In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
dataset = pd.read_csv("UberDataset.csv")
```

In [6]:
```
dataset
```

Out[6]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPO
0	01-01-2016 21:11	01-01-2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entert
1	01-02-2016 01:25	01-02-2016 01:37	Business	Fort Pierce	Fort Pierce	5.0	N
2	01-02-2016 20:25	01-02-2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Suppl
3	01-05-2016 17:31	01-05-2016 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeti
4	01-06-2016 14:42	01-06-2016 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer V
...	.....	.....	.....	.....	.....	.....	.....
1151	12/31/2016 13:24	12/31/2016 13:42	Business	Kar?chi	Unknown Location	3.9	Temporary S
1152	12/31/2016 15:03	12/31/2016 15:38	Business	Unknown Location	Unknown Location	16.2	Meeti
1153	12/31/2016 21:32	12/31/2016 21:50	Business	Katunayake	Gampaha	6.4	Temporary S
1154	12/31/2016 22:08	12/31/2016 23:51	Business	Gampaha	Ilukwatta	48.2	Temporary S
1155	Totals	NaN	NaN	NaN	NaN	12204.7	N

1156 rows × 7 columns

In [8]:
```
dataset.shape
```

Out[8]:
```
(1156, 7)
```

In [10]:
```
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
```

Data Preprocessing

In [15]:
```
dataset['PURPOSE'].fillna("NOT", inplace = True)
```

```
C:\Users\swati\AppData\Local\Temp\ipykernel_31136\4083644620.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.meth
od({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to pe
rform the operation inplace on the original object.


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

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

Out[17]:

	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

In [19]:
```
dataset['START_DATE'] = pd.to_datetime(dataset['START_DATE'], errors = 'coerce')

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

In [21]:
```
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
```

In [23]:
```
from datetime import datetime

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

In [25]:
```
dataset.head()
```

Out[25]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	t
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	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	
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	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	
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	

In [27]:
```
dataset['day-night'] = pd.cut(x=dataset['time'],bins = [0,10,15,19,24],labels =
```

In [29]:
```
dataset.head()
```

Out[29]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	t
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	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	
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	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	
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	

In [33]:
```
dataset.dropna(inplace = True)
```

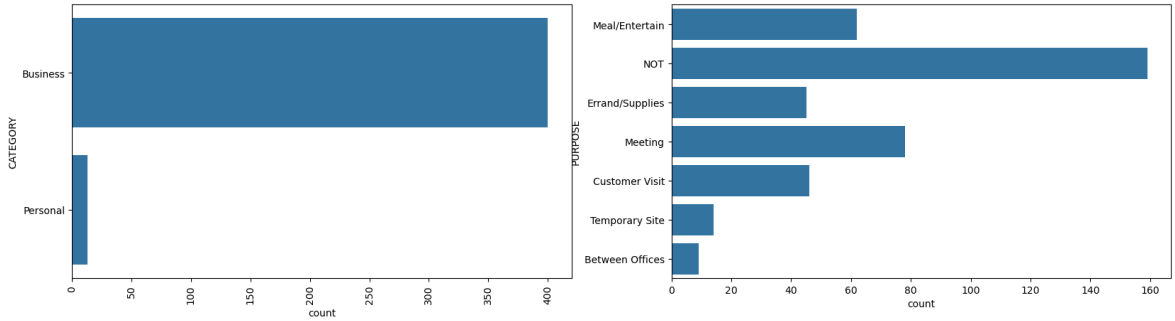
In [35]:
```
dataset.shape
```

Out[35]: (413, 10)

Data Visualization

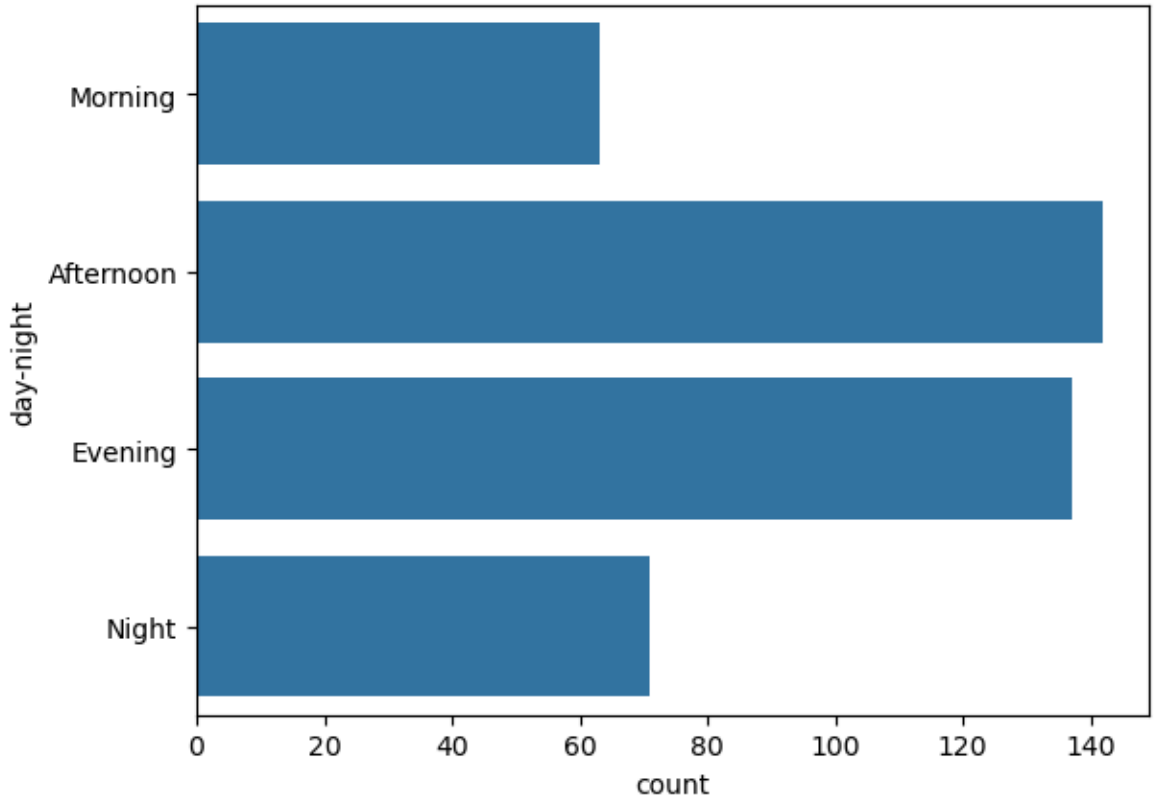
In [46]:
```
plt.figure(figsize=(20,5))

plt.subplot(1,2,1)

sns.countplot(dataset['CATEGORY'])
plt.xticks(rotation =90)

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

Out[46]: <Axes: xlabel='count', ylabel='PURPOSE'>



In [48]:
```
sns.countplot(dataset['day-night'])
```

Out[48]: <Axes: xlabel='count', ylabel='day-night'>



In [50]:
```
dataset.head()
```

Out[50]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	t
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	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	
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	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	
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	

In [52]:
```
dataset['MONTH'] = pd.DatetimeIndex(dataset['START_DATE']).month# START_DATE se

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'} # Months ko st
```

```
dataset["MONTH"] = dataset.MONTH.map(month_label) # Number months ko string name

mon = dataset.MONTH.value_counts(sort=False)  # Har month ke counts calculate ka
```

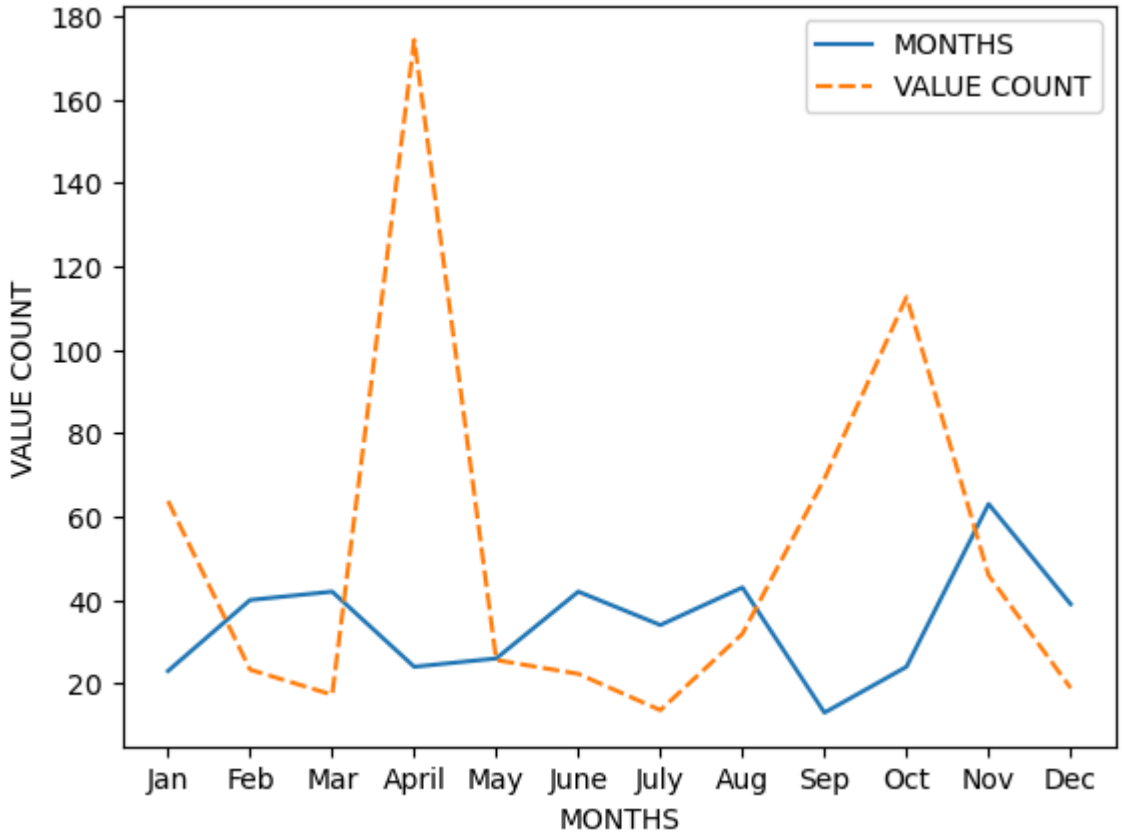
In [54]:
```
dataset.head()
```

Out[54]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	t
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	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	
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	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	
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	

In [58]:
```
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
```

Out[58]:
```
[Text(0.5, 0, 'MONTHS'), Text(0, 0.5, 'VALUE COUNT')]
```



In [60]:
```
dataset.head()
```

Out[60]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	t
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	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	
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	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	
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	

In [64]:
```
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)
```

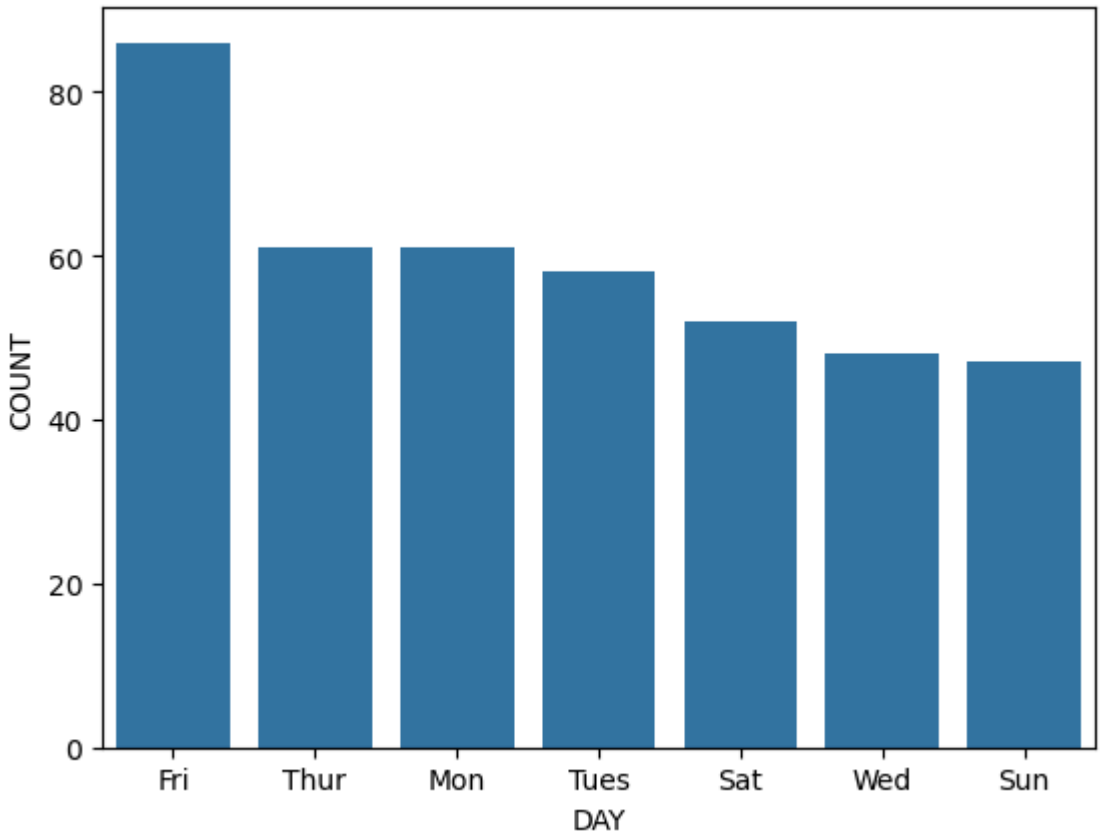
In [66]:
```
dataset.head()
```

Out[66]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	t
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	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	
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	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	
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	

In [68]:
```
day_label =dataset.DAY.value_counts()

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

Out[68]:
```
Text(0, 0.5, 'COUNT')
```



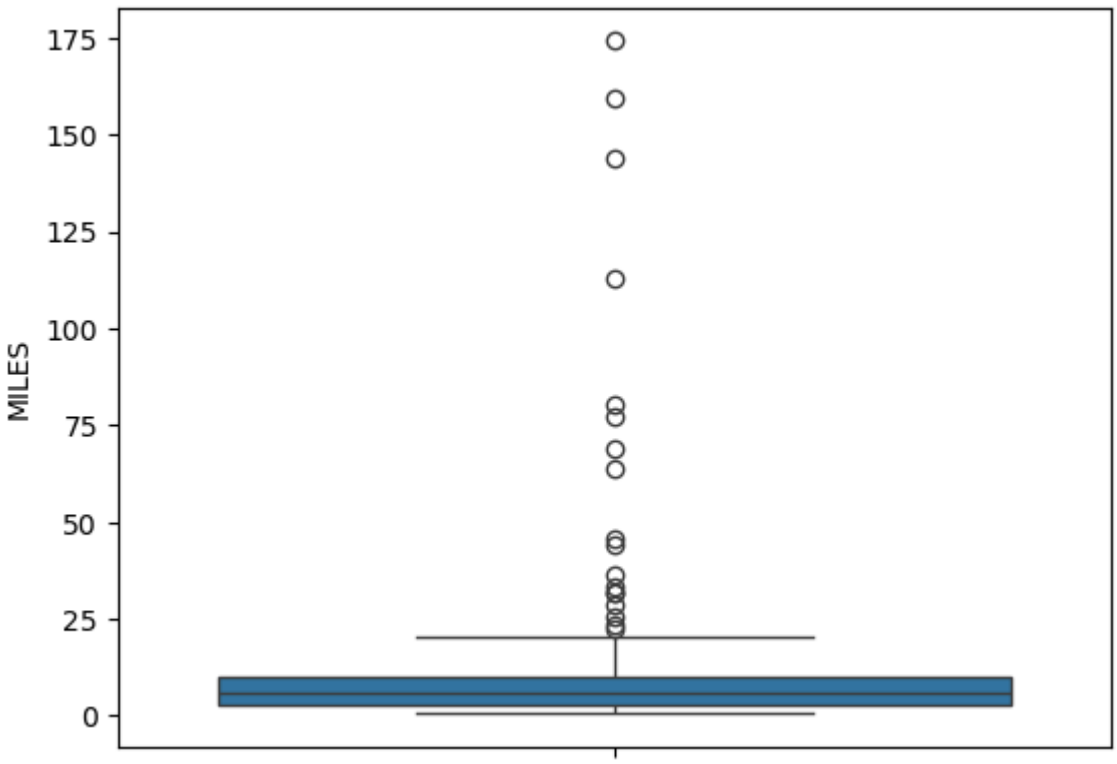
In [70]:
```
dataset.head()
```

Out[70]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	date	t
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	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NOT	2016-01-02	
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	
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	2016-01-05	
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	

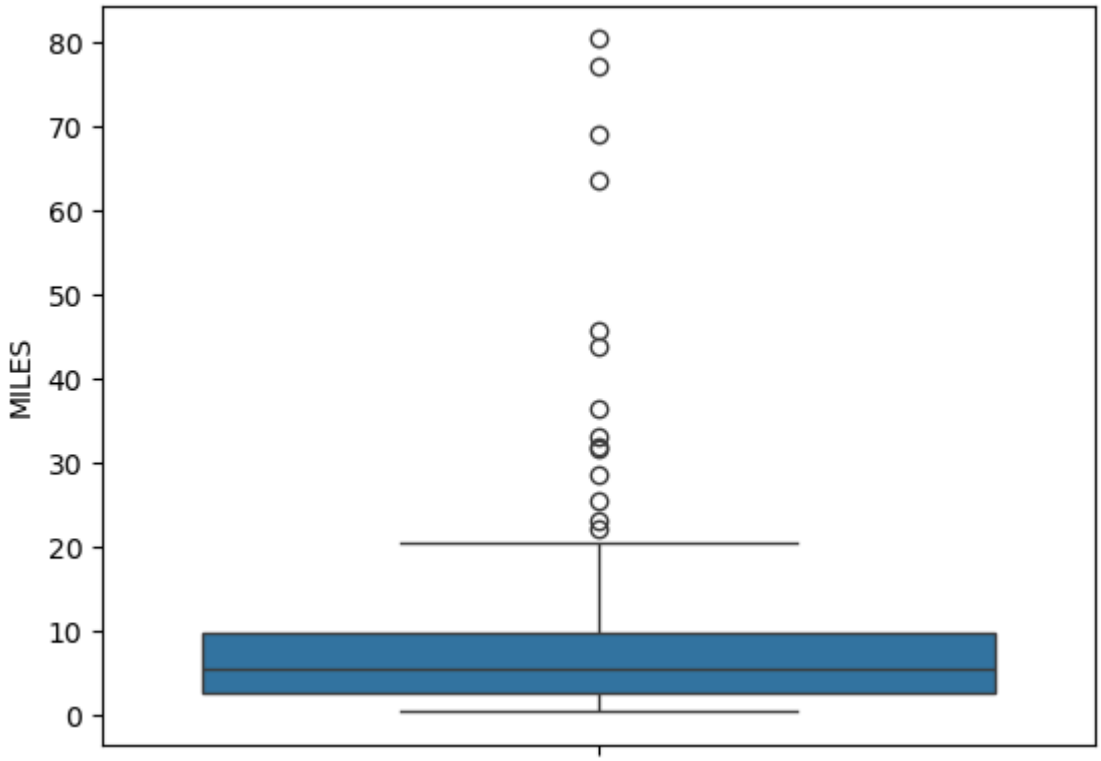
In [74]:
```
sns.boxplot(dataset['MILES'])
```

Out[74]:
```
<Axes: ylabel='MILES'>
```



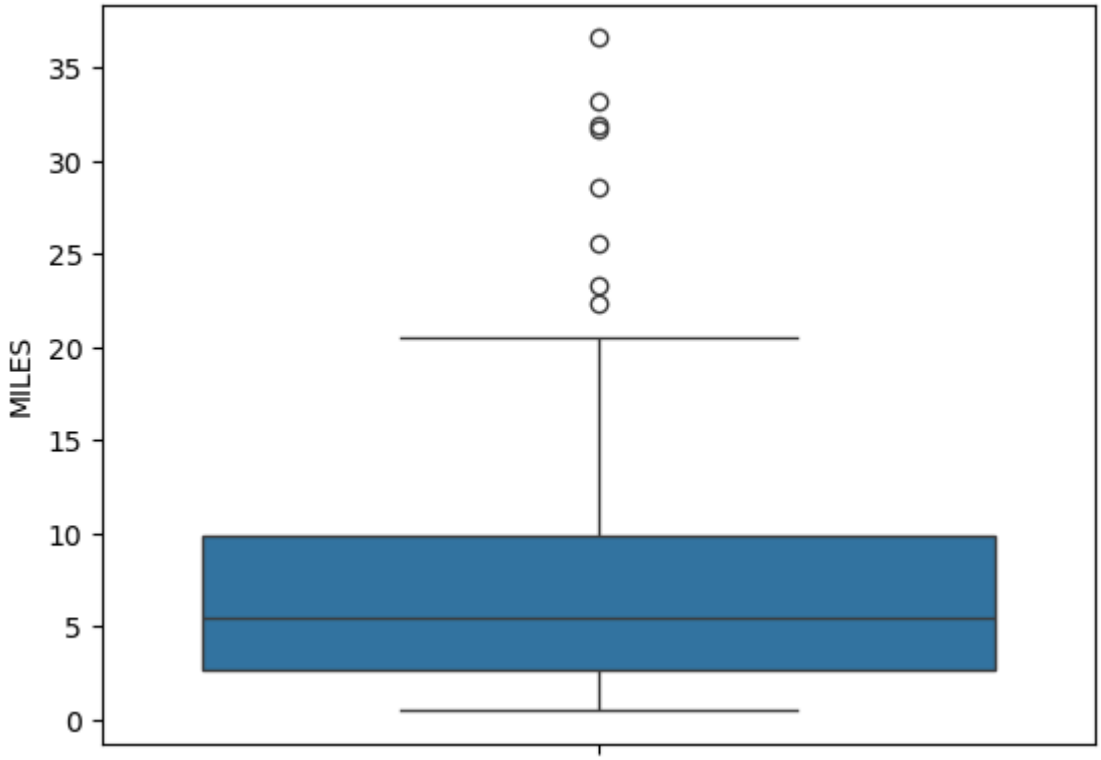
In [78]:
```
sns.boxplot(dataset[dataset['MILES']<100]['MILES'])
```

Out[78]:
```
<Axes: ylabel='MILES'>
```



In [82]:
```
sns.boxplot(dataset[dataset['MILES']<40]['MILES'])
```

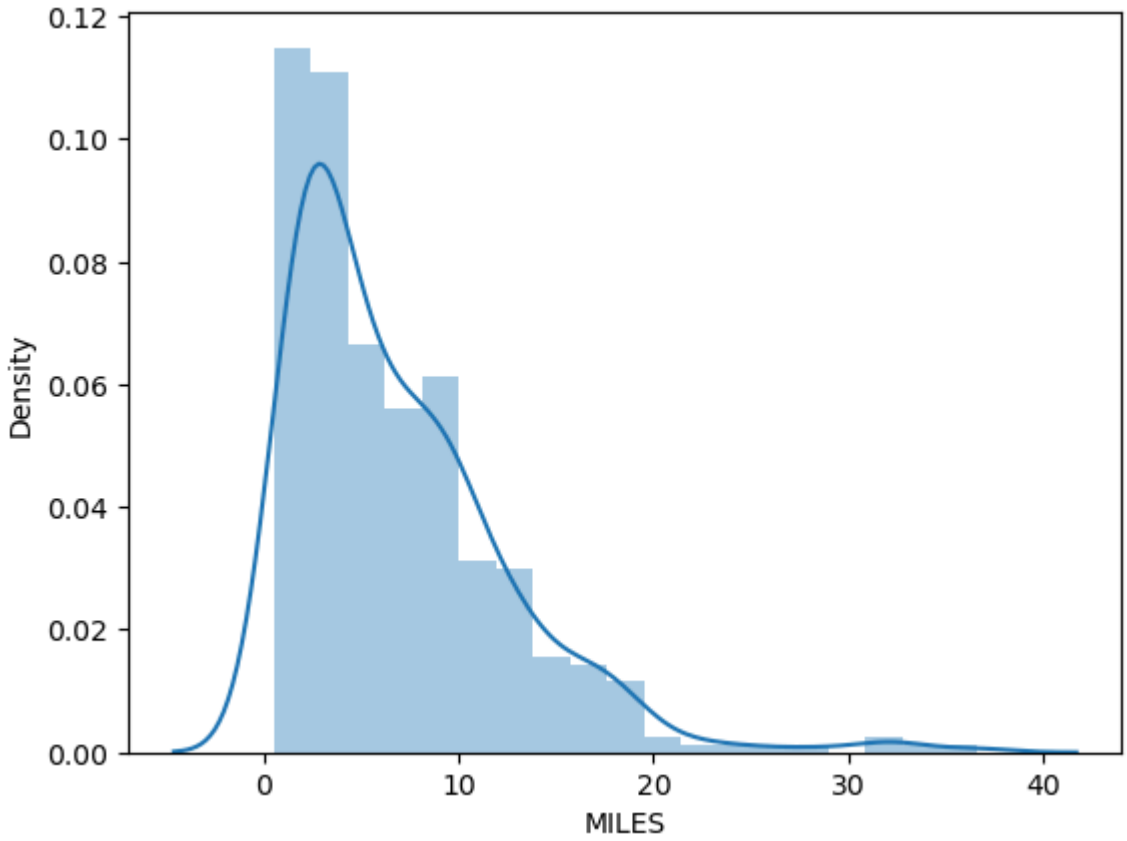
Out[82]: <Axes: ylabel='MILES'>



In [86]:
```
sns.distplot(dataset[dataset['MILES']<40]['MILES'])
```

```
C:\Users\swati\AppData\Local\Temp\ipykernel_31136\1678554178.py: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[dataset['MILES']<40]['MILES'])
```

Out[86]: <Axes: xlabel='MILES', ylabel='Density'>



In [ ]: