Analyze_ab_test_results_notebook

September 7, 2019

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.

Part I - Probability

To get started, let's import our libraries.

```
In [57]: 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 [58]: df = pd.read_csv('ab_data.csv')
         df.head()
```

```
Out[58]:    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 [59]: df.shape[0]

Out[59]: 294478
```

c. The number of unique users in the dataset.

```
In [60]: df["user_id"].nunique()

Out[60]: 290584
```

d. The proportion of users converted.

```
In [5]: df['converted'].mean() * 100

Out[5]: (35237, 5)
```

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

```
In [61]: mismatch_g1 = df.query('group == "treatment" and landing_page == "old_page"')
         len(mismatch_g1)

Out[61]: 1965

In [62]: mismatch_g2 = df.query("group == 'control' and landing_page == 'new_page'")
         len(mismatch_g2)

Out[62]: 1928

In [63]: len(mismatch_g1) + len(mismatch_g2)

Out[63]: 3893
```

f. Do any of the rows have missing values?

```
In [7]: df[df.isnull()].count()

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

```
In [64]: 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** 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 [68]: df.drop(df.query("group =='treatment' and  landing_page =='old_page'").index,inplace=Tr
         df.drop(df.query("group =='control' and  landing_page =='new_page'").index,inplace=True

In [69]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 290585 entries, 0 to 294477
Data columns (total 5 columns):
user_id         290585 non-null int64
timestamp       290585 non-null object
group           290585 non-null object
landing_page    290585 non-null object
converted       290585 non-null int64
dtypes: int64(2), object(3)
memory usage: 13.3+ MB


In [70]: df.to_csv('ab_edited.csv',index=False)

In [71]: df2 = pd.read_csv('ab_edited.csv')

In [72]: # 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[72]: 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 [73]: len(df2['user_id'].unique())
```

```
Out[73]: 290584
```

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

```
In [74]: sum(df2['user_id'].duplicated())
```

```
Out[74]: 1
```

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

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

```
Out[11]:       user_id                   timestamp      group landing_page  converted
         1876   773192  2017-01-09 05:37:58.781806  treatment     new_page          0
         2862   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 [12]: tim_dup = "2017-01-09 05:37:58.781806"
         df2 = df2[df2.timestamp != tim_dup]
```

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 [13]: df['converted'].mean()
```

```
Out[13]: 0.11959667567149027
```

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

```
In [14]: df_grp = df.groupby('group')
         df_grp.describe()
         #Thus, given that an individual was in the control group, the probability they converte
```

```
Out[14]:           converted                                                 user_id  \
                       count      mean       std  min  25%  50%  75%  max     count
         group
         control     145274.0  0.120386  0.325414  0.0  0.0  0.0  0.0  1.0  145274.0
         treatment   145311.0  0.118807  0.323563  0.0  0.0  0.0  0.0  1.0  145311.0

                                                                                \
                             mean           std       min       25%       50%
         group
         control     788164.072594  91287.914601  630002.0  709279.5  788128.5
         treatment   787845.618446  91161.258854  630000.0  708746.5  787874.0


                          75%       max
         group
         control     867208.25  945998.0
         treatment   866718.50  945999.0
```

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

```
In [ ]: Thus, given that an individual was in the treatment group, the probability they converte
```

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

```
In [75]: new_users = len(df.query("group == 'treatmemt'"))
         users = df.shape[0]
         new_users_p = new_users/users

         print(new_users_p)
         print(new_users)
         print(users)
```

```
0.0
0
290585
```

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.

Evidence that one page leads to more conversions? - given that an individual was in the treatment group, the probability they have converted is 0.118807 - given that an individual was in the control group, the probability they have converted is 0.120386 - we are able to find that old page does better , but by a very small margin. - changed aversion, the test span duration and other potentially influencing factors have not been acoounted for. So,we cannot state that one page leads to more conversions.This is very important as both pages show similar performance

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.

Hypothesis H0 : pnew <= pold 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 **conversion rate** for p_{new} under the null?

```
In [16]: p_new = df2['converted'].mean()
         print(p_new)
```

```
0.119597087245
```

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

```
In [17]: p_old = df2['converted'].mean()
         print(p_old)
```

```
0.119597087245
```

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

```
In [18]: n_new = len(df2.query("group =='treatment'"))
         print(n_new)
```

```
145310
```

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

```
In [19]: n_old = len(df2.query("group=='control'"))
         print(n_old)
```

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

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

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

```
In [22]: new_page_converted=new_page_converted[:145274]
         p_diff = new_page_converted/n_new - old_page_converted/n_old
         print(p_diff)
```

```
[ -6.88354420e-06   0.00000000e+00   0.00000000e+00 ...,   0.00000000e+00
   0.00000000e+00   6.88183883e-06]
```

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 [ ]: p_diffs = []

        for _ in range(10000):
            new_page_converted = np.random.choice([1, 0], size=n_new, p=[p_new, (1-p_new)]).mean
            old_page_converted = np.random.choice([1, 0], size=n_old, p=[p_old, (1-p_old)]).mean
            diff = new_page_converted - old_page_converted
            p_diffs.append(diff)
        print (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 [ ]: 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 [ ]: act_diff = df[df['group'] == 'treatment']['converted'].mean() -  df[df['group'] == 'cont
        act_diff
        p_diffs = np.array(p_diffs)
        p_diffs
```

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?

we are computing p values here This is the probability of observing our statistic,if the null hypothesis is true or not The most extreme in favor of the alternative portion of this statement determines the shading associated with your p-value we find that there is no conversion advantage in the new page.So we can conclude that null hypothesis is true as old and new perform almolst the same.As the number shows the old page performed slightly better

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 [33]: import statsmodels.api as sm
         from pandas.core import datetools

         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'"))
```

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

```
1.31092419842 0.189883374482
```

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.**?

from scipy.stats import norm print(norm.cdf(z_score)) print(norm.ppf(1-(0.05))) #Tells us what our critical value at 95% confidence is

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?

Logistic Regression

b. The goal is to use **statsmodels** to fit the regression model you specified in part **a.** to see if there is a significant difference in conversion based on which page a customer receives. However, you first need to create in df2 a column for the intercept, and create a dummy variable column for which page each user received. Add an **intercept** column, as well as an **ab_page** column, which is 1 when an individual receives the **treatment** and 0 if **control**.

```
In [42]: df['intercept']=1
         df[['control', 'treatment']] = pd.get_dummies(df['group'])
```

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 [39]: 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 [40]: results = logit.fit()
         results.summary()

Optimization terminated successfully.
         Current function value: 0.366118
         Iterations 6


Out[40]: <class 'statsmodels.iolib.summary.Summary'>
         """
                                  Logit Regression Results
         ==============================================================================
         Dep. Variable:              converted   No. Observations:               290585
         Model:                          Logit   Df Residuals:                   290583
         Method:                           MLE   Df Model:                            1
         Date:                Sat, 07 Sep 2019   Pseudo R-squ.:               8.085e-06
         Time:                        18:48:59   Log-Likelihood:            -1.0639e+05
         converged:                       True   LL-Null:                   -1.0639e+05
                                                 LLR p-value:                    0.1897
         ==============================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
         ------------------------------------------------------------------------------
         intercept     -1.9888      0.008   -246.669      0.000      -2.005      -1.973
         treatment     -0.0150      0.011     -1.312      0.190      -0.037       0.007
         ==============================================================================
         """
```

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

Answer: -Ourhypothesishereis: - H0 : pnew - pold =0- H1 : pnew - pold !=0

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?

We should consider other factors into the regression model as they might influence the conversions too. For instance student segments [new v/s returning candidates] might create change aversion or even, the opposite as a predisposition to conversion. Seasonality like new terms or New years might mean more interest in new skills/ resolutions. Timestamps are inl- cuded but without regionality, they do not indicate if seasonality was a factor or not. [as different countries follow different term and weather patterns. - Factors like device on which tests were taken or course which was looked at, prior academic background, age, might alter experience and ultimately, conversions. These are limitations which should be at least kept in mind while making the final decision. - The disadvantages to adding additional terms into the regression model is that even with additional factors we can never account for all influencing factors or accomodate them. Plus, small pilots and pivots sometimes work better in practice than long-drawn research without execution.

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 [51]: countries_df = pd.read_csv('./countries.csv')
         countries_df.head()

         df_new = countries_df.set_index('user_id').join(df2.set_index('user_id'), how='inner')
         df_new.head()
         df_new['country'].value_counts()

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

         df_new['country'].astype(str).value_counts()

Out[51]: US    203619
         UK     72466
         CA     14499
         Name: country, dtype: int64
```

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 [55]: ### Fit Your Linear Model And Obtain the Results
         df['intercept'] = 1
         log_mod = sm.Logit(df_new['converted'], df_new[['CA', 'US']])
         results = log_mod.fit()
         results.summary()

         np.exp(results.params)
         1/_
         df.groupby('group').mean()['converted']

Optimization terminated successfully.
         Current function value: 0.447174
         Iterations 6


Out[55]: group
         control      0.120386
         treatment    0.118807
         Name: converted, dtype: float64
```

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 [56]:
```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])
```

Out[56]: 0

In [ ]:
```
##Conclusions from Regression:
As in this logistic regression model too, we find that the values do not show a substan
This indicates that we can acceot the Null Hypothesis and keep the existing page as is.
## Conclusions
The performance of the old page was found better (by miniscule values only) as computed
Hence, we accept the Null Hypothesis and Reject the Alternate Hypothesis.
These inferences are strictly based on data on hand. This analysis acknowledges its lim
tions due to factors not included in the data.
```