Analyze_ab_test_Szymon_Debski

March 23, 2021

0.1 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!

0.2 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.

[55]:

```
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:

```
[56]: df = pd.read_csv('ab_data.csv')
```

```
[57]: df.head(5)
```

```
[57]:    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.

```
[58]: df.info()
```

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

   c. The number of unique users in the dataset.

```
[59]: len(df.user_id.unique())
```

```
[59]: 290584
```

   d. The proportion of users converted.

```
[60]: round(df.converted.mean() * 100, 2)
```

```
[60]: 11.97
```

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

```
[61]: (df.query('group == "treatment" and landing_page != "new_page"')['user_id'].
       ↪count()
      + df.query('group != "treatment" and landing_page == "new_page"')['user_id'].
       ↪count())
```

```
[61]: 3893
```

f. Do any of the rows have missing values?

```
[62]: df.isnull().sum()
```

```
[62]: 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**.

```
[63]: df2 = df.drop(df[(df['group'] == "treatment") & (df['landing_page'] !=␣
       ↪"new_page")].index)
      df2 = df2.drop(df2[(df2['group'] != "treatment") & (df2['landing_page'] ==␣
       ↪"new_page")].index)
```

```
[64]: # 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]
```

```
[64]: 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**?

```
[65]: df2.shape[0]
```

```
[65]: 290585
```

```
[66]: len(df2.user_id.unique())
```

```
[66]: 290584
```

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

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

```
[67]:       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
```

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

Information is the same except the timestamp therefore I will remove the second row.

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

```
[68]: df2 = df2.drop_duplicates(subset='user_id')
```

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?

```
[69]: round(df2.converted.mean() * 100, 2)
```

```
[69]: 11.96
```

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

```
[70]: round(df2.query('group == "control"')['converted'].mean() * 100, 2)
```

```
[70]: 12.04
```

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

```
[71]: round(df2.query('group == "treatment"')['converted'].mean() * 100, 2)
```

```
[71]: 11.88
```

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

```
[72]: round(df2.query('landing_page == "new_page"').count()[0]/df2.shape[0] * 100, 2)
```

```
[72]: 50.01
```

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.

0.3 Answere:

*** The conversion rate is slightly higher for the control group - 12.04 % vs 11.88% treatment group**

*** However based on this information we cannot conclude with any certainty which 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.

* **Null hypothesis - if the p value >= 5% the old page has better conversion rate**

* **Alternative hypothesis - if the p value < 5% the new page has a better conversion rate** 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.

[73]:
```
df2.head(1)
```

[73]:
```
   user_id                   timestamp    group landing_page  converted
0   851104  2017-01-21 22:11:48.556739  control     old_page          0
```

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

[74]:
```
p_new = df2.converted.mean()
p_new
```

[74]:
```
0.11959708724499628
```

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

[75]:
```
p_old = df2.converted.mean()
p_old
```

[75]:
```
0.11959708724499628
```

c. What is n_{new}?

[76]:
```
n_new = df2.query('landing_page == "new_page"').count()[0]
n_new
```

[76]:
```
145310
```

d. What is n_{old}?

[77]:
```
n_old = df2.query('landing_page == "old_page"').count()[0]
n_old
```

```
[77]: 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**.

```
[78]: new_page_converted = np.random.binomial(1, p_new, n_new)
      new_page_converted.mean()
```

```
[78]: 0.11807859059940816
```

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**.

```
[79]: old_page_converted = np.random.binomial(1, p_old, n_old)
      old_page_converted.mean()
```

```
[79]: 0.1197599019783306
```

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

```
[80]: #new_page and old page have diffrent sizes thats why we use means
      new_page_converted.mean() - old_page_converted.mean()
```

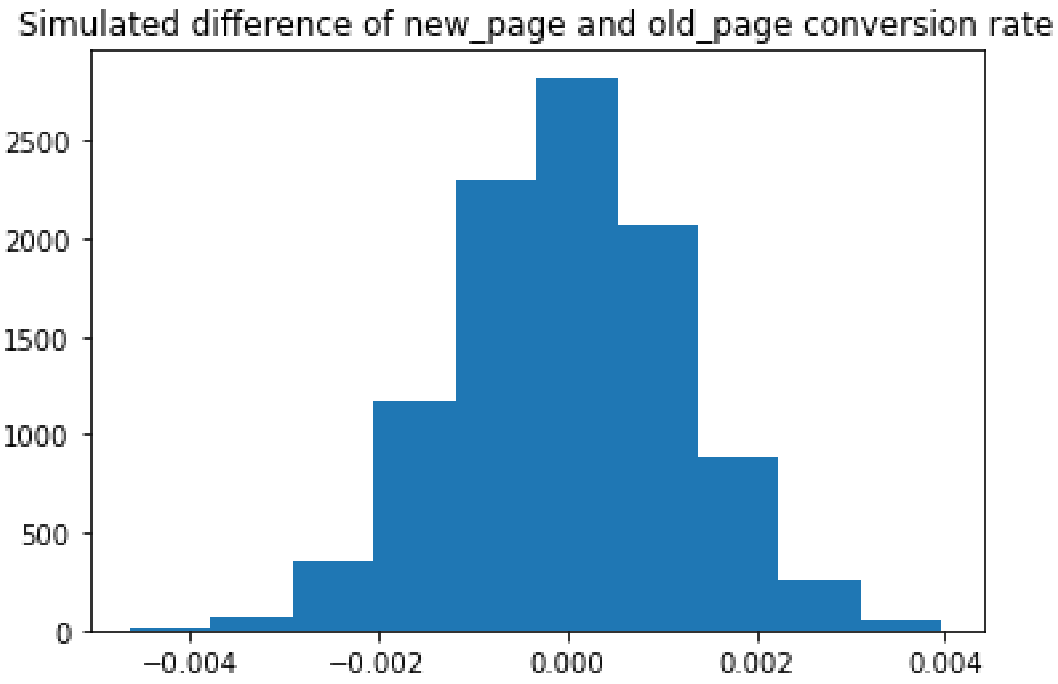
```
[80]: -0.0016813113789224399
```

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**.

```
[81]:  new_page_conv = np.random.binomial(n_new, p_new, 10000)/n_new
       old_page_conv = np.random.binomial(n_old, p_old, 10000)/n_old
       p_diffs = new_page_conv - old_page_conv
```

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.

```
[82]: plt.hist(p_diffs);
      plt.title('Simulated difference of new_page and old_page conversion rate');
```



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

```
[83]: act_new_conv = df2.query('landing_page == "new_page"')['converted'].mean()
      act_old_conv = df2.query('landing_page == "old_page"')['converted'].mean()

      act_diff = act_new_conv - act_old_conv
      act_diff
```

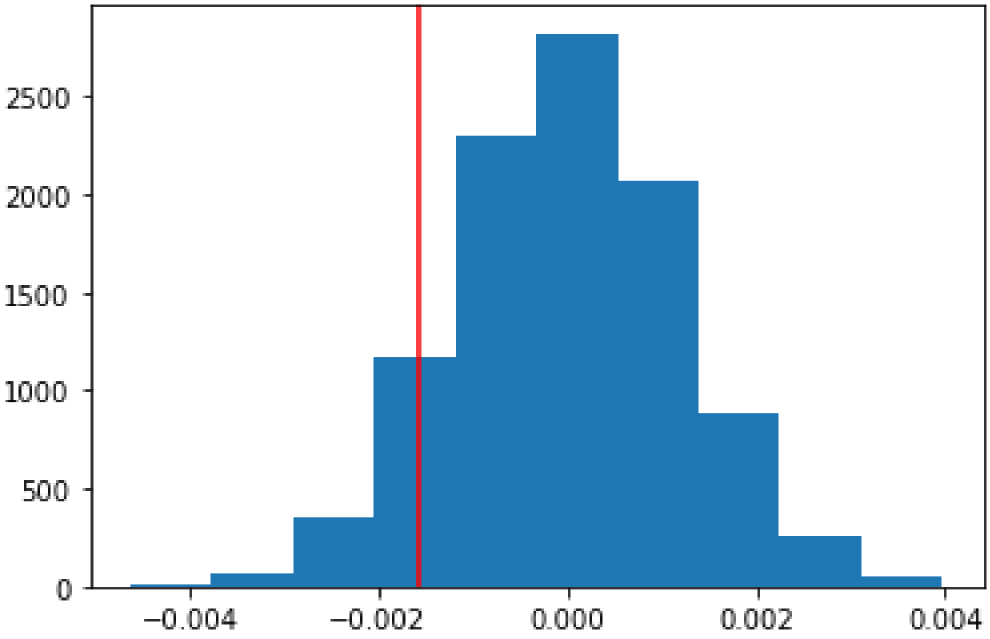
```
[83]: -0.0015782389853555567
```

```
[84]: p_diffs = np.array(p_diffs)
```

```
[85]: round((p_diffs > act_diff).mean() * 100, 2)
```

```
[85]: 90.62
```

```
[86]: plt.hist(p_diffs);
      plt.axvline(act_diff, c='red');
```



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?

In part J we computed the P-Value which in our example is 90%. The greater the value is the more random our results are and therefore NOT significant.

Based on this result we fail to reject the null hypothesis. This means the company should stay with the old page as there is no statistical evidence that the new page has a better conversion rate.

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.

[87]:

```
import statsmodels.api as sm

convert_old = df2.query('landing_page == "old_page"')['converted'].sum()
convert_new = df2.query('landing_page == "new_page"')['converted'].sum()
n_old = df2.query('landing_page == "old_page"').count()[0]
n_new = df2.query('landing_page == "new_page"').count()[0]

convert_old, n_old, convert_new, n_new
```

```
[87]: (17489, 145274, 17264, 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.

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

```
[88]: (-1.3109241984234394, 0.9050583127590245)
```

```
[89]: from scipy.stats import norm
      print(norm.ppf(1-(0.05))) #critical value for onesided test at 95%
```

```
1.6448536269514722
```

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 p-value and the z_score further prove our previous conclusion that we cannot reject the null hypothesis.

The p-value corresponds to the p-value we computed in part J - which confirms that our results are not significant.

The z-score is less than the critical value (1.64) which also proves that we cannot reject our null hypothesis.

### Part III - A regression approach

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

a. Since each row is either a conversion or no conversion, what type of regression should you be performing in this case?

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**.

```
[90]: df2.head(1)
```

```
[90]:    user_id                   timestamp    group landing_page  converted
      0   851104  2017-01-21 22:11:48.556739  control     old_page          0
```

```
[91]: df2['intercept'] = 1
```

```
df2['ab_page'] = pd.get_dummies(df2['group'])['treatment']

df2.head(1)
```

```
[91]:    user_id                   timestamp    group landing_page  converted  \
      0   851104  2017-01-21 22:11:48.556739  control     old_page          0

         intercept  ab_page
      0          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.

```
[92]: import statsmodels.api as sm
      log_mod = 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.

```
[93]: results = log_mod.fit()
      results.summary()
```

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

```
[93]: <class 'statsmodels.iolib.summary.Summary'>
      """
                                 Logit Regression Results
      ==============================================================================
      Dep. Variable:              converted   No. Observations:               290584
      Model:                          Logit   Df Residuals:                   290582
      Method:                           MLE   Df Model:                            1
      Date:                Tue, 23 Mar 2021   Pseudo R-squ.:               8.077e-06
      Time:                        12:08:42   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 the **Part II**?

The p-value for the above model is 0.19 which greater than 5% which indicates that the alternative hypothesis is not significant.

When comparing this with the model in Part II the conclusion is the same - that the conversion rate for the old_page is the same or better.

One thing to note that the p-value is different than in part II.

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?

It is a good idea to consider other factors in the model. However, there is always the risk the model will become too complex and it will be hard to come up with any clear conclusions.

g. Now along with testing if the conversion rate changes for different pages, also add an effect based on which country a user lives. You will need to read in the **countries.csv** dataset and merge together your datasets on the approporiate rows. Here 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.

```
[94]: 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(1)
```

```
[94]:         country                   timestamp    group landing_page  converted  \
      user_id
      834778       UK  2017-01-14 23:08:43.304998  control     old_page          0

               intercept  ab_page
      user_id
      834778           1        0
```

```
[95]: df_new.groupby(['country'])['converted'].mean()
```

```
[95]: country
      CA    0.115318
      UK    0.120594
      US    0.119547
      Name: converted, dtype: float64
```

```
[96]: df_new.country.unique()
```

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

[97]:
```
df_new[['UK','US', 'CA']]= pd.get_dummies(df_new['country'])
df_new.head(5)
```

[97]:
```
        country                   timestamp      group landing_page  \
user_id
834778       UK  2017-01-14 23:08:43.304998    control     old_page
928468       US  2017-01-23 14:44:16.387854  treatment     new_page
822059       UK  2017-01-16 14:04:14.719771  treatment     new_page
711597       UK  2017-01-22 03:14:24.763511    control     old_page
710616       UK  2017-01-16 13:14:44.000513  treatment     new_page

         converted  intercept  ab_page  UK  US  CA
user_id
834778           0          1        0   0   1   0
928468           0          1        1   0   0   1
822059           1          1        1   0   1   0
711597           0          1        0   0   1   0
710616           0          1        1   0   1   0
```

[98]:
```
df_new['intercept'] = 1

mlr = sm.Logit(df_new['converted'], df_new[['intercept', 'ab_page', 'UK',␣
 ↪'CA']])
results = mlr.fit()
results.summary()
```

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

[98]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:              converted   No. Observations:               290584
Model:                          Logit   Df Residuals:                   290580
Method:                           MLE   Df Model:                            3
Date:                Tue, 23 Mar 2021   Pseudo R-squ.:               2.323e-05
Time:                        12:08:49   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
UK            -0.0506      0.028     -1.784      0.074      -0.106       0.005
CA            -0.0099      0.013     -0.743      0.457      -0.036       0.016
```

```
==============================================================================
"""
```

Looking at the results we can conclude that the country does not impact the conversion rate as the p-value for both cases is > 5%.

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.

```
[99]: df_new['UK_ab_page'] = df_new['UK'] * df_new['ab_page']
      df_new.head(1)
```

```
[99]:         country                   timestamp    group landing_page  converted  \
      user_id
      834778       UK  2017-01-14 23:08:43.304998  control     old_page          0

               intercept  ab_page  UK  US  CA  UK_ab_page
      user_id
      834778           1        0   0   1   0           0
```

```
[100]: df_new['CA_ab_page'] = df_new['CA'] * df_new['ab_page']
       df_new.head(1)
```

```
[100]:         country                   timestamp    group landing_page  converted  \
       user_id
       834778       UK  2017-01-14 23:08:43.304998  control     old_page          0

                intercept  ab_page  UK  US  CA  UK_ab_page  CA_ab_page
       user_id
       834778           1        0   0   1   0           0           0
```

```
[101]: df_new['intercept'] = 1

       mlr = sm.Logit(df_new['converted'], df_new[['intercept', 'ab_page', 'UK', 'CA',␣
        ↪'UK_ab_page', 'CA_ab_page']])
       results = mlr.fit()
       results.summary()
```

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

```
[101]: <class 'statsmodels.iolib.summary.Summary'>
       """
                                  Logit Regression Results
       ==============================================================================
Dep. Variable:              converted   No. Observations:               290584
Model:                          Logit   Df Residuals:                   290578
Method:                           MLE   Df Model:                            5
Date:                Tue, 23 Mar 2021   Pseudo R-squ.:               3.482e-05
Time:                        12:08:52   Log-Likelihood:            -1.0639e+05
converged:                       True   LL-Null:                   -1.0639e+05
Covariance Type:            nonrobust   LLR p-value:                    0.1920
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -1.9922      0.016   -123.457      0.000      -2.024      -1.961
ab_page        0.0108      0.023      0.475      0.635      -0.034       0.056
UK            -0.0118      0.040     -0.296      0.767      -0.090       0.066
CA             0.0057      0.019      0.306      0.760      -0.031       0.043
UK_ab_page    -0.0783      0.057     -1.378      0.168      -0.190       0.033
CA_ab_page    -0.0314      0.027     -1.181      0.238      -0.084       0.021
==============================================================================
"""
```

## Conclusions

Based on the results above and the interactions between country and the ab_page we can conclude that none of the variables have significant p-values. Haveing this in mind we would fail to reject the null hypothesis.

Taking into account all the results from all the tests my conclusion is that we should stick with the old page as there is no evidence that the new page is better. Moreover the old page seems to have a slightly better conversion rate.