Analyze_ab_test_results_notebook

May 8, 2020

0.1 Analyze A/B Test Results

You may either submit your notebook through the workspace here, or you may work from your local machine and submit through the next page. Either way assure that your code passes the project RUBRIC. **Please save regularly.**

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

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.

```
In [144]: #import packages we need for this analysis
          import pandas as pd # handle and wrangle data
          import numpy as np  # create arrays
          import random # provides access to functions that support many operations.
          import matplotlib.pyplot as plt #plot data
          %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 [145]: # Load your data and print out a few lines. Perform operations to inspect data
          df = pd.read_csv("ab_data.csv") #read csv
          df.head() #print the first row of the dataframe
```

```
Out[145]:    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 [146]: # look at the shape and return the number of row and column
          df.shape
```

```
Out[146]: (294478, 5)
```

0.2.1 This dataset has 294478 rows, and 5 columns.

c. The number of unique users in the dataset.

```
In [147]: # number of unique users
          df.user_id.nunique()
```

```
Out[147]: 290584
```

0.2.2 There are 290583 of unique users in this dataset.

d. The proportion of users converted.

```
In [148]: #find the mean
          df.converted.mean()
```

```
Out[148]: 0.11965919355605512
```

0.2.3 The proportion of users converted = 12 %

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

```
In [149]: ## rows where treatment users are land incorrectly on old_page

          df.query('landing_page == "new_page" and group == "contrl"').user_id.size
          df.query('landing_page == "old_page" and group == "treatment"').user_id.nunique()
```

```
Out[149]: 1965
```

0.2.4 The number of times treatment users on old_page are 1965.

```
In [85]: # rows where control users are land incorrectly on new_page
         df.query('landing_page == "old_page" and group == "treatment"').user_id.size
         df.query('landing_page == "new_page" and group == "control"').user_id.nunique()

Out[85]: 1928
```

0.2.5 The number of times control user on new_page are 1928.

```
In [150]: #counting rows where the new_page and treatment don't line up
          df.query('group=="treatment" and landing_page != "new_page" or group=="control" and la

Out[150]: user_id         3893
          timestamp       3893
          group           3893
          landing_page    3893
          converted       3893
          dtype: int64
```

0.2.6 The number of times the new_page and treatment don't match is 3893.

f. Do any of the rows have missing values?

```
In [151]: #Counts all null values
          df.isnull().sum()

Out[151]: user_id         0
          timestamp       0
          group           0
          landing_page    0
          converted       0
          dtype: int64
```

```
In [152]: # see the column info and null values in the dataset
          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
```

0.2.7 The rows have no missing values.

2. For the rows where **treatment** does not match with **new_page** or **control** does not match with **old_page**, we cannot be sure if this row truly received the new or old page. Use **Quiz 2** in the classroom to figure out 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 [154]: df2 = df[((df['group'] == 'treatment') & (df['landing_page'] == 'new_page')) | ((df['g
```

```
In [155]: # Double Check all of the correct rows were removed - this should be 0
          df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) == False].s

Out[155]: 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**?

```
In [156]: # find out unique user_ids in df2
          df2.user_id.nunique()

Out[156]: 290584
```

0.2.8 There are 290584 unique user_ids are in df2.

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

```
In [157]: # look at the shape and return the number of row and column
          df2.shape

Out[157]: (290585, 5)
```

```
In [158]: # inspect repeated user_id

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

Out[158]: 1899    773192
          2893    773192
          Name: user_id, dtype: int64
```

0.2.9 user_id 773192 is repeated in df2.

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

```
In [159]: #details of rows with repeated user ids
          df2[df2.duplicated(['user_id'], keep=False)]

Out[159]:       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**, but keep your dataframe as **df2**.

```
In [160]: # Remove the row with a duplicate user_id
          df2 = df2.drop_duplicates(subset = 'user_id')

In [161]: # inspect number of entries
          df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 290584 entries, 0 to 294477
Data columns (total 5 columns):
user_id         290584 non-null int64
timestamp       290584 non-null object
group           290584 non-null object
landing_page    290584 non-null object
converted       290584 non-null int64
dtypes: int64(2), object(3)
memory usage: 13.3+ MB
```

```
In [162]: #check unique value of user id
          len(df['user_id'].unique())

Out[162]: 290584
```

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?

```
In [163]: #find the mean
          df2.converted.mean()

Out[163]: 0.11959708724499628
```

0.2.10 The probbability of an individual converting regardless of the page they receive = 12%

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

```
In [164]: # compute the statistics using describe function

          df_grp = df.groupby('group')
          df_grp.describe()

Out[164]:           converted                                                      user_id  \
                       count      mean       std  min  25%  50%  75%  max     count
          group
          control   147202.0  0.120399  0.325429  0.0  0.0  0.0  0.0  1.0  147202.0
          treatment 147276.0  0.118920  0.323695  0.0  0.0  0.0  0.0  1.0  147276.0

                                                                                 \
                    mean           std       min       25%       50%
group
control    788123.098035  91278.896888  630002.0  709287.0  788053.5
treatment  787825.226283  91142.800641  630000.0  708729.5  787837.5


                 75%       max
group
control    867155.50  945998.0
treatment  866693.75  945999.0
```

0.2.11 The probability of an individual in control group = 12%

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

```
In [165]: # compute the statistics using describe function

          df_grp = df.groupby('group')
          df_grp.describe()
```

```
Out[165]:           converted                                                user_id  \
                count      mean       std  min  25%  50%  75%  max     count
          group
          control    147202.0  0.120399  0.325429  0.0  0.0  0.0  0.0  1.0  147202.0
          treatment  147276.0  0.118920  0.323695  0.0  0.0  0.0  0.0  1.0  147276.0

                                                                                \
                              mean           std       min       25%       50%
          group
          control    788123.098035  91278.896888  630002.0  709287.0  788053.5
          treatment  787825.226283  91142.800641  630000.0  708729.5  787837.5


                           75%       max
          group
          control    867155.50  945998.0
          treatment  866693.75  945999.0
```

0.2.12 The probability of an individual in treatment group = 11.89%

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

```
In [166]: df2.query('landing_page=="new_page"').count()/df2.shape[0]
```

```
Out[166]: user_id         0.500062
          timestamp       0.500062
          group           0.500062
          landing_page    0.500062
```