# Analyze A/B Test Results

You may either submit your notebook through the workspace here, or you may work from your local machine and submit through the next page. Either way assure that your code passes the project RUBRIC (https://review.udacity.com/#!/projects/37e27304-ad47-4eb0-a1ab-8c12f60e43d0/rubric). **Please save regularly.**

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

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).

#### 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 s
et 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 [2]:
```
#Store and read data
df=pd.read_csv('ab_data.csv')
df.head()
```

Out[2]:

| | 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 cell below to find the number of rows in the dataset.

In [3]:
```
#Find the number of rows
df.shape[0]
```

Out[3]: 294478

c. The number of unique users in the dataset.

In [4]:
```
#Find the number of unique users
df['user_id'].nunique()
```

Out[4]: 290584

d. The proportion of users converted.

In [5]:
```
#Find the proportion of users converted
df['converted'].mean()
```

Out[5]: 0.11965919355605512

e. The number of times the `new_page` and `treatment` don't match.

In [6]:
```
#Create counts for control and treatment groups in a new dataframe
df_counts = df.query("(group == 'control' and landing_page == 'new_page') or
 (group == 'treatment' and landing_page == 'old_page')")
#Find total number of rows for this dataframe
df_counts.shape[0]
```

Out[6]:
```
3893
```

f. Do any of the rows have missing values?

In [7]:
```
#Check for rows with missing values
df.info()
```

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

There are no missing values as we have the same amount of rows in each column.

2. For the rows where **treatment** does not match with **new_page** or **control** does not match with **old_page**, we cannot be sure if this row truly received the new or old page. Use **Quiz 2** in the classroom to figure out 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 [8]:
```
#Remove rows with no match as needed
df2 = df.query("(group == 'control' and landing_page == 'old_page') or (group
 == 'treatment' and landing_page == 'new_page')")
```

In [9]:
```
# 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[9]:
```
0
```

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

a. How many unique **user_id**s are in **df2**?

In [10]:
```
#Find the number of unique users in new dataframe
df2['user_id'].nunique()
```

Out[10]: 290584

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

In [11]:
```
#Find the duplicated user id
df2[df2.duplicated('user_id')]
```

Out[11]:

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

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

In [12]:
```
#Obtain row information for the repeat user_id
df2[df2['user_id']==773192]
```

Out[12]:

| | 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 [13]:
```
#Remove one of the rows
df2 = df2.copy()
df2.drop(labels=1899, axis=0, inplace=True)
```

In [14]:
```
#Confirm changes
df2[df2['user_id']==773192]
```

Out[14]:

| | 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 cells below 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 [15]:
```
#Calculate the probability of conversions regardless of group
df2['converted'].mean()
```

Out[15]: 0.11959708724499628

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

In [16]:
```
#Find probability of control group conversions
df2[df2['group']=='control']['converted'].mean()
```

Out[16]: 0.1203863045004612

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

In [17]:
```
#Find probability of treatment group conversions
df2[df2['group']=='treatment']['converted'].mean()
```

Out[17]: 0.11880806551510564

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

In [18]:
```
#Calculate probability of individual receiving new page
new_page=(df2.landing_page == "new_page").mean()
new_page
```

Out[18]: 0.50006194422266881

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

**Based on my results from part 4a to 4d, there is not sufficient evidence to conclude that the new treatment page leads to more conversions.** >

> The conversions from the control group (12.04%) was greater than the conversions from the treatment group (11.88%), however it is only a small difference to call it significant.
>
> At this stage of analysis, there is not enough information to support that the new treatment would lead to more conversions as effects of change aversion, novelty effect, or experiment duration have not be accounted for. These possible effects can create misleading results from this analysis so it is important to consider them before making any final conclusions.

## 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.

**Null Hypothesis (H0):** $pnew$ **<=** $pold$

**Alternative Hypothesis (H1):** $pnew$ **>** $pold$

Null Hypothesis (H0): New page has a lower or equalivent conversion rate than old page

Alternative Hypothesis (H1): New page has a greater conversion rate than old page

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 **conversion rate** for $p_{new}$ under the null?

In [19]:
```
#Calculate conversion rate for p_new under null using mean function
p_new=df2['converted'].mean()
p_new
```

Out[19]: 0.11959708724499628

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

In [20]:
```
#Calculate conversion rate for p_old under null using mean function
p_old=df2['converted'].mean()
p_old
```

Out[20]: 0.11959708724499628

c. What is $n_{new}$, the number of individuals in the treatment group?

In [21]:
```
#Find the number of individuals in treatment group
n_new=df2[df2['group']=='treatment'].shape[0]
n_new
```

Out[21]: 145310

d. What is $n_{old}$, the number of individuals in the control group?

In [22]:
```
#Find the number of individuals in control group
n_old=df2[df2['group']=='control'].shape[0]
n_old
```

Out[22]: 145274

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

In [23]:
```
#Find simulated value for new page conversions
new_page_converted = np.random.binomial(n_new,p_new)
new_page_converted
```

Out[23]: 17184

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

In [24]:
```
#Find simulated value for old page conversions
old_page_converted = np.random.binomial(n_old,p_old)
old_page_converted
```

Out[24]: 17348

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

In [25]:
```
#Find simulated value for the difference of new page and old page conversions
difference= new_page_converted/n_new - old_page_converted/n_old
difference
```

Out[25]: -0.0011582063594498815

h. Create 10,000 $p_{new}$ - $p_{old}$ values using the same simulation process you used in parts (a) through (g) above. Store all 10,000 values in a NumPy array called **p_diffs**.

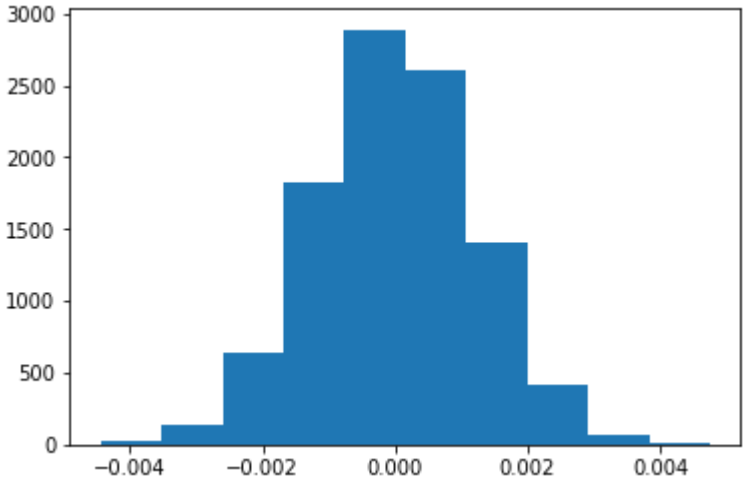
In [26]:
```
#Create 10000 simulated samples and store in NumPy

p_diffs = []
for _ in range(10000):
    new_page_converted = np.random.binomial(n_new,p_new)
    old_page_converted = np.random.binomial(n_old, p_old)
    difference = new_page_converted/n_new - old_page_converted/n_old
    p_diffs.append(difference)
```

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 [27]:
```
#Plot histogram
plt.hist(p_diffs)
```

Out[27]:
```
(array([   19.,   136.,   637.,  1819.,  2889.,  2600.,  1406.,   415.,
          66.,    13.]),
 array([-0.00444143, -0.0035226 , -0.00260378, -0.00168495, -0.00076612,
         0.00015271,  0.00107154,  0.00199037,  0.0029092 ,  0.00382803,
         0.00474686]),
 <a list of 10 Patch objects>)
```




The histogram is plotted as expected to represent the p_diffs, the difference between old and new page under the null with 10,000 sample values.

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

In [28]:
```
#Calculate actual difference with values from p_diffs
actual_difference = (df2[df2['group'] == "treatment"]['converted'].mean()) - (
df2[df2['group'] == "control"]['converted'].mean())
actual_difference
```

Out[28]:
```
-0.0015782389853555567
```

In [29]:
```
#Find proportion or p-value
p_value=(p_diffs > actual_difference).mean()
p_value
```

Out[29]:
```
0.90800000000000003
```

k. Please explain using the vocabulary you've learned in this course 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?

**With finding the difference with calculated results and actual results, we have the p-value.**

The p-value offers a sense as to whether or not, a particular varable would be useful for predicting the response to a test experiment. The p-value is the probability of obtaining observed results with holding the null hypothesis as true. A high p-value suggests that we fail to reject the null hypothesis while a low p-value suggests that we reject the null hypothesis.

In terms of this analysis, the calculated p-value of 0.90 means that we fail to reject the null hypothesis. This further tells us that the new page conversion will more than likely be lower or equalivent to the old page conversion. So, there is not a significant difference between the new and old pages in regard to conversion rate.

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 the the number of rows associated with the old page and new pages, respectively.

In [30]:
```
import statsmodels.api as sm

#Calculate the number of conversions for each page
convert_old = df2.query(" landing_page == 'old_page' and converted == 1").shap
e[0]
convert_new = df2.query(" landing_page == 'new_page' and converted == 1").shap
e[0]

#Calculate the number of individuals who received each page
n_old = df2[df2['group'] == 'control'].shape[0]
n_new = df2[df2['group'] == 'treatment'].shape[0]
```

```
/opt/conda/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: Futur
eWarning: The pandas.core.datetools module is deprecated and will be removed
in a future version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
```

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

In [31]:
```
#Compute test statistic and p-value
z_score, p_value = sm.stats.proportions_ztest([convert_new, convert_old], [n_n
ew, n_old], alternative = 'larger')
z_score, p_value
```

Out[31]:
```
(-1.3109241984234394, 0.90505831275902449)
```

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.**?

**Given a negative z-score and a high p-value, these calculations suggest that we fail to reject the null hypothesis. The conversion rates of old and new pages are not statistically significant. This is in agreement with findings in parts j and k that the new page conversion will be lower or equalivent to the old page conversion rates.**

## Part III - A regression approach

1. In this final part, you will see that the result you achieved in the A/B test in Part II above 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?

**Logistic Regression**

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 in df2 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 [32]:
```
#Create an intercept column
df2['intercept']=1

#Create a dummy variable column for which page each user received
df2[['control', 'ab_page']] = pd.get_dummies(df2['group'])
df2.head()
```

Out[32]:

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

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

In [33]:
```
#Perform statsmodels to logistic regression model
log = sm.Logit(df2['converted'], df2[['intercept','ab_page']])
```

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

In [34]:
```
#Obtain summary of results
results = log.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.366118
         Iterations 6
```

Out[34]:

Logit Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | converted | **No. Observations:** | 290584 |
| **Model:** | Logit | **Df Residuals:** | 290582 |
| **Method:** | MLE | **Df Model:** | 1 |
| **Date:** | Wed, 17 Apr 2019 | **Pseudo R-squ.:** | 8.077e-06 |
| **Time:** | 06:06:44 | **Log-Likelihood:** | -1.0639e+05 |
| **converged:** | True | **LL-Null:** | -1.0639e+05 |
| | | **LLR p-value:** | 0.1899 |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **intercept** | -1.9888 | 0.008 | -246.669 | 0.000 | -2.005 | -1.973 |
| **ab_page** | -0.0150 | 0.011 | -1.311 | 0.190 | -0.037 | 0.007 |

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 **Part II**?

**The p-value associated with ab_page is 0.190 and it differs from the value found in Part II (0.905) because an intercept was added.**

The null and alternative hypotheses associated with the logisitic regression model is whether the new page is equal or not equal to the old page.

For a logistic regression approach, it predicts only two possible outcomes and in this analysis, we are looking to see if (1) the new page equals old page or (2) new page does not equal old page.

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?

**It is a good idea to consider other factors to add to regression model because implementing a new page is not the only influencing factor to whether or not an individual converts. It is possible to consider the duration of the A/B test and characteristics of testers.**

There are disadvantages to adding additional terms to regression model with the chance of correlated errors, outliers, and multicollinearity which all can lower the R-squared value for the fit of our model to the data.

g. Now along with testing if the conversion rate changes for different pages, also add an effect based on which country a user lives in. 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 [35]:
```
#Add effect of country that users lives in
countries_df = pd.read_csv('countries.csv')
countries_df.head()
```

Out[35]:

| | user_id | country |
|---|---|---|
| **0** | 834778 | UK |
| **1** | 928468 | US |
| **2** | 822059 | UK |
| **3** | 711597 | UK |
| **4** | 710616 | UK |

In [36]:
```
#Merge datasets
df_new = countries_df.set_index('user_id').join(df2.set_index('user_id'), how=
'inner')
df_new.head()
```

Out[36]:

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

In [37]:
```
#Create dummy variables
df_new[['US', 'UK']] = pd.get_dummies(df_new['country'])[['US','UK']]
df_new.head()
```

Out[37]:

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

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 [38]:
```
#Create necessary additional columns
df_new['intercept'] = 1
log_new = sm.Logit(df_new['converted'], df_new[['US', 'UK', 'intercept', 'ab_page']])

#Fit the new model
results = log_new.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.366113
         Iterations 6
```

Out[38]:

Logit Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | converted | **No. Observations:** | 290584 |
| **Model:** | Logit | **Df Residuals:** | 290580 |
| **Method:** | MLE | **Df Model:** | 3 |
| **Date:** | Wed, 17 Apr 2019 | **Pseudo R-squ.:** | 2.323e-05 |
| **Time:** | 06:06:48 | **Log-Likelihood:** | -1.0639e+05 |
| **converged:** | True | **LL-Null:** | -1.0639e+05 |
| | | **LLR p-value:** | 0.1760 |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **US** | 0.0408 | 0.027 | 1.516 | 0.130 | -0.012 | 0.093 |
| **UK** | 0.0506 | 0.028 | 1.784 | 0.074 | -0.005 | 0.106 |
| **intercept** | -2.0300 | 0.027 | -76.249 | 0.000 | -2.082 | -1.978 |
| **ab_page** | -0.0149 | 0.011 | -1.307 | 0.191 | -0.037 | 0.007 |

# Analysis Findings

From my analysis, I fail to reject the null hypothesis as conversions for the new page does not perform better than the old page. With considering location of users, I still find that I fail to reject the null hypothesis as p-values are greater than 0.05 and so there is not enough significant evidence to reject the null hypothesis.

# References

https://stackoverflow.com/questions/38147027/action-with-pandas-settingwithcopywarning (https://stackoverflow.com/questions/38147027/action-with-pandas-settingwithcopywarning)

https://www.statsmodels.org/devel/generated/statsmodels.stats.proportion.proportions_ztest.html (https://www.statsmodels.org/devel/generated/statsmodels.stats.proportion.proportions_ztest.html)

https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.binomial.html (https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.binomial.html)

https://stackoverflow.com/questions/48820601/obtaining-summary-from-logistic-regressionpython (https://stackoverflow.com/questions/48820601/obtaining-summary-from-logistic-regressionpython)

# Finishing Up

> Congratulations! You have reached the end of the A/B Test Results project! You should be very proud of all you have accomplished!
>
> **Tip**: Once you are satisfied with your work here, check over your report to make sure that it is satisfies all the areas of the rubric (found on the project submission page at the end of the lesson). You should also probably remove all of the "Tips" like this one so that the presentation is as polished as possible.

# Directions to Submit

> Before you submit your project, you need to create a .html or .pdf version of this notebook in the workspace here. To do that, run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the orange Jupyter icon in the upper left).
>
> Alternatively, you can download this report as .html via the **File** > **Download as** submenu, and then manually upload it into the workspace directory by clicking on the orange Jupyter icon in the upper left, then using the Upload button.
>
> Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right here. This will create and submit a zip file with this .ipynb doc and the .html or .pdf version you created. Congratulations!

In [ ]: