ba-860-assignment-1

April 1, 2024

0.1 TEAM MEMBERS: SAHASRA KONKALA, JASMINE GOHIL, MAHIKA BHARTARI

Link to Colab File here

0.2 LOADING THE DATASET:

[ ]:
```
import pandas as pd
```

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[ ]:
```
df = pd.read_csv('/content/drive/MyDrive/BA-860/PS1-MSBA-4000416-loc.csv')
```

0.3 EXPLORING THE DATASET:

[ ]:
```
df
```

[ ]:

	country	treatment	saw_ads	sales	past_sales
0	US	1	0	0.00	0.0
1	US	0	1	0.00	0.0
2	FR	1	1	0.00	0.0
3	US	0	1	0.00	0.0
4	UK	1	1	16.65	0.0
…	…	…	…	…	…
4000411	US	0	0	0.00	0.0
4000412	US	0	0	0.00	0.0
4000413	US	1	1	0.00	0.0
4000414	UK	1	0	0.00	0.0
4000415	US	0	1	0.00	0.0

[4000416 rows x 5 columns]

[ ]:
```
us_count = ((df['country'] == 'US') & (df['saw_ads'] == 1) & (df['treatment']␣
↪== 1)).sum()
```

```
us_count
```

[ ]: 1084882

[ ]:
```
uk_count = ((df['country'] == 'UK') & (df['saw_ads'] == 1) & (df['treatment']␣
    ↪== 1)).sum()
uk_count
```

[ ]: 525201

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000416 entries, 0 to 4000415
Data columns (total 5 columns):
 #   Column      Dtype
---  ------      -----
 0   country     object
 1   treatment   int64
 2   saw_ads     int64
 3   sales       float64
 4   past_sales  float64
dtypes: float64(2), int64(2), object(1)
memory usage: 152.6+ MB
```

[ ]:
```
df.describe()
```

[ ]:

	treatment	saw_ads	sales	past_sales
count	4.000416e+06	4.000416e+06	4.000416e+06	4.000416e+06
mean	5.997951e-01	7.539341e-01	1.257225e+00	5.716948e-01
std	4.899398e-01	4.307175e-01	9.824927e+00	5.769595e+00
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
50%	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
75%	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
max	1.000000e+00	1.000000e+00	2.331490e+03	1.307610e+03

[ ]:
```
df.isna().sum()
```

[ ]:
```
country       0
treatment     0
saw_ads       0
sales         0
past_sales    0
dtype: int64
```

1 Question 1.

Before analyzing the experiment's results, we want to verify that the experiment properly randomized users. To do this, we compare the treatment and control groups by the users' characteristics.

1.1 1a.

Verify the randomization by user country location i. (10 pts) Use the difference-in-means estimator to compare the proportions of UK and US consumers in Treatment versus Control groups. (Hint: create two indicator variables using the "country" variable: "IsUK" and "IsUS", and compute the difference for each variable separately).

[ ]:
```
import pandas as pd
import warnings
import numpy as np
import scipy.stats as stat
```

[ ]:
```
df['IsUK'] = df['country'].apply(lambda x: 1 if x == 'UK' else 0)
df['IsUS'] = df['country'].apply(lambda x: 1 if x == 'US' else 0)

vT = df.loc[(df['treatment']==1)&(df['saw_ads']==1), 'IsUK']
vC = df.loc[(df['treatment']==0)&(df['saw_ads']==1), 'IsUK']

diff_uk = np.mean(vT) - np.mean(vC)
se_uk = np.sqrt(np.std(vT)**2/len(vT) + np.std(vC)**2/len(vC))

CI_lower_uk = np.round(diff_uk - 1.96*se_uk, 3)
CI_upper_uk = np.round(diff_uk + 1.96*se_uk, 3)

print('Mean difference: ', np.round(diff_uk, 3))
print('Standard error: ', np.round(se_uk, 3))
print('95% CI: [', CI_lower_uk, ', ', CI_upper_uk, ']')
```

```
Mean difference:  -0.0
Standard error:  0.001
95% CI: [ -0.001 ,  0.001 ]
```

[ ]:
```
vT1 = df.loc[(df['treatment']==1)&(df['saw_ads']==1), 'IsUS']
vC1 = df.loc[(df['treatment']==0)&(df['saw_ads']==1), 'IsUS']

diff_us = np.mean(vT1) - np.mean(vC1)
se_us = np.sqrt(np.std(vT1)**2/len(vT1) + np.std(vC1)**2/len(vC1))

CI_lower_us = np.round(diff_uk - 1.96*se_us, 3)
CI_upper_us = np.round(diff_uk + 1.96*se_us, 3)

print('Mean difference: ', np.round(diff_us, 3))
print('Standard error: ', np.round(se_us, 3))
```

```
print('95% CI: [', CI_lower_us, ', ', CI_upper_us, ']')
```

```
Mean difference:  -0.0
Standard error:  0.001
95% CI: [ -0.001 ,  0.001 ]
```

- From the above we can see that the randomization was done perferctly since the mean difference is 0. It also means that there is no difference in proportion of UK, US customers between control and the treatment groups.
- Standard error of 0.001 states that the mean difference obtained is very accurate and there is very less chance of it changing due to random chance.
- At 95% CI, our results show that there is no statistical significance difference in the proportions of UK, US customers between the Treatment and Control groups.

1.2 1b.

Verify the randomization by past sales

i. (5 pts) Use the difference-in-means estimator to compare the average sales in the 2 weeks before the experiment in the Treatment versus Control groups.

```
[ ]: avg_psales_t = df[df['treatment'] == 1]['past_sales'].mean()
avg_psales_c = df[df['treatment'] == 0]['past_sales'].mean()

avg_psales_t_std = df[df['treatment'] == 1]['past_sales'].std()
avg_psales_c_std = df[df['treatment'] == 0]['past_sales'].std()

n_treatment = len(df[df['treatment'] == 1])
n_control = len(df[df['treatment'] == 0])

diff_avg_psales = avg_psales_t - avg_psales_c

se_psales = np.sqrt((avg_psales_t_std**2 / n_treatment) + (avg_psales_c_std**2 /
 ↪ n_control))

CI_lower_psales = np.round(diff_avg_psales - 1.96*se_psales, 3)
CI_upper_psales = np.round(diff_avg_psales + 1.96*se_psales, 3)


print("Difference in average sales before the experiment:", diff_avg_psales)
print('Standard error: ', np.round(se_psales, 3))
print('95% CI: [', CI_lower_psales, ', ', CI_upper_psales, ']')
```

```
Difference in average sales before the experiment: -0.0012113621629216142
Standard error:  0.006
95% CI: [ -0.013 ,  0.01 ]
```

- The negative symbol in "Difference in average sales before the experiment" through past sales shows that the treatment group has lesser sales compared to the control group.

- The standard error 0.006 indicates the level of variability or uncertainty associated with this estimated difference.
- The 95% CI [-0.013, 0.01] here indicates that we are 95% confident that the true difference in average sales between the treatment and control groups falls within this range. As the range also includes 0, it means that there is no significant statistical difference in average sales between the two groups before the experiment.

ii. (5 pts) What do you conclude about the validity of the randomization in terms of past sales?

- In terms of past sales also the randomization check proved to be accurate as the standard error is quite low and the confidence interval range includes 0 showing that there is no significant statistical difference in average sales between the two groups before the experiment. The mean difference is also very minute so we can therefore say that there is no evidence of significant difference in average sales between the treatment and control groups before the experiment.

#Question 2

(10 pts) What would be your estimate for the ad effect on per-user sales if the experiment did not have ghost ads? Compute the Intention-to-Treat estimate of the ad effect. ##

[ ]:
```
avg_sales_t = df[df['treatment'] == 1]['sales'].mean()
avg_sales_c = df[df['treatment'] == 0]['sales'].mean()

se_sales_t = df[df['treatment'] == 1]['sales'].std() / np.sqrt(n_treatment)
se_sales_c = df[df['treatment'] == 0]['sales'].std() / np.sqrt(n_control)
se_diff_sales = np.sqrt(se_sales_t**2 + se_sales_c**2)

itt_estimate = avg_sales_t - avg_sales_c

ci_difference_avg_sales = np.round((itt_estimate - 1.96 * se_diff_sales,
                           itt_estimate + 1.96 * se_diff_sales), 3)

lift = (itt_estimate / avg_sales_c) * 100


print("Intention-to-Treat estimate of ad effect on per-user sales:", np.
 ↪round(itt_estimate, 3))
print('Standard error: ', np.round(se_diff_sales, 3))
print('95% CI:', ci_difference_avg_sales)
print('Lift: ', np.round(lift, 3))
```

```
Intention-to-Treat estimate of ad effect on per-user sales: 0.498
Standard error:  0.01
95% CI: [0.479 0.517]
Lift:  51.955
```

- The ITT of 0.498 tells that when a person is assigned to treatment group, it would result in a 0.498 units increase in sales.
- The 95% confidence interval [0.479, 0.517] tells that we are 95% confident that the true effect of the ads on per-user sales falls within this range.

- Lift value of 51.955 shows the percentage increase in average user sales when in a treatment group compared to the control group.
- Therefore without the ghost ads the estimate of the ad effect on avg sales is 0.498.

2 Question 3

This experiment used ghost ads. Verify if the ghost ads were deployed the same way as the retailer ads by comparing the exposed users between the Treatment and Control groups. ##

##3a.

(15 pts) Verify the equivalence of Treatment exposed and Control exposed users by user location (repeat both steps in question 1A).

[ ]:
```
#counting exposed users in treatment group for country
treatment_exposed_country = df[(df['treatment'] == 1) & (df['saw_ads'] == 1)].
↪groupby('country').size()

#counting exposed users in control group for country
control_exposed_country = df[(df['treatment'] == 0) & (df['saw_ads'] == 1)].
↪groupby('country').size()

print("Treatment group exposed by country:")
print(treatment_exposed_country)
print("\nControl group exposed by country:")
print(control_exposed_country)
```

```
Treatment group exposed by country:
country
FR     199043
UK     525201
US    1084882
dtype: int64

Control group exposed by country:
country
FR    132547
UK    350505
US    723872
dtype: int64
```

[ ]:
```
#ratio of treatment to control for every group
ratio_by_country = treatment_exposed_country / control_exposed_country

print("\nRatio of treatment to control for every group:")
print(ratio_by_country)
```

```

Ratio of treatment to control for every group:
```

```
country
FR    1.501679
UK    1.498412
US    1.498721
dtype: float64
```

- For all the countries, the treatment group is 1.5 times larger than the control showing equivalence among the distributions.

```
[ ]: df['IsUK'] = df['country'].apply(lambda x: 1 if x == 'UK' else 0)
df['IsUS'] = df['country'].apply(lambda x: 1 if x == 'US' else 0)
df['IsFR'] = df['country'].apply(lambda x: 1 if x == 'FR' else 0)

vT = df.loc[(df['treatment']==1)&(df['saw_ads']==1), 'IsUK']
vC = df.loc[(df['treatment']==0)&(df['saw_ads']==1), 'IsUK']

diff_uk = np.mean(vT) - np.mean(vC)
se_uk = np.sqrt(np.std(vT)**2/len(vT) + np.std(vC)**2/len(vC))

CI_lower_uk = np.round(diff_uk - 1.96*se_uk, 3)
CI_upper_uk = np.round(diff_uk + 1.96*se_uk, 3)

print('Mean difference: ', np.round(diff_uk, 3))
print('Standard Error:', np.round(diff_uk, 3))
print('Lower 95% CI (UK)', np.round(CI_lower_uk,3))
print(' Upper 95% CI (UK)', np.round(CI_upper_uk,3))
```

```
Mean difference:  -0.0
Standard Error: -0.0
Lower 95% CI (UK) -0.001
 Upper 95% CI (UK) 0.001
```

```
[ ]: vT = df.loc[(df['treatment']==1)&(df['saw_ads']==1), 'IsUS']
vC = df.loc[(df['treatment']==0)&(df['saw_ads']==1), 'IsUS']

diff_us = np.mean(vT) - np.mean(vC)
se_us = np.sqrt(np.std(vT)**2/len(vT) + np.std(vC)**2/len(vC))

CI_lower_us = np.round(diff_us - 1.96*se_us, 3)
CI_upper_us = np.round(diff_us + 1.96*se_us, 3)

print('Mean difference: ', np.round(diff_us, 3))
print('Standard Error:', np.round(se_us, 3))
print('Lower 95% CI (US)', np.round(CI_lower_us,3))
print('Upper 95% CI (US)', np.round(CI_upper_us,3))
```

```
Mean difference:  -0.0
Standard Error: 0.001
Lower 95% CI (US) -0.001
```

```
Upper 95% CI (US) 0.001
```

[ ]:
```
vT = df.loc[(df['treatment']==1)&(df['saw_ads']==1), 'IsFR']
vC = df.loc[(df['treatment']==0)&(df['saw_ads']==1), 'IsFR']

diff_fr = np.mean(vT) - np.mean(vC)
se_fr = np.sqrt(np.std(vT)**2/len(vT) + np.std(vC)**2/len(vC))

CI_lower_fr = np.round(diff_fr - 1.96*se_fr, 3)
CI_upper_fr = np.round(diff_fr + 1.96*se_fr, 3)

print('Mean difference: ', np.round(diff_fr, 3))
print('Standard Error:', np.round(se_fr, 3))
print('Lower 95% CI (FR)', np.round(CI_lower_fr,3))
print(' Upper 95% CI (FR)', np.round(CI_upper_fr,3))
```

```
Mean difference:  0.0
Standard Error: 0.0
Lower 95% CI (FR) -0.001
 Upper 95% CI (FR) 0.001
```

- For all countries, namely US, UK and France- the mean differnece is 0.0 showing that there is no differenec between the treatment & control groups of all the countires (except the treatment itself). There seems to be no standard error. Since the Confidence Interval contains 0 we cannot be confident that there is a significant difference between the treatment and control groups for all the countries.

2.1 3b.

(10 pts) Verify the equivalence of Treatment exposed and Control exposed users by past sales (repeat both steps in question 1B).

[ ]:
```
#exposed users by past sales in Treatment group
treatment_exposed_past_sales = df.loc[(df['treatment'] == 1) & (df['saw_ads']␣
↪== 1)]['past_sales'].mean()

#exposed users by past sales in Control group
control_exposed_by_past_sales = df.loc[(df['treatment'] == 0) & (df['saw_ads']␣
↪== 1)]['past_sales'].mean()
diff_avg_psales = treatment_exposed_past_sales - control_exposed_by_past_sales

print("Difference in average sales before the experiment:",np.
↪round(diff_avg_psales,3))
```

```
Difference in average sales before the experiment: -0.007
```

- The Average Treatment Effect of -0.007 suggests that, based on past sales data, there was a decrease in average sales before the experiment by approximately 0.007 units.

[ ]:
```python
import scipy.stats as stats
#point estimate
point_estimate = diff_avg_psales

#standard error
std_error = np.sqrt(df.loc[(df['treatment'] == 1) & (df['saw_ads'] ==␣
↪1)]['past_sales'].var() / len(df.loc[(df['treatment'] == 1) & (df['saw_ads']␣
↪== 1)]) +
                    df.loc[(df['treatment'] == 0) & (df['saw_ads'] ==␣
↪1)]['past_sales'].var() / len(df.loc[(df['treatment'] == 0) & (df['saw_ads']␣
↪== 1)]))

#Calculate degrees of freedom
df1 = len(df.loc[(df['treatment'] == 1) & (df['saw_ads'] == 1)]) + len(df.
↪loc[(df['treatment'] == 0) & (df['saw_ads'] == 1)]) - 2

#95% confidence interval
margin_of_error = stats.t.ppf(0.975, df1) * std_error
lower_bound = point_estimate - margin_of_error
upper_bound = point_estimate + margin_of_error
lower_bound = np.round(lower_bound, 3)
upper_bound = np.round(upper_bound, 3)

print("Point estimate:", np.round(point_estimate, 3))
print("Standard error:",np.round(std_error, 3))
print("95% confidence interval: [", lower_bound, ", ", upper_bound, "]")
```

```
Point estimate: -0.007
Standard error: 0.007
95% confidence interval: [ -0.021 ,  0.006 ]
```

- The Standard error seems to be less. Since the confidence interval contains both positive and negative values, it suggests that we cannot be confident that there is a significant difference in the outcome between the treatment and control groups.

3 Question 4.

(10 pts) How does your ad effect estimate change when you use the ghost ads? Compute the Treatment on Treated (TOT) estimate for users who saw ads. ##

TOT = (Mean outcome in Treatment group among treated) - (Mean outcome in Control group among treated)

[ ]:
```python
mean_sales_treatment_ = df[(df['treatment'] == 1)]['sales'].mean()
mean_sales_control_ = df[(df['treatment'] == 0)]['sales'].mean()

#ATE estimate
ATE_estimate = mean_sales_treatment_ - mean_sales_control_
```

```
print("ATE estimate for Ghost ads:", round(ATE_estimate, 3))
```

```
ATE estimate for Ghost ads: 0.498
```

```
[ ]: mean_sales_treatment_exposed = df[(df['treatment'] == 1) & (df['saw_ads'] ==␣
      ↪1)]['sales'].mean()
     mean_sales_control_exposed = df[(df['treatment'] == 0) & (df['saw_ads'] ==␣
      ↪1)]['sales'].mean()

     #TOT estimate
     TOT_estimate_saw_ads = mean_sales_treatment_exposed - mean_sales_control_exposed

     print("Treatment on Treated (TOT) estimate for people who saw the ads:",␣
      ↪round(TOT_estimate_saw_ads, 3))
```

```
Treatment on Treated (TOT) estimate for people who saw the ads: 0.67
```

- The Treatment on Treated estimate of 0.67 suggests that among those who were exposed to the ads (the treated group), sales increased by 0.67 units on average compared to what it would have been if they hadn't been exposed to the ads.

```
[ ]: n_treatment = len(df[(df['treatment'] == 1) & (df['saw_ads'] == 1)])
     n_control = len(df[(df['treatment'] == 0) & (df['saw_ads'] == 1)])


     mean_sales_treatment_exposed = df[(df['treatment'] == 1) & (df['saw_ads'] ==␣
      ↪1)]['sales'].mean()
     mean_sales_control_exposed = df[(df['treatment'] == 0) & (df['saw_ads'] ==␣
      ↪1)]['sales'].mean()

     # TOT estimate
     TOT_estimate_saw_ads = mean_sales_treatment_exposed - mean_sales_control_exposed
     print("Treatment on Treated (TOT) estimate for people who saw the ads:",␣
      ↪round(TOT_estimate_saw_ads, 3))


     avg_tot_t_std = df[(df['treatment'] == 1) & (df['saw_ads'] == 1)]['sales'].std()
     avg_tot_c_std = df[(df['treatment'] == 0) & (df['saw_ads'] == 1)]['sales'].std()
     se = np.sqrt((avg_tot_t_std**2 / n_treatment) + (avg_tot_c_std**2 / n_control))
     CI_lower = TOT_estimate_saw_ads - 1.96 * se
     CI_upper = TOT_estimate_saw_ads + 1.96 * se

     # Print the results
     print("Point Estimate:", round(TOT_estimate_saw_ads, 3))
     print("Standard Error:", round(se, 3))
     print("95% Confidence Interval: [", round(CI_lower, 3), ",", round(CI_upper,␣
      ↪3), "]")
```

```
Treatment on Treated (TOT) estimate for people who saw the ads: 0.67
Point Estimate: 0.67
Standard Error: 0.011
95% Confidence Interval: [ 0.648 , 0.692 ]
```

- The standard error seems to be less. Since this cofidence interval is entirely positive and does not include zero, it suggests that we can be confident that the treatment (seeing the ads) had a statistically significant positive effect on sales for the individuals who were exposed to them.

[ ]:
```
#lift for users who saw ads(TOT)
lift_exposed = ( TOT_estimate_saw_ads/ mean_sales_control_exposed) * 100
lift_exposed = lift_exposed.round(3)
print("Lift for users who saw ads:", lift_exposed)
```

```
Lift for users who saw ads: 70.321
```

- There was a 70.321% increase in sales among individuals who were exposed to the ads compared to those who were not exposed to the ads.This means that the ads had a substantial positive impact on sales, resulting in a significant increase in sales among the group that saw the ads compared to the group that did not.

4 Question 5.

(5 pts each) Questions 2 and 4 asked you the ad effect on per-user sales. What is the effect of the ad campaign on total sales (i.e. gross revenue)? (Hint: this question asks you to calculate the total incremental gross revenue due to the ad campaign.) ##

4.1 5a.

Compute the effect using the ITT estimate.

[ ]:
```
total_sales_treatment = df.loc[df['treatment'] == 1, 'sales'].sum()
total_sales_control = df.loc[df['treatment'] == 0, 'sales'].sum()
incremental_gross_revenue =  np.round(total_sales_treatment -␣
 ↪total_sales_control, 3)

print("Incremental Gross Revenue due to the ad campaign:",␣
 ↪incremental_gross_revenue)
```

```
Incremental Gross Revenue due to the ad campaign: 1960249.89
```

- The result indicates that the incremental gross revenue generated by the ad campaign over the span of two weeks is $1,960,249.89. This figure represents the additional revenue attributed directly to the advertisement campaign during this specific timeframe.

[ ]:
```
T_sales = df.loc[df['treatment'] == 1, 'sales']
C_sales = df.loc[df['treatment'] == 0, 'sales']

diff = np.mean(T_sales) - np.mean(C_sales)
```

```
std_t_sales = T_sales.std()
std_c_sales = C_sales.std()
se = np.sqrt(std_t_sales**2 + std_c_sales**2)

CI_lower = np.round(diff - 1.96*se, 3)
CI_upper = np.round(diff + 1.96*se, 3)

lift_itt = (diff / np.mean(df[df['treatment'] == 0]['sales'])) * 100

print('Point Estimate of the Mean Difference (ITT Estimate): ', np.round(diff,␣
↪3))
print('Standard error (ITT Estimate): ', np.round(se, 3))
print('95% CI: [', CI_lower, ', ', CI_upper, ']')
print('Lift: {:.2f}%'.format(lift_itt))
```

```
Point Estimate of the Mean Difference (ITT Estimate):  0.498
Standard error (ITT Estimate):  13.608
95% CI: [ -26.174 ,  27.17 ]
Lift: 51.95%
```

- **Point Estimate of the Mean Difference (ITT Estimate):** The positive value indicates that, on average, the treatment group had slightly higher sales compared to the control group.
- **Standard Error (ITT Estimate):** The standard error associated with the ITT estimate is 13.608. This value reflects the variability or uncertainty in the estimate of the mean difference. A higher standard error suggests more uncertainty in the estimate.
- **95% Confidence Interval (ITT Estimate):** Since this interval includes both positive and negative values, it indicates that the observed difference is not statistically significant, and there is considerable uncertainty about the true effect of the ad campaign.
- **Lift:** The lift is calculated as 51.95%. It represents the percentage increase in sales in the treatment group compared to the control group. In this context, a lift of 51.95% suggests that, on average, the treatment group experienced a 51.95% increase in sales compared to the control group.

4.2 5b.

Compute the effect using the TOT estimate.

[ ]:
```
TOT_T_sales = df.loc[(df['treatment'] == 1) & (df['saw_ads'] == 1), 'sales']
TOT_C_sales = df.loc[(df['treatment'] == 0) & (df['saw_ads'] == 1), 'sales']

tot_diff = np.mean(TOT_T_sales) - np.mean(TOT_C_sales)
std_tot_t_sales = TOT_T_sales.std()
std_tot_c_sales = TOT_C_sales.std()
tot_se = np.sqrt(std_tot_t_sales**2 + std_tot_c_sales**2)

CI_tot_lower = np.round(tot_diff - 1.96*tot_se, 3)
CI_tot_upper = np.round(tot_diff + 1.96*tot_se, 3)
```

```
lift_tot = (tot_diff / np.mean(df[(df['treatment'] == 0) & (df['saw_ads'] ==␣
↪1)]['sales'])) * 100

print('Point Estimate of the Mean Difference (TOT Estimate): ', np.
↪round(tot_diff, 3))
print('Standard error (TOT Estimate): ', np.round(tot_se, 3))
print('95% CI: [', CI_tot_lower, ', ', CI_tot_upper, ']')
print('Lift: {:.2f}%'.format(lift_tot))
```

```
Point Estimate of the Mean Difference (TOT Estimate):  0.67
Standard error (TOT Estimate):  13.94
95% CI: [ -26.653 ,  27.993 ]
Lift: 70.32%
```

- **Point Estimate of the Mean Difference (TOT Estimate):** The point estimate of the mean difference is 0.67. As this is higher than the ITT estimate, we can observe that on an average in the group of people who actually saw the ads, have higher sales compared to the group of people who saw the ghost ads.
- **Standard Error (TOT Estimate):** The standard error associated with the TOT estimate is 13.94. This value reflects the variability or uncertainty in the estimate of the mean difference.
- **95% Confidence Interval (TOT Estimate):** Since this interval includes both positive and negative values, it indicates that the observed difference is not statistically significant, and there is considerable uncertainty about the true effect of the ad campaign.
- **Lift:** A lift of 70.32% suggests that, on average, the treatment group experienced a 70.32% increase in sales compared to the control group.

4.3 5c.

Based on your analysis in Question 3, which of the two estimates should you report from this experiment? Why?

[ ]:
```
# Compute effect using ITT estimate
ITT_effect = diff * len(df[df['treatment'] == 1])

# Compute effect using TOT estimate
TOT_effect = tot_diff * df[(df['treatment'] == 1) & (df['saw_ads'] ==␣
↪1)]['saw_ads'].sum()

print("Effect using ITT estimate:", ITT_effect)
print("Effect using TOT estimate:", TOT_effect)
```

```
Effect using ITT estimate: 1194920.7492791698
Effect using TOT estimate: 1211645.6650882412
```

- Given the aim of our study, which is to measure the effectiveness of a display ad campaign emphasizing environmentally friendly aspects of Selene's production, both the ITT (Intention-to-Treat) and TOT (Treatment-on-the-Treated) estimates could be valuable. However, considering the nature of our campaign's message and its potential impact on consumer behavior, the TOT estimate may be particularly effective in this context.

Here's why:

- **Focus on Actual Exposure:** The TOT estimate captures the effect of the campaign on individuals who were actually exposed to the advertisement and potentially influenced by its message. Since our ad campaign highlights Selene's environmentally friendly practices, individuals who are exposed to the ad and resonate with its message are more likely to engage with Selene's products in a manner consistent with the campaign's objectives.
- **Relevance to Campaign Objectives:** Our campaign aims to highlight Selene's environmentally friendly production practices and encourage consumers to support these efforts. The TOT estimate provides a direct assessment of the impact of the campaign on individuals who actively engage with Selene's message, aligning closely with the campaign's objectives.
- **Accounting for Compliance:** In the context of our study, compliance with the treatment (exposure to the ad) is crucial for understanding its effectiveness. The TOT estimate explicitly accounts for compliance by focusing on individuals who were exposed to the campaign message, providing a more targeted assessment of the campaign's impact among those who received the intended treatment.
- **Practical Implications:** Understanding the effectiveness of the campaign among individuals who actively engage with its message can offer valuable insights for future marketing strategies and resource allocation. By focusing on the TOT estimate, we can better assess the real-world impact of the campaign on consumer behavior and decision-making related to Selene's products.

Overall, while both ITT and TOT estimates offer valuable insights, the TOT estimate may be particularly effective in assessing the impact of your environmentally focused ad campaign, as it provides a targeted assessment of the campaign's effectiveness among individuals who were exposed to and potentially influenced by its message.

4.4 5d.

Using your preferred estimator, summarize your results for a manager. What are the managerial and statistical implications of your results?

Managerial Implications:

- The positive point estimate and lift suggest that the ad campaign may have some positive impact on sales. However, since the confidence interval includes zero and the effect is not statistically significant, the manager should interpret the results cautiously.
- It's essential to consider other factors that may influence sales, such as seasonality, market conditions, and competitive factors, before making any decisions based solely on the ad campaign.
- Further analysis and monitoring may be necessary to assess the long-term effectiveness of the campaign and to identify any potential trends or patterns in customer behavior.
- The manager should also consider the cost-effectiveness of the ad campaign and compare it with other marketing strategies to determine the best allocation of resources.

5 Question 6.

(5 pts each) What would be your estimate for the effect of the campaign if you didn't have an experiment? (Hint: you wouldn't have control-group data in this case) ##

5.1 6a.

Compute the observational estimate.

[ ]:
```
T_sales = df.loc[(df['treatment'] == 1) & (df['saw_ads'] == 1), 'sales']
T_sales_not_see_ads = df.loc[(df['treatment'] == 1) & (df['saw_ads'] == 0),␣
↪'sales']

t_diff = np.mean(T_sales) - np.mean(T_sales_not_see_ads)
std_tot_t_sales = T_sales.std()
std_tot_t_sales_not_see_ads = T_sales_not_see_ads.std()
t_se = np.sqrt(std_tot_t_sales**2 + std_tot_t_sales_not_see_ads**2)

CI_t_lower = np.round(t_diff - 1.96*t_se, 3)
CI_t_upper = np.round(t_diff + 1.96*t_se, 3)

print('Point Estimate of the Mean Difference: ', np.round(t_diff, 3))
print('Standard error: ', np.round(t_se, 3))
print('95% CI: [', CI_t_lower, ', ', CI_t_upper, ']')
```

```
Point Estimate of the Mean Difference:  0.673
Standard error:  14.307
95% CI: [ -27.368 ,  28.715 ]
```

- In this case, the estimated mean difference is 0.673, suggesting that, on average, the people who saw the ads had higher sales compared to the people who did not see the ads.

A standard error of 14.307 indicates the degree of variability in the estimated mean difference. A larger standard error suggests greater uncertainty in the estimate.

Since the confidence interval includes both positive and negative values, it suggests that the true effect of the ad campaign could be either positive (indicating increased sales) or negative (indicating decreased sales), and the observed mean difference may not be statistically significant.

5.2 6b.

Suppose a manager had not run an experiment and only had the observational estimate. What would they get wrong?

- If a manager had not run an experiment and relied solely on observational estimates, they would likely encounter several limitations and potential inaccuracies in their analysis. Here are some key points illustrating what the manager might get wrong:

Causality vs. Correlation: Observational estimates may only show correlations between variables, without establishing causality. Without experimental controls, it's challenging to determine whether observed effects are truly due to the ad campaign or influenced by other confounding factors.

Selection Bias: Observational estimates may suffer from selection bias, where certain types of users are more likely to be exposed to the ads or respond to them. This bias can distort the estimated effects of the ad campaign, leading to inaccurate conclusions about its effectiveness.

Unobserved Variables: Observational estimates may not account for all relevant variables that could influence sales or revenue. Factors such as seasonality, market trends, competitor activities, or changes in consumer behavior may confound the analysis, making it difficult to isolate the true impact of the ad campaign.

Missing Counterfactual: Without experimental controls, observational estimates lack a clear counterfactual scenario (e.g., control group) for comparison. This absence makes it challenging to quantify the true incremental effect of the ad campaign, as there is no reference point to assess what would have happened in the absence of the ads.

Generalizability: Observational estimates may not generalize well to other contexts or populations. The observed effects of the ad campaign in one setting may not apply universally, limiting the manager's ability to draw actionable insights or make informed decisions.

6 Question 7.

(5 pts each) Consider location as a segmentation variable. In particular, your manager has asked you to examine whether ads are more effective on UK customers. ##

6.1 7a.

Using your preferred estimator from question 5(c), what is the average ad effect per UK customer?

[ ]:
```
df['IsUK'] = df['country'].apply(lambda x: 1 if x == 'UK' else 0)
effect = tot_diff * len(df[(df['treatment'] == 1) & (df['IsUK'] == 1)])
UK_effect = np.round(effect / len(df['IsUK'] == 1), 3)

print("Ad effect per UK Customer", UK_effect)
```

```
Ad effect per UK Customer 0.117
```

6.2 7b.

Using your preferred estimator from question 5(c), what is the average ad effect per customer outside of UK?

UK:

[ ]:
```
df['IsUK'] = df['country'].apply(lambda x: 1 if x == 'UK' else 0)
effect_1 = tot_diff * len(df[(df['treatment'] == 1) & (df['IsUK'] == 0)])
outside_UK_effect = np.round(effect_1 / len(df['IsUK'] == 0),3)

print("Ad effect per Customer outside of UK", outside_UK_effect)
```

```
Ad effect per Customer outside of UK 0.285
```

6.3 7c.

Summarize the managerial and statistical implications of your results for a manager who needs to decide how to allocate the ad budget across locations. How will you recommend allocating the budget?

US:

[ ]:
```python
df['IsUS'] = df['country'].apply(lambda x: 1 if x == 'US' else 0)
effect_2 = tot_diff * len(df[(df['treatment'] == 1) & (df['IsUS'] == 1)])
US_effect = np.round(effect_2 / len(df['IsUS'] == 1),3)

print("Ad effect per US Customer", US_effect)
```

```
Ad effect per US Customer 0.241
```

FR:

[ ]:
```python
df['IsFR'] = df['country'].apply(lambda x: 1 if x == 'FR' else 0)
effect_3 = tot_diff * len(df[(df['treatment'] == 1) & (df['IsFR'] == 1)])
FR_effect = np.round(effect_3 / len(df['IsFR'] == 1),3)

print("Ad effect per France Customer", FR_effect)
```

```
Ad effect per France Customer 0.044
```

Managerial Implications:

- **Focus on Customers Outside the UK:** Advertising seems to have a significantly higher impact on customers outside the UK, with an average effect of 0.285 compared to 0.117 for UK customers. This suggests a potentially higher return on investment (ROI) by allocating more budget to non-UK locations.
- **Prioritize US and Avoid France:** Among non-UK countries, the US shows the strongest ad effect (0.241) compared to France (0.044). This suggests focusing efforts on the US market for better customer acquisition through advertising.