

Analyze A/B Test Results

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](https://review.udacity.com/#!/projects/37e27304-ad47-4eb0-a1ab-8c12f60e43d0/rubric) (<https://review.udacity.com/#!/projects/37e27304-ad47-4eb0-a1ab-8c12f60e43d0/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]: 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]: len(df.index)
```

```
Out[3]: 294478
```

c. The number of unique users in the dataset.

```
In [4]: len(df["user_id"].unique())
```

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

d. The proportion of users converted.

```
In [5]: df.converted.mean()
```

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

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

```
In [6]: df.query("group == 'treatment' and landing_page == 'old_page').shape[0]+df.query(
```

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

f. Do any of the rows have missing values?

```
In [7]: df.isnull().sum()
```

```
Out[7]: 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_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]: df2=df[((df['group'] == 'treatment') == (df['landing_page'] == 'new_page')) == True]
```

```
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]
```

```
Out[9]: 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 [10]: df2.user_id.unique().shape[0]
```

```
Out[10]: 290584
```

```
In [11]: df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 290585 entries, 0 to 294477  
Data columns (total 5 columns):  
#   Column          Non-Null Count  Dtype    
---  ---            -  
0   user_id         290585 non-null int64    
1   timestamp       290585 non-null object   
2   group           290585 non-null object   
3   landing_page    290585 non-null object   
4   converted       290585 non-null int64    
dtypes: int64(2), object(3)  
memory usage: 13.3+ MB
```

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

```
In [12]: df2['user_id'].duplicated().sum()
```

```
Out[12]: 1
```

```
In [13]: df2.user_id.value_counts()
```

```
Out[13]: 773192      2
          851104      1
          688307      1
          718297      1
          838144      1
          ..
          755610      1
          804629      1
          837875      1
          889019      1
          715931      1
          Name: user_id, Length: 290584, dtype: int64
```

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

```
In [14]: df2[df2['user_id'].duplicated()]
```

```
Out[14]:
```

	user_id	timestamp	group	landing_page	converted
2893	773192	2017-01-14 02:55:59.590927	treatment	new_page	0

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

```
In [15]: # Remove one of the rows with a duplicate user_id..
# Hint: The dataframe.drop_duplicates() may not work in this case because the rows
df2 = df2[df2.user_id != 773192]
# Check again if the row with a duplicate user_id is deleted or not
df2.user_id.duplicated().sum()
```

```
Out[15]: 0
```

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 [16]: df2.converted.mean()
```

```
Out[16]: 0.11959749882133504
```

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

```
In [17]: c_prob=df2.query('group=="control"')['converted'].mean()
          c_prob
```

```
Out[17]: 0.1203863045004612
```

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

```
In [18]: t_prob=df2.query('group=="treatment"')['converted'].mean()  
t_prob
```

```
Out[18]: 0.11880888313869065
```

```
In [19]: # Calculate the actual difference (obs_diff) between the conversion rates for the t  
obs_diff = t_prob - c_prob  
obs_diff
```

```
Out[19]: -0.0015774213617705535
```

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

```
In [20]: len(df2.query('landing_page=="new_page"'))/len(df2)
```

```
Out[20]: 0.5000602237570677
```

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.

I think there is no sufficient evidence to say that. Because when I compare average converted belongs of control and treatment groups, 0.12 and 0.118 are similar rates. So we can not say that according to this probabilities.

Part II - A/B Test

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

However, then the hard question is do you stop as soon as one page is considered significantly better than another or does it need to happen consistently for a certain amount of time? How long do you run to render a decision that neither page is better than another?

These questions are the difficult parts associated with A/B tests in general.

1. For now, consider you need to make the decision just based on all the data provided. If you want to assume that the old page is better unless the new page proves to be definitely better at a Type I error rate of 5%, what should your null and alternative hypotheses be? You can state your hypothesis in terms of words or in terms of p_{old} and p_{new} , which are the converted rates for the old and new pages.

H0: $p_{new} - p_{old} \leq 0$

H1: $p_{new} - p_{old} > 0$ (right tailed)

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 **convert rate** for p_{new} under the null?

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

```
Out[21]: 0.11959749882133504
```

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

```
In [22]: p_old = df2['converted'].mean()  
p_old
```

```
Out[22]: 0.11959749882133504
```

c. What is n_{new} ?

```
In [23]: n_new = df2[df2['group'] == 'treatment'].shape[0]  
n_new
```

```
Out[23]: 145309
```

d. What is n_{old} ?

```
In [24]: n_old = df2[df2['group'] == 'control'].shape[0]  
n_old
```

```
Out[24]: 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 [25]: # Simulate a Sample for the treatment Group
new_page_converted = np.random.choice([1, 0], size=n_new, p=[p_new, (1-p_new)])
new_page_converted
```

```
Out[25]: array([0, 0, 0, ..., 0, 0, 0])
```

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 [26]: # Simulate a Sample for the control Group
old_page_converted = np.random.choice([1, 0], size=n_old, p=[p_old, (1-p_old)])
old_page_converted
```

```
Out[26]: array([0, 0, 0, ..., 1, 0, 1])
```

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

```
In [27]: new_page_converted.mean() - old_page_converted.mean()
```

```
Out[27]: -0.0011300390711643793
```

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 [30]: # Sampling distribution
p_diffs = []
for _ in range(10000):

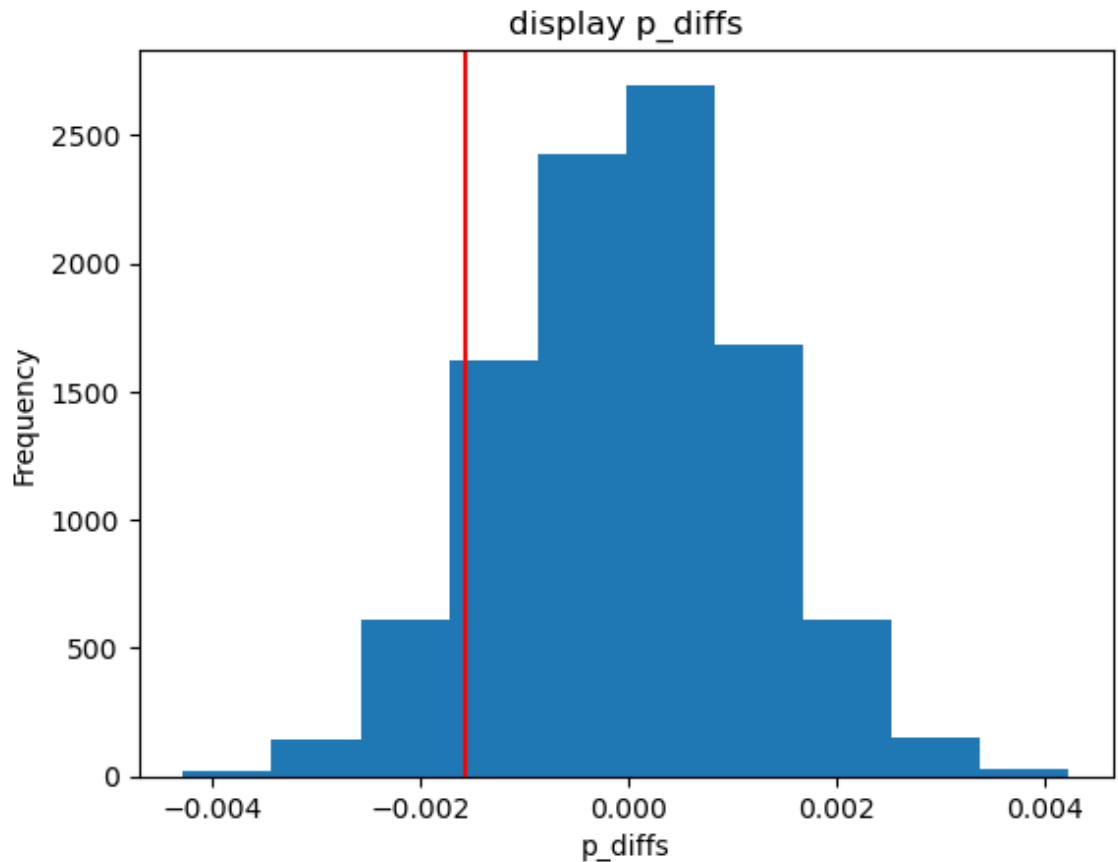
    new_page_converted = np.random.choice([1, 0], size=n_new, p=[p_new, (1-p_new)])
    old_page_converted = np.random.choice([1, 0], size=n_old, p=[p_old, (1-p_old)])
    p_diffs.append(new_page_converted.mean() - old_page_converted.mean())
```

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 [31]: # convert to numpy array
p_diffs = np.array(p_diffs)
```

```
In [32]: plt.hist(p_diffs)
plt.axvline(x=obs_diff, color='red')
plt.title('display p_diffs')
plt.xlabel('p_diffs')
plt.ylabel('Frequency')
```

```
Out[32]: Text(0, 0.5, 'Frequency')
```



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

```
In [33]: (p_diffs > obs_diff).mean()
```

```
Out[33]: 0.9008
```

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?

What I computed in part j. is called p-value in scientific studies. p-value is the probability of observing your statistic (or one more extreme in favor of the alternative) if the null hypothesis is true. In our case the p-value(0.9008) is greater than $\alpha=0.05$. We can say that we have to fail to reject null hypothesis. So our null hypothesis is True.

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 to the number of rows associated with the old page and new pages, respectively.

```
In [34]: import statsmodels.api as sm

convert_old = df2[df2.group == 'control'].converted.sum()
convert_new = df2[df2.group == 'treatment'].converted.sum()
n_old = df2[df2.group == 'control'].converted.count()
n_new = df2[df2.group == 'treatment'].converted.count()
convert_old, convert_new, n_old, n_new
```

```
Out[34]: (17489, 17264, 145274, 145309)
```

m. Now use `stats.proportions_ztest` to compute your test statistic and p-value. [Here \(http://knowledgegetack.com/python/statsmodels/proportions_ztest/\)](http://knowledgegetack.com/python/statsmodels/proportions_ztest/) is a helpful link on using the built in.

```
In [35]: import statsmodels.api as sm
z_score, p_value = sm.stats.proportions_ztest([convert_new, convert_old], [n_new, n_old])
print(z_score, p_value)

-1.3102408579271012 0.9049428161159749
```

n. What do the z-score and p-value you computed in the previous question mean for the conversion rates of the old and new pages? Do they agree with the findings in parts **j.** and **k.**?

The z-score means that the difference between our test statistic (the difference between conversion rates) and the null hypothesis is -1.31 standard deviations above the mean. p-value is 0.9049, which is bigger than our alpha of 0.05. This p-value is similar to the previous p-value of 0.9008, so the z-test appears to agree with the previous findings. With this computation, we can confidently say we fail to reject null hypothesis

Part III - A regression approach

1. In this final part, you will see that the result you achieved in the previous A/B test 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?

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 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]: df2.head()
```

```
Out[36]:
```

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

```
In [37]: # copy the dataframe to a new one, for clarity
df2_copy = df2.copy()
```

```
In [38]: # another sanity check, if True, go ahead
df2_copy.shape == df2.shape
```

```
Out[38]: True
```

```
In [39]: df2_copy['intercept'] = 1
df2_copy['ab_page'] = pd.get_dummies(df2_copy['group'])['treatment']
df2_copy.head(5)
```

```
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 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 [41]: logit_mod = sm.Logit(df2_copy['converted'], df2_copy[['intercept', 'ab_page']])
results = logit_mod.fit()
```

```
Optimization terminated successfully.
      Current function value: 0.366119
      Iterations 6
```

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

```
In [42]: results.summary()
```

Out[42]:

Logit Regression Results

Dep. Variable:	converted	No. Observations:	290583
Model:	Logit	Df Residuals:	290581
Method:	MLE	Df Model:	1
Date:	Tue, 14 Mar 2023	Pseudo R-squ.:	8.068e-06
Time:	16:19:12	Log-Likelihood:	-1.0639e+05
converged:	True	LL-Null:	-1.0639e+05
Covariance Type:	nonrobust	LLR p-value:	0.1901

	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	-1.310	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 the **Part II**?

When I compare `p_value(0.1901)` with `treashold=0.05`, `p_value` is greater than `treashold`. So we have to fail to reject null hypothesis. It's mean thaat we found statistical evidence to accept null hypothesis.

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?

Adding a relevant variable can prevent bias in the estimate of the other regression coefficient but can also increase variance of other regression coefficient.

Adding an irrelevant variable can increase the variance of the estimate of other correlation coefficient and will not have any benefits.

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 appropriate rows. [Here \(https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.join.html\)](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.join.html) are the docs for joining tables.

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

```
In [43]: countries_df = pd.read_csv('./countries.csv')
df_new = countries_df.set_index('user_id').join(df2_copy.set_index('user_id'), how='left')
```

```
In [44]: df_new.head()
```

```
Out[44]:
```

	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

```
In [45]: df_new.country.unique()
```

```
Out[45]: array(['UK', 'US', 'CA'], dtype=object)
```

```
In [46]: df_new[['ca', 'uk', 'us']] = pd.get_dummies(df_new['country'])
```

```
In [47]: df_new.head()
```

```
Out[47]:
```

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

```
In [53]: # fitting linear model
df_new['intercept'] = 1

logit_mod = sm.Logit(df_new['converted'], df_new[['intercept', 'ab_page', 'ca', 'us']])
results = logit_mod.fit()
```

```
Optimization terminated successfully.
Current function value: 0.366114
Iterations 6
```

```
In [54]: results.summary()
```

Out[54]: Logit Regression Results

Dep. Variable:	converted	No. Observations:	290583
Model:	Logit	Df Residuals:	290579
Method:	MLE	Df Model:	3
Date:	Tue, 14 Mar 2023	Pseudo R-squ.:	2.322e-05
Time:	16:46:44	Log-Likelihood:	-1.0639e+05
converged:	True	LL-Null:	-1.0639e+05
Covariance Type:	nonrobust	LLR p-value:	0.1761

	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	-1.306	0.191	-0.037	0.007
ca	-0.0506	0.028	-1.784	0.074	-0.106	0.005
us	-0.0099	0.013	-0.743	0.458	-0.036	0.016

Based on the p-values above, it also does not appear as though country has a significant impact on conversion.

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 [55]: df_new['ca_ab'] = df_new['ca']*df_new['ab_page']
df_new['us_ab'] = df_new['us']*df_new['ab_page']
logit_mod = sm.Logit(df_new['converted'], df_new[['intercept', 'ca_ab', 'us_ab', 'ca', 'us', 'ab_page']])
results = logit_mod.fit()
results.summary()
```

```
Optimization terminated successfully.
Current function value: 0.366110
Iterations 6
```

Out[55]: Logit Regression Results

Dep. Variable:	converted	No. Observations:	290583
Model:	Logit	Df Residuals:	290577
Method:	MLE	Df Model:	5
Date:	Tue, 14 Mar 2023	Pseudo R-squ.:	3.481e-05
Time:	16:50:48	Log-Likelihood:	-1.0639e+05
converged:	True	LL-Null:	-1.0639e+05
Covariance Type:	nonrobust	LLR p-value:	0.1922

	coef	std err	z	P> z	[0.025	0.975]
intercept	-1.9922	0.016	-123.457	0.000	-2.024	-1.961
ca_ab	-0.0783	0.057	-1.378	0.168	-0.190	0.033
us_ab	-0.0314	0.027	-1.180	0.238	-0.084	0.021
ca	-0.0118	0.040	-0.296	0.767	-0.090	0.066
us	0.0057	0.019	0.306	0.760	-0.031	0.043
ab_page	0.0108	0.023	0.475	0.635	-0.034	0.056

Conclusions

In this project, I used A/B test to check which web page run the experiment longer to make users decision. Firstly I covered my data and i dropped my duplicate value and counted some probabilities. According to the hypoythesis test in Part2, I found p_value=0,9008 which is bigger than alpha=0.05. So I failed to reject null hypothesis. After z-test I found p-value=0,9049 which is smilar I found it in previous part. I cheked p_value with different regressions regarding countries for ab_page. Ultimately, we do not have enough evidence to reject the null hypothesis based on any of our A/B testing. As a result, there is no reason to switch to the new page, when the old one performs just as well.

Gather Submission Materials

Once you are satisfied with the status of your Notebook, you should save it in a format that will make it easy for others to read. You can use the **File -> Download as -> HTML (.html)** menu to save your notebook as an .html file. If you are working locally and get an error about "No module name", then

open a terminal and try installing the missing module using `pip install <module_name>` (don't include the "<" or ">" or any words following a period in the module name).

You will submit both your original Notebook and an HTML or PDF copy of the Notebook for review. There is no need for you to include any data files with your submission. If you made reference to other websites, books, and other resources to help you in solving tasks in the project, make sure that you document them. It is recommended that you either add a "Resources" section in a Markdown cell at the end of the Notebook report, or you can include a `readme.txt` file documenting your sources.

Submit the Project

When you're ready, click on the "Submit Project" button to go to the project submission page. You can submit your files as a .zip archive or you can link to a GitHub repository containing your project files. If you go with GitHub, note that your submission will be a snapshot of the linked repository at time of submission. It is recommended that you keep each project in a separate repository to avoid any potential confusion: if a reviewer gets multiple folders representing multiple projects, there might be confusion regarding what project is to be evaluated.

It can take us up to a week to grade the project, but in most cases it is much faster. You will get an email once your submission has been reviewed. If you are having any problems submitting your project or wish to check on the status of your submission, please email us at dataanalyst-project@udacity.com (<mailto:dataanalyst-project@udacity.com>). In the meantime, you should feel free to continue on with your learning journey by beginning the next module in the program.

Sources: https://verascity.github.io/ab_test.html (https://verascity.github.io/ab_test.html)
<https://stats.stackexchange.com/questions/304707/costs-and-benefits-of-adding-more-variables-to-multiple-regression> (<https://stats.stackexchange.com/questions/304707/costs-and-benefits-of-adding-more-variables-to-multiple-regression>) <https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-what-are-the-differences-between-one-tailed-and-two-tailed-tests/>
(<https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-what-are-the-differences-between-one-tailed-and-two-tailed-tests/>)