Uber Trip Analysis

```
import pandas as pd
import numpy as np
import plotly.express as px

df=pd.read_csv("D:\\Excel_project\\UberDataset.csv")
print("csv loaded succussecfully",df)
```

```
csv loaded succussecfully             START_DATE           END_DATE
CATEGORY             START  \
0     01-01-2016 21:11  01-01-2016 21:17  Business       Fort Pierce

1     01-02-2016 01:25  01-02-2016 01:37  Business       Fort Pierce

2     01-02-2016 20:25  01-02-2016 20:38  Business       Fort Pierce

3     01-05-2016 17:31  01-05-2016 17:45  Business       Fort Pierce

4     01-06-2016 14:42  01-06-2016 15:49  Business       Fort Pierce

...                ...               ...       ...               ...

1151  12/31/2016 13:24  12/31/2016 13:42  Business           Kar?chi

1152  12/31/2016 15:03  12/31/2016 15:38  Business  Unknown Location

1153  12/31/2016 21:32  12/31/2016 21:50  Business        Katunayake

1154  12/31/2016 22:08  12/31/2016 23:51  Business           Gampaha

1155            Totals               NaN       NaN               NaN


                   STOP    MILES          PURPOSE
0          Fort Pierce      5.1   Meal/Entertain
1          Fort Pierce      5.0              NaN
2          Fort Pierce      4.8  Errand/Supplies
3          Fort Pierce      4.7          Meeting
4      West Palm Beach     63.7   Customer Visit
...                ...      ...              ...
1151  Unknown Location      3.9   Temporary Site
1152  Unknown Location     16.2          Meeting
1153           Gampaha      6.4   Temporary Site
1154         Ilukwatta     48.2   Temporary Site
1155               NaN  12204.7              NaN

[1156 rows x 7 columns]
```

```
#DATA CLEANING
df.describe()
```

```
              MILES
count   1156.000000
mean      21.115398
std      359.299007
min        0.500000
25%        2.900000
50%        6.000000
75%       10.400000
max    12204.700000
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1156 entries, 0 to 1155
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   START_DATE  1156 non-null   object 
 1   END_DATE    1155 non-null   object 
 2   CATEGORY    1155 non-null   object 
 3   START       1155 non-null   object 
 4   STOP        1155 non-null   object 
 5   MILES       1156 non-null   float64
 6   PURPOSE     653 non-null    object 
dtypes: float64(1), object(6)
memory usage: 63.3+ KB
```

```
df.isna().sum()
```

```
START_DATE      0
END_DATE        1
CATEGORY        1
START           1
STOP            1
MILES           0
PURPOSE       503
dtype: int64
```

```
df['PURPOSE']=df['PURPOSE'].fillna("Unknown")
df.dropna(inplace=True)
df.isna().sum()
```

```
START_DATE    0
END_DATE      0
CATEGORY      0
START         0
STOP          0
MILES         0
PURPOSE       0
dtype: int64
```

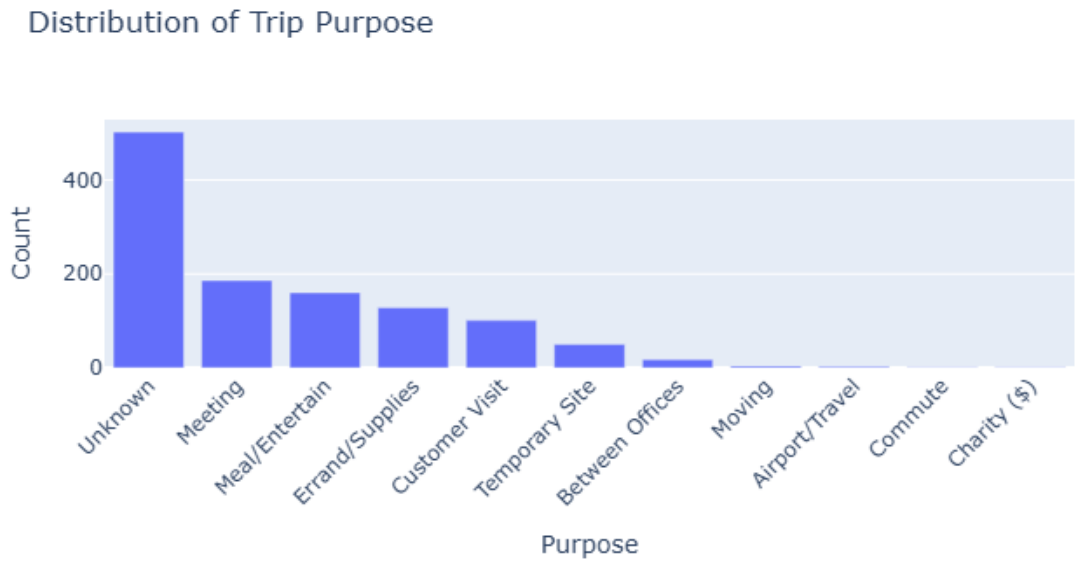
```
df.duplicated().sum()
```

```
np.int64(1)
```

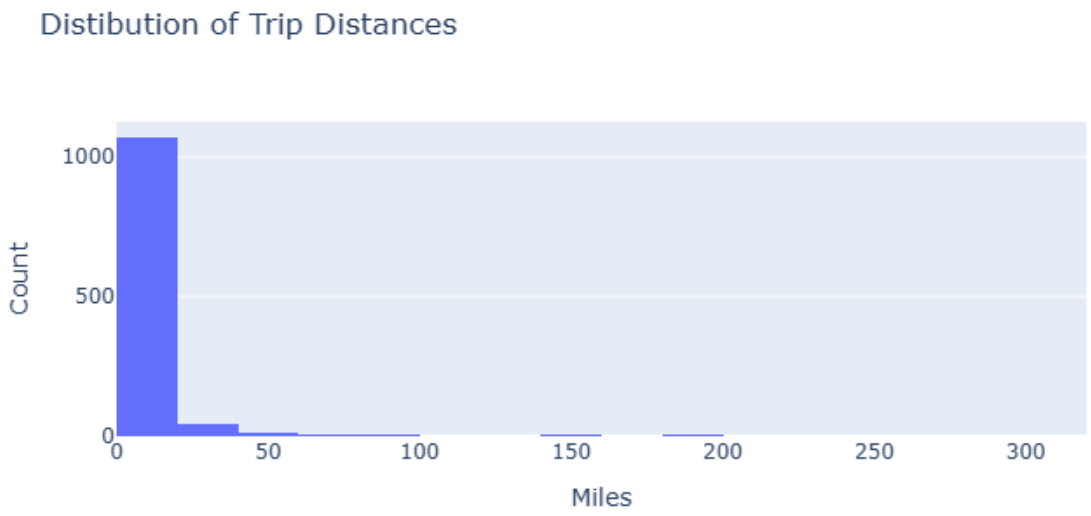
```
df.drop_duplicates(inplace=True)
df.duplicated().sum()
```

```
np.int64(0)
```

```
#EDA
#Visualize trip purposes
purpose_count=df['PURPOSE'].value_counts()
fig=px.bar(x=purpose_count.index,
y=purpose_count.values,labels={'x':"Purpose","y":'Count'},title='Distr
ibution of Trip Purpose')
fig.update_layout(xaxis_tickangle=-45)
fig.show()
```




```
#Visualize trip distances
fig=px.histogram(df,x='MILES',nbins=20,title='Distibution of Trip
Distances')
fig.update_xaxes(title='Miles')
fig.update_yaxes(title='Count')
fig.show()
```




```
#Feature Engineering
from dateutil.parser import parse
df['START_DATE']=df['START_DATE'].apply(lambda x:parse(x))
df['start_day']=df['START_DATE'].dt.strftime('%A')
df['start_day']
```

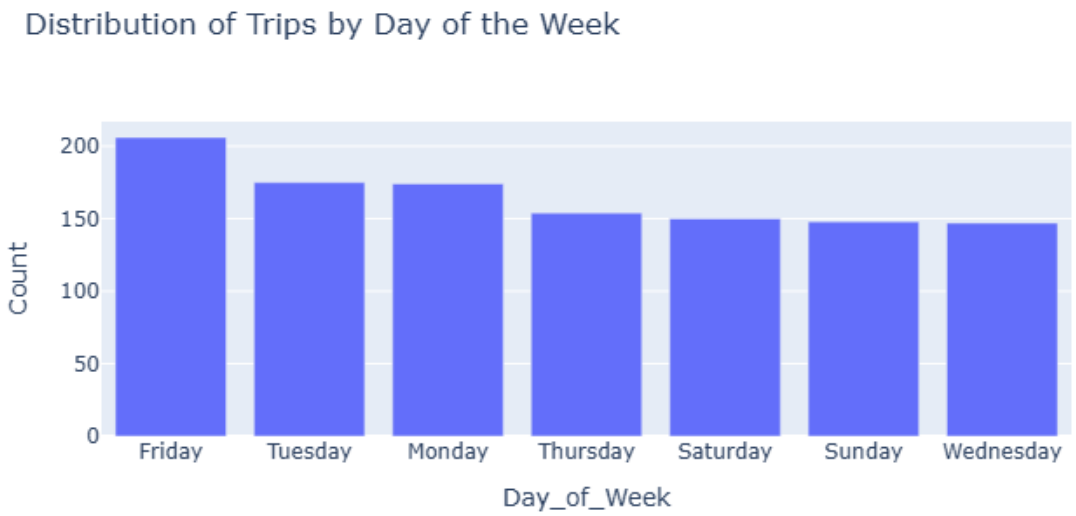
```
0          Friday
1        Saturday
2        Saturday
3         Tuesday
4       Wednesday
           ...
1150     Saturday
1151     Saturday
1152     Saturday
1153     Saturday
1154     Saturday
Name: start_day, Length: 1154, dtype: object
```

```
#plottin the number of trips per each day
day_counts = df['start_day'].value_counts().reset_index()

day_counts.columns = ['Day_of_Week', 'Count']

colors = px.colors.qualitative.Plotly[:7]

fig = px.bar(day_counts, x='Day_of_Week', y='Count',
             color_discrete_sequence=colors,
             labels={'x': 'Day of the Week', 'y': 'Number of Trips'},
             title='Distribution of Trips by Day of the Week')
fig.show()
```


Distribution of Trips by Day of the Week
Count
200
150
100
50
0
Friday
Tuesday
Monday
Thursday
Saturday
Sunday
Wednesday
Day_of_Week