UDACITY

‹ Return to Classroom

Analyze A/B Test Results

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Felicidades 😄

You have met all the requirements for this project, I really appreciate the hard work you put into this project. Keep up the great work and good luck with your upcoming project(s). 👏

Like any other language or tool, Python has some best practices to follow before, during, and after the process of writing your code. These make the code readable and create a standard across the industry. Find some interesting resources for best coding practices in Python

- 10 Essential Python Tips And Tricks For Programmers
- 20 Python Programming Tips and Tricks for Beginners

“LEARNING IS NOT ATTAINED BY CHANCE, IT MUST BE SOUGHT FOR WITH ARDOR AND ATTENDED TO WITH DILIGENCE.” —ABIGAIL ADAMS

#UdaciousAlways 🎉

Code Quality

All code cells can be run without error.

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

When possible, it is always more computationally efficient to use NumPy built-in operations over explicit for loops. The short reason is that NumPy -based operations attack a computational problem based on vectors by computing large chunks simultaneously. Additionally, using loops to simulate 10000 can take a considerable amount of time vs using numpy .

```
new_converted_simulation = np.random.binomial(n_new, p_new, 10000)/n_new
old_converted_simulation = np.random.binomial(n_old, p_old, 10000)/n_old
p_diffs = new_converted_simulation - old_converted_simulation
```

Suggestion

I could not find comments on main code blocks, and this is very important both for the code writer and any other programmers who will work on the same project. You can refer to this article for best practices for commenting in the code.

Statistical Analyses

All results from different analyses are correctly interpreted.

Spot On!!! Great intuition with the relationship between the different hypothesis statements.

Another reason for the differences in ab testing between part II and III has to do with the type of test. In the Part III, we have a two-tailed test. For more details, Differences between 1-tailed and 2-tailed test

It was good to see "collinearity" mentioned as a problem when adding too many factors to the model. Also, a very common reason why adding too many factors to our models has a detrimental effect, is multicollinearity. Multicollinearity is when there's a close relationship between three or more variables and it's usually hard to detect. For more details, read Multicollinearity Explained and VIF/Multicollinearity

For all numeric values, you should provide the correct results of the analysis.

Awesome! Getting the stats calculations for both the simulation and z-test correct is difficult at this stage. 💪

Conclusions should include not only statistical reasoning, but also practical reasoning for the situation.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START