UDACITY

‹ Return to Classroom

Analyze A/B Test Results

REVIEW

HISTORY

Requires Changes

3 specifications require changes

This is an excellent first submission for this project. *(The bulk of the work has been completed, only minor issues remain to be resolved to meet all requirements).*

Your interpretation of the results that you derived are spot on for the work included. For example:

> *"From the histogram above, it seems that there are plenty of data from the red line to the right (alternative hypothesis), and if we calculate the p-value which is the mean of p_diffs data having greater value than the obs_diff, it is 0.903, which is greater than the Type I error rate of 0.05. Thus we cannot reject the null hypothesis, which assume that there is no difference between the treatment group and the controlgroup."*

However, *reviewers are ruled by the **Rubric*** and there is still a main requirement that has not been met:

- The models for **Part III g) and h)** are not correctly specified.
- (there are other very minor changes to make, which will take just a few minutes - *(where **"very minor issues"** take about a minute to resolve)*).

This is explained in the sections below.

Note: Given that the structure of your project is sound (there are no major changes required), it will not take long to make the changes required to meet all of the requirements.

Overall, very nice work!! Best wishes for your re-submission!

p.s if you have any questions about any aspect of this review, you can ask mentors questions on Knowledge p.s if you have any questions about any aspect of this review, you can ask mentors questions on Knowledge

---

Important

- Where changes are required, a simple list of required changes is provided.
    - *If you feel comfortable making those changes, all other comments can be ignored. They are included as guides, to help make the required changes.*
- Sections marked `Note` are **"for your information"** *(and, if you wish, can be read later, after you have finished your project).*
- Sections marked `Suggestions/Tips` detail ways that you can improve your project *(which you can incorporate into your project but are **optional**).*

Code Quality

All code cells can be run without error.

After resolving a minor issue, the code in your notebook evaluates as expected (without errors). Nice work!

However, the HTML file submitted does not have all of the results displayed. This is required.

ISSUE

*This issue relates to the version of python you are using, which is why **it is not** a required change.*

p_diffs has to be a *numpy array* to perform mathematical calculations with it *(in your notebook it is a list)*:

```
In [44]:
# proportion of the p_diffs are greater than the obs_diff
(p_diffs>obs_diff).mean()

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/var/folders/xj/2spvn73d36j93q5j6c21y3000000gn/T/ipykernel_69774/1975949983.py in <module>
      1 # proportion of the p_diffs are greater than the obs_diff
----> 2 (p_diffs>obs_diff).mean()

TypeError: '>' not supported between instances of 'list' and 'float'
```

```
In [31]:
# proportion of the p_diffs are greater than the obs_diff
(np.array(p_diffs)>obs_diff).mean()

0.9063
```

Click On Images To Enlarge Them

You should resolve this so that your code runs in future versions of python.

Also, there are 3 notebooks in your *workspace*.

All files in your workspace are included in your *workspace*. So it is not immediately clear which notebook you want *evaluated*. Leave only one notebook in your *workspace*.

CHANGE REQUIRED (VERY MINOR)

- Include a *HTML* file that shows all of the results evaluated. It is part of the submission requirements.

ADDITIONAL NOTES

Most of the test results are not evaluated in the HTML submitted:

Then I store these n_{old} 1's and 0's in the `old_page_converted` numpy array.

```
In [136]: # Simulate a Sample for the control Group
np_old = np.array(df2[df2['landing_page']=='old_page'].converted)
old_page_converted = np.random.choice([0,1], size=n_old, p=[p_old, 1-p_old])
old_page_converted
```

```
Out[136]: array([0, 1, 1, ..., 1, 1, 1])
```

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

```
In [137]: # difference in the "converted" probability for simulated samples
sam_diff = new_page_converted.mean() - old_page_converted.mean()
sam_diff
```

```
Out[137]: -0.00061035800429909415
```

h. Sampling distribution
I will 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 I used in parts (a) through (g) above.

Then I will store all (p'_{new} - p'_{old}) values in a NumPy array called `p_diffs`.

```
In [ ]: # Sampling distribution
p_diffs = []
for i 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])
    diff = new_page_converted.mean() - old_page_converted.mean()
    p_diffs.append(diff)
```

i. Histogram
I will plot a histogram of the **p_diffs**.

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

```
In [ ]: plt.hist(p_diffs)
plt.axvline(x=obs_diff, color='red') # the actual difference observed in the df2
plt.title('Differences in Convertion Rate Between New_page and Old_page')
plt.xlabel('Differences in convertion rate')
plt.ylabel('counts from the simulated data')
```

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

```
In [ ]: # proportion of the p_diffs are greater than the obs_diff
(p_diffs>obs_diff).mean()
```

From the above results, I can find that the p-value is 0.9028.

If you see the histogram above, it seems that there are plenty of data from the red line to the right (alternative hypothesis), and if we calculate the p-value which is the mean of `p_diffs` data having greater value than the `obs_diff`, it is 0.903, which is greater than the `Type I error rate` of 0.05. Thus we cannot reject the null hypothesis, which assume that there is no difference between the `treatment` group and the `control` group.

l. Using Built-in Methods for Hypothesis Testing

Here are the submission requirements, from the *Project Submission webpage:*

> 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.

TIPS

- To help develop your skills using python, I highly recommend working through the examples in this free online text
- For guides for most aspects of the *Python Programming Language* this site is a great resource

Docstrings, comments, and variable names enable readability of the code.

- The variable names that you use are clear and make it easier to follow your code.
- The comments in your code, and in your Markdown that relate to the code, makes it easy to follow your code. Very nice work!

ADDITIONAL NOTES

- In data analysis, in a work environment, commenting code, so that your colleagues understand the intent of the code that you write, is a basic necessity.
  - *(It is always more difficult to read other people's code)*.
  - It only takes a few minutes to clearly comment code like this but the benefits are substantial. Overtime you will built up a library of scripts/notebook that you will refer back to. With clearly commented code, you can scan those resources quickly.
  - Commenting code is a good habit to develop, because it communicates to colleagues, or to yourself at some future time, the intent of the code that you have written (in a succinct way that avoids you having to examine the code line by line). It is difficult to overstate the importance of clear code commenting, in a work environment, in data analysis.

Excellent work!!

Statistical Analyses

All results from different analyses are correctly interpreted.

- **In "Part II - A/B Test", student should correctly interpret the test statistic and p-value.**
- **In "Part III - A regression approach", student should correctly analyze the interaction effects on all of p-value and statistical significance to predict conversions.**

Your interpretation of the results of the three different approaches are excellent. For example:

> *"From the histogram above, it seems that there are plenty of data from the red line to the right (alternative hypothesis), and if we calculate the p-value which is the mean of p_diffs data having greater value than the obs_diff, it is 0.903, which is greater than the Type I error rate of 0.05. Thus we cannot reject the null hypothesis, which assume that there is no difference between the treatment group and the controlgroup."*

However, there are parts that were not answered:

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.

CHANGE REQUIRED (MINOR)

- *Double check* that all required interpretations are included.

All statistical numeric values are calculated correctly.
Tip: Students can optionally attempt the classroom quizzes to ensure they are calculating the right value in many cases.

All numerical analysis **upto** *Part III g)* is correct. Excellent work!

CHANGE REQUIRED (MINOR)

- Update your responses to Part III g) and h)

ADDITIONAL NOTES

- The entire analysis is answering the question: *Should I change my webpage to the new version?"*
- You are using three different methods to determine whether you should reject the Null Hypothesis that the conversion rate is the same for both pages.
- For Part III g) and h) you are **building** on the work that you did in Part III e)
  - You are now **adding** features to your regression model to see if these factors will influence your decision about changing to the new page). You do this in **two ways**:

- **g)** You add new features, **countries**, as each may show that there are different responses.

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 [37]:

```python
# Read the countries.csv
df_coun = pd.read_csv('countries.csv')
df_coun.head()
```

	user_id	country
0	834778	UK
1	928468	US
2	822059	UK
3	711597	UK
4	710616	UK

In [38]:

```python
# Join with the df2 dataframe
df_merged = df2.join(df_coun.set_index('user_id'), on='user_id')
df_merged.head()
```

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

No model is included for this part

In [39]:

```python
# Create the necessary dummy variables
df_merged = df_merged.join(pd.get_dummies(df_merged['country']))
df_merged.head()
```

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

In [40]:

```python
df_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 290584 entries, 0 to 294477
Data columns (total 11 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   user_id       290584 non-null  int64
 1   timestamp     290584 non-null  object
 2   group         290584 non-null  object
 3   landing_page  290584 non-null  object
 4   converted     290584 non-null  int64
 5   intercept     290584 non-null  int64
 6   ab_page       290584 non-null  uint8
 7   country       290584 non-null  object
 8   CA            290584 non-null  uint8
 9   UK            290584 non-null  uint8
 10  US            290584 non-null  uint8
dtypes: int64(3), object(4), uint8(4)
memory usage: 26.9+ MB
```

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.

Click On Images To Enlarge Them

- **h)** You add **interactive terms**, countries and treatment (Here is a discussion of interactive variables that uses a simple example. Here is a more technical analysis).

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]:

```
# Fit your model, and summarize the results
country_model = sm.Logit(df_merged['converted'], df_merged[['intercept', 'ab_page', 'US', 'UK']])
country_result = country_model.fit()
country_result.summary2()
```

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

Model:	Logit	Pseudo R-squared:	0.000
Dependent Variable:	converted	AIC:	212781.1253
Date:	2021-12-26 10:14	BIC:	212823.4439
No. Observations:	290584	Log-Likelihood:	-1.0639e+05
Df Model:	3	LL-Null:	-1.0639e+05
Df Residuals:	290580	LLR p-value:	0.17599
Converged:	1.0000	Scale:	1.0000
No. Iterations:	6.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
intercept	-2.0300	0.0266	-76.2488	0.0000	-2.0822	-1.9778
ab_page	-0.0149	0.0114	-1.3069	0.1912	-0.0374	0.0075
US	0.0408	0.0269	1.5161	0.1295	-0.0119	0.0934
UK	0.0506	0.0284	1.7835	0.0745	-0.0050	0.1063

This is the correct model for Part III g). Include it there

For Part III h), use this model and ADD 2 interactive variables (for the same countries).

In [42]:

```
# Exponentiate the coefficents from the summary
np.exp(country_result.params)
```

```
intercept    0.131332
ab_page      0.985168
US           1.041599
UK           1.051944
dtype: float64
```

Click On Images To Enlarge Them

TIPS

As explained above, the logit models are sequential. Your base model is correct.

- *Note: If you omit* `ab_page` *from any of your regression models then the model is no longer an* ***A/B testing*** *model.* `ab_page` *is the old page/new page variable and its significance determines the relationship between conversion and the new page.*

You now need to use that **base model** and **add variables to it** (i.e. you don't create separate models). This is because, *throughout your analysis*, the aim is always the same: **to test the effectiveness of the treatment** - so the **treatment variable** has to be in all of your models.

The extra variables are added to the base model to rule out *confounding factors* - that is, because you are using *observational data* (rather than *experimental data*), you need to ensure that there are no variables that are different in both your treatment and control groups that may *bias* the results that you get.

So, to resolve this:

1. Add the country dummy variables to your base model (instead of creating a completely new model

that excludes the treatment) - **(Part III g)** - *Already completed, move it from Part III h) to Part III g)*

2. Then, to that model (1.), add interactive dummy variables. These are simply created by multiplying the country and treatment dummy variables - for each country - **(Part III h)**):

   For example, for the US:

   ```
   df_merged['US_inter'] = df_merged['US']*df_merged['ab_page']
   ```

   and the same for `UK`.

Relevant Sections in the course are:

- Logistic Regression Models: *Lesson 16*
- Interaction between variables in Regression Models - *Lesson 15, Sections 23 to 25*
  - Even though those sections relate to the Linear Regression Model, they are still relevant here.

(Because each cohort of students have different links to course sections, unfortunately I cannot provide direct links here)

Submission Comment

"I would like to get a detailed explanation regarding the regression model and sampling distribution."

- I explained the regression model in the notes above
- There are *Four "Steps"* involved in understanding the *sampling distribution*

STEP ONE

- You have a sample of data.
- If you took a different sample, you would get different observations.
  - This ***sampling variation*** is the **only reason** that you need to perform any tests.
  - That is, if there was **no** sampling variation, you would just look at the conversion rates in your sample of data and you would be done.

STEP TWO

- The sample of data is **the only information** that you have.
- The difference in conversions rates (new page vs old page) is the **best estimate** that you have.
  - But because of *sampling variation* you cannot simply look at that number to determine if the new page is better than the old page.
  - You need to determine if it is *statistically significantly* better.

STEP THREE

- You set up your test: Your *Null* and *Alternative* hypotheses. Which are:
  - Statistically, there is no difference (I just got a difference because of the sample that I have) - **Null**
  - Statistically, the conversion rate for the new page is greater than the old page - **Alternative**
  - You did this correctly

STEP FOUR: **TEST**

- If the Null is *True* ...
- ... how *likely* is the *actual* sample estimate that I got?
    - If it is *"unlikely" (low p-value)*, I *reject* the Null
    - If it is *"likely" (high p-value)*, I *fail to reject* the Null

"likely" is given by the p-value. That is, if I assume that the Null is true, and the actual statistic is likely, that is evidence in favor of the null.

Conclusions should include both - statistical reasoning and practical reasoning for the situation.

TIP

- Always add a conclusion section to reports like this, that summarize **all** of the results in your report. *(Even if it does repeat part of your previous analysis)*
- You can think of this section as the *Executive Summary* (the part that people flip to first before reading the rest).

RESUBMIT

DOWNLOAD PROJECT