Uber Data Analysis

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

```
dataset = pd.read_csv('/content/UberDataset.csv')
```

```
dataset.head(10)
```

	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
5	01-06-2016 17:15	01-06-2016 17:19	Business	West Palm Beach	West Palm Beach	4.3	Meal/Entertain
6	01-06-2016 17:30	01-06-2016 17:35	Business	West Palm Beach	Palm Beach	7.1	Meeting
7	01-07-2016 13:27	01-07-2016 13:33	Business	Cary	Cary	0.8	Meeting
8	01-10-2016 08:05	01-10-2016 08:25	Business	Cary	Morrisville	8.3	Meeting
9	01-10-2016 12:17	01-10-2016 12:44	Business	Jamaica	New York	16.5	Customer Visit

Next steps: Generate code with dataset | View recommended plots | New interactive sheet

```
#check column
dataset.columns
```

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

```
#check shape
dataset.shape
```

```
(1156, 7)
```

```
#check info
dataset.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
```

```
#check missiong value in every column
dataset.isnull().sum()
```

	0
START_DATE	0
END_DATE	1
CATEGORY	1
START	1
STOP	1
MILES	0
PURPOSE	503

dtype: int64

Data preprocessing

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

```
dataset.head()
```

	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	NOT
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

Next steps: Generate code with dataset | View recommended plots | New interactive sheet

```
#change the formate of date column
dataset['START_DATE'] = pd.to_datetime(dataset['START_DATE'] , errors='coerce')
dataset['END_DATE'] = pd.to_datetime(dataset['END_DATE'] , errors='coerce')
```

```
dataset.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  421 non-null    datetime64[ns]
 1   END_DATE    420 non-null    datetime64[ns]
 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     1156 non-null   object
dtypes: datetime64[ns](2), float64(1), object(4)
memory usage: 63.3+ KB
```

```
#creating new column for weekday
from datetime import datetime
dataset['date'] = pd.DatetimeIndex(dataset['START_DATE']).date
dataset['time'] = pd.DatetimeIndex(dataset['START_DATE']).hour
```

```
dataset.head()
```

	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	NOT	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

Next steps: Generate code with dataset | View recommended plots | New interactive sheet

```
#create a new column for weekday
dataset['day night'] = pd.cut(x=dataset['time'], bins = [0,10,15,19,24], labels=['Morning','Afternoon','Evening','Night'])
```

```
dataset.head()
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	time	day night
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	Night
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	1.0	Morning
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	Night
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	Evening
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	Afternoon

Next steps: Generate code with dataset | View recommended plots | New interactive sheet

```
#remove nulls
dataset.dropna(inplace=True)
```

```
#check null
dataset.isnull().sum()
```

	0
START_DATE	0
END_DATE	0
CATEGORY	0
START	0
STOP	0
MILES	0
PURPOSE	0
date	0
time	0
day night	0

dtype: int64

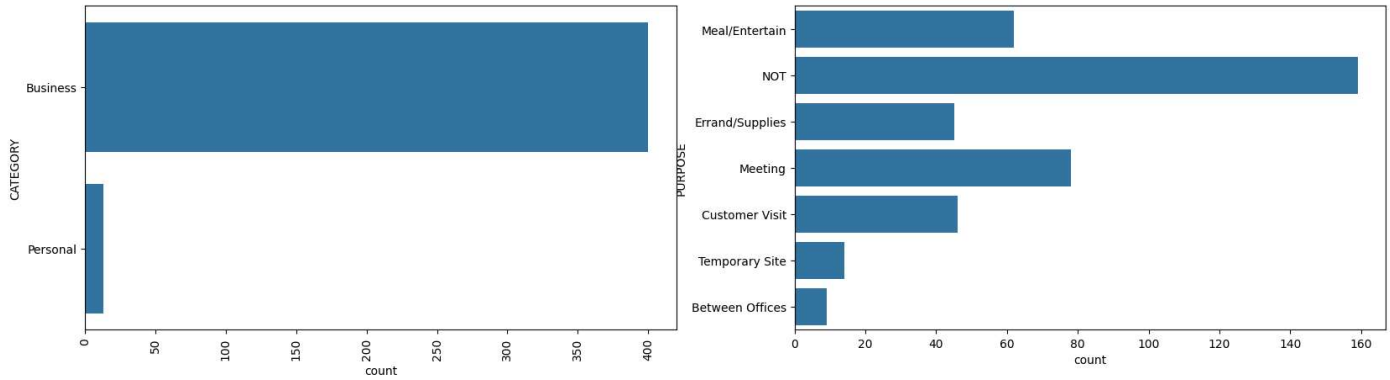
Data Visualizations

Questions to Answer

1. In which category do people book the most Uber rides?
2. For which purpose do people book Uber rides the most?
3. At what time do people book cabs the most from Uber?
4. In which months do people book Uber rides less frequently?
5. On which days of the week do people book Uber rides the most?
6. How many miles do people usually book a cab for through Uber?

```
plt.figure(figsize=(20,5))
plt.subplot(1,2,1)
sns.countplot(dataset['CATEGORY'])
plt.xticks(rotation=90)
plt.subplot(1,2,2)
sns.countplot(dataset['PURPOSE'])
```

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



1. In which category do people book the most Uber rides?

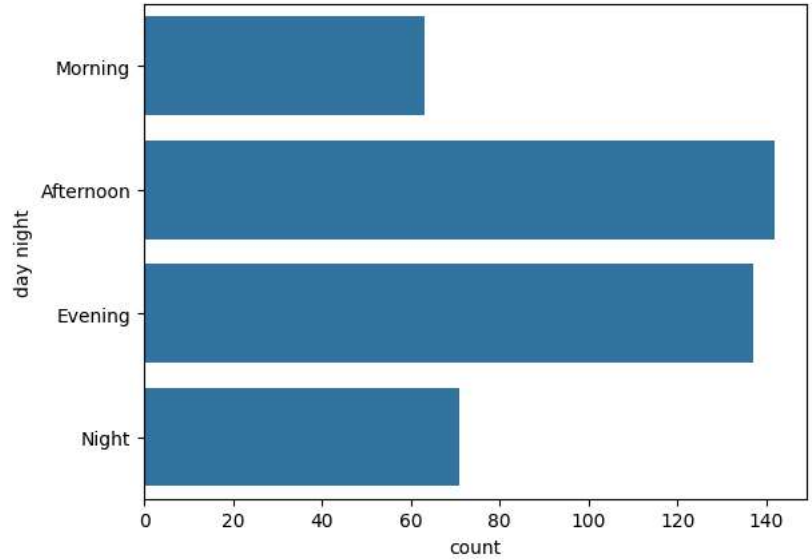
Answer: the most popular category do people book the most is 'Business'

2. For which purpose do people book Uber rides the most?

Answer: the most popular purpose for booking Uber rides is 'Meeting'

```
sns.countplot(dataset['day night'])
```

```
<Axes: xlabel='count', ylabel='day night'>
```



3. At what time do people book cabs the most from Uber?

Answer : uber mostly use in afternoon time

```
dataset.head()
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	time	day night
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	Night
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	1.0	Morning
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	Night
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	Evening
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	Afternoon

Next steps: Generate code with dataset | View recommended plots | New interactive sheet

```
dataset['MONTH'] = pd.DatetimeIndex(dataset['START_DATE']).month
month_label = {1:'Jan', 2:'Feb', 3:'Mar', 4:'Apr',
               5:'May', 6:'Jun', 7:'Jul', 8:'Aug',
               9:'Sep', 10:'Oct', 11:'Nov', 12:'Dec'}
dataset['MONTH']  = dataset['MONTH'].map(month_label)
mon = dataset.MONTH.value_counts(sort=False)
```

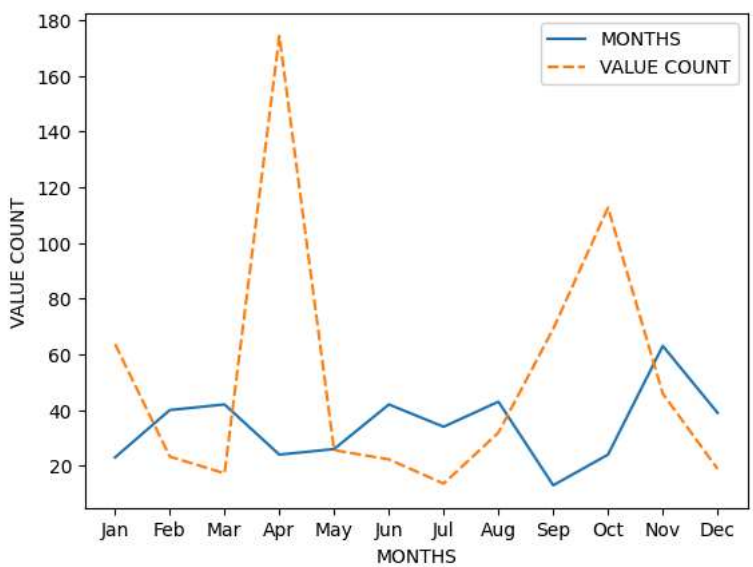
```
dataset.head()
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	time	day night	MONTH
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	Night	Jan
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	1.0	Morning	Jan
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	Night	Jan

Next steps: Generate code with dataset | View recommended plots | New interactive sheet

```
df = pd.DataFrame({
"MONTHS": mon.values, # Har month ka total count.
"VALUE COUNT": dataset.groupby('MONTH', sort=False)['MILES'].max() # Har mo
})
p = sns.lineplot(data=df) # Line plot banata hai.
p.set(xlabel="MONTHS", ylabel="VALUE COUNT") # Axis labels set karta ha
```

```
[Text(0.5, 0, 'MONTHS'), Text(0, 0.5, 'VALUE COUNT')]
```



4. In which months do people book Uber rides less frequently?

Ans: People book Uber rides less frequently in the month of January, February, November, December.

```
dataset['DAY'] = dataset.START_DATE.dt.weekday
day_label = {
```

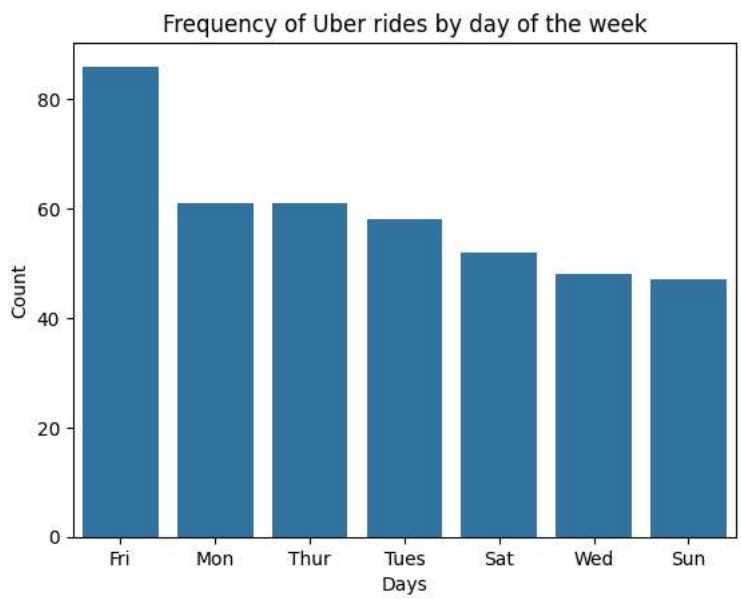
```
0: 'Mon', 1:'Tues', 2:'Wed', 3:'Thur',4:'Fri', 5:'Sat', 6:'Sun'}
dataset['DAY'] = dataset['DAY'].map(day_label)
```

```
dataset.head()
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	time	day night	MONTH	DAY
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	Night	Jan	Fri
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	1.0	Morning	Jan	Sat
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	Night	Jan	Sat

Next steps: Generate code with dataset | View recommended plots | New interactive sheet

```
dat_label = dataset.DAY.value_counts()
sns.barplot(x=dat_label.index, y=dat_label)
plt.xlabel('Days')
plt.ylabel('Count')
plt.title('Frequency of Uber rides by day of the week')
plt.show()
```

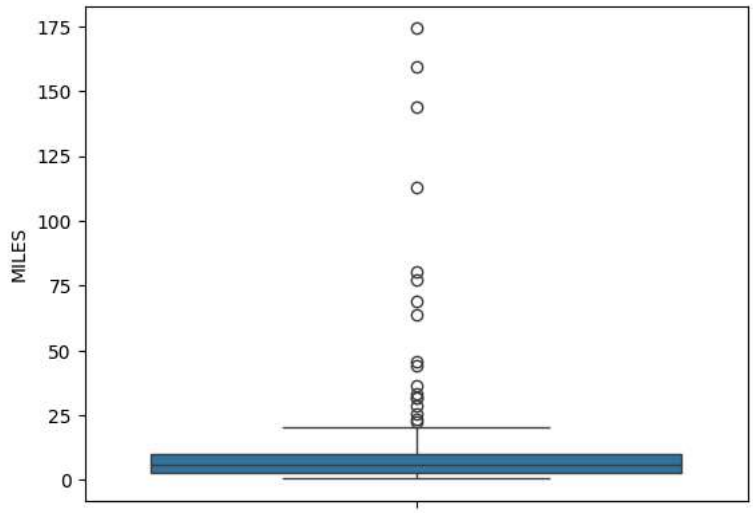


5. On which days of the week do people book Uber rides the most?

Ans: People book Uber rides the most on Fridays and Monday.

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

```
<Axes: ylabel='MILES'>
```

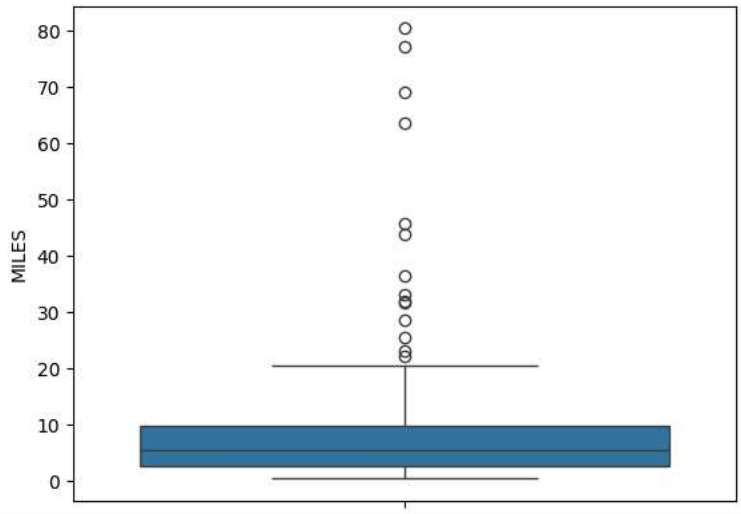



```
#check for 100miles
dataset[dataset['MILES'] > 100]
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	time	day night	MONTH	DAY
297	2016-04-02 19:38:00	2016-04-02 22:36:00	Business	Jacksonville	Ridgeland	174.2	Customer Visit	2016-04-02	19.0	Evening	Apr	Sat
298	2016-04-02 23:11:00	2016-04-03 01:34:00	Business	Ridgeland	Florence	144.0	Meeting	2016-04-02	23.0	Night	Apr	Sat

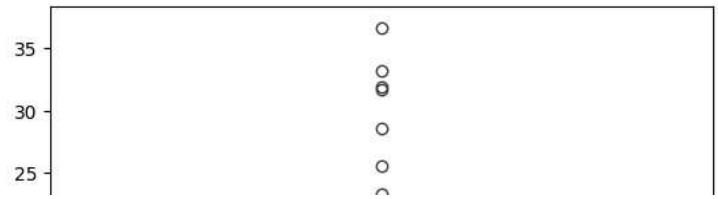
```
sns.boxplot(dataset['MILES'][dataset['MILES'] < 100])
```

```
<Axes: ylabel='MILES'>
```




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

```
<Axes: ylabel='MILES'>
```




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

```
<ipython-input-72-9e249e6bc7e9>:1: UserWarning:
`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

  sns.distplot(dataset['MILES'][dataset['MILES'] < 40])
<Axes: xlabel='MILES', ylabel='Density'>
```