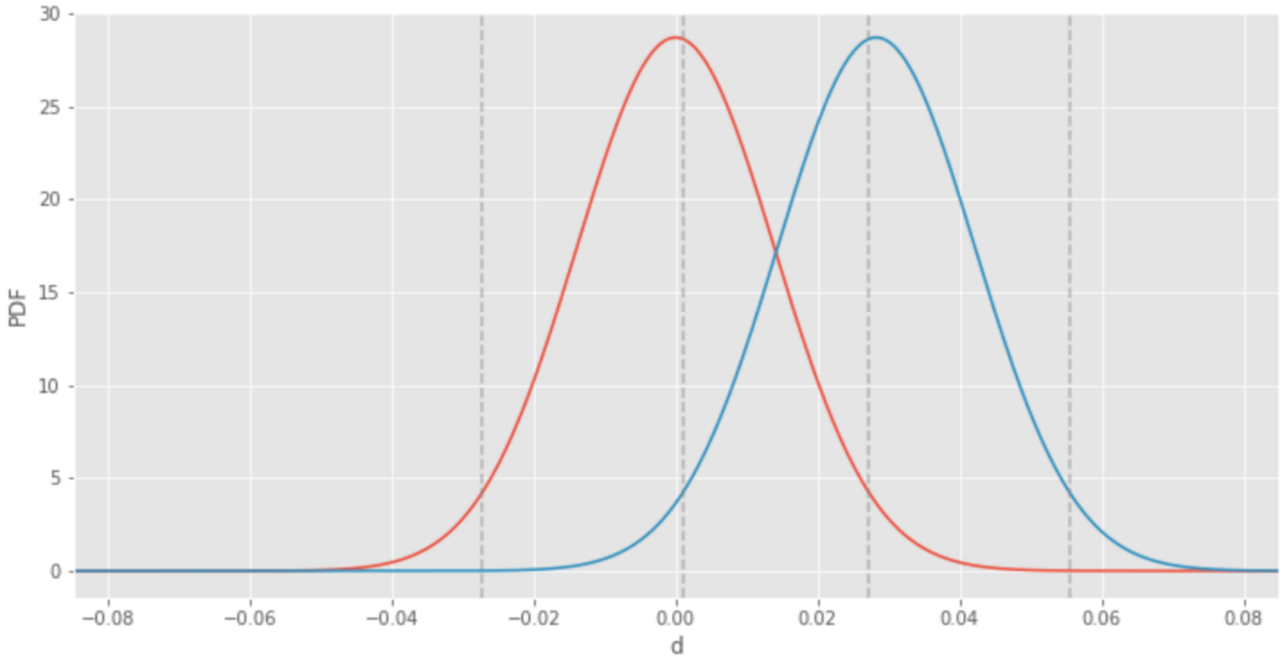
$$H_A : d = p_B - p_A$$

The alternative hypothesis

$$\hat{d}_A \sim Normal(d,\ SE_{pool})$$

Now that we understand the derivation of the pooled standard error, we can just directly plot the null and alternative hypotheses for future experiments. I wrote a script for quickly plotting the null and alternative hypotheses, `abplot`, which can be found here.

```
# define the parameters for abplot()
# use the actual values from the experiment for bcr and d_hat
# p_A is the conversion rate of the control group
# p_B is the conversion rate of the test group

n = N_A + N_B
bcr = p_A
d_hat = p_B - p_A
abplot(n, bcr, d_hat)
```



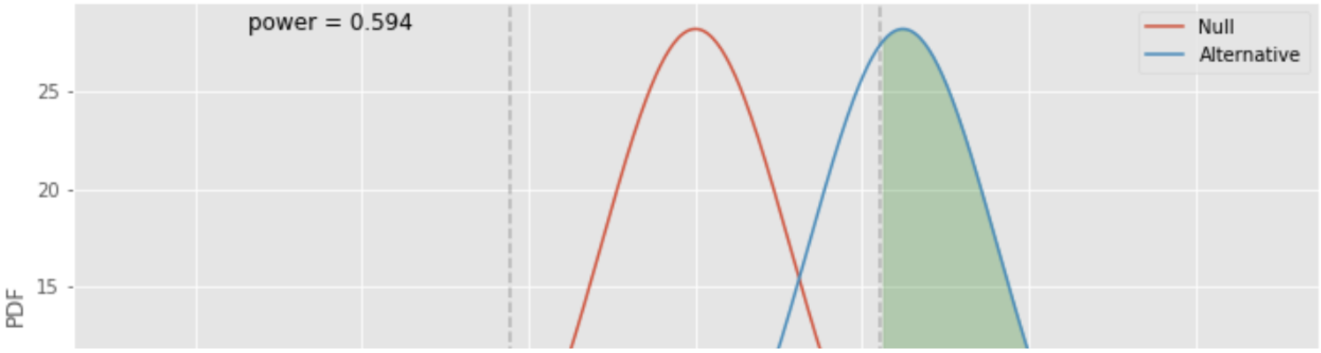
Null hypothesis (red) vs. alternative hypothesis (blue)

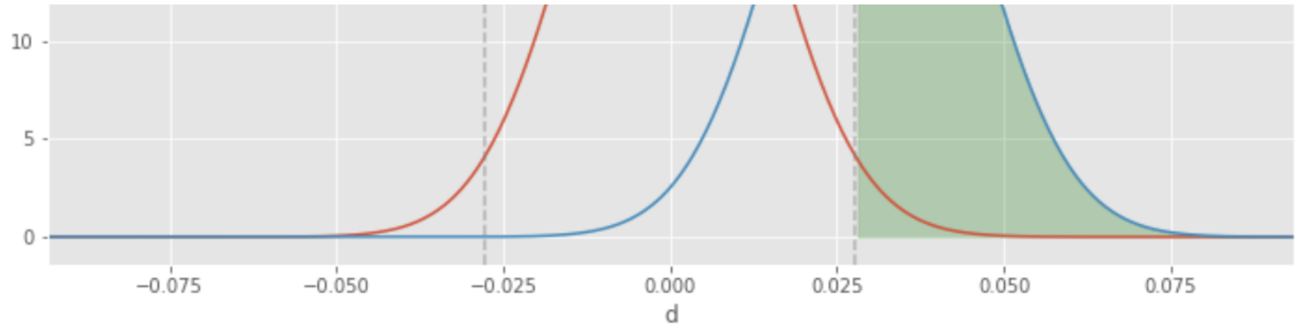
Visually, the plot for the null and alternative hypothesis looks very similar to the other plots above. Fortunately, both curves are identical in shape, so we can just compare the distance between the means of the two distributions. We can see that the alternative hypothesis curve suggests that the test group has a higher conversion rate than the control group. This plot also can be used to directly determine the statistical power.

4. Statistical Power and Significance Level

I think it is easier to define statistical power and significance level by first showing how they are represented in the plot of the null and alternative hypothesis. We can return a visualization of the statistical power by adding the parameter `show_power=True` .

```
abplot(N_A, N_B, bcr, d_hat, show_power=True)
```





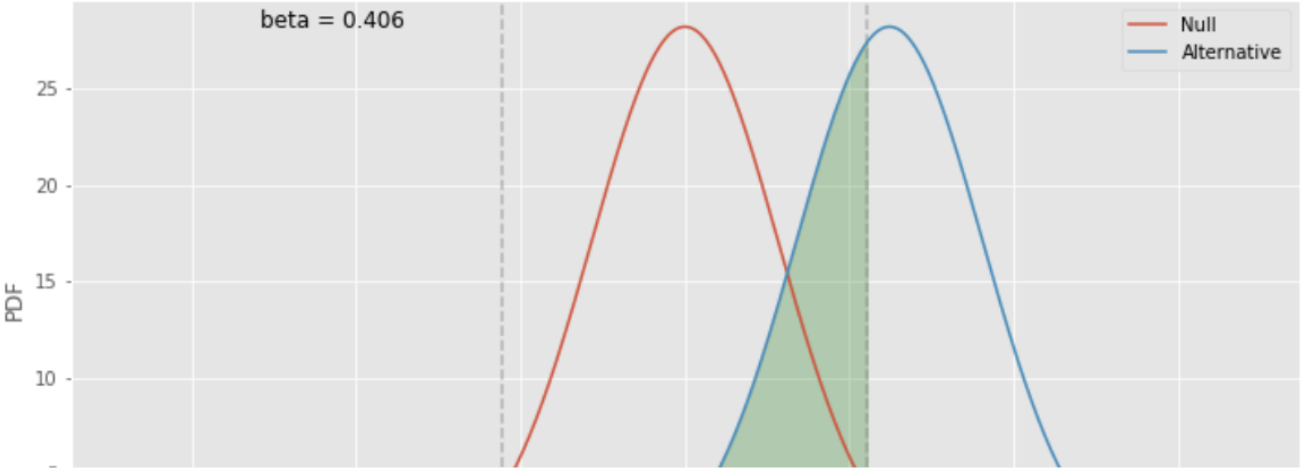
Statistical power shown in green

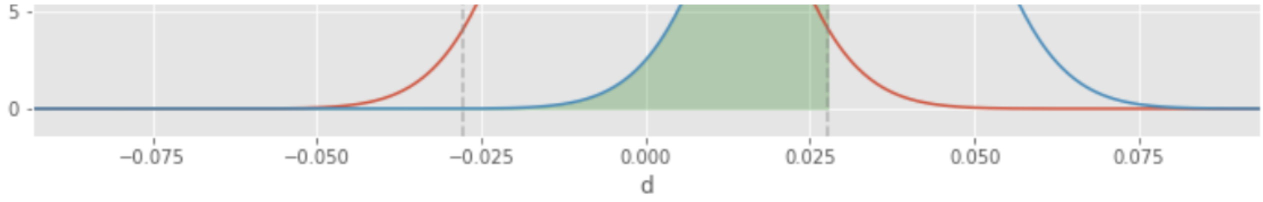
The green shaded area represents the statistical power, and the calculated value for power is also displayed on the plot. The gray dashed lines in the plot above represent the confidence interval (95% for the plot above) for the null hypothesis. Statistical power is calculated by finding the area under the alternative hypothesis distribution and outside of the confidence interval of the null hypothesis.

After running our experiment, we get a resulting conversion rate for both groups. If we calculate the difference between the conversion rates, we end up with one result, the difference or the effect of the design change. Our task is to determine which population this result came from, the null hypothesis or the alternative hypothesis.

The area under the alternative hypothesis curve is equal to 1. If the alternative design is truly better, the power is the probability that we accept the alternative hypothesis and reject the null hypothesis and is equal to the area shaded green **(true positive)**. The opposite area under the alternative curve is the probability that we accept the null hypothesis and reject the alternative hypothesis **(false negative)**. This is referred to as **beta** in A/B testing or hypothesis testing and is shown below.

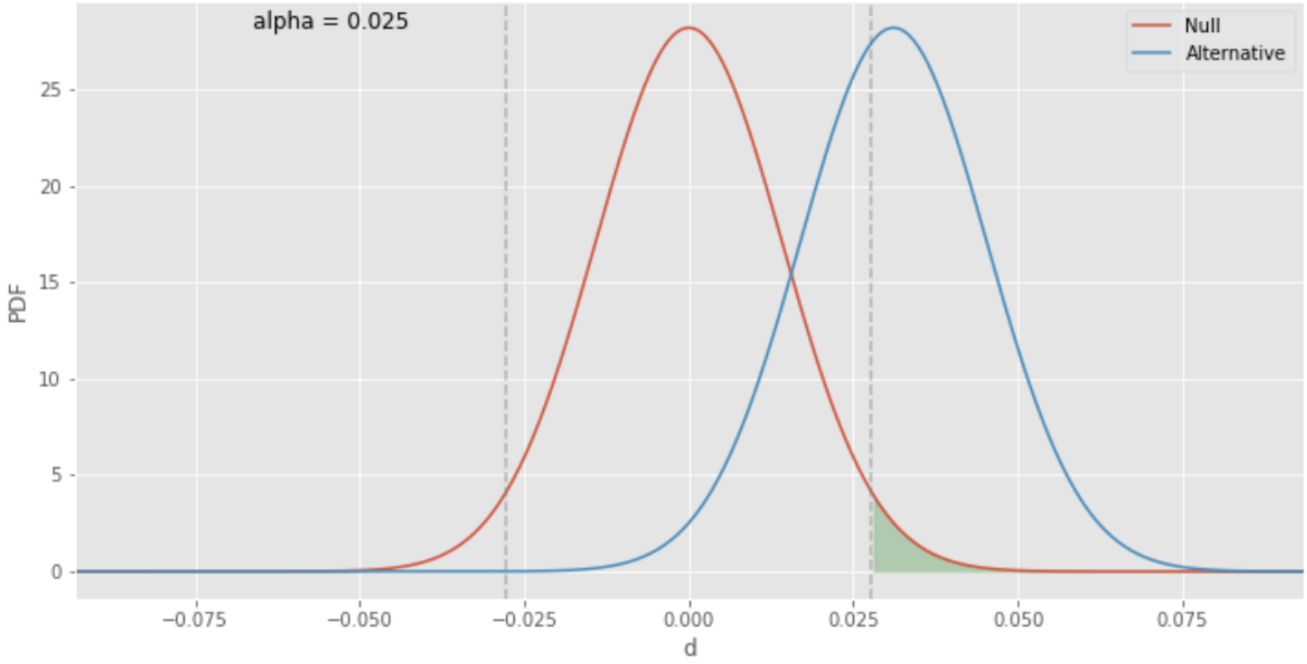
```
abplot(N_A, N_B, bcr, d_hat, show_beta=True)
```





Beta shown in green

The gray dashed line that divides the area under the alternative curve into two also directly segments the area associated with the significance level, often denoted with the greek letter alpha.



The green shaded area has an area equal to 0.025, which represents alpha.

If the null hypothesis is true and there **truly is no difference** between the control and test groups, then the significance level is the probability that we would reject the null hypothesis and accept the alternative hypothesis **(false positive)**. A false positive is when we mistakenly conclude that the new design is better. This value is low because we want to limit this probability.

Oftentimes, a problem will be given with a desired confidence level instead of the significance level. A typical 95% confidence level for an A/B test corresponds to a significance level of 0.05.

$$\alpha = 100\% - \text{Confidence Level}$$

Significance Level (alpha) and Confidence Level

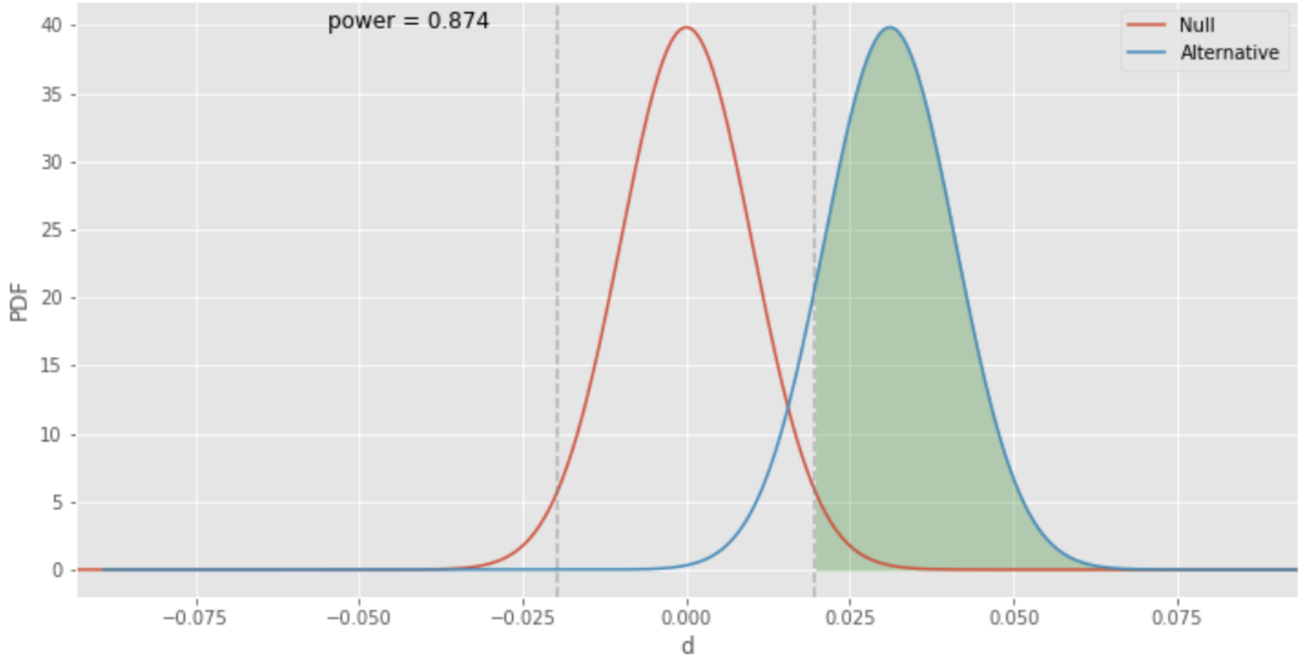
It might be helpful to refer to a confusion matrix when you are evaluating the results of an A/B test and the different probability of outcomes.

Experiments are typically set up for a minimum desired power of 80%. If our new design is truly better, we want our experiment to show that there is at least an 80% probability that this is the case. Unfortunately, our current experiment only has a power of 0.594. We know that if we increase the sample size for each group, we will decrease the pooled variance for our null and alternative hypothesis. This will make our distributions much narrower and may increase the statistical power. Let's take a look at how sample size will directly affect our results.

5. Sample Size

If we run our test again with a sample size of 2000 instead of 1000 for each group, we get the following results.

```
abplot(2000, 2000, bcr, d_hat, show_power=True)
```



Our curves for the null and alternative hypothesis have become more narrow and more of the area under the alternative curve is located on the right of the gray dashed line. The result for power is greater than 0.80 and meets our benchmark for statistical power. We can now say that our results are statistically significant.

A problem you may encounter is determining the minimum sample size you will need for your experiment. This is a common interview question and it is useful to know because it is directly related to how quickly you can complete your experiments and deliver statistically significant results to your design team. You can use the calculators available on the web, such as the ones below:

Sample Size Calculator (Evan's Awesome A/B Tools)

Visual, interactive sample size calculator ideal for A/B tests.

www.evanmiller.org

A/B Test Sample Size Calculator

With this methodology, you no longer need to use the sample size calculator to ensure the validity of your results...

www.optimizely.com

You will need the baseline conversion rate (bcr) and the minimum detectable effect, which is the minimum difference between the control and test group that you or your team will determine to be worth the investment of making the design change in the first place.

I wanted to write a script that would do the same calculation but needed to find the equation that was being used. After much searching, I found and tested this equation from this Stanford Lecture. **(Warning: link opens Powerpoint download.)**

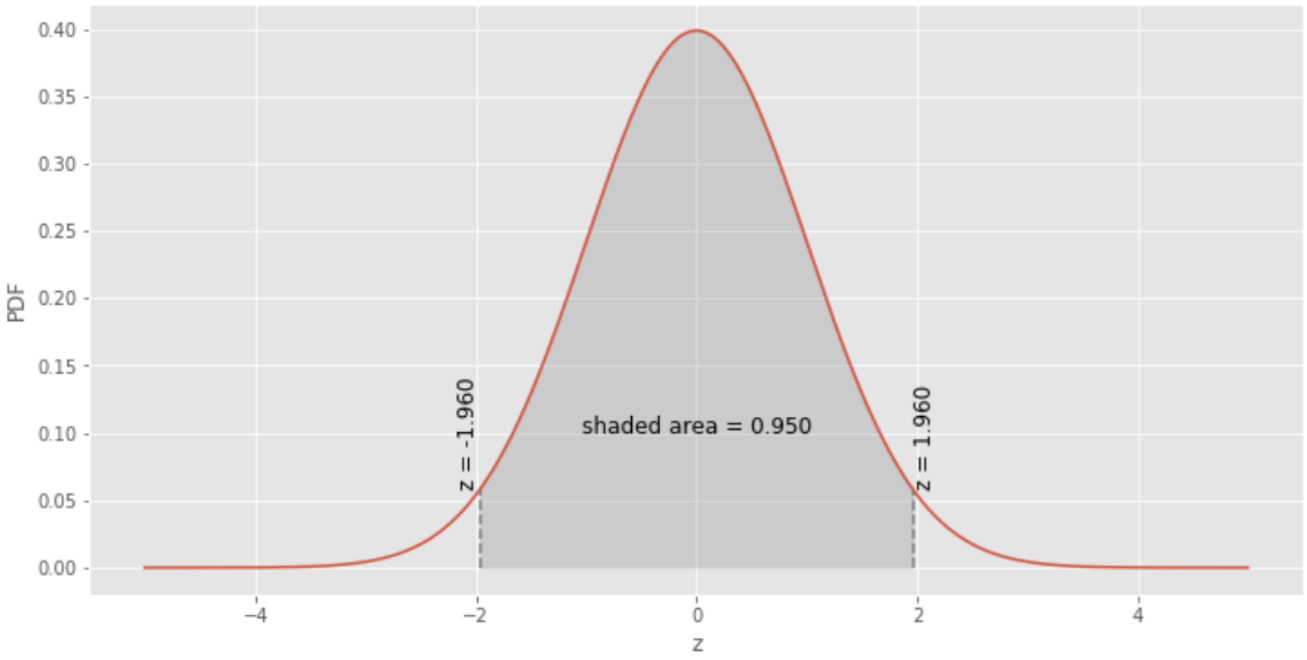
$$n = \frac{2(\bar{p})(1-\bar{p})(Z_\beta + Z_{\alpha/2})^2}{(p_B - p_A)^2}$$

Equation for minimum sample size

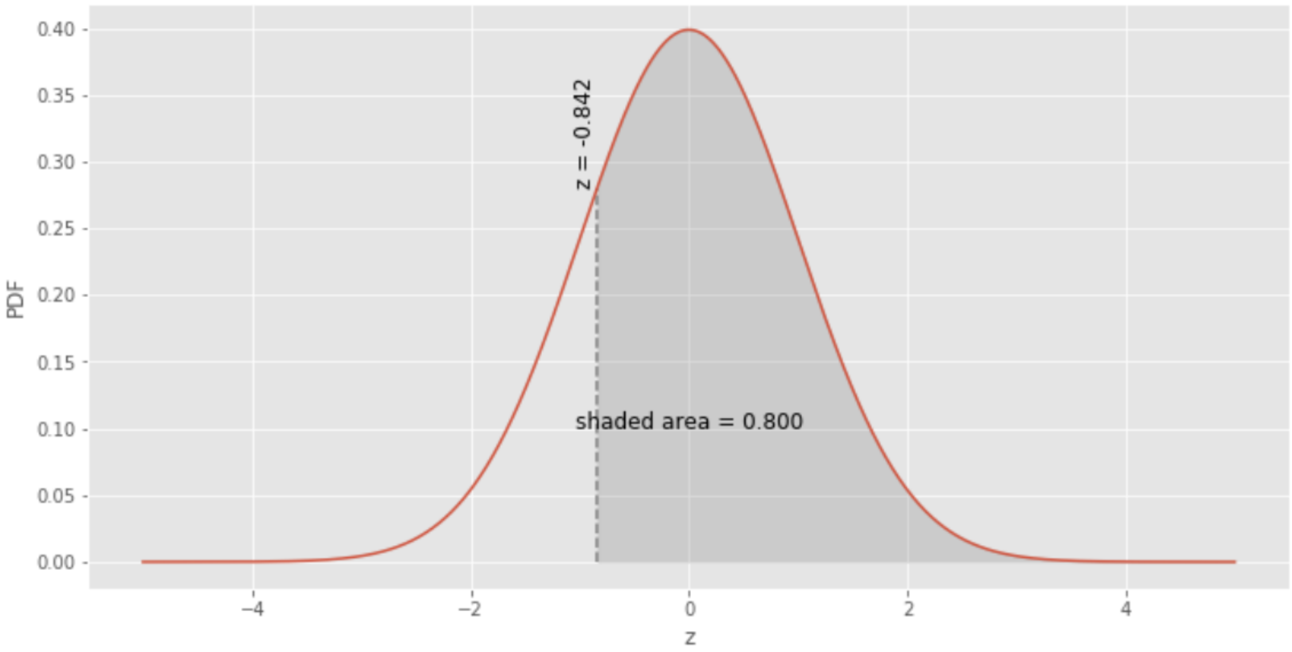
Z_β : z-score that corresponds to the level of statistical power
$Z_{\alpha/2}$: z-score that corresponds to the level of significance or confidence level

$\bar{p}$: pooled probability or average of p_A and p_B
p_A : success rate of control group
p_B : success rate of test group

Many people calculate Z from tables such as those shown here and here. However, I am more of a visual learner and I like to refer to the plot of the Z-distribution from which the values are derived.



Plot for typical significance level of 0.05 or confidence level of 0.95 (z = 1.96)



Typical z-score for power level of 0.80 (z = 0.842

The code for these z-plots can be found in my Github repo here.

Here is the Python code that performs the same calculation for minimum sample size:

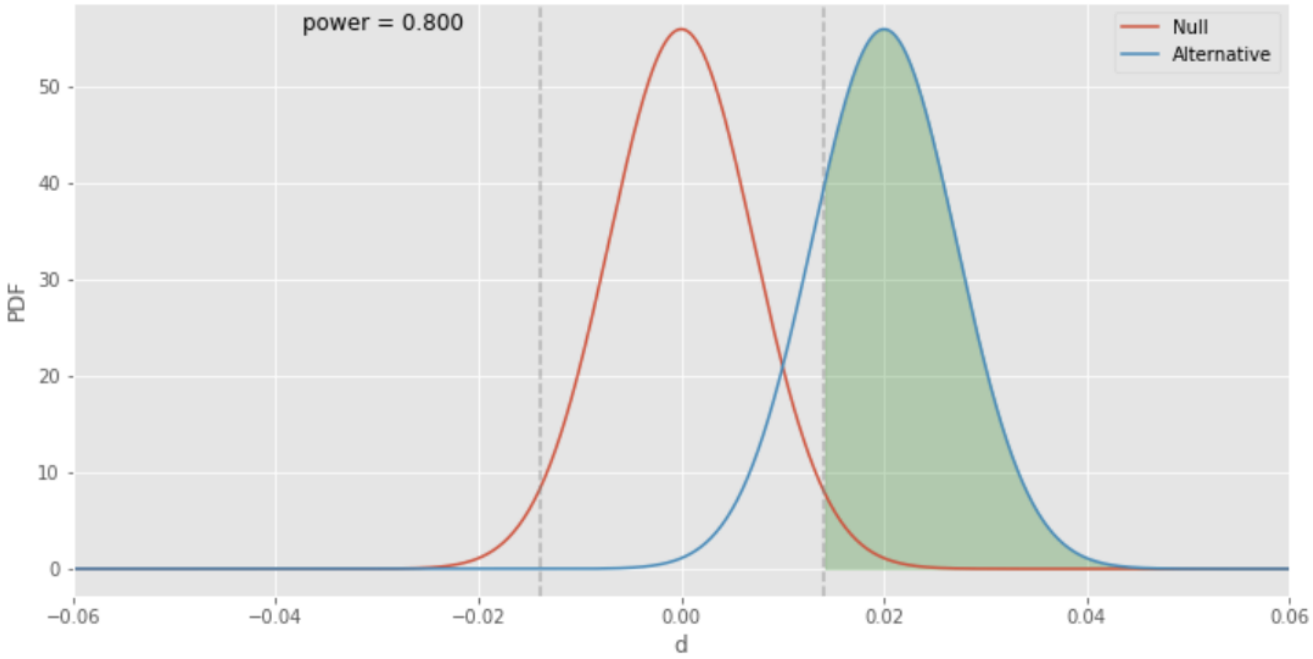
```
import scipy.stats as scs
```

```python

def min_sample_size(bcr, mde, power=0.8, sig_level=0.05):
    """Returns the minimum sample size to set up a split test

    Arguments:
        bcr (float): probability of success for control, sometimes
        referred to as baseline conversion rate

        mde (float): minimum change in measurement between control
        group and test group if alternative hypothesis is true, sometimes
        referred to as minimum detectable effect

        power (float): probability of rejecting the null hypothesis when the
        null hypothesis is false, typically 0.8

        sig_level (float): significance level often denoted as alpha,
        typically 0.05

    Returns:
        min_N: minimum sample size (float)

    References:
        Stanford lecture on sample sizes
        http://statweb.stanford.edu/~susan/courses/s141/hopower.pdf
    """
    # standard normal distribution to determine z-values
    standard_norm = scs.norm(0, 1)

    # find Z_beta from desired power
    Z_beta = standard_norm.ppf(power)

    # find Z_alpha
    Z_alpha = standard_norm.ppf(1-sig_level/2)

    # average of probabilities from both groups
    pooled_prob = (bcr + bcr+mde) / 2

    min_N = (2 * pooled_prob * (1 - pooled_prob) * (Z_beta + Z_alpha)**2
             / mde**2)

    return min_N
```

min_sample_size.py hosted with ❤ by GitHub view raw

I can demonstrate that this equation returns a correct answer by running another A/B experiment with the sample size that results from the equation.

```
min_sample_size(bcr=0.10, mde=0.02)
Out: 3842.026

abplot(3843, 3843, 0.10, 0.02, show_power=True)
```



The calculated power for this sample size was approximately 0.80. Therefore, if our design change had an improvement in conversion of about 2 percent, we would need at least 3843 samples in each group for a statistical power of at least 0.80.

That was a very long but basic walkthrough of A/B tests. Once you have developed an understanding and familiarity with the procedure, you will probably be able to run an experiment and go directly to the plots for the null and alternative hypothesis to determine if your results achieved enough power. By calculating the minimum sample size you need prior to the experiment, you can determine how long it will take to get the results back to your team for a final decision.

. . .

If you have any questions, I can try to answer them in the comments below. If you liked this article please 👏 . Shoutout to Brian McGarry for editing notes. Thank you for reading!

Ab Testing Data Analysis Data Visualization Data Science Product Development

About Help Legal