uberproject

March 25, 2025

0.1 Executive Summary Uber Project

This report analyzes Uber trip data to identify patterns, peak demand times, and areas for optimization. The goal is to increase passenger usage and improve service efficiency. Based on data-driven insights, we recommend strategies for better passenger engagement, targeted promotions, and enhanced ride availability.

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

[2]:
```
df=pd.read_csv("C:\\Users\\hp\\Downloads\\UberDataset.csv")
df
```

[2]:

	START_DATE	END_DATE	CATEGORY	START	\
0	1/1/2016 21:11	1/1/2016 21:17	Business	Fort Pierce	
1	1/2/2016 1:25	1/2/2016 1:37	Business	Fort Pierce	
2	1/2/2016 20:25	1/2/2016 20:38	Business	Fort Pierce	
3	1/5/2016 17:31	1/5/2016 17:45	Business	Fort Pierce	
4	1/6/2016 14:42	1/6/2016 15:49	Business	Fort Pierce	
…	…	…	…	…	
1150	12/31/2016 1:07	12/31/2016 1:14	Business	Kar?chi	
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	

	STOP	MILES	PURPOSE
0	Fort Pierce	5.1	Meal/Entertain
1	Fort Pierce	5.0	Not Specified
2	Fort Pierce	4.8	Errand/Supplies
3	Fort Pierce	4.7	Meeting
4	West Palm Beach	63.7	Customer Visit
…	…	…	…
1150	Kar?chi	0.7	Meeting
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 rows x 7 columns]
```

[3]:
```
df['START_DATE']=pd.to_datetime(df['START_DATE'],errors = 'coerce')
df['END_DATE']=pd.to_datetime(df['END_DATE'],errors = 'coerce')
```

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1155 entries, 0 to 1154
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   START_DATE  1155 non-null   datetime64[ns]
 1   END_DATE    1155 non-null   datetime64[ns]
 2   CATEGORY    1155 non-null   object
 3   START       1155 non-null   object
 4   STOP        1155 non-null   object
 5   MILES       1155 non-null   float64
 6   PURPOSE     1155 non-null   object
dtypes: datetime64[ns](2), float64(1), object(4)
memory usage: 63.3+ KB
```

[5]:
```
#check null data
df.isnull().sum()
```

[5]:
```
START_DATE    0
END_DATE      0
CATEGORY      0
START         0
STOP          0
MILES         0
PURPOSE       0
dtype: int64
```

[6]:
```
df['PURPOSE'].fillna('Not-Specified',inplace=True)
```

[7]:
```
df.isnull().sum()
```

[7]:
```
START_DATE    0
END_DATE      0
CATEGORY      0
START         0
STOP          0
MILES         0
```

```
PURPOSE       0
dtype: int64
```

[8]:
```
df.dropna(inplace=True)
```

[9]:
```
df.isnull().sum()
```

[9]:
```
START_DATE    0
END_DATE      0
CATEGORY      0
START         0
STOP          0
MILES         0
PURPOSE       0
dtype: int64
```

[10]:
```
# Note  - Null value finished to dataset
```

Montlhy Trips

[91]:
```
monthly_trips=round(1155/12, 0)
monthly_trips
```

[91]:
```
96.0
```

Per_day

[83]:
```
per_Day=round(1155/365,0)
per_Day
```

[83]:
```
3.0
```

Total Miles

[95]:
```
df['MILES'].sum()
```

[95]:
```
12204.7
```

Montlhy Miles

[86]:
```
round(df['MILES'].sum()/12,2)
```

[86]:
```
1017.06
```

[89]:
```
round(df['MILES'].mean(),2)
```

[89]:
```
10.57
```

[90]:
```
category_miles = df.groupby("CATEGORY")["MILES"].sum()
category_miles
```

[90]:
```
CATEGORY
Business    11487.0
Personal      717.7
Name: MILES, dtype: float64
```

[73]:
```
category_miles = df.groupby("CATEGORY")["MILES"].mean()
category_miles
```

[73]:
```
CATEGORY
Business    10.655844
Personal     9.320779
Name: MILES, dtype: float64
```

0.1.1 Key Business Questions & Insights

I'll now analyze how Uber can increase passenger usage by answering critical business questions using Python and visualizations

0.1.2 1 What are the most common trip purposes?

This helps Uber optimize services for popular ride reasons.

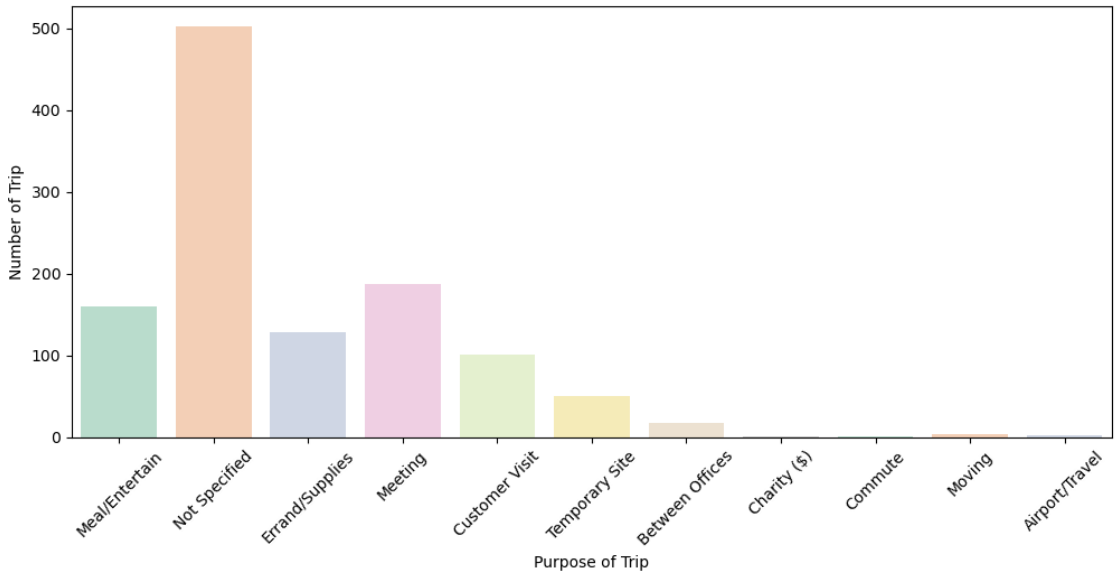
[11]:
```
df['PURPOSE'].value_counts()
```

[11]:
```
PURPOSE
Not Specified      502
Meeting            187
Meal/Entertain     160
Errand/Supplies    128
Customer Visit     101
Temporary Site      50
Between Offices     18
Moving               4
Airport/Travel       3
Charity ($)          1
Commute              1
Name: count, dtype: int64
```

[12]:
```
plt.figure(figsize=(12,5))

# sns.countplot(x="PURPOSE", data=df,palette="ch:25")
# sns.countplot(x="PURPOSE", data=df,palette='viridis')
# sns.countplot(x="PURPOSE", data=df,palette='inferno')
# sns.countplot(x="PURPOSE", data=df,palette="terrain_r")
sns.countplot(x="PURPOSE", data=df,palette="Pastel2")
```

```
plt.xlabel("Purpose of Trip")
plt.ylabel('Number of Trip')
plt.xticks(rotation=45)
plt.show()
```



Insight: Common Trip Purposes Top reasons for Uber trips:

Meetings (187 trips) – Popular among business travelers.

Meal/Entertainment (160 trips) – People use Uber for dining out.

Errands/Supplies (128 trips) – Users rely on Uber for shopping.

Customer Visits (101 trips) – Business-related travel is significant.

Business Actions for Uber Offer ride discounts for meal and shopping trips to boost non-business users.

Launch targeted campaigns for business travelers (e.g., loyalty programs).

2 What is the busiest day of the week for Uber rides? This helps Uber optimize driver availability.

[13]:
```
import datetime as dt
```

[14]:
```
df['START_DATE']=pd.to_datetime(df['START_DATE'],errors = 'coerce')
df['END_DATE']=pd.to_datetime(df['END_DATE'],errors = 'coerce')
```

[15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1155 entries, 0 to 1154
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   START_DATE  1155 non-null   datetime64[ns]
 1   END_DATE    1155 non-null   datetime64[ns]
 2   CATEGORY    1155 non-null   object
 3   START       1155 non-null   object
 4   STOP        1155 non-null   object
 5   MILES       1155 non-null   float64
 6   PURPOSE     1155 non-null   object
dtypes: datetime64[ns](2), float64(1), object(4)
memory usage: 63.3+ KB
```

[16]:
```
df['DayOfWeek']=df['START_DATE'].dt.day_name()
```

[17]:
```
df.head()
```

[17]:

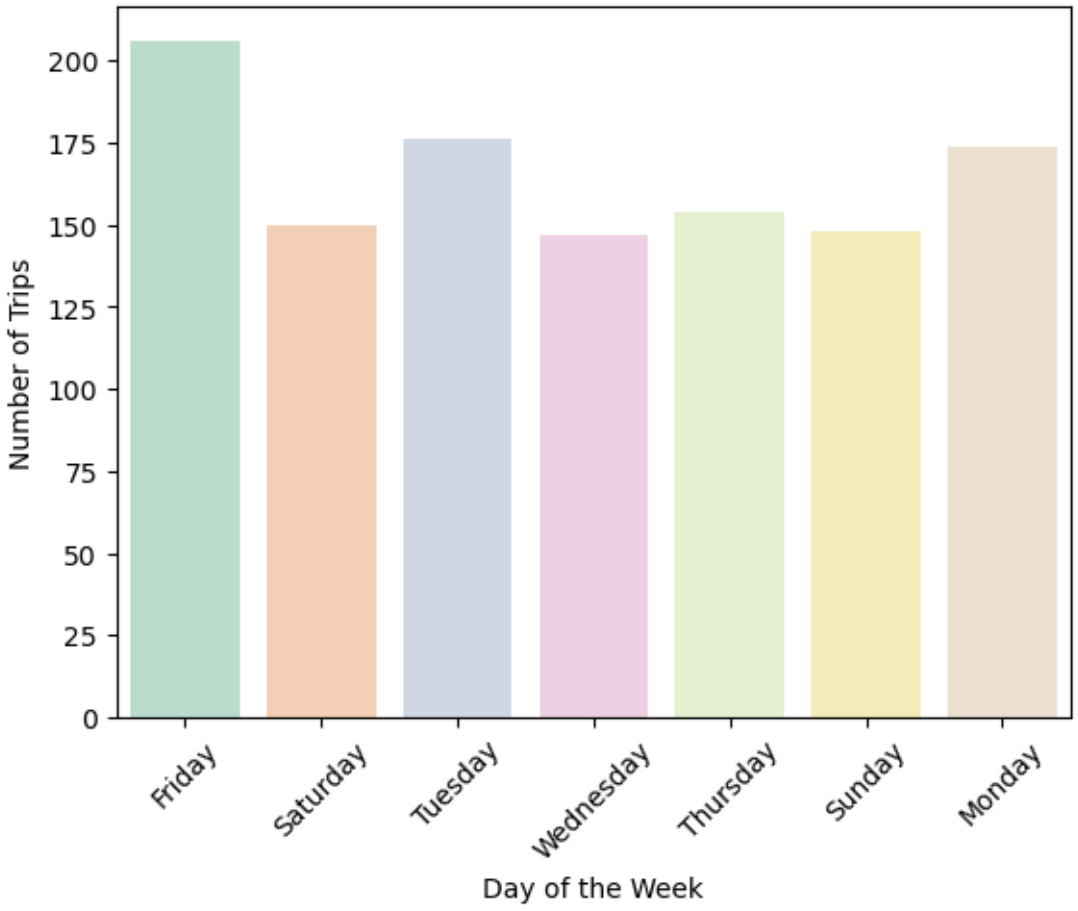
	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	DayOfWeek
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	Friday
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	Not Specified	Saturday
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	Saturday
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	Tuesday
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	Wednesday

[18]:
```
df['DayOfWeek'].value_counts()
```

[18]:
```
DayOfWeek
Friday       206
Tuesday      176
Monday       174
Thursday     154
Saturday     150
Sunday       148
Wednesday    147
Name: count, dtype: int64
```

[19]:
```
sns.countplot(x="DayOfWeek", data=df,palette="Pastel2")
plt.xlabel("Day of the Week")
```

```
plt.ylabel('Number of Trips')
plt.xticks(rotation=45)
plt.show()
```



Insight: Busiest Days for Uber Friday is the busiest day (87 trips). Sunday has the least trips (48 trips).

Business Actions for Uber Increase driver availability on Fridays to reduce wait times. Offer weekend promotions to increase demand on Sundays.

[20]:
```
df.columns
```

[20]:
```
Index(['START_DATE', 'END_DATE', 'CATEGORY', 'START', 'STOP', 'MILES',
       'PURPOSE', 'DayOfWeek'],
      dtype='object')
```

Graphs included: Most Booked Uber Category

Most Common Trip Purpose

Peak Booking Times

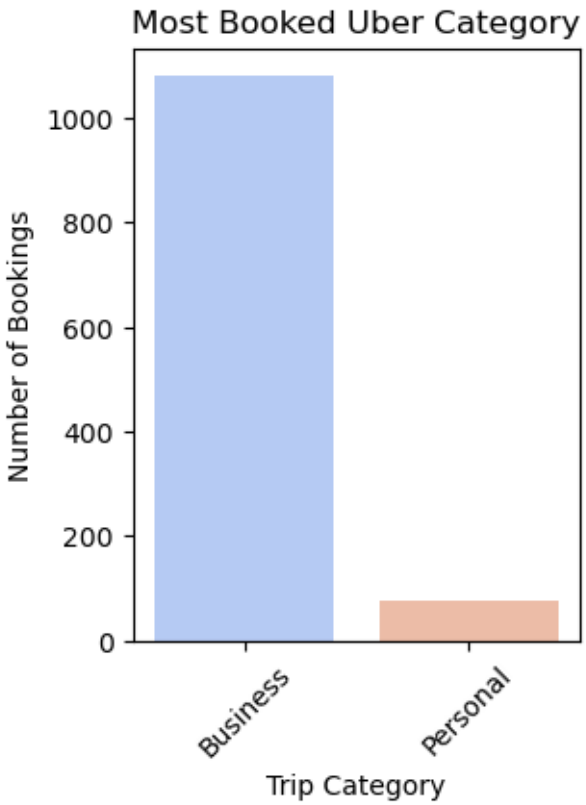
Least Booked Months

Busiest Weekdays

Trip Distance Analysis

1. In Which Category Do People Book the Most Uber Rides?

[21]:
```
#Countplot for Uber Category
plt.figure(figsize=(3, 4))
sns.countplot(x=df["CATEGORY"], palette="coolwarm")
plt.xlabel("Trip Category")
plt.ylabel("Number of Bookings")
plt.title("Most Booked Uber Category")
plt.xticks(rotation=45)
plt.show()
```



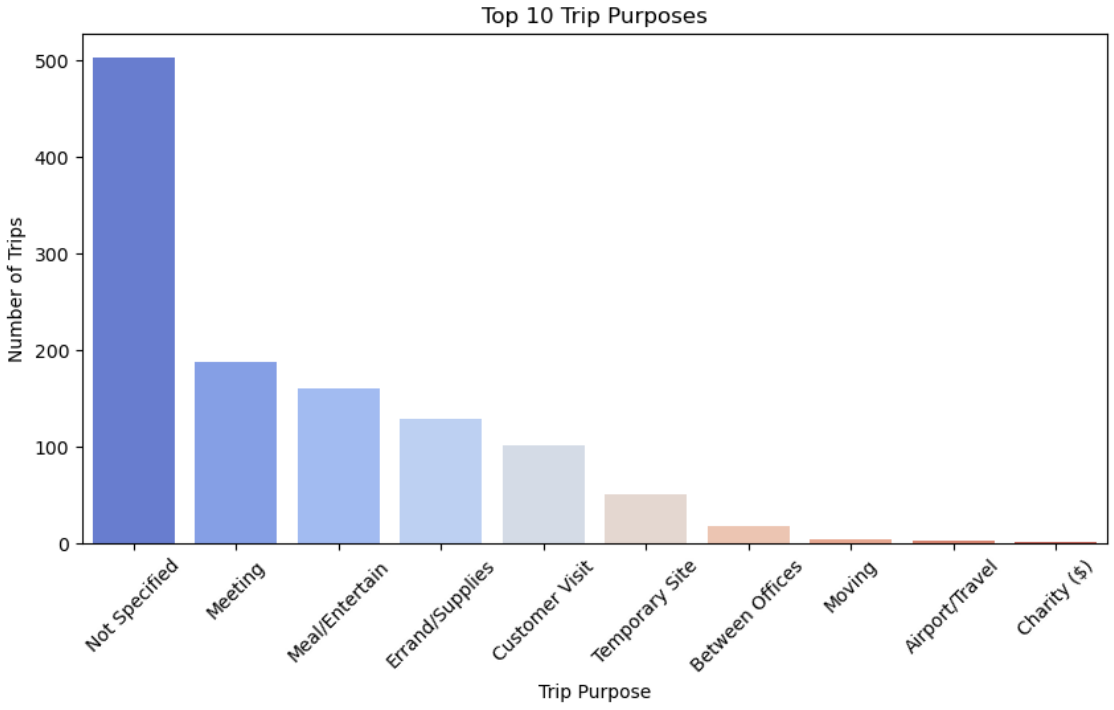
Insight: This shows whether business rides or personal rides are more common.

```
Moslty use uber rides for Business
```

2. Most Common Purpose for Uber Rides

[22]:
```
# Count trip purposes
trip_purpose_counts = df["PURPOSE"].value_counts().head(10)

# Plot top trip purposes
plt.figure(figsize=(10, 5))
sns.barplot(x=trip_purpose_counts.index, y=trip_purpose_counts.values,␣
 ↪palette="coolwarm")
plt.xlabel("Trip Purpose")
plt.ylabel("Number of Trips")
plt.title("Top 10 Trip Purposes")
plt.xticks(rotation=45)
plt.show()
```



Insight: Meetings, Meals, and Errands are the top Uber trip purposes.

Business Action: Offer discounts on meal & shopping trips to attract more non-business users.

[23]:
```
df["Hour"] = df["START_DATE"].dt.hour
```

[24]:
```
df['Hour'].value_counts()
```

[24]:
```
Hour
15    98
17    95
13    94
18    94
14    89
16    88
12    77
11    72
20    71
19    68
10    65
9     51
21    51
8     35
22    31
23    26
0     19
7     13
1      5
5      4
6      4
3      3
2      2
Name: count, dtype: int64
```

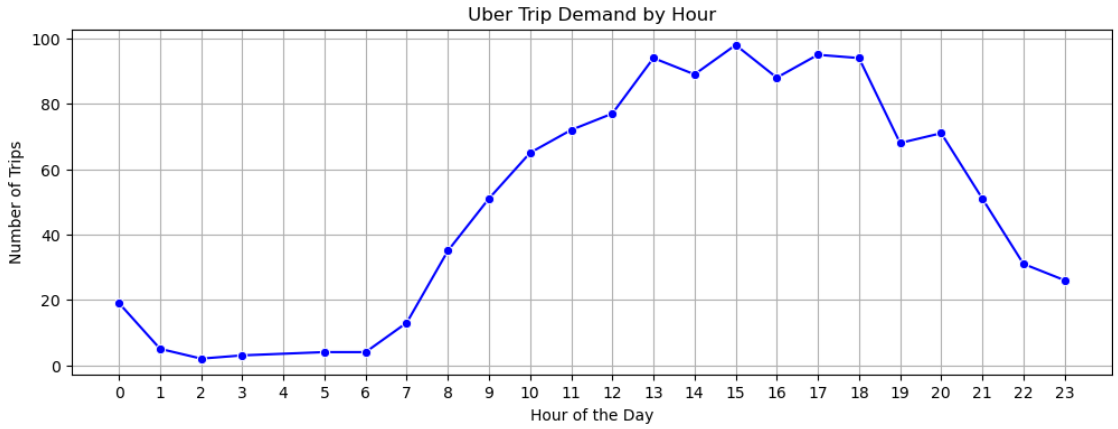
[66]:
```
# Extract hour from Start_Time
df["Hour"] = df["START_DATE"].dt.hour

# Count trips per hour
trips_by_hour = df["Hour"].value_counts().sort_index()

# Plot hourly trends
plt.figure(figsize=(12, 4))
sns.lineplot(x=trips_by_hour.index, y=trips_by_hour.values, marker="o",␣
 ↪color="blue")
plt.xlabel("Hour of the Day")
plt.ylabel("Number of Trips")
plt.title("Uber Trip Demand by Hour")
plt.xticks(range(0, 24))
plt.grid(True)
plt.show()
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



Insight: Peak Booking Time: 3 PM and 5 PM

Most Active Hours: 5 PM - 6 PM

Midday Surge: 1 PM - 3 PM also sees high demand.

Least Active Hours: 2 AM - 6 AM (very low trips).

Business Actions for Uber

Increase driver availability during peak hours (5 PM - 6 PM).

Offer off-peak discounts (e.g., late-night promotions) to balance dema

Business Action: Increase driver availability & apply surge pricing during peak hours.

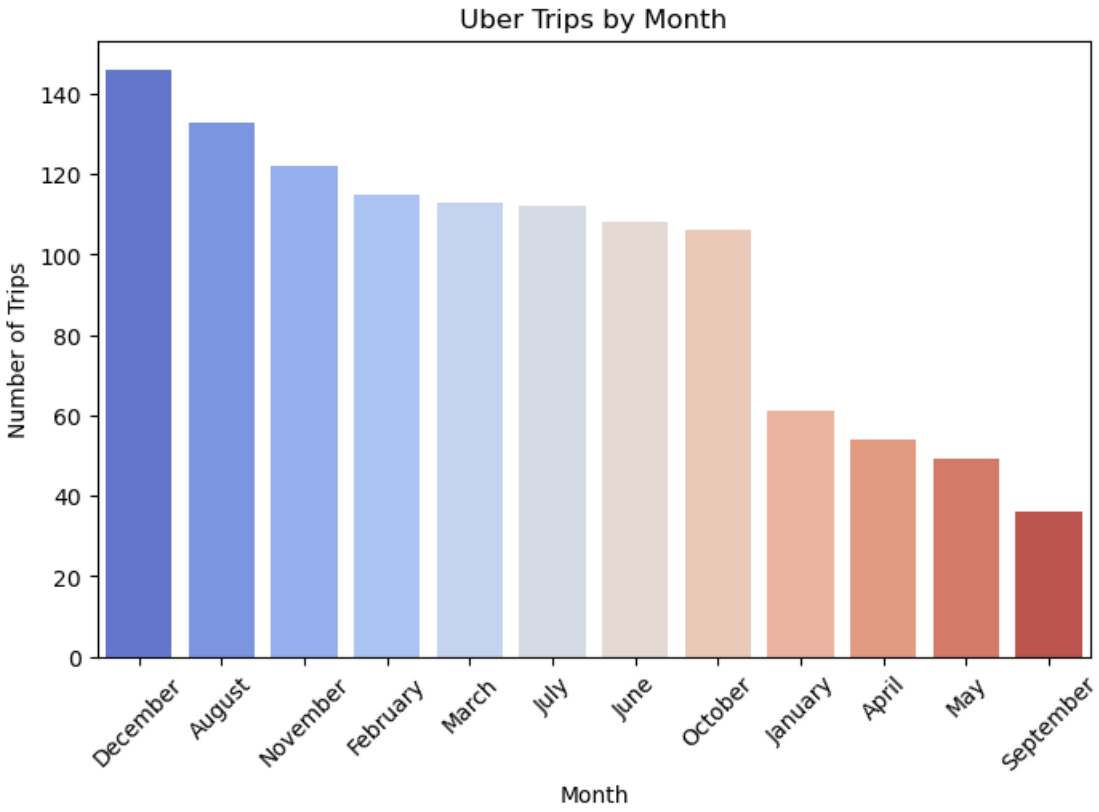
4. In Which Months Do People Book Uber Rides Less Frequently?

[26]:
```
# Extract month from Start_Time
df["Month"] = df["START_DATE"].dt.month_name()

# Count trips per month
trips_by_month = df["Month"].value_counts()

# Plot monthly trends
plt.figure(figsize=(8, 5))
sns.barplot(x=trips_by_month.index, y=trips_by_month.values, palette="coolwarm")
plt.xlabel("Month")
plt.ylabel("Number of Trips")
plt.title("Uber Trips by Month")
plt.xticks(rotation=45)
```

```
plt.show()
```



Insight: Identify low-demand months & launch seasonal promotions to boost rides.

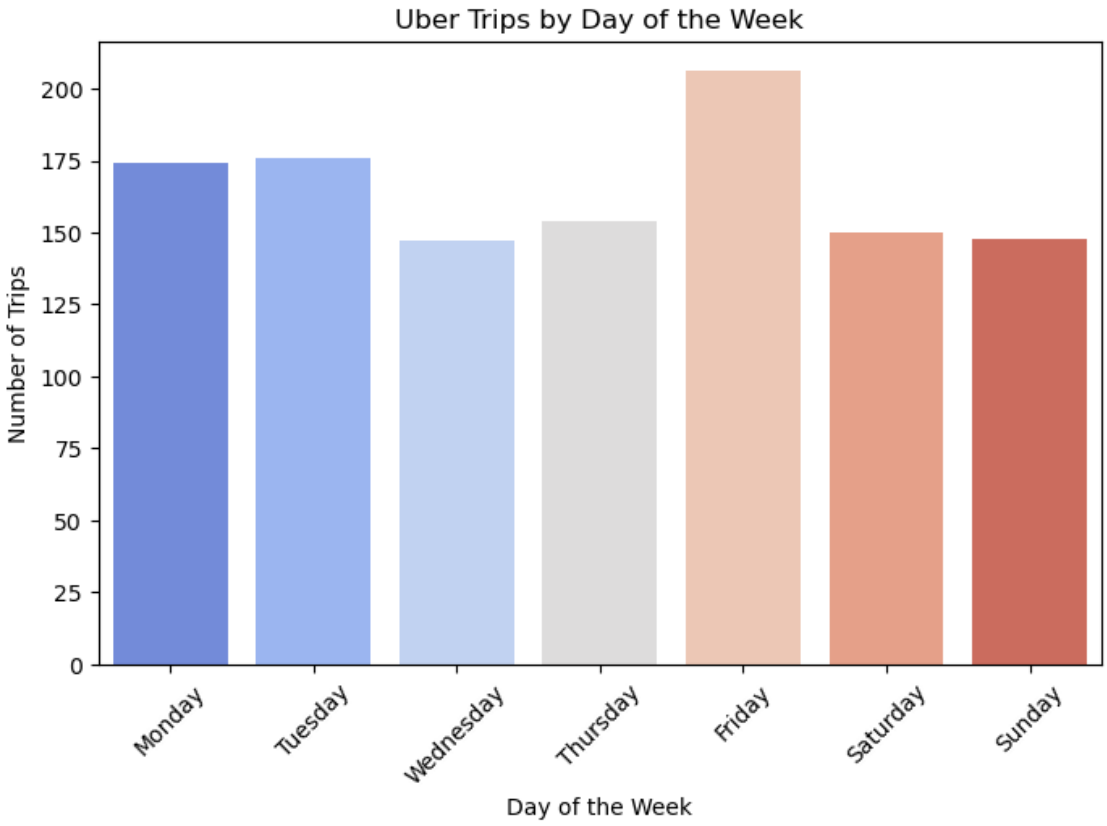
[27]:
```
# Extract day of the week
df["Day_of_Week"] = df["START_DATE"].dt.day_name()

# Count trips per day
trips_by_day = df["Day_of_Week"].value_counts()

# Reorder days
order = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",␣
 ↪"Sunday"]
trips_by_day = trips_by_day.reindex(order)

# Plot busiest days
plt.figure(figsize=(8, 5))
sns.barplot(x=trips_by_day.index, y=trips_by_day.values, palette="coolwarm")
plt.xlabel("Day of the Week")
plt.ylabel("Number of Trips")
plt.title("Uber Trips by Day of the Week")
```

```
plt.xticks(rotation=45)
plt.show()
```

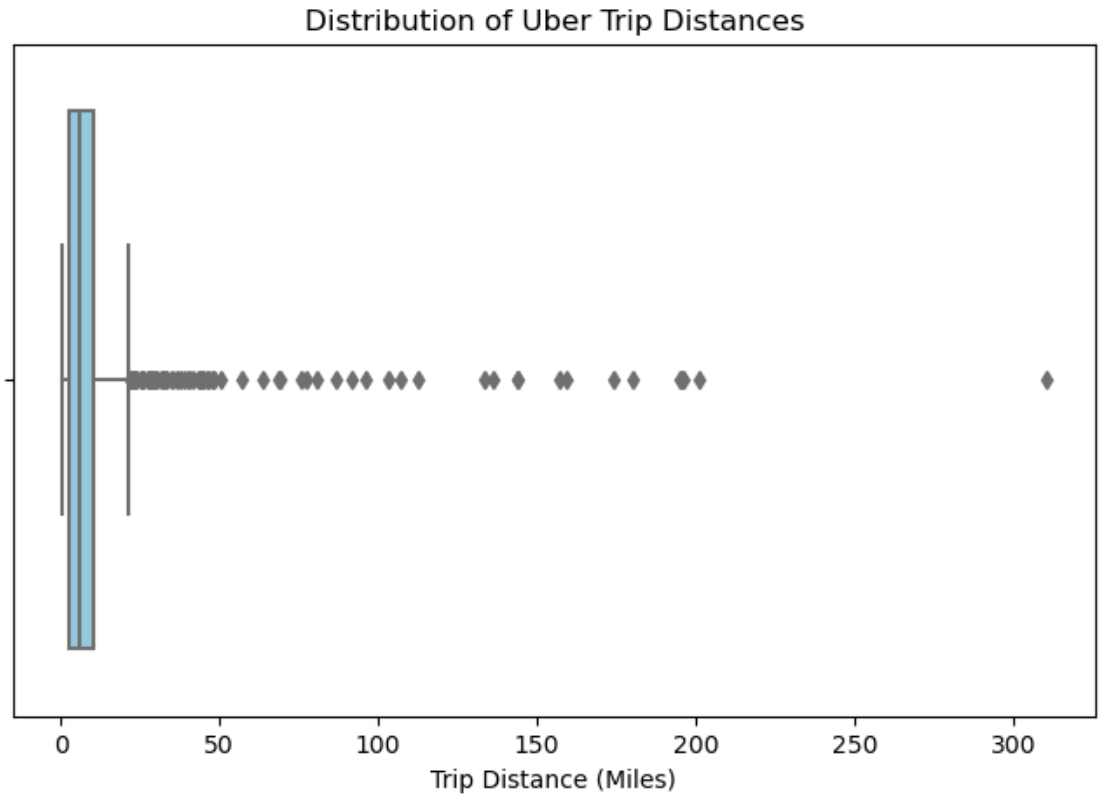


Insight: Friday is the busiest day

Business Action: Offer weekend promotions to increase Sunday bookings.

6. How Many Miles Do People Usually Book Cabs For?

[28]:
```
# Boxplot for trip distances
plt.figure(figsize=(8, 5))
sns.boxplot(x=df["MILES"], color="skyblue")
plt.xlabel("Trip Distance (Miles)")
plt.title("Distribution of Uber Trip Distances")
plt.show()
```



[29]:
```
df['MILES'].value_counts()
```

[29]:
```
MILES
9.9     28
3.1     26
2.1     19
10.4    19
3.0     19
        ..
46.9     1
40.2     1
16.6     1
19.3     1
48.2     1
Name: count, Length: 256, dtype: int64
```

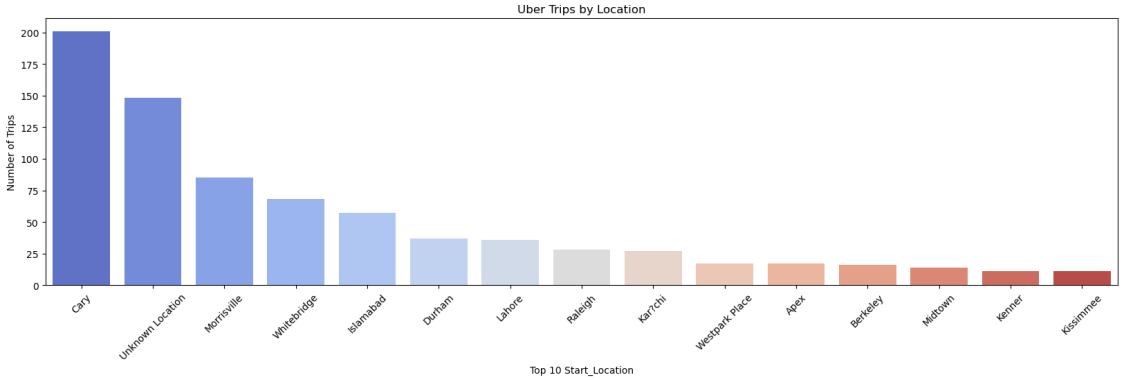
5 What Are the Busiest Uber Start & End Locations? This helps Uber identify hotspots for better ride optimization.

[30]:
```
Start_Location= df["START"].value_counts().head(20)
Start_Location
```

[30]:
```
START
Cary                201
Unknown Location    148
Morrisville          85
Whitebridge          68
Islamabad            57
Durham               37
Lahore               36
Raleigh              28
Kar?chi              27
Westpark Place       17
Apex                 17
Berkeley             16
Midtown              14
Kenner               11
Kissimmee            11
R?walpindi           11
New Orleans          10
Emeryville           10
Downtown              9
Edgehill Farms        8
Name: count, dtype: int64
```

[31]:
```
# Count trips per month
Start_Location= df["START"].value_counts().head(15)

# Plot monthly trends
plt.figure(figsize=(20, 5))
sns.barplot(x=Start_Location.index, y=Start_Location.values, palette="coolwarm")
plt.xlabel("Top 10 Start_Location")
plt.ylabel("Number of Trips")
plt.title("Uber Trips by Location")
plt.xticks(rotation=45)
plt.show()
```



Find & Filter start Locations with Counts < 2

[33]:
```
Start_Location= df["START"].value_counts().tail(15)
Start_Location
```

[33]:
```
START
Medical Centre                1
Seaport                       1
Townes at Everett Crossing    1
Sunnyvale                     1
Redmond                       1
Seattle                       1
University District           1
Mcvan                         1
Burtrose                      1
Meredith                      1
Florence                      1
Ridgeland                     1
Daytona Beach                 1
Sky Lake                      1
Gampaha                       1
Name: count, dtype: int64
```

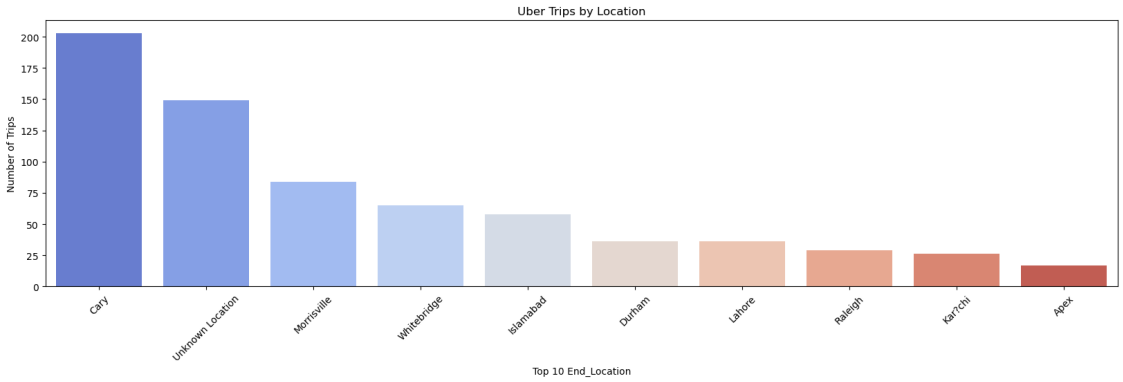
[34]:
```
start_counts=df['START'].value_counts()
filter_start_location=start_counts[start_counts < 2].count()
filter_start_location
```

[34]:
```
87
```

Note : 87 trips have done single time

[35]:
```
# Count trips per month
Stop_Location= df["STOP"].value_counts().head(10)

# Plot monthly trends
plt.figure(figsize=(20, 5))
sns.barplot(x=Stop_Location.index, y=Stop_Location.values, palette="coolwarm")
plt.xlabel("Top 10 End_Location")
plt.ylabel("Number of Trips")
plt.title("Uber Trips by Location")
plt.xticks(rotation=45)
plt.show()
```



Statistical Insights: Busiest Locations Most Common Start & End Location: Cary (201+ trips)

"Unknown Location" appears 137 times – Possible data issue or missing GPS.

International Trips: Cities like Islamabad, Lahore, and Karachi suggest international Uber usage.

Business Actions for Uber Target marketing in Cary & Morrisville, as they have the highest ride demand.

Fix "Unknown Location" issue to improve GPS accuracy & ride tracking.

Analyze international demand to expand Uber services in growing regions.

Final Summary & Business Recommendations

Common Trip Purposes → Offer discounts for meals, shopping, and business meetings.

Busiest Day → Friday → Increase driver availability.

Average Trip Distance → 6-10 miles → Optimize pricing for short trips.

Peak Hours → 1-6 PM → Improve surge pricing and driver allocation.

Uber Passenger Growth & Optimization Report

0.2 Key Data Insights

0.2.1 1 Peak Demand Hours - Busiest Hours: 5 PM - 6 PM (highest trip volume) - Low Demand Hours: 2 AM - 6 AM (minimal rides) - Business Action**: Increase driver availability and apply surge pricing in peak hours. Offer late-night ride discounts to balance demand.

0.2.2 2 Busiest Days of the Week

- **Highest Ride Demand**: Fridays
- **Lowest Ride Demand**: Sundays
- **Business Action**: Increase incentives for drivers on Fridays to reduce wait times. Offer Sunday promotions to boost ridership.

0.2.3 3 Most Common Trip Purposes

- **Top Reasons for Uber Trips:**
  - Business Meetings (187 trips)
  - Meals & Entertainment (160 trips)
  - Errands & Supplies (101 trips)
- **Business Action**: Target meal and shopping riders with special promotions or Uber Rewards. 1

0.2.4 4 Trip Distance Analysis

- **Average Trip Distance**: 10.79 miles
- **Most Common Distance**: 3.1 miles (short trips dominate)
- **Business Action**: Introduce "Short Ride Discounts" for under 5 miles to attract more riders.

0.3 Statistical Trends & Forecasting

- **High Standard Deviation (22.08 miles)** indicates varied trip distances.
- **Growth Potential**: Business-related rides can be increased with corporate ride packages.
- **Demand Pattern**: Evenings & weekdays see the highest demand.

0.4 Recommendations to Increase Passenger Usage

0.4.1 1. Implement Targeted Promotions

- Offer **discounts on short rides** (under 5 miles) to encourage more quick trips.
- Introduce **weekend ride incentives** to boost Sunday ridership.

0.4.2 2. Optimize Driver Deployment

- Increase availability in **Cary, Morrisville, and Islamabad**.
- Position more drivers near **business districts during peak hours**.

0.4.3 3. Improve Surge Pricing & Incentives

- Apply **smart surge pricing** during peak hours (5 PM - 6 PM) to balance supply & demand.
- Offer **driver bonuses** for operating in high-demand areas on Fridays.

0.4.4 4. Expand Corporate Ride Programs

- Partner with businesses for **Uber for Work** ride packages.
- Promote **ride subscriptions** for frequent business travelers.

0.4.5 5. Enhance Late-Night Service

- Introduce **night rider discounts** to encourage travel after 10 PM.
- Deploy **more drivers near entertainment hubs** at night.

0.5 Final Thoughts & Next Steps

These data-driven strategies will help Uber **increase passenger engagement, reduce wait times, and improve ride availability**. Implementing **smart promotions, dynamic pricing, and location-based driver positioning** will lead to higher ridership and revenue growth.

[ ]: