Analyze A/B Test Results

This project will assure you have mastered the subjects covered in the statistics lessons. The hope is 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 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 w through this notebook to help the company understand if they should implement the new page, keep experiment longer to make their decision.

As you work through this notebook, follow along in the classroom and answer the corresponding qu each question. The labels for each classroom concept are provided for each question. This will assu you work through the project, and you can feel more confident in your final submission meeting the c you meet all the criteria on the RUBRIC.

Part I - Probability

To get started, let's import our libraries.

```
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 ir**

a. Read in the dataset and take a look at the top few rows here:

```
df= pd.read_csv("ab_data.csv")
df.head(5)
```

	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.

```
df.shape[0]
```

```
294478
```

c. The number of unique users in the dataset.

```
len(df.user_id.unique())
```

```
290584
```

d. The proportion of users converted.

```
x= (df[['converted']]==1).sum()
y= (df[['converted']]==0).sum()
percent= ((x)/(x+ y))*100
percent
```

```
converted    11.965919
dtype: float64
```

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

```
df2 = df.query("(group == 'control' and landing_page == 'new_page') or (group == 'treatment
df2.shape[0]
```

```
3893
```

f. Do any of the rows have missing values?

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

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

2. For the rows where **treatment** is not aligned with **new_page** or **control** is not aligned with **old_pag** truly received the new or old page. Use **Quiz 2** in the classroom to provide how we should handle thes

a. Now use the answer to the quiz to create a new dataset that meets the specifications from the qui **df2**.

```
df2 = df.query("(group == 'control' and landing_page == 'old_page') or (group == 'treatment

# 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
```

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**?

```
df2.user_id.nunique
```

```
<bound method IndexOpsMixin.nunique of 0         851104
1         804228
2         661590
3         853541
4         864975
           ...
294473    751197
294474    945152
294475    734608
294476    697314
294477    715931
Name: user_id, Length: 290585, dtype: int64>
```

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

```
df2[df2.duplicated(['user_id'])]['user_id'].unique()
```

```
array([773192])
```

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

```
df2[df2.duplicated(['user_id'], keep=False)]
```

	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**.

```
df2 = df2.drop_duplicates(['user_id'], keep='first')
```

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?

```
x= (df2[['converted']]==1).sum()
y= (df2[['converted']]==0).sum()
percent= ((x)/(x+ y))
percent
```

```
converted    0.119597
dtype: float64
```

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

```
control_group= df2[df2['group']=='control']
x= (control_group[['converted']]==1).sum()
y= (control_group[['converted']]==0).sum()
percent= ((x)/(x+ y))
percent
```

```
converted    0.120386
dtype: float64
```

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

```
treatment_group= df2[df2['group']=='treatment']
x= (treatment_group[['converted']]==1).sum()
y= (treatment_group[['converted']]==0).sum()
percent= ((x)/(x+ y))
percent
```

```
converted    0.118808
dtype: float64
```

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

```
x= (df2[['landing_page']]=="new_page").sum()
y= (df2[['landing_page']]=="old_page").sum()
percent= ((x)/(x+ y))
percent
```

```
landing_page    0.500062
dtype: float64
```

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

- **The control group converted at a slightly higher rate that the treatment group**
- **Probability of a person of received the new page is .50 it indicates that it is not possible for th conversion based on being given more opportunities to do so**

Part II - A/B Test

Notice that because of the time stamp associated with each event, you could technically run a hypotl observation was observed.

However, then the hard question is do you stop as soon as one page is considered significantly bette happen consistently for a certain amount of time? How long do you run to render a decision that neit

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 t better unless the new page proves to be definitely better at a Type I error rate of 5%, what should youı hypotheses be? You can state your hypothesis in terms of words or in terms of p_{old} and p_{new}, which 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 **con** of page - that is p_{new} and p_{old} are equal. Furthermore, assume they are equal to the **converted** rate iı 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 estimate from the null.

Use the cells below to provide the necessary parts of this simulation. If this doesn't make complete s you are going to work through the problems below to complete this problem. You can use **Quiz 5** in tl are on the right track.

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

```
p_new = df2['converted'].mean()
p_new
```

```
0.11959708724499628
```

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

```
p_old = df2['converted'].mean()
p_old
```

```
0.11959708724499628
```

c. What is n_{new}?

```
n_new = len(df2.query("landing_page == 'new_page'"))
n_new
```

```
145310
```

d. What is n_{old}?

```
n_old = len(df2.query("landing_page == 'old_page'"))
n_old
```

```
145274
```

e. Simulate n_{new} transactions with a convert rate of p_{new} under the null. Store these n_{new} 1's and 0'

```
new_page_converted = np.random.binomial(n_new,p_new)
new_page_converted
```

```
17568
```

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 = np.random.binomial(n_old,p_old)
old_page_converted
```

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

```
(new_page_converted/n_new) - (old_page_converted/n_old)
```

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

```
p_diffs = []
for _ in range(10000):
    new_page_converted = np.random.binomial(n_new,p_new)
    old_page_converted = np.random.binomial(n_old, p_old)
    p_diff = new_page_converted/n_new - old_page_converted/n_old
    p_diffs.append(p_diff)
```

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

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

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

```
p_diff_orig = df[df['landing_page'] == 'new_page']['converted'].mean() -  df[df['landing_pa
p_diffs = np.array(p_diffs)
p_diff_proportion = (p_diff_orig < p_diffs).mean()
p_diff_proportion
```

k. In words, explain what you just computed in part **j.** What is this value called in scientific studies? W
terms of whether or not there is a difference between the new and old pages?

- **If null hypothesis is true pvalue gives the probability of statistics tested.**
- **In this case, the new page doesn't have better conversion rates than the old page**

l. We could also use a built-in to achieve similar results. Though using the built-in might be easier to c
walkthrough of the ideas that are critical to correctly thinking about statistical significance. Fill in the
of conversions for each page, as well as the number of individuals who received each page. Let `n_ol`
number of rows associated with the old page and new pages, respectively.

```
import statsmodels.api as sm

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

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

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

n. What do the z-score and p-value you computed in the previous question mean for the conversion ra

```
from scipy.stats import norm
print(norm.cdf(z_score))
print(norm.ppf(1-(0.05)))
```

zscore is a measure of how many standard deviations below or above the population mean a raw sco
of 1.3109241984234394 is less than the critical value of 1.6448536269514722 which means we can
can conclude that old page conversions are slightly better than new page conversions. Eventhough th
findings in parts j. and k but it suggests there is no significant difference between old page and new p

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 ach

a. Since each row is either a conversion or no conversion, what type of regression should you be perf

Logistic Regression

b. The goal is to use **statsmodels** to fit the regression model you specified in part **a.** to see if there is
conversion based on which page a customer receives. However, you first need to create a column for
dummy variable column for which page each user received. Add an **intercept** column, as well as an **a**
when an individual receives the **treatment** and 0 if **control**.

```
df2['intercept'] = 1
df2[['control', 'ab_page']]=pd.get_dummies(df2['group'])
df2.drop(labels=['control'], axis=1, inplace=True)
df2.head()
```

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

```
import statsmodels.api as sm
import scipy.stats as stats
logit = sm.Logit(df2['converted'],df2[['intercept' ,'ab_page']])
results = logit.fit()
```

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

```
stats.chisqprob = lambda chisq, df: stats.chi2.sf(chisq, df)
results.summary()
```

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 d
alternative hypotheses in the **Part II**?

**p-value is 0.190.The p-value here suggests that that new page is not statistically significant as 0.19
sided test and in this section it was a two sided test. Here we test for not equal in our hypotheses w
different.**

f. Now, you are considering other things that might influence whether or not an individual converts. D
consider other factors to add into your regression model. Are there any disadvantages to adding add
regression model?

There can be multiple factors that affect whether or not an individual converts. Factors like age can ensure that it's best fit it's always better to include more features however at the same time we shou features since we do not want to overfit

g. Now along with testing if the conversion rate changes for different pages, also add an effect basec You will need to read in the **countries.csv** dataset and merge together your datasets on the appropor joining tables.

Does it appear that country had an impact on conversion? Don't forget to create dummy variables for **You will need two columns for the three dummy variables.** Provide the statistical output as well as a question.

```
countries_df = pd.read_csv('./countries.csv')
df_new = countries_df.set_index('user_id').join(df2.set_index('user_id'), how='inner')
```

```
df_new[['CA','UK','US']]=pd.get_dummies(df_new['country'])
mod = sm.Logit(df_new['converted'], df_new[['intercept', 'CA', 'UK']])
results = mod.fit()
results.summary()
```

h. Though you have now looked at the individual factors of country and page on conversion, we woul interaction between page and country to see if there significant effects on conversion. Create the nec fit the new model.

Provide the summary results, and your conclusions based on the results.

```
### Fit Your Linear Model And Obtain the Results
mod = sm.Logit(df_new['converted'], df_new[['intercept', 'CA', 'UK','ab_page']])
results = mod.fit()
```

```
results = mod.fit()
results.summary()
```

bold text

Conclusions

We can accept Null Hypothesis as there is no significant difference in conversion rates. We can rejec results are based on given dataset. There may be limitations due to incorrect data or missing columr