Analyze_ab_test_results_notebook

September 25, 2018

0.1 Analyze A/B Test Results

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[0]
```

```
Out[3]: 294478
```

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 line up.

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

```
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** 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**.

Since we can't know what the true data are, we will handle these rows by removing them.

```
In [8]: # Get indices for rows to be removed
        indices = df.query('group != "treatment" and landing_page == "new_page" or group == "t

        # Drop rows
        df2 = df.drop(indices)

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].sh

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['user_id'][df2.duplicated(subset='user_id') == True]

Out[11]: 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 = df2.drop(2893)
```

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[df2['landing_page'] == 'new_page'].shape[0] / df2.shape[0]

Out[17]: 0.5000619442226688
```

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

From the results above, we do not have sufficient evidence to say that the new treatment page leads to more conversions. If anything, we have some evidence that the control group has a higher conversion than the treatment group, but we do not know if this difference can be attributed to a real differences or chance.

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.

$$H_0 : p_{new} - p_{old} \leq 0$$

$$H_A : p_{new} - p_{old} > 0$$

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.

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.query('landing_page == "new_page"').shape[0]
         n_new

Out[20]: 145310
```

d. What is n_{old}?

```
In [21]: n_old = df2.query('landing_page == "old_page"').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(2, n_new, p=[1-p_new, p_new])
```

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(2, n_old, p=[1-p_old, p_old])
```

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]: -0.0006629510105762432
```

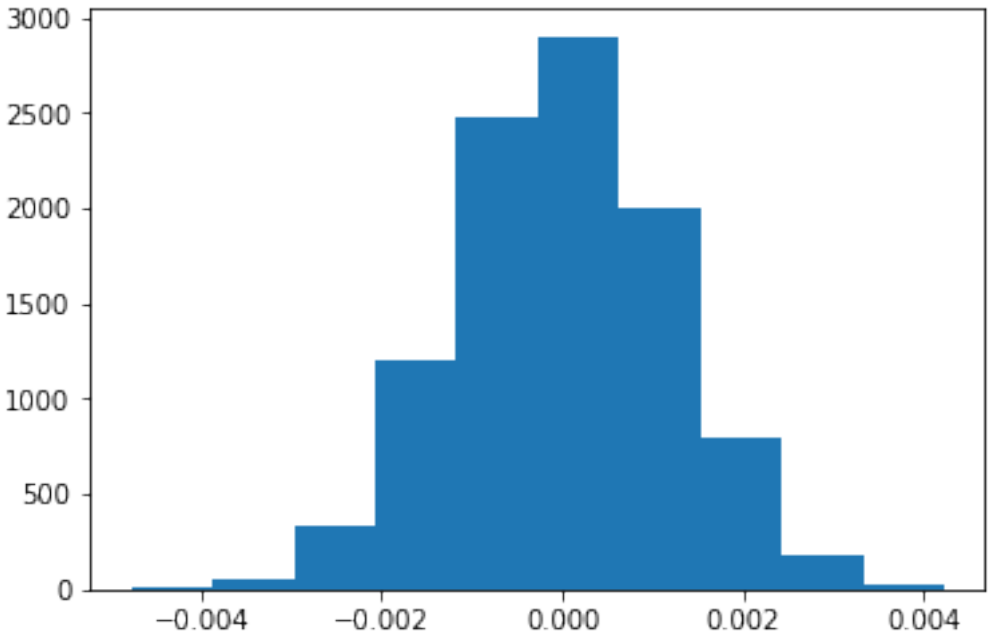
h. Simulate 10,000 p_{new} - p_{old} values using this same process similarly to the one you calculated in parts **a. through g.** above. Store all 10,000 values in a numpy array called **p_diffs**.

```
In [25]: p_diffs = []
         for _ in range(10000):
             new_page_converted = np.random.choice(2, n_new, p=[1-p_new, p_new])
             old_page_converted = np.random.choice(2, n_old, p=[1-p_old, p_old])
             p_diffs.append(new_page_converted.mean() - old_page_converted.mean())

         p_diffs = np.array(p_diffs)
```

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]: obs_diff = df2.query('landing_page == "new_page"').converted.mean() - df2.query('landi
         obs_diff

Out[27]: -0.0015782389853555567

In [28]: (p_diffs > obs_diff).mean()

Out[28]: 0.9096
```

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?

I computed the probability of finding the observed value (obs_diff) when the null hypothesis is true. THis is known as the p-value. A p-value of 0.91 means that we cannot reject the null hypothesis because there is a 91% probability of finding the observed value (or a more extreme value) if the null hypothesis is true. By not rejecting the null hypothesis, we say that there is no evidence that there is a true difference between 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 [29]: 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]
```

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]: sm.stats.proportions_ztest([convert_old, convert_new], [n_old, n_new])

Out[30]: (1.3109241984234394, 0.18988337448195103)
```

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 above are consistent with the findings in parts j. and k. Namely, the p-value of 0.997 leads us to not reject the null hypothesis, which means we have no evidence to indicate that the proportion of conversion in the new page is different from the proporsion of conversion in the old page.

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?

I would be performing a logistical 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 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 [31]: # Add intercept column
         df2['intercept'] = 1

         # Add dummy variable column for treatment
         df2[['ab_page', 'old']] = pd.get_dummies(df2['landing_page'])
         df2.drop('old', axis=1, inplace=True)

         df2.head()
```

```
Out[31]:    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 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 [32]: log_reg_mod = sm.Logit(df2['converted'], df2[['intercept', 'ab_page']])
         results = log_reg_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 [33]: results.summary()
```

```
Out[33]: <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:                Tue, 25 Sep 2018   Pseudo R-squ.:               8.077e-06
Time:                        14:44:31   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 the **Part II**?

The null hypothesis in the regression model is that the coefficient is equal to zero, essentially meaning that there is no relationship between the x variable and the y variable. The alternative hypothesis is that the coefficient is not zero. As mentioned in part II, the null hypothesis in that case was that the difference between proportions for the two groups (experiment and control) was zero.

While they both imply similar things (that the page you view has no impact on conversion), we can see above that the null hypothesis are actually different, so we can expect p-values to be different as well.

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?

Our model has a very low pseudo R-squared (8.077e-06) and the ab_page coeffiecient has a p-value that is not statistically significant. So basically, our current model is not good at all at explaining whether or not an individual converts. Therefore, it would be useful to consider other factors.

There are a couple of issues to keep in mind if we add more terms to the model. For starters, if we add too many of them, we might get some statistically significant p-values by chance alone. Secondly, we might start including terms that exhibit collinearity, so we must be careful not to add too many and which ones we add.

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 variables.** Provide the statistical output as well as a written response to answer this question.

In [34]:
```
countries_df = pd.read_csv('./countries.csv')
df_new = countries_df.set_index('user_id').join(df2.set_index('user_id'), how='inner')
```

In [35]:
```
df_new = pd.get_dummies(df_new, columns=['country'])
df_new.head()
```

Out[35]:

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

In [36]:
```
log_reg_mod = sm.Logit(df_new['converted'], df_new[['intercept', 'ab_page', 'country_C
results = log_reg_mod.fit()
results.summary()
```

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

Out[36]:

```
<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:                Tue, 25 Sep 2018   Pseudo R-squ.:              2.323e-05
Time:                        14:44:43   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
country_CA    -0.0408      0.027     -1.516      0.130      -0.093       0.012
country_UK     0.0099      0.013      0.743      0.457      -0.016       0.036
```

```
==============================================================================
"""
```

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 [37]:
```
# Create interaction variables
df_new['ab*CA'], df_new['ab*UK'] = df_new['ab_page']*df_new['country_CA'], df_new['ab
df_new.head()
```

Out[37]:

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

In [38]:
```
log_reg_mod = sm.Logit(df_new['converted'], df_new[['intercept', 'ab_page', 'country_
results = log_reg_mod.fit()
results.summary()
```

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

Out[38]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:              converted   No. Observations:               290584
Model:                          Logit   Df Residuals:                   290578
Method:                           MLE   Df Model:                            5
Date:                Tue, 25 Sep 2018   Pseudo R-squ.:               3.482e-05
Time:                        14:45:08   Log-Likelihood:            -1.0639e+05
converged:                       True   LL-Null:                   -1.0639e+05
                                        LLR p-value:                    0.1920
```

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -1.9865      0.010   -206.344      0.000      -2.005      -1.968
ab_page       -0.0206      0.014     -1.505      0.132      -0.047       0.006
country_CA    -0.0175      0.038     -0.465      0.642      -0.091       0.056
country_UK    -0.0057      0.019     -0.306      0.760      -0.043       0.031
ab*CA         -0.0469      0.054     -0.872      0.383      -0.152       0.059
ab*UK          0.0314      0.027      1.181      0.238      -0.021       0.084
==============================================================================
"""
```

Conclusions

- There is no evidence that the new webpage leads to increased conversion rates when compared to the old page, when looking at overall data (global).
- There is no evidence that the new webpage leads to higher conversion rates than the old page, for any particular country (CA, UK, US).
- If we are able to get more data (additional terms to include in our model), we might be able to find a subset of users for which the new page is better.
- In the meantime, I recommend not using the new page. We have no evidence that it is better than the old one and we would probably have to invest resources in updating the webpage.