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

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

```
        START_DATE          END_DATE  CATEGORY        START
STOP  \
0  01-01-2016 21:11  01-01-2016 21:17  Business  Fort Pierce        Fort
Pierce
1  01-02-2016 01:25  01-02-2016 01:37  Business  Fort Pierce        Fort
Pierce
2  01-02-2016 20:25  01-02-2016 20:38  Business  Fort Pierce        Fort
Pierce
3  01-05-2016 17:31  01-05-2016 17:45  Business  Fort Pierce        Fort
Pierce
4  01-06-2016 14:42  01-06-2016 15:49  Business  Fort Pierce  West
Palm Beach

   MILES          PURPOSE
0    5.1   Meal/Entertain
1    5.0              NaN
2    4.8  Errand/Supplies
3    4.7          Meeting
4   63.7   Customer Visit
```

```python
dataset.shape
```

```
(1156, 7)
```

```python
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
```

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

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

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'])

dataset.dropna(inplace=True)

dataset.drop_duplicates(inplace=True)

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
```

```
{'CATEGORY': 1, 'START': 69, 'STOP': 74, 'PURPOSE': 6, 'date': 85}
```

```python
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)
```
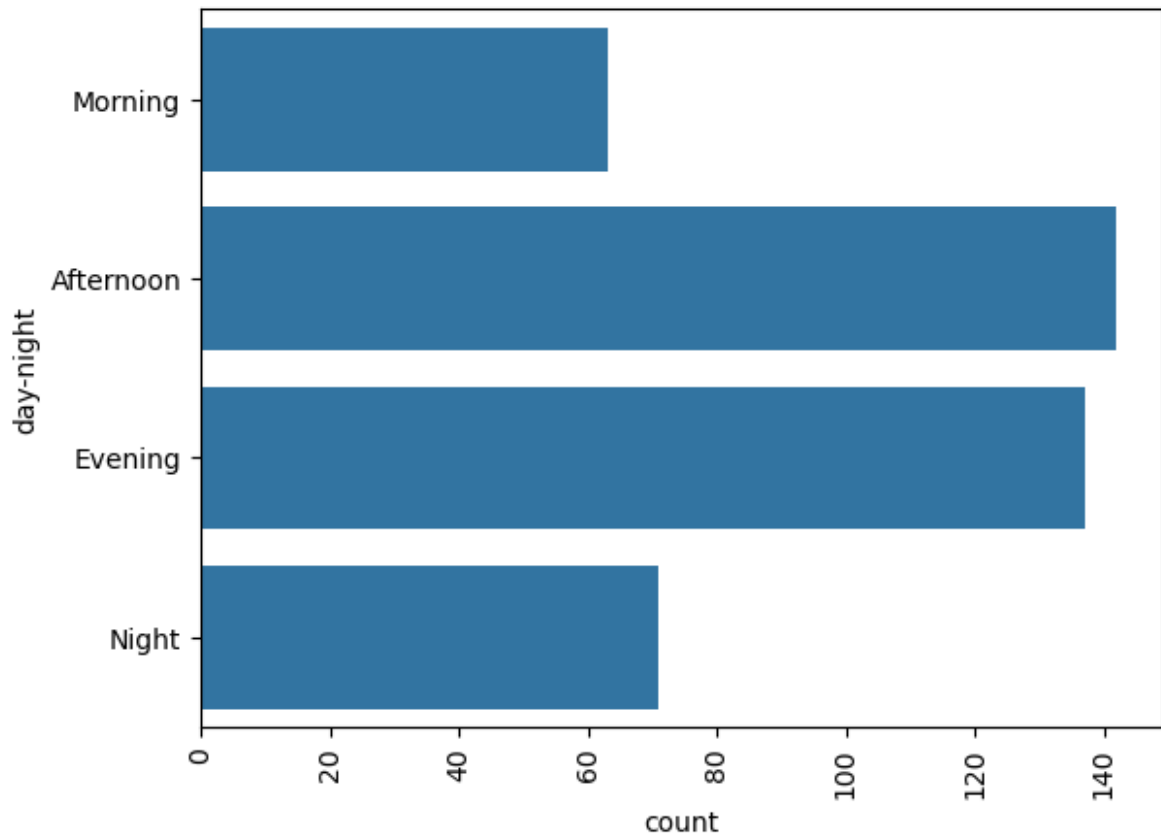
```
(array([  0.,  20.,  40.,  60.,  80., 100.]),
 [Text(0.0, 0, '0'),
  Text(20.0, 0, '20'),
  Text(40.0, 0, '40'),
  Text(60.0, 0, '60'),
  Text(80.0, 0, '80'),
  Text(100.0, 0, '100')])
```
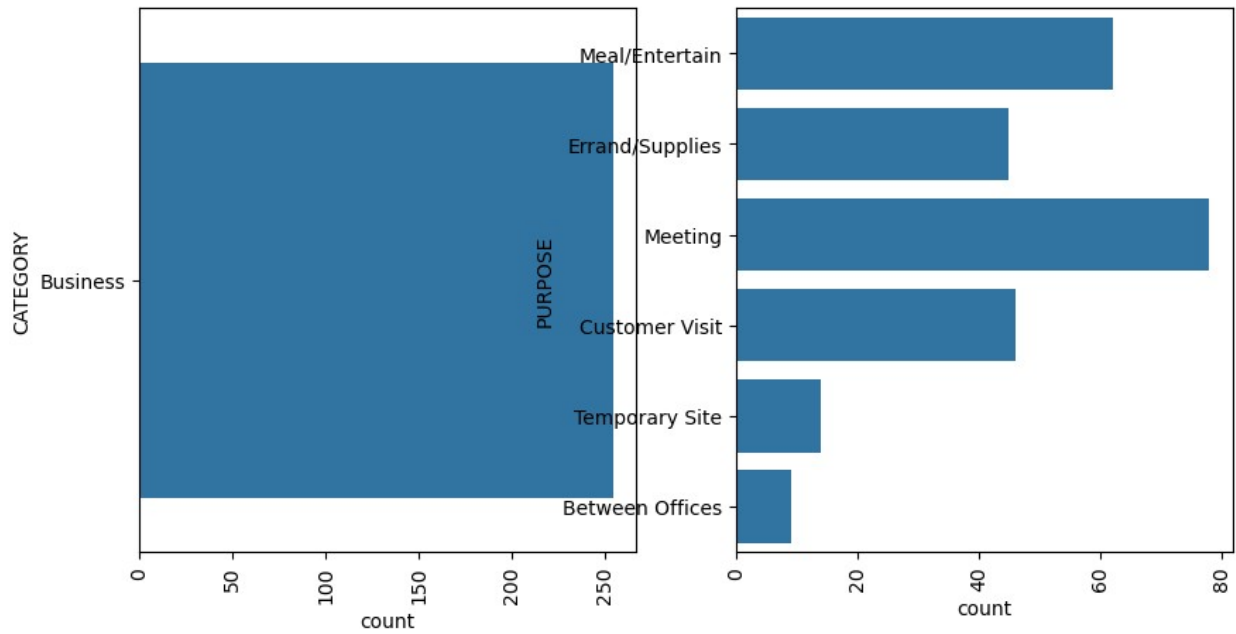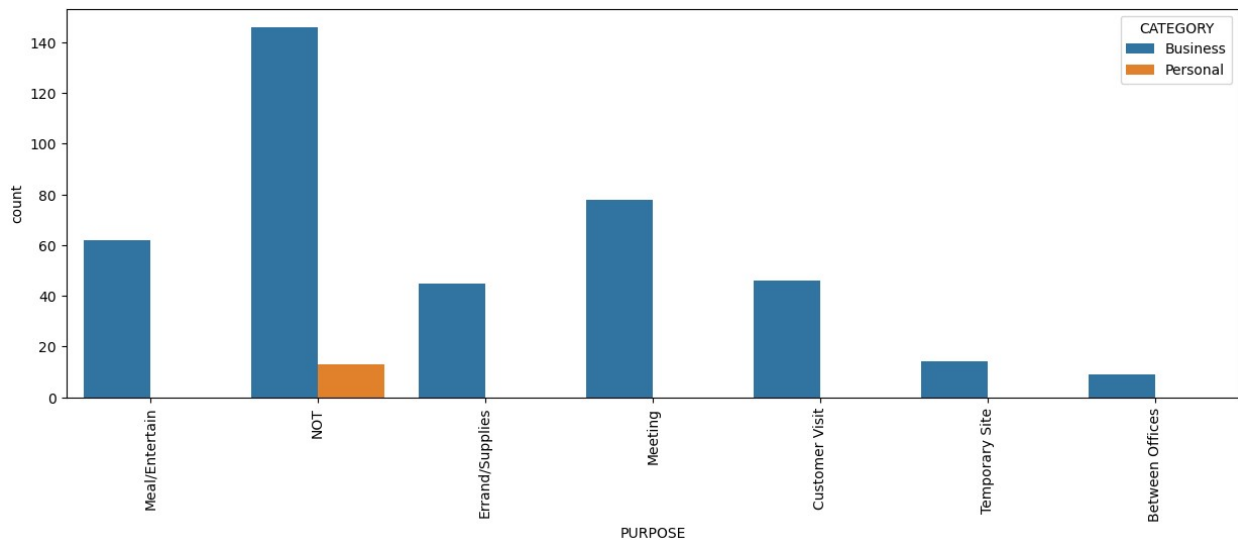
```
sns.countplot(dataset['day-night'])
plt.xticks(rotation=90)

(array([  0.,  20.,  40.,  60.,  80., 100., 120., 140., 160.]),
 [Text(0.0, 0, '0'),
  Text(20.0, 0, '20'),
  Text(40.0, 0, '40'),
  Text(60.0, 0, '60'),
  Text(80.0, 0, '80'),
  Text(100.0, 0, '100'),
  Text(120.0, 0, '120'),
  Text(140.0, 0, '140'),
  Text(160.0, 0, '160')])
```

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

```python
# 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)

# This code is modified by Susobhan Akhuli
```

```
<Axes: >
```



```python
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")

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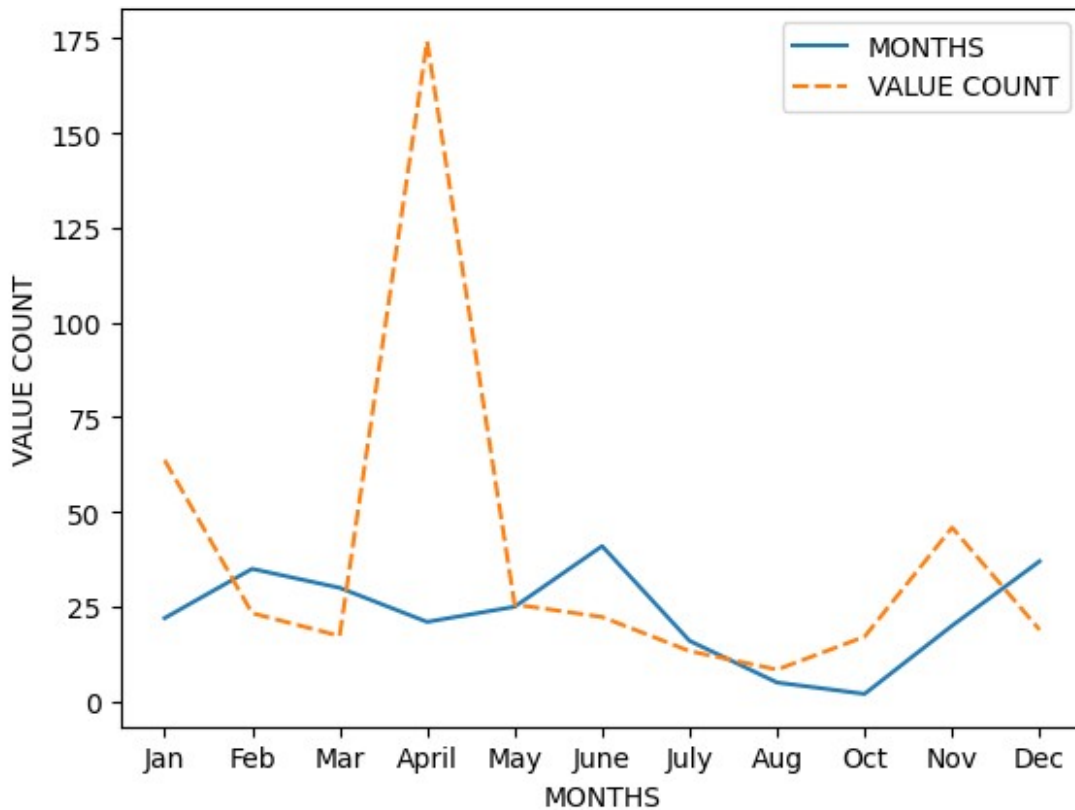


```
dataset['DAY'] = dataset.START_DATE.dt.weekday
day_label = {
    0: 'Mon', 1: 'Tues', 2: 'Wed', 3: 'Thus', 4: 'Fri', 5: 'Sat', 6:
'Sun'
}
dataset['DAY'] = dataset['DAY'].map(day_label)

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

Text(0, 0.5, 'COUNT')
```
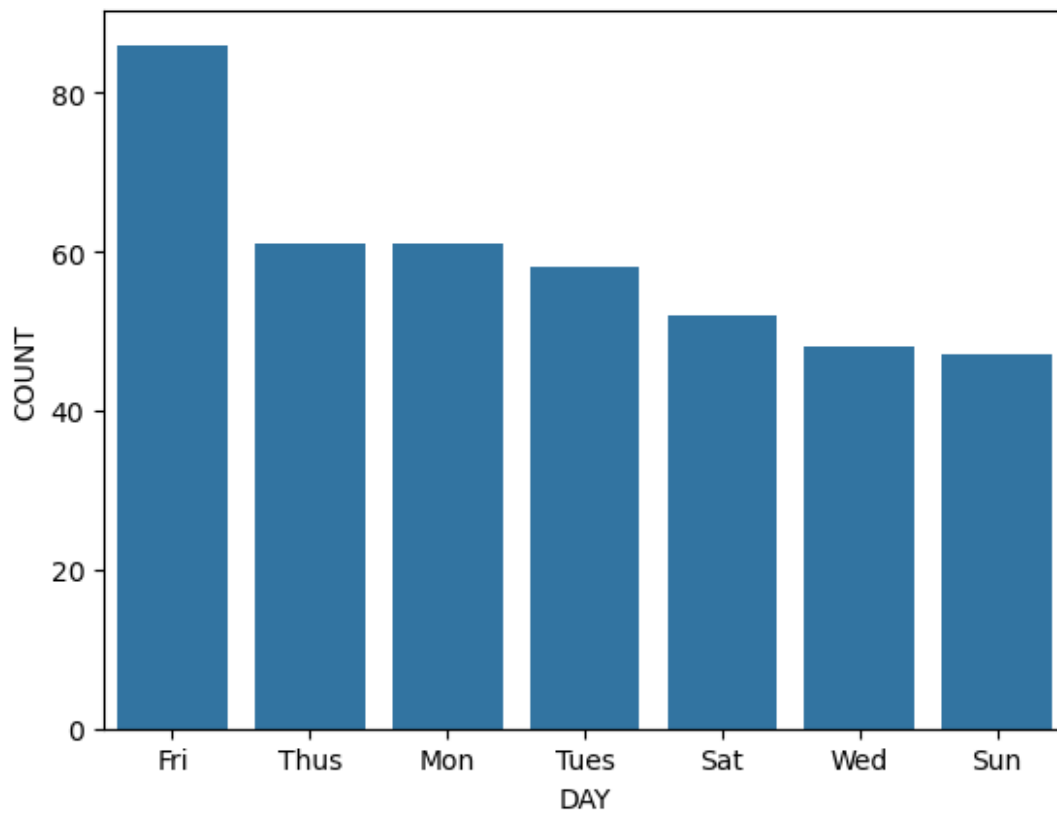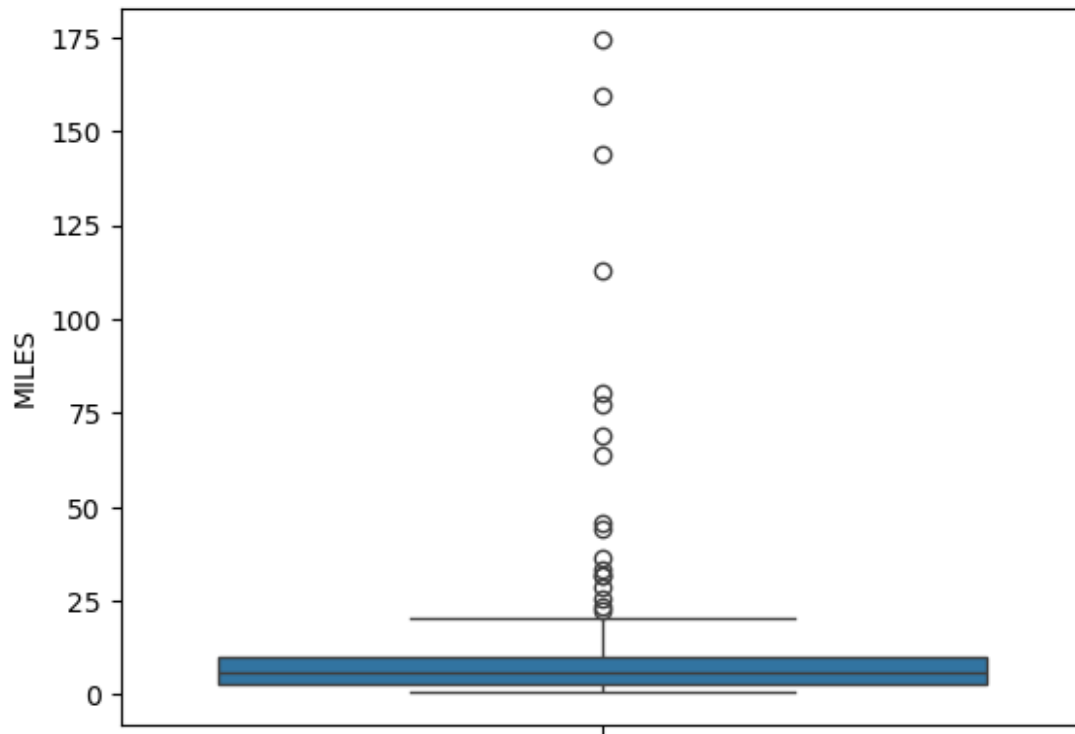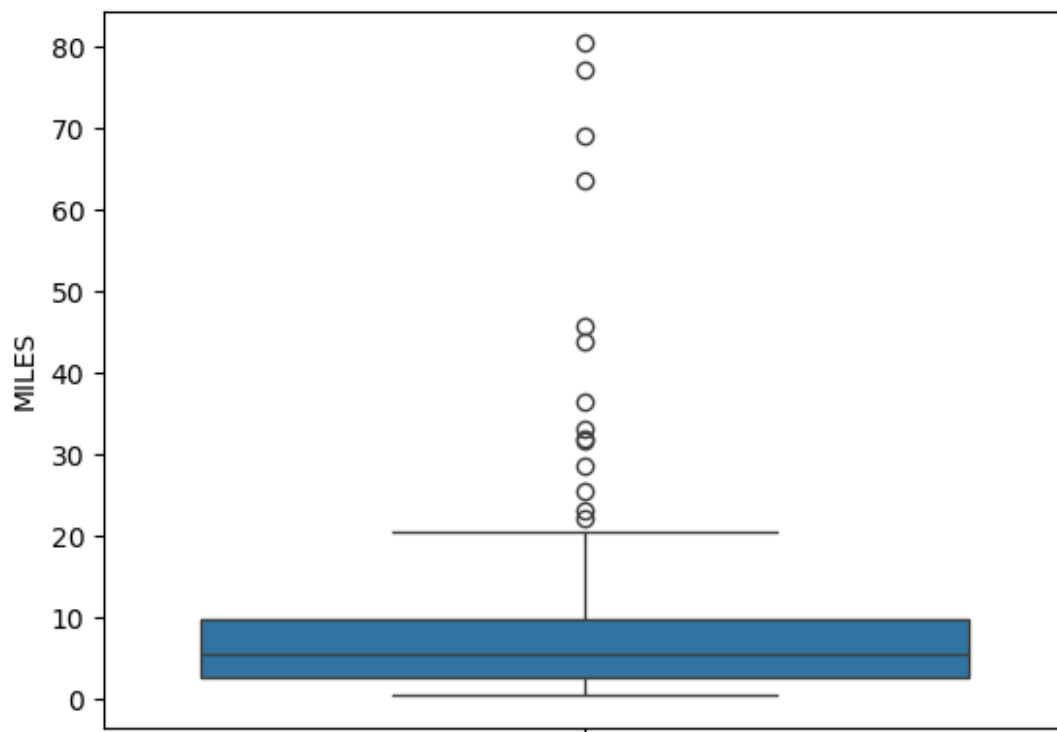
```
sns.boxplot(dataset['MILES'])
```

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

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

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

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

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_10632\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'])
```

```
<Axes: xlabel='MILES', ylabel='Density'>
```