Case #1

Author: Yanbing Chen 1009958752

1. Why didn't Uber launch Express Pool with simple A/B testing for 6 cities? Why not roll out Express to 50% of users in a given city and leave the remaining 50% with the existing product offering to see how Express fares?

Answer:

Uber opted for a synthetic control experiment instead of simple A/B testing for several reasons:

(1). Complexisty and Scale: A synthetic control experiment provides a more rigorous and controlled environment for testing. This method allows for comparison against control cities where the market conditions are held constant,thus isolating the impact of the Express Pool feature more effectively.

(2). Clean Data Collection: The synthetic control allows for cleaner data collection. By avoiding changes in the treatment cities for five weeks, Uber's data scientists could attribute changes in the market directly to the introduction of Express Pool. A/B testing within a single city might introduce confounding variables that could muddy the interpretation of the data. For instance, external factors like weather, local events, or economic changes might differently impact the behavior of users in the same city, making it harder to isolate the effect of the new service.

(3). Stabilization: Before the experiment in Boston, the Express Pool product had been available for three months, which allowed the markets to stabilize after introducing the new product. This stabilization period is important for ensuring that any observed effects are due to the treatment itself rather than initial fluctuations following a new product launch.

2. Why did Uber launch the switch-back experiment in Boston if they were already testing Express with a synthetic control lunch experiment in 12 cities?

Answer:

The switch-back experiment conducted in Boston was a targeted experiment to evalute the effects of longer wait time on the efficiency and economics of the rider service as well as on the customer experience.

While the synthetic control experiment compared the overall performance of the Express Pol with the control cities, the switch-back experiment in Boston was designed to fine-tune the wait time. It aimed to assess whether changing the wait time between two to five minitues would lead to more efficient matches, and consequently more profitbale pricing structures due to better utilization of seating capacity.

In addition, the swithc-back experiment was likely aimed at quantifying the economic benefits of longer wait times and understanding the trade-off between immediate cost savings and potential long-term impacts on customer satisfaction and market performance.

3. What is the difference between a double match rate of 2-minute wait time against a 5-minute wait time?

- Plot the histogram and interpret the result.
- Write one-sided hypothesis tests.
- Is the difference statistically significant at a 95% confidence level? If you are using parametric test, you need to validate their assumptions.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# read the data
df = pd.read_excel('/Users/violet/Desktop/Rotman/RSM8415 - Service Analytics/I
df.head()
```

Out[2]:

	city_id	period_start	wait_time	treat	commute	trips_pool	trips_express_pool	rider_cance
0	Boston	2018-02-19 07:00:00	2 mins	False	True	1415	3245	
1	Boston	2018-02-19 09:40:00	5 mins	True	False	1461	2363	
2	Boston	2018-02-19 12:20:00	2 mins	False	False	1362	2184	
3	Boston	2018-02-19 15:00:00	5 mins	True	True	1984	3584	
4	Boston	2018-02-19 17:40:00	2 mins	False	False	1371	2580	

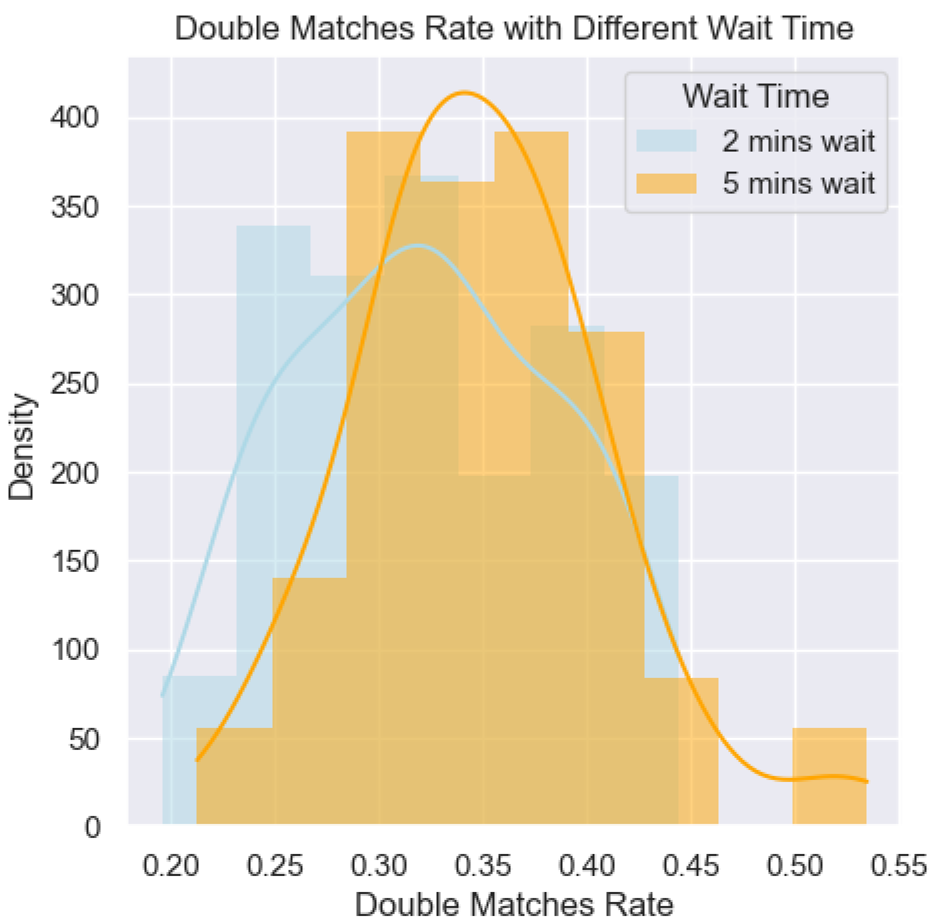
In [3]:
```
# derieve double match rate
df['total_ride'] = df['trips_express_pool'] + df['trips_pool'] + df['rider_canc
df['double_matches_rate'] = df['total_double_matches']/(df['trips_express_pool
```

Histogram and Interpretation

In [4]:
```
sns.set(rc={'figure.figsize':(5,5)})
sns.set(font_scale=1)

df_2_mins = df[df['wait_time'] == '2 mins']
df_5_mins = df[df['wait_time'] == '5 mins']

sns.histplot(data=df_2_mins, x='double_matches_rate', kde=True, stat='frequency
sns.histplot(data=df_5_mins, x='double_matches_rate', kde=True, stat='frequency

plt.xlabel('Double Matches Rate')
plt.ylabel('Density')
```

```
plt.legend(title='Wait Time')
plt.title('Double Matches Rate with Different Wait Time')
plt.show()
```

```
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed in a fut
ure version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a fut
ure version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed in a fut
ure version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a fut
ure version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



Interpretation:

The plot above displays to distribution of the double matches rate for two different wait times: 2 mins and 5 mins. There are some overlap between the two distributions, but not identical. The blue curve appears to right-skewed compared to the organce curve, suggesting that the rate of double matches might be lower on average for the 2-min wait time than for the 5-min wait time.

Besides, from the plot we can see that maybe waiting longer might be associated with a higher double matches rate, which could indicate more efficient use of capacity. This could be due to the algorithm having more time to find suitable matches for riders, thus potentially increasing the efficiency of the service.

One-sided hypothesis test

Null Hypothesis (H0): The mean double matches rate for 2 minutes wait time is less than, or equal to the mean double matches rate for 5 minutes wait time. $\mu_{5_{min}} \leq \mu_{2_{min}}$.

Alternative Hypothesis (H1): The mean double match rate for 2 minutes wait time is greater than the mean double match rate for 5 minutes wait time. $\mu_{5_{min}} > \mu_{2_{min}}$.

The assumptions of the t-test are:

- Normality: The data from 2 minute wait time, and 5 minute wait time for double matches comes from a normal distribution.
- Equal Variance: The variance for 2 minute wait time double matches is the same as the variance from 5 minute wait time double matches.

Validation for Normality

In [5]:
```
from scipy.stats import shapiro

wait_2_s = shapiro(df_2_mins['double_matches_rate'])
wait_5_s = shapiro(df_5_mins['double_matches_rate'])

print(f"Shapiro test for 2-minute wait time: {wait_2_s}")
print(f"Shapiro test for 5-minute wait time: {wait_5_s}")
```

```
Shapiro test for 2-minute wait time: ShapiroResult(statistic=0.97149181365966
8, pvalue=0.15116670727729797)
Shapiro test for 5-minute wait time: ShapiroResult(statistic=0.976454198360443
1, pvalue=0.2682681083679199)
```

- The Shapiro-Wilk test results for both the 2-minute and 5-minute wait time data show p-values of 0.1512 and 0.2683, respectively. Since both p-values are greater than the common alpha level of 0.05, there is not enough statistical evidence to reject the hypothesis that the data is normally distributed. This implies that, based on this test, both datasets could be considered as following a normal distribution, though this doesn't conclusively prove normality.

Validation for equal variance

In [8]:
```
from scipy.stats import f_oneway

statistic, p_value = f_oneway(df_2_mins['double_matches_rate'],df_5_mins['doub

print(f"Test result is: %.3f, P-value=%.3f" %(statistic,p_value))
```

```
Test result is: 6.163, P-value=0.014
```

- The T statistic is 6.613, and the p-value is approximately 0.014. Since the p-value is leass than the typical alpha level of 0.05, we do have sufficient evidence to reject the null hypothesis, which states that the variances are equal. Therefore, the result suggests that there's significant difference in variances between the two groups for the 'double_matches_rate' variable, indicating that the assumption of equal variances (homogeneity of variances) for these two datasets is not reasonable.

In [11]:
```
from scipy.stats import ttest_ind

one_side_ttest=ttest_ind(df_5_mins['double_matches_rate'],df_2_mins['double_ma
one_side_ttest
```

Out[11]:
```
Ttest_indResult(statistic=2.4824995426548577, pvalue=0.007193690166830475)
```

Answer: For 95% percent confidence interval, the typical alpha levelis 0.05. From the result of the one-side t-test above we can see the p-value is 0.00719, which is smaller than 0.05, so we can conclude that the difference between the two groups (2 mins & 5 mins) is statistically significant at the 95% confidence level. This means that there is sufficient evidence to suggest that the true mean of the population from which 2mins group is drawn is less than the true mean of the population from which 5mins group is drawn, in the direction of the alternative hypothesis.

4. How do the results of cancellation rate of 2-minute against a 5-minute wait time change during commute hours ? Use simple linear regression to answer the question. Use visualization to support your results.

In [30]:
```
import statsmodels.api as sm
```

In [31]:
```
# filter data in commute time
df1 = df[df['commute'] == True]

# derieve cancellation rate
df1['cancellation_rate'] = df1['rider_cancellations'] / (df1['total_ride'])
```

```
/var/folders/rp/ghfj6g1n259fz3zjrkpghf4w0000gn/T/ipykernel_81246/2228961639.p
y:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1['cancellation_rate'] = df1['rider_cancellations'] / (df1['total_ride'])
```

In [32]:
```
df1['treat'] = df['treat'].astype(int)

X = sm.add_constant(df1['treat'])

y = df1['cancellation_rate']

model = sm.OLS(y, X).fit()
```

```
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      cancellation_rate   R-squared:                       0.722
Model:                            OLS   Adj. R-squared:                  0.707
Method:                 Least Squares   F-statistic:                     46.77
Date:                Thu, 29 Feb 2024   Prob (F-statistic):           2.12e-06
Time:                        22:20:00   Log-Likelihood:                 80.094
No. Observations:                  20   AIC:                            -156.2
Df Residuals:                      18   BIC:                            -154.2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0463      0.001     31.512      0.000       0.043       0.049
treat          0.0142      0.002      6.839      0.000       0.010       0.019
==============================================================================
Omnibus:                        0.132   Durbin-Watson:                   2.102
Prob(Omnibus):                  0.936   Jarque-Bera (JB):                0.353
Skew:                          -0.037   Prob(JB):                        0.838
Kurtosis:                       2.354   Cond. No.                         2.62
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
```

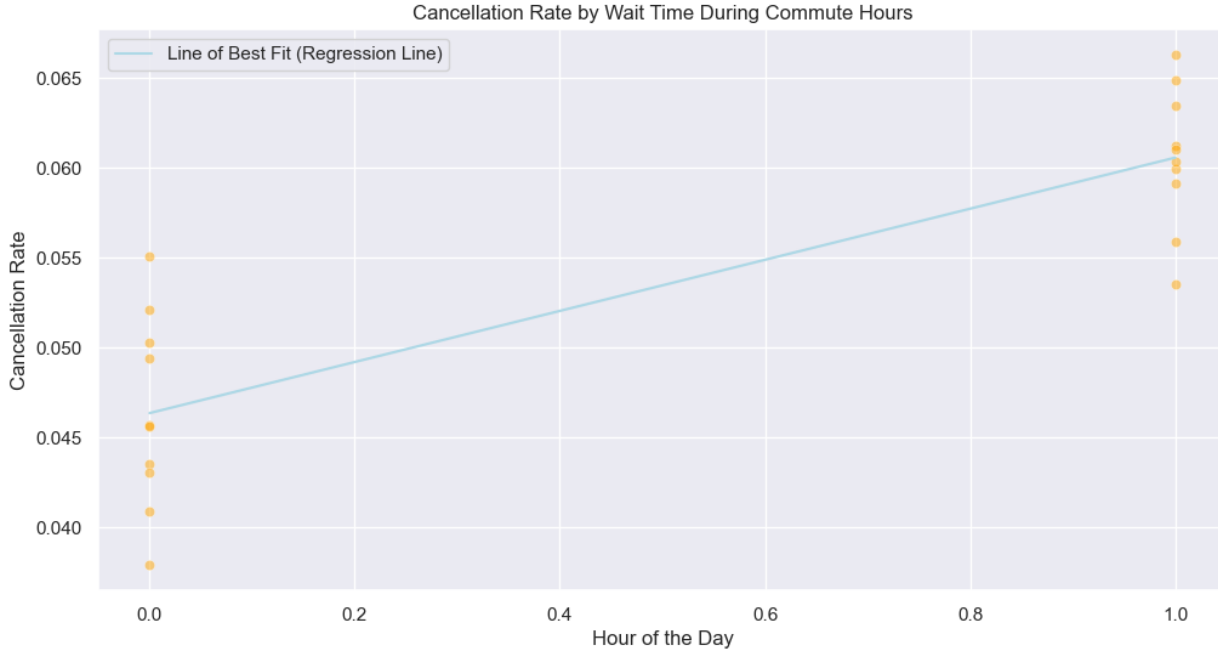
```
/var/folders/rp/ghfj6g1n259fz3zjrkpghf4w0000gn/T/ipykernel_81246/3506291587.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1['treat'] = df['treat'].astype(int)
```

In [33]:
```
y_pred = model.predict(X)
y_pred
```

Out[33]:
```
0      0.046333
3      0.060553
9      0.060553
12     0.046333
18     0.046333
21     0.060553
27     0.060553
30     0.046333
36     0.046333
39     0.060553
63     0.060553
66     0.046333
72     0.046333
75     0.060553
81     0.060553
84     0.046333
90     0.046333
93     0.060553
99     0.060553
102    0.046333
dtype: float64
```

In [41]:
```
plt.figure(figsize=(12, 6))

sns.scatterplot(data = df1,x = 'waiting',y = 'cancellation_rate',color = 'orang
sns.lineplot(x = df1['waiting'],y = y_pred,color = 'lightblue',label = 'Line o

plt.xlabel('Hour of the Day')
plt.ylabel('Cancellation Rate')
plt.title('Cancellation Rate by Wait Time During Commute Hours')
plt.legend()
plt.show()
```

```
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed in a fut
ure version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed in a fut
ure version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed in a fut
ure version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed in a fut
ure version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a fut
ure version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/violet/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a fut
ure version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



5. Based on the data available to you, what would you recommend doing? Would you increase the wait time from 2 to 5 minutes in the six treatment cities of the launch experiment? (explain in words)

Answer:

Based on the data available we have, I proposed these recommendation below before making the final decision regarding increasing the wait time.

- Efficiency Gains: The density plot indicates that a 5-minute wait may lead to a higher rate of double matches. If the goal is to optimize for efficiency, this data suggests that a longer wait time does improve the double match rate, which could lead to better utilization of drivers and vehicles.

- Customer Experience: The switch-back experiment in Boston showed that longer wait times led to more efficient matches and lower costs per ride. However, customer experience can be negatively affected by longer waits, potentially reducing customer satisfaction and loyalty. This trade-off needs to be carefully evaluated.

- Economic Impact: According to the case study, by not increasing wait times, Uber stands to lose a significant amount of money in the treatment cities. If the financial benefits are substantial and if Uber can mitigate the potential negative impact on customer experience, it might justify the increase in wait time.

- Data Consistency and Quality: The case study emphasizes the importance of clean data collection after a product launch. Changing the wait time during the experiment could confound the results, making it difficult to attribute changes in the market to specific causes.

Given these considerations, if the primary goal is to improve operational efficiency and if the negative impact on customer experience can be managed (perhaps through customer education or incentives), I would recommend cautiously increasing the wait time to 5 minutes. However, it would be important to continuously monitor customer feedback and market performance to ensure that the change does not lead to a significant loss of ridership. It's also crucial to communicate the reasons for the change to customers clearly, to manage their expectations effectively.