Maximizing Revenue for Drivers

THROUGH MODE OF PAYMENT

The objective of this project is to analyse various payments methods accepted by the NYC Taxi drivers and provide an optimized solution to increase there revenue by payment type.

AGENDA:

- **Problem Statement**
- **Research Question**
- **Data Overview**
- **Methodology**
- **Analysis and Findings**
- **Hypothesis Testing**
- **Recommendations**

DATASET:

https://drive.google.com/file/d/12mIgNKzAirbSOWUBqEg3SudVUSU3twtP/view?usp=drive_link

Problem Statement:

In the fast-paced taxi booking sector, making the most of revenue is essential for long-term success and driver happiness. Our goal is to use data-driven insights to maximise revenue streams for taxi drivers in order to meet this need. Our research aims to determine whether payment methods have an impact on fare pricing by focusing on the relationship between payment type and fare amount.

Research Question/ Hypothesis:

Is there a relationship between total fare amount and payment type?

Can we nudge customers toward payment methods that generate higher revenue for drivers, without negatively impacting customer experience?

Data Overview:

For this analysis, we utilized the comprehensive dataset of NYC Taxi Trip Record, performed data cleaning and feature engineering procedures to concentrate solely on the relevant columns essential for our investigation.

```
data
```

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID
0	1.0	2020-01-01 00:28:15	2020-01-01 00:33:03	1.0	1.20	1.0	N	238	239
1	1.0	2020-01-01 00:35:39	2020-01-01 00:43:04	1.0	1.20	1.0	N	239	238
2	1.0	2020-01-01 00:47:41	2020-01-01 00:53:52	1.0	0.60	1.0	N	238	238
3	1.0	2020-01-01 00:55:23	2020-01-01 01:00:14	1.0	0.80	1.0	N	238	151
4	2.0	2020-01-01 00:01:58	2020-01-01 00:04:16	1.0	0.00	1.0	N	193	193
...	...	...	...	...	...	...	...	...	...
6405003	NaN	2020-01-31 22:51:00	2020-01-31 23:22:00	NaN	3.24	NaN	NaN	237	234
6405004	NaN	2020-01-31 22:10:00	2020-01-31 23:26:00	NaN	22.13	NaN	NaN	259	45
6405005	NaN	2020-01-31 22:50:07	2020-01-31 23:17:57	NaN	10.51	NaN	NaN	137	169
6405006	NaN	2020-01-31 22:25:53	2020-01-31 22:48:32	NaN	5.49	NaN	NaN	50	42
6405007	NaN	2020-01-31 22:44:00	2020-01-31 23:06:00	NaN	11.60	NaN	NaN	179	205

6405008 rows × 18 columns

Methodology:

❖ Descriptive Analysis:

Performed statistical analysis to summarize key aspects of the data, focusing on fare amounts and payment types.

❖ Hypothesis Testing:

Conducted a T-Test to evaluate the relationship between payment type and fare amount, testing the hypothesis that different payment methods influence fare amount.

- **Performing Exploratory Data Analysis:**

```
df.shape
```

```
(6405008, 18)
```

```
df.describe()
```

	VendorID	passenger_count	trip_distance	RatecodeID	PULocationID	DOLocationID	payment_type	fare_amount	extra	mta_tax
count	6.339567e+06	6.339567e+06	6.405008e+06	6.339567e+06	6.405008e+06	6.405008e+06	6.339567e+06	6.405008e+06	6.405008e+06	6.405008e+06
mean	1.669624e+00	1.515333e+00	2.929644e+00	1.059908e+00	1.647323e+02	1.626627e+02	1.270298e+00	1.269411e+01	1.115456e+00	4.923182e-01
std	4.703484e-01	1.151594e+00	8.315911e+01	8.118432e-01	6.554374e+01	6.991261e+01	4.739985e-01	1.212730e+01	1.260054e+00	7.374184e-02
min	1.000000e+00	0.000000e+00	-3.062000e+01	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	-1.238000e+03	-2.700000e+01	-5.000000e-01
25%	1.000000e+00	1.000000e+00	9.600000e-01	1.000000e+00	1.320000e+02	1.130000e+02	1.000000e+00	6.500000e+00	0.000000e+00	5.000000e-01
50%	2.000000e+00	1.000000e+00	1.600000e+00	1.000000e+00	1.620000e+02	1.620000e+02	1.000000e+00	9.000000e+00	5.000000e-01	5.000000e-01
75%	2.000000e+00	2.000000e+00	2.930000e+00	1.000000e+00	2.340000e+02	2.340000e+02	2.000000e+00	1.400000e+01	2.500000e+00	5.000000e-01
max	2.000000e+00	9.000000e+00	2.102401e+05	9.900000e+01	2.650000e+02	2.650000e+02	5.000000e+00	4.265000e+03	1.130100e+02	3.080000e+01

Features impacting Fare Amount:

- ❖ Trip distance
- ❖ Trip Duration
- ❖ Pickup and Drop Location

- **Checking the Data Type of the Features:**

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6405008 entries, 0 to 6405007
Data columns (total 18 columns):
 #   Column                 Dtype  
---  ------                 -----  
 0   VendorID               float64
 1   tpep_pickup_datetime   object 
 2   tpep_dropoff_datetime  object 
 3   passenger_count        float64
 4   trip_distance          float64
 5   RatecodeID             float64
 6   store_and_fwd_flag     object 
 7   PULocationID           int64  
 8   DOLocationID           int64  
 9   payment_type           float64
 10  fare_amount            float64
 11  extra                  float64
 12  mta_tax                float64
 13  tip_amount             float64
 14  tolls_amount           float64
 15  improvement_surcharge  float64
 16  total_amount           float64
 17  congestion_surcharge   float64
dtypes: float64(13), int64(2), object(3)
memory usage: 879.6+ MB
```

- **Data Cleaning – Handling the missing values & Filtering the Data:**

Filtering the Data --- Using only required Features

```
df = df[['passenger_count','payment_type','fare_amount','trip_distance','duration']]
```

Handling Missing Values

```
df.isnull().sum()
```

```
passenger_count    65441
payment_type       65441
fare_amount            0
trip_distance          0
duration               0
dtype: int64
```

```python
print('Percentage of Missing Values in the Dataset is ',(65441/len(df))*100)
```

```
Percentage of Missing Values in the Dataset is  1.021716132126611
```

```python
df.dropna(inplace = True)
df
```

	passenger_count	payment_type	fare_amount	trip_distance	duration
0	1.0	1.0	6.0	1.20	4.800000
1	1.0	1.0	7.0	1.20	7.416667
2	1.0	1.0	6.0	0.60	6.183333
3	1.0	1.0	5.5	0.80	4.850000
4	1.0	2.0	3.5	0.00	2.300000
...	...	...	...	...	...
6339562	1.0	1.0	11.0	2.10	14.233333
6339563	1.0	1.0	13.0	2.13	19.000000
6339564	1.0	1.0	12.5	2.55	16.283333
6339565	1.0	2.0	8.5	1.61	9.633333
6339566	1.0	1.0	0.0	0.00	1.066667

6339567 rows × 5 columns

- Since the missing values contributing only 1.02% of the data, they were dropped and the cleaned data is used for further analysis.
- Similarly, the duplicates were also showing no impact on the data, they were also dropped. This makes data more efficient and lighter which is good for analysis.

```python
df = df[df['payment_type']<3]
df = df[(df['passenger_count']>0)& (df['passenger_count']<6)]
df.shape
```

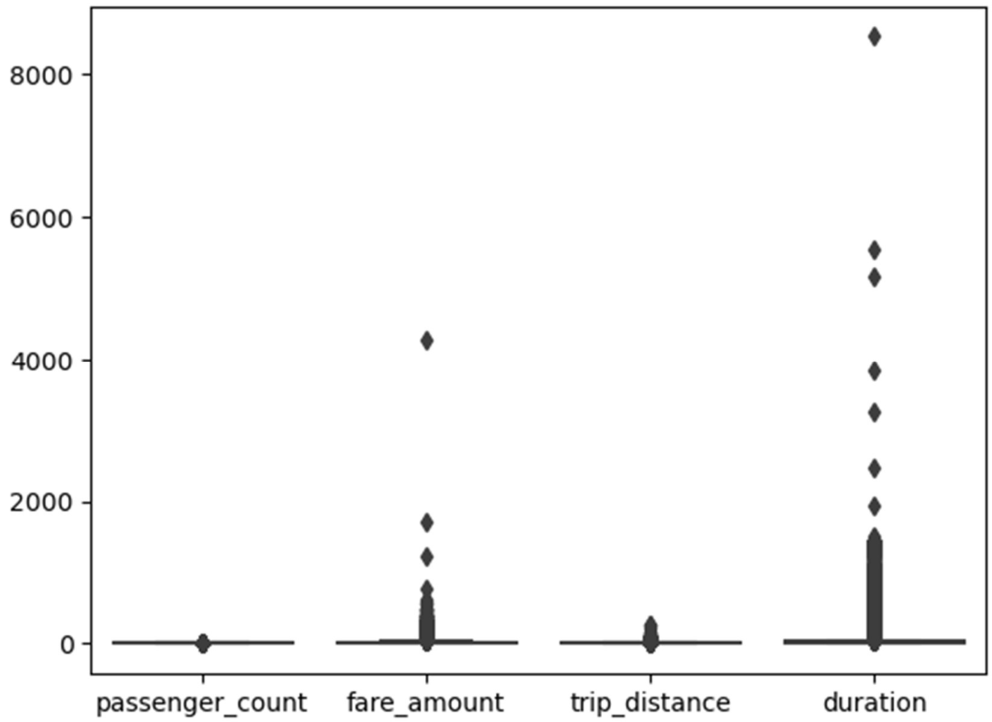
```
(2780283, 5)
```

```python
df['payment_type'].replace([1,2],['Online','Cash'], inplace = True)
```

```python
df.head()
```

	passenger_count	payment_type	fare_amount	trip_distance	duration
0	1	Online	6.0	1.2	4.800000
1	1	Online	7.0	1.2	7.416667
2	1	Online	6.0	0.6	6.183333
3	1	Online	5.5	0.8	4.850000
4	1	Cash	3.5	0.0	2.300000

- **Outlier Detection and Handling the Outliers:**

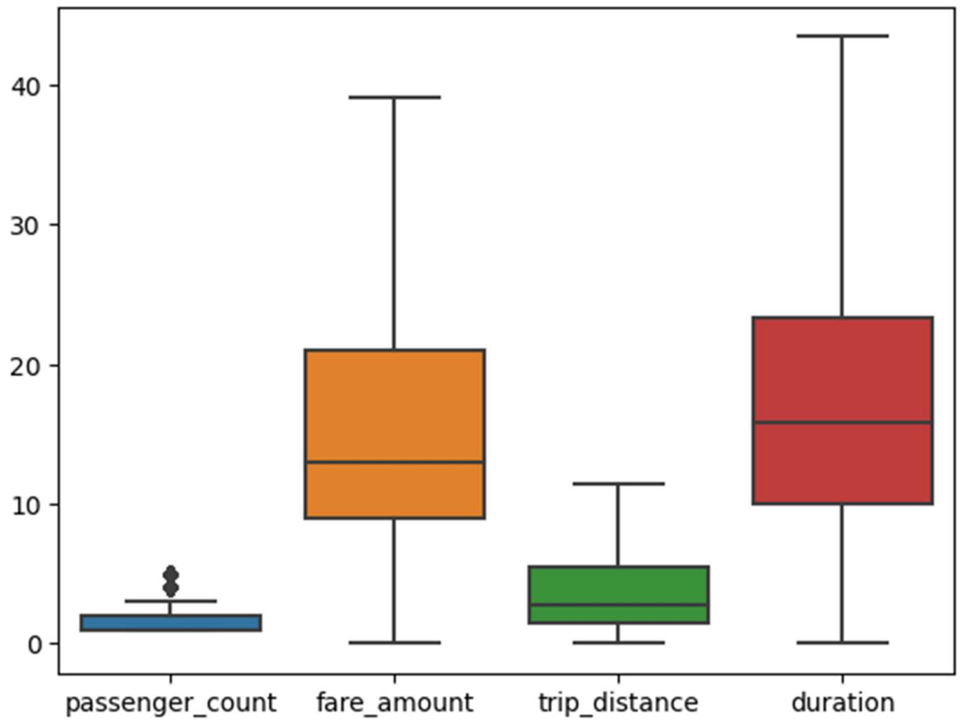


❖ Handling the outliers play a major role in data analysis, as the statistical values such as mean which is the pillar behind the standard deviation is sensitive to outliers and gives wrong interpretations.

```
for col in ['fare_amount','trip_distance','duration']:
    q1 = df[col].quantile(0.25)
    q3 = df[col].quantile(0.75)
    iqr = q3 - q1

    ll = q1-1.5*iqr
    ul = q3+1.5*iqr

    df[col].clip(lower=ll,upper=ul,inplace=True)
```



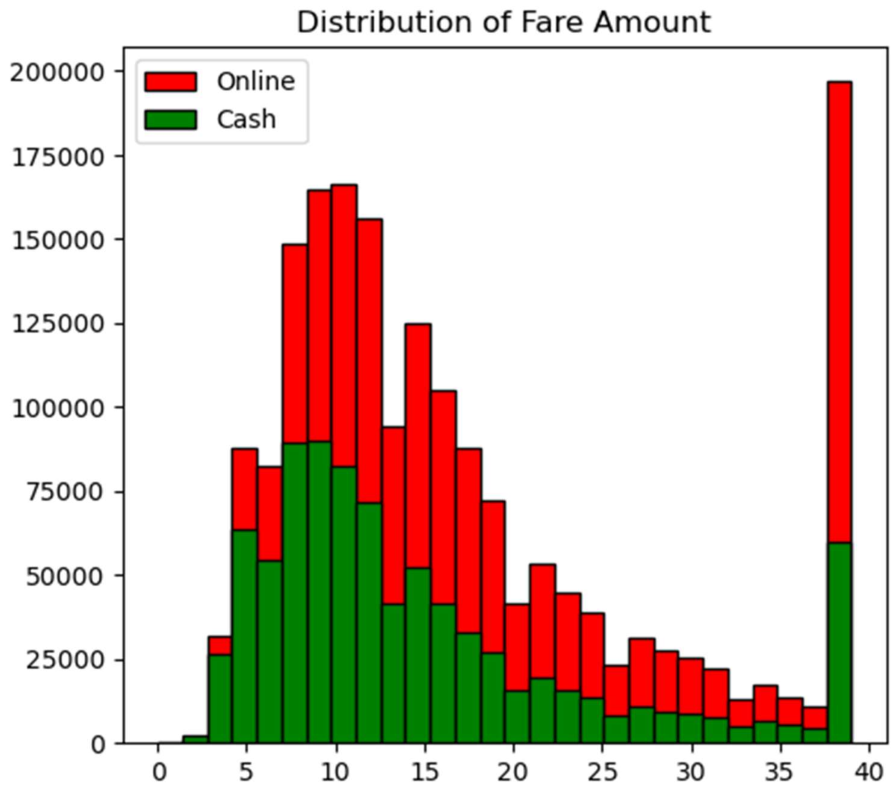
Since the outliers were gone, we can further proceed with our analysis.

Insights From The Data:

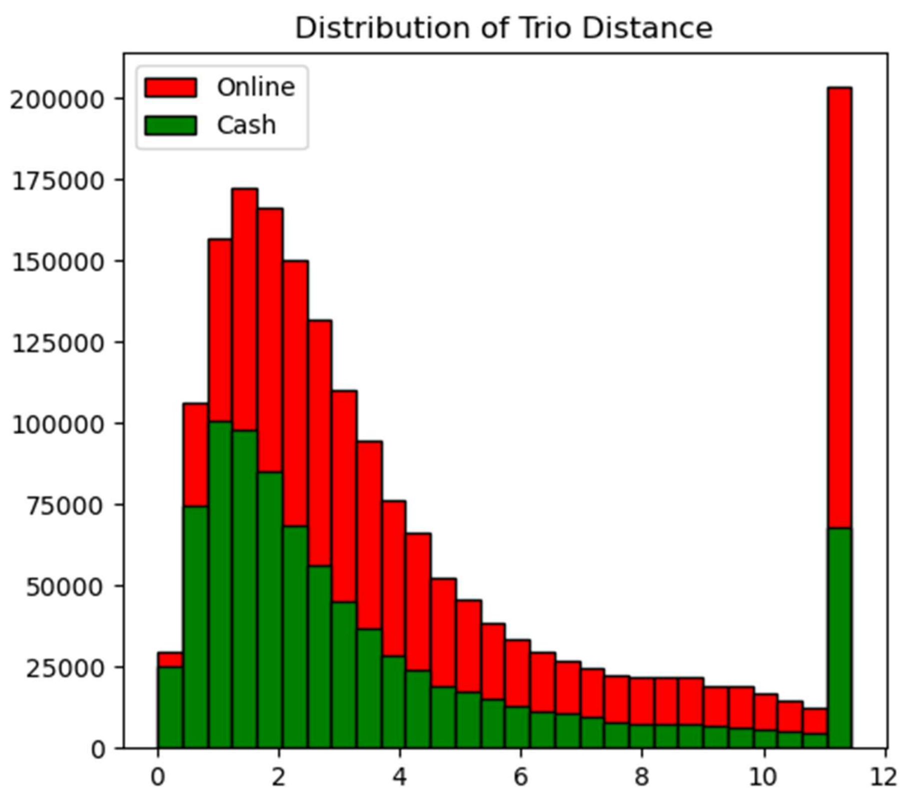
```
: plt.figure(figsize=(12,5))
  plt.subplot(1,2,1)
  plt.title('Distribution of Fare Amount')
  plt.hist(df[df['payment_type'] == 'Online']['fare_amount'], histtype = 'barstacked', bins = 28, edgecolor = 'k', color = 'red', l
  plt.hist(df[df['payment_type'] == 'Cash']['fare_amount'], histtype = 'barstacked',bins = 28, edgecolor = 'k', color = 'green',lab
  plt.legend()

  plt.figure(figsize=(12,5))
  plt.subplot(1,2,1)
  plt.title('Distribution of Trio Distance')
  plt.hist(df[df['payment_type'] == 'Online']['trip_distance'], histtype = 'barstacked', bins = 28, edgecolor = 'k', color = 'red',
  plt.hist(df[df['payment_type'] == 'Cash']['trip_distance'], histtype = 'barstacked',bins = 28, edgecolor = 'k', color = 'green',l
  plt.legend()
  plt.show()
```

	Payment Type	Mean	Standard Deviation
Fare Amount	**Online**	**17.09**	**10.32**
	Cash	**14.76**	**9.61**
Trip Distance	**Online**	**4.26**	**3.43**
	Cash	**3.58**	**3.22**

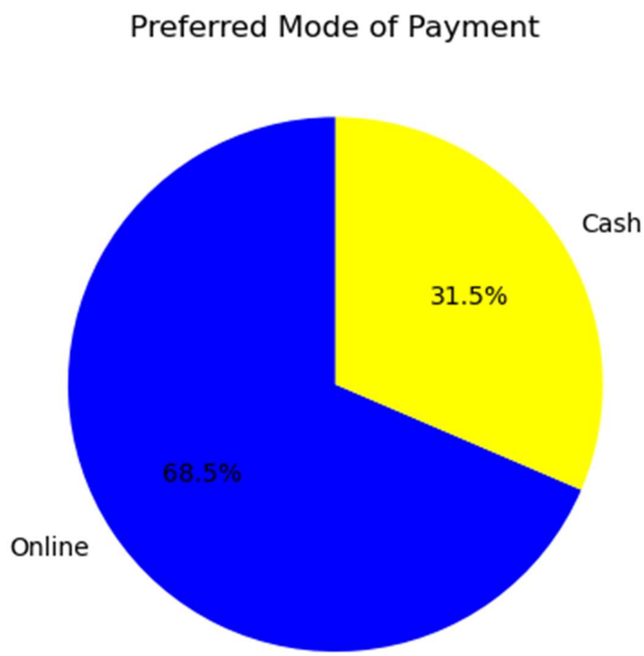


Distribution of Fare Amount
Online
Cash
200000
175000
150000
125000
100000
75000
50000
25000
0
0
5
10
15
20
25
30
35
40



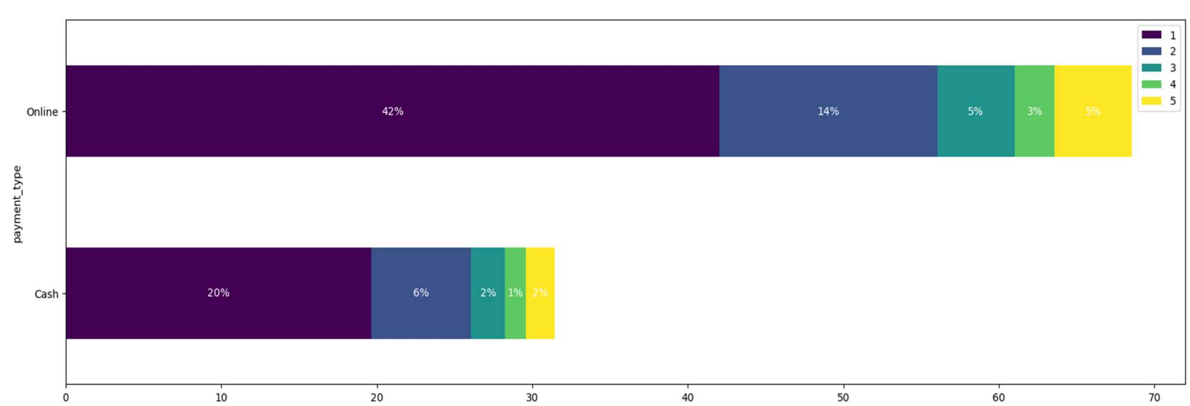
Distribution of Trio Distance
Online
Cash
200000
175000
150000
125000
100000
75000
50000
25000
0
0
2
4
6
8
10
12

- ❖ Customers who pay through online payment tend to have a slightly higher average trip distance and fare amount compares to those paying with cash.
- ❖ Customers prefers to pay more with online when they have high fare amount and long trip distance.



- ❖ 67.3% of the passenger prefer online mode for payments and 32.7% of the passengers pay by cash.
- ❖ Most of the customers are preferring online payments over cash transactions.

- **Distribution of Payment:**

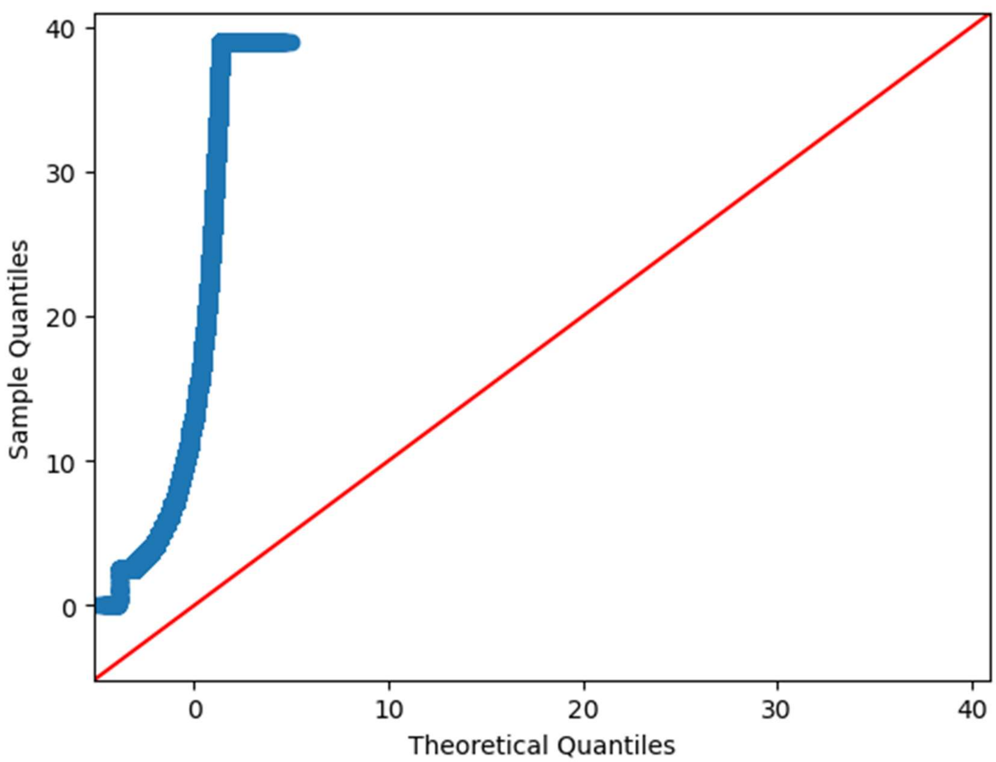


- ❖ Among card payments, rides with single passenger dominates with largest portion comprising of 42% of all online transactions.
- ❖ Similarly, cash transactions are also predominantly dominated by single -ride passengers making up to 20% of all cash payments.
- ❖ There is a noticeable decrease in the percentage of transactions as the passenger count increases, suggesting that as the no. of passengers/ group increases the usage of taxis are less or opting for alternative payment methods.
- ❖ These insights emphasize the importance of considering both payment method and passenger count when analysing transaction data, as they provide valuable insights into customer behaviour and preferences.

Hypothesis Testing:

- ➢ **Null Hypothesis (Ho): -** There is no significant difference in average fare between customers who use online payments and cash.

- ➢ **Alternative Hypothesis(H1): -** There is significant difference in average fare between customers who use online payments and cash.



Hence, the data is not normal and standard deviation is unknown we use T-Test.

```
online_sample = df[df['payment_type']== 'Online']['fare_amount']
cash_sample = df[df['payment_type']== 'Cash']['fare_amount']
```

```
t_stats, p_value = st.ttest_ind(a = online_sample,
                               b = cash_sample,
                               equal_var = False)
level_of_significance = 0.05 # error

if p_value<= level_of_significance:
    print('We reject Ho')
    print('Hence, it is proved. There is significant difference in average fare between customers who use online payments and cas
else:
    print('We do not reject Ho')

print('T statistic', t_stats)
print('P-value',p_value)
```

```
We reject Ho
Hence, it is proved. There is significant difference in average fare between customers who use online payments and cash.
T statistic 182.0837116780057
P-value 0.0
```

- **Since, the 'p_value' is less than level of significance (0.05) we rejected the Null Hypothesis. Hence, there is a significance difference in fare amount between online payment and cash.**

Recommendations:

- Encourage customers to use online payments to capitalize on the potential for generating more revenue for taxi cab drivers.
- Implement strategies such coupons or discounts on transaction or other benefits to increase more customer base and choose online payment methods.
- Provide seamless and secure gateways for online payments to ensure customer safety and convenience.