Analyze_ab_test_results_notebook-Copy1

June 21, 2020

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 [28]: 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 [6]: df=pd.read_csv('ab_data.csv')
        df.head()
```

```
Out[6]:    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 [7]: df.shape
```

```
Out[7]: (294478, 5)
```

c. The number of unique users in the dataset.

```
In [8]: df.nunique()
```

```
Out[8]: user_id         290584
        timestamp       294478
        group                2
        landing_page         2
        converted            2
        dtype: int64
```

d. The proportion of users converted.

```
In [9]: len(df[df['converted'] ==1])/df.shape[0]
```

```
Out[9]: 0.11965919355605512
```

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

```
In [10]: ((df.group=='treatment') & (df.landing_page!='new_page')).sum()+((df.group!='treatment'
```

```
Out[10]: 3893
```

f. Do any of the rows have missing values?

```
In [11]: 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 [12]: op=df['landing_page'] == 'old_page'
         np= df['landing_page'] == 'new_page'
         ctrl= df['group'] == 'control'
         tmt= df['group'] == 'treatment'
         df1 = (df[(tmt)&(op)]+ df[(ctrl)& (np)]).index
         df2=df.drop(df1)
```

```
In [13]: # 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[13]: 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 [14]: df2.nunique()

Out[14]: user_id         290584
         timestamp       290585
         group                2
         landing_page         2
         converted            2
         dtype: int64
```

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

```
In [15]: df2[df2.duplicated(subset=['user_id'], keep=False)]

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
```

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

```
In [16]: # The User is not converted and in the treatment group on the new page.
```

d. Remove **one** of the rows with a duplicate **user_id**, but keep your dataframe as **df2**.

```
In [17]: df2.drop_duplicates(subset='user_id', keep='first', inplace=True)
```

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 [18]: len(df[df['converted']== 1])/df.shape[0]

Out[18]: 0.11965919355605512
```

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

```
In [19]: df2.query('group == "control"')["converted"].mean()

Out[19]: 0.1203863045004612
```

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

```
In [20]: df2.query('group == "treatment"')["converted"].mean()

Out[20]: 0.11880806551510564
```

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

```
In [21]: len(df[df["landing_page"] == "new_page"])/df.shape[0]

Out[21]: 0.5
```

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 results show, that the conversion rate from the control group is slightly higher than from the treatment group. It is though, only minimally higher, so in my opinion it does not count as a sufficient evidence that any of the two pages lead to more conversions, as taking another sample, or a bigger sample would probably even out the number, so that you can say that for both pages the conversion rate is very similar. And therefore you can conclude that both pages lead to approximately the same amount of 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.

In the null hypothesis, H0 we start by first of all saying that the new version is either worse than, or the same as the old version, this is the one that we assume to be true in the beginning of our testing/ analysis. In terms that would mean: p_new = p_old or p_new<p_old. The Alternative hypothesis, H1 states, that the new version is better than the old one and this is the one that we want to prove as right, so therefore we would say: p_new > p_old. `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 [22]: p_new = df2.converted.mean()
         print(p_new)
```

```
0.119597087245
```

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

```
In [23]: p_old = df2.converted.mean()
         print(p_old)
```

```
0.119597087245
```

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

```
In [24]: n_new = sum(df2.landing_page == 'new_page')
         print(n_new)
```

```
145310
```

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

```
In [25]: n_old = sum (df2.landing_page == 'old_page')
         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 [29]:
```
new_page_converted =np.random.choice([0,1], size=n_new, p=(1-p_new, p_old))
```

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 [30]:
```
old_page_converted = np.random.choice([0,1], size=n_old, p=(1-p_old, p_old))
```

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

In [31]:
```
pnewpold= new_page_converted.mean()-old_page_converted.mean()
print(pnewpold)
```

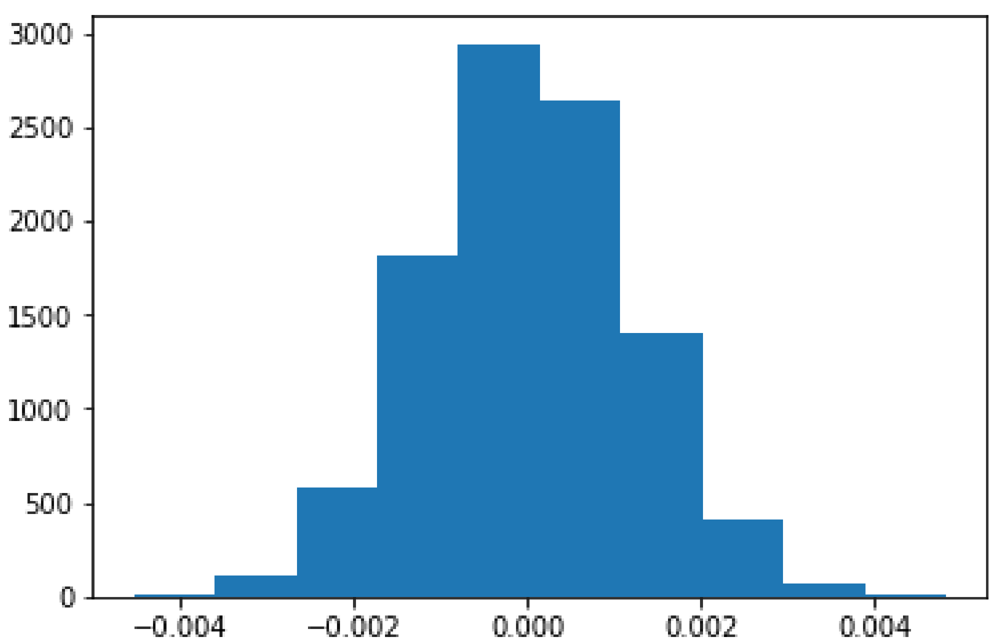
```
0.00147781588704
```

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 [32]:
```
p_diffs= []
for _ in range(10000):
    new_page_converted= np.random.choice([0,1], size =n_new, p=(1-p_new, p_new))
    old_page_converted= np.random.choice([0,1], size=n_old, p=(1-p_old, p_old))
    p_diffs.append(new_page_converted.mean()-old_page_converted.mean())
```

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 [33]:
```
plt.hist(p_diffs);
```



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

```
In [34]: obs_diff =df2.query('group=="treatment"')['converted'].mean()-df2.query('group=="contro
         obs_diff
         (p_diffs > obs_diff).mean()
```

```
Out[34]: 0.9114999999999998
```

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?

In part J what I have computed is the so called p value. It is a way to accept or to reject the null hypothesis. The smaller the p_value is, the closer it is to 0, the more likely we are to reject the null hypothesis. Here our pvalue is high, 0,9, which is why we stick to the null hypothesis as true. In terms of our analysis that means, that the mean of the ones that have converted to the new page is lower than the ones who have converted to the old page. As answered in a question above, we see the null hypothesis as true, if the new version is not better, or even worse than the old one, and this is the case here. To conclude, in the terms that we have learned in the lessons we say, that we FAIL 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 [63]: import statsmodels.api as sm
         convert_old=df2.query('converted == 1 and landing_page == "old_page"').count()[0]
         convert_new=df2.query('converted==1 and landing_page =="old_page"').count()[0]
         n_old = df2.query ('landing_page == "old_page"').count()[0]
         n_new= df2.query ('landing_page == "new_page"').count()[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 [64]: from scipy.stats import norm
         critical_value= norm.ppf(1-(0.05/2))
         z_score, p_value= sm.stats.proportions_ztest([convert_new, convert_old], [n_new, n_old]
         print("z_score= {}, p_value={} and critical_value={}".format(round(z_score,2), round(p_
```

```
z_score= -0.02, p_value=0.51 and critical_value=1.96
```

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.**?

We have a negative z_score. The zscore shows how far away data points are from the mean. While a positice zscore means its higher, a negative one means lower than the mean. , which means that we have to stick to the null hypothesis as being true, because this means not many converted, here it is 0,02 below the mean, that means less have converted. Also the p_value I have computed here is 0.51, this is a high p_value.which means a relatively high confidence not to reject the null hyothesis, which is why here we also fail to reject this one.

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?

Since we have only the two possibilities, conversion and non conversion, which are both categorical, I should use the logistic regression, as this is the common way to do this, when you have got two categorical data.

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 [65]: df2['intercept']=1
         df2['ab_page']= (df2.group == 'treatment').astype(int)
```

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 [66]: rmodel=sm.Logit(df2.converted, df2[['intercept', 'ab_page']])
```

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

```
In [67]: result=rmodel.fit()
         result.summary2()
```

```
Optimization terminated successfully.
         Current function value: 0.366118
         Iterations 6
```

```
Out[67]: <class 'statsmodels.iolib.summary2.Summary'>
         """
                                  Results: Logit
         ==================================================================
         Model:              Logit            No. Iterations:   6.0000
         Dependent Variable: converted        Pseudo R-squared: 0.000
```

```
Date:                  2020-06-21 13:50 AIC:              212780.3502
No. Observations:      290584           BIC:              212801.5095
Df Model:              1                Log-Likelihood:   -1.0639e+05
Df Residuals:          290582           LL-Null:          -1.0639e+05
Converged:             1.0000           Scale:            1.0000
-------------------------------------------------------------------
              Coef.   Std.Err.      z      P>|z|    [0.025   0.975]
-------------------------------------------------------------------
intercept    -1.9888    0.0081  -246.6690 0.0000  -2.0046  -1.9730
ab_page      -0.0150    0.0114    -1.3109 0.1899  -0.0374   0.0074
==================================================================

"""
```

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 that we have found out here for the ab_page is 0.1899. The difference between here and in part II is , that in part II we have conducted a one sided test, and here we have performed a two sided test.

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 things, that may be looked at that state if an individual converts or not could for example be the country it lives in, or also maybe the gender. It is always good to look at more terms in a model, to get an even more detailed result, but you have to be aware of the fact that if you add too many new factors at once, that you may not get an accurate view over what may have an impact. It would be more clever, to always add one more factor at a time, to really see how it moves, even though that would be really time consuming, but it is the only way to get a clear vision. What is also a problem when adding to many new factors, that it could happen that you dont know anymore what has an impact on what.

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 [68]:
```
countries_df=pd.read_csv('countries.csv')
df3=pd.merge(df2, countries_df, on='user_id')
df3.head()
```

```
Out[68]:    user_id                   timestamp      group landing_page  converted  \
         0   851104  2017-01-21 22:11:48.556739    control     old_page          0
         1   804228  2017-01-12 08:01:45.159739    control     old_page          0
         2   661590  2017-01-11 16:55:06.154213  treatment     new_page          0
         3   853541  2017-01-08 18:28:03.143765  treatment     new_page          0
         4   864975  2017-01-21 01:52:26.210827    control     old_page          1

            intercept  ab_page country
         0          1        0      US
         1          1        0      US
         2          1        1      US
         3          1        1      US
         4          1        0      US
```

above I have joined the df2 and countries datasets, next i am going to use value_counts, to see all the countries that appear, so I know for what i will have to make dummy variables.

```
In [69]: df3['country'].value_counts()

Out[69]: US    203619
         UK     72466
         CA     14499
         Name: country, dtype: int64

In [70]: df3[['US', 'UK', 'CA']]=pd.get_dummies(df3['country'])
         df3.head()

Out[70]:    user_id                   timestamp      group landing_page  converted  \
         0   851104  2017-01-21 22:11:48.556739    control     old_page          0
         1   804228  2017-01-12 08:01:45.159739    control     old_page          0
         2   661590  2017-01-11 16:55:06.154213  treatment     new_page          0
         3   853541  2017-01-08 18:28:03.143765  treatment     new_page          0
         4   864975  2017-01-21 01:52:26.210827    control     old_page          1

            intercept  ab_page country  US  UK  CA
         0          1        0      US   0   0   1
         1          1        0      US   0   0   1
         2          1        1      US   0   0   1
         3          1        1      US   0   0   1
         4          1        0      US   0   0   1
```

now we are going to fit the model, but as we have learned in the lessons, we need to drop a column, so that the matrice is full rank which means that the columns are linearly independent, that is why we are only going to use us and uk in our model! In other words, the number of columns should be the number of categorical variables minus 1, so I will be dropping CA

```
In [71]: model=sm.Logit(df3['converted'], df3[['intercept', 'UK', 'US']])
         result=model.fit()
         result.summary2()
```

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

```
Out[71]: <class 'statsmodels.iolib.summary2.Summary'>
"""
                          Results: Logit
==================================================================
Model:              Logit            No. Iterations:   6.0000
Dependent Variable: converted        Pseudo R-squared: 0.000
Date:               2020-06-21 13:51 AIC:              212780.8333
No. Observations:   290584           BIC:              212812.5723
Df Model:           2                Log-Likelihood:   -1.0639e+05
Df Residuals:       290581           LL-Null:          -1.0639e+05
Converged:          1.0000           Scale:            1.0000
-------------------------------------------------------------------
              Coef.  Std.Err.     z      P>|z|   [0.025   0.975]
-------------------------------------------------------------------
intercept    -1.9967   0.0068  -292.3145 0.0000 -2.0101  -1.9833
UK            0.0099   0.0133     0.7458 0.4558 -0.0161   0.0360
US           -0.0408   0.0269    -1.5178 0.1291 -0.0935   0.0119
==================================================================

"""
```

So what we can see here when we look at the coefficient, we can see that it is slightly negative for the US, that UK is only slightly positive above 0 and that the Intercept, our predicted value is negative. What that means for our dependent variable relationship is, that there is a negative relationship between people converting from the US and only slightly positive relationship for people converting from the UK. The difference though is so small, that you can not build an opinion from that, because the difference is so extremely small.

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 [75]: df3['US_ab_page']=df3['US']*df3['ab_page']
         df3['UK_ab_page']=df3['UK']*df3['ab_page']
         print(df3)
```

```
        user_id                   timestamp      group landing_page  \
0        851104  2017-01-21 22:11:48.556739    control     old_page
1        804228  2017-01-12 08:01:45.159739    control     old_page
2        661590  2017-01-11 16:55:06.154213  treatment     new_page
3        853541  2017-01-08 18:28:03.143765  treatment     new_page
4        864975  2017-01-21 01:52:26.210827    control     old_page
```

5	936923	2017-01-10 15:20:49.083499	control	old_page
6	679687	2017-01-19 03:26:46.940749	treatment	new_page
7	719014	2017-01-17 01:48:29.539573	control	old_page
8	817355	2017-01-04 17:58:08.979471	treatment	new_page
9	839785	2017-01-15 18:11:06.610965	treatment	new_page
10	929503	2017-01-18 05:37:11.527370	treatment	new_page
11	834487	2017-01-21 22:37:47.774891	treatment	new_page
12	803683	2017-01-09 06:05:16.222706	treatment	new_page
13	944475	2017-01-22 01:31:09.573836	treatment	new_page
14	718956	2017-01-22 11:45:11.327945	treatment	new_page
15	644214	2017-01-22 02:05:21.719434	control	old_page
16	847721	2017-01-17 14:01:00.090575	control	old_page
17	888545	2017-01-08 06:37:26.332945	treatment	new_page
18	650559	2017-01-24 11:55:51.084801	control	old_page
19	935734	2017-01-17 20:33:37.428378	control	old_page
20	740805	2017-01-12 18:59:45.453277	treatment	new_page
21	759875	2017-01-09 16:11:58.806110	treatment	new_page
22	793849	2017-01-23 22:36:10.742811	treatment	new_page
23	905617	2017-01-20 14:12:19.345499	treatment	new_page
24	746742	2017-01-23 11:38:29.592148	control	old_page
25	892356	2017-01-05 09:35:14.904865	treatment	new_page
26	773302	2017-01-12 08:29:49.810594	treatment	new_page
27	913579	2017-01-24 09:11:39.164256	control	old_page
28	736159	2017-01-06 01:50:21.318242	treatment	new_page
29	690284	2017-01-13 17:22:57.182769	control	old_page
...	...	...	...	...
290554	776137	2017-01-12 05:53:12.386730	treatment	new_page
290555	883344	2017-01-22 23:15:58.645325	treatment	new_page
290556	825594	2017-01-06 12:37:08.897784	treatment	new_page
290557	875688	2017-01-14 07:19:49.042869	control	old_page
290558	927527	2017-01-12 10:52:11.084740	control	old_page
290559	789177	2017-01-17 18:17:56.215378	control	old_page
290560	937338	2017-01-19 03:23:22.236666	treatment	new_page
290561	733101	2017-01-23 12:52:58.711914	treatment	new_page
290562	679096	2017-01-02 16:43:49.237940	treatment	new_page
290563	691699	2017-01-09 23:42:35.963486	treatment	new_page
290564	807595	2017-01-22 10:43:09.285426	treatment	new_page
290565	924816	2017-01-20 10:59:03.481635	control	old_page
290566	846225	2017-01-16 15:24:46.705903	treatment	new_page
290567	740310	2017-01-10 17:22:19.762612	control	old_page
290568	677163	2017-01-03 19:41:51.902148	treatment	new_page
290569	832080	2017-01-19 13:18:27.352570	control	old_page
290570	834362	2017-01-17 01:51:56.106436	control	old_page
290571	925675	2017-01-07 20:38:26.346410	treatment	new_page
290572	923948	2017-01-09 16:33:41.104573	control	old_page
290573	857744	2017-01-05 08:00:56.024226	control	old_page
290574	643562	2017-01-02 19:20:05.460595	treatment	new_page
290575	755438	2017-01-18 17:35:06.149568	control	old_page

290576	908354	2017-01-11 02:42:21.195145	control	old_page
290577	718310	2017-01-21 22:44:20.378320	control	old_page
290578	822004	2017-01-04 03:36:46.071379	treatment	new_page
290579	751197	2017-01-03 22:28:38.630509	control	old_page
290580	945152	2017-01-12 00:51:57.078372	control	old_page
290581	734608	2017-01-22 11:45:03.439544	control	old_page
290582	697314	2017-01-15 01:20:28.957438	control	old_page
290583	715931	2017-01-16 12:40:24.467417	treatment	new_page

	converted	intercept	ab_page	country	US	UK	CA	US_ab_page	\
0	0	1	0	US	0	0	1	0	
1	0	1	0	US	0	0	1	0	
2	0	1	1	US	0	0	1	0	
3	0	1	1	US	0	0	1	0	
4	1	1	0	US	0	0	1	0	
5	0	1	0	US	0	0	1	0	
6	1	1	1	CA	1	0	0	1	
7	0	1	0	US	0	0	1	0	
8	1	1	1	UK	0	1	0	0	
9	1	1	1	CA	1	0	0	1	
10	0	1	1	UK	0	1	0	0	
11	0	1	1	US	0	0	1	0	
12	0	1	1	US	0	0	1	0	
13	0	1	1	US	0	0	1	0	
14	0	1	1	US	0	0	1	0	
15	1	1	0	US	0	0	1	0	
16	0	1	0	US	0	0	1	0	
17	1	1	1	US	0	0	1	0	
18	0	1	0	CA	1	0	0	0	
19	0	1	0	US	0	0	1	0	
20	0	1	1	US	0	0	1	0	
21	0	1	1	UK	0	1	0	0	
22	0	1	1	US	0	0	1	0	
23	0	1	1	UK	0	1	0	0	
24	0	1	0	US	0	0	1	0	
25	1	1	1	UK	0	1	0	0	
26	0	1	1	US	0	0	1	0	
27	1	1	0	US	0	0	1	0	
28	0	1	1	US	0	0	1	0	
29	0	1	0	US	0	0	1	0	
...	...	...	...	...	..	..	..	...	
290554	0	1	1	US	0	0	1	0	
290555	0	1	1	CA	1	0	0	1	
290556	0	1	1	UK	0	1	0	0	
290557	0	1	0	US	0	0	1	0	
290558	0	1	0	US	0	0	1	0	
290559	0	1	0	US	0	0	1	0	
290560	0	1	1	UK	0	1	0	0	

290561	0	1	1	US	0	0	1	0
290562	0	1	1	US	0	0	1	0
290563	0	1	1	US	0	0	1	0
290564	0	1	1	US	0	0	1	0
290565	0	1	0	US	0	0	1	0
290566	0	1	1	US	0	0	1	0
290567	0	1	0	US	0	0	1	0
290568	0	1	1	US	0	0	1	0
290569	0	1	0	US	0	0	1	0
290570	0	1	0	US	0	0	1	0
290571	0	1	1	US	0	0	1	0
290572	0	1	0	US	0	0	1	0
290573	0	1	0	US	0	0	1	0
290574	0	1	1	CA	1	0	0	1
290575	0	1	0	US	0	0	1	0
290576	0	1	0	US	0	0	1	0
290577	0	1	0	US	0	0	1	0
290578	0	1	1	CA	1	0	0	1
290579	0	1	0	US	0	0	1	0
290580	0	1	0	US	0	0	1	0
290581	0	1	0	US	0	0	1	0
290582	0	1	0	US	0	0	1	0
290583	0	1	1	UK	0	1	0	0

	UK_ab_page
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	1
9	0
10	1
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	1
22	0

```
23                1
24                0
25                1
26                0
27                0
28                0
29                0
...             ...
290554            0
290555            0
290556            1
290557            0
290558            0
290559            0
290560            1
290561            0
290562            0
290563            0
290564            0
290565            0
290566            0
290567            0
290568            0
290569            0
290570            0
290571            0
290572            0
290573            0
290574            0
290575            0
290576            0
290577            0
290578            0
290579            0
290580            0
290581            0
290582            0
290583            1

[290584 rows x 13 columns]
```

```
In [77]: lm=sm.Logit(df3['converted'], df3[['intercept', 'UK_ab_page', 'US_ab_page']])
         result=lm.fit()
         result.summary2()
```

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

```
Iterations 6
```

Out[77]:
```
<class 'statsmodels.iolib.summary2.Summary'>
"""
                          Results: Logit
==================================================================
Model:              Logit            No. Iterations:   6.0000
Dependent Variable: converted        Pseudo R-squared: 0.000
Date:               2020-06-21 13:56 AIC:              212779.0384
No. Observations:   290584           BIC:              212810.7773
Df Model:           2                Log-Likelihood:   -1.0639e+05
Df Residuals:       290581           LL-Null:          -1.0639e+05
Converged:          1.0000           Scale:            1.0000
-------------------------------------------------------------------
              Coef.  Std.Err.     z     P>|z|   [0.025   0.975]
-------------------------------------------------------------------
intercept    -1.9963   0.0062 -322.0487 0.0000 -2.0084 -1.9841
UK_ab_page    0.0149   0.0173    0.8617 0.3888 -0.0190  0.0488
US_ab_page   -0.0752   0.0376   -1.9974 0.0458 -0.1489 -0.0014
==================================================================

"""
```

Now we can see, that the conversion rates in the uk_ab_page and US_ab_page are quite similar as well, even though one is negative, but this only strengthens the hypothesis that country does not have an impact on our conversion rate here.

https://statisticsbyjim.com/regression/interpret-coefficients-p-values-regression/
http://resources.esri.com/help/9.3/arcgisengine/java/gp_toolref/spatial_statistics_toolbox/what_is_a_z_scor

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

In [ ]: