Analyze_ab_test_results_notebook

October 10, 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 [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]: df.shape
```

```
Out[3]: (294478, 5)
```

number of rows = 294478

c. The number of unique users in the dataset.

```
In [4]: df.user_id.nunique()
```

```
Out[4]: 290584
```

The number of unique =290584

```
In [5]: sum(df.duplicated())
```

```
Out[5]: 0
```

d. The proportion of users converted.

```
In [6]:  df['converted'].mean()
```

```
Out[6]: 0.11965919355605512
```

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

```
In [7]: no_mat = df.query("(group == 'control' and landing_page == 'new_page') or (group == 'tre
        no_mat
```

```
Out[7]: 3893
```

The number of times the new_page and treatment =3893

f. Do any of the rows have missing values?

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294478 entries, 0 to 294477
Data columns (total 5 columns):
user_id         294478 non-null int64
timestamp       294478 non-null object
group           294478 non-null object
landing_page    294478 non-null object
converted       294478 non-null int64
dtypes: int64(2), object(3)
memory usage: 11.2+ MB
```

```
In [9]: df.isnull().sum()
```

```
Out[9]: user_id         0
        timestamp       0
        group           0
        landing_page    0
        converted       0
        dtype: int64
```

no missing values

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 [10]: df2 = df.query("(group == 'treatment' and landing_page == 'new_page') or (group == 'con
         df2.head(20)
```

Out[10]:

	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
5	936923	2017-01-10 15:20:49.083499	control	old_page	0
6	679687	2017-01-19 03:26:46.940749	treatment	new_page	1
7	719014	2017-01-17 01:48:29.539573	control	old_page	0
8	817355	2017-01-04 17:58:08.979471	treatment	new_page	1
9	839785	2017-01-15 18:11:06.610965	treatment	new_page	1
10	929503	2017-01-18 05:37:11.527370	treatment	new_page	0
11	834487	2017-01-21 22:37:47.774891	treatment	new_page	0
12	803683	2017-01-09 06:05:16.222706	treatment	new_page	0
13	944475	2017-01-22 01:31:09.573836	treatment	new_page	0

```
14   718956  2017-01-22 11:45:11.327945  treatment     new_page          0
15   644214  2017-01-22 02:05:21.719434    control     old_page          1
16   847721  2017-01-17 14:01:00.090575    control     old_page          0
17   888545  2017-01-08 06:37:26.332945  treatment     new_page          1
18   650559  2017-01-24 11:55:51.084801    control     old_page          0
19   935734  2017-01-17 20:33:37.428378    control     old_page          0
```

In [11]:
```
# 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[11]: 0

In [12]:
```
df2.shape
```

Out[12]: (290585, 5)

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 [13]:
```
# the unique user_ids in df2

df2.user_id.nunique()
```

Out[13]: 290584

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

In [14]:
```
#print the user_id repeated in df2 by calling duplicated function

duplicat = df2.loc[df2['user_id'].duplicated(), 'user_id']
print(duplicat)
```

```
2893    773192
Name: user_id, dtype: int64
```

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

In [15]:
```
df2[df2.user_id.duplicated()]
```

Out[15]:

	user_id	timestamp	group	landing_page	converted
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 [16]:
```
df.drop_duplicates(['user_id'],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 [17]: # probability of an individual converting in df2

         df2['converted'].mean()

Out[17]: 0.11959667567149027
```

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

```
In [18]: contpro=df2[df2['group'] == 'control']['converted'].mean()
```

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

```
In [19]: tpro=df2[df2['group'] == 'treatment']['converted'].mean()
```

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

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

Out[20]: 0.5000636646764286
```

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.

No . it does not feel like one page leadd to nore convsions
the new page reportedly contributed to a lower conversion rate than the old one
and
From above results , we can find this Answers :

1) What is the probability of an individual converting regardless of the page they receive? 0.11959708724499628

2) Given that an individual was in the control group, what is the probability they converted? 0.1203863045004612

3) Given that an individual was in the treatment group, what is the probability they converted? 0.11880806551510564

4) What is the probability that an individual received the new page? 0.5000619442226688.

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.

H0: <=

H1: >

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 [21]: pnew = df2['converted'].mean()
         pnew
```

```
Out[21]: 0.11959667567149027
```

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

```
In [22]: pold=df2['converted'].mean()
         pold
```

```
Out[22]: 0.11959667567149027
```

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

```
In [23]: nnew = df2.query('group == "treatment"').converted.count()
         nnew
```

```
Out[23]: 145311
```

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

```
In [24]: nold = df2.query('group == "control"').converted.count()
         nold
```

```
Out[24]: 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 [25]: new_page_converted = np.random.binomial(nnew , pnew)
         new_page_converted
```

Out[25]: 17427

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 [26]: old_page_converted = np.random.binomial(nold , pold)
         old_page_converted
```

Out[26]: 17473

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

```
In [27]: pnew-pold
```

Out[27]: 0.0

from the hypothies test we found h0 null hyopthies is true becuse H0: <=

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**.

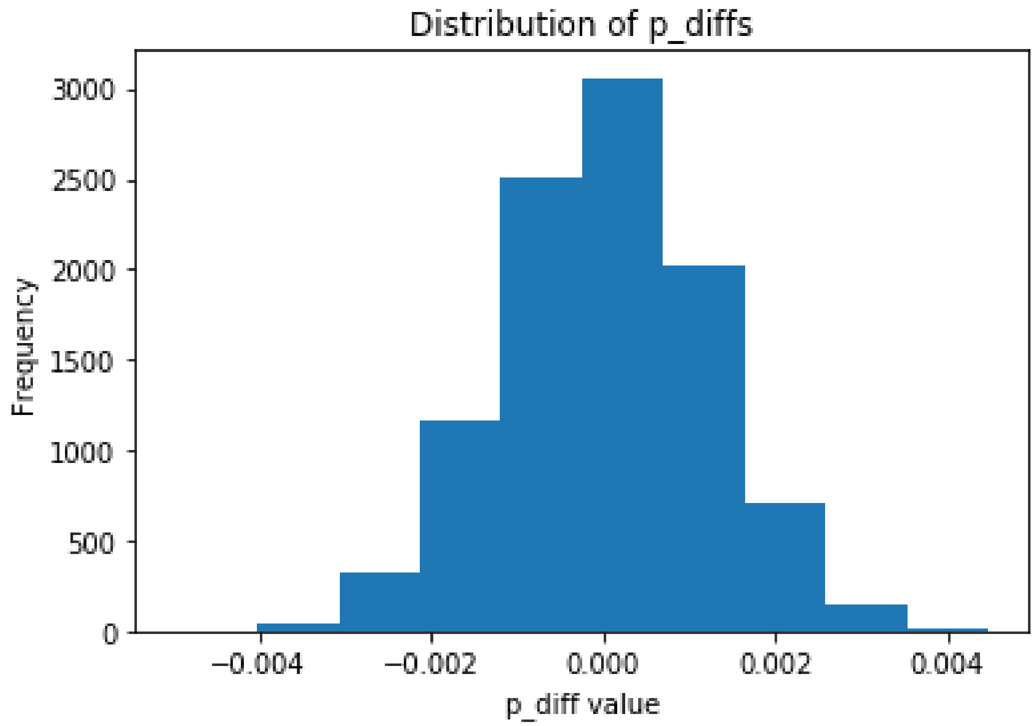
```
In [28]: p_diffs = []
         new_page_converted = np.random.binomial(nnew,pnew,10000)/nnew
         old_page_converted = np.random.binomial(nold,pold,10000)/nold
         p_diffs = new_page_converted - old_page_converted
         p_diffs
```

```
Out[28]: array([-0.00127622, -0.0006982 ,  0.00107746, ...,  0.0015317 ,
                 0.00070581, -0.00031946])
```

```
In [29]: p_diffs = np.array(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 [30]: plt.hist(p_diffs)
         plt.title('Distribution of p_diffs')
         plt.xlabel('p_diff value')
         plt.ylabel('Frequency');
```

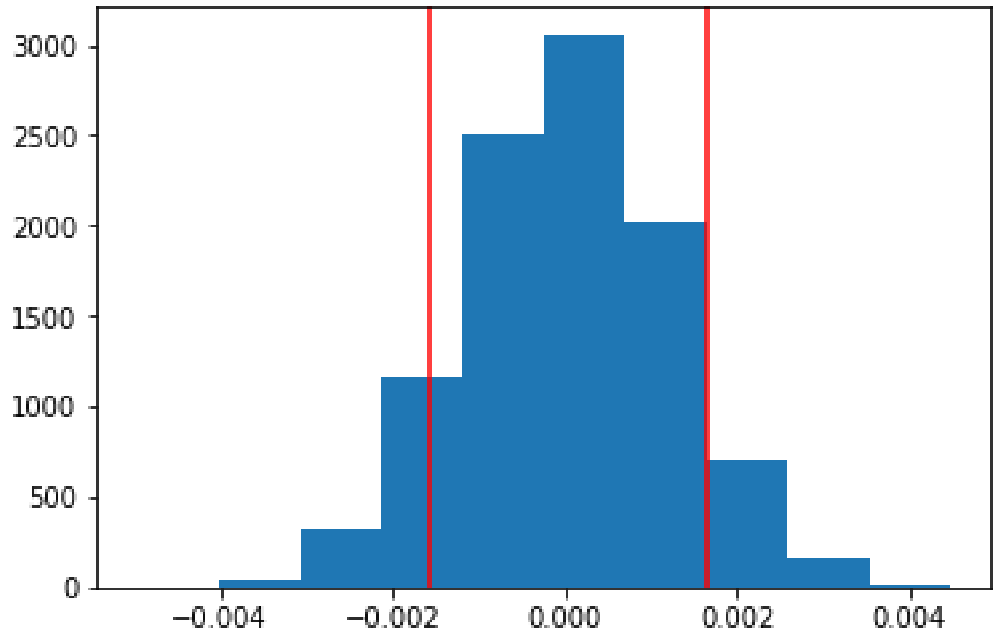


j. What proportion of the **p_diffs** are greater than the actual difference observed in **ab_data.csv**?

```
In [31]: obser_pro = tpro - contpro

         low_prob = (p_diffs < obser_pro).mean()
         high_prob = (p_diffs.mean() + (p_diffs.mean() - obser_pro) < p_diffs).mean()

         plt.hist(p_diffs);
         plt.axvline(obser_pro, color='red')
         plt.axvline(p_diffs.mean() + (p_diffs.mean() - obser_pro), color='red');
```




```
In [32]: low_prob = (p_diffs > obser_pro).mean()
         print(low_prob)
```

```
0.9052
```

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?

If p-value <= 0.05 (small): strong evidence against the null
If p-value > 0.05 (large): weak evidence against the null
https://www.simplypsychology.org/p-value.html
it appears that the p-value is above 0.05,(the Type I error) so We failed to reject the null hypothesis, and that the processing page did not have higher conversion rates than the control page on a statistically significant basis. Note that the value of p-value is large (~0.9)

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
```

```
convert_old = sum(df2.query("landing_page == 'old_page'")['converted'])
convert_new = sum(df2.query("landing_page == 'new_page'")['converted'])
n_old = len(df2.query("landing_page == 'old_page'"))
n_new = len(df2.query("landing_page == 'new_page'"))
```

```
/opt/conda/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The panda
  from pandas.core import datetools
```

```
In [34]: print ('convert_old ' , convert_old) ,
         print ('convert_new ' , convert_new) ,
         print ('n_old ' , n_old) ,
         print ('n_new ' , n_new)
```

```
convert_old  17489
convert_new  17264
n_old  145274
n_new  145311
```

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

```
z_score  1.31160753391
p_value   0.905173705141
```

```
In [36]: from scipy.stats import norm

         print(norm.cdf(z_score))
         print(norm.ppf(1-(0.05)))
```

```
0.905173705141
1.64485362695
```

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

The z-score is the difference between our test statistic or conversion rates z-score or (null hypothesis) is 1.31092419842 in this case (standard deviations above the mean) This is less than the critical value (1.96) We'd have to reject the null hypothesis. and p_value = 0.905173705141 which isn't below alpha of 0.05 , This p-value is similar to the previous p-value so the z-test appears to agree with the previous findings in parts j. and k .

https://www.simplypsychology.org/z-score.html
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 : because we want to kaow the odds of conversion

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 [37]:
```
df2['intercept'] = 1
df2[['new_page','old_page']] = pd.get_dummies(df2['landing_page'])
df2['ab_page'] = pd.get_dummies(df2['group'])['treatment']
df2.head()
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  """Entry point for launching an IPython kernel.
/opt/conda/lib/python3.6/site-packages/pandas/core/frame.py:3140: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  self[k1] = value[k2]
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[37]:
```
   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  new_page  old_page  ab_page
```

```
0          1          0          1          0
1          1          0          1          0
2          1          1          0          1
3          1          1          0          1
4          1          0          1          0
```

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 [38]: import statsmodels.api as sm
         from scipy import stats
```

```
In [39]: log_model = sm.Logit(df2['converted'],df2[['intercept' ,'ab_page']])
         results = log_model.fit()
```

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

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

```
In [40]: print(results.summary2())
         #(results.summary) was not working Please note this because it was difficult
```

```
                          Results: Logit
=================================================================
Model:              Logit            No. Iterations:   6.0000
Dependent Variable: converted        Pseudo R-squared: 0.000
Date:               2020-10-10 11:54 AIC:              212780.6032
No. Observations:   290585           BIC:              212801.7625
Df Model:           1                Log-Likelihood:   -1.0639e+05
Df Residuals:       290583           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    -1.3116 0.1897  -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 associated with the ab_page is 0.190, which is different from that found in PartII (0.9). But the greater p_value always shows the same conclusion that the old page is better than or equal to the current page.

null and alternative in case P_new and P_old is :

H0:P_new=P_old

H0:P_new=P_old

H1:P_newP_old

H1:P_newP_old

Here the alternative is 'not equal' and is a two-sided test, although our conclusions were different in the A / B 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?

Since the current hypotheses only used a single conversion factor, further variables could be integrated into the model.

We have a number of influences that can affect human conversions such as gender , culture and age group New trends can be found using other variables

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 [41]: countries_df = pd.read_csv('countries.csv')
         df_join = countries_df.set_index('user_id').join(df2.set_index('user_id'), how='inner')
         df_join.head()

Out[41]:         country                   timestamp      group landing_page  \
         user_id
         630000       US  2017-01-19 06:26:06.548941  treatment     new_page
         630001       US  2017-01-16 03:16:42.560309  treatment     new_page
         630002       US  2017-01-19 19:20:56.438330    control     old_page
         630003       US  2017-01-12 10:09:31.510471  treatment     new_page
         630004       US  2017-01-18 20:23:58.824994  treatment     new_page

                  converted  intercept  new_page  old_page  ab_page
         user_id
         630000           0          1         1         0        1
         630001           1          1         1         0        1
         630002           0          1         0         1        0
         630003           0          1         1         0        1
         630004           0          1         1         0        1
```

```
In [42]: df_join['country'].value_counts()
```

```
Out[42]: US    203620
         UK     72466
         CA     14499
         Name: country, dtype: int64
```

```
In [43]: df_join[['CA','UK','US']]=pd.get_dummies(df_join['country'])
         df_join.head()
```

```
Out[43]:        country                   timestamp      group landing_page  \
         user_id
         630000      US  2017-01-19 06:26:06.548941  treatment     new_page
         630001      US  2017-01-16 03:16:42.560309  treatment     new_page
         630002      US  2017-01-19 19:20:56.438330    control     old_page
         630003      US  2017-01-12 10:09:31.510471  treatment     new_page
         630004      US  2017-01-18 20:23:58.824994  treatment     new_page

                  converted  intercept  new_page  old_page  ab_page  CA  UK  US
         user_id
         630000           0          1         1         0        1   0   0   1
         630001           1          1         1         0        1   0   0   1
         630002           0          1         0         1        0   0   0   1
         630003           0          1         1         0        1   0   0   1
         630004           0          1         1         0        1   0   0   1
```

```
In [44]: mod = sm.Logit(df_join['converted'], df_join[['intercept', 'CA', 'UK']])
         results = mod.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.366115
         Iterations 6
```

```
In [45]: results.summary2()
```

```
Out[45]: <class 'statsmodels.iolib.summary2.Summary'>
         """
                                   Results: Logit
         ==================================================================
         Model:              Logit            No. Iterations:   6.0000
         Dependent Variable: converted        Pseudo R-squared: 0.000
         Date:               2020-10-10 11:54 AIC:              212781.0880
         No. Observations:   290585           BIC:              212812.8269
         Df Model:           2                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    -1.9967    0.0068  -292.3154  0.0000  -2.0101  -1.9833
CA           -0.0408    0.0269    -1.5176  0.1291  -0.0935   0.0119
UK            0.0099    0.0133     0.7462  0.4555  -0.0161   0.0360
=====================================================================

"""
```

Also, on the basis of the above p-values, it does not appear as if the country has a major effect on conversion.

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 [49]:
```
df_join['CA_page'] = df_join['CA']*df_join['ab_page']
df_join['UK_page'] = df_join['UK']*df_join['ab_page']
df_join['US_page'] = df_join['US']*df_join['ab_page']
mod = sm.Logit(df_join['converted'], df_join[['intercept', 'CA_page', 'UK_page','US_pag
results = mod.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.366108
         Iterations 6
```

In [50]:
```
results.summary2()
```

Out[50]:
```
<class 'statsmodels.iolib.summary2.Summary'>
"""
                          Results: Logit
=====================================================================
Model:              Logit            No. Iterations:   6.0000
Dependent Variable: converted        Pseudo R-squared: 0.000
Date:               2020-10-10 11:56 AIC:              212779.1904
No. Observations:   290585           BIC:              212821.5090
Df Model:           3                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    -1.9888    0.0081  -246.6690  0.0000  -2.0046  -1.9730
CA_page      -0.0827    0.0380    -2.1763  0.0295  -0.1571  -0.0082
UK_page       0.0074    0.0180     0.4098  0.6819  -0.0279   0.0427
US_page      -0.0183    0.0126    -1.4495  0.1472  -0.0430   0.0064
=====================================================================

"""
```

We do not have enough proof to dismiss the null hypothesis based on any of our A / B tests. As a consequence, there is no need to move to a new website, when the old one is performing just as well.

1 Conclusions

My conclusion to this project is in the form of advice. I recommend not wasting resources and time creating a new web feed because it was a waste of time and money. Where the indicators showed no satistical or practical significance Since the p_value is > 0.05

```
In [48]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])
```

```
Out[48]: 0
```