Analyze_ab_test_results_notebook

February 12, 2019

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

- Section **??**
- Section **??**
- Section **??**
- Section **??**

### 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 [59]: 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 [60]: df = pd.read_csv('ab_data.csv')
         df.head()
```

```
Out[60]:    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 [61]: df.shape
```

```
Out[61]: (294478, 5)
```

c. The number of unique users in the dataset.

```
In [62]: df['user_id'].nunique()
```

```
Out[62]: 290584
```

d. The proportion of users converted.

```
In [63]: df_convert = df.query('converted=="1"')
         df_convert.count()
```

```
Out[63]: user_id         35237
         timestamp       35237
         group           35237
         landing_page    35237
         converted       35237
         dtype: int64
```

```
In [64]: ratio_convert = (35237 / 290584)*100
         ratio_convert
```

```
Out[64]: 12.126269856564711
```

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

```
In [65]: df_notlineup = df[((df['group'] == 'treatment') & (df['landing_page'] != 'new_page')) |
         df_notlineup.shape
```

```
Out[65]: (3893, 5)
```

f. Do any of the rows have missing values?

```
In [66]: 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
```

```
In [67]: # no value missing
```

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 [68]: df2 = df.drop(df.query("group == 'control' and landing_page == 'new_page'").index)\
                 .drop(df.query("group == 'treatment' and landing_page == 'old_page'").index)
         df2.shape

Out[68]: (290585, 5)
```

```
In [69]: # 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[69]: 0
```

```
In [70]: len(df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) == False

Out[70]: 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 [71]: df2.user_id.nunique()

Out[71]: 290584
```

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

```
In [73]: df2.user_id.value_counts().index[0]
```

```
Out[73]: 773192
```

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

```
In [15]: df2[df2.user_id==773192]

Out[15]:       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 [74]: df2 = df2.drop_duplicates('user_id')

         # df2.drop(df.index[2893], axis=0, inplace=True)

In [75]: df2.shape

Out[75]: (290584, 5)
```

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

Out[76]: 0.11959708724499628
```

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

```
In [77]: pro_convert_control = df2.query("group == 'control'").converted.mean()

In [78]: pro_convert_control

Out[78]: 0.1203863045004612
```

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

```
In [79]: pro_convert_treatment = df2.query("group == 'treatment'").converted.mean()
         pro_convert_treatment

Out[79]: 0.11880806551510564

In [80]: observed_value = pro_convert_treatment - pro_convert_control
         observed_value

Out[80]: -0.0015782389853555567
```

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

```
In [81]: pro_newpage = len(df2.query("landing_page == 'new_page'")) / len(df2)

         pro_newpage

Out[81]: 0.5000619442226688
```

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.

based on the calculated conversion probabilities, we tend to say that there is not enough evidence to suggest that the new web design increases the conversion rate in any way. Moreover, looking at the size of the difference, it seems clear that this result has no practical meaning.

We need to do some statistical analysis to get more evidence to show whether the new design can improve the conversion rate.

### Part II - A/B Test

Notice that because of the time stamp associated with each event, you could technically run a hypothesis test continuously as each observation was observed.

However, then the hard question is do you stop as soon as one page is considered significantly better than another or does it need to happen consistently for a certain amount of time? How long do you run to render a decision that neither page is better than another?

These questions are the difficult parts associated with A/B tests in general.

`1.` For now, consider you need to make the decision just based on all the data provided. If you want to assume that the old page is better unless the new page proves to be definitely better at a Type I error rate of 5%, what should your null and alternative hypotheses be? You can state your hypothesis in terms of words or in terms of p_{old} and p_{new}, which are the converted rates for the old and new pages.

The null and alternative hypotheses will be:

$$H_0 : p_{new} \leq p_{old} \Leftrightarrow H_0 : p_{new} - p_{old} \leq 0$$
$$H_1 : p_{new} > p_{old} \Leftrightarrow H_1 : p_{new} - p_{old} > 0$$

`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.

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 [82]: p_new = df2.converted.mean()
         p_new

Out[82]: 0.11959708724499628
```

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

```
In [83]: p_old = df2.converted.mean()
         p_old
```

```
Out[83]: 0.11959708724499628
```

c. What is n_{new}?

```
In [84]: n_new = len(df2[df2.landing_page == 'new_page'])
         n_new
```

```
Out[84]: 145310
```

d. What is n_{old}?

```
In [85]: n_old = len(df2[df2.landing_page == 'old_page'])
         n_old
```

```
Out[85]: 145274
```

1 Pay attention

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

```
Out[86]: 17272
```

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

```
Out[87]: 17541
```

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

```
In [89]: p_new - p_old
```

```
Out[89]: 0.0
```

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**.

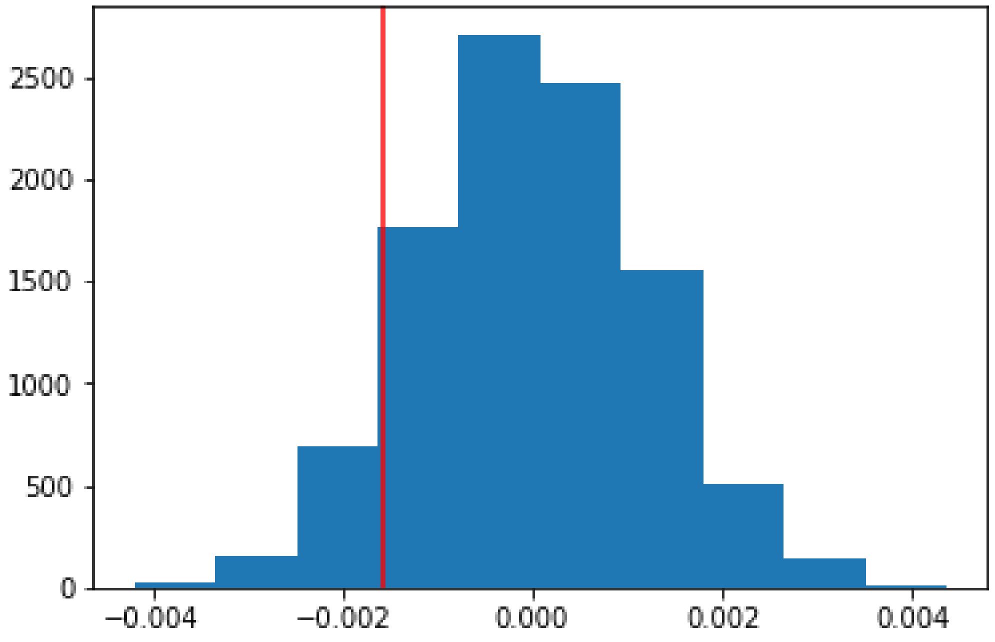
```
In [90]: new_page_converted = np.random.binomial(n_new, p_new, 10000) / n_new
         old_page_converted = np.random.binomial(n_old, p_old, 10000) / n_old

         p_diffs = new_page_converted - old_page_converted
         p_diffs
```

```
Out[90]: array([ 0.00019062,  0.00136749, -0.00111029, ..., -0.00252139,
                 0.00122972,  0.00065198])
```

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 [91]: plt.hist(p_diffs)
         plt.axvline(observed_value, color = "r")
```

```
Out[91]: <matplotlib.lines.Line2D at 0x7f52b4352320>
```



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

```
In [92]: p_value = (p_diffs > observed_value).mean()
         p_value
```

```
Out[92]: 0.90759999999999996
```

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?

we calculate the statistical probability under the null hypothesis. We set the significance level of this hypothesis test to = 0.05. Therefore, any p_value above 0.05 (5%) does not reject the null hypothesis.

The p_value is 0.9075, close to 90%. This indicates that the behavior of our test values (the difference in conversion ratio between old and new pages) is based on the null hypothesis, meaning that the old page is converted better than the new page at a very high probability, or at least equal to the new page. So we failed to reject 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 [93]: import statsmodels.api as sm

         convert_old = df2[(df2['landing_page'] == "old_page") & (df2['converted'] == 1)].shape[
         convert_new = df2[(df2['landing_page'] == "new_page") & (df2['converted'] == 1)].shape[

         n_old = df2[df2['landing_page'] == "old_page"].shape[0]
         n_new = df2[df2['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 [94]: import statsmodels.api as sm

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

Out[94]: (-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 falls inside the critical value of -1.96 - 1.96, and the p-value is 0.9050, almost same as manully computed p_value, all close to 90%. This means the z-score and p-value agree with the findings in parts j and k - there is no significant statistical difference in the conversions of new page and old page, therefore we cannot reject the null hypothesis.

### 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?

Logistic Regression is a better option. Because the outcome (dependent variable) has only a limited number of possible values, in our test, it is either a conversion or no conversion. So we choose the 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 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 [96]: df2['intercept'] = 1

In [97]: df2 = df2.join(pd.get_dummies(df['landing_page']))
         df2 = df2.drop('old_page', axis = 1)

         # df2 = df2.rename({'new_page' : 'ab_page'}, axis = 'columns')

In [98]: df2.columns

Out[98]: Index(['user_id', 'timestamp', 'group', 'landing_page', 'converted',
                'intercept', 'new_page'],
               dtype='object')

In [99]: df2.rename(columns = {'new_page':'ab_page'}, inplace = True)

In [100]: df2.columns

Out[100]: Index(['user_id', 'timestamp', 'group', 'landing_page', 'converted',
                 'intercept', 'ab_page'],
                dtype='object')

In [101]: pd.__version__

Out[101]: '0.20.3'
```

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 [102]: model = sm.Logit(endog = df2['converted'], exog = df2[['intercept', 'ab_page']])
          res = model.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 [103]: print(res.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:              converted   No. Observations:               290584
Model:                          Logit   Df Residuals:                   290582
Method:                           MLE   Df Model:                            1
Date:                Tue, 12 Feb 2019   Pseudo R-squ.:               8.077e-06
Time:                        09:32:35   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 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**?

The p-value associated with ab_page is 0.19. This value is still larger than the Type 1 error rate of 0.05. It means that the landing page is not statistically significant in the prediction of converts.

This p_value is different from the value in Part II. Because the null and alternative hypotheses in two part are different. In the regression model are

$$H0 : p_{new} = p_{old}$$

$$H1 : p_{new}! = p_{old}$$

, which just predicts a difference in the two values.

In the part two, the null and alternative hypotheses is

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

$$H_1 : p_{new} > p_{old}$$

, which want to predict which page gets more conversions.

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?

In general, for regression testing, we want to use as many linear independent predictors as possible with our response variables (conversions), as this will increase the chances that the model will correctly determine which predictors are more likely to convert, and we will be able to find out Which predictors are important in increasing conversion rates.

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

```
In [107]: df_countries.head()
```

```
Out[107]:    user_id country
          0   834778      UK
          1   928468      US
          2   822059      UK
          3   711597      UK
          4   710616      UK
```

```
In [108]: df_new = df_new.join(pd.get_dummies(df_new['country']))
```

```
In [109]: model = sm.Logit(endog = df_new['converted'], exog = df_new[['intercept', 'ab_page', '
          res = model.fit()
```

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

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 [110]: print(res.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:              converted   No. Observations:               290584
Model:                          Logit   Df Residuals:                   290580
Method:                           MLE   Df Model:                            3
Date:                Tue, 12 Feb 2019   Pseudo R-squ.:               2.323e-05
Time:                        09:32:54   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
==============================================================================
```

By adding the catergorical predictor `country`, we are trying to figure out whether it is statistically significant when our users come from the US, UK or Canada.

Base on the large p-values of 0.13 and 0.457, we can understand that, regardless of whether our users come from the baseline category (US) or any of the other possible countriesCA,UK, the conversion rate for these users will not increase much.

## 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.

1.1 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 [111]: from subprocess import call
          call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])

Out[111]: 0

In [ ]:
```