

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](#)
- [Part I - Probability](#)
- [Part II - A/B Test](#)
- [Part III - Regression](#)

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 [189... 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 [190... df=pd.read_csv("ab_data.csv")
df.head(10)
```

```
Out[190... 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
5  936923  2017-01-10 15:20:49.083499  control    old_page         0
6  679687  2017-01-19 03:26:46.940749  treatment  new_page         1
7  719014  2017-01-17 01:48:29.539573  control    old_page         0
8  817355  2017-01-04 17:58:08.979471  treatment  new_page         1
9  839785  2017-01-15 18:11:06.610965  treatment  new_page         1
```

```
In [191... time_duration = pd.to_datetime(df['timestamp'].max()) - pd.to_datetime(df['timestamp'].min())
print('A/B Test data available for {} days'.format(time_duration))
```

A/B Test data available for 21 days 23:59:49.081927 days

```
In [192... df.groupby(['group', 'landing_page']).count()
```

```
Out[192... user_id  timestamp  converted
```

group	landing_page	user_id	timestamp	converted
control	new_page	1928	1928	1928
	old_page	145274	145274	145274
treatment	new_page	145311	145311	145311
	old_page	1965	1965	1965

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

```
In [193... df.shape
print("Number of rows are: ", df.shape[0])
```

Number of rows are: 294478

c. The number of unique users in the dataset.

```
In [194... print("The number of unique users in the dataset is : ", df['user_id'].nunique())
```

The number of unique users in the dataset is : 290584

d. The proportion of users converted.

```
In [195... proportion = (df.query('converted==1')['converted'].count()/df.shape[0]) * 100
print("\nThe proportion of users converted: ", proportion, "%")
```

The proportion of users converted: 11.96591935560551 %

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

```
In [196... incorrect_lineup = len(df.query("group == 'treatment' and landing_page == 'old_page')) + len(df.query("group == 'control' and landing_page == 'new_page'))
print("The number of times the new_page and treatment don't line up is: ", incorrect_lineup)
```

The number of times the new_page and treatment don't line up is: 3893

f. Do any of the rows have missing values?

```
In [197... print(df.isnull().sum(), "\n")
print("Since sum of null values is zero for all the columns, none of the rows have missing values.\n")
```

```
user_id      0
timestamp    0
```

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

Since sum of null values is zero for all the columns, none of the rows have missing values.

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 [198... df2=pd.read_csv("ab_data.csv")
df2.drop(df.query("group == 'treatment' and landing_page == 'old_page').index, inplace=True)
df2.drop(df.query("group == 'control' and landing_page == 'new_page').index, inplace=True)
df2.shape
```

```
Out[198... (290585, 5)
```

```
In [199... # 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[199... 0
```

3. Use **df2** and the cells below to answer questions for **Quiz3** in the classroom.

a. How many unique **user_ids** are in **df2**?

```
In [200... print("Number of unique user_ids is: ", df2['user_id'].nunique())

#print('\n\n', df2[df2['user_id'].duplicated()])
```

```
Number of unique user_ids is: 290584
```

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

```
In [201... print("The repeated user_id is: ", df2[df2['user_id'].duplicated()]['user_id'].values[0])
```

```
The repeated user_id is: 773192
```

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

```
In [202... duplicate_user_id = df2[df2['user_id'].duplicated()]['user_id'].values[0]
df2[df2.user_id == duplicate_user_id]
```

	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 [203... print("Before remove: ", df2.shape)
df2 = df2.drop(df2[df2['user_id'].duplicated()].index[0])
print("After remove: ", df2.shape)
```

```
Before remove: (290585, 5)
After remove: (290584, 5)
```

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 [204... df2['converted'].mean()
```

```
Out[204... 0.11959708724499628
```

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

```
In [205... df2.query('group == "control"')['converted'].mean()
```

```
Out[205... 0.1203863045004612
```

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

```
In [206... df2.query('group == "treatment"')['converted'].mean()
```

```
Out[206... 0.11880806551510564
```

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

```
In [207... df2.query('landing_page == "new_page"')['landing_page'].count() / df2['landing_page'].count()
```

```
Out[207... 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.

As per the findings above, the probability of conversion for the treatment group is less than that for the control group. So, it can not be concluded that the treatment page leads to more conversions.

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.

- $H_0 : p_{old} \geq p_{new}$
- $H_1 : p_{old} < p_{new}$

Or, we can say:

- $H_0 : p_{new} - p_{old} \leq 0$
- $H_1 : p_{new} - p_{old} > 0$

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

```
Out[208... 0.11959708724499628
```

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

```
In [209... p_old = df2['converted'].mean()  
p_old
```

```
Out[209... 0.11959708724499628
```

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

```
In [210... n_new = df2.query('group == "treatment"')['user_id'].count()  
n_new
```

```
Out[210... 145310
```

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

```
In [211... n_old = df2.query('group == "control"')['user_id'].count()  
n_old
```

```
Out[211... 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 [212... new_page_converted = np.random.binomial(1, p_new, n_new)  
new_page_converted
```

```
Out[212... 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 [213... old_page_converted = np.random.binomial(1, p_old, n_old)
old_page_converted
```

```
Out[213... array([0, 0, 0, ..., 0, 0, 1])
```

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

```
In [214... new_page_converted = new_page_converted[:145274]
sim_diff = new_page_converted.mean() - old_page_converted.mean()
sim_diff
```

```
Out[214... -0.00032352657736414714
```

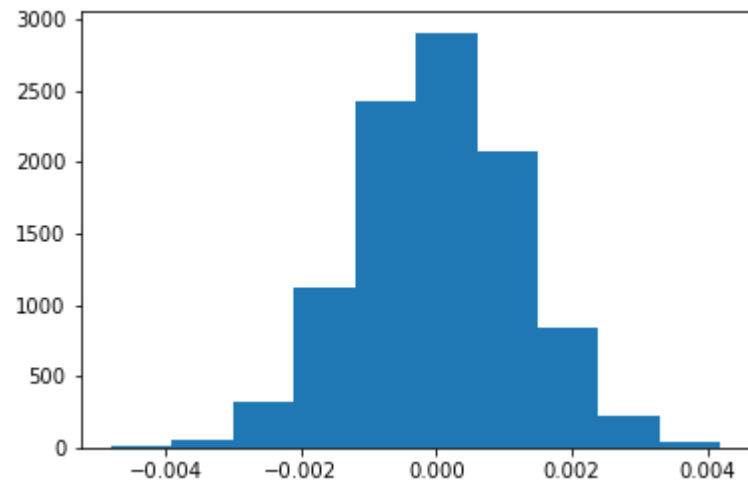
h. Create 10,000 $p_{new} - p_{old}$ values using the same simulation process you used in parts (a) through (g) above. Store all 10,000 values in a NumPy array called **p_diffs**.

```
In [215... p_diffs = []

for _ in range(10000):
    new_page_converted = np.random.binomial(1, p_new, n_new)
    old_page_converted = np.random.binomial(1, p_old, n_old)
    new_page_p = new_page_converted.mean()
    old_page_p = old_page_converted.mean()
    p_diffs.append(new_page_p - old_page_p)
```

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 [216... plt.hist(p_diffs);
```

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

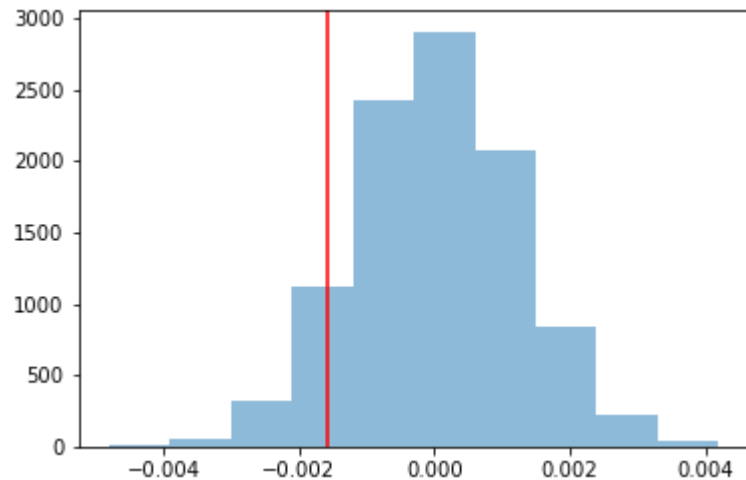
```
In [217... actual_diff = (df2[df2['group'] == "treatment"]['converted'].mean()) - (df2[df2['group'] == "control"]['converted'].mean())
actual_diff
```

```
Out[217... -0.0015782389853555567
```

```
In [218... ddff=pd.DataFrame({ 'p_diffs': p_diffs })
p_value = (actual_diff < ddff['p_diffs']).mean()
p_value
```

```
Out[218... 0.911
```

```
In [219... plt.hist(p_diffs, alpha=.5)
plt.axvline(x=actual_diff, color='red');
```



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?

Put your answer here.

- In part j we calculated what proportion of the `p_diffs` array is greater than the actual difference. That is the actual difference from the dataset between the control group conversion rate and the treatment group conversion rate. The `p_value` is 0.9051.
- In scientific studies the definition of the p-value is "The p-value is the level of marginal significance within a statistical hypothesis test representing the probability of the occurrence of a given event. The p-value is used as an alternative to rejection points to provide the smallest level of significance at which the null hypothesis would be rejected."
- We computed a `p_value` of 0.9051 which means that 90.51% of the differences can be attributed to pure chance. `p_diffs` (difference in means) is higher than the actual difference we see in our dataset. This was only a simulation. Not the actual data. The actual difference in the data is only higher than ~ 9% of the simulated difference assuming both pages conversion rates are the same. This is well above the .05 null hypothesis threshold. This is not a significant result. So, I fail to reject the null hypothesis.

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

In [220...

```
import statsmodels.api as sm

convert_old = df2.query('group == "control" & converted == 1')['converted'].count()
```

```
convert_new = df2.query('group == "treatment" & converted == 1')['converted'].count()
n_old = df2.query('group == "control"')['user_id'].count()
n_new = df2.query('group == "treatment"')['user_id'].count()

convert_old, convert_new, n_old, n_new
```

Out[220... (17489, 17264, 145274, 145310)

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 [221... z_score, p_value = sm.stats.proportions_ztest([convert_new, convert_old], [n_new, n_old], alternative='larger')
z_score, p_value
```

Out[221... (-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.

A negative z-score suggests and the value of p-value suggests that we should fail to reject the null hypothesis.

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?

Put your answer here.

Logistic regression

b. The goal is to use **statsmodels** to fit the regression model you specified in part **a.** to see if there is a significant difference in conversion based on which page a customer receives. However, you first need to create in `df2` a column for the intercept, and create a dummy variable column for which page each user received. Add an **intercept** column, as well as an **ab_page** column, which is 1 when an individual receives the **treatment** and 0 if **control**.

```
In [222... #Create intercept column
df2['intercept']=1
```

```
#Create dummies
ab_page = ['treatment', 'control']
df2['ab_page'] = pd.get_dummies(df2.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 [223... logit = 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 [224... results = logit.fit()
results.summary()
```

Optimization terminated successfully.

Current function value: 0.366118

Iterations 6

Out[224...

Logit Regression Results

Dep. Variable:	converted	No. Observations:	290584
Model:	Logit	Df Residuals:	290582
Method:	MLE	Df Model:	1
Date:	Sun, 23 May 2021	Pseudo R-squ.:	8.077e-06
Time:	09:57:09	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**?

Put your answer here.

The logistic regression determines only two possible outcomes. If the new page is equal to the old page or different.

- $H_0 : p_{new} - p_{old} = 0$
- $H_1 : p_{new} - p_{old} \neq 0$

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?

Put your answer here.

We could consider introducing the timestamp metric to determine in which part of the day the individuals converted the most. For example, if we find that the evening is the period that users spend most of their time on the internet we might also take it into consideration.

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 [225... 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.head()
```

```
Out[225...      country      timestamp      group  landing_page  converted  intercept  ab_page
user_id
834778      UK  2017-01-14 23:08:43.304998    control    old_page         0         1         0
928468      US  2017-01-23 14:44:16.387854    treatment    new_page         0         1         1
822059      UK  2017-01-16 14:04:14.719771    treatment    new_page         1         1         1
711597      UK  2017-01-22 03:14:24.763511    control    old_page         0         1         0
710616      UK  2017-01-16 13:14:44.000513    treatment    new_page         0         1         1
```

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 [226... df_new[['CA', 'UK', 'US']] = pd.get_dummies(df_new['country'])[['CA', 'UK', 'US']]
df_new.sample(10)
```

Out[226...

	country	timestamp	group	landing_page	converted	intercept	ab_page	CA	UK	US
user_id										
849838	US	2017-01-16 14:07:29.962778	treatment	new_page	0	1	1	0	0	1
783503	UK	2017-01-19 01:48:15.270835	control	old_page	0	1	0	0	1	0
867937	US	2017-01-20 06:16:52.333549	treatment	new_page	0	1	1	0	0	1
691545	US	2017-01-21 09:48:53.910803	control	old_page	0	1	0	0	0	1
729737	US	2017-01-23 00:01:54.490731	control	old_page	1	1	0	0	0	1
882328	UK	2017-01-17 08:04:23.715137	treatment	new_page	0	1	1	0	1	0
739100	US	2017-01-08 06:21:55.235259	treatment	new_page	0	1	1	0	0	1
694357	CA	2017-01-23 10:40:13.960919	control	old_page	0	1	0	1	0	0
692898	US	2017-01-16 09:33:21.219137	control	old_page	0	1	0	0	0	1
830289	US	2017-01-23 17:12:40.719069	control	old_page	1	1	0	0	0	1

```
In [227... mod = sm.Logit(df_new['converted'], df_new[['intercept', 'ab_page', 'US', 'CA']])
results = mod.fit()
results.summary()
```

Optimization terminated successfully.
Current function value: 0.366113
Iterations 6

Logit Regression Results

Dep. Variable:	converted	No. Observations:	290584
Model:	Logit	Df Residuals:	290580
Method:	MLE	Df Model:	3
Date:	Sun, 23 May 2021	Pseudo R-squ.:	2.323e-05
Time:	09:57:10	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	-1.9794	0.013	-155.415	0.000	-2.004	-1.954
ab_page	-0.0149	0.011	-1.307	0.191	-0.037	0.007
US	-0.0099	0.013	-0.743	0.457	-0.036	0.016
CA	-0.0506	0.028	-1.784	0.074	-0.106	0.005

Conclusions

According to the analysis performed I found that the old page was better than the new page, therefore I fail to reject the null hypothesis. Moreover, the histogram shows that the new page is not better than the old page.

From the regression above we see that the p-value is higher in US than in Canada, which means that users in the US are more likely to convert, but still not enough evidence to reject the null hypothesis.

Should we run these tests longer to come to a final conclusion?

Ideal duration for A/B tests is 1-3 weeks or maybe more for some cases. As per my finding, we have data for 21+ days. So, I don't think there is any need of running the test for longer duration.

Do we need more variables?

In my opinion, this dataset contains all the required variables. So, there is no need of more variables.

Do we need more data?

As per my findings from various other sites, there should be minimum 100 conversions per variation before taking any decision. In this case, we have more than 1000 conversions for each group and landing_page category. So, I don't think the test requires more data.

Finishing Up

Congratulations! You have reached the end of the A/B Test Results project! You should be very proud of all you have accomplished!

Tip: Once you are satisfied with your work here, check over your report to make sure that it satisfies all the areas of the rubric (found on the project submission page at the end of the lesson). You should also probably remove all of the "Tips" like this one so that the presentation is as polished as possible.

Directions to Submit

Before you submit your project, you need to create a .html or .pdf version of this notebook in the workspace here. To do that, run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the orange Jupyter icon in the upper left).

Alternatively, you can download this report as .html via the **File > Download as** submenu, and then manually upload it into the workspace directory by clicking on the orange Jupyter icon in the upper left, then using the Upload button.

Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right here. This will create and submit a zip file with this .ipynb doc and the .html or .pdf version you created. Congratulations!

```
In [228... from subprocess import call
call(['python', '-m', 'nbconvert', 'Analyze_ab_test_results_notebook.ipynb'])
```

```
Out[228... 1
```