# Analyze_ab_test_results_notebook

February 1, 2019

## 0.1 Analyze A/B Test Results

## 0.2 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.

#### 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 [2]: 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]: df.shape
```

```
Out[3]: (294478, 5)
```

c. The number of unique users in the dataset.

```
In [4]: df['user_id'].nunique()
```

```
Out[4]: 290584
```

d. The proportion of users converted.

```
In [5]: df['converted'].mean()
```

```
Out[5]: 0.11965919355605512
```

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

```
In [6]: df.query('(landing_page == "new_page" and group != "treatment") or (landing_page != "new
```

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

f. Do any of the rows have missing values?

```
In [7]: 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
```

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]: df2 = df[(df.group == 'treatment') & (df.landing_page == 'new_page')]
        df2 = df2.append(df[(df.group == 'control') & (df.landing_page == 'old_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].sha
```

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

```
Out[10]: 290584
```

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

```
In [11]: df2[df2.duplicated(['user_id'], keep=False)]['user_id']
```

```
Out[11]: 1899    773192
         2893    773192
         Name: user_id, dtype: int64
```

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

```
In [12]: 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]: df2.drop(df.index[1899], inplace = True)
```

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 [14]: df2['converted'].mean()
```

```
Out[14]: 0.11959708724499628
```

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

```
In [15]: df2.query('group == "control"')['converted'].mean()
```

```
Out[15]: 0.1203863045004612
```

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

```
In [16]: df2.query('group == "treatment"')['converted'].mean()
```

```
Out[16]: 0.11880806551510564
```

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

```
In [17]: df2.query('landing_page == "new_page"')['user_id'].count() / df2['user_id'].count()
```

```
Out[17]: 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.

No, I do not think there is sufficient evidence based on the above information to conclude that the new page leads to more conversions. The new page actually has a slightly lower converesion rate than the old page and neither the rate of conversion for the new page nor the old page differs much at all from the overall conversion rate.

### 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.

h0: The old page is better than the new page / no difference

ha: The new page is better than the old page at a p value of 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 **conversion rate** for $p_{new}$ under the null?

```
In [18]: p_new = df2['converted'].mean()
         p_new
```

Out[18]: 0.11959708724499628

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

```
In [19]: p_old = df2['converted'].mean()
         p_old
```

Out[19]: 0.11959708724499628

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

```
In [20]: n_new = df2.query('group == "treatment"')['user_id'].count()
         n_new
```

Out[20]: 145310

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

```
In [21]: n_old = df2.query('group == "control"')['user_id'].count()
         n_old
```

Out[21]: 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 [22]: new_page_converted = np.random.binomial(n_new, p_new)
```

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 [23]: old_page_converted = np.random.binomial(n_old, p_old)
```

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

```
In [24]: (new_page_converted / n_new) - (old_page_converted / n_old)
```

Out[24]: -0.0013509694722351612

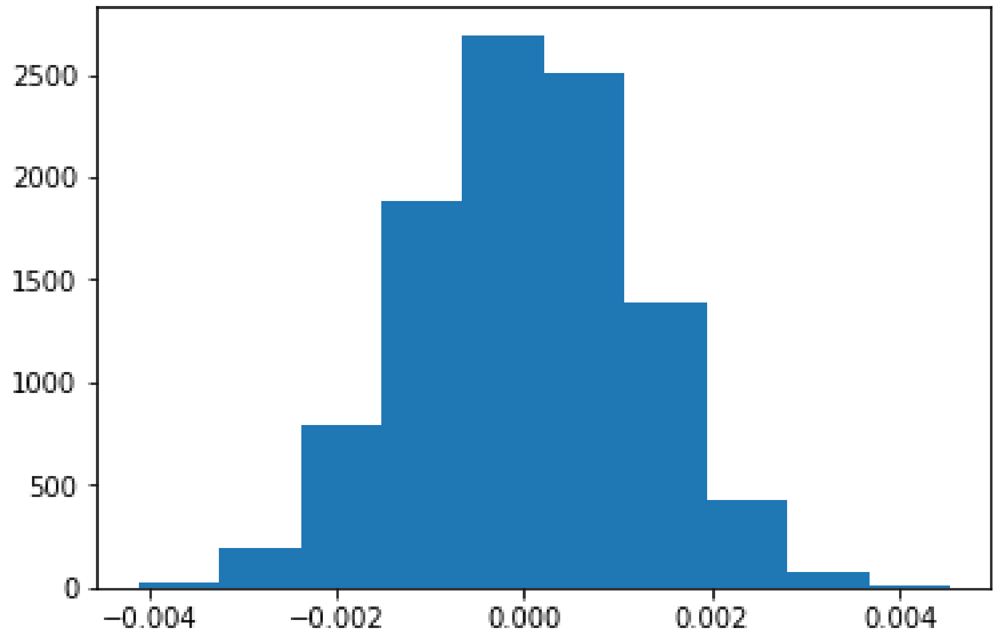
h. Create 10,000 $p_{new}$ - $p_{old}$ values using the same simulation process you used in parts (a) through (g) above. Store all 10,000 values in a NumPy array called **p_diffs**.

```
In [25]: 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)
             p_diffs.append((new_page_converted / n_new) - (old_page_converted / n_old))
```

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 [26]:
```
plt.hist(p_diffs);
```




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

In [27]:
```
actual_dif = (df2[df2['group'] == 'treatment']['converted'].mean()) - (df2[df2['group']
p_diffs = np.array(p_diffs)
p_value = (p_diffs > actual_dif).mean()
p_value
```

Out[27]: 0.9082000000000001

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?

**What we just computed in part j is called the p-value. The p-value is a measure of the probability of getting a specific result if the null hypothesis is true. A large p-value (close to 1) means that the null hypothesis is very likely true and we should accept the null, whereas a small p-value (close to 0) means that the alternative hypothesis is likely true and we should reject the null. Since the p-value we just computed based on the ab data is very large (p > 0.9) we can fairly safely conclude that we should accept the null hypothesis and say that the new page did not work better than the old page.**

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 [28]: import statsmodels.api as sm

         convert_old = df2.query("landing_page == 'old_page' and converted == 1")['user_id'].cou
         convert_new = df2.query("landing_page == 'old_page' and converted == 1")['user_id'].cou
         n_old = df2[df2['landing_page'] == 'old_page']['user_id'].count()
         n_new = df2[df2['landing_page'] == 'new_page']['user_id'].count()
```

```
/opt/conda/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The panda
  from pandas.core import datetools
```

m. Now use `stats.proportions_ztest` to compute your test statistic and p-value. Here is a helpful link on using the built in.

```
In [29]: z_score, p_value = sm.stats.proportions_ztest([convert_old, convert_new], [n_old, n_new
         print(z_score)
         print(p_value)
```

```
0.0247046451343
0.509854725034
```

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.**?

**The z-score and p-value computed in part m indicate that the null hypothesis is supported by the data and we should accept the null hypothesis that the new page is not better than the old page. This is in agreement with the earlier findings.**
### 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 [30]: df2['intercept'] = 1
         df2[['control', 'ab_page']] = pd.get_dummies(df2['group'])
         df2.drop(['control'], axis=1, inplace=True)
         df2.head()
```

```
Out[30]:    user_id                   timestamp      group landing_page  converted  \
         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
         6   679687  2017-01-19 03:26:46.940749  treatment     new_page          1
         8   817355  2017-01-04 17:58:08.979471  treatment     new_page          1
         9   839785  2017-01-15 18:11:06.610965  treatment     new_page          1

            intercept  ab_page
         2          1        1
         3          1        1
         6          1        1
         8          1        1
         9          1        1
```

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 [31]: logistic = sm.Logit(df2['converted'], df2[['intercept', 'ab_page']])
         result = logistic.fit()
```

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

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

```
In [32]: result.summary()
```

```
Out[32]: <class 'statsmodels.iolib.summary.Summary'>
         """
                                    Logit Regression Results
         ==============================================================================
         Dep. Variable:              converted   No. Observations:               290584
         Model:                          Logit   Df Residuals:                   290582
         Method:                           MLE   Df Model:                            1
         Date:                Fri, 01 Feb 2019   Pseudo R-squ.:               8.077e-06
         Time:                        16:48:23   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
================================================================================
"""
```

In [33]:
```
np.exp(-0.015)
```

Out[33]:
```
0.98511193960306265
```

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 based on the logistic regression is 0.190. This is different from the value found in Part II because the alternative hypotheses between the two methods are different. In Part II we used a one-sided alternative hypothesis that the new page would be better than the old page. In Part III we used a two-sided alternative hypothesis that the new page would be differnt from the old page either better or worse.**

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 would be a good idea to consider other factors in our regression model because there are likely other factors that will influence whether an individual converts or not. As in most any statistical analysis situation, there are likely to be more factors affecting the outcome other than the intended independant variable. Often times, there are confounding variables that we cannot control and should consider. In this situation, some other factors that may have an impact on the outcome are age, socio-economic status, or nationality. It may be a good idea to try to acount for some of these other factors. However, sometimes including these other factors can unnecessarily complicate the analysis. In many cases it may be unnecessary to include other factors in our analysis because it is likely that if our sample size is large enough, there will be a nearly equal distribution of these other factors and they will effectivly cancel out.**

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 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 [34]:
```
df_countries = pd.read_csv('countries.csv')
```

In [35]:
```
df3 = df_countries.set_index('user_id').join(df2.set_index('user_id'), how='inner')
df3.head()
```

Out[35]:
```
        country                   timestamp      group landing_page  \
user_id
834778       UK  2017-01-14 23:08:43.304998    control     old_page
928468       US  2017-01-23 14:44:16.387854  treatment     new_page
822059       UK  2017-01-16 14:04:14.719771  treatment     new_page
711597       UK  2017-01-22 03:14:24.763511    control     old_page
710616       UK  2017-01-16 13:14:44.000513  treatment     new_page

         converted  intercept  ab_page
user_id
834778           0          1        0
928468           0          1        1
822059           1          1        1
711597           0          1        0
710616           0          1        1
```

In [36]:
```
df3[['CA', 'UK', 'US']] = pd.get_dummies(df3['country'])
df3.head()
```

Out[36]:
```
        country                   timestamp      group landing_page  \
user_id
834778       UK  2017-01-14 23:08:43.304998    control     old_page
928468       US  2017-01-23 14:44:16.387854  treatment     new_page
822059       UK  2017-01-16 14:04:14.719771  treatment     new_page
711597       UK  2017-01-22 03:14:24.763511    control     old_page
710616       UK  2017-01-16 13:14:44.000513  treatment     new_page

         converted  intercept  ab_page  CA  UK  US
user_id
834778           0          1        0   0   1   0
928468           0          1        1   0   0   1
822059           1          1        1   0   1   0
711597           0          1        0   0   1   0
710616           0          1        1   0   1   0
```

In [37]:
```
logistic2 = sm.Logit(df3['converted'], df3[['intercept', 'CA', 'UK']])
result2 = logistic2.fit()
result2.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.366116
         Iterations 6
```

Out[37]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:              converted   No. Observations:               290584
```

```
Model:                          Logit   Df Residuals:                   290581
Method:                           MLE   Df Model:                            2
Date:                Fri, 01 Feb 2019   Pseudo R-squ.:               1.521e-05
Time:                        16:48:25   Log-Likelihood:            -1.0639e+05
converged:                       True   LL-Null:                   -1.0639e+05
                                        LLR p-value:                    0.1984
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -1.9967      0.007   -292.314      0.000      -2.010      -1.983
CA            -0.0408      0.027     -1.518      0.129      -0.093       0.012
UK             0.0099      0.013      0.746      0.456      -0.016       0.036
==============================================================================
"""
```

In [38]:
```
np.exp(result2.params)
```

Out[38]:
```
intercept    0.135779
CA           0.960018
UK           1.009966
dtype: float64
```

**Using the US as the baseline, meaning the null hypothesis is that Canada and the UK are not any different than the US and the alternative hypothesis is that Canada or the UK are different than the US, we find that Canadian users are about 4% less likely to convert and that UK users are about 1% more likley to convert than American users. These results, however, are not statistically significant as the p-values for both Canada and UK are much too high.**

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 [39]:
```
df3['CA_new'] = df3['ab_page'] * df3['CA']
df3['UK_new'] = df3['ab_page'] * df3['UK']
df3['US_new'] = df3['ab_page'] * df3['US']
df3.head()
```

Out[39]:
```
        country                   timestamp      group landing_page  \
user_id
834778       UK  2017-01-14 23:08:43.304998    control     old_page
928468       US  2017-01-23 14:44:16.387854  treatment     new_page
822059       UK  2017-01-16 14:04:14.719771  treatment     new_page
711597       UK  2017-01-22 03:14:24.763511    control     old_page
710616       UK  2017-01-16 13:14:44.000513  treatment     new_page

         converted  intercept  ab_page  CA  UK  US  CA_new  UK_new  US_new
user_id
```

```
834778          0          1         0   0   1   0         0         0         0
928468          0          1         1   0   0   1         0         0         1
822059          1          1         1   0   1   0         0         1         0
711597          0          1         0   0   1   0         0         0         0
710616          0          1         1   0   1   0         0         1         0
```

In [40]:
```python
logistic3 = sm.Logit(df3['converted'], df3[['intercept', 'ab_page', 'CA_new', 'UK_new']
result3 = logistic3.fit()
result3.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.366109
         Iterations 6
```

Out[40]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:              converted   No. Observations:               290584
Model:                          Logit   Df Residuals:                   290580
Method:                           MLE   Df Model:                            3
Date:                Fri, 01 Feb 2019   Pseudo R-squ.:               3.351e-05
Time:                        16:48:25   Log-Likelihood:            -1.0639e+05
converged:                       True   LL-Null:                   -1.0639e+05
                                        LLR p-value:                   0.06785
==============================================================================
                 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       -0.0183      0.013     -1.449      0.147      -0.043       0.006
CA_new        -0.0644      0.038     -1.679      0.093      -0.140       0.011
UK_new         0.0257      0.019      1.363      0.173      -0.011       0.063
==============================================================================
"""
```

In [41]:
```python
np.exp(result3.params)
```

Out[41]:
```
intercept    0.136863
ab_page      0.981901
CA_new       0.937618
UK_new       1.025986
dtype: float64
```

**The logistic regression shows that Canadian users who received the new page are about 4% less likely to convert than all users who received the new page and that UK users who received the new page are about 4.5% more likely to convert than all users who received the new page. None of these results are statistically significant though because the p-values are too high.**

## Conclusion

**Overall, none of the results of this study have low enough p-values to be considered statistically significant. It seems that the pages perform similarly, perhaps with the new page being a little worse performing than the old, but this difference is neither statistically nor practically significant. My conclusion is that the test results are inconclusive and the test needs to be run longer in order to reach a definitive result.**

In [42]:
```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])
```

Out[42]: 0