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!

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]:

```
# openning the ab_data.csv file and store it in df
df = pd.read_csv('ab_data.csv')
# reading the first five rows of the data set
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

user_id	timestamp	group	landing_page	converted

b. Use the cell below to find the number of rows in the dataset.

In [3]:

```
# finding the number of rows in the data set
df.shape[0]
```

Out[3]:

```
294478
```

c. The number of unique users in the dataset.

In [4]:

```
# finding the number of unique users in the data set
df['user_id'].nunique()
```

Out[4]:

```
290584
```

d. The proportion of users converted.

In [5]:

```
# finding the proportion of unique users who converted (bought the company's product)
num_unique=df['user_id'].nunique()

df.query('converted == "1"').user_id.nunique() / num_unique
```

Out[5]:

```
0.12104245244060237
```

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

In [6]:

```
# finding the number of times the new_page and treatment_page didn't match
df.query('group == "treatment" and landing_page=="old_page"').user_id.count() + df.query('group ==
"control" and landing_page=="new_page"').user_id.count()
```

Out[6]:

```
3893
```

f. Do any of the rows have missing values?

In [7]:

```
# checking if there are rows with missing values
df.isnull().sum().any()
```

Out[7]:

```
False
```

`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 [8]:

```
# deleting rows where treatment does not match with new_page or control does not match with old_pa
ge
# by finding the rows indexes where treatment does not match with new_page
index1 = df[ (df['group'] == "treatment") & (df['landing_page'] =="old_page")].index
# and then dropping rows of these indexes
df2=df.drop(index1 )
# and by finding the rows indexes where control does not match with old_page
index2 = df[ (df['group'] == "control") & (df['landing_page'] =="new_page")].index
# and then dropping rows of these indexes
# and finally storing the new data frame in df2
df2=df2.drop(index2 )
```

In [9]:

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

Out[9]:

```
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 [10]:

```
# finding the number of unique users in the new data set
df2['user_id'].nunique()
```

Out[10]:

```
290584
```

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

In [11]:

```
# finding the repeated user_id in the new data set
df2[df2.duplicated(['user_id'], keep=False)]
```

Out[11]:

	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 [12]:

```
# finding the row information for the repeated user_id
df2.query('user_id=="773192"')
```

Out[12]:

	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**.

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

In [13]:

```
# removing one of the rows with the duplicated user_id "773192"
df2 = df2.drop(2893)
```

In [14]:

```
# checking that the row with the specified index is removed from the data set
df2.query('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

`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 [15]:

```
# finding the probability of an individual converting regardless
# of the page they receive
df2.query('converted=="1"').user_id.count() / df2.shape[0]
```

Out[15]:

```
0.11959708724499628
```

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

In [16]:

```
# finding the probability that an individual from the control group is converted
p_old =(df2.query('group=="control"')['converted']==1).mean()
p_old
```

Out[16]:

```
0.1203863045004612
```

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

In [17]:

```
# finding the probability that an individual from the treatment group is converted
p_new =(df2.query('group=="treatment"')['converted']==1).mean()
p_new
```

Out[17]:

```
0.11880806551510564
```

In [18]:

```
# checking the conversion difference between the new page and the old page
stat_diffs = p_new - p_old
stat_diffs
```

Out[18]:

```
-0.0015782389853555567
```

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

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

In [19]:

```
# finding the propbaility that an individual received the new page
df2.query('landing_page=="new_page"').user_id.count() / df2.shape[0]
```

Out[19]:

```
0.50006194422266881
```

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 propbability of converting regardless of the page is nearly 0.12. And the propbabilities of converting for both control and treatment groups is also nearly 0.12. Therefore , I don't see sufficient evidence to conclude that the new treatment page leads to more conversions than the old page since the propbabilities small differences can't be considered as significant practicly. They all have an approximate propbability of 0.12. Furthermore, based on the last propbability that receiving the new or the old page is 0.5 which means a fair chance for the two pages to be displayed to users , I don't see any unfairness that mights affect the results. Therefore, based on the above reasons, I fail to reject the null hypothesis that the propbability of the conversion rate if the user receieved the new page is less than or equal to the propbability of the conversion rate if the user receieved the old page.

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.

- The Null Hypotheses is H0: Pnew - Pold <=0
- The Alternative Hypotheses is 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]:

```
# the conversion rate for pnew under the null hypothesis
p_new = df2['converted'].mean()
p_new
```

Out[20]:

```
0.11959708724499628
```

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

In [21]:

```
# the conversion rate for pold under the null hypothesis
p_old = df2['converted'].mean()
p_old
```

Out[21]:

```
0.11959708724499628
```

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

In [22]:

```
# the number of individuals in the treatment group
n_new =df2[(df2['group'] == 'treatment')].user_id.count()
n_new
```

Out[22]:

```
145310
```

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

In [23]:

```
# the number of individuals in the control group
n_old= df2[(df2['group'] == 'control')].user_id.count()
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]:

```
# simulating n_new ( number of people in the treatment group) transactions with a conversion rate
of p_new under the null
# hypothesis and then storing the resulted 1's and 0's in new_page_converted
new_page_converted = np.random.choice([0,1],size=n_new,p=[(1-p_new),p_new])
# checking that the values are computed
new_page_converted
```

Out[24]:

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

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]:

```
# simulating n_old ( number of people in the control group) transactions with a conversion rate of
p_old under the null
# hypotheses and then storing the resulted 1's and 0's in old_page_converted
old_page_converted = np.random.choice([0,1],size=n_old,p=[(1-p_old),p_old])
```

```
# checking that the values are computed
old_page_converted
```

Out[25]:

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

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

In [26]:

```
# finding the new probability of conversion for the new page ( p_new)
new_page_converted.mean()
```

Out[26]:

```
0.12022572431353658
```

In [27]:

```
# finding the new probability of conversion for the old page (p_old)
old_page_converted.mean()
```

Out[27]:

```
0.11999394248110468
```

In [28]:

```
# finding the difference between p_new and p_old for the simulated values
# from part (e) and (f)
new_page_converted.mean()- old_page_converted.mean()
```

Out[28]:

```
0.00023178183243190154
```

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 [29]:

```
#Creating 10,000  p_new and p_old values using the same simulation process used in parts (a) throu
gh (g) above
#finding the difference between p_new and p_old and
#storing all 10,000 values in a numpy array called p_diffs
p_diffs=[]
for _ in range(10000):
    new_page_convertedsim = np.random.choice([0,1],size=n_new,p=[(1-p_new),p_new]).mean()
    old_page_convertedsim = np.random.choice([0,1],size=n_old,p=[(1-p_old),p_old]).mean()
    p_diffs.append(new_page_convertedsim-old_page_convertedsim)
```

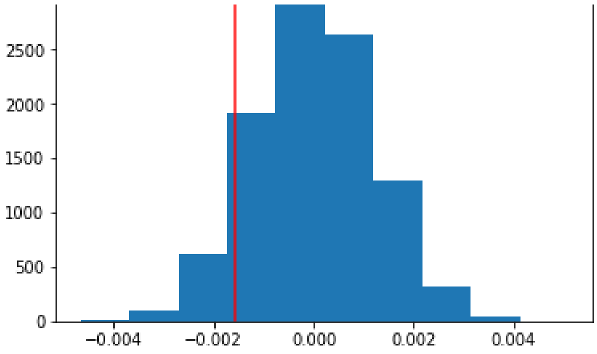
i. Plot a histogram of the **p_diffs**. Does this plot look like what you expected? Use the matching problem in the classroom to assure you fully understand what was computed here.

In [30]:

```
# plotting the p_diffs based on the null hypotheses
plt.hist(p_diffs)
# ploting line for the observed statistic - from partI
plt.axvline(stat_diffs, c='red')
```

Out[30]:

```
<matplotlib.lines.Line2D at 0x7f2900e91b00>
```



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

In [31]:

```
# computing the p-value
p_diffs = np.array(p_diffs)
(p_diffs > stat_diffs).mean()
```

Out[31]:

```
0.9062999999999999
```

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 above histogram displays the observed statistic (the difference between the new page conversion rates and the the old page conversion rates) from the data set compared to the computed conversion rates difference from the null hypothesis (that the propbability of the conversion rate if the user receieved the new page is less than or equal to the propbability of the conversion rate if the user receieved the old page).

Then the sampling distribuation from the null hypothesis was compared to the observed statistic to obtain the proportion of conversion rates difference that are greater than the conversion rates difference observed from the data set.

This computed proportion is called the p-value. The p-value is the probability of observing the statistic observed from the data (in this case the difference between the the conversion rates from the new page and old page) in favor of the alternative hypotheses if the null hypothesis is true.

The alterantive hypothesis : (that the propbability of the conversion rate if the user recieved the new page is greater than the propbability of the conversion rate if the user recieved the old page).

The null hypothesis : (that the propbability of the conversion rate if the user receieved the new page is less than or equal to the propbability of the conversion rate if the user receieved the old page).

Based on the fact that this study accepts Type I error rate of 5% :

I can compare the p-value and the Type I error rate of 5% :

p-value = 0.9

Type I error rate = 0.05

0.9 > 0.05

Since the p-value is greater than the Type I error rate , I fail to reject the null hypothesis that the propbability of the conversion rate if the user receieved the new page is less than or equal to the propbability of the conversion rate if the user receieved the old page. The p-value states that the observed statistic is likely from the null hypothesis.

Therefore , I stay with the null hypothesis as my decision , so I recommened Audacity to stay with the old 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

# finding the number of poeple from the control group who converted
convert_old = df2[ (df2['group'] == "control") & (df2['converted'] ==1)].user_id.count()
# finding the number of people from the treatment group who converted
convert_new = df2[ (df2['group'] == "treatment") & (df2['converted'] ==1)].user_id.count()
# finding the number of people who recieved the old page
n_old = df2[(df2['group']=="control")].user_id.count()
# finding the number of people who recieved the new page
n_new = df2[(df2['group']=="treatment")].user_id.count()
```

```
/opt/conda/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The
pandas.core.datetools module is deprecated and will be removed in a future version. Please use the
pandas.tseries module instead.
  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 [33]:

```
# finding the z-score and the p-value
z_score, p_value = sm.stats.proportions_ztest([convert_new, convert_old], [n_new, n_old] , alternat
ive='larger')
z_score, p_value
```

Out[33]:

```
(-1.3109241984234394, 0.90505831275902449)
```

In [34]:

```
from scipy.stats import norm
# how significant the z-score is
norm.cdf(z_score)
```

Out[34]:

```
0.094941687240975514
```

In [35]:

```
# what is the critical value at 95% confidence is
norm.ppf(1-(0.05/2))
```

Out[35]:

```
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.**?

Since the z-score of -1.31 doesn't exceed the critical value range of 1.96 , I fail to reject the null hypothesis that the propbability of the conversion rate if the user receieved the new page is less than or equal to the propbability of the conversion rate if the user receieved the old page.

Furthermore, since the p-value of 0.9 (I can say it's nearly the same as the p-value comupted in part j) is larger than the Type I error rate of 0.05 , I also fail to reject the previous null hypothesis.

Based on these two conclusions , there is no difference between the new page and the old page conversion rates , so no need to implment the new page. Audacity should keep the old page. In addition , these findings are similar to the findings of part 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 in this statistical study, I am dealing only with a qualitive variable (There is a conversion or no conversion represented by 1 and 0 respectively), I will need to use the Logistic Regression to predict the qualitative response. The qualitative response will be bound between 0 and 1.

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 [36]:

```
# preparing for the logistic regression
# adding intercept column to the data set
# adding a dummy varibale column for which page each user received [ 1: when an individual receives the treatment] and
# [0: when an individual receives the control]
df2['intercept']=1
df2[['ba_page','ab_page']]=pd.get_dummies(df2['group'])
```

In [37]:

```
# checking that the new columns are added to the data set
df2.head()
```

Out[37]:

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

In [38]:

```
# dropping one of the genreted columns for the dummy variable
df2=df2.drop('ba_page',axis=1)
```

In [39]:

```
# checking the modified data frame
df2.head()
```

Out[39]:

	user_id	timestamp	group	landing_page	converted	intercept	ab_page
0	851104	2017-01-21 22:11:48.556739	control	old_page	0	1	0
1	804228	2017-01-12 08:01:45.159739	control	old_page	0	1	0
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0	1	1
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0	1	1
4	864975	2017-01-21 01:52:26.210827	control	old_page	1	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 [40]:

```
# using statsmodels to instantiate regression model on the two columns created previously
# using statsmodels to instantiate regression model on the two columns created previously
logit_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]:

```
# fitting the model using the two columns created to predict whether or not an individual converts
results = logit_mod.fit()
results.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, 21 Feb 2019	**Pseudo R-squ.:**	8.077e-06
Time:	16:11:05	**Log-Likelihood:**	-1.0639e+05
converted:	True	**LL-Null:**	-1.0639e+05
		LLR p-value:	0.1899

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
intercept	-1.9888	0.008	-246.669	0.000	-2.005	-1.973
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 ab_page is 0.19 while the p-value found in Part II from the sampling distribution was nearly 0.9 and from the z-test was nearly 0.9. The difference between the p values exists because in the logistic regression a dependent variable (conversion rate) was studied against another independent variable (page type: new or old) to predict if there is a significant difference in conversion based on which page a customer receives. The null hypothesis can be here: there is no significant difference in conversion based on which page a customer receives, and the alternative hypothesis: there is a significant difference in conversion based on which page a customer receives.

Mathematically : H0: Pnew = Pold , H1: Pnew != Pold

On the other hand, in the sampling distribution only the existing of the difference between the conversion rate of the new page and the old page without considering any influence from the page type. The null hypothesis was: that the propbability of the conversion rate if the user received the new page is less than or equal to the propbability of the conversion rate if the user received the old page. And the alternative hypothesis: that the propbability of the conversion rate if the user received the new page is greater than the propbability of the conversion rate if the user received the old page. I can say then that the factor that affected the p-value is that in the simple distribuation study , the new page was specified before the old page to test the difference between them in the null hypothesis. However , in the logistic regression study , only the relationship between the page type and the conversion rate without specifying the pages as in the first study in the null hypothesis were considered.

Mathematically : H0: Pnew - Pold <= 0 , H1: Pnew - Pold > 0

In addition :

Based on the p-value of 0.19 compared to the error rate of 0.05, the p-value is larger than the error rate. Thus, I fail to reject the null hypothesis that there is no significant difference in conversion based on which page a customer receives.

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 to consider other factors into the regression to have a clear picture about what might influence the dependent variable other than the independent variables studied. For example , if the user country is considered as an independent variable , there may exist a relationship between the user country and the conversion rate. The user might be living in a poor country with weak economic , so the conversion rate will probably be very low . As a result , Audacity might need to work on displaying the new website page to a more stable countries and see the results. On the other hand , there are some disadvantages to adding additional terms into the regression model. Multicollinearity is one of the disadvantage. It's when there are independent variables that are correlated with one another. One of the main concerns of multicollinearity is that it can lead to changing the test results and manipulating the studied relationship. In addtion, there is the problem of outliers that might come with more variables studied. The outliers can increase the spread of data and manipulate the results. If the data was pulled from multpile sources. It's possible that some data values are incorrect.

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 [42]:

```
# storing the countries data set in df_countries to make it ready for the merge with df2 (data set
)
df_countries = pd.read_csv("countries.csv")
```

In [43]:

```
# joining the df2 and df_countries data sets together
df2=df2.join(df_countries.set_index("user_id"),on="user_id",how='inner')
```

In [44]:

```
# checking the join of the two data sets
df2.head()
```

Out[44]:

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

In [45]:

```
# checking how many countries are there in the data set
df2['country'].unique()
```

Out[45]:

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

In [46]:

```
# creating dummay variables for the countries column
df2[['CA','UK','US']]=pd.get_dummies(df2['country'])
```

In [47]:

```
# checkign the new changes to the data set
# checkign the new changes to the data set
df2.head()
```

Out[47]:

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

In [48]:

```
# using statsmodels to instantiate regression model on the two columns created previously
logit_mod = sm.Logit(df2['converted'],df2[['intercept','ab_page','CA','UK']])
```

In [49]:

```
# fitting the model using the two columns created to predict whether or not an individual converts
results = logit_mod.fit()
results.summary()
```

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

Out[49]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	290584
Model:	Logit	**Df Residuals:**	290580
Method:	MLE	**Df Model:**	3
Date:	Thu, 21 Feb 2019	**Pseudo R-squ.:**	2.323e-05
Time:	16:11:21	**Log-Likelihood:**	-1.0639e+05
converged:	True	**LL-Null:**	-1.0639e+05
		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

In [50]:

```
# to have an accurate coefficients , exponentiating is used for each coefficient
np.exp(-0.0408),np.exp(0.0099)
```

Out[50]:

```
(0.9600211497165088, 1.0099491671175422)
```

Based on the previous logistic regression and the fact that US is used to compare the results(meaning based on US) , for each user from Canada he or she is likely to convert by 0.96 times than a user from US holding all else constant. And for each user from UK , he or she is likely to convert by 1.01 times than a user from US holding all else constant. All the p-values are not equal to zero which mean they are not statistically significant , so there is no relationship between the user country and whether if he or she will convert. Therefore, based on these conclusions , I can say that there is no statistically significant relationship between the user's country and the conversion 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 [51]:

```
# to look at the interaction between page and country , the user country column is multyplied by t
he ab_page
df2['CA_new']= df2['CA']*df2['ab_page']
df2['UK_new'] = df2['UK']*df2['ab_page']
```

In [52]:

```
# to check changes made to the data set
df2.head()
```

Out[52]:

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

In [53]:

```
# using statsmodels to instantiate regression model to study the interaction between page and coun
try to see if there is
# significant effect on conversion
logit_mod = sm.Logit(df2['converted'],df2[['intercept','ab_page','CA','UK','CA_new','UK_new']])
```

In [54]:

```
# fitting the logistic model
results = logit_mod.fit()
results.summary()
```

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

Out[54]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	290584
Model:	Logit	**Df Residuals:**	290578
Method:	MLE	**Df Model:**	5
Date:	Thu, 21 Feb 2019	**Pseudo R-squ.:**	3.482e-05
Time:	16:11:29	**Log-Likelihood:**	-1.0639e+05
converged:	True	**LL-Null:**	-1.0639e+05
		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

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_new	-0.0469	0.054	-0.872	0.383	-0.152	0.059
UK_new	0.0314	0.027	1.181	0.238	-0.021	0.084

From the above table summary , I can conclude from the p-values (0.383 , 0.238) that are not equal to zero that there is no significance evidence that the interaction between page and country have a significant effect on conversion.

Conclusion

Based on the probability study , there was no difference between the conversion rate of the new page and the old page. Therefore , there wasn't sufficient evidence to conclude that the new treatment page leads to more conversions than the old page . The probability of converting was almost the same regardless of the page. As a result , there isn't sufficient evidence to conclude that the new treatment page leads to more conversions than the old page.

Based on the sampling distribuation and the study of the statistic compared to the null hypothesis , the p-value was greater than the Type I error rate , so there was a fail to reject the null hypothesis that the propbability of the conversion rate if the user receieved the new page is less than or equal to the propbability of the conversion rate if the user receieved the old page. The p-value states that the observed statistic is likely from the null hypothesis.

Based on the z-score test , the z-score was in the critical value range , so there was a fail to reject the null hypothesis that the propbability of the conversion rate if the user receieved the new page is less than or equal to the propbability of the conversion rate if the user receieved the old page. Furthermore, the p-value was larger than the Type I error rate of 0.05 , so there was a fail to reject the also the null hypothesis.

Based on the logistic regression that studied if there is a significant difference in conversion based on which page a customer receives , the p-value was larger than the error rate of 0.05. Thus, there was a fail to reject the null hypothesis that there is no significant difference in conversion based on which page a customer receives.

Based on the second logistic regression that studied if there is a statistically significant relationship between the user's country and the conversion rate , all the p-values were not equal to zero which mean they are not statistically significant. As a result, there is no relationship between the user country and the conversion rate. Therefore, based on these conclusions , I can say that there is no statistically significant relationship between the user's country and the conversion rate.

Based on the last logisitc regression that studied if the interaction between page and country to see if there is significant effects on conversion , the p-values were not equal to zero , and that means that there is no significance evidence that the interaction between page and country have a significant effect on conversion.

Statistically and based on these previous results , I fail to reject the null hypothesis that the propbability of the conversion rate if the user receieved the new page is less than or equal to the propbability of the conversion rate if the user receieved the old page.

However there are some practical factors for Audacity to consider like :

- **Novelty effect and change aversion when existing users first experience a change. Some users might reject the change while other can get exicted about it.**
- **Running the experiment for a long enough time to account for changes in behavior based on time of day/week or seasonal events.**
- **Practical significance of a conversion rate (the cost of launching the new page vs. the gain from the increase in conversion)**.

In [55]:

```
# to check the runtime for the experiment of launching the new page and the old page
df2['timestamp'].max(),df2['timestamp'].min()
```

Out[55]:

```
('2017-01-24 13:41:54.460509', '2017-01-02 13:42:05.378582')
```

It's obvious from the runtime of the experiment of 23 days that if Audacity wants to truly get reliable data to run the

experiment with a larger time frame to take their final decision!

Refrences

Link 1

Link 2

Link 3

Link 4

Link 5

Link 6

Link 7

Link 8

Link 9

Link 10

Link 11

Link 12

In [97]:

```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Analyze_ab_test2_results_notebook.ipynb'])
```

Out[97]:

```
255
```

In [ ]: