Analyze_ab_test_results_notebook

June 27, 2018

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 [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 below cell 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.head()
        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 line up.

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

```
Out[6]: user_id         3893
        timestamp       3893
        group           3893
        landing_page    3893
        converted       3893
        dtype: int64
```

f. Do any of the rows have missing values?

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

```
<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
```

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

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

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

```
In [8]: df2 = df.query("(group == 'control' and landing_page == 'old_page') or (group == 'treatm
```

```
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]: df.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'])]['user_id'].unique()
```

```
Out[11]: array([773192])
```

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

```
In [12]: df2[df2.duplicated(['user_id'], keep=False)]
```

```
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(labels=1899, axis=0, inplace=True)
```

```
/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

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  """Entry point for launching an IPython kernel.
```

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

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

```
In [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"').count()[0])/df2.shape[0]
```

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

e. Use the results in the previous two portions of this question to suggest if you think there is evidence that one page leads to more conversions? Write your response below.

#The Control Group (i.e, group with old page) converted with higher rate than that of Treatment Group (i.e, group with new page). But the important point to note is that the difference in the rate of conversion is very less roughly around 0.2%, so we can't conclude with this. #The Probability that an individual received a new page is approximately is 0.5, which means that it's not possible to conclude with respect to conversion. We need to analyze even further to conclude.

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

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

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

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

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

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

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

Out[18]: 0.11959708724499628
```

b. What is the **convert 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}?

```
In [20]: n_new = df2[df2['group'] == 'treatment'].shape[0]
         n_new

Out[20]: 145310
```

d. What is n_{old}?

```
In [21]: n_old = df2[df2['group'] == 'control'].shape[0]
         n_old

Out[21]: 145274
```

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

```
In [22]: new_page_converted = np.random.choice([1, 0], size=n_new, p=[p_new, (1-p_new)])
         new_page_converted.mean()
```

```
Out[22]: 0.1196545316908678
```

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

```
In [23]: old_page_converted = np.random.choice([1, 0], size=n_old, p=[p_old, (1-p_old)])
         old_page_converted.mean()
```

```
Out[23]: 0.11912661591200077
```

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

```
In [24]: new_page_converted.mean()-old_page_converted.mean()
```

```
Out[24]: 0.00052791577886703023
```

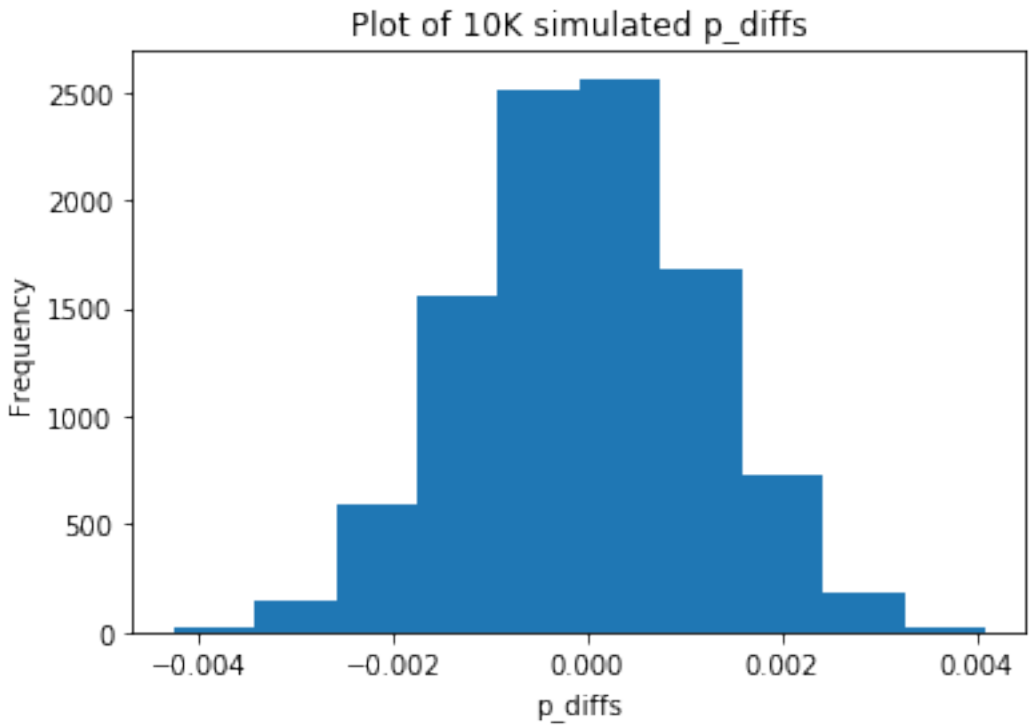
h. Simulate 10,000 p_{new} - p_{old} values using this same process similarly to the one you calculated in parts **a. through g.** above. Store all 10,000 values in **p_diffs**.

```
In [25]: p_diffs = []

         for _ in range(10000):
             new_page_converted = np.random.choice([1, 0], size=n_new, p=[p_new, (1-p_new)]).mea
             old_page_converted = np.random.choice([1, 0], size=n_old, p=[p_old, (1-p_old)]).mea
             diff = new_page_converted - old_page_converted
             p_diffs.append(diff)
```

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);
         plt.xlabel('p_diffs')
         plt.ylabel('Frequency')
         plt.title('Plot of 10K simulated p_diffs');
```



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

```
In [27]: act_diff = df[df['group'] == 'treatment']['converted'].mean() -  df[df['group'] == 'con
         act_diff

Out[27]: -0.0014795997940775518

In [28]: p_diffs = np.array(p_diffs)
         (act_diff < p_diffs).mean()

Out[28]: 0.88490000000000002
```

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

1. The above computed value is P-value.
2. P-value gives the probability of observing our statistic if null hypothesis is true.
3. Based on the p-value we can conclude that thers's conversion advantage with the new page. Old page performed slightly better based on the above results. Hence we fail to reject the null hypothesis.
4. The p-value is very high and we calculated that more than 90% of our sample values lies above the observed difference when compared to p-value to 0.5.

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 [29]: import statsmodels.api as sm
         convert_old = sum(df2.query("group == 'control'")['converted'])
         convert_new = sum(df2.query("group == 'treatment'")['converted'])
         n_old = len(df2.query("group == 'control'"))
         n_new = len(df2.query("group == 'treatment'"))
```

```
/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 [30]: z_score, p_value = sm.stats.proportions_ztest([convert_old, convert_new], [n_old, n_new
         print(z_score, p_value)
```

```
1.31092419842 0.905058312759
```

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.**?

```
In [31]: from scipy.stats import norm

         print(norm.cdf(z_score))
         # Tells us how significant our z-score is

         # for our single-sides test, assumed at 95% confidence level, we calculate:
         print(norm.ppf(1-(0.05)))
         # Tells us what our critical value at 95% confidence is
         # Here, we take the 95% values as specified in PartII.1
```

```
0.905058312759
1.64485362695
```

1. We find that the z-score of 1.31092419842 is less than the critical value of 1.64485362695, So, we accept the null hypothesis.
2. As regards the conversion rates of the old and new pages, we find that old pages are only minutely better than new pages.
3. These values can suggest us that the findings in part J and K are true.

Part III - A regression approach

1. In this final part, you will see that the result you acheived in the previous A/B test can also be acheived 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?

logisitic Regression (beacause of categorical variable)

b. The goal is to use **statsmodels** to fit the regression model you specified in part **a.** to see if there is a significant difference in conversion based on which page a customer receives. However, you first need to create a colun 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 [43]: df['intercept']=1
         df[['control', 'treatment']] = pd.get_dummies(df['group'])
```

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

```
In [44]: import statsmodels.api as sm
         logit = sm.Logit(df['converted'],df[['intercept','treatment']])
```

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

```
In [45]: results = logit.fit()
         results.summary()

Optimization terminated successfully.
         Current function value: 0.366243
         Iterations 6


Out[45]: <class 'statsmodels.iolib.summary.Summary'>
         """
                                  Logit Regression Results
         ==============================================================================
         Dep. Variable:              converted   No. Observations:               294478
         Model:                          Logit   Df Residuals:                   294476
         Method:                           MLE   Df Model:                            1
         Date:                Sat, 26 May 2018   Pseudo R-squ.:              7.093e-06
         Time:                        12:27:12   Log-Likelihood:            -1.0785e+05
         converged:                       True   LL-Null:                   -1.0785e+05
                                                 LLR p-value:                    0.2161
         ==============================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
```

```
--------------------------------------------------------------------------------
intercept     -1.9887       0.008   -248.297       0.000       -2.004       -1.973
treatment     -0.0140       0.011     -1.237       0.216       -0.036        0.008
================================================================================
"""
```

e. What is the p-value associated with **ab_page**? Why does it differ from the value you found in the **Part II**? **Hint**: What are the null and alternative hypotheses associated with your regression model, and how do they compare to the null and alternative hypotheses in the **Part II**?

0.3 Our hypothesis here is :

0.3.1 H0: Pnew = Pold

0.3.2 H1: Pnew != Pold

In part II we had compared with greater than or lesser than operators in our hypothesis for analysis, here we use equal or not equal in our regression mode.

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?

1. This analysis is based on limited number of factors, the other factors that might influence are age, gender, nationality(Country). Including one or more of these factors we might have different results. Here I have mentioned only few factors, there might be other factors also which could have been included.
2. Adding factors might give us a precise solution whether or not to implement a new page as we would have considered most of influencing factors.
3. Whereas I find a disadvantage that is if we add multiple factors there's possibility of Multicollinearity which would affect our results. So these we have to be mindful of the factors and do appropriate 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. You will need to read in the **countries.csv** dataset and merge together your datasets on the approporiate 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 varaibles.** Provide the statistical output as well as a written response to answer this question.

```
In [46]: countries_df = pd.read_csv('./countries.csv')
         countries_df.head()
```

Out[46]:

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

```
In [48]: df_new = countries_df.set_index('user_id').join(df.set_index('user_id'), how='inner')
         df_new.head()
```

Out[48]:

user_id	country	timestamp	group	landing_page	converted	intercept	control	treatment
630000	US	2017-01-19 06:26:06.548941	treatment	new_page	0	1	0	1
630001	US	2017-01-16 03:16:42.560309	treatment	new_page	1	1	0	1
630002	US	2017-01-19 19:20:56.438330	control	old_page	0	1	1	0
630003	US	2017-01-12 10:09:31.510471	treatment	new_page	0	1	0	1
630004	US	2017-01-18 20:23:58.824994	treatment	new_page	0	1	0	1

```
In [49]: df_new['country'].value_counts()
```

```
Out[49]: US    206364
         UK     73419
         CA     14695
         Name: country, dtype: int64
```

```
In [50]: df_new[['CA', 'UK', 'US']] = pd.get_dummies(df_new['country'])
         df_new = df_new.drop(['CA'], axis=1)
         df_new.head()
```

Out[50]:

user_id	country	timestamp	group	landing_page	converted	intercept	control	treatment	UK	US
630000	US	2017-01-19 06:26:06.548941	treatment	new_page	0	1	0	1	0	1
630001	US	2017-01-16 03:16:42.560309	treatment	new_page	1	1	0	1	0	1
630002	US	2017-01-19 19:20:56.438330	control	old_page	0	1	1	0	0	1
630003	US	2017-01-12 10:09:31.510471	treatment	new_page	0	1	0	1	0	1
630004	US	2017-01-18 20:23:58.824994	treatment	new_page	0	1	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 [58]: ### Fit Your Linear Model And Obtain the Results
         df_new['intercept'] = 1
         df_new['UK_ind_treatment'] = df_new['UK']*df_new['treatment']
         logit_mod = sm.Logit(df_new['converted'], df_new[['intercept','treatment', 'US', 'UK',
         results = logit_mod.fit()
         results.summary()

Optimization terminated successfully.
         Current function value: 0.366236
         Iterations 6


Out[58]: <class 'statsmodels.iolib.summary.Summary'>
         """
                                   Logit Regression Results
         ==============================================================================
         Dep. Variable:              converted   No. Observations:               294478
         Model:                          Logit   Df Residuals:                   294473
         Method:                           MLE   Df Model:                            4
         Date:                Sat, 26 May 2018   Pseudo R-squ.:              2.461e-05
         Time:                        12:35:23   Log-Likelihood:            -1.0785e+05
         converged:                       True   LL-Null:                   -1.0785e+05
                                                 LLR p-value:                    0.2570
         ===============================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
         -------------------------------------------------------------------------------
         intercept          -2.0213      0.027    -76.017      0.000      -2.073      -1.969
         treatment          -0.0212      0.013     -1.614      0.107      -0.047       0.005
         US                  0.0357      0.027      1.338      0.181      -0.017       0.088
         UK                  0.0307      0.031      0.990      0.322      -0.030       0.092
         UK_ind_treatment    0.0285      0.026      1.090      0.276      -0.023       0.080
         ===============================================================================
         """

In [53]: np.exp(results.params)

Out[53]: intercept    0.132005
         treatment    0.986090
         US           1.036322
         UK           1.045968
         dtype: float64

In [54]: 1/_

Out[54]: 0.00010001000100010001

In [55]: df.groupby('group').mean()['converted']
```

```
Out[55]: group
         control      0.120399
         treatment    0.118920
         Name: converted, dtype: float64
```

Conclusion for Logistic Regression including Country factor:

1. As this is logistic regression, we find that values do not show substantial difference in the conversion rates for control and treatment group.
2. Hence there is no strong reason for us to reject null Hypothesis. Which implies that we should accept the Null Hypothesis that is keep the existing page(old page) and try adding something to new page and perform the test again to check if we can implement the new page.
3. Including a new factor considering the interaction between an AB page and country also seem there wasn't statistical significance. Hence clearly we should accept null hypothesis

0.4 Overall Conclusions:

1. The performance of old page was better than new page by a very minute difference based on different techniques we used.
2. This values suggested as to accept null hypothesis by rejecting the alternate hypothesis.
3. The intutions we have given here purely based on the data what we had, variables were limited only to few number. Therefore the results might not be the same when the other factors are added. So insights might well change by additional parameters.

Finishing Up

Congratulations! You have reached the end of the A/B Test Results project! This is the final project in Term 1. 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.

0.5 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 [59]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])
```

```
Out[59]: 0
```