

Analyze A/B Test Results

This project will assure you have mastered the subjects covered in the statistics lessons. The hope is to have this project be as comprehensive of these topics as possible. Good luck!

Table of Contents

- [Introduction](#)
- [Part I - Probability](#)
- [Part II - A/B Test](#)
- [Part III - Regression](#)

Introduction

A/B tests are very commonly performed by data analysts and data scientists. It is important that you get some practice working with the difficulties of these

For this project, you will be working to understand the results of an A/B test run by an e-commerce website. Your goal is to work through this notebook to help the company understand if they should implement the new page, keep the old page, or perhaps run the experiment longer to make their decision.

As you work through this notebook, follow along in the classroom and answer the corresponding quiz questions associated with each question. The labels for each classroom concept are provided for each question. This will assure you are on the right track as you work through the project, and you can feel more confident in your final submission meeting the criteria. As a final check, assure you meet all the criteria on the [RUBRIC \(https://review.udacity.com/#!/projects/37e27304-ad47-4eb0-a1ab-8c12f60e43d0/rubric\)](https://review.udacity.com/#!/projects/37e27304-ad47-4eb0-a1ab-8c12f60e43d0/rubric).

Part I - Probability

To get started, let's import our libraries.

In [1]:

```
import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt
%matplotlib inline
#We are setting the seed to assure you get the same answers on quizzes as we set up
random.seed(42)
```

1. Now, read in the `ab_data.csv` data. Store it in `df`. **Use your dataframe to answer the questions in Quiz 1 of the classroom.**

a. Read in the dataset and take a look at the top few rows here:

In [3]:

```
df = pd.read_csv('ab_data.csv')
df.head()
```

Out[3]:

	user_id	timestamp	group	landing_page	converted
0	851104	2017-01-21 22:11:48.556739	control	old_page	0
1	804228	2017-01-12 08:01:45.159739	control	old_page	0
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0
4	864975	2017-01-21 01:52:26.210827	control	old_page	1

b. Use the below cell to find the number of rows in the dataset.

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294478 entries, 0 to 294477
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   user_id          294478 non-null  int64
1   timestamp        294478 non-null  object
2   group            294478 non-null  object
3   landing_page     294478 non-null  object
4   converted        294478 non-null  int64
dtypes: int64(2), object(3)
memory usage: 11.2+ MB
```

c. The number of unique users in the dataset.

In [9]:

```
df['user_id'].nunique()
```

Out[9]:

290584

d. The proportion of users converted.

In [13]:

```
df['converted'].value_counts()
print(35237/294477*100)
```

11.965959990084116

e. The number of times the new_page and treatment don't line up.

In [37]:

```
#count control in group if new_page in landing_page column. count treatment in group if old_page in landing_page column
#sum them
count_to = df[(df.group == 'treatment') & (df.landing_page == 'old_page')].count()
count_cn = df[(df.group == 'control') & (df.landing_page == 'new_page')].count()
dont_line_up = count_to['group'] + count_cn['group']

dont_line_up
```

Out[37]:

3893

f. Do any of the rows have missing values?

In [16]:

```
df.isna().sum()
```

Out[16]:

```
user_id      0
timestamp    0
group         0
landing_page  0
converted     0
dtype: int64
```

2. For the rows where **treatment** is not aligned with **new_page** or **control** is not aligned with **old_page**, we cannot be sure if this row truly received the new or old page. Use **Quiz 2** in the classroom to provide how we should handle these rows.

a. Now use the answer to the quiz to create a new dataset that meets the specifications from the quiz. Store your new dataframe in **df2**.

In [41]:

```
df2 = df
to = df2[(df2.group == 'treatment') & (df2.landing_page == 'old_page')].index
cn = df2[(df2.group == 'control') & (df2.landing_page == 'new_page')].index
df2.drop(to, inplace=True)
df2.drop(cn, inplace=True)
```

In [42]:

```
# Double Check all of the correct rows were removed - this should be 0
df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) == False].shape[0]
```

Out[42]:

0

3. Use **df2** and the cells below to answer questions for **Quiz3** in the classroom.

a. How many unique **user_ids** are in **df2**?

In [43]:

```
df2['user_id'].nunique()
```

Out[43]:

290584

b. There is one **user_id** repeated in **df2**. What is it?

In [46]:

```
df2.user_id.value_counts().loc[lambda x: x>1].index
```

Out[46]:

```
Int64Index([773192], dtype='int64')
```

c. What is the row information for the repeat **user_id**?

In [47]:

```
df.loc[df['user_id'] == 773192]
```

Out[47]:

	user_id	timestamp	group	landing_page	converted
1899	773192	2017-01-09 05:37:58.781806	treatment	new_page	0
2893	773192	2017-01-14 02:55:59.590927	treatment	new_page	0

d. Remove **one** of the rows with a duplicate **user_id**, but keep your dataframe as **df2**.

In [50]:

```
#code line for drop: df2.drop([1899], axis=0, inplace=True)
df2.loc[df['user_id'] == 773192] #check if dropped
```

Out[50]:

	user_id	timestamp	group	landing_page	converted
2893	773192	2017-01-14 02:55:59.590927	treatment	new_page	0

4. Use **df2** in the below cells to answer the quiz questions related to **Quiz 4** in the classroom.

a. What is the probability of an individual converting regardless of the page they receive?

In [53]:

```
#Divide number of converted with all users
df2.query('converted == 1').user_id.nunique() / df.user_id.nunique()
```

Out[53]:

```
0.11959708724499628
```

b. Given that an individual was in the `control` group, what is the probability they converted?

In [67]:

```
#Divide number of cases where control group converted with all users that are in group control
#df2.query('converted == 1 & group == "control").user_id.nunique() / df.query('group == "control").user_id.nunique()
#Above is correct but following is a bit shorter. Doing the same thing. Keeping for note to self
df2.query('group == "control")['converted'].mean()
```

Out[67]:

```
0.1203863045004612
```

c. Given that an individual was in the `treatment` group, what is the probability they converted?

In [68]:

```
#Divide number of cases where treatment group converted with all users that are in group treatment
#df2.query('converted == 1 & group == "treatment").user_id.nunique() / df.query('group == "treatment").user_id.nunique()
#Above is correct but following is a bit shorter. Doing the same thing. Keeping for note to self
df2.query('group == "treatment")['converted'].mean()
```

Out[68]:

```
0.11880806551510564
```

d. What is the probability that an individual received the new page?

In [70]:

```
#Divide number of cases where landing_page is new_page with all users. Cannot use mean directly here as when user id is not bo
df2.query('landing_page == "new_page").user_id.nunique() / df.user_id.nunique()
```

Out[70]:

```
0.5000619442226688
```

e. Consider your results from a. through d. above, and explain below whether you think there is sufficient evidence to say that the new treatment page leads to more conversions.

The conversation rate for control group is about 12.0% and for the treatment group 11.9%. This does not give enough evidence to state that the new page performs better, rather it performs equal to the old page. Portion of visitors to each page was fifty fifty so it was a fair judgement. It would require to drill down on user details to perhaps learn more how the new page vs the old page perform. But based on this result there is not enough evidence to change, yet if there is a wish to change, because you want to, there is nothing from this test showing it would result in drop of conversion. Maybe flip a coin if you have a hard time to chose to stay or go ;)

Part II - A/B Test

Notice that because of the time stamp associated with each event, you could technically run a hypothesis test continuously as each observation was observed.

However, then the hard question is do you stop as soon as one page is considered significantly better than another or does it need to happen consistently for a certain amount of time? How long do you run to render a decision that neither page is better than another?

These questions are the difficult parts associated with A/B tests in general.

1. For now, consider you need to make the decision just based on all the data provided. If you want to assume that the old page is better unless the new page proves to be definitely better at a Type I error rate of 5%, what should your null and alternative hypotheses be? You can state your hypothesis in terms of words or in terms of p_{old} and p_{new} , which are the converted rates for the old and new pages.

$H_0 = P_{old}$ is better if P_{new} type one error is less or equal to 0.05. P_{old} is larger than 0.05. $H_1 = P_{new}$ is better if P_{old} type one error is larger than 0.05 p_{new} is less or equal to 0.05 Type one error is when chosen site should convert but it does not. Type two error is when chosen site should not convert but it did $H_0 = P_{old} \leq \alpha 0.05$ $H_1 = P_{new} > \alpha 0.05$

2. Assume under the null hypothesis, p_{new} and p_{old} both have "true" success rates equal to the **converted** success rate regardless of page - that is p_{new} and p_{old} are equal. Furthermore, assume they are equal to the **converted** rate in **ab_data.csv** regardless of the page.

Use a sample size for each page equal to the ones in **ab_data.csv**.

Perform the sampling distribution for the difference in **converted** between the two pages over 10,000 iterations of calculating an estimate from the null.

Use the cells below to provide the necessary parts of this simulation. If this doesn't make complete sense right now, don't worry - you are going to work through the problems below to complete this problem. You can use **Quiz 5** in the classroom to make sure you are on the right track.

a. What is the **convert rate** for p_{new} under the null?

In [85]:

```
#because they are equal I check the conversion rate on all
df2['converted'].mean()
```

Out[85]:

```
0.11959708724499628
```

b. What is the **convert rate** for p_{old} under the null?

In [86]:

```
#same as p.new
df2['converted'].mean()
```

Out[86]:

```
0.11959708724499628
```

c. What is n_{new} ?

In [83]:

```
df2.query('group == "treatment").count()
```

Out[83]:

```
user_id      145310
timestamp    145310
group         145310
landing_page  145310
converted     145310
dtype: int64
```

d. What is n_{old} ?

In [84]:

```
df2.query('group == "control").count()
```

Out[84]:

```
user_id      145274
timestamp    145274
group         145274
landing_page  145274
converted     145274
dtype: int64
```

e. Simulate n_{new} transactions with a convert rate of p_{new} under the null. Store these n_{new} 1's and 0's in **new_page_converted**.

In [91]:

```
#Method found here: https://stackoverflow.com/questions/47300430/how-to-generate-random-numbers-with-specific-probabilities-in-
new_page_converted = []

for i in range(145310):
    #creates one number out of 0 or 1 with prob p 0.8804 for 0 and 0.1196 for 1
    convert_rate = np.random.choice(np.arange(0, 2), p=[0.8804, 0.1196])
    new_page_converted.append(convert_rate)
```

f. Simulate n_{old} transactions with a convert rate of p_{old} under the null. Store these n_{old} 1's and 0's in **old_page_converted**.

In [92]:

```
#Same as for new but another storage name and slightly less number of transactions
old_page_converted = []

for i in range(145274):
    #creates one number out of 0 or 1 with prob p 0.8804 for 0 and 0.1196 for 1
    convert_rate = np.random.choice(np.arange(0, 2), p=[0.8804, 0.1196])
    old_page_converted.append(convert_rate)
```

g. Find $p_{new} - p_{old}$ for your simulated values from part (e) and (f).

In [103]:

```
#rounded to fewer decimals, because found out too many decimals make the subtract calculation go bananas
new_mean = round(np.array(new_page_converted).mean(), 3)
old_mean = round(np.array(old_page_converted).mean(), 3)

new_subtract_old = new_mean - old_mean

print(new_mean, ' - ', old_mean, ' = ', new_subtract_old)
```

```
0.12 - 0.12 = 0.0
```

h. Simulate 10,000 $p_{new} - p_{old}$ values using this same process similarly to the one you calculated in parts **a. through g.** above. Store all 10,000 values in a numpy array called **p_diffs**.

In [106]:

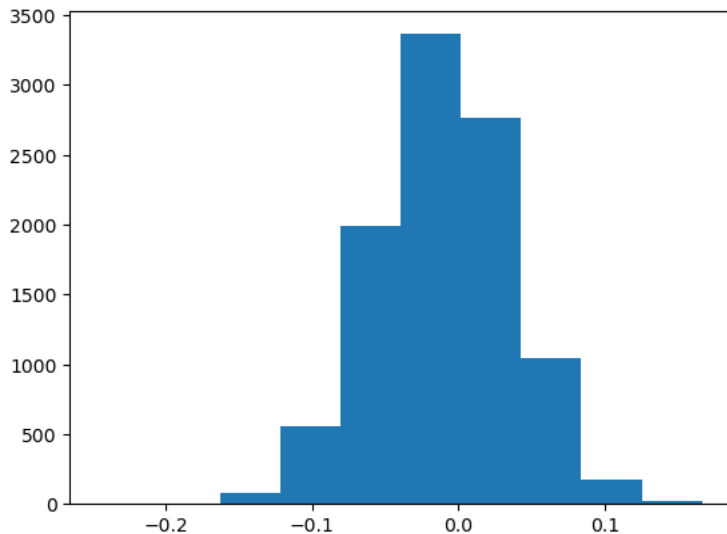
```
sample_df2 = df2.sample(200)
old_means, new_means, p_diffs = [], [], []

for _ in range(10000):
    bootsamp = sample_df2.sample(200, replace = True)
    new_mean = bootsamp[bootsamp['landing_page'] == 'new_page']['converted'].mean()
    old_mean = bootsamp[bootsamp['landing_page'] == 'old_page']['converted'].mean()
    # append the info
    new_means.append(new_mean)
    old_means.append(old_mean)
    p_diffs.append(new_mean - old_mean)
```

i. Plot a histogram of the **p_diffs**. Does this plot look like what you expected? Use the matching problem in the classroom to assure you fully understand what was computed here.

In [109]:

```
p_diffs = np.array(p_diffs)
plt.hist(p_diffs);
```



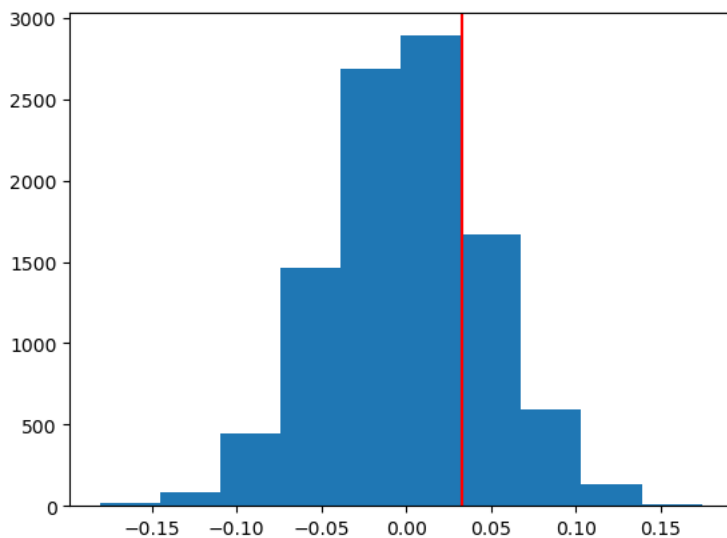
j. What proportion of the **p_diffs** are greater than the actual difference observed in **ab_data.csv**?

In [114]:

```
null_vals = np.random.normal(0, p_diffs.std(), p_diffs.size)
# plot null distribution
plt.hist(null_vals);

# Compute observed difference in completion rates
obs_diff = new_mean - old_mean

# plot line for observed statistic
plt.axvline(obs_diff, c='red');
```



k. In words, explain what you just computed in part j. What is this value called in scientific studies? What does this value mean in terms of whether or not there is a difference between the new and old pages?

As described in the Hypothesis Testing part of the course: The p-value is the probability of getting our statistic or a more extreme value if the null is true. Therefore, small p-values suggest our null is not true. Rather, our statistic is likely to have come from a different distribution than the null. When the p-value is large, we have evidence that our statistic was likely to come from the null hypothesis. Therefore, we do not have evidence to reject the null. By comparing our p-value to our type I error threshold (α), we can make our decision about which hypothesis we will choose.

l. We could also use a built-in to achieve similar results. Though using the built-in might be easier to code, the above portions are a walkthrough of the ideas that are critical to correctly thinking about statistical significance. Fill in the below to calculate the number of conversions for each page, as well as the number of individuals who received each page. Let `n_old` and `n_new` refer to the number of rows associated with the old page and new pages, respectively.

In [148]:

```
import statsmodels.api as sm
from statsmodels.stats.proportion import proportions_ztest

convert_old = df2.query('landing_page == "old_page")['converted'].sum()
convert_new = df2.query('landing_page == "new_page")['converted'].sum()
n_old = 145274 #from previous calculations
n_new = 145310
```

m. Now use `stats.proportions_ztest` to compute your test statistic and p-value. [\[Here\]](#). (http://knowledgestack.com/python/statsmodels/proportions_ztest/) is a helpful link on using the built in.

In [149]:

```
ig this input page instead of yours (which led to a 404)
s://www.statsmodels.org/dev/generated/statsmodels.stats.proportion.proportions_ztest.html#statsmodels.stats.proportion.proportions_ztest

re, p_value = sm.stats.proportions_ztest([convert_old,convert_new],[n_old,n_new], alternative='smaller')

(z_score, p_value)

1.3109241984234394 0.9050583127590245
```

n. What do the z-score and p-value you computed in the previous question mean for the conversion rates of the old and new pages? Do they agree with the findings in parts **j** and **k**?

Answer found here: <https://statisticsbyjim.com/basics/z-score/> (<https://statisticsbyjim.com/basics/z-score/>). "A z-score measures the distance between a data point and the mean using standard deviations. Z-scores can be positive or negative. The sign tells you whether the observation is above or below the mean."

Part III - A regression approach

1. In this final part, you will see that the result you achieved in the previous A/B test can also be achieved by performing regression.

a. Since each row is either a conversion or no conversion, what type of regression should you be performing in this case?

Simple linear regression, because we will compare two quantitative variables to one another.

b. The goal is to use **statsmodels** to fit the regression model you specified in part **a**. to see if there is a significant difference in conversion based on which page a customer receives. However, you first need to create a column for the intercept, and create a dummy variable column for which page each user received. Add an **intercept** column, as well as an **ab_page** column, which is 1 when an individual receives the **treatment** and 0 if **control**.

In [191]:

```
#intercept add
df2['intercept'] = 1

#ab_page column, ref: https://stackoverflow.com/questions/68836813/adding-column-in-a-dataframe-with-0-1-values-based-on-another-column
df2_newish = df2
landing_pages = ['new_page']
df2_newish['ab_page'] = df2['landing_page'].isin(landing_pages).astype(int)
```

In [192]:

```
df2_newish.head()
```

Out[192]:

	user_id	timestamp	group	landing_page	converted	intercept	ab_page
0	851104	2017-01-21 22:11:48.556739	control	old_page	0	1	0
1	804228	2017-01-12 08:01:45.159739	control	old_page	0	1	0
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0	1	1
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0	1	1
4	864975	2017-01-21 01:52:26.210827	control	old_page	1	1	0

c. Use **statsmodels** to import your regression model. Instantiate the model, and fit the model using the two columns you created in part **b**. to predict whether or not an individual converts.

In [195]:

```
lm = sm.OLS(df2_newish['converted'], df2_newish[['intercept', 'ab_page']])
results = lm.fit()
```

d. Provide the summary of your model below, and use it as necessary to answer the following questions.

In [196]:

```
results.summary()
```

Out[196]:

OLS Regression Results

Dep. Variable:	converted	R-squared:	0.000
Model:	OLS	Adj. R-squared:	0.000
Method:	Least Squares	F-statistic:	1.719
Date:	Thu, 05 Jan 2023	Prob (F-statistic):	0.190
Time:	19:50:17	Log-Likelihood:	-85267.
No. Observations:	290584	AIC:	1.705e+05
Df Residuals:	290582	BIC:	1.706e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
intercept	0.1204	0.001	141.407	0.000	0.119	0.122
ab_page	-0.0016	0.001	-1.311	0.190	-0.004	0.001

Omnibus:	125553.456	Durbin-Watson:	1.995
Prob(Omnibus):	0.000	Jarque-Bera (JB):	414313.355
Skew:	2.345	Prob(JB):	0.00
Kurtosis:	6.497	Cond. No.	2.62

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

e. What is the p-value associated with **ab_page**? Why does it differ from the value you found in **Part II**?

Hint: What are the null and alternative hypotheses associated with your regression model, and how do they compare to the null and alternative hypotheses in the **Part II**?

Intercept p value is 0.000 and for ab_page it is 0.190. In this case the intercept, the null, is zero and the alternative, the hypothesis, is different suggesting the alternative to be false. But as when the hypothesis is larger than 0.05 we fail to reject the null. More evidence needed to take a decision based on data.
External Ref: <https://www.bumc.bu.edu/public-health-learning-modules/lessons/introduction-and-learning-objectives/p-values/#:~:text=If%20the%20p%2Dvalue%20is,the%20means%20are%20probably%20different> (<https://www.bumc.bu.edu/public-health-learning-modules/lessons/introduction-and-learning-objectives/p-values/#:~:text=If%20the%20p%2Dvalue%20is,the%20means%20are%20probably%20different>).

f. Now, you are considering other things that might influence whether or not an individual converts. Discuss why it is a good idea to consider other factors to add into your regression model. Are there any disadvantages to adding additional terms into your regression model?

The result from both tests show quite equal spread even though the regression show to go for the new page. Adding other factors, such as time could be an option to look at without having to join in an extra csv.

g. Now along with testing if the conversion rate changes for different pages, also add an effect based on which country a user lives. You will need to read in the **countries.csv** dataset and merge together your datasets on the appropriate rows. [Here \(https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.join.html\)](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.join.html) are the docs for joining tables.

Does it appear that country had an impact on conversion? Don't forget to create dummy variables for these country columns - **Hint: You will need two columns for the three dummy variables**. Provide the statistical output as well as a written response to answer this question.

In [187]:

```
countries_df = pd.read_csv('./countries.csv')
df_new = countries_df.set_index('user_id').join(df2.set_index('user_id'), how='inner')
df_new.head()
```

Out[187]:

	country	timestamp	group	landing_page	converted	intercept	ab_page
user_id							
834778	UK	2017-01-14 23:08:43.304998	control	old_page	0	1	0
928468	US	2017-01-23 14:44:16.387854	treatment	new_page	0	1	1
822059	UK	2017-01-16 14:04:14.719771	treatment	new_page	1	1	1
711597	UK	2017-01-22 03:14:24.763511	control	old_page	0	1	0
710616	UK	2017-01-16 13:14:44.000513	treatment	new_page	0	1	1

In [188]:

```
### Create the necessary dummy variables
country_dummies = pd.get_dummies(df_new['country'])
df_new = df_new.join(country_dummies)
df_new.head()
```

Out[188]:

	country	timestamp	group	landing_page	converted	intercept	ab_page	CA	UK	US
user_id										
834778	UK	2017-01-14 23:08:43.304998	control	old_page	0	1	0	0	1	0
928468	US	2017-01-23 14:44:16.387854	treatment	new_page	0	1	1	0	0	1
822059	UK	2017-01-16 14:04:14.719771	treatment	new_page	1	1	1	0	1	0
711597	UK	2017-01-22 03:14:24.763511	control	old_page	0	1	0	0	1	0
710616	UK	2017-01-16 13:14:44.000513	treatment	new_page	0	1	1	0	1	0

h. Though you have now looked at the individual factors of country and page on conversion, we would now like to look at an interaction between page and country to see if there significant effects on conversion. Create the necessary additional columns, and fit the new model.

Provide the summary results, and your conclusions based on the results.

In [197]:

```
### Fit Your Linear Model And Obtain the Results
lm = sm.OLS(df_new['ab_page'], df_new[['intercept', 'UK', 'US']]) #Drop CA that will be baseline
results = lm.fit()
results.summary()
```

Out[197]:

OLS Regression Results

Dep. Variable:	ab_page	R-squared:	0.000
Model:	OLS	Adj. R-squared:	-0.000
Method:	Least Squares	F-statistic:	0.8946
Date:	Thu, 05 Jan 2023	Prob (F-statistic):	0.409
Time:	19:52:12	Log-Likelihood:	-2.1090e+05
No. Observations:	290584	AIC:	4.218e+05
Df Residuals:	290581	BIC:	4.218e+05
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
intercept	0.5036	0.004	121.267	0.000	0.495	0.512
UK	-0.0053	0.005	-1.166	0.244	-0.014	0.004
US	-0.0031	0.004	-0.720	0.472	-0.012	0.005

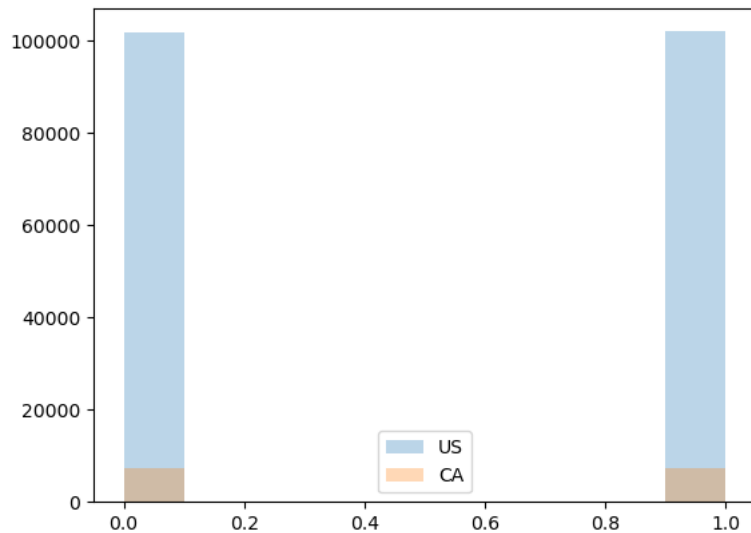
Omnibus:	989752.634	Durbin-Watson:	1.996
Prob(Omnibus):	0.000	Jarque-Bera (JB):	48429.474
Skew:	-0.000	Prob(JB):	0.00
Kurtosis:	1.000	Cond. No.	9.94

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [202]:

```
plt.hist(df_new.query("US == 1")['ab_page'], alpha = 0.3, label = 'US');
plt.hist(df_new.query("CA == 1")['ab_page'], alpha = 0.3, label = 'CA');
plt.legend();
```



Conclusion is that it all looks like something is wrong in my calculations *LOL :D or it actually shows that even though adding country as term it will not provide us with any significant insight to whereas old or new page performs better. The country data just show that more people visit the page from US than Canada. I would suggest do qualitative testing instead, perhaps focus groups with customers to find out what is expected to be on the page.

Conclusions

Congratulations on completing the project!

Gather Submission Materials

Once you are satisfied with the status of your Notebook, you should save it in a format that will make it easy for others to read. You can use the **File -> Download as -> HTML (.html)** menu to save your notebook as an .html file. If you are working locally and get an error about "No module name", then open a terminal and try installing the missing module using `pip install <module_name>` (don't include the "<" or ">" or any words following a period in the module name).

You will submit both your original Notebook and an HTML or PDF copy of the Notebook for review. There is no need for you to include any data files with your submission. If you made reference to other websites, books, and other resources to help you in solving tasks in the project, make sure that you document them. It is recommended that you either add a "Resources" section in a Markdown cell at the end of the Notebook report, or you can include a `readme.txt` file documenting your sources.

Submit the Project

When you're ready, click on the "Submit Project" button to go to the project submission page. You can submit your files as a .zip archive or you can link to a GitHub repository containing your project files. If you go with GitHub, note that your submission will be a snapshot of the linked repository at time of submission. It is recommended that you keep each project in a separate repository to avoid any potential confusion: if a reviewer gets multiple folders representing multiple projects, there might be confusion regarding what project is to be evaluated.

It can take us up to a week to grade the project, but in most cases it is much faster. You will get an email once your submission has been reviewed. If you are having any problems submitting your project or wish to check on the status of your submission, please email us at dataanalyst-project@udacity.com (<mailto:dataanalyst-project@udacity.com>). In the meantime, you should feel free to continue on with your learning journey by beginning the next module in the program.

In []: