Uber Ride Project Report

This project analyses the Uber dataset using Exploratory Data Analysis (EDA) to uncover key insights into ride patterns, user behaviour, and overall platform performance.

Prepared By: -	**Submitted to: -**
Vishal Singh	Mr. Anshum Banga
Skill Circle (WDLK01)	Skill Circle
05-01-2025	Chandigarh

UBER RIDES – EXPLORATORY DATA ANALYSIS

Project Overview

This report provides an analysis of Uber ride data, focusing on key patterns in trip details, such as ride frequency, peak hours, and geographic trends. By exploring factors like ride duration, distance, and customer preferences, the analysis offers insights into Uber's operations and helps identify opportunities for improving service efficiency and user experience.

About Company

Uber Technologies, Inc., commonly known as Uber, is a global leader in the ride-hailing and transportation industry. Founded in 2009 by Garrett Camp and Travis Kalanick, Uber has transformed urban mobility through its innovative, technology-driven platform. By connecting riders with drivers via a user-friendly mobile application, the company has revolutionized how people commute, making transportation more convenient, affordable, and accessible. Headquartered in San Francisco, California, Uber operates in over 70 countries and 10,000 cities worldwide, offering services such as ride-sharing, food delivery (Uber Eats), freight transportation, and electric bike and scooter rentals.

Purpose and Goals

Purpose: To create opportunities through movement by connecting people to reliable, efficient, and safe transportation solutions.

Goal: To make transportation as reliable as running water while supporting sustainability and reducing reliance on private vehicle ownership.

Dataset Used

The Uber ride dataset includes trip-level details such as request times, locations, durations, distances, and fares. It also incorporates factors like driver availability, surge pricing, weather, and traffic patterns. This data helps optimize algorithms, improve route efficiency, and predict demand for a seamless user experience.

UBER RIDES – EXPLORATORY DATA ANALYSIS

Dataset Overview

Source of Dataset: - The Uber Ride dataset was sourced from Kaggle

- File Size: Approximately 0.08 MB.
- Number of Rows: 1,156 entries.
- Number of Columns: 7 columns.
- Column Names: START_DATE, END_DATE, CATEGORY, START, STOP, MILES, PURPOSE.

➢ Project Plan for Uber Ride Analysis

1. **Data Collection and Access**:
   - Obtain Uber ride datasets from sources like Kaggle or public repositories.
   - Ensure dataset includes trip details, fares, distances, and ride categories.
   - Supplement with external data (e.g., weather, traffic) to fill data gaps.

2. **Data Pre-processing and Cleaning**:
   - Handle missing values, standardize formats, and encode categorical variables.
   - Remove duplicates and outliers to ensure data quality and reliability.
   - Conduct exploratory checks for data distribution, anomalies, and inconsistencies.

3. **Data Exploration and Visualization**:
   - Analyse ride trends, durations, and fares using univariate and bivariate analysis.
   - Create visualizations like histograms, scatter plots, line charts, and heatmaps.
   - Explore user behaviour, ride preferences, and external factors influencing demand.
   - Design interactive dashboards in Power BI or Tableau for deeper trend insights.

UBER RIDES – EXPLORATORY DATA ANALYSIS

Here we will use Python and its different libraries to analyze the Uber Rides Data.

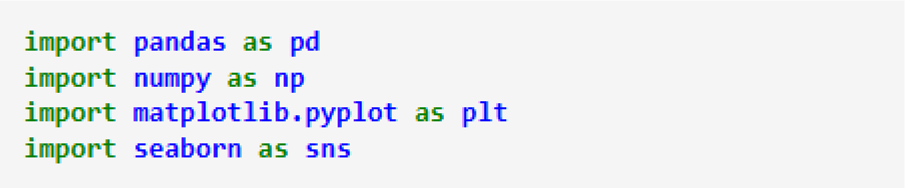
➢ Analysis Steps

1. Importing Libraries

The analysis will be done using the following libraries:

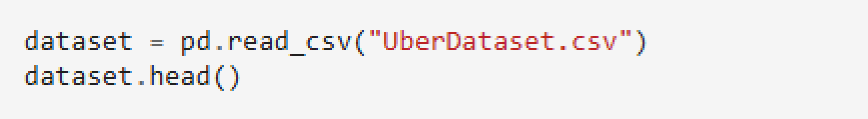
- **Pandas**: This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.
- **NumPy**: NumPy arrays are very fast and can perform large computations in a very short time.
- **Matplotlib / Seaborn**: This library is used to draw visualizations.

2. To importing all these libraries, we can use the below code:



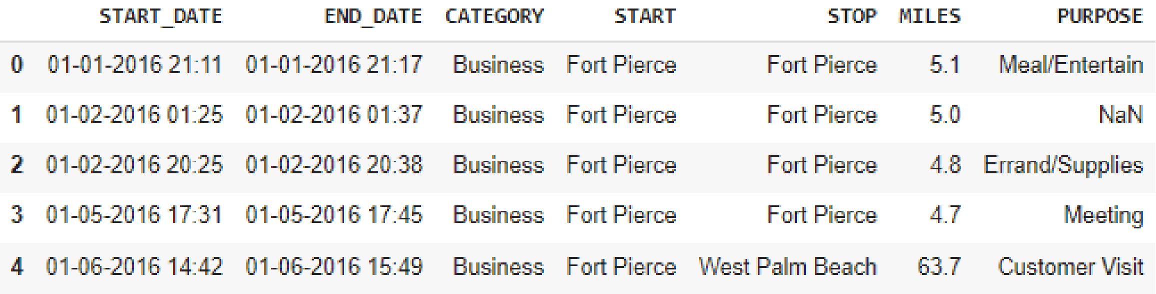
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

3. Once downloaded, you can import the dataset using the panda's library.



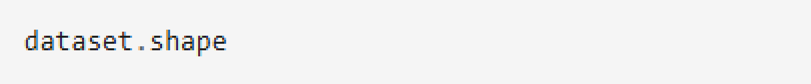
```
dataset = pd.read_csv("UberDataset.csv")
dataset.head()
```

Output:



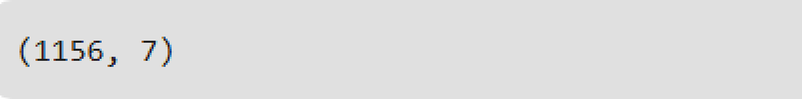
	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE
0	01-01-2016 21:11	01-01-2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	01-02-2016 01:25	01-02-2016 01:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	01-02-2016 20:25	01-02-2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	01-05-2016 17:31	01-05-2016 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	01-06-2016 14:42	01-06-2016 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit

4. To find the **shape** of the dataset, we can use dataset. **Shape**



```
dataset.shape
```

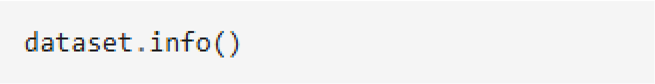
Output:



```
(1156, 7)
```

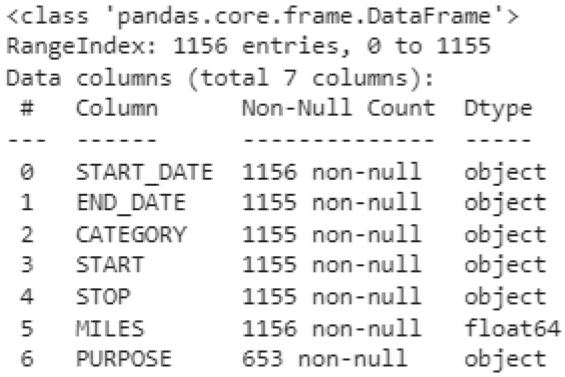
Vishal Singh

5. To understand the data more deeply, we need to know about the null values **count, datatype**, etc. So, for that we will use the below code.



```
dataset.info()
```

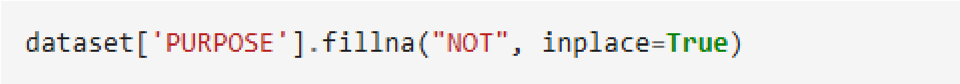
Output:



```
<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
```

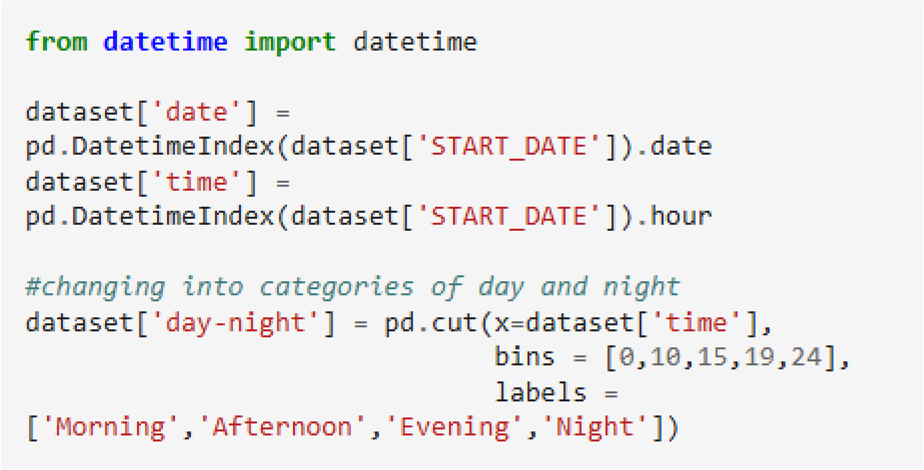
6. Data Pre-processing

As we understood that there are a lot of null values in PURPOSE column, so for that we will me filling the null values with a NOT keyword. You can try something else too.



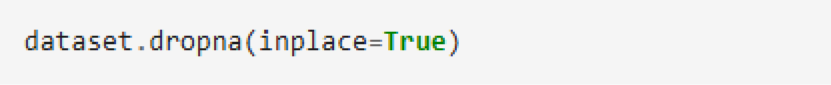
```
dataset['PURPOSE'].fillna("NOT", inplace=True)
```

7. Splitting the START_DATE to date and time column and then converting the time into four different categories i.e. Morning, Afternoon, Evening, Night



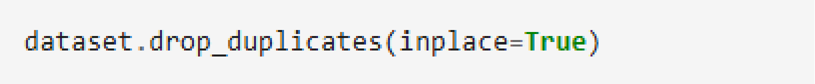
```
from datetime import datetime

dataset['date'] =
pd.DatetimeIndex(dataset['START_DATE']).date
dataset['time'] =
pd.DatetimeIndex(dataset['START_DATE']).hour

#changing into categories of day and night
dataset['day-night'] = pd.cut(x=dataset['time'],
                              bins = [0,10,15,19,24],
                              labels =
['Morning','Afternoon','Evening','Night'])
```

8. Once we are done with **creating new columns**, we can now **drop rows** with **null values**.



```
dataset.dropna(inplace=True)
```

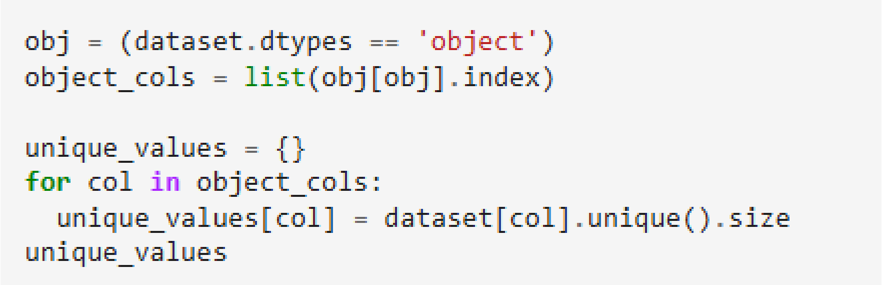
9. It is also important to drop the **duplicates rows** from the dataset. To do that, refer the code below.



```
dataset.drop_duplicates(inplace=True)
```

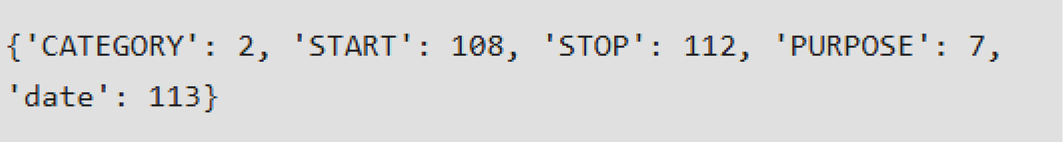
10. Data Visualization

In this section, we will try to understand and compare all columns.
Let’s start with checking the unique values in dataset of the columns with object **datatype**.



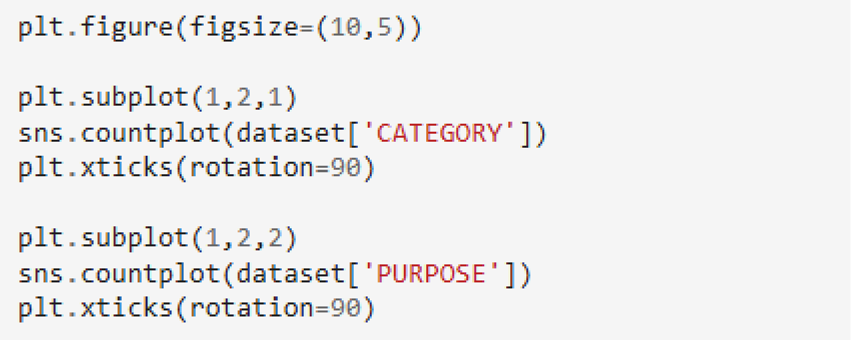
```
obj = (dataset.dtypes == 'object')
object_cols = list(obj[obj].index)

unique_values = {}
for col in object_cols:
  unique_values[col] = dataset[col].unique().size
unique_values
```

Output:



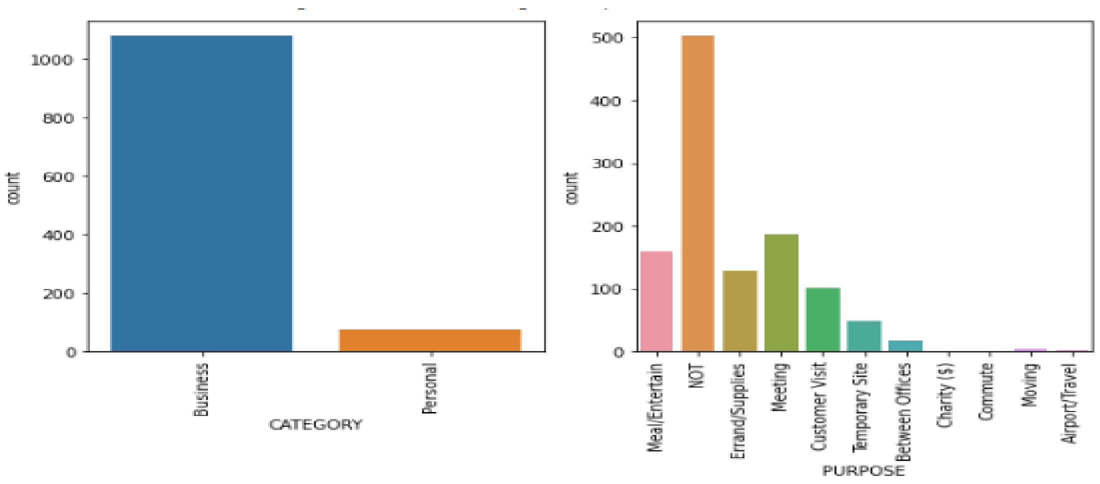
```
{'CATEGORY': 2, 'START': 108, 'STOP': 112, 'PURPOSE': 7,
'date': 113}
```

11. Now, we will be using **matplotlib** and **seaborn library** for **countplot** the **CATEGORY** and **PURPOSE** columns.

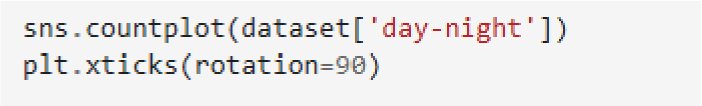


```
plt.figure(figsize=(10,5))

plt.subplot(1,2,1)
sns.countplot(dataset['CATEGORY'])
plt.xticks(rotation=90)

plt.subplot(1,2,2)
sns.countplot(dataset['PURPOSE'])
plt.xticks(rotation=90)
```

Output:

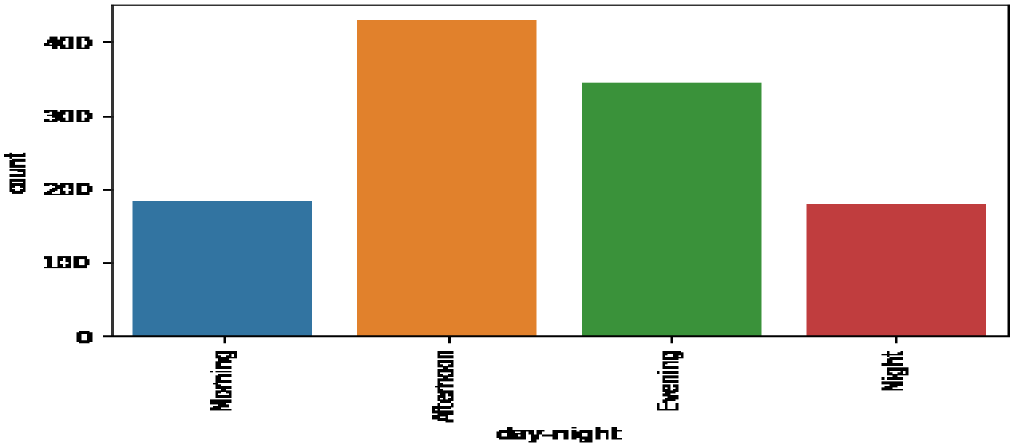


12. Let's do the same for time column, here we will be using the time column which we have extracted above.

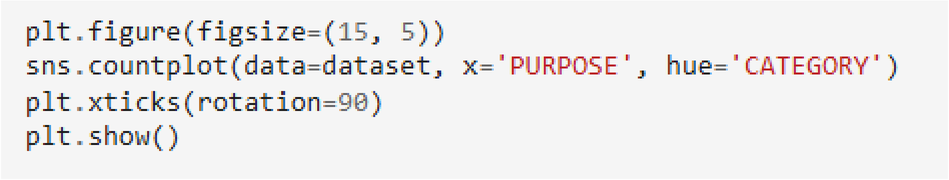


```
sns.countplot(dataset['day-night'])
plt.xticks(rotation=90)
```

Output:

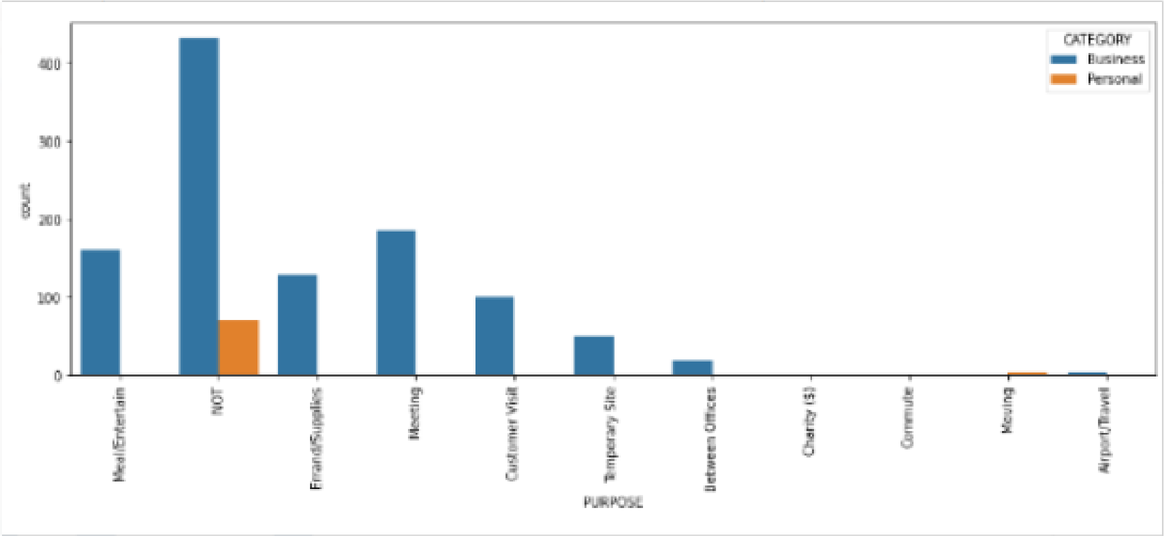


13. Now, we will be comparing the two different categories along with the **PURPOSE** of the user.



```
plt.figure(figsize=(15, 5))
sns.countplot(data=dataset, x='PURPOSE', hue='CATEGORY')
plt.xticks(rotation=90)
plt.show()
```

Output:



14. ***Insights from the above count-plots:***

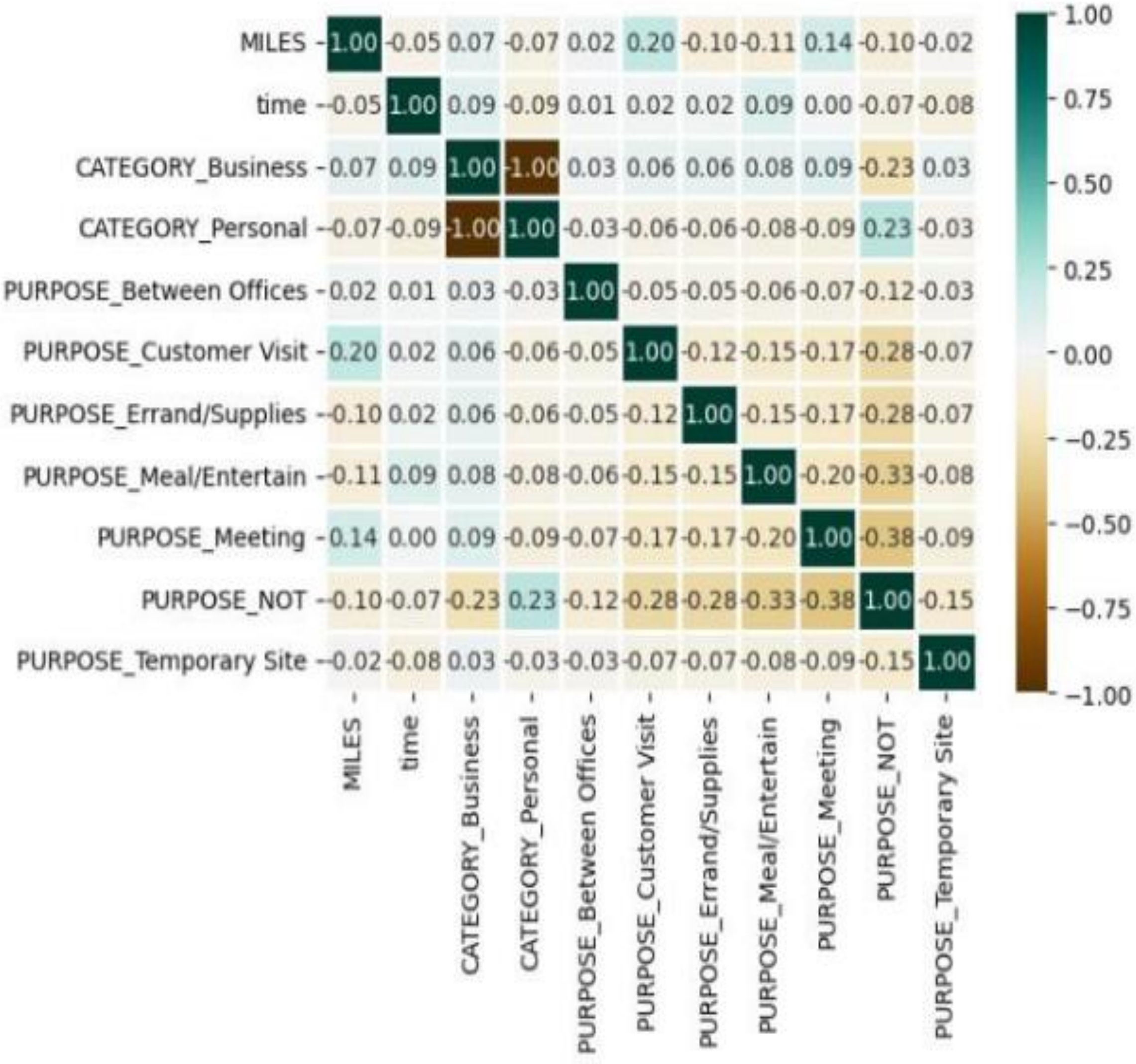
- Most of the rides are booked for business purpose.
- Most of the people book cabs for Meetings and Meal / Entertain purpose.
- Most of the cabs are booked in the time duration of 10am-5pm (Afternoon).

15. After that, we can now find the correlation between the columns using **heatmap**.

```
# Select only numerical columns for correlation
calculation
numeric_dataset = dataset.select_dtypes(include=
['number'])

sns.heatmap(numeric_dataset.corr(),
            cmap='BrBG',
            fmt='.2f',
            linewidths=2,
            annot=True)
```

Output:

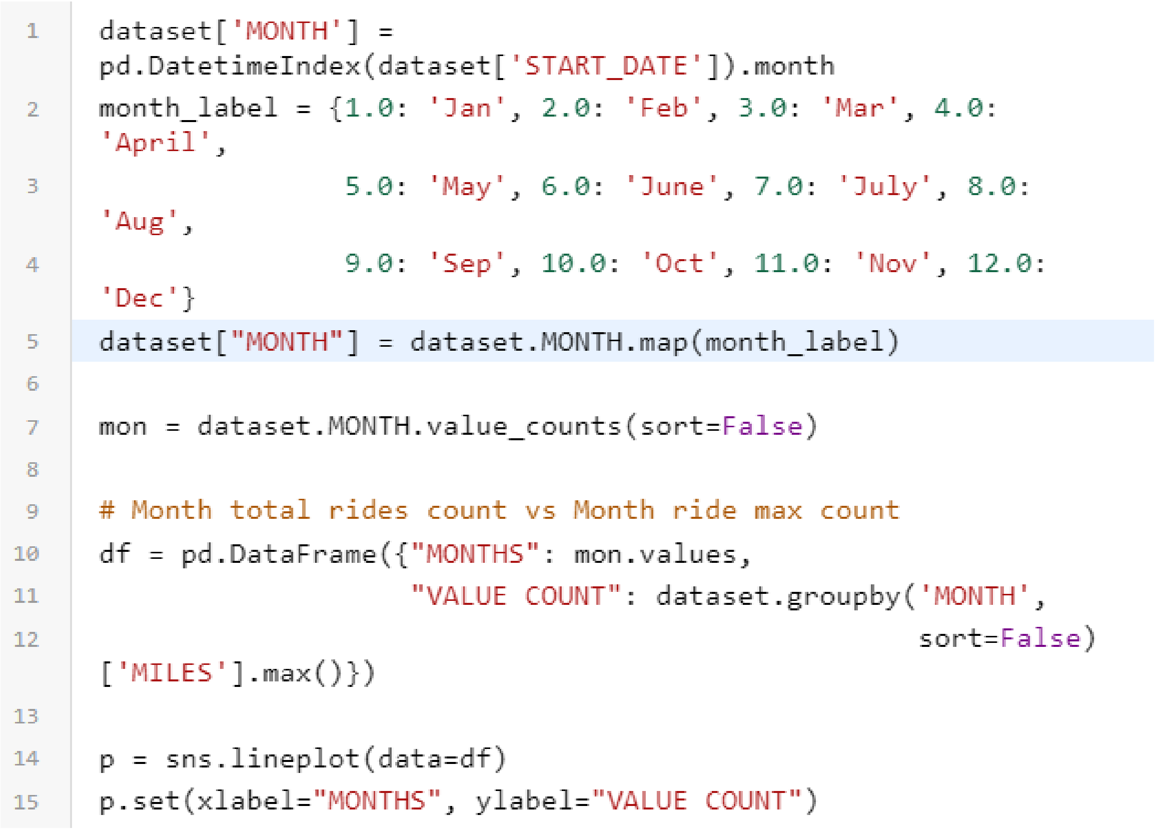
UBER RIDES – EXPLORATORY DATA ANALYSIS



➢ ***Insights from the heatmap:***

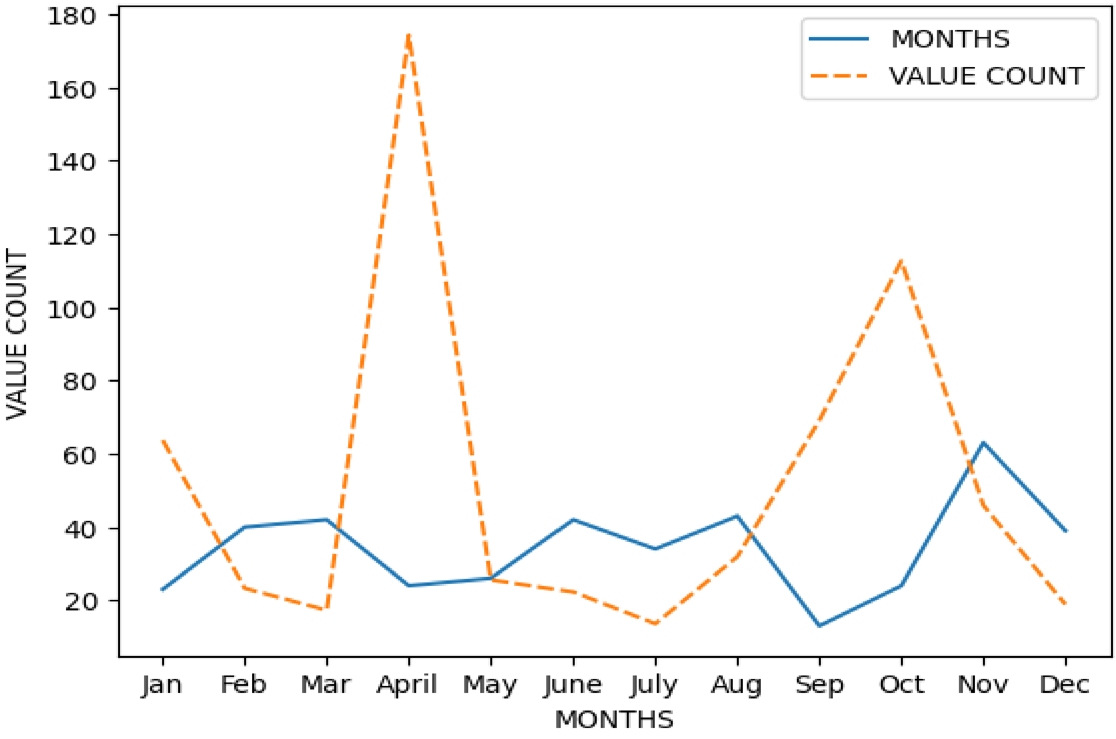
- Business and Personal Category are highly negatively correlated, this have already proven earlier. So, this plot, justifies the above conclusions.
- There is not much correlation between the features.

16. Now, as we need to visualize the month data. This can we same as done before (for hours).



```
dataset['MONTH'] = pd.DatetimeIndex(dataset['START_DATE']).month
month_label = {1.0: 'Jan', 2.0: 'Feb', 3.0: 'Mar', 4.0: 'April',
               5.0: 'May', 6.0: 'June', 7.0: 'July', 8.0: 'Aug',
               9.0: 'Sep', 10.0: 'Oct', 11.0: 'Nov', 12.0: 'Dec'}
dataset["MONTH"] = dataset.MONTH.map(month_label)

mon = dataset.MONTH.value_counts(sort=False)

# Month total rides count vs Month ride max count
df = pd.DataFrame({"MONTHS": mon.values,
                   "VALUE COUNT": dataset.groupby('MONTH',
                                                  sort=False)['MILES'].max()})

p = sns.lineplot(data=df)
p.set(xlabel="MONTHS", ylabel="VALUE COUNT")
```

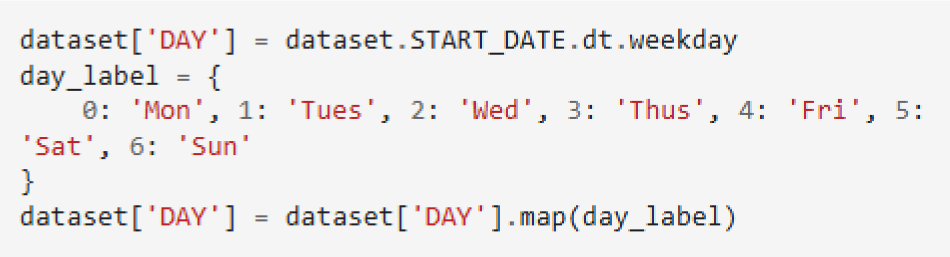
Output:



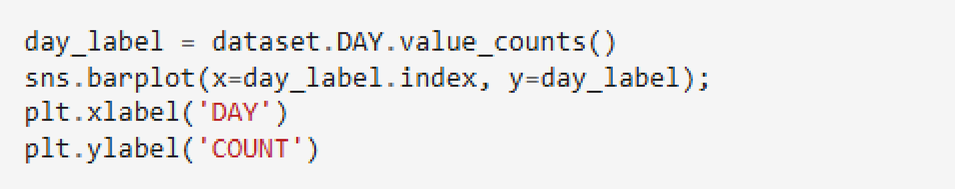
➢ ***Insights from the above plot:***

- The counts are very irregular.
- Still it’s very clear that the counts are very less during Nov, Dec, Jan, which justifies the fact that time winters are there in Florida, US.

17. Visualization for days data.

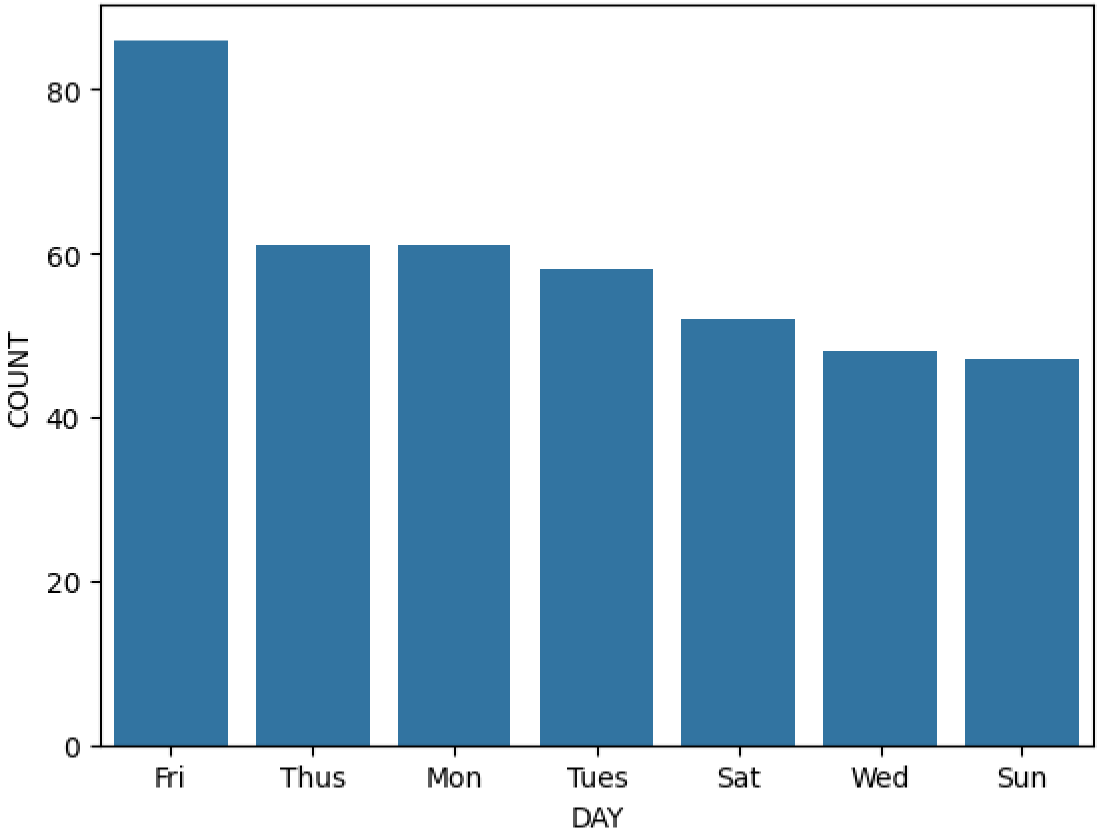


```
dataset['DAY'] = dataset.START_DATE.dt.weekday
day_label = {
    0: 'Mon', 1: 'Tues', 2: 'Wed', 3: 'Thus', 4: 'Fri', 5:
'Sat', 6: 'Sun'
}
dataset['DAY'] = dataset['DAY'].map(day_label)
```

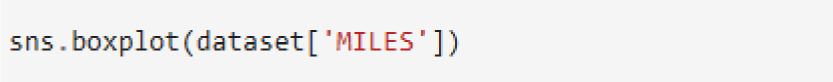


```
day_label = dataset.DAY.value_counts()
sns.barplot(x=day_label.index, y=day_label);
plt.xlabel('DAY')
plt.ylabel('COUNT')
```

Output:

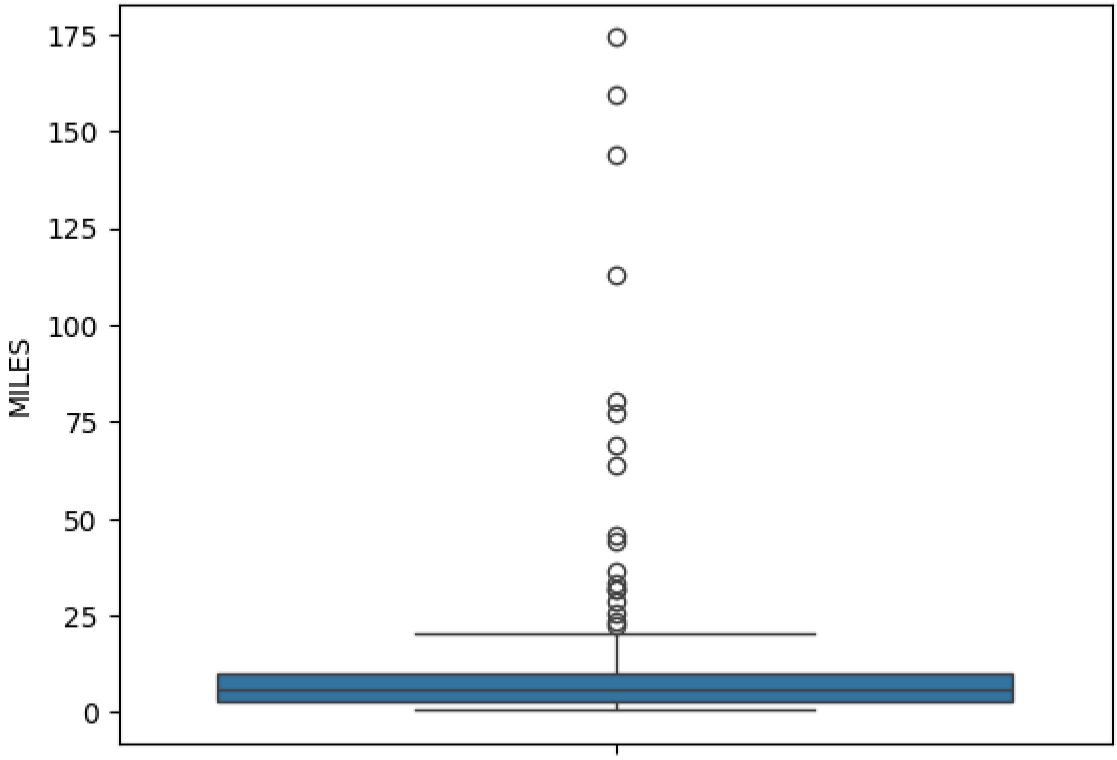


18. Now, let’s explore the **MILES** Column. We can use boxplot to check the distribution of the column

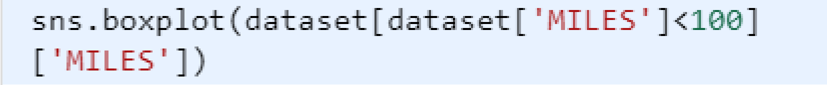


```
sns.boxplot(dataset['MILES'])
```

Output:

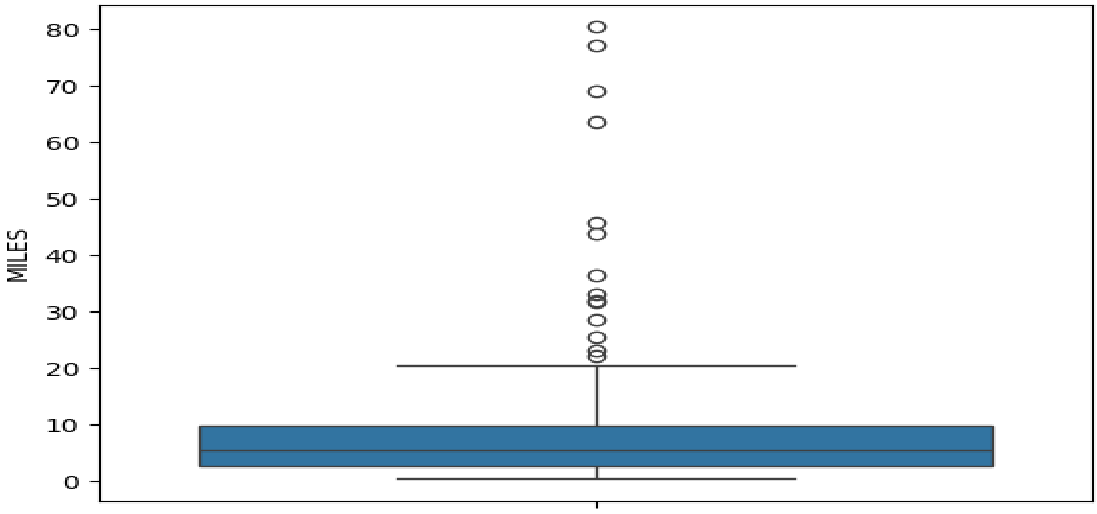


19. As the graph is not clearly understandable. Let's zoom in it for values lees than 100.

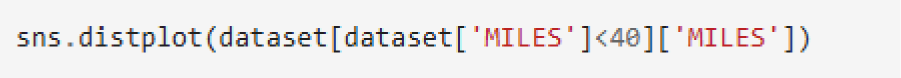


```
sns.boxplot(dataset[dataset['MILES']<100]
['MILES'])
```

Output:

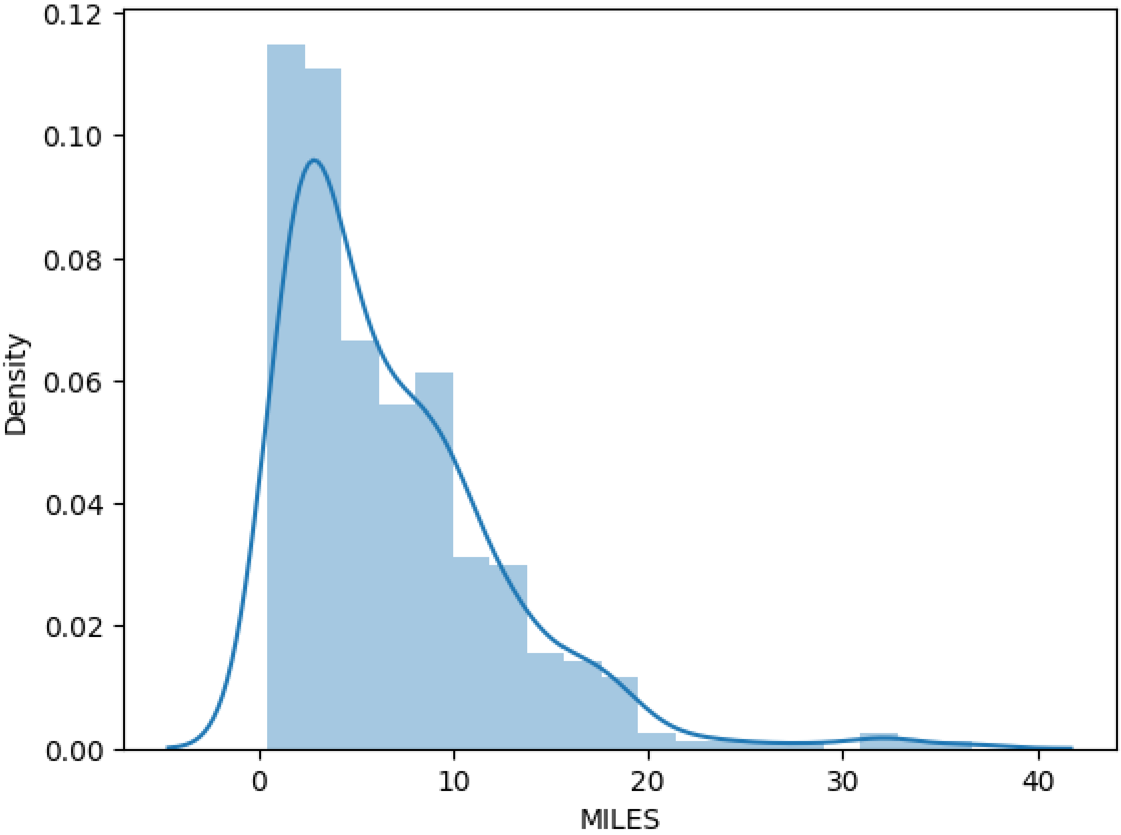


20. It’s bit visible. But to get more clarity we can use distplot for values less than 40.



```
sns.distplot(dataset[dataset['MILES']<40]['MILES'])
```

Output:



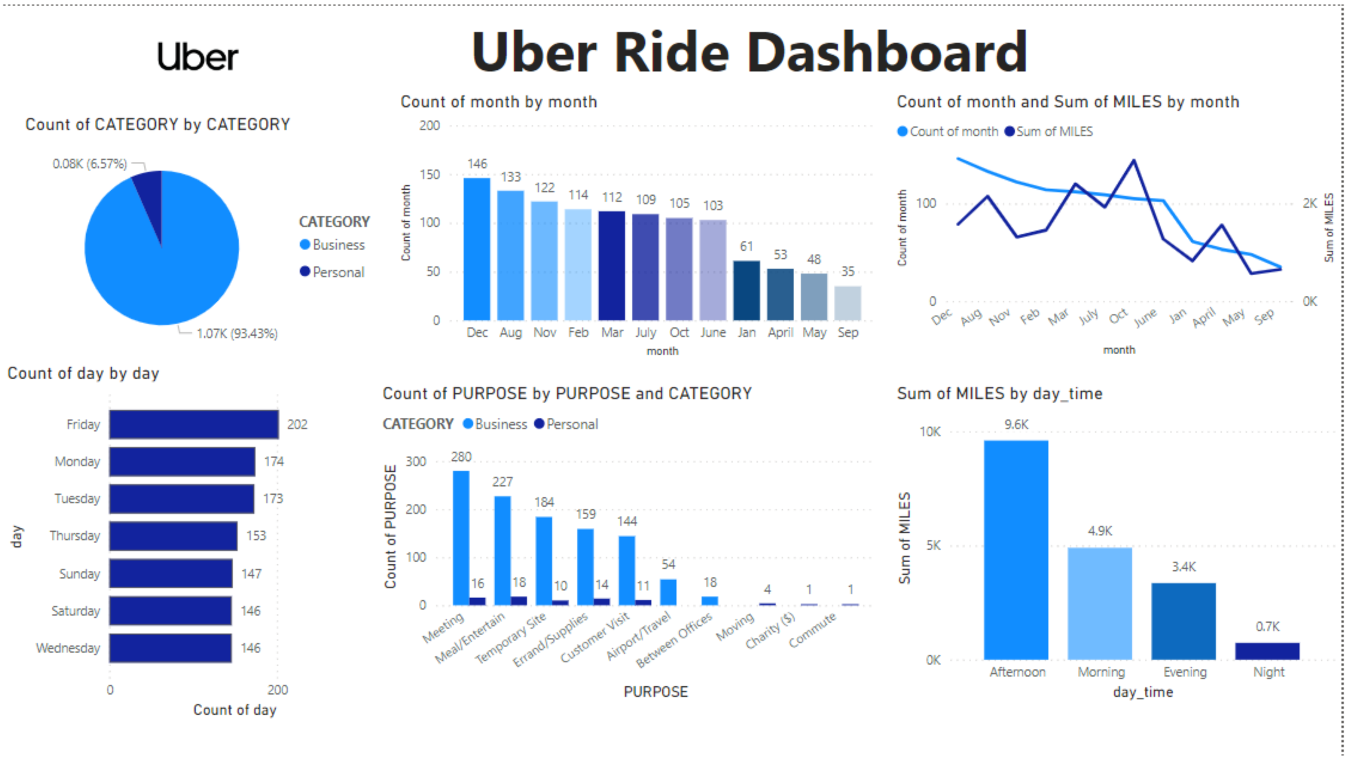
➢ <u>*Insights from the above plots:*</u>

- Most of the cabs booked for the distance of 4-5 miles.
- Majorly people choose cabs for the distance of 0-20 miles.
- For distance more than 20 miles cab counts is nearly negligible.

UBER RIDES – EXPLORATORY DATA ANALYSIS

➢ Uber Ride Dashboard: Key Trends and Highlights

- **Ride Category:** The majority of rides are categorized as "Personal" (93.43%), with "Business" accounting for only 6.57%.
- **Day Analysis:** Fridays are the most common day for rides (202 rides), while Wednesdays see the least usage (146 rides).
- **Miles by Time of Day:** Most miles are travelled in the afternoon (9.6K), followed by the morning (4.9K). Night rides account for the least (0.7K miles).
- **Monthly Trends:** December has the highest number of rides (146), with a noticeable decline from March to September.
- **Purpose of Business Rides:** "Meetings" drive the majority of business rides (280), followed by "Meals/Entertainment" and "Temporary Site visits."



UBER RIDES – EXPLORATORY DATA ANALYSIS

➢ Recommendations: -

1. **Trip Insights**:

   - **Peak Hours**: Analyse trip times to identify peak demand hours for better resource allocation.
   - **Popular Routes**: Highlight frequently travelled routes to optimize driver placement and coverage.

2. **Rider Behaviour**:

   - **Trip Purpose Trends**: Examine trip purposes (e.g., business or leisure) to tailor marketing campaigns.
   - **Distance Patterns**: Identify average trip distances to improve fare estimations and pricing strategies.

3. **Driver Utilization**:

   - **Driver Availability**: Evaluate driver activity during peak times to minimize ride cancellations.
   - **Idle Time Analysis**: Assess idle time to reduce inefficiencies and increase driver earnings.

4. **Revenue Optimization**:

   - **Fare Insights**: Analyse fare trends by distance, time, and category to refine pricing models.
   - **Surge Pricing**: Study surge pricing impact to balance profitability and rider satisfaction.

5. **Operational Efficiency**:

   - **Outlier Detection**: Investigate unusually long trips or distances for possible operational issues.
   - **Duplicate Trips**: Eliminate duplicate entries to ensure clean and accurate reporting.

6. **Visualizations for Clarity**:

   - Include heatmaps for popular areas, line charts for peak hours, and histograms for fare distributions to provide a clear, visual understanding of the data.

7. **Actionable Insights**:

   - Recommend driver incentives during peak hours to increase availability.

Vishal Singh

- Suggest route-based promotions to attract more riders during off-peak times.
- Explore partnerships with businesses for frequent business trip discounts.

- GitHub - https://github.com/VishalSingh11510
- LinkedIn - https://www.linkedin.com/in/vishal-singh-20103a201/