Analyze_ab_test_results_notebook

June 25, 2020

0.1 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. **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!

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 [188]:
```
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 [189]: df = pd.read_csv('ab_data.csv')
          df.head()
```

Out[189]:

	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 [190]: print('Number of Rows is {}'.format(df.shape[0]))
```

```
Number of Rows is 294478
```

c. The number of unique users in the dataset.

```
In [191]: print('Number of unique users is {}'.format(df.user_id.nunique()))
```

```
Number of unique users is 290584
```

d. The proportion of users converted.

```
In [192]:  n = df.groupby('converted').user_id.nunique()/df.user_id.nunique()

          print('Proportion of users converted is {}'.format(n[1]))
```

```
Proportion of users converted is 0.12104245244060237
```

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

```
In [193]: treat = df.query('group == "treatment"')
          cont = df.query('group == "control"')

In [194]: treat_old = treat.query('landing_page == "old_page"')
          cont_new = cont.query(('landing_page == "new_page"'))

          n = treat.query('landing_page == "old_page"').shape[0] + cont.query('landing_page == "
          print('The number of times the new_page and treatment dont match is {}'.format(n))
```

```
The number of times the new_page and treatment dont match is 3893
```

f. Do any of the rows have missing values?

```
In [195]: df.isna().sum()

Out[195]: 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 [196]: print(df.shape)
          treat.query('landing_page == "old_page"').shape[0], cont.query('landing_page == "new_p

(294478, 5)


Out[196]: (1965, 1928)

In [197]: df2 = df.drop(treat_old.index, axis = 0)
          df2 = df2.drop(cont_new.index, axis = 0)

          df2.shape[0] - df.shape[0]

Out[197]: -3893

In [198]: # Double Check all of the correct rows were removed - this should be 0
          df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) == False].s

Out[198]: 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 [200]: n = df2.user_id.nunique()
          print("Number of Unique user id's {}".format(n))

Number of Unique user id's 290584
```

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

```
In [14]: df2[df2.duplicated('user_id') == True]

         df2.query('user_id == 773192')
```

Out[14]:

	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

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

```
In [15]: df2[df2.duplicated('user_id') == True].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1 entries, 2893 to 2893
Data columns (total 5 columns):
user_id         1 non-null int64
timestamp       1 non-null object
group           1 non-null object
landing_page    1 non-null object
converted       1 non-null int64
dtypes: int64(2), object(3)
memory usage: 48.0+ bytes
```

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

```
In [16]: df2.drop(2893, axis = 0, inplace = True)

In [17]: df2.head()
```

Out[17]:

	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

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 [18]: df2.converted.mean()
```

```
Out[18]: 0.11959708724499628
```

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

```
In [19]: df2.groupby('group').converted.mean()
```

```
Out[19]: group
         control      0.120386
         treatment    0.118808
         Name: converted, dtype: float64
```

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

```
In [20]: df2.groupby('group').converted.mean()

Out[20]: group
         control      0.120386
         treatment    0.118808
         Name: converted, dtype: float64
```

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

```
In [21]: df2.landing_page.value_counts()/df2.shape[0]

Out[21]: new_page    0.500062
         old_page    0.499938
         Name: landing_page, dtype: float64

In [ ]:

In [ ]:
```

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.

Although the evidence shows that the control group had more conversions than the treatment group, we cannot conclude the same since

- The tests need to be done for longer

- More Samples need to be taken

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.

$$H_o : P_o >= P_n$$

$$H_1 : P_n > P_o$$

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 [26]: p_new = df2.converted.mean()
```

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

```
In [27]: p_old = df2.converted.mean()
```

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

```
In [202]: n_new = df2.query("group == 'treatment'").shape[0]
          n_new

Out[202]: 145311
```

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

```
In [29]: n_old = df2.query("group == 'control'").shape[0]
```

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 [147]: new_page_converted = np.random.binomial(n_new,p_new)
          new_page_converted

Out[147]: 17478
```

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

Out[148]: 17388
```

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

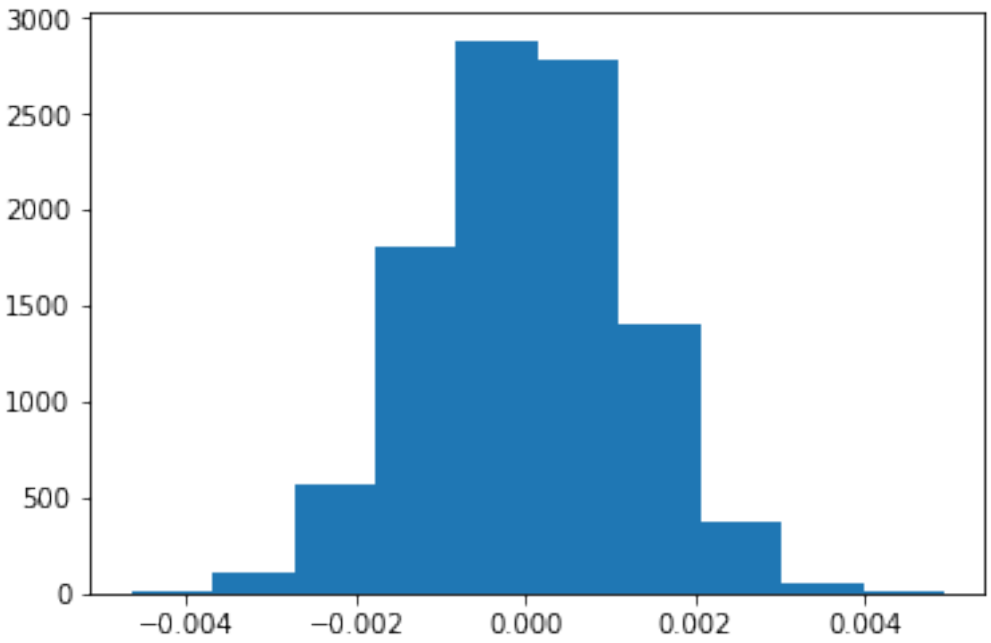
```
In [156]: p_diff = (new_page_converted/n_new) - (old_page_converted/n_old)

          p_diff


Out[156]: 0.0005897124878170151
```

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 [150]: np.random.seed(42)

          p_diffs = []

          for i in range(10000):
              x = np.random.binomial(n_new,p_new)
              y = np.random.binomial(n_old,p_old)
              p_diffs.append(-(y/n_old) + (x/n_new))
```

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 [151]: plt.hist(p_diffs)

Out[151]: (array([   17.,   108.,   562.,  1808.,  2883.,  2786.,  1397.,   377.,
                     56.,     6.]),
           array([-0.00465482, -0.00369605, -0.00273728, -0.00177851, -0.00081974,
                   0.00013903,  0.0010978 ,  0.00205657,  0.00301534,  0.00397411,
                   0.00493288]),
           <a list of 10 Patch objects>)
```



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

```
In [157]: p_obsdiff = - df2.query("landing_page == 'old_page'").converted.mean() + df2.query("la
          p_obsdiff
```

```
Out[157]: -0.0015782389853555567
```

```
In [158]: (p_diffs > p_obsdiff).mean()
```

```
Out[158]: 0.90449999999999997
```

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?

Essentially we are calculating the parameter called P-Value. Here the P-Value is greater than 0.5 and hence the Null hypothesis is significant or we fail to reject the null hypothesis

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 [98]: 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.query("landing_page == 'old_page'").shape[0]
         n_new = df2.query("landing_page == 'new_page'").shape[0]

         convert_old, convert_new, n_old, n_new
```

```
Out[98]: (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 [105]: z_stat, p_value = sm.stats.proportions_ztest([convert_old, convert_new], [n_old,n_new]

          z_stat, p_value
```

```
Out[105]: (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.**?

The Z score is essentially how many std dev from the mean the Raw score is actually is or how far from the mean the data lies. We know that the type 1 error is = 5% and from the p_value as seen above is more than 5% which is essentially showing that we cannot reject the null hypothesis

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?

The best fit for our analysis seems to be 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 [116]: df2['intercept'] = 1

          df2[['ab', 'ab_page']] = pd.get_dummies(df2['group'])
          df2.drop('ab', axis = 1, inplace = True)
```

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 [124]: from scipy import stats
          stats.chisqprob = lambda chisq, df: stats.chi2.sf(chisq, df)

          log_mod = sm.Logit(df2['converted'], df2[['intercept', 'ab_page']])
          result = 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 [125]: result.summary()

Out[125]: <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, 25 Jun 2020   Pseudo R-squ.:               8.077e-06
Time:                        00:03:16   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**?

For Part2

$$H_0 : P_n <= P_o$$

$$H_0 : P_n > P_o$$

For Part3

$$H_0 : P_n = P_o$$

$$H_0 : P_n! = P_o$$

The difference in hypothesis for the 2 sections can be seen above
Here the P-Value shows to be 0.19 which is greater than 0.05. This proves that the treatment group does not have a statistical significance to the converted data and hence we fail to reject the Null hypothesis.

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?

There are a lot of factors that could influence our analysis. For example, Gender can be a good variable to see the influence of Male and Female choices on the 2 different websites. Another factor might be the the reviews associated with the people who took the courses. In short, although multiple factors can be added or removed, we must understand that there could also be other factors such as multicollinearity that could also influence the other variables for our analysis

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 [142]: cf = pd.read_csv('countries.csv')
          cf.info()
          #cf_df = cf.join(df2)

          c = pd.merge(df2, cf, on = 'user_id')
          c.head()
          c.info()

          c[['CA','UK','US']] = pd.get_dummies(c['country'])

          c.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 290584 entries, 0 to 290583
Data columns (total 2 columns):
user_id    290584 non-null int64
country    290584 non-null object
dtypes: int64(1), object(1)
memory usage: 4.4+ MB
<class 'pandas.core.frame.DataFrame'>
Int64Index: 290584 entries, 0 to 290583
Data columns (total 8 columns):
user_id         290584 non-null int64
timestamp       290584 non-null object
group           290584 non-null object
landing_page    290584 non-null object
converted       290584 non-null int64
intercept       290584 non-null int64
ab_page         290584 non-null uint8
country         290584 non-null object
dtypes: int64(3), object(4), uint8(1)
memory usage: 18.0+ MB
```

Out[142]:

	user_id	timestamp	group	landing_page	converted	intercept	ab_page	country	CA	UK	US
0	851104	2017-01-21 22:11:48.556739	control	old_page	0	1	0	US	0	0	1
1	804228	2017-01-12 08:01:45.159739	control	old_page	0	1	0	US	0	0	1
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						

```
2          1        1      US   0   0   1
3          1        1      US   0   0   1
4          1        0      US   0   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 [144]: log_mod = sm.Logit(c['converted'], c[['intercept', 'ab_page', 'CA', 'UK']])
          result = log_mod.fit()
          result.summary()
```

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

```
Out[144]: <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, 25 Jun 2020   Pseudo R-squ.:               2.323e-05
          Time:                        00:36:09   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
          ==============================================================================
          """
```

Here looking at the P-Value, we can confirm that none of the variables have any statistical significance due to their high values(> 0.05). This shows that we reject to fail the Null Hypothesis.

Conclusions

In this project we have done multiple analysis to see which website would be better for the new website. We can see that all the analysis leads us to believe that the New Website will not improve the conversion to sales. For now the advice is to stick to the present website until a new layout can be made and then proceed with the steps on top. Moreover, other variables can be chosen such as genders or review ratings. Nonetheless, more information needs to be provided to make sure no other variables have any statistical significance for the same

```
In [203]: from subprocess import call
          call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])
```

```
Out[203]: 0
```

```
In [ ]:
```