NYC TLC Project Part 3

November 5, 2024

1 NYC TLC Project Part 3

To analyze the relationship between fare amount and payment type and to conduct an A/B test.

2 Statistical analysis

The project covers fundamental concepts such as descriptive statistics and hypothesis testing.

The purpose of this project is to demostrate knowledge of how to prepare, create, and analyze A/B tests. The A/B test results should aim to find ways to generate more revenue for taxi cab drivers.

Note: For the purpose of this project, assume that the sample data comes from an experiment in which customers are randomly selected and divided into two groups: 1) customers who are required to pay with credit card, 2) customers who are required to pay with cash. Without this assumption, we cannot draw causal conclusions about how payment method affects fare amount.

The goal is to apply descriptive statistics and hypothesis testing in Python. The goal for this A/B test is to sample data and analyze whether there is a relationship between payment type and fare amount. For example: discover if customers who use credit cards pay higher fare amounts than customers who use cash.

This activity has four parts:

Part 1: Imports and data loading

Part 2: Conduct EDA and hypothesis testing

3 Conduct an A/B test

Import packages and libraries needed to compute descriptive statistics and conduct a hypothesis test.

Hint:

Before you begin, recall the following Python packages and functions that may be useful:

Main functions: stats.ttest_ind(a, b, equal_var)

Other functions: mean()

Packages: pandas, stats.scipy

[1]:
```
import pandas as pd
import numpy as np
from scipy import stats
```

[2]:
```
# Load dataset into dataframe
taxi_data = pd.read_csv("2017_Yellow_Taxi_Trip_Data.csv", index_col = 0)
```

3.0.1 Task 2. Data exploration

Use descriptive statistics to conduct Exploratory Data Analysis (EDA).

Note: In the dataset, `payment_type` is encoded in integers: * 1: Credit card * 2: Cash * 3: No charge * 4: Dispute * 5: Unknown

[3]:
```
print(taxi_data.describe())
print(taxi_data.shape)
print(taxi_data.info())
```

```
          VendorID  passenger_count  trip_distance    RatecodeID  \
count  22699.000000     22699.000000   22699.000000  22699.000000
mean       1.556236         1.642319       2.913313      1.043394
std        0.496838         1.285231       3.653171      0.708391
min        1.000000         0.000000       0.000000      1.000000
25%        1.000000         1.000000       0.990000      1.000000
50%        2.000000         1.000000       1.610000      1.000000
75%        2.000000         2.000000       3.060000      1.000000
max        2.000000         6.000000      33.960000     99.000000

       PULocationID  DOLocationID  payment_type   fare_amount         extra  \
count  22699.000000  22699.000000  22699.000000  22699.000000  22699.000000
mean     162.412353    161.527997      1.336887     13.026629      0.333275
std       66.633373     70.139691      0.496211     13.243791      0.463097
min        1.000000      1.000000      1.000000   -120.000000     -1.000000
25%      114.000000    112.000000      1.000000      6.500000      0.000000
50%      162.000000    162.000000      1.000000      9.500000      0.000000
75%      233.000000    233.000000      2.000000     14.500000      0.500000
max      265.000000    265.000000      4.000000    999.990000      4.500000

            mta_tax    tip_amount  tolls_amount  improvement_surcharge  \
count  22699.000000  22699.000000  22699.000000           22699.000000
mean       0.497445      1.835781      0.312542               0.299551
std        0.039465      2.800626      1.399212               0.015673
min       -0.500000      0.000000      0.000000              -0.300000
25%        0.500000      0.000000      0.000000               0.300000
50%        0.500000      1.350000      0.000000               0.300000
75%        0.500000      2.450000      0.000000               0.300000
```

```
max        0.500000    200.000000     19.100000                0.300000

       total_amount
count  22699.000000
mean      16.310502
std       16.097295
min     -120.300000
25%        8.750000
50%       11.800000
75%       17.800000
max     1200.290000
(22699, 17)
<class 'pandas.core.frame.DataFrame'>
Int64Index: 22699 entries, 24870114 to 17208911
Data columns (total 17 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   VendorID               22699 non-null  int64
 1   tpep_pickup_datetime   22699 non-null  object
 2   tpep_dropoff_datetime  22699 non-null  object
 3   passenger_count        22699 non-null  int64
 4   trip_distance          22699 non-null  float64
 5   RatecodeID             22699 non-null  int64
 6   store_and_fwd_flag     22699 non-null  object
 7   PULocationID           22699 non-null  int64
 8   DOLocationID           22699 non-null  int64
 9   payment_type           22699 non-null  int64
 10  fare_amount            22699 non-null  float64
 11  extra                  22699 non-null  float64
 12  mta_tax                22699 non-null  float64
 13  tip_amount             22699 non-null  float64
 14  tolls_amount           22699 non-null  float64
 15  improvement_surcharge  22699 non-null  float64
 16  total_amount           22699 non-null  float64
dtypes: float64(8), int64(6), object(3)
memory usage: 3.1+ MB
None
```

We are interested in the relationship between payment type and the fare amount the customer pays. One approach is to look at the average fare amount for each payment type.

[4]:
```
payment_grouped = taxi_data.groupby(["payment_type"])
mean_fare = payment_grouped["fare_amount"].mean()
print(mean_fare)
```

```
payment_type
1    13.429748
2    12.213546
```

```
3    12.186116
4     9.913043
Name: fare_amount, dtype: float64
```

Based on the averages shown, it appears that customers who pay in credit card tend to pay a larger fare amount than customers who pay in cash. However, this difference might arise from random sampling, rather than being a true difference in fare amount. To assess whether the difference is statistically significant, we conduct a hypothesis test.

3.0.2 Task 3. Hypothesis testing

H_0: There is no difference in the average fare amount between customers who use credit cards and customers who use cash.

H_A: There is a difference in the average fare amount between customers who use credit cards and customers who use cash.

Oour goal in this step is to conduct a two-sample t-test. The steps for conducting a hypothesis test are:

1. State the null hypothesis and the alternative hypothesis
2. Choose a signficance level
3. Find the p-value
4. Reject or fail to reject the null hypothesis

We choose 5% as the significance level and proceed with a two-sample t-test.

[5]:
```
significance_level = 0.05
credit_card = taxi_data[taxi_data['payment_type'] == 1]['fare_amount']
cash = taxi_data[taxi_data['payment_type'] == 2]['fare_amount']
stats.ttest_ind(a=credit_card, b=cash, equal_var=False)
```

[5]:
```
Ttest_indResult(statistic=6.866800855655372, pvalue=6.797387473030518e-12)
```

Since the p-value is significantly smaller than the significance level of 5%, you reject the null hypothesis.

We conclude that there is a statistically significant difference in the average fare amount between customers who use credit cards and customers who use cash.

3.0.3 Outcomes

1. The key business insight is that encouraging customers to pay with credit cards can generate more revenue for taxi cab drivers.
2. This project requires an assumption that passengers were forced to pay one way or the other, and that once informed of this requirement, they always complied with it. The data was not collected this way; so, an assumption had to be made to randomly group data entries to perform an A/B test. This dataset does not account for other likely explanations. For example, riders might not carry lots of cash, so it's easier to pay for longer/farther trips with

a credit card. In other words, it's far more likely that fare amount determines payment type, rather than vice versa.