ganesh_analyze_ab_test

August 21, 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

### 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 cell below to find the number of rows in the dataset.

```
In [3]: total_users = df.shape[0]
        total_users
```

```
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]: users_converted=float(df.query('converted == 1')['user_id'].nunique())
        p1 = (users_converted/total_users)
        print("The proportion of users converted is {0:.2%}".format(p1))
```

```
The proportion of users converted is 11.94%
```

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

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

```
Out[6]: 3893
```

f. Do any of the rows have missing values?

```
In [7]: df.isnull().values.any()
```

```
Out[7]: False
```

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

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

```
In [8]: df2 = df.drop(df.query('(group == "treatment" and landing_page != "new_page") or (group
```

```
In [9]: # Double Check all of the correct rows were removed - this should be 0
        df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) == False].sha
```

```
Out[9]: 0
```

3. Use **df2** and the cells below to answer questions for **Quiz3** in the classroom.

a. How many unique **user_id**s are in **df2**?

```
In [10]: df2['user_id'].nunique()
```

```
Out[10]: 290584
```

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

```
In [11]: df2[df2.duplicated(['user_id'], keep=False)]['user_id']
```

```
Out[11]: 1899    773192
         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(df2[(df2.user_id == 773192) & (df2['timestamp'] == '2017-01-09 05:37:58.
         df2[df2['user_id'] == 773192]
```

```
Out[13]:       user_id                   timestamp      group landing_page  converted
         2893   773192  2017-01-14 02:55:59.590927  treatment     new_page          0
```

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

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

```
In [14]: converted_users2 = float(df2.query('converted == 1')['user_id'].nunique())
         p2 = converted_users2/float(df2.shape[0])
         print("The probability of an individual converting regardless of the page they receive
```

```
The probability of an individual converting regardless of the page they receive is 11.96%
```

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

```
In [15]: converted_controlusers2 = float(df2.query('converted == 1 and group == "control"')['use
         control_users2 =float(df2.query('group == "control"')['user_id'].nunique())
         cp2 = converted_controlusers2 /control_users2
         print(" Given that an individual was in the control group, the probability they convert
```

```
 Given that an individual was in the control group, the probability they converted is 12.04%
```

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

```
In [16]: converted_controlusers2 = float(df2.query('converted == 1 and group == "treatment"')['u
         treat_users2 =float(df2.query('group == "treatment"')['user_id'].nunique())
         tp2 = converted_controlusers2 /treat_users2
         print(" Given that an individual was in the treatment group, the probability they conve
```

```
 Given that an individual was in the treatment group, the probability they converted is 11.88%
```

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

```
In [17]: new_page_users2 = float(df2.query('landing_page == "new_page"')['user_id'].nunique())
         Newpage_p2 = new_page_users2/float(df2.shape[0])
         print("The probability that an individual received the new page is {0:.2%}".format(Newp
```

```
The probability that an individual received the new page is 50.01%
```

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.

The probability of an individual converting regardless of the page they receive is 11.96%, Given that an individual was in the control group, the probability they converted is 12.04% Given that an individual was in the treatment group, the probability they converted is 11.88%. The probablity users converted in both control and treatment group are pretty similar to each other and probability of an individual converting regardless of the page they receive. therefore, there is no evidence that ne page leads to more conversions.

### 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 hypothese is H0: p_new - p_old <= 0
Alternative hypothese is H1: 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 **conversion rate** for p_{new} under the null?

```
In [18]: p_new = round(float(df2.query('converted == 1')['user_id'].nunique())/float(df2['user_i
         p_new
```

```
Out[18]: 0.1196
```

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

```
In [19]: p_old = round(float(df2.query('converted == 1')['user_id'].nunique())/float(df2['user_i
         p_old
```

```
Out[19]: 0.1196
```

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

```
In [20]: N_new = df2.query('landing_page == "new_page"')['user_id'].nunique()
         N_new
```

```
Out[20]: 145310
```

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

```
In [21]: N_old = df2.query('landing_page == "old_page"')['user_id'].nunique()
         N_old
```

```
Out[21]: 145274
```

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

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

```
Out[22]: array([1, 1, 1, ..., 1, 1, 0])
```

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

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

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

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]: 0.00062840538668740287
```

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 [25]: import timeit
         start = timeit.default_timer()

         # Create sampling distribution for difference in completion rates
         # with boostrapping
         p_diffs = []
         size = df.shape[0]
         for _ in range(10000):
             samp = df2.sample(size, replace = True)
             new_page_converted = np.random.choice([0,1],N_new, p=(p_new,1-p_new))
             old_page_converted = np.random.choice([0,1],N_old, p=(p_old,1-p_old))
             p_diffs.append(new_page_converted.mean() - old_page_converted.mean())

         #Compute python running time.
         stop = timeit.default_timer()
         print (stop - start)

         p_diffs = np.array(p_diffs)
```

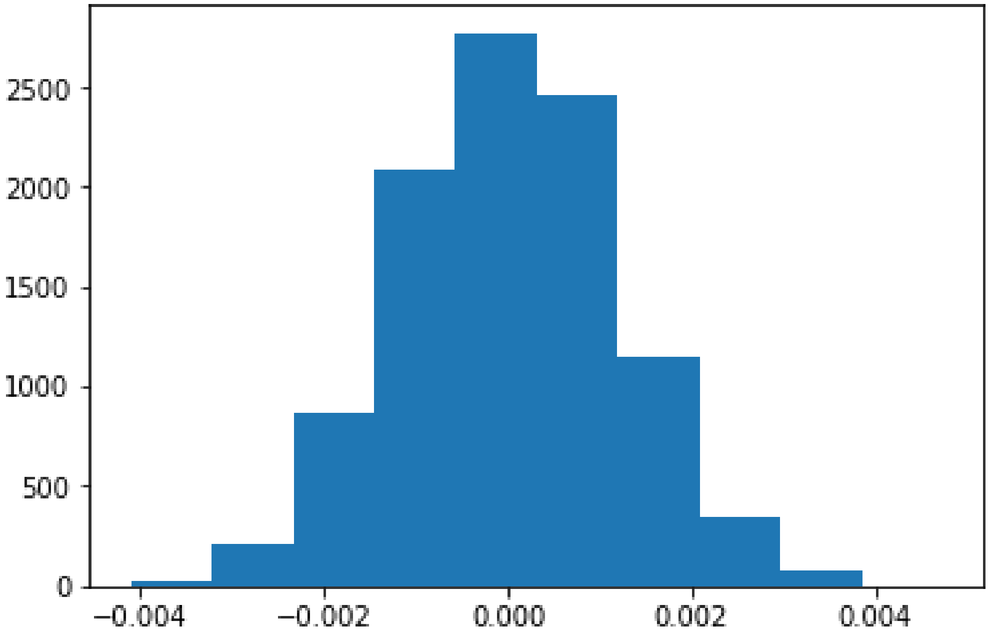
```
824.111282871
```

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)

         convert_new = df2.query('converted == 1 and landing_page == "new_page"')['user_id'].nun
         convert_old = df2.query('converted == 1 and landing_page == "old_page"')['user_id'].nun

         actual_cvt_new = float(convert_new)/ float(n_new)
```

```
actual_cvt_old = float(convert_old)/ float(n_old)

obs_diff = actual_cvt_new - actual_cvt_old

# Display observed difference in converted rate
obs_diff
```

```
---------------------------------------------------------------------------

NameError                                 Traceback (most recent call last)

<ipython-input-26-34738f68ac23> in <module>()
  4 convert_old = df2.query('converted == 1 and landing_page == "old_page"')['user_id'].
  5
----> 6 actual_cvt_new = float(convert_new)/ float(n_new)
  7 actual_cvt_old = float(convert_old)/ float(n_old)
  8


NameError: name 'n_new' is not defined
```



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

```
In [ ]: null_vals = np.random.normal(0, p_diffs.std(), p_diffs.size)


        #Plot Null distribution
        plt.hist(null_vals)
        #Plot vertical line for observed statistic
        plt.axvline(x=obs_diff,color ='red')


        #Compute proportion of the p_diffs are greater than the actual difference observed in ab
        (null_vals > obs_diff).mean()
```

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?

Type I error rate of 5%, and Pold > Alpha, we fail to reject the null. Therefore, the data show, with a type I error rate of 0.05, that the old page has higher probablity of convert rate than new page.

P-Value: The probability of observing our statistic or a more extreme statistic from 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.

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 [ ]: import statsmodels.api as sm


        convert_old = sum(old_df.converted)
        convert_new = sum(new_df.converted)
        n_old = len(old_df)
        n_new = len(new_df)
```

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 [ ]: z_score, p_value = sm.stats.proportions_ztest(np.array([convert_new,convert_old]
        z_score, p_value
```

```
# it's a one tail test so a z-score past 1.96 will be significant.),np.array([n_new,n_ol
from scipy.stats import norm

norm.cdf(z_score)
# 0.09494168724097551 4 # Tells us how significant our z-score is
norm.ppf(1-(0.05/2))
# 1.959963984540054 # Tells us what our critical value at 95% confidence is
```

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.**?

Since the z-score of 1.3109241984234394 does not exceed the critical value of 1.959963984540054, we fail to reject the null hypothesis that old page users has a better or equal converted rate than old page users.

Therefore, the converted rate for new page and old page have no difference. This result is the same as parts J. and K. result.

### 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?

Put your answer here.

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 [ ]: df2['intercept'] = 1


        #create a dummy variable column for which page each user received
        df2= df2.join(pd.get_dummies(df2['landing_page']))

        df2['ab_page'] = pd.get_dummies(df['group']) ['treatment']
        df2.head()
```

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 [ ]: lo = sm.Logit(df2['converted'], df2[['intercept','ab_page']])
        result = lo.fit()
```

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

```
In [ ]: print result.summary()
```

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**?

The p-value associated with ab_page is 0.190. The null in c-e part is that there is no difference between the treatment and control group. Alternative hypotheses is that there is difference between between the treatment and control group

Part II assumes the old page is better unless the new page proves to be definitely better at a Type I error rate of 5%, compared to question c-e,they have different explainory varibale or factor for the result.

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?

Other factor can be the time(timestamp variable). We can check if the converted rate depends on certain time of the day or certain day when user browerse the website.

For timestamp variable, we can further convert time as categorical variable which includes "Morning, afternoon, and evening", or "weekday and weekend".

Disadavantage for adding additional terms into regression model is that it will make interpretate the model more complex and also, if new terms are dependable variable with the exisiting explanatory term, we need to add higher order term to help predict the result better.

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 [ ]: c = pd.read_csv('countries.csv')
        c.head()

        #Join ab dataset with country dataset
        df3 = df2.merge(c, on ='user_id', how='left')
        df3.head()

        c['country'].unique()


        df3[['CA','UK','US']] = pd.get_dummies(df3['country'])
        df3 = df3.drop(df3['CA'])


        #Create intercept variable
```

```
df3['intercept'] = 1

#Create Logit regression model for conveted and country, and us CA and old page as basel
logit3 = sm.Logit(df3['converted'], df3[['intercept','new_page','UK','US']])
result = logit3.fit()
result.summary()

1/np.exp(-0.0150),np.exp(0.0506),np.exp(0.0408)


Interpreting Result:

For every unit for new_page decrease, convert will be 1.5% more likely to happen, holdin

For every unit for UK increases, convert is 5.2% more to happen, holding all other varib

For every unit for US increases, convert is 4.2% more to happen, holding all other varib
```

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 [ ]: # What is  the p-value for each country? what is the p-critical, does the coefficient ca
        # Create a new intereacton variable between new page and country US and UK
        df3['UK_new_page'] = df3['new_page']* df3['UK']
        df3['US_new_page'] = df3['new_page']* df3['US']

        #Create logistic regression for the intereaction variable between new page and country u
        logit4 = sm.Logit(df3['converted'], df3[['intercept','new_page','UK_new_page','US_new_pa
        result4 = logit4.fit()
        result4.summary()

        #exponentiated the CV to inteprete the result
        np.exp(result4.params)

        Interpreting Result:

        From the above Logit Regression Results, we can see the coefficient of intereaction vari

        Also,only intercept's p-value is less than 0.05, which is statistically significant enou
        Additionally, Z-score for all X variables are not large enough to be significant for pre

        Therefore, the country a user lives is not significant on the converted rate considering

        For every unit for new_page decreases, convert will be 7.0% more likely to happen, holdi
```

```
Convert is 1.08 times more likely to happen for UK and new page users than CA and new pa

Convert is 1.04 times more likely to happen for US and new page users than CA and new pa

Convert is 1.18 % more likely to happen for the users in UK than CA, holding all other v

Convert is 1.76 % more likely to happen for the users in US than CA, holding all other v
```

```
In [ ]: df3['UK_new_page'] = df3['new_page']* df3['UK']
        df3['US_new_page'] = df3['new_page']* df3['US']
```

```
In [ ]: logit4 = sm.Logit(df3['converted'], df3[['intercept','new_page','UK_new_page','US_new_pa
        result4 = logit4.fit()
        result4.summary()
```

```
In [ ]: np.exp(result4.params)
```

## Finishing Up

Congratulations! You have reached the end of the A/B Test Results project! 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.

0.3 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 [ ]: #Import sklearn model to split, test and score data,and fit data model
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import confusion_matrix, precision_score, recall_score, accuracy_sc
        from sklearn.model_selection import train_test_split
```

```
In [ ]: x = df3[['new_page','UK_new_page','US_new_page','UK','US']]
        y = df3['converted']
```

```
In [ ]: X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=0)
```

```
In [ ]:
```

```
In [ ]: lm = LinearRegression()
```

```
In [ ]: lm.fit(X_train,y_train) # fit the train data
```

```
In [ ]: print(lm.score(X_test,y_test))
```

```
In [ ]: # The score result is very low, which mean the page and country dataset are not a good f
```

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