Analyze_ab_test_results_notebook

August 26, 2020

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

- Section **??**
- Section **??**
- Section **??**
- Section **??**

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 [2]: 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 [3]: #importing the dataset
        df = pd.read_csv('ab_data.csv')

        #Showing the first 5 rows of the dataset
        df.head()
```

Out[3]:

	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 [4]: #Showing number of the rows
        df.shape[0]
```

```
Out[4]: 294478
```

c. The number of unique users in the dataset.

```
In [5]: df.nunique()
```

```
Out[5]: user_id         290584
        timestamp       294478
        group                2
        landing_page         2
        converted            2
        dtype: int64
```

d. The proportion of users converted.

```
In [6]: len(df.query('converted == True'))/(df.shape[0])
```

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

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

```
In [7]: len(df.query('group == "treatment" and not landing_page == "new_page"')) + (len(df.query
```

```
Out[7]: 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
```

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 [9]: df2 = df.drop(df[((df['group'] == 'treatment') == (df['landing_page'] == 'new_page')) ==
```

```
In [10]: # 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[10]: 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 [11]: df2.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 [12]: df2.nunique()
```

```
Out[12]: 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 [13]: df2[df2['user_id'].duplicated()]
```

```
Out[13]:       user_id                   timestamp      group landing_page  converted
         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 [14]: df2[df2['user_id'].duplicated()]
```

```
Out[14]:       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 [15]: df2 = df2.drop_duplicates(subset="user_id")
```

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 [16]: df2['converted'].mean()
```

```
Out[16]: 0.11959708724499628
```

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

```
In [17]: df2.query('group == "control"').converted.mean()
```

```
Out[17]: 0.1203863045004612
```

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

```
In [18]: df2.query('group == "treatment"').converted.mean()
```

```
Out[18]: 0.11880806551510564
```

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

```
In [19]: len(df2.query('landing_page == "new_page"'))/(df2.shape[0])
```

```
Out[19]: 0.5000619442226688
```

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.

- As we can see above, the probability of converting regardless of page is 0.1196.
- And the probability of converting for the control group whom have the old page is 0.1204.
- And the probability of converting for the treatment group whom have the new page is 0.1188.
- I see that further ivestigarion is needed by the hypothesis test to see how far these values are from the P-Value. As of right now the diffrence of converted between the old and new page is just too little and it is hard to determine what to do because of that.

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:** Pnew Pold <= 0
- **H1:** Pnew Pold > 0

`2.` Assume under the null hypothesis, p_{new} and p_{old} both have "true" success rates equal to the **converted** success rate regardless of page - that is p_{new} and p_{old} are equal. Furthermore, assume they are equal to the **converted** rate in **ab_data.csv** regardless of the page.

Use a sample size for each page equal to the ones in **ab_data.csv**.

Perform the sampling distribution for the difference in **converted** between the two pages over 10,000 iterations of calculating an estimate from the null.

Use the cells below to provide the necessary parts of this simulation. If this doesn't make complete sense right now, don't worry - you are going to work through the problems below to complete this problem. You can use **Quiz 5** in the classroom to make sure you are on the right track.

a. What is the **conversion rate** for p_{new} under the null?

```
In [20]: Pnew = df2['converted'].mean()
         print(Pnew)
```

```
0.119597087245
```

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

```
In [21]: Pold = df2['converted'].mean()
         print(Pold)
```

```
0.119597087245
```

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

```
In [22]: n_new = df2.query('group == "treatment"').user_id.count()
         print(n_new)
```

```
145310
```

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

```
In [23]: n_old = df2.query('group == "control"').user_id.count()
         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 [24]: new_page_converted = np.random.choice([1, 0], size=n_new, p=[Pnew, (1-Pnew)])
         len(new_page_converted)
```

```
Out[24]: 145310
```

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 [25]: old_page_converted = np.random.choice([1, 0], size=n_old, p=[Pold, (1-Pold)])
         len(old_page_converted)
```

```
Out[25]: 145274
```

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

```
In [26]: new_page_converted.mean() - old_page_converted.mean()
```

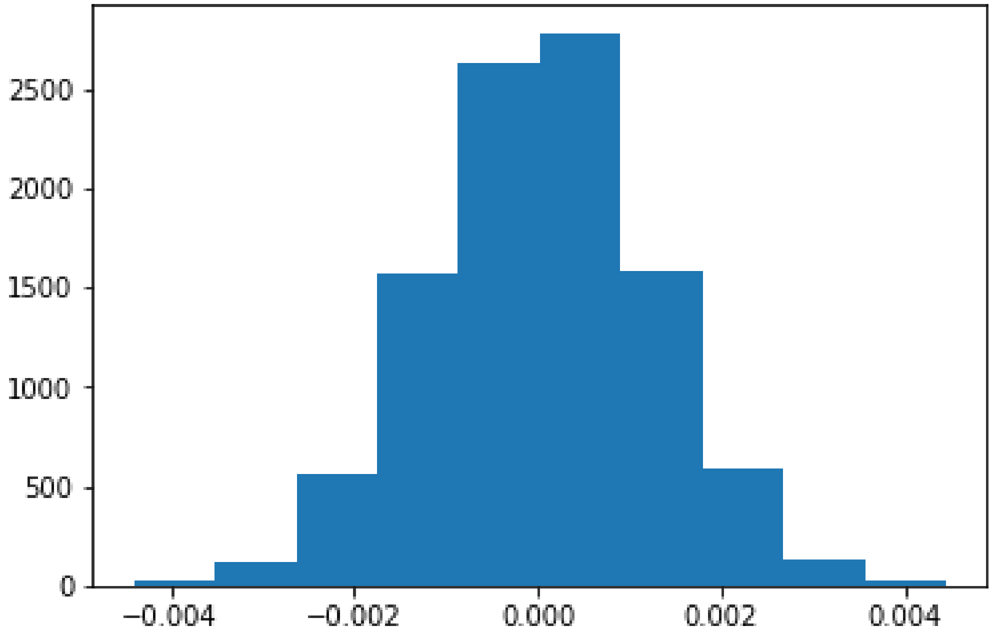
```
Out[26]: -0.0012752777719939878
```

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 [27]: #Running the simulation 10000 times
         p_diffs = []

         for x in range(10000):

             new_page_converted = np.random.choice([1, 0], size=n_new, p=[Pnew, (1-Pnew)])
             old_page_converted = np.random.choice([1, 0], size=n_old, p=[Pold, (1-Pold)])

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



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

```
In [29]: #Showing the obsereved difference
         diff = df2.query('group == "treatment"').converted.mean() - (df2.query('group == "contr
         (p_diffs > diff).mean()
```

```
Out[29]: 0.9068000000000005
```

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?

- The value is called P-value, and it has to be < 0.05 or > 0.95 so that the null hypothesis can be rejected, and as obsereved above our P-value does is not able 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 [30]: import statsmodels.api as sm

         convert_old = len(df2.query('landing_page == "old_page" and converted == True'))
         convert_new = len(df2.query('landing_page == "new_page" and converted == True'))
         n_old = df2.query('group == "control"').shape[0]
         n_new = df2.query('group == "treatment"').shape[0]
```

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

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

```
In [32]: print("z-score : {}".format(z_score))
         print("p-value : {}".format(p_value))
```

```
z-score : 1.3109241984234394
p-value : 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.**?

- z-score test further proofs what our p-value indicated, that we can not reject the null hypothesis because it is less than 1.6448.
- convertion is better from the old page than the new one, that is what our p-value idicates.
- Agreed with the findings in J and K

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 are predicting a categorical response, and we are going to predict two possible outcomes, therefore we are going to use **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 [33]: #Showing 1 row of the dataset
         df2.head(1)
```

```
Out[33]:    user_id                   timestamp    group landing_page  converted
         0   851104  2017-01-21 22:11:48.556739  control     old_page          0
```

```
In [34]: import statsmodels.api as sm

         #Creating an intercept column
```

```
df2['intercept'] = 1

#Creating a "ab_page" column which contains a 1 for treatment group and 0 for the contr
df2['ab_page'] = pd.get_dummies(df2['group'])['treatment']

df2.head()
```

Out[34]:

```
   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
0          1        0
1          1        0
2          1        1
3          1        1
4          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 [35]:

```
Logit_mod = sm.Logit(df2['converted'], df2[['intercept','ab_page']])

results= Logit_mod.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 [36]:

```
results.summary2()
```

Out[36]:

```
<class 'statsmodels.iolib.summary2.Summary'>
"""
                          Results: Logit
==================================================================
Model:              Logit            No. Iterations:   6.0000
Dependent Variable: converted        Pseudo R-squared: 0.000
Date:               2020-08-13 10:08 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**?

- P-value associated with ab_page is 0.1899.
- The difference comes from that in part 2 it was a one-sided test.
- but in part 3 it is two-sided test, two possible outcomes.

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?

- It is a good idea because we can explore different variables that might have an effect on the conversion rate (Response variable)
- One disadvantage is that we are going to have multiple regression model with two or more expantory variables, making it complex.

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 [37]: #Storing the countries.csv dataset into the variable
         df_countries = pd.read_csv('./countries.csv')
         df_countries.head()
```

```
Out[37]:    user_id country
         0   834778      UK
         1   928468      US
         2   822059      UK
         3   711597      UK
         4   710616      UK
```

```
In [38]: # Merging two datasets "df2" & "df_countries"
         df2_countries = pd.merge(df2, df_countries, how='inner', on='user_id')
         df2_countries.head()
```

```
Out[38]:    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
```

```
In [39]: df2_countries['country'].unique()
```

```
Out[39]: array(['US', 'CA', 'UK'], dtype=object)
```

```
In [40]: df2_countries['intercept'] = 1

         df2_countries[['US', 'UK']] = pd.get_dummies(df2_countries['country'])[['US', 'UK']]
         df2_countries.head()
```

```
Out[40]:    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
        0          1        0      US   1   0
        1          1        0      US   1   0
        2          1        1      US   1   0
        3          1        1      US   1   0
        4          1        0      US   1   0
```

```
In [41]: df['intercept'] = 1

         Logit_mod2 = sm.Logit(df2_countries['converted'], df2_countries[['intercept', 'US', 'UK
         results = Logit_mod2.fit()
```

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

```
In [44]: results.summary2()
```

Out[44]:

```
<class 'statsmodels.iolib.summary2.Summary'>
"""
                          Results: Logit
==================================================================
Model:              Logit            No. Iterations:   6.0000
Dependent Variable: converted        Pseudo R-squared: 0.000
Date:               2020-08-13 10:26 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     -2.0375    0.0260  -78.3639  0.0000  -2.0885  -1.9866
US             0.0408    0.0269    1.5178  0.1291  -0.0119   0.0935
UK             0.0507    0.0284    1.7863  0.0740  -0.0049   0.1064
==================================================================

"""
```

- We can not say that a country has any influnce on whether a person converts or not.
- Both countries P-value is > 0.05 and < 0.95 so it is true that we can not say that they have an effect on the convertion rate.

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 [45]:

```python
df2_countries.head()
```

Out[45]:

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

In [46]:

```python
df2_countries['US_new'] = df2_countries['US'] * df2_countries['ab_page']
df2_countries['UK_new'] = df2_countries['UK'] * df2_countries['ab_page']
```

```
df2_countries.head()
```

Out[46]:
```
   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  US_new  UK_new
0          1        0      US   1   0       0       0
1          1        0      US   1   0       0       0
2          1        1      US   1   0       1       0
3          1        1      US   1   0       1       0
4          1        0      US   1   0       0       0
```

In [47]:
```
df2['intercept'] = 1

Logit_mod3 = sm.Logit(df2_countries['converted'], df2_countries[['intercept', 'ab_page'
results = Logit_mod3.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.366109
         Iterations 6
```

In [49]:
```
results.summary2()
```

Out[49]:
```
<class 'statsmodels.iolib.summary2.Summary'>
"""
                          Results: Logit
==================================================================
Model:              Logit            No. Iterations:   6.0000
Dependent Variable: converted        Pseudo R-squared: 0.000
Date:               2020-08-13 10:34 AIC:              212782.6602
No. Observations:   290584           BIC:              212846.1381
Df Model:           5                Log-Likelihood:   -1.0639e+05
Df Residuals:       290578           LL-Null:          -1.0639e+05
Converged:          1.0000           Scale:            1.0000
-------------------------------------------------------------------
               Coef.  Std.Err.     z      P>|z|    [0.025   0.975]
-------------------------------------------------------------------
intercept     -2.0040   0.0364  -55.0077  0.0000  -2.0754  -1.9326
ab_page       -0.0674   0.0520   -1.2967  0.1947  -0.1694   0.0345
US             0.0175   0.0377    0.4652  0.6418  -0.0563   0.0914
US_new         0.0469   0.0538    0.8718  0.3833  -0.0585   0.1523
UK             0.0118   0.0398    0.2957  0.7674  -0.0663   0.0899
UK_new         0.0783   0.0568    1.3783  0.1681  -0.0330   0.1896
```

```
==============================================================================
"""
```

- The P-value for both "US_new" & "UK_new" is > 0.05 & < 0.95
- The P-value concludes that there is no significant effect on the conversion rate.

0.3 Conclusion

- Based on the probability, A/B testing, and Regression, and the P-values we got from these tests, we can say that we will be accepting the null hypothesis.
- As a result of accepting the null hypothesis, I sugest we should restrict our selfs to the old_page.

In [50]:
```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])
```

Out[50]: 0

In [ ]: