Analyze_ab_test_results_notebook

January 10, 2019

0.1 Analyze A/B Test Results

This project is to assure I 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!

### Introduction

A/B tests are very commonly performed by data analysts and data scientists. It is important that I get some practice working with the difficulties of these

For this project, I will be working to understand the results of an A/B test run by an e-commerce website. My 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 I 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 I am on the right track as I work through the project, and I can feel more confident in my final submission meeting the criteria. As a final check, assure I meet all the criteria on the RUBRIC.

#### Part I - Probability

To get started, let's import our libraries.

```
In [139]: 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 [140]: df = pd.read_csv('ab_data.csv')
          df.head(5)

Out[140]:    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 [141]: 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
```

c. The number of unique users in the dataset.

```
In [142]: # Reference : https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.n
          df['user_id'].nunique()

Out[142]: 290584
```

d. The proportion of users converted.

```
In [143]: # Get the mean of users that converted
          df['converted'].mean()

Out[143]: 0.11965919355605512
```

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

```
In [144]: ab_df = df.query("(group == 'control' and landing_page == 'new_page') or (group == '
          len(ab_df)

Out[144]: 3893
```

f. Do any of the rows have missing values?

```
In [145]: #Get the number of NaN values in the dataset
          df.isnull().sum()

Out[145]: user_id         0
          timestamp       0
          group           0
          landing_page    0
          converted       0
          dtype: int64
```

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

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

```
In [146]: df2 = df.query("(group == 'control' and landing_page == 'old_page') or (group == 'tre
```

```
In [147]: # Double Check all of the correct rows were removed - this should be 0
          df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) == False]
```

```
Out[147]: 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 [148]: df2['user_id'].nunique()
```

```
Out[148]: 290584
```

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

```
In [149]: df2[df2['user_id'].duplicated()]['user_id'].unique()
```

```
Out[149]: array([773192])
```

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

```
In [150]: #Reference : http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.
          df2[df2['user_id'].duplicated(keep = False)]
```

```
Out[150]:       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 [151]: #Remove the first duplicate of user_id = 773192 and Check if the duplicates user_id
          df2 = df2.drop_duplicates(['user_id'] , keep = 'last' )
          df2.query("(user_id == '773192')")
```

```
Out[151]:       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 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 [152]: # Calculate the probability of an individual converting regardless of the page they
          df2['converted'].mean()
```

```
Out[152]: 0.11959708724499628
```

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

```
In [153]: # Calculate the probability of an individual converting given the individual was in
          df2[df2['group'] == 'control']['converted'].mean()
```

```
Out[153]: 0.1203863045004612
```

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

```
In [154]: # Calculate the probability of an individual converting given the individual was in
          df2[df2['group'] == 'treatment']['converted'].mean()
```

```
Out[154]: 0.11880806551510564
```

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

```
In [155]: # Count the number of individual receiving the new page
          count_new_page = df2[df2['landing_page'] == 'new_page']['user_id'].count()

          # Calculate the probability of an individual receiving the new page
          probability_new_page = count_new_page / df2.shape[0]
          probability_new_page
```

```
Out[155]: 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.

There is no sufficient evidence to say that the new treatment page leads to more conversions.

0.1.1 Reasons of insufficient evidence

1. Contrary to what has been expected, the proportion of users that has converted in the control group (0.12) is actually higher than the treatment group (0.1188)

2. The percentage of an individual received the new page is actually 50% which means the A/B test has been fair but the probability of individual has converted is approximately 11.9%

3. There are also other factors that are not accounted for such as change aversion and test span duration.

4. The proportion of users that has converted in the control group and treatment group has a marginal effect which leads to insufficient evidence to reject the null hypothesis.

### 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.

Type 1 error rate = 5%

0.1.2 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 **convert rate** for p_{new} under the null?

```
In [156]: # Calculate the convert rate for pnew under the null
          new_p = df2['converted'].mean()
          new_p
```

```
Out[156]: 0.11959708724499628
```

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

```
In [157]: # Calculate the convert rate for pold under the null
          old_p = df2['converted'].mean()
          old_p
```

```
Out[157]: 0.11959708724499628
```

c. What is n_{new}?

```
In [158]: # Calculate the number of individuals in the treatment group
          new_n = df2[df2['group'] == 'treatment'].shape[0]
          new_n
```

```
Out[158]: 145310
```

d. What is n_{old}?

```
In [159]: # Calculate the number of individuals in the control group
          old_n = df2[df2['group'] == 'control'].shape[0]
          old_n

Out[159]: 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 [160]: # Calculate the  binomial distribution for the converted new page
          new_page_converted = np.random.binomial(new_n , new_p)
          new_page_converted

Out[160]: 17445
```

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 [161]: # Calculate the binomial distribution for the converted old page
          old_page_converted = np.random.binomial(old_n , old_p)
          old_page_converted

Out[161]: 17285
```

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

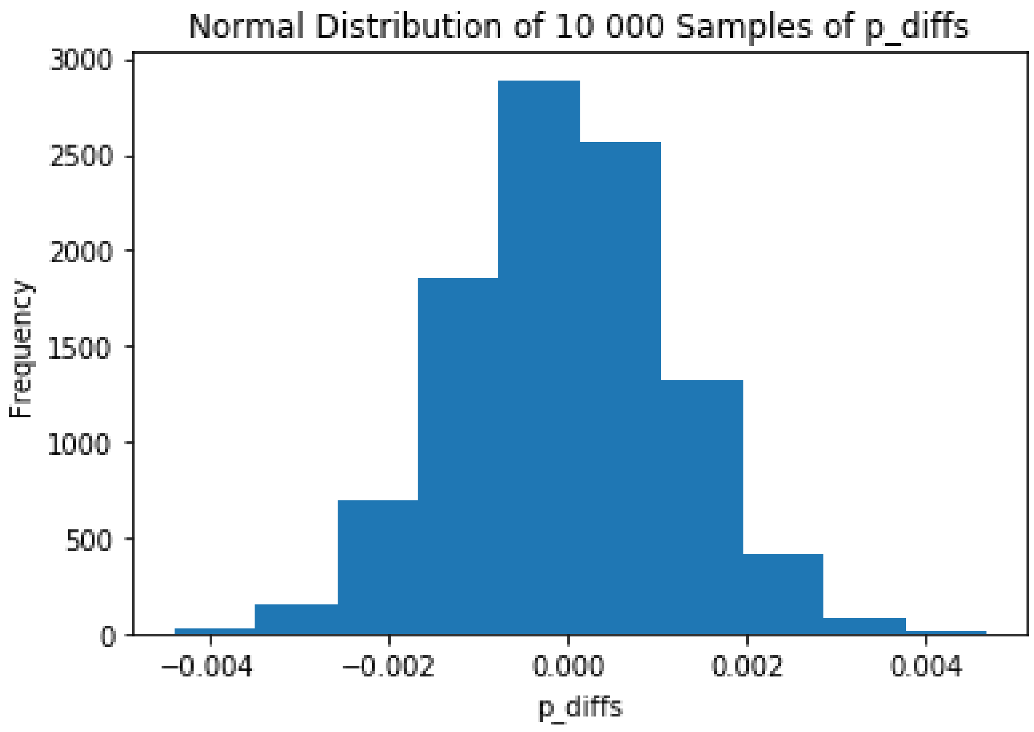
```
In [162]: # Difference between pnew and pold
          difference_probability_page_converted =  (new_page_converted / new_n) - (old_page_co
          difference_probability_page_converted

Out[162]: 0.0010716168590364228
```

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 [163]: # Simulate with 10 000 samples to calculate the difference between pnew and pold
          p_diffs = []
          for _ in range(10000):
              new_page_converted = np.random.binomial(new_n,new_p)
              old_page_converted = np.random.binomial(old_n, old_p)
              diff = new_page_converted/new_n - old_page_converted/old_n
              p_diffs.append(diff)
```

i. Plot a histogram of the **p_diffs**. Does this plot look like what you expected? Use the matching problem in the classroom to assure you fully understand what was computed here.

```
In [167]: plt.hist(p_diffs)
          plt.xlabel('p_diffs')
          plt.ylabel('Frequency')
          plt.title('Normal Distribution of 10 000 Samples of p_diffs');
```



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

```
In [168]: # Calculate the actual difference
          act_diff = df2[df2['group'] == 'treatment']['converted'].mean() - df2[df2['group'] ==
          act_diff

Out[168]: -0.0015782389853555567

In [169]: # Create an array for p_diffs
          p_diffs = np.array(p_diffs)

In [171]: # Calculate the proportion of p_diffs that are greater than our actual diffs observe
          (p_diffs > act_diff).mean()

Out[171]: 0.902
```

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

1. The value in scientific studies is called p-value

2. The p-value is the probability of getting results you did (or more extreme results) given the null hypothesis is true

3. Once you have calculated the proportion of p_diffs that are greater than our actual diffs observed, we found out that due to the high proportion (0.902), there are no comparative advantage of the new page. Hence , we have failed to reject 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.

```
In [172]: import statsmodels.api as sm

          # Find the number of conversions for old_page
          convert_old = df2.query(" landing_page == 'old_page' and converted == 1").shape[0]

          # Find the number of conversions for new_page
          convert_new = df2.query(" landing_page == 'new_page' and converted == 1").shape[0]

          # Find the number of individuals for control group
          n_old = df2[df2['group'] == 'control'].shape[0]

          # Find the number of individuals for treatment group
          n_new = df2[df2['group'] == 'treatment'].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 [174]: # Compute the z-score and p-value
          z_score , p_value = sm.stats.proportions_ztest([convert_old , convert_new], [n_old ,
          z_score , p_value

Out[174]: (1.3109241984234394, 0.9050583127590245)
```

n. What do the z-score and p-value you computed in the previous question mean for the conversion rates of the old and new pages? Do they agree with the findings in parts **j.** and **k.**?

```
In [135]: # Add more comment
          from scipy.stats import norm
          # Tells us how significant our z-score is
          print(norm.cdf(z_score))

          # for our single-sides test, assumed at 95% confidence level, we calculate:
          print(norm.ppf(1-(0.05)))

0.1035291448032774
1.6448536269514722
```

Summary from A/B test

1. The coversion rate of old page is only marginally better than the conversion rate of new page.

2. Z-Score (1.31) is lower than the critical valaue of 1.64. Hence, we have failed to 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?

0.1.3 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 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 [175]: # Create a column for the intercept
          df2['intercept'] = 1

          # Create a dummy variable for which page each user received
          df2[['control' , 'treatment']] = pd.get_dummies(df2['group'])
```

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

```
In [176]: # Use statsmodels to import your regression model
          import statsmodels.api as sm

          # Instantiate the model and fit the model using two columns
          logit = sm.Logit(df2['converted'] , df2[['intercept' , 'treatment']])

          # Predict whether or not an individual converts
          results = logit.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 [178]: # Summary of our regression model
          results.summary()

Out[178]: <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:                Sun, 25 Nov 2018   Pseudo R-squ.:               8.077e-06
          Time:                        16:52:28   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
          treatment     -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**?

H0 : pnew = pold H1 : Pnew != pold

f. Now, you are considering other things that might influence whether or not an individual converts. Discuss why it is a good idea to consider other factors to add into your regression model. Are there any disadvantages to adding additional terms into your regression model?

1. With additional factors , it would make a more predictive models and a greater understanding of the user's taste.
2. With additional factors such as locations, it might be possible that there will be difference pages that have better conversion rate for different locations
3. The disadvantage with adding more factors would be misinterpretation because of human errors due to the failure of understanding what the data really entails.

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 [179]: # Impoert countries data
          countries_df = pd.read_csv('./countries.csv')

          # Merging the dataset
          df_new = countries_df.set_index('user_id').join(df2.set_index('user_id'), how='inner
```

```
In [180]: ### Create the necessary dummy variables
          df_new['intercept'] = 1
          df_new[['CA','US']] = pd.get_dummies(df_new['country'])[['CA','US']]
```

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 [181]: ### Fit Your Linear Model And Obtain the Results
          mod = sm.Logit(df_new['converted'], df_new[['intercept', 'US', 'CA']])

          results = mod.fit()

          results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.366116
         Iterations 6
```

```
Out[181]: <class 'statsmodels.iolib.summary.Summary'>
          """
                                     Logit Regression Results
          ==============================================================================
          Dep. Variable:              converted   No. Observations:               290584
          Model:                          Logit   Df Residuals:                   290581
          Method:                           MLE   Df Model:                            2
          Date:                Sun, 25 Nov 2018   Pseudo R-squ.:               1.521e-05
          Time:                        17:04:47   Log-Likelihood:            -1.0639e+05
          converged:                       True   LL-Null:                   -1.0639e+05
                                                  LLR p-value:                    0.1984
          ==============================================================================
                           coef    std err          z      P>|z|      [0.025      0.975]
          ------------------------------------------------------------------------------
          intercept     -1.9868      0.011   -174.174      0.000      -2.009      -1.964
          US            -0.0099      0.013     -0.746      0.456      -0.036       0.016
          CA            -0.0507      0.028     -1.786      0.074      -0.106       0.005
          ==============================================================================
          """
```

0.1.4 Conclusion of the model in h

1. We found out that the conversion rate has only low marginal effect on the control and treatment group

2. Hence, we failed to reject the null hypothesis.

## Conclusions
Congratulations on completing the project!

0.1.5 Gather Submission Materials

Once you are satisfied with the status of your Notebook, you should save it in a format that will make it easy for others to read. You can use the **File -> Download as -> HTML (.html)** menu to save your notebook as an .html file. If you are working locally and get an error about "No module name", then open a terminal and try installing the missing module using `pip install <module_name>` (don't include the "<" or ">" or any words following a period in the module name).

You will submit both your original Notebook and an HTML or PDF copy of the Notebook for review. There is no need for you to include any data files with your submission. If you made reference to other websites, books, and other resources to help you in solving tasks in the project, make sure that you document them. It is recommended that you either add a "Resources" section in a Markdown cell at the end of the Notebook report, or you can include a `readme.txt` file documenting your sources.

0.1.6 Submit the Project

When you're ready, click on the "Submit Project" button to go to the project submission page. You can submit your files as a .zip archive or you can link to a GitHub repository containing your project files. If you go with GitHub, note that your submission will be a snapshot of the linked repository at time of submission. It is recommended that you keep each project in a separate repository to avoid any potential confusion: if a reviewer gets multiple folders representing multiple projects, there might be confusion regarding what project is to be evaluated.

It can take us up to a week to grade the project, but in most cases it is much faster. You will get an email once your submission has been reviewed. If you are having any problems submitting your project or wish to check on the status of your submission, please email us at dataanalyst-project@udacity.com. In the meantime, you should feel free to continue on with your learning journey by beginning the next module in the program.