Analyze A/B Test Results

Table of Contents

Introduction

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

In [3]:
```
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 [4]:
```
df['user_id'].nunique()
```

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

d. The proportion of users converted.

In [5]:
```
df[df['converted']==1].shape[0]/df.shape[0]
```

Out[5]:
```
0.11965919355605512
```

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

In [6]:
```
a=df[df['landing_page']=='new_page']
b=df[df['group']=='treatment']
(a[a['group']!='treatment'].shape[0])+(b[b['landing_page']!='new_page'].shape
```

Out[6]:
```
3893
```

f. Do any of the rows have missing values?

In [7]:
```
#no
```

2. For the rows where **treatment** is not aligned with **new_page** or **control** is not aligned with **old_page**, we cannot be sure if this row truly received the new or old page. Use **Quiz 2** in the classroom to provide 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]:
```python
a=df[df['landing_page']=='new_page']
ind1=a[a['group']!='treatment'].index

b=df[df['group']=='treatment']
ind2=b[b['landing_page']!='new_page'].index

df2=df
df2.drop(ind1,inplace=True)
df2.drop(ind2,inplace=True)
```

In [9]:
```python
 a=df[df['landing_page']=='old_page']
ind3=a[a['group']!='control'].index

b=df[df['group']=='control ']
ind4=b[b['landing_page']!='old_page'].index

df2.drop(ind3,inplace=True)
df2.drop(ind4,inplace=True)
```

In [10]:
```python
# Double Check all of the correct rows were removed - this should be 0
df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) ==
```

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]:
```python
df['user_id'].nunique()
```

Out[11]: 290584

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

In [12]:
```python
df2[df2.duplicated(['user_id'], keep=False)]['user_id']
```

Out[12]:
```
1899    773192
2893    773192
Name: user_id, dtype: int64
```

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

In [13]:
```python
df2[df2.duplicated(['user_id'], keep=False)]
```

Out[13]:

	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 [14]:
```
df.drop(index=1899,inplace=True)
```

4. Use **df2** in the below cells 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]:
```
df['converted'].mean()
```

Out[15]: 0.11959708724499628

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

In [16]:
```
df[df['group']=='control']['converted'].mean()
```

Out[16]: 0.1203863045004612

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

In [17]:
```
df[df['group']=='treatment']['converted'].mean()
```

Out[17]: 0.11880806551510564

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

In [18]:
```
df[df['landing_page']=='new_page'].shape[0]/df.shape[0]
```

Out[18]: 0.5000619442226688

e. Consider your results from a. through d. above, and explain below whether you think there is sufficient evidence to say that the new treatment page leads to more conversions.

Your answer goes here.

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.

Put your answer here.

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.

In [19]:
```
df.sample(1)
```

Out[19]:

	user_id	timestamp	group	landing_page	converted
233349	677983	2017-01-06 16:48:35.600166	treatment	new_page	0

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

In [20]:
```
p_new = df2.converted.mean()
p_new
```

Out[20]: 0.11959708724499628

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

In [21]:
```
p_old = df2.converted.mean()
p_old
```

Out[21]: 0.11959708724499628

c. What is n_{new}?

In [22]:
```
n_new=df2.landing_page.value_counts()[0]
n_new
```

Out[22]: 145310

d. What is n_{old}?

In [23]:
```
n_old= df2.landing_page.value_counts()[1]
n_old
```

Out[23]: 145274

e. Simulate n_{new} transactions with a convert 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(2, size=n_new ,p=[p_new,1 - p_new])
new_page_converted.mean()
```

Out[24]: 0.8814534443603331

f. Simulate n_{old} transactions with a convert 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(2, size=n_old ,p=[p_old,1 - p_old])
old_page_converted.mean()
```

Out[25]: 0.8805567410548343

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

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

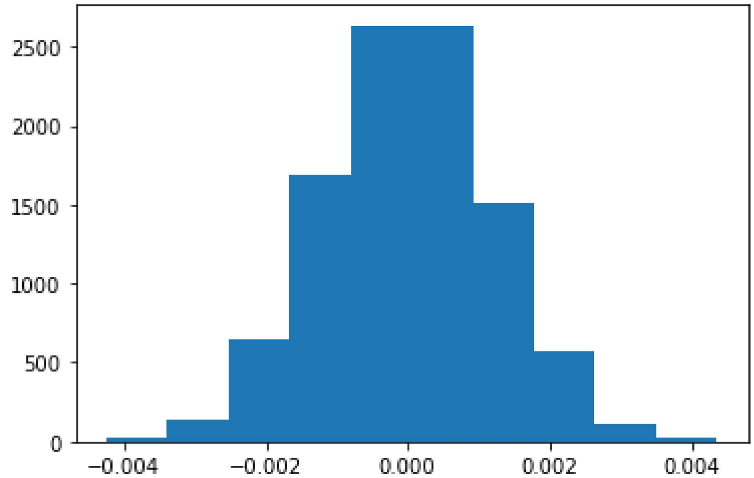
Out[26]: 0.0008967033054988471

h. Simulate 10,000 p_{new} - p_{old} values using this same process similarly to the one you calculated in parts **a. through g.** above. Store all 10,000 values in a numpy array called **p_diffs**.

In [27]:
```
p_diffs, bs_new_means, bs_old_means = [], [], []
for _ in range (10000):
    bs_new = np.random.choice(2, size=n_new ,p=[p_new,1 - p_new])
    bs_old = np.random.choice(2, size=n_old ,p=[p_old,1 - p_old])
    bs_new_means.append(bs_new.mean())
    bs_old_means.append(bs_old.mean())
    p_diffs.append(bs_new.mean() - bs_old.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 [30]:
```
p_diffs = np.array(p_diffs)
plt.hist(p_diffs);
```

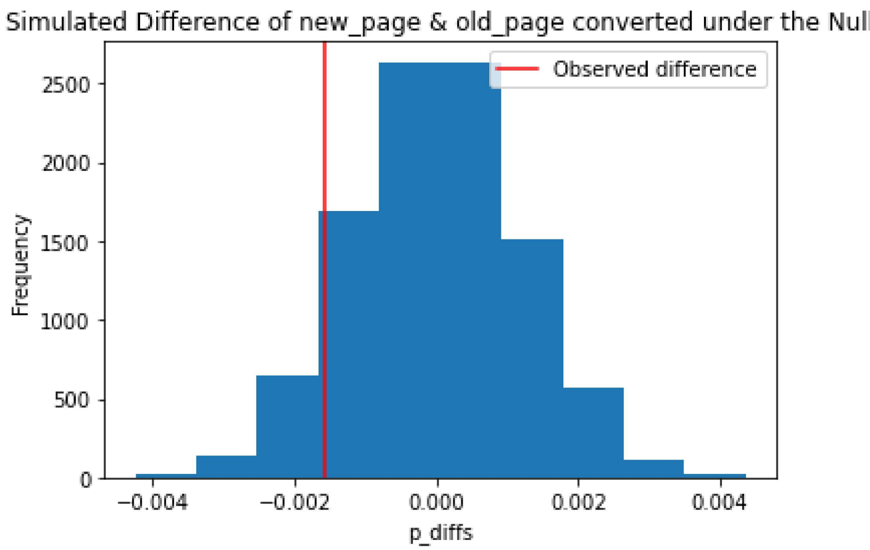


j. What proportion of the **p_diffs** are greater than the actual difference observed in **ab_data.csv**?

In [32]:
```
actual_diff = df2.query('landing_page == "new_page"').converted.mean() - \
                        df2.query('landing_page == "old_page"').converted.mear
(p_diffs>actual_diff).mean()
```

Out[32]: 0.905

In [33]:
```
# plot line for observed statistic
plt.hist(p_diffs)
plt.axvline(x=actual_diff, color='r', label="Observed difference")
plt.xlabel('p_diffs')
plt.ylabel('Frequency')
plt.title('Simulated Difference of new_page & old_page converted under the Nu
plt.legend()
plt.show()
```



k. In words, explain 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?

Put your answer here.

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 [36]:
```
n_new = len(df2.query('landing_page == "new_page"'))
n_old = len(df2.query('landing_page == "old_page"'))
convert_new = len(df2.query('landing_page == "new_page" & converted == 1'))
convert_old = len(df2.query('landing_page == "old_page" & converted == 1'))
```

In [37]:
```
(convert_new, convert_old)
```

Out[37]: (17264, 17489)

In [38]:
```
n_new, n_old
```

Out[38]: (145310, 145274)

m. Now use `stats.proportions_ztest` to compute your test statistic and p-value. [Here](#)

(http://knowledgetack.com/python/statsmodels/proportions_ztest/) is a helpful link on using the built in.|

In [42]:

```
import statsmodels.api as sm
z_score, p_value = sm.stats.proportions_ztest([convert_old, convert_new], [n_

z_score, p_value
```

Out[42]: (1.3109241984234394, 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.**?

Put your answer here.

Part III - A regression approach

`1.` In this final part, you will see that the result you acheived in the previous A/B test can also be acheived 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?

Put your answer here.

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 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 [43]:

```
df2[['control','treatment']]= pd.get_dummies(df2['group'])
df2 = df2.drop('control',axis = 1)

df2.head()
```

Out[43]:

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

In [45]:
```python
df2 = df2.rename(columns={'treatment': 'ab_page'})
df2.head()
```

Out[45]:

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

c. Use **statsmodels** to import your regression model. Instantiate the model, and fit the model using the two columns you created in part **b.** to predict whether or not an individual converts.

In [46]:
```python
from scipy import stats
stats.chisqprob = lambda chisq, df3: stats.chi2.sf(chisq, df3)

df2['intercept'] = 1

lm = sm.Logit(df2['converted'],df2[['intercept','ab_page']])
results = lm.fit()
results.summary()
```

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

Out[46]:

Logit Regression Results

Dep. Variable:	converted	**No. Observations:**	290584
Model:	Logit	**Df Residuals:**	290582
Method:	MLE	**Df Model:**	1
Date:	Thu, 26 Aug 2021	**Pseudo R-squ.:**	8.077e-06
Time:	20:32:29	**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

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

Summary: Holding all other variables constant, the number of converted is 1.015 times more likely to be converted than those that are not converted. This means that the old page and new page are both equal in chance of converting users. We should not assume that the new page is better than the old page.

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 the **Part II**?

Answer:

The p-value found in the logistic regression model(0.19) is different than what we found in parts j and k because our null and alternative hypthesis model assumed that there is an equal probability of the old and new page converting users. In the logistic regression model, this is not the case. Also, the Logistic Regression performed is a two-tailed test, whereas the computation done in Part II is a one-tailed 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?

Answer:

Other factors that influence whether an individual converts could be age. Older users may prefer more information on the pages as opposed to a kid, where they may prefer more pictures and a more casual theme. Adding more factors into the regression model will increase or decrease confidence intervals. A disadvantage of multiple factors in a logistic regression model is that it reduces the power of analysis.

g. Now along with testing if the conversion rate changes for different pages, also add an effect based on which country a user lives. You will need to read in the **countries.csv** dataset and merge together your datasets on the approporiate 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 [49]:
```
df3 = pd.read_csv('countries.csv')
df3.head()
```

Out[49]:

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

In [50]:
```
# create dummy variables for country column
df3[['CA','UK', 'US']] = pd.get_dummies(df3['country'])
# drop the country column since this is not necessary
df3 = df3.drop('country', 1)
df3.head()
```

Out[50]:

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

In [53]:
```
new_df = df2.join(df3.set_index('user_id'), on='user_id')
new_df.head()
```

Out[53]:

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

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 [54]:
```
new_df['US_ab_page'] = new_df['US'] * new_df['ab_page']
new_df['UK_ab_page'] = new_df['UK'] * new_df['ab_page']
new_df['CA_ab_page'] = new_df['CA'] * new_df['ab_page']
```

In [55]:
```
lm = sm.Logit(new_df['converted'], new_df[['intercept', 'US_ab_page', 'UK_ab_
results = lm.fit()
results.summary2()
```

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

Out[55]:

Model:	Logit	Pseudo R-squared:	0.000
Dependent Variable:	converted	AIC:	212778.9383
Date:	2021-08-26 20:36	BIC:	212821.2568
No. Observations:	290584	Log-Likelihood:	-1.0639e+05
Df Model:	3	LL-Null:	-1.0639e+05
Df Residuals:	290580	LLR p-value:	0.067853
Converged:	1.0000	Scale:	1.0000
No. Iterations:	6.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
intercept	-1.9888	0.0081	-246.6690	0.0000	-2.0046	-1.9730
US_ab_page	-0.0183	0.0126	-1.4486	0.1475	-0.0430	0.0064
UK_ab_page	0.0074	0.0180	0.4098	0.6819	-0.0279	0.0427
CA_ab_page	-0.0827	0.0380	-2.1763	0.0295	-0.1571	-0.0082

Conclusions

Conclusions Based on the statistical tests we used, the Z-test, logistic regression model, and actual difference observed, the results have shown that the new and old page have an approximately equal chance of converting users. We fail to reject the null hypothesis. I recommend to the e-commerce company to keep the old page. This will save time and money on creating a new page.