ORIE 5355/INFO 5370 HW 4: Experimentation

- Name:
- Net-id:
- Date:
- Late days used for this assignment:
- Total late days used (counting this assignment):
- People with whom you discussed this assignment:

After you finish the homework, please complete the following (short, anonymous) post-homework survey: https://forms.gle/YFrnPW63HsmrTZL38 and include the survey completion code below.

Question 0 [2 points]

Survey completion code:

Conceptual component [6 points]

Personal reflection

Think back to a time that you wanted to evaluate an idea or product. If you have not had such an idea before, you may answer these questions about an article in the news that reported such a feature, or a feature that you think might be in deployment at a company or organization with which you interact (for example, Amazon, Google, Facebook, etc).

Briefly summarize the scenario in no more than two sentences.

In [ ]:

What was the objective that you cared about/wanted to optimize with the product/idea? What was the measurement that you could feasibly measure during the experimental period? In what ways did the measurement not match the objective you cared about? Answer in no more than 3 sentences.

In [ ]:

Did the setting have interference (such as due to a network setting, interference through a 2 sided marketplace or capacity constraints, etc.)? If so, how did it effect your experimental design and results? If your answer is no, why are you sure that such interference did not happen? Answer in no more than 3 sentences.

In [ ]:

Given what we have learned in class so far, what would you do differently if faced with the same

scenario again? Answer in no more than 3 sentences.

In [ ]:

Programming component

Helper code

In [52]:
```
import numpy as np
import pandas as pd
import os, sys, math
import matplotlib.pyplot as plt
```

In [53]:
```
df_headlines = pd.read_csv('headline-experiment-heds.csv')
df = pd.read_csv('headline-experiment-impressions.csv')
```

In [54]:
```
for x in df_headlines.hed:
    print(x)
```

```
She's Not Just Destined For Greatness, She's Destined To Do Great Things For Women
This Young Woman Just Took Silicon Valley By Storm And She's Not Stopping There
Feminism 101: This Girl Is Going Places And She's Taking Other Girls With Her
Remember When Math Was "Too Hard" For The Ladies? Not So Much.
```

In [55]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14950 entries, 0 to 14949
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   hed     14950 non-null  int64
 1   click   14950 non-null  int64
dtypes: int64(2)
memory usage: 233.7 KB
```

In [56]:
```
df.hed.value_counts()
```

Out[56]:
```
3    3763
1    3756
4    3737
2    3694
Name: hed, dtype: int64
```

In [57]:
```
df.groupby('hed')['click'].mean()
```

Out[57]:
```
hed
1    0.010650
2    0.006497
3    0.010098
```

```
4    0.004549
Name: click, dtype: float64
```

df_headlines has a list of 4 headlines for the same article from Upworthy. df is a dataframe where each row represents a user. `hed` indicates which headline was shown to the user, and `click` is a binary indicator for whether the user clicked on the headline. A `1` represents a click, and so, for example, headline 2 was clicked on 0.6\% of the time. Each headline was shown to about 3700 users.

I recommend reading the following post: https://towardsdatascience.com/ab-testing-with-python-e5964dd66143

In this homework, we will only be working with the first two headlines:

In [58]:
```
df = df.query('hed==1 or hed==2')
```

In [59]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7450 entries, 1 to 14949
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   hed     7450 non-null   int64
 1   click   7450 non-null   int64
dtypes: int64(2)
memory usage: 174.6 KB
```

Problem 1: Simple A/B tests, and dependence on sample size

Problem 1a: Simple A/B testing

First, what do the results look like if we use all the data?

Here, you will want to use the functions under "4. Testing the hypothesis" in the above blog post. In particular you will want to test the "1 sided" hypothesis that headline 1 is better than headline 2. (In statsmodels.stats.proportion.proportions_ztest, use alternative='larger', and put headline 1 first in the data.)

https://www.statsmodels.org/stable/generated/statsmodels.stats.proportion.proportions_ztest.html

If you use all the data (all the entries in the dataframe), what is the mean click through rate for each headline?

In [ ]:
```

```

If you use all the data, what is the p-value for the hypothesis that the first headline is better than the second headline?

In [ ]:
```

```

If you use all the data, what are the confidence intervals for the click through rates for each headline?

In [ ]:

Interpret the above, in no more than 3 sentences

In [ ]:

Problem 1b: Experimentation with lower sample sizes

Now, we'll see how often we would make the "wrong" decision if we instead had run an experiment with a lower sample size. We do this via a method called "bootstrapping" -- we 're-sample' from the data that we actually saw, in order to estimate what would have happened via counter-factual experiments.

Complete the following function, which does the following: it simulates 1000 fake experiments; each fake experiment, we sample `overall_sample_size` users and pretend that those users made up the experiment.

We want to store:

- the distribution of click-through-rate estimates for each headline (we do this for you)
- the fraction of experiments in which headline 1 was found to be better than headline 2

In [61]:

```
def get_estimates_from_bootstrapping(df, overall_sample_size = 100):
    estimates = {hed: [] for hed in df.hed.unique()} # for each headline, store the mea
    number_of_headlines_1_better_than_2 = 0
    for _ in range(1000): # simulate 1000 fake experiments ("bootstrapping")
        df_sample = df.sample(overall_sample_size)
        means = df_sample.groupby('hed')['click'].mean()
        for en, mean in enumerate(means):
            estimates[en+1].append(mean)
        ### TODO complete code here for number_of_headlines_1_better_than_2

    return estimates, number_of_headlines_1_better_than_2/1000
```

In [ ]:

For each of overall_sample_size in [100, 1000, 5000] plot a histogram of the estimates for each headline. You should have 3 plots, each plot corresponding to 1 sample size number and containing 2 histograms, 1 for each headline.

In [ ]:

In [62]:
```
sample_size_numbers = list(range(100, 6000, 500))
```

For each of overall_sample_size in sample_size_numbers, get the fraction of experiments in which headline 1 was found to be better than headline 2. Plot a line plot where the X axis is the sample size, and the Y axis is the fraction of times. Note that this code might take a minute or so to run. Note: your line plot should be increasing in the sample size (Why?)

In [ ]:
```

```

Intepret the above, in no more than 3 sentences.

In [ ]:
```

```

Problem 2: Peeking (6 points)

Now, we'll illustrate the problem of "peeking" in experiments. Suppose you're a headline writer, and you personally wrote headline 2 and are now running the AB test. So, you have a maximum experiment budget of 2000 users. Each user comes in sequentially and is assigned either the first or second headline. Now, you also realize that experimentation is wasteful, and so you want to minimize the amount of time you're spending in the experiment.

So, you do the following: after each 20th user comes in and either clicks on the headline or doesn't, you run a 1-sided hypothesis test that headline 2 is better than headline 1 (similar to Problem 1a, but now testing headline 2 being better). If the p-value is less than 0.05, then you declare victory: that headline 2 is statistically significantly better than headline 1, and so you stop the experiment. If the p-value is greater than 0.05, you continue the experiment.

Now, we'll want to calculate: how often does the above procedure lead to you declaring victory, that headline 2 is statistically significantly better than headline 1?

Here, we will walk you through simulating the above procedure. As before, we will simulate 1000 fake experiments, to get a good estimate of what the above procedure behaves like.

Finish the below code, to calculate number_of_headlines_2_better_than_1 using the above procedure

In [66]:
```
number_of_headlines_2_better_than_1 = 0

for _ in range(1000): # simulate 1000 fake experiments
    df_sample = df.sample(2000)
    for number_users in range(20, 2001, 20):
        df_users_to_far = df_sample.iloc[0:number_users] #grab the first number_users u
        #TODO: calculate p-value of experiment, the 1-sided hypothesis test that headli
        #TODO potentially end experiment. The "break" keyword in python might come in h
        # Note that you want to break the inner for loop but not the outer loop (think
        # Note, you sometimes may get "unlucky", and all the first 20 users received th
```

In [ ]:
```

```

What fraction of the time does the above procedure declare that headline 2 is better than headline 1?

In [ ]:

Interpret the above answer, in no more than 3 sentences. What went wrong?

In [ ]: