UBER

UBER Analysis

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

In [2]:
```
db=pd.read_csv("UberDb.csv")
db.head()
```

Out[2]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE
0	1/1/2016 21:11	1/1/2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	1/2/2016 1:25	1/2/2016 1:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	1/2/2016 20:25	1/2/2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	1/5/2016 17:31	1/5/2016 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	1/6/2016 14:42	1/6/2016 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit

In [3]:
```
db.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
```

In [4]:
```
print('-There are 1156 entries\n- All columns are non-null(Except Purpose)\n')
```

```
-There are 1156 entries
- All columns are non-null(Except Purpose)
```

Data Preprocessing

In [5]:
```
db['PURPOSE'].isna().sum()
```

Out[5]:
```
503
```

In [6]:
```
print('-Their are 503 Null data in Purpose column')
```

```
-Their are 503 Null data in Purpose column
```

In [7]:

```
db['PURPOSE'] = db['PURPOSE'].fillna('Unknown')
```

In [8]:

```
db.dtypes
```

Out[8]:

```
START_DATE     object
END_DATE       object
CATEGORY       object
START          object
STOP           object
MILES         float64
PURPOSE        object
dtype: object
```

In [9]:

```
print("-Datatype of 'START_DATE' and 'END_DATE' are in object we would convert
```

```
-Datatype of 'START_DATE' and 'END_DATE' are in object we would convert it in
to datetime format
```

In [10]:

```
db['START_DATE']=pd.to_datetime(db['START_DATE'],errors='coerce')
```

In [11]:

```
db['END_DATE']=pd.to_datetime(db['END_DATE'],errors='coerce')
```

In [12]:

```
db.dtypes
```

Out[12]:

```
START_DATE    datetime64[ns]
END_DATE      datetime64[ns]
CATEGORY              object
START                 object
STOP                  object
MILES                float64
PURPOSE               object
dtype: object
```

In [13]:

```
print("Creating 2 new columns of Starting Date and Time")
```

```
Creating 2 new columns of Starting Date and Time
```

In [14]:

```
db['DATE'] = db['START_DATE'].dt.date
db['Time']=db['START_DATE'].dt.hour
```

In [15]:
```
db.head()
```

Out[15]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	DATE	Time
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	2016-01-01	21.0
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	Unknown	2016-01-02	1.0
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	2016-01-02	20.0
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	17.0
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	2016-01-06	14.0

We Will create a new column to categories time on the basis of morning,afternoon,evening,Night

Using 'CUT' Function we are going to divide the data into interval

In [16]:
```
db['Day-Night']=pd.cut(x=db['Time'],bins=[0,10,15,19,24],labels=['Morning','Af
```

In [17]:
```
db.head()
```

Out[17]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	DATE	Time	
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	2016-01-01	21.0	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	Unknown	2016-01-02	1.0	
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	2016-01-02	20.0	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	17.0	
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	2016-01-06	14.0	A

```
In [19]: db.head()
```

Out[19]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	DATE	Time	
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	2016-01-01	21.0	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	Unknown	2016-01-02	1.0	
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	2016-01-02	20.0	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	17.0	
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	2016-01-06	14.0	A

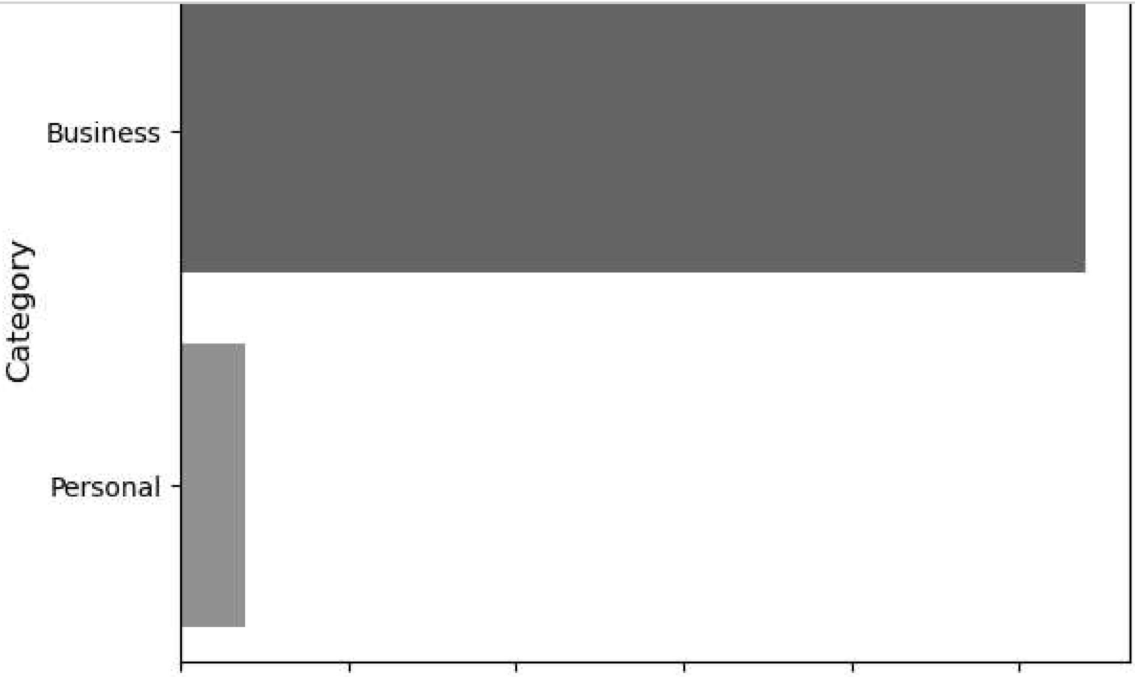
```
In [20]: db.shape
Out[20]: (1156, 10)
```

In Which Category do people book the most uber rides?

```
In [21]: category_counts = db['CATEGORY'].value_counts()
```

In [22]:

```
sns.countplot(y=db['CATEGORY'])
plt.xticks(rotation=90)
plt.ylabel('Category', fontsize=12)
plt.xlabel('Number of Rides', fontsize=12)
plt.title('Number of Uber Rides by Category', fontsize=14)
```



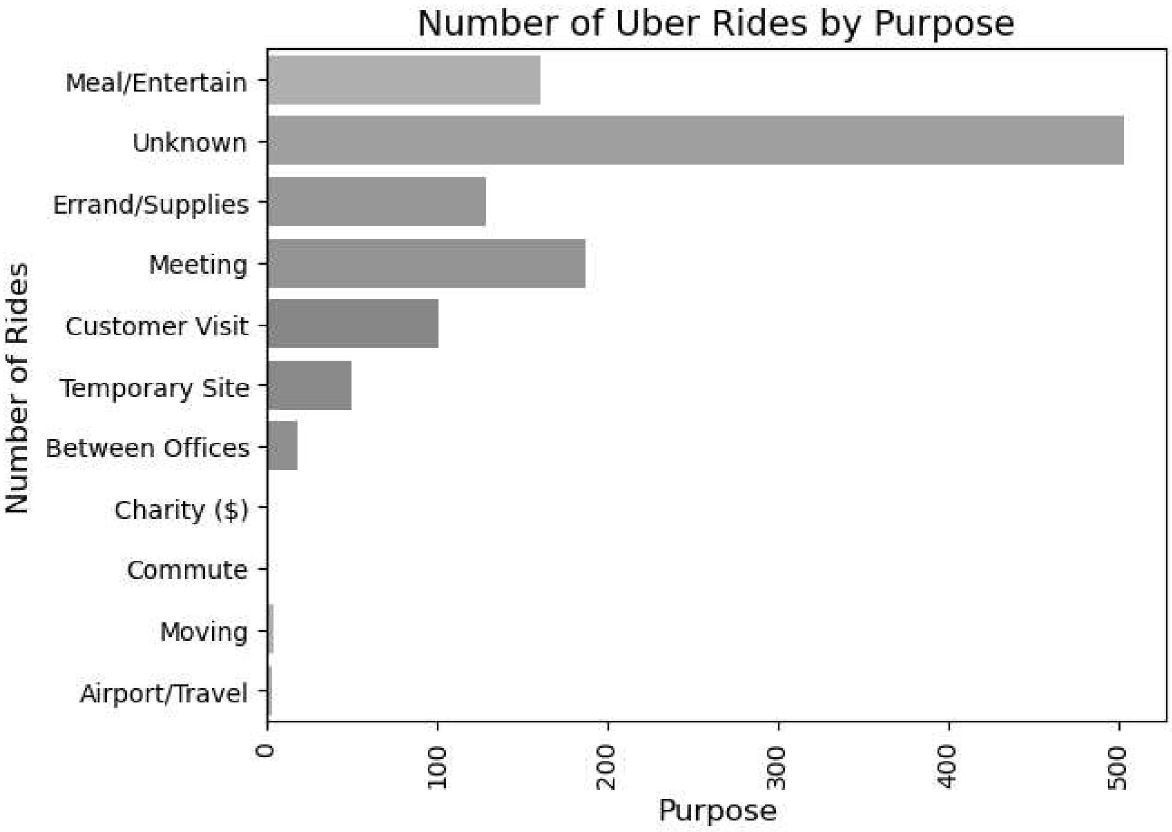
Q)For What purpose Do people book Uber rides the most?

In [23]:

```
sns.countplot(y=db['PURPOSE'])
plt.xticks(rotation=90)
plt.xlabel('Purpose', fontsize=12)
plt.ylabel('Number of Rides', fontsize=12)
plt.title('Number of Uber Rides by Purpose', fontsize=14)
```

Out[23]:

```
Text(0.5, 1.0, 'Number of Uber Rides by Purpose')
```



Q)In which month do people book cabs the most from uber?

In [24]:

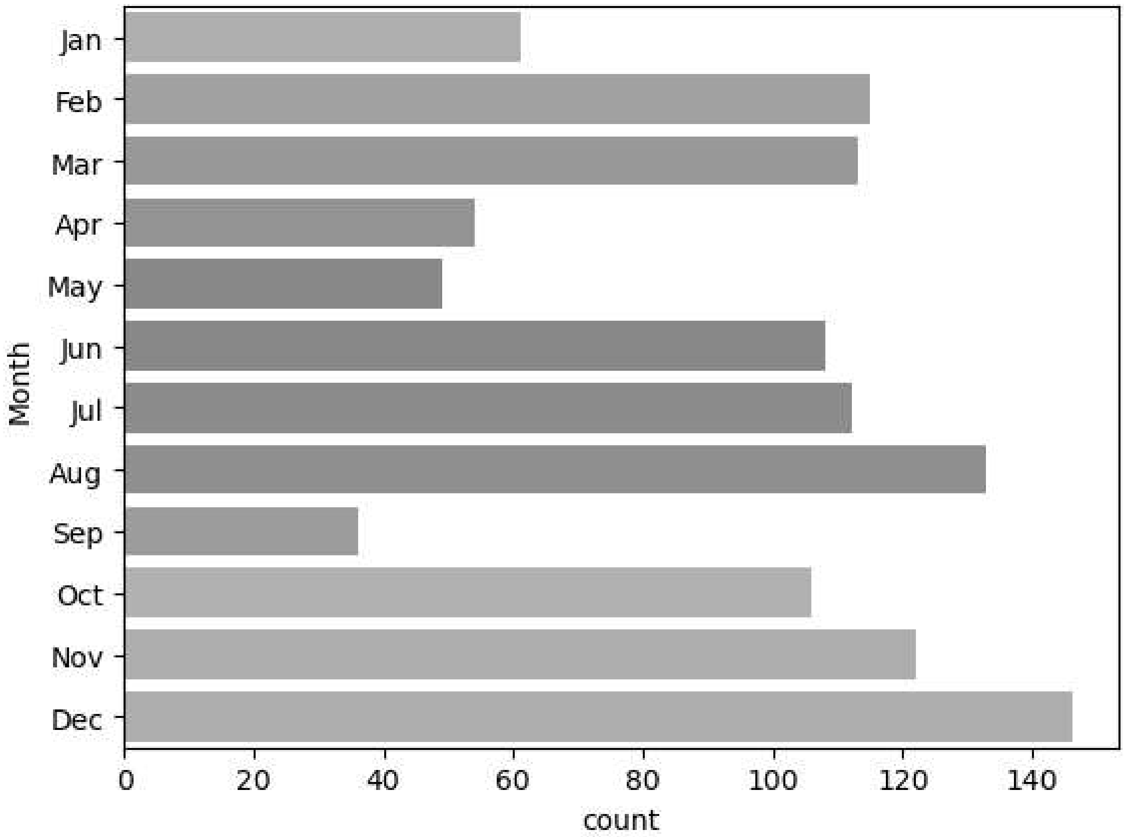
```
db['Month']=db['START_DATE'].dt.month
```

In [25]:

```
month_map = {
    1: 'Jan', 2: 'Feb', 3: 'Mar', 4: 'Apr', 5: 'May', 6: 'Jun',
    7: 'Jul', 8: 'Aug', 9: 'Sep', 10: 'Oct', 11: 'Nov', 12: 'Dec'
}

db['Month'] = db['Month'].map(month_map)
```

```
In [26]: sns.countplot(y=db['Month'])
```

```
Out[26]: <Axes: xlabel='count', ylabel='Month'>
```



```
In [27]: db.head()
```

Out[27]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	DATE	Time	
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	2016-01-01	21.0	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	Unknown	2016-01-02	1.0	
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	2016-01-02	20.0	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	17.0	
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	2016-01-06	14.0	A

In Which days of the week do people book uber rides the most?

In [28]:

```
db['Day']=db['START_DATE'].dt.weekday
```

In [29]:

```
day_label = {
    0: 'Mon', 1: 'Tue', 2: 'Wed', 3: 'Thu', 4: 'Fri', 5: 'Sat', 6: 'Sun'
}
```

In [30]:

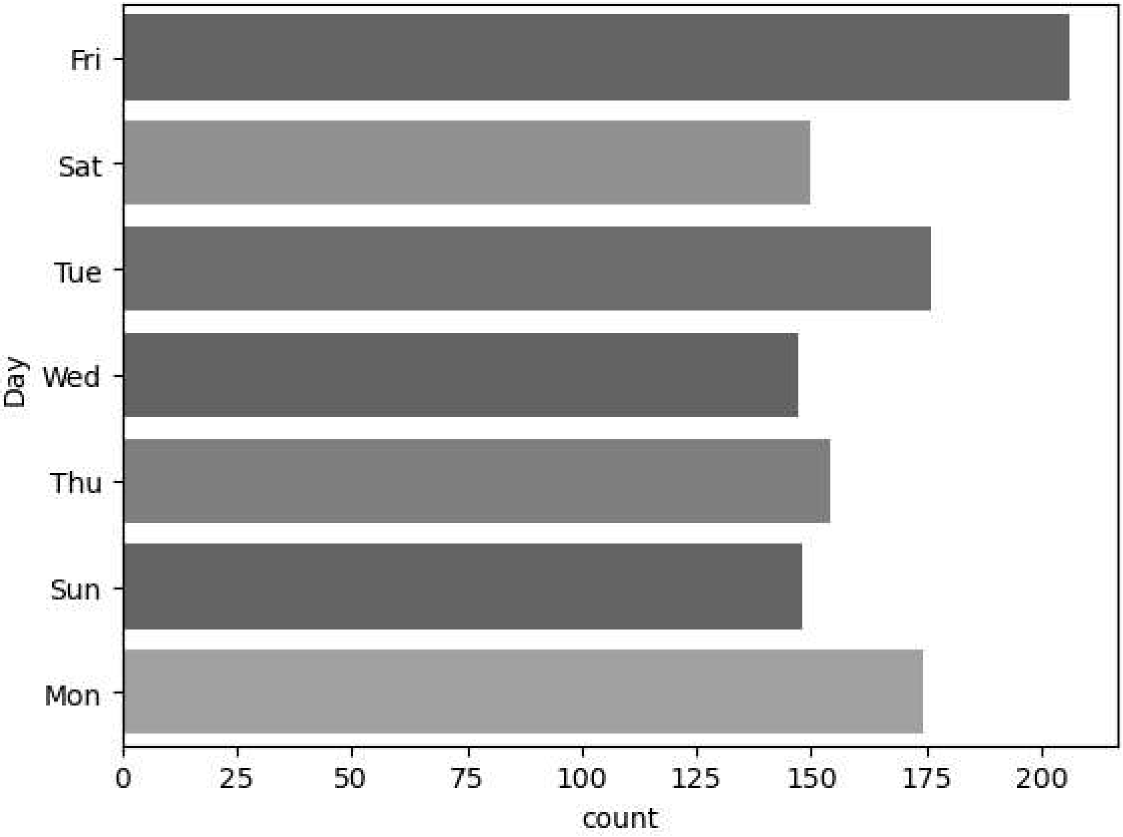
```
db['Day']=db['Day'].map(day_label)
```

In [31]:

```
sns.countplot(y=db['Day'])
```

Out[31]:

```
<Axes: xlabel='count', ylabel='Day'>
```



```
In [32]: db.head()
```

Out[32]:

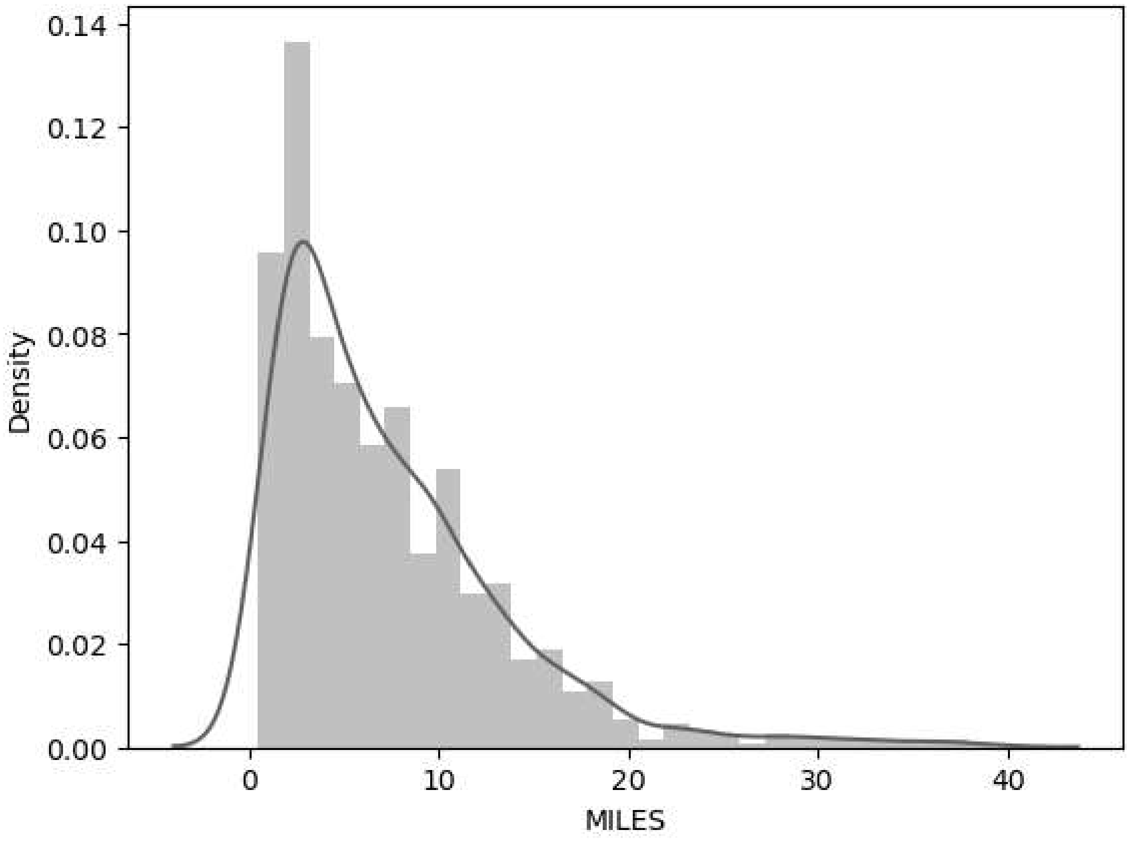
	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	DATE	Time	
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	2016-01-01	21.0	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	Unknown	2016-01-02	1.0	
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	2016-01-02	20.0	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	17.0	
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	2016-01-06	14.0	A

How many miles do people usually book a cab for through uber?

```
In [41]: sns.distplot(db[db['MILES']<40]['MILES'])
```

```
C:\Users\91981\AppData\Local\Temp\ipykernel_12548\497687722.py:1: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

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

```
Out[41]: <Axes: xlabel='MILES', ylabel='Density'>
```



Conclusion

Most Popular Category: Business trips dominate Uber bookings.

Top Purpose for Rides: Meetings are the most common reason for booking Uber rides. (Excluding Unknowns)

Peak Booking Month: December sees the highest number of Uber bookings.

Time of Day Analysis: Rides are categorized into Morning, Afternoon, Evening, and Night, with most rides happening in the evening.

In [ ]: