Analyze_ab_test_results_notebook

August 27, 2020

0.1 Analyze A/B Test Results

0.2 Table of Contents

Introduction
For this project, an A/B test will be performed on an e-commerce website and the 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.
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[((df['group'] == 'treatment') == (df['landing_page'] == 'new_page')) == False].shape[

Out[6]: 3893
```

f. Do any of the rows have missing values?

```
In [7]: df.isnull().sum()

Out[7]: user_id         0
        timestamp       0
        group           0
        landing_page    0
        converted       0
        dtype: int64
```

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')) != False]

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.user_id.duplicated()]
```

```
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]: 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(1899, inplace=True)
         df2[df2.user_id == 773192]
```

```
/opt/conda/lib/python3.6/site-packages/pandas/core/frame.py:3697: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  errors=errors)
```

```
Out[13]:       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 [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]: control_convert = df2.query('group == "control"')['converted'].mean()
         control_convert
```

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

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

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

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

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

```
In [17]: (df2['landing_page'] == "new_page").mean()
```

```
Out[17]: 0.50006194422266881
```

```
In [18]: # Timestamp range of the collected data
         max(df2.timestamp), min(df2.timestamp)
```

```
Out[18]: ('2017-01-24 13:41:54.460509', '2017-01-02 13:42:05.378582')
```

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.

There is no sufficient evidentce to conclude that the new treatment page leads to more conversions, as control group conversion probability is nearly the same as treatment group. And treatment group converted probability is actually slightly less than the control group, while the probability of individual received the new page is 50%.

Part II - A/B Test

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.

$H_0 : p_{new}$ - p_{old} 0; $H_1 : p_{new}$ - $p_{old} > 0$

`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.

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

```
In [19]: p_new = (df2.converted == 1).mean()
         p_new
```

```
Out[19]: 0.11959708724499628
```

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

```
In [20]: p_old = (df2.converted == 1).mean()
         p_old
```

```
Out[20]: 0.11959708724499628
```

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

```
In [21]: n_new = (df2['landing_page'] == "new_page").sum()
         n_new

Out[21]: 145310
```

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

```
In [22]: n_old = (df2['landing_page'] == "old_page").sum()
         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]: new_page_converted = np.random.choice([0, 1], size=n_new, p=[1-p_new, p_new])
         new_page_converted

Out[23]: array([0, 1, 0, ..., 0, 0, 0])
```

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]: old_page_converted = np.random.choice([0, 1], size=n_old, p=[1-p_old, p_old])
         old_page_converted

Out[24]: array([0, 0, 0, ..., 0, 0, 1])
```

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

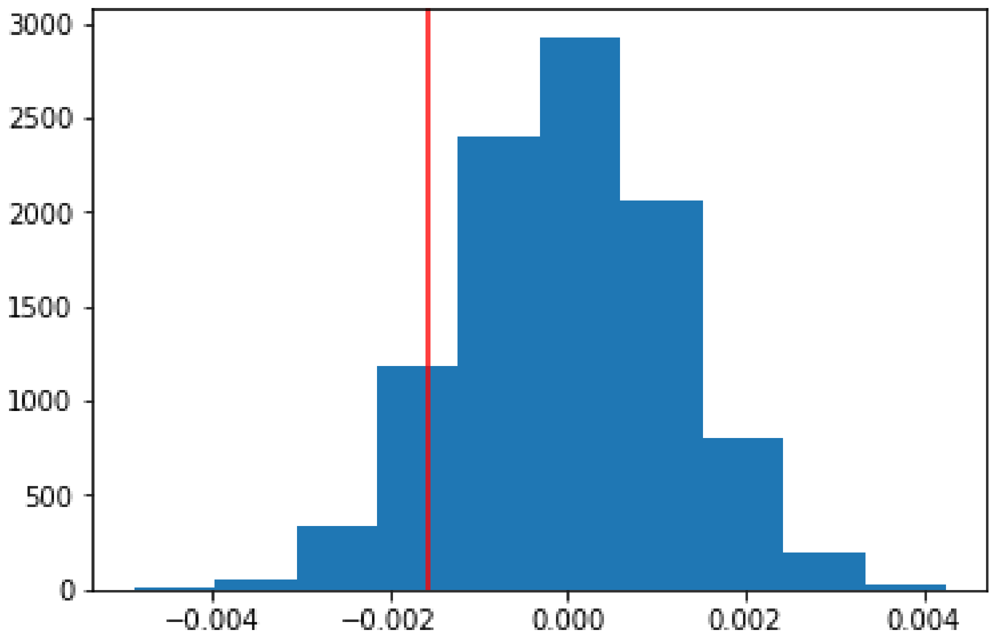
```
In [25]: new_page_converted.mean() - old_page_converted.mean()

Out[25]: -0.00016724885426412651
```

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 [32]: p_diffs = []
         for _ in range(10000):
             new_page_converted = np.random.choice([0, 1], size=n_new, p=[(1-p_new), p_new])
             old_page_converted = np.random.choice([0, 1], size=n_old, p=[(1-p_old), p_old])
             p_diffs.append(new_page_converted.mean() - old_page_converted.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 [33]: p_diffs = np.array(p_diffs)
         obs_diff = treatment_convert - control_convert
         plt.hist(p_diffs);
         plt.axvline(x=obs_diff, color='red');
```



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

```
In [34]: (p_diffs > obs_diff).mean()
```

```
Out[34]: 0.90159999999999996
```

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? **This value is called p-value in statistics, it represents the probability of obtaining a sample more extreme than the ones observed in our data, assuming null hypothesis is True. Since this is a right tail test with type I error rate () of 0.05 and p-value is 0.9, p-value is greater than , we fail to reject null hypothesis, which means There is no sufficient evidence to conclude there is converted probability difference between the new and old pages.**

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 [35]: import statsmodels.api as sm

         convert_old = df2.query('landing_page == "old_page"')['converted'].sum()
```

```
convert_new = df2.query('landing_page == "new_page"')['converted'].sum()
n_old = (df2['landing_page'] == "old_page").sum()
n_new = (df2['landing_page'] == "new_page").sum()
convert_old, convert_new, n_old, n_new
```

```
Out[35]: (17489, 17264, 145274, 145310)
```

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 [36]: z_score, p_value = sm.stats.proportions_ztest([convert_new, convert_old], [n_new, n_old
         z_score, p_value
```

```
Out[36]: (-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.**?

z-score is the deviation from the mean in units of standard deviation; p-value is the probability of obtaining a sample more extreme than the ones observed in our data when the null hypothesis is true. Based on z-score, we can get the corresponding p-value (normally from z-test table) and from there, p-value of 0.9 could be captured. P-value of 0.9 is greater than the given , 0.05, thus, we fail to reject the null hypothesis that new pages have better conversion rates than new pages. With the same p-value computed in parts j. and k., 0.9, the findings are agree with each other.

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 should be performed as it predicts a probability between 0 and 1.

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 [37]: df2['intercept'] = 1
         df2['ab_page'] = pd.get_dummies(df2['group'])['treatment']
         df2.head()
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
```

```
  """Entry point for launching an IPython kernel.
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
```

```
Out[37]:    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

            intercept  ab_page
         0          1        0
         1          1        0
         2          1        1
         3          1        1
         4          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 [38]: log_mod = sm.Logit(df2['converted'], df2[['intercept',  'ab_page']])
         results = log_mod.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 [39]: from scipy import stats
         stats.chisqprob = lambda chisq, df: stats.chi2.sf(chisq, df)
         results.summary()
```

```
Out[39]: <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:                 Thu, 27 Aug 2020   Pseudo R-squ.:              8.077e-06
Time:                         03:36:59   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**?

P-value associated with ab_page is 0.19, which is higher than 0.05, it means ab_page does not have impact on predicting the conversion rates. Part II model null hypothesis assumes that the old page is significantly better than the new page in terms of conversion rate; Part III model is used to examine the association of independent variable (ab_page) with dependent variable (converted) to see if conversion rate will change with ab_page, for example, null hypothesis is: when ab_page is 0, converted = 1 and the alternative hypothesis: when ab_page = 0, converted = 0.

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?

Adding in more factors or adopting multiple regression model is more accurate in prediction than simple regression model. There are factors such as features of users, age, country, occupation etc. that might also influence the outcome of prediction. It also helps us to fit the estimated lines better.

There are also potential problems associated with Multiple Linear Regression, when adding in new factors we need to identify whether any of the following problems exists and address them accordingly:

1. Non-linearity of the response-predictor relationships
2. Correlation of error terms
3. Non-constant Variance and Normally Distributed Errors
4. Outliers/ High leverage points
5. Multicollinearity

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.

```
In [40]: df_countries = pd.read_csv('countries.csv')
         df_countries.head()
```

```
Out[40]:    user_id country
         0   834778      UK
         1   928468      US
         2   822059      UK
         3   711597      UK
         4   710616      UK
```

```
In [41]: df3 = df_countries.set_index('user_id').join(df2.set_index('user_id'), how='inner')
         df3.head()
```

```
Out[41]:         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 [42]: df3['country'].unique()
```

```
Out[42]: array(['UK', 'US', 'CA'], dtype=object)
```

```
In [43]: df3[['CA', 'UK', 'US']] = pd.get_dummies(df3['country'])
         df3.head()
```

```
Out[43]:         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 [44]:
```
log_mod = sm.Logit(df3['converted'], df3[['intercept',  'ab_page', 'CA', 'UK']])
results = log_mod.fit()
results.summary()
```

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

Out[44]:
```
<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:                Thu, 27 Aug 2020   Pseudo R-squ.:               2.323e-05
Time:                        03:37:03   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]
------------------------------------------------------------------------------
intercept     -1.9893      0.009   -223.763      0.000      -2.007      -1.972
ab_page       -0.0149      0.011     -1.307      0.191      -0.037       0.007
CA            -0.0408      0.027     -1.516      0.130      -0.093       0.012
UK             0.0099      0.013      0.743      0.457      -0.016       0.036
==============================================================================
"""
```

Findings: Based on the result summary, all of the variables have p-value greater than 0.05, countries do not have impact on predicting the conversion rates.

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 [45]:
```
df3['CA_page'] = df3['ab_page'] * df3['CA']
df3['UK_page'] = df3['ab_page'] * df3['UK']
df3['US_page'] = df3['ab_page'] * df3['US']
df3.head()
```

Out[45]:
```
        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_page  UK_page  US_page
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 [46]:
```python
log_mod = sm.Logit(df3['converted'], df3[['intercept', 'CA_page', 'UK_page']])
results = log_mod.fit()
results.summary()
```

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

Out[46]:
```
<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:                Thu, 27 Aug 2020   Pseudo R-squ.:               2.364e-05
Time:                        03:37:07   Log-Likelihood:            -1.0639e+05
converged:                       True   LL-Null:                   -1.0639e+05
                                        LLR p-value:                   0.08085
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -1.9963      0.006   -322.049      0.000      -2.008      -1.984
CA_page       -0.0752      0.038     -1.997      0.046      -0.149      -0.001
UK_page        0.0149      0.017      0.862      0.389      -0.019       0.049
==============================================================================
"""
```

In [47]:
```python
1/np.exp(results.params)
```

Out[47]:
```
intercept    7.361591
CA_page      1.078076
UK_page      0.985222
dtype: float64
```

Findings: 1. P-values for CA_page is less than 0.05, which means that the interation between Canada users and new page is statistically significant in predicting the conversion of users; 2. While holding UK_page constant, the chance of coversion is 1.078 times more likely for CA_page.

Conclusion

For this project, both A/B test and logistiic regression approach were performed on an e-commerce website 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. Based on results from both A/B test and logistic regression model, there is no sufficient evidence to conclude that the new page works better than the old page in terms of conversion rate.

As we only analyzed dataset over a relatively short time frame, less than 1 month, it is suggested to run the experiment longer to make decision, especially there are potentially bias results when tested on existing users, due to factors like change aversion and novelty effect. Meanwhile, other factors such as click through rate could also be examined. Finally, practical significance of a conversion rate should also be taken into consideration, for example the cost of launching a new page vs the revenue obtained from the increase in conversion.

```
In [49]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])

Out[49]: 0

In [ ]:
```