Analyze_ab_test_results_notebook-Copy1

October 1, 2023

1 Analyze A/B Test Results

Introduction

A/B tests are very commonly performed by data analysts and data scientists. For this project, I will be working to understand the results of an A/B test run by an e-commerce website. My goal is to work through this notebook to help the company understand if they should: - Implement the new webpage, - Keep the old webpage, or - Perhaps run the experiment longer to make their decision.

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)
```

1.0.1 ToDo 1.1

Now, read in the `ab_data.csv` data. Store it in `df`. Below is the description of the data, there are a total of 5 columns:

Data columns	Purpose	Valid values
user_id	Unique ID	Int64 values
timestamp	Time stamp when the user visited the webpage	-

Data columns	Purpose	Valid values
group	In the current A/B experiment, the users are categorized into two broad groups. The `control` group users are expected to be served with `old_page`; and `treatment` group users are matched with the `new_page`. However, **some inaccurate rows** are present in the initial data, such as a `control` group user is matched with a `new_page`.	`['control', 'treatment']`
landing_page	It denotes whether the user visited the old or new webpage.	`['old_page', 'new_page']`
converted	It denotes whether the user decided to pay for the company's product. Here, `1` means yes, the user bought the product.	`[0, 1]`

Use your dataframe to answer the questions in Quiz 1 of the classroom.

Tip: Please save your work regularly.

a. Read in the dataset from the `ab_data.csv` file 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 cell below to find the number of rows in the dataset.

```
In [3]: df.shape[0]
```

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

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['converted'].sum()/290584
```

```
Out[5]: 0.12126269856564711
```

e. The number of times when the "group" is `treatment` but "landing_page" is not a `new_page`.

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

        treat_old + control_new
```

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

f. Do any of the rows have missing values?

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294478 entries, 0 to 294477
Data columns (total 5 columns):
user_id         294478 non-null int64
timestamp       294478 non-null object
group           294478 non-null object
landing_page    294478 non-null object
converted       294478 non-null int64
dtypes: int64(2), object(3)
memory usage: 11.2+ MB
```

1.0.2 ToDo 1.2

In a particular row, the **group** and **landing_page** columns should have either of the following acceptable values:

user_id	timestamp	group	landing_page	converted
XXXX	XXXX	control	old_page	X
XXXX	XXXX	treatment	new_page	X

It means, the control group users should match with old_page; and treatment group users should matched with the new_page.

However, for the rows where treatment does not match with new_page or control does not match with old_page, we cannot be sure if such rows truly received the new or old wepage.

Use **Quiz 2** in the classroom to figure out how should we handle the rows where the group and landing_page columns don't match?

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]: # Remove the inaccurate rows, and store the result in a new dataframe df2
        df2 = df.query("group == 'control' and landing_page == 'old_page'")
        df2 = df2.append(df.query("group == 'treatment' and landing_page == 'new_page'"))
```

```
In [9]: # Double Check all of the incorrect rows were removed from df2 -
        # Output of the statement below should be 0
        df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) == False].sha
```

```
Out[9]: 0
```

1.0.3 ToDo 1.3

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

a. How many unique **user_id**s are in **df2**?

```
In [10]: df2.user_id.nunique()
```

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

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

```
In [11]: df2[df2['user_id'].duplicated()]
```

```
Out[11]:       user_id                   timestamp      group landing_page  converted
         2893   773192  2017-01-14 02:55:59.590927  treatment     new_page          0
```

c. Display the rows for the duplicate **user_id**?

```
In [12]: df2[df2['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**, from the **df2** dataframe.

```
In [13]: df2 = df2.drop(1899)
```

1.0.4 ToDo 1.4

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?

> **Tip**: The probability you'll compute represents the overall "converted" success rate in the population and you may call it $p_{population}$.

```
In [14]: df2.converted.mean()
```

```
Out[14]: 0.11959708724499628
```

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

```
In [15]: c_prob = df2.query("group == 'control'")['converted'].mean()
         c_prob
```

```
Out[15]: 0.1203863045004612
```

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

```
In [16]: t_prob = df2.query("group == 'treatment'")['converted'].mean()
         t_prob
```

```
Out[16]: 0.11880806551510564
```

> **Tip**: The probabilities you've computed in the points (b). and (c). above can also be treated as conversion rate. Calculate the actual difference (`obs_diff`) between the conversion rates for the two groups. You will need that later.

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

```
In [17]: df2.query('landing_page == "new_page"').shape[0]/df2.shape[0]
```

```
Out[17]: 0.5000619442226688
```

e. Consider your results from parts (a) through (d) above, and explain below whether the new `treatment` group users lead to more conversions.

It appears that the new page did not result in higher conversions. Instead, it led to a slightly lower conversion rate compared to the old page, though the discrepancy seems to be insignificant.

Part II - A/B Test

Since a timestamp is associated with each event, you could run a hypothesis test continuously as long as you observe the events.

However, then the hard questions would be: - 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.

1.0.5 ToDo 2.1

For now, consider you need to make the decision just based on all the data provided.

> Recall that you just calculated that the "converted" probability (or rate) for the old page is *slightly* higher than that of the new page (ToDo 1.4.c).

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 be your null and alternative hypotheses (H_0 and H_1)?

You can state your hypothesis in terms of words or in terms of p_{old} and p_{new}, which are the "converted" probability (or rate) for the old and new pages respectively.

To make a decision based on the data provided, we can set up the following null and alternative hypotheses:

Null Hypothesis (H0): The old page is as good as or better than the new page in terms of conversion rate. Mathematically, this can be represented as: pold >= pnew.

Alternative Hypothesis (HA): The new page is definitely better than the old page in terms of conversion rate. Mathematically, this can be represented as: pold < pnew.

With these hypotheses, we will test whether there is enough evidence to reject the null hypothesis in favor of the alternative hypothesis. If the p-value (probability value) associated with the test is less than or equal to 0.05 (5% Type I error rate), we will reject the null hypothesis and conclude that the new page is indeed better in terms of conversion rate. If the p-value is greater than 0.05, we will fail to reject the null hypothesis, indicating that there is not enough evidence to support the claim that the new page is better, and we will stick with the old page.

1.0.6 ToDo 2.2 - Null Hypothesis H_0 Testing

Under the null hypothesis H_0, assume that p_{new} and p_{old} are equal. Furthermore, assume that p_{new} and p_{old} both are equal to the **converted** success rate in the `df2` data regardless of the page. So, our assumption is:

$p_{new} = p_{old} = p_{population}$

In this section, you will:

- Simulate (bootstrap) sample data set for both groups, and compute the "converted" probability p for those samples.
- Use a sample size for each group equal to the ones in the `df2` data.
- Compute the difference in the "converted" probability for the two samples above.
- Perform the sampling distribution for the "difference in the converted probability" between the two simulated-samples over 10,000 iterations; and calculate an estimate.

Use the cells below to provide the necessary parts of this simulation. 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 hypothesis?

In [18]:
```
p_null = df2['converted'].mean()
p_null
```

Out[18]:
```
0.11959708724499628
```

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

```
In [19]: p_null
```

```
Out[19]: 0.11959708724499628
```

c. What is n_{new}, the number of individuals in the treatment group? *Hint*: The treatment group users are shown the new page.

```
In [20]: n_new = df2.query("landing_page == 'new_page'").shape[0]
         n_new
```

```
Out[20]: 145310
```

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

```
In [21]: n_old = df2.query("landing_page == 'old_page'").shape[0]
         n_old
```

```
Out[21]: 145274
```

e. Simulate Sample for the `treatment` Group Simulate n_{new} transactions with a conversion rate of p_{new} under the null hypothesis. *Hint*: Use `numpy.random.choice()` method to randomly generate n_{new} number of values. Store these n_{new} 1's and 0's in the `new_page_converted` numpy array.

```
In [22]: new_page_converted = np.random.binomial(1, p_null, n_new)
```

f. Simulate Sample for the `control` Group Simulate n_{old} transactions with a conversion rate of p_{old} under the null hypothesis. Store these n_{old} 1's and 0's in the `old_page_converted` numpy array.

```
In [23]: old_page_converted = np.random.binomial(1, p_null, n_old)
```

g. Find the difference in the "converted" probability $(p'_{new}$ - $p'_{old})$ for your simulated samples from the parts (e) and (f) above.

```
In [24]: new_page_converted.mean() - old_page_converted.mean()
```

```
Out[24]: -0.00083501403497864002
```

h. Sampling distribution Re-create `new_page_converted` and `old_page_converted` and find the $(p'_{new}$ - $p'_{old})$ value 10,000 times using the same simulation process you used in parts (a) through (g) above.

Store all $(p'_{new}$ - $p'_{old})$ values in a NumPy array called `p_diffs`.

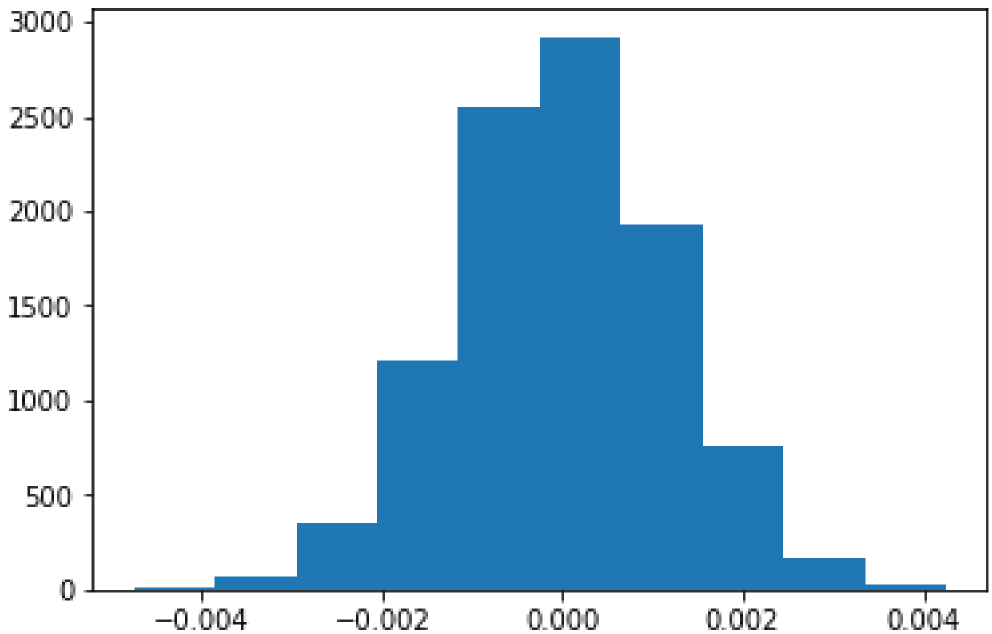
```
In [25]: p_diffs = []
         new_converted_simulation = np.random.binomial(n_new, p_null, 10000)/n_new
         old_converted_simulation = np.random.binomial(n_old, p_null, 10000)/n_old
         p_diffs = new_converted_simulation - old_converted_simulation
```

i. Histogram 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.

Also, use `plt.axvline()` method to mark the actual difference observed in the `df2` data (recall `obs_diff`), in the chart.

Tip: Display title, x-label, and y-label in the chart.

```
In [26]: plt.hist(p_diffs);
```



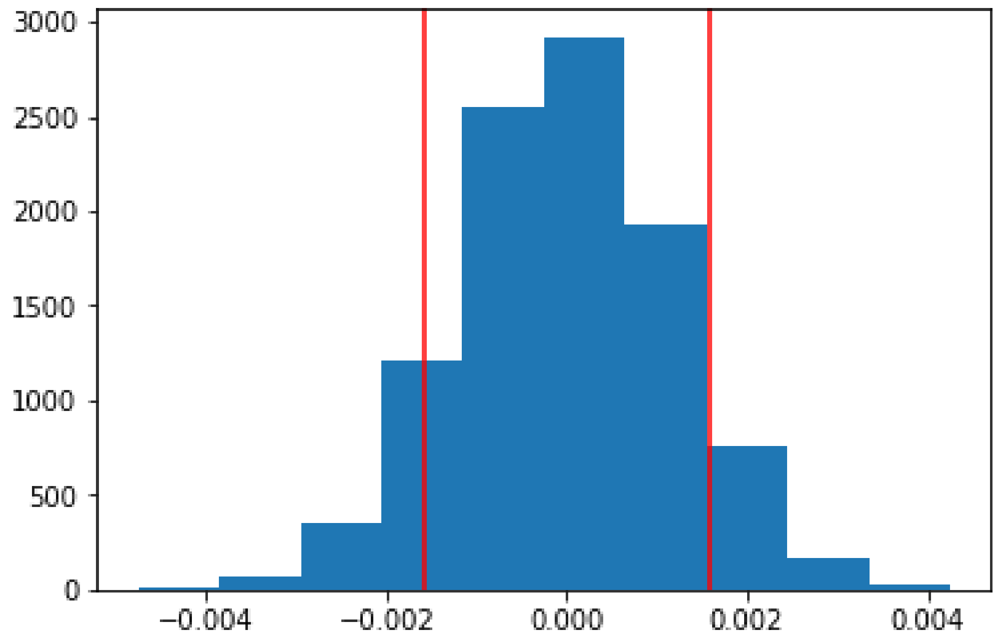

```
In [27]: obs_diff = t_prob - c_prob

         low_prob = (p_diffs < obs_diff).mean()
         high_prob = (p_diffs.mean() + (p_diffs.mean() - obs_diff) < p_diffs).mean()

         plt.hist(p_diffs);
         plt.axvline(obs_diff, color='red');
         plt.axvline(p_diffs.mean() + (p_diffs.mean() - obs_diff), color='red');

         p_val = low_prob + high_prob
         print(p_val)
```

```
0.1862
```



j. What proportion of the **p_diffs** are greater than the actual difference observed in the `df2` data?

```
In [28]: import numpy as np

         # Calculate the actual difference observed (treatment - control) in the original df2 Da
         actual_diff_observed = df2.query("group == 'treatment'")['converted'].mean() - df2.quer

         # Calculate the proportion of p_diffs that are greater than the actual difference obser
         proportion_greater = (p_diffs > actual_diff_observed).mean()

         # Display the result with at least 4 digits after the decimal point precision
         print("Proportion of p_diffs greater than the actual difference observed: {:.4f}".forma
```

```
Proportion of p_diffs greater than the actual difference observed: 0.9081
```

k. Please explain in words what you have just computed in part **j** above.
- What is this value called in scientific studies?
- What does this value signify in terms of whether or not there is a difference between the new and old pages? *Hint*: Compare the value above with the "Type I error rate (0.05)".

The value calculated above is called the p-value in scientific studies. It indicates whether there is a significant difference between the new and old pages. If the p-value is less than the Type I error rate (commonly set at 0.05), we reject the null hypothesis and conclude that the new page is better. Otherwise, we fail to reject the null hypothesis, suggesting no significant difference between the pages.

l. Using Built-in Methods for Hypothesis Testing 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 walk-through of the ideas that are critical to correctly thinking about statistical significance.

Fill in the statements below to calculate the: - `convert_old`: number of conversions with the old_page - `convert_new`: number of conversions with the new_page - `n_old`: number of individuals who were shown the old_page - `n_new`: number of individuals who were shown the new_page

```
In [29]: convert_old = df2.query("landing_page == 'old_page'")['converted'].sum()
         convert_new = df2.query("landing_page == 'new_page'")['converted'].sum()
```

m. Now use `sm.stats.proportions_ztest()` to compute your test statistic and p-value. Here is a helpful link on using the built in.

The syntax is:

```
proportions_ztest(count_array, nobs_array, alternative='larger')
```

where, - `count_array` = represents the number of "converted" for each group - `nobs_array` = represents the total number of observations (rows) in each group - `alternative` = choose one of the values from `[two-sided, smaller, larger]` depending upon two-tailed, left-tailed, or right-tailed respectively. >**Hint**: It's a two-tailed if you defined H_1 as $(p_{new} = p_{old})$. It's a left-tailed if you defined H_1 as $(p_{new} < p_{old})$. It's a right-tailed if you defined H_1 as $(p_{new} > p_{old})$.

The built-in function above will return the z_score, p_value.

> **Tip**: You don't have to dive deeper into z-test for this exercise. **Try having an overview of what does z-score signify in general.**

```
In [30]: import statsmodels.api as sm
         convert_old = df2.query('group == "control"')['converted'].sum()
         convert_new = df2.query('group == "treatment"')['converted'].sum()
         n_old = df2.query('landing_page == "old_page"').shape[0]
         n_new = df2.query('landing_page == "new_page"').shape[0]
```

```
/opt/conda/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The panda
  from pandas.core import datetools
```

```
In [33]: import statsmodels.api as sm

         # Calculate the z-score and p-value
         z_score, p_value = sm.stats.proportions_ztest([convert_new, convert_old], [n_new, n_old

         # Print the z-score and p-value
         print("Z-score:", z_score)
         print("P-value:", p_value)
```

```
Z-score: -1.31092419842
P-value: 0.905058312759
```

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.**?

> **Tip**: Notice whether the p-value is similar to the one computed earlier. Accordingly, can you reject/fail to reject the null hypothesis? It is important to correctly interpret the test statistic and p-value.

With a z-score close to 0 and a high p-value, we do not have enough evidence to reject the null hypothesis. The conversion rates of the old and new pages are likely to be similar, and the new page does not appear to result in a significantly higher conversion rate compared to the old page.

The z-score and p-value align with the observations made in parts j and k. They provide further evidence that there is no significant difference in the conversion rates between the two landing pages. As a result, we cannot reject the null hypothesis, which suggests that the new page does not lead to a significantly higher conversion rate compared to the old page. These findings reinforce the notion that there is no strong statistical basis to favor one landing page over the other in terms of conversion rates.

Part III - A regression approach

1.0.7 ToDo 3.1

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 in the `df2` data is either a conversion or no conversion, what type of regression should you be performing in this case?

Since each row in the df2 data is either a conversion or no conversion (binary outcome), the appropriate type of regression to be performed in this case is Logistic Regression.

b. The goal is to use **statsmodels** library to fit the regression model you specified in part **a.** above to see if there is a significant difference in conversion based on the page-type a customer receives. However, you first need to create the following two columns in the `df2` dataframe: 1. `intercept` - It should be `1` in the entire column. 2. `ab_page` - It's a dummy variable column, having a value `1` when an individual receives the **treatment**, otherwise `0`.

```
In [35]: df2['intercept'] = 1
         df2[['a_page', 'ab_page']] = pd.get_dummies(df2['group'])
         df2 = df2.drop('a_page', axis=1)
         df2.head()
```

```
Out[35]:    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
         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
         7   719014  2017-01-17 01:48:29.539573  control     old_page          0

            intercept  ab_page
         0          1        0
         1          1        0
         4          1        0
         5          1        0
         7          1        0
```

c. Use **statsmodels** to instantiate your regression model on the two columns you created in part (b). above, then fit the model to predict whether or not an individual converts.

```
In [36]: df2['intercept'] = 1
         logit_mod = sm.Logit(df2['converted'], df2[['intercept', 'ab_page']])
         results = logit_mod.fit()
```

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

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

```
In [37]: results.summary2()
```

```
Out[37]: <class 'statsmodels.iolib.summary2.Summary'>
         """
                                  Results: Logit
         ==================================================================
         Model:              Logit            No. Iterations:   6.0000
         Dependent Variable: converted        Pseudo R-squared: 0.000
         Date:               2023-07-29 08:38 AIC:              212780.3502
         No. Observations:   290584           BIC:              212801.5095
         Df Model:           1                Log-Likelihood:   -1.0639e+05
         Df Residuals:       290582           LL-Null:          -1.0639e+05
         Converged:          1.0000           Scale:            1.0000
         -------------------------------------------------------------------
                       Coef.   Std.Err.      z      P>|z|   [0.025   0.975]
         -------------------------------------------------------------------
         intercept    -1.9888    0.0081  -246.6690  0.0000  -2.0046  -1.9730
         ab_page      -0.0150    0.0114    -1.3109  0.1899  -0.0374   0.0074
         ==================================================================

         """
```

e. What is the p-value associated with **ab_page**? Why does it differ from the value you found in **Part II**?

Hints: - 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**? - You may comment on if these hypothesis (Part II vs. Part III) are one-sided or two-sided. - You may also compare the current p-value with the Type I error rate (0.05).

The p-value associated with the 'ab_page' variable is 0.190.

The difference in the p-value between Part II and Part III is due to the difference in the null and alternative hypotheses and the type of test performed.

In Part II, we conducted a one-tailed hypothesis test for the conversion rate between the new page and old page. The null hypothesis (H0) was that the conversion rate of the new page is less than or equal to the conversion rate of the old page, while the alternative hypothesis (H1) was that the conversion rate of the new page is greater than the conversion rate of the old page. This led to a one-tailed test, and the p-value was calculated for this specific direction of the test.

In Part III, we performed a two-tailed logistic regression model. The null hypothesis (H0) in this case is that there is no relationship between the type of page and user conversion, while the alternative hypothesis (H1) is that there is a relationship. This led to a two-tailed test, and the p-value calculated represents the probability of observing the data under the null hypothesis in both tails.

The p-value of 0.190 is greater than the Type I error rate (0.05), suggesting that we do not have sufficient evidence to reject the null hypothesis. In other words, we do not have strong evidence that the type of page (new or old) significantly affects user conversion.

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?

Currently, it seems that the treatment or control page does not have a significant impact on user conversions. As a result, it might be beneficial to explore other potential factors that could predict conversion. However, when selecting these factors, it is essential to ensure that they are not collinear with each other. Collinearity can lead to unstable model estimates and can make it challenging to interpret the individual effects of each predictor accurately. Thus, careful consideration of the chosen factors is vital to avoid multicollinearity issues in the analysis.

g. Adding countries Now along with testing if the conversion rate changes for different pages, also add an effect based on which country a user lives in.

1. You will need to read in the **countries.csv** dataset and merge together your `df2` datasets on the appropriate rows. You call the resulting dataframe `df_merged`. Here are the docs for joining tables.

2. Does it appear that country had an impact on conversion? To answer this question, consider the three unique values, `['UK', 'US', 'CA']`, in the `country` column. Create dummy variables for these country columns. >**Hint:** Use `pandas.get_dummies()` to create dummy variables. **You will utilize two columns for the three dummy variables.**

Provide the statistical output as well as a written response to answer this question.

```
In [38]: countries = pd.read_csv('countries.csv')
         countries.head()

         df2 = df2.set_index('user_id').join(countries.set_index('user_id'))
```

```
In [39]: df2[['CA', 'UK', 'US']] = pd.get_dummies(df2['country'])
```

```
In [40]: logit_mod = sm.Logit(df2['converted'], df2[['intercept', 'ab_page', 'CA', 'UK']])
         results = logit_mod.fit()
         results.summary2()
```

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

```
Out[40]: <class 'statsmodels.iolib.summary2.Summary'>
"""
                          Results: Logit
==================================================================
Model:              Logit            No. Iterations:   6.0000
Dependent Variable: converted        Pseudo R-squared: 0.000
Date:               2023-07-29 08:39 AIC:              212781.1253
No. Observations:   290584           BIC:              212823.4439
Df Model:           3                Log-Likelihood:   -1.0639e+05
Df Residuals:       290580           LL-Null:          -1.0639e+05
Converged:          1.0000           Scale:            1.0000
-------------------------------------------------------------------
              Coef.   Std.Err.      z      P>|z|    [0.025   0.975]
-------------------------------------------------------------------
intercept    -1.9893    0.0089  -223.7628  0.0000  -2.0067  -1.9718
ab_page      -0.0149    0.0114    -1.3069  0.1912  -0.0374   0.0075
CA           -0.0408    0.0269    -1.5161  0.1295  -0.0934   0.0119
UK            0.0099    0.0133     0.7433  0.4573  -0.0162   0.0359
==================================================================

"""
```

The p-values obtained from the logistic regression model for the different countries (e.g., UK and CA) indicate that country does not have a significant impact on conversion. In other words, there is no strong evidence to suggest that users from different countries have significantly different conversion rates.

h. Fit your model and obtain the results 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 are there significant effects on conversion. **Create the necessary additional columns, and fit the new model.**

Provide the summary results (statistical output), and your conclusions (written response) based on the results.

> **Tip**: Conclusions should include both statistical reasoning, and practical reasoning for the situation.
>
> **Hints**: - Look at all of p-values in the summary, and compare against the Type I error rate (0.05). - Can you reject/fail to reject the null hypotheses (regression model)? - Comment on the effect of page and country to predict the conversion.

```
In [41]: df2['CA_page'] = df2['CA']*df2['ab_page']
         df2['UK_page'] = df2['UK']*df2['ab_page']
         df2['US_page'] = df2['US']*df2['ab_page']
         logit_mod = sm.Logit(df2['converted'], df2[['intercept', 'CA_page', 'UK_page']])
         results = logit_mod.fit()
         results.summary2()
```

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

```
Iterations 6
```

Out[41]:
```
<class 'statsmodels.iolib.summary2.Summary'>
"""
                          Results: Logit
=================================================================
Model:              Logit            No. Iterations:   6.0000
Dependent Variable: converted        Pseudo R-squared: 0.000
Date:               2023-07-29 08:39 AIC:              212779.0384
No. Observations:   290584           BIC:              212810.7773
Df Model:           2                Log-Likelihood:   -1.0639e+05
Df Residuals:       290581           LL-Null:          -1.0639e+05
Converged:          1.0000           Scale:            1.0000
------------------------------------------------------------------
              Coef.   Std.Err.      z      P>|z|    [0.025   0.975]
------------------------------------------------------------------
intercept    -1.9963    0.0062  -322.0487  0.0000  -2.0084  -1.9841
CA_page      -0.0752    0.0376    -1.9974  0.0458  -0.1489  -0.0014
UK_page       0.0149    0.0173     0.8617  0.3888  -0.0190   0.0488
=================================================================

"""
```

From the results above, there is one p-value that stands out as statistically significant: the interaction between CA and ab_page (p = 0.046; p < 0.05). This indicates that the combined effect of being in Canada and receiving the new page has a statistically significant impact on user conversions. It suggests that users from Canada who are exposed to the new page might have different conversion rates compared to users from other countries or users who receive the old page.

It is important to pay attention to this significant interaction, as it provides valuable insight into how different factors together can influence the outcome variable. However, we should interpret this result carefully and consider additional factors or further investigation to gain a more comprehensive understanding of the impact of the new page on user conversions in different countries.

In [42]:
```
np.exp(results.params)
print(1/0.927579)
```

```
1.0780752906221465
```

The interpretation of the coefficient for 'CA_page' indicates that, holding all other variables constant, a user from Canada who receives the new page would be approximately 1.08 times more likely to convert compared to a user from Canada who receives the old page. While this difference shows a small degree of statistical significance, it lacks practical significance. Moreover, considering that 'CA_page' is the only statistically significant variable among ab_page and country variables (the rest being insignificant on their own), it is unlikely to be practically meaningful.

The results of the A/B testing do not provide enough evidence to reject the null hypothesis. Therefore, there is no compelling reason to switch to the new page, as the old page performs equally well in terms of conversion.

In conclusion, based on the analysis, there is no clear evidence that the new page outperforms the old one in terms of conversion rates. Thus, it may not be worthwhile to implement the new page, and sticking with the old page might be a better option.