Project 3: Analyze A/B Test Results

Table of Contents

Introduction

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

For this project, we will be working to understand the results of an A/B test run by an e-commerce website. Our 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 we 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 that we are on the right track as we work through the project, and we can feel more confident in our final submission meeting the criteria.

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]:

```
df=pd.read_csv("ab_data.csv")
```

In [4]:

```
df.head()
```

Out[4]:

	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 [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294478 entries, 0 to 294477
Data columns (total 5 columns):
 #   Column        Non-Null Count   Dtype 
---  ------        --------------   ----- 
 0   user_id       294478 non-null  int64 
 1   timestamp     294478 non-null  object
 2   group         294478 non-null  object
 3   landing_page  294478 non-null  object
 4   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 [6]:

```
df.user_id.nunique()
```

Out[6]:

```
290584
```

d. The proportion of users converted.

In [7]:

```
df.converted.mean()
```

Out[7]:

```
0.11965919355605512
```

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

In [8]:

```
df[(df.group=="treatment") & (df.landing_page!="new_page")].user_id.count()+df[(df.group!="
```

Out[8]:

```
3893
```

f. Do any of the rows have missing values?
No

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
user_id         0
timestamp       0
group           0
landing_page    0
converted       0
dtype: int64
```

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.copy()
dele=df[((df.group=="treatment") & (df.landing_page!="new_page")) | ((df.group!="treatment"
df2.drop(dele.index,inplace=True)
df2.head()
```

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

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].shape[
```

Out[11]:

```
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 [12]:

```
df2.user_id.nunique()
```

Out[12]:

```
290584
```

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==773192]
```

Out[14]:

	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 [15]:

```
df2.drop(1899,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 [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[df2.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[df2.group=="treatment"].converted.mean()
```

Out[18]:

```
0.11880806551510564
```

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

In [19]:

```
(df2[df2.landing_page=="new_page"].user_id.count())/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.

Your answer goes here.

- The probability of an individual converting regardless of the page they receive is just 11.95%.
- The probability of an individual converting from control group is 12.03%
- The probability of an individual converting from treatment group is 11.88%
- The probability that an individual received the new page is 50.00%

No, we need to further testing to prove whether it's true or not.

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: p_old - p_new>=0
H1: p_old - p_new<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]:

```
p_new =df2[df2['converted']==1].user_id.count()/df2.shape[0]
p_new
```

Out[20]:

```
0.11959708724499628
```

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

In [21]:

```
p_old = df2[df2['converted']==1].user_id.count()/df2.shape[0]
p_old
```

Out[21]:

```
0.11959708724499628
```

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

In [22]:

```
n_new = df2[df2.group == "treatment"].user_id.nunique()
n_new
```

Out[22]:

```
145310
```

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

In [23]:

```
n_old = df2[df2.group == "control"].user_id.nunique()
n_old
```

Out[23]:

```
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([0,1],n_new, p=(p_new,1-p_new))
new_page_converted
```

Out[24]:

```
array([1, 1, 1, ..., 1, 1, 1])
```

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([0,1],n_new, p=(p_old,1-p_old))
old_page_converted
```

Out[25]:

```
array([0, 1, 0, ..., 1, 1, 1])
```

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

In [26]:

```
obs_diff = new_page_converted.mean()-old_page_converted.mean()
obs_diff
```

Out[26]:

```
-0.0006881838827333953
```

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]:

```
p_diffs = []
new_page_converted = np.random.binomial(n_new,p_new,10000)/n_new
old_page_converted = np.random.binomial(n_old,p_old,10000)/n_old
p_diffs = new_page_converted - old_page_converted
p_diffs
```

Out[27]:

```
array([-0.00029124,  0.00158823,  0.00058291, ...,  0.00107145,
       -0.00130938, -0.00113783])
```

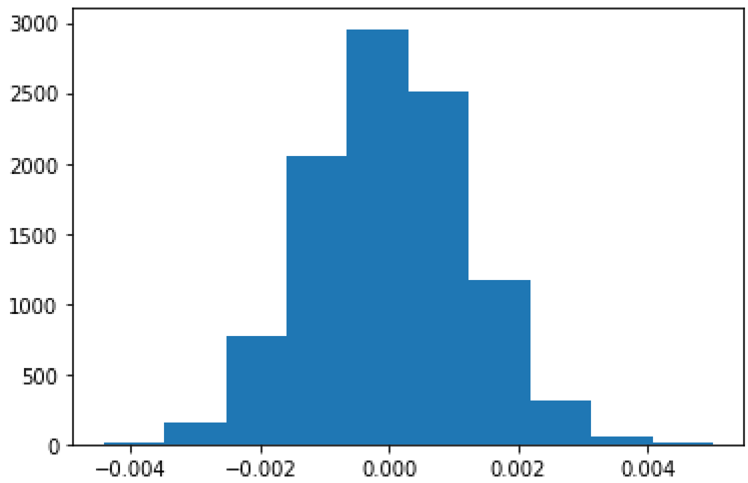
i. Plot a histogram of the **p_diffs**. Does this plot look like what you expected? Use the matching problem in the classroom to assure you fully understand what was computed here.

In [28]:

```
plt.hist(p_diffs)
```

Out[28]:

```
(array([  11.,  148.,  773., 2057., 2956., 2512., 1170.,  318.,   49.,
           6.]),
 array([-0.00442776, -0.00348412, -0.00254048, -0.00159684, -0.0006532 ,
         0.00029044,  0.00123407,  0.00217771,  0.00312135,  0.00406499,
         0.00500863]),
 <a list of 10 Patch objects>)
```



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

In [29]:

```
convert_new = df2[(df2.converted==1) & (df.landing_page == "new_page")].user_id.nunique()
convert_old = df2[(df2.converted==1) & (df.landing_page == "old_page")].user_id.nunique()
actual_new=convert_new/ n_new
actual_old=convert_old/n_old
obs_diff = actual_new - actual_old
obs_diff
```

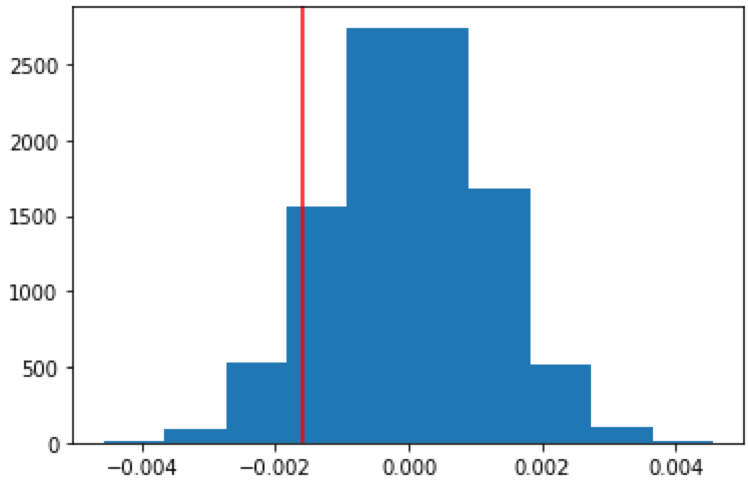
```
C:\Users\ABC\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarnin
g: Boolean Series key will be reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
C:\Users\ABC\anaconda3\lib\site-packages\ipykernel_launcher.py:2: UserWarnin
g: Boolean Series key will be reindexed to match DataFrame index.
```

Out[29]:

```
-0.0015782389853555567
```

In [30]:

```
null_vals = np.random.normal(0, np.std(p_diffs), np.array(p_diffs).size)
plt.hist(null_vals)
plt.axvline(x=obs_diff,color ='red');
```



In [31]:

```
(null_vals > obs_diff).mean()
```

Out[31]:

```
0.9086
```

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 obtained p-value is 0.9074 which is greater than alpha, we fail to reject the null hypothesis. Hence the data indicates that with a type I error rate of 0.05, the old page has higher probablity of convert rate than the new page.

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 [32]:

```
import statsmodels.api as sm

new = df2[(df2.converted == 1) & (df2.landing_page == "new_page")]['user_id'].nunique()
old = df2[(df2.converted == 1) & (df2.landing_page == "old_page")]['user_id'].nunique()
n_old = df2[df2.landing_page == "old_page"]['user_id'].nunique()
n_new = df2[df2.landing_page == "new_page"]['user_id'].nunique()
```

m. Now use `stats.proportions_ztest` to compute your test statistic and p-value. [Here (https://docs.w3cub.com/statsmodels/generated/statsmodels.stats.proportion.proportions_ztest/)](https://docs.w3cub.com/statsmodels/generated/statsmodels.stats.proportion.proportions_ztest/) is a helpful link on using the built in.

In [33]:

```
z_score,p_value = sm.stats.proportions_ztest(np.array([convert_new,convert_old]),np.array([
```

In [34]:

```
z_score, p_value
```

Out[34]:

```
(-1.3109241984234394, 0.9050583127590245)
```

In [35]:

```
from scipy.stats import norm
norm.cdf(z_score)
```

Out[35]:

```
0.09494168724097551
```

In [36]:

```
norm.ppf(1-(0.05/2))
```

Out[36]:

```
1.959963984540054
```

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 of 1.310 does not exceed the critical value of 1.959, therefore we failed to reject the null hypothesis that old page has a better or equal converted rate than old page. Yes, they agree with the findings in parts 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 each row is either a conversion or no conversion, a logistic regression should be performed..

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
```

In [38]:

```
df2= df2.join(pd.get_dummies(df2['landing_page']))
```

In [39]:

```
df2['ab_page']=pd.get_dummies(df['group'])['treatment']
```

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 [40]:

```
log_mod=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 [41]:

```
result = log_mod.fit()
result.summary()
```

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

Out[41]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	290584
Model:	Logit	**Df Residuals:**	290582
Method:	MLE	**Df Model:**	1
Date:	Thu, 28 May 2020	**Pseudo R-squ.:**	8.077e-06
Time:	06:27:33	**Log-Likelihood:**	-1.0639e+05
converged:	True	**LL-Null:**	-1.0639e+05
Covariance Type:	nonrobust	**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
ab_page	-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 **Part II**?`

- The p-value associated with the ab_page is 0.19. This is because the approach for the calculating the p-value is different for each case. In the first case, we calculated the probability receiving a observed statistic if the null hypothesis is true. Therefore this is a one-sided test.

- On the other hand the ab_page p-value is the result of a two sided test, because the null hypothesis for this case is, "there is no significant relationship between the conversion rate and ab_page".The alternate hypothesis is "there is significant relationship between the conversion rate and ab_page"

- Based p_value we can say, that the conversion is not significant dependent on the page

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 Factors:
- Difference in browsing time of each user, because of which the conversion rate may vary.
- We have categorical variable which includes "Morning", "Afternoon", and "Evening", or "Weekday and Weekend" which can affect conversion rate.
The main disadavantage for adding additional terms into regression model is makeS the model more

complex and creates complication in interpreting the model output. There is a possibility of having multi-collinearity and overfitting if these new variables are not taken care off.

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 (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.join.html)](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.join.html) 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 [42]:

```
countries = pd.read_csv('countries.csv')
```

In [43]:

```
countries.head()
```

Out[43]:

	user_id	country
0	834778	UK
1	928468	US
2	822059	UK
3	711597	UK
4	710616	UK

In [44]:

```
countries.country.value_counts()
```

Out[44]:

```
US    203619
UK     72466
CA     14499
Name: country, dtype: int64
```

In [45]:

```
#Join ab dataset with country dataset
df3 = df2.merge(countries, on ='user_id', how='left')
df3.head()
```

Out[45]:

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

In [46]:

```
df3 = df3.join(pd.get_dummies(df3['country']))
df3.head()
```

Out[46]:

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

In [47]:

```
logit_model = sm.Logit(df3['converted'], df3[['intercept','ab_page','CA','UK']])
result = logit_model.fit()
result.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.366113
         Iterations 6
```

Out[47]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	290584
Model:	Logit	**Df Residuals:**	290580
Method:	MLE	**Df Model:**	3
Date:	Thu, 28 May 2020	**Pseudo R-squ.:**	2.323e-05
Time:	06:27:39	**Log-Likelihood:**	-1.0639e+05
converged:	True	**LL-Null:**	-1.0639e+05
Covariance Type:	nonrobust	**LLR p-value:**	0.1760

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	-1.9893	0.009	-223.763	0.000	-2.007	-1.972
ab_page	-0.0149	0.011	-1.307	0.191	-0.037	0.007
CA	-0.0408	0.027	-1.516	0.130	-0.093	0.012
UK	0.0099	0.013	0.743	0.457	-0.016	0.036

Note: Considering p-values, we can say that countries doesn't have a significant impact on the coversion 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 [52]:

```
#Create a new interaction variable between ab page and country CA, US and UK
df3['CA_page'] = df3['ab_page']* df3['CA']
df3['UK_page'] = df3['ab_page']* df3['UK']
df3['US_page'] = df3['ab_page']* df3['US']
df3.head(2)
```

Out[52]:

	user_id	timestamp	group	landing_page	converted	intercept	new_page	old_page	a
0	851104	2017-01-21 22:11:48.556739	control	old_page	0	1	0	1	
1	804228	2017-01-12 08:01:45.159739	control	old_page	0	1	0	1	

In [50]:

```
logit_model2=sm.Logit(df3['converted'], df3[['intercept','ab_page','CA','UK','CA_page','UK_
result = logit_model2.fit()
result.summary()
```

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

Out[50]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	290584
Model:	Logit	**Df Residuals:**	290578
Method:	MLE	**Df Model:**	5
Date:	Thu, 28 May 2020	**Pseudo R-squ.:**	3.482e-05
Time:	06:27:43	**Log-Likelihood:**	-1.0639e+05
converged:	True	**LL-Null:**	-1.0639e+05
Covariance Type:	nonrobust	**LLR p-value:**	0.1920

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	-1.9865	0.010	-206.344	0.000	-2.005	-1.968
ab_page	-0.0206	0.014	-1.505	0.132	-0.047	0.006
CA	-0.0175	0.038	-0.465	0.642	-0.091	0.056
UK	-0.0057	0.019	-0.306	0.760	-0.043	0.031
CA_page	-0.0469	0.054	-0.872	0.383	-0.152	0.059
UK_page	0.0314	0.027	1.181	0.238	-0.021	0.084

Note: Considering p-values we can conclude that these features doesn't have a significant impact on conversion rate.

Conclusion

We failed to reject the null hypothesis that old page has a better or equal converted rate than old page. Therefore we can say there won't be significant positive change if the the company go for the the new page. In worst case scenario, it may lead to fewer convert ratio than before and result in loss of the company.