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

In [3]:
```
df=pd.read_csv('UberDataset - UberDataset.csv')
df
```

Out[3]:

	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 1:25	01-02-2016 1:37	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 [4]:
```
(r,c)=df.shape
print("rows:-",r)
print("columns:-",c)
```

```
rows:- 1156
columns:- 7
```

In [5]:
```
df.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 [6]:
```
df.columns
```

Out[6]:
```
Index(['START_DATE', 'END_DATE', 'CATEGORY', 'START', 'STOP', 'MILES',
       'PURPOSE'],
      dtype='object')
```

In [7]:
```
df.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max
MILES	1156.0	21.115398	359.299007	0.5	2.9	6.0	10.4	12204.7

In [8]:
```
df.dtypes
```

Out[8]:
```
START_DATE     object
END_DATE       object
CATEGORY       object
START          object
STOP           object
MILES         float64
PURPOSE        object
dtype: object
```

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

Out[9]:
```
START_DATE      0
END_DATE        1
CATEGORY        1
START           1
STOP            1
MILES           0
PURPOSE       503
dtype: int64
```

In [10]:
```
df['PURPOSE'].fillna("Not.",inplace=True)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\2373822972.py:1: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assi
gnment 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.


  df['PURPOSE'].fillna("Not.",inplace=True)
```

In [11]:
```
df
```

Out[11]:

	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 1:25	01-02-2016 1:37	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 [12]:
```
df['START_DATE']=pd.to_datetime(df['START_DATE'],errors='coerce')
df['END_DATE']=pd.to_datetime(df['END_DATE'],errors='coerce')
df
```

Out[12]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPO
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Enterta
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	N
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Suppl
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeti
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Vi
...	...	...	...	...	...	...	
1151	NaT	NaT	Business	Kar?chi	Unknown Location	3.9	Temporary S
1152	NaT	NaT	Business	Unknown Location	Unknown Location	16.2	Meeti
1153	NaT	NaT	Business	Katunayake	Gampaha	6.4	Temporary S
1154	NaT	NaT	Business	Gampaha	Ilukwatta	48.2	Temporary S
1155	NaT	NaT	NaN	NaN	NaN	12204.7	N

1156 rows × 7 columns

In [13]:
```
df['START_DATE'].isnull().sum()
```

Out[13]:
```
np.int64(735)
```

In [14]:
```
df['START_DATE'].fillna(method='ffill',inplace=True)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\3100504763.py:1: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assi
gnment 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.


  df['START_DATE'].fillna(method='ffill',inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\3100504763.py:1: FutureWarning: Se
ries.fillna with 'method' is deprecated and will raise in a future version. Use o
bj.ffill() or obj.bfill() instead.
  df['START_DATE'].fillna(method='ffill',inplace=True)
```

In [15]:
```
df['END_DATE'].isnull().sum()
```

Out[15]:
```
np.int64(736)
```

In [16]:
```
df['END_DATE'].fillna(method='ffill',inplace=True)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\4058995039.py:1: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assi
gnment 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.


  df['END_DATE'].fillna(method='ffill',inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\4058995039.py:1: FutureWarning: Se
ries.fillna with 'method' is deprecated and will raise in a future version. Use o
bj.ffill() or obj.bfill() instead.
  df['END_DATE'].fillna(method='ffill',inplace=True)
```

In [17]:
```
df['CATEGORY'].fillna(method='ffill',inplace=True)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\2029814591.py:1: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assi
gnment 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.


  df['CATEGORY'].fillna(method='ffill',inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\2029814591.py:1: FutureWarning: Se
ries.fillna with 'method' is deprecated and will raise in a future version. Use o
bj.ffill() or obj.bfill() instead.
  df['CATEGORY'].fillna(method='ffill',inplace=True)
```

In [18]:
```
df['START'].fillna(method='ffill',inplace=True)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\582321262.py:1: FutureWarning: A v
alue is trying to be set on a copy of a DataFrame or Series through chained assig
nment 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.


  df['START'].fillna(method='ffill',inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\582321262.py:1: FutureWarning: Ser
ies.fillna with 'method' is deprecated and will raise in a future version. Use ob
j.ffill() or obj.bfill() instead.
  df['START'].fillna(method='ffill',inplace=True)
```

In [19]:
```
df['STOP'].fillna(method='ffill',inplace=True)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\2867485326.py:1: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assi
gnment 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.


  df['STOP'].fillna(method='ffill',inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\2867485326.py:1: FutureWarning: Se
ries.fillna with 'method' is deprecated and will raise in a future version. Use o
bj.ffill() or obj.bfill() instead.
  df['STOP'].fillna(method='ffill',inplace=True)
```

In [20]:
```
df['MILES'].fillna(method='ffill',inplace=True)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\3362399603.py:1: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assi
gnment 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.


  df['MILES'].fillna(method='ffill',inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\3362399603.py:1: FutureWarning: Se
ries.fillna with 'method' is deprecated and will raise in a future version. Use o
bj.ffill() or obj.bfill() instead.
  df['MILES'].fillna(method='ffill',inplace=True)
```

In [21]:
```
df
```

Out[21]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPO
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Enterta
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	N
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Suppl
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeti
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Vi
...	...	...	...	...	...	...	
1151	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Kar?chi	Unknown Location	3.9	Temporary S
1152	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Unknown Location	Unknown Location	16.2	Meeti
1153	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Katunayake	Gampaha	6.4	Temporary S
1154	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Gampaha	Ilukwatta	48.2	Temporary S
1155	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Gampaha	Ilukwatta	12204.7	N

1156 rows × 7 columns

In [22]:
```
df.dtypes
```

Out[22]:
```
START_DATE    datetime64[ns]
END_DATE      datetime64[ns]
CATEGORY              object
START                 object
STOP                  object
MILES                float64
PURPOSE               object
dtype: object
```

In [23]:
```
df['Date']=pd.DatetimeIndex(df['START_DATE']).date
```

In [24]:
```
df.head(10)
```

Out[24]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	Da
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	201 01-
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	Not.	201 01-
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	201 01-
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	201 01-
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	201 01-
5	2016-01-06 17:15:00	2016-01-06 17:19:00	Business	West Palm Beach	West Palm Beach	4.3	Meal/Entertain	201 01-
6	2016-01-06 17:30:00	2016-01-06 17:35:00	Business	West Palm Beach	Palm Beach	7.1	Meeting	201 01-
7	2016-01-07 13:27:00	2016-01-07 13:33:00	Business	Cary	Cary	0.8	Meeting	201 01-
8	2016-01-10 08:05:00	2016-01-10 08:25:00	Business	Cary	Morrisville	8.3	Meeting	201 01-
9	2016-01-10 12:17:00	2016-01-10 12:44:00	Business	Jamaica	New York	16.5	Customer Visit	201 01-

In [25]:
```
df['time']=pd.DatetimeIndex(df['START_DATE']).hour
```

In [26]:
```
df.head(10)
```

Out[26]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	Da
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	201 01-
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	Not.	201 01-
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	201 01-
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	201 01-
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	201 01-
5	2016-01-06 17:15:00	2016-01-06 17:19:00	Business	West Palm Beach	West Palm Beach	4.3	Meal/Entertain	201 01-
6	2016-01-06 17:30:00	2016-01-06 17:35:00	Business	West Palm Beach	Palm Beach	7.1	Meeting	201 01-
7	2016-01-07 13:27:00	2016-01-07 13:33:00	Business	Cary	Cary	0.8	Meeting	201 01-
8	2016-01-10 08:05:00	2016-01-10 08:25:00	Business	Cary	Morrisville	8.3	Meeting	201 01-
9	2016-01-10 12:17:00	2016-01-10 12:44:00	Business	Jamaica	New York	16.5	Customer Visit	201 01-

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

In [28]:
```
df.head(10)
```

Out[28]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	Da
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	201 01-
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	Not.	201 01-
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	201 01-
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	201 01-
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	201 01-
5	2016-01-06 17:15:00	2016-01-06 17:19:00	Business	West Palm Beach	West Palm Beach	4.3	Meal/Entertain	201 01-
6	2016-01-06 17:30:00	2016-01-06 17:35:00	Business	West Palm Beach	Palm Beach	7.1	Meeting	201 01-
7	2016-01-07 13:27:00	2016-01-07 13:33:00	Business	Cary	Cary	0.8	Meeting	201 01-
8	2016-01-10 08:05:00	2016-01-10 08:25:00	Business	Cary	Morrisville	8.3	Meeting	201 01-
9	2016-01-10 12:17:00	2016-01-10 12:44:00	Business	Jamaica	New York	16.5	Customer Visit	201 01-

In [29]:
```
df.dropna(inplace=True)
```

In [30]:
```
df
```

Out[30]:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPO
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Enterta
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	N
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Suppl
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeti
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Vi
...	...	...	...	...	...	...	
1151	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Kar?chi	Unknown Location	3.9	Temporary S
1152	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Unknown Location	Unknown Location	16.2	Meeti
1153	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Katunayake	Gampaha	6.4	Temporary S
1154	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Gampaha	Ilukwatta	48.2	Temporary S
1155	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Gampaha	Ilukwatta	12204.7	N

1156 rows × 10 columns

In [31]:
```
(c,r)=df.shape
print("columns=",c)
print("row=",r)
```

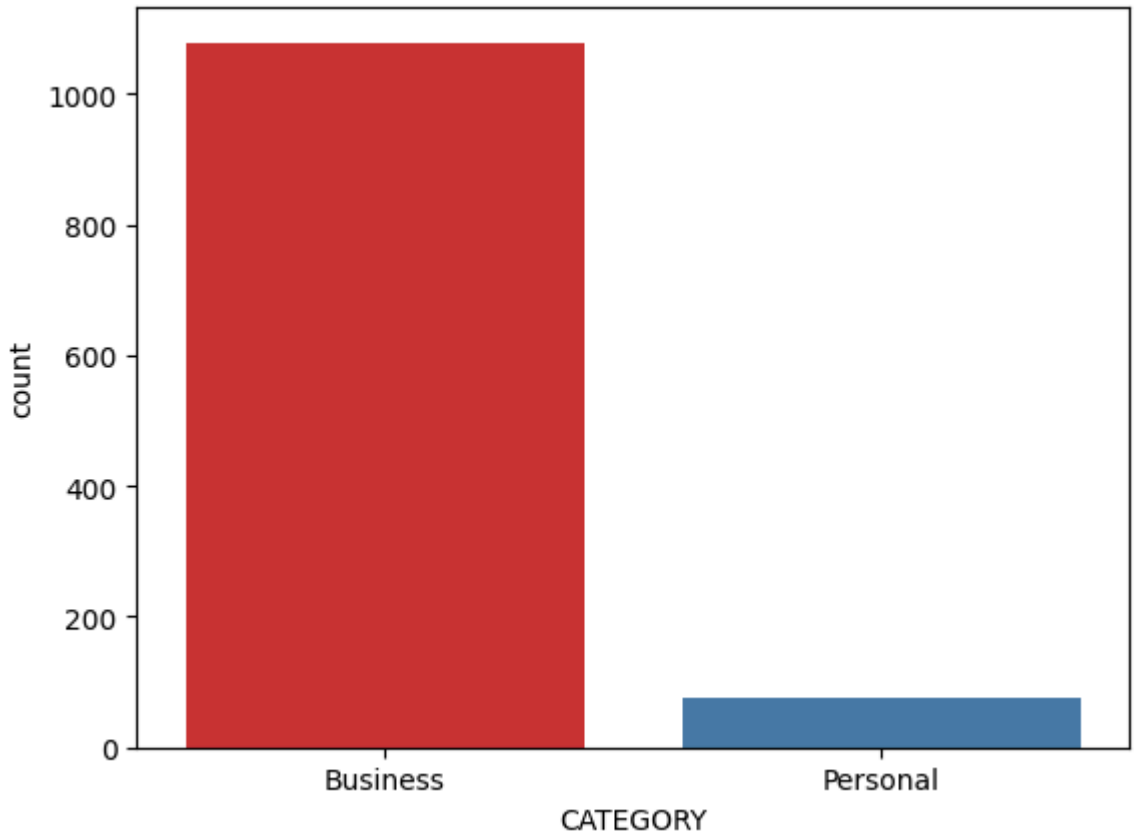
```
columns= 1156
row= 10
```

Data visualization

In [32]:
```
sns.countplot(x='CATEGORY', data=df,palette="Set1")
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\22903052.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(x='CATEGORY', data=df,palette="Set1")
```

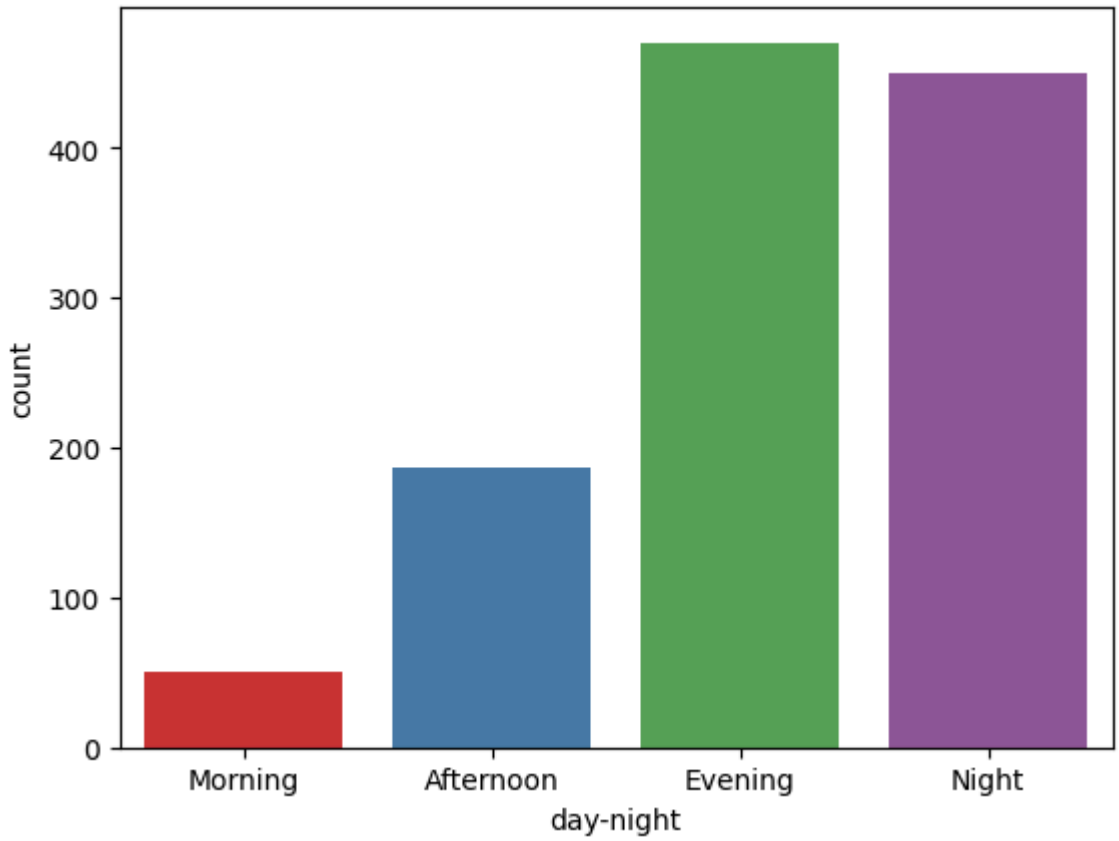
Out[32]: <Axes: xlabel='CATEGORY', ylabel='count'>



In [33]:
```
sns.countplot(x='day-night', data=df,palette="Set1")
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\4094278088.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(x='day-night', data=df,palette="Set1")
```

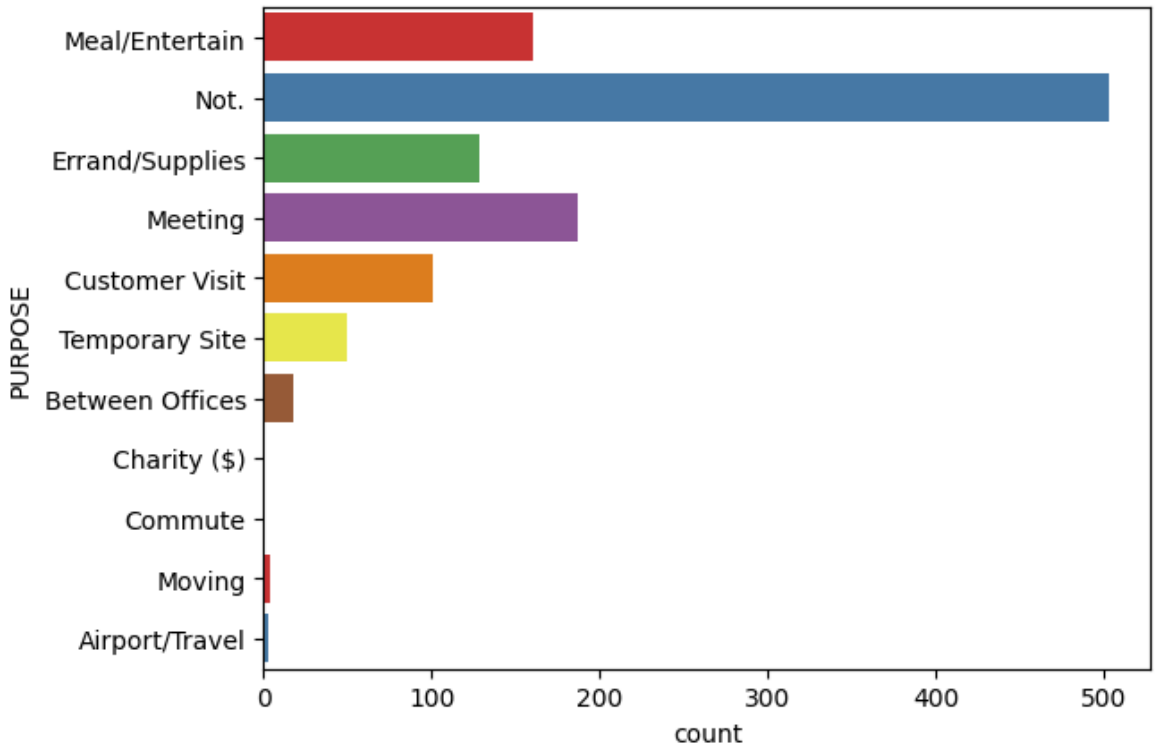
Out[33]: <Axes: xlabel='day-night', ylabel='count'>



In [34]:
```
sns.countplot(y='PURPOSE', data=df,palette="Set1")
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\895393509.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(y='PURPOSE', data=df,palette="Set1")
```

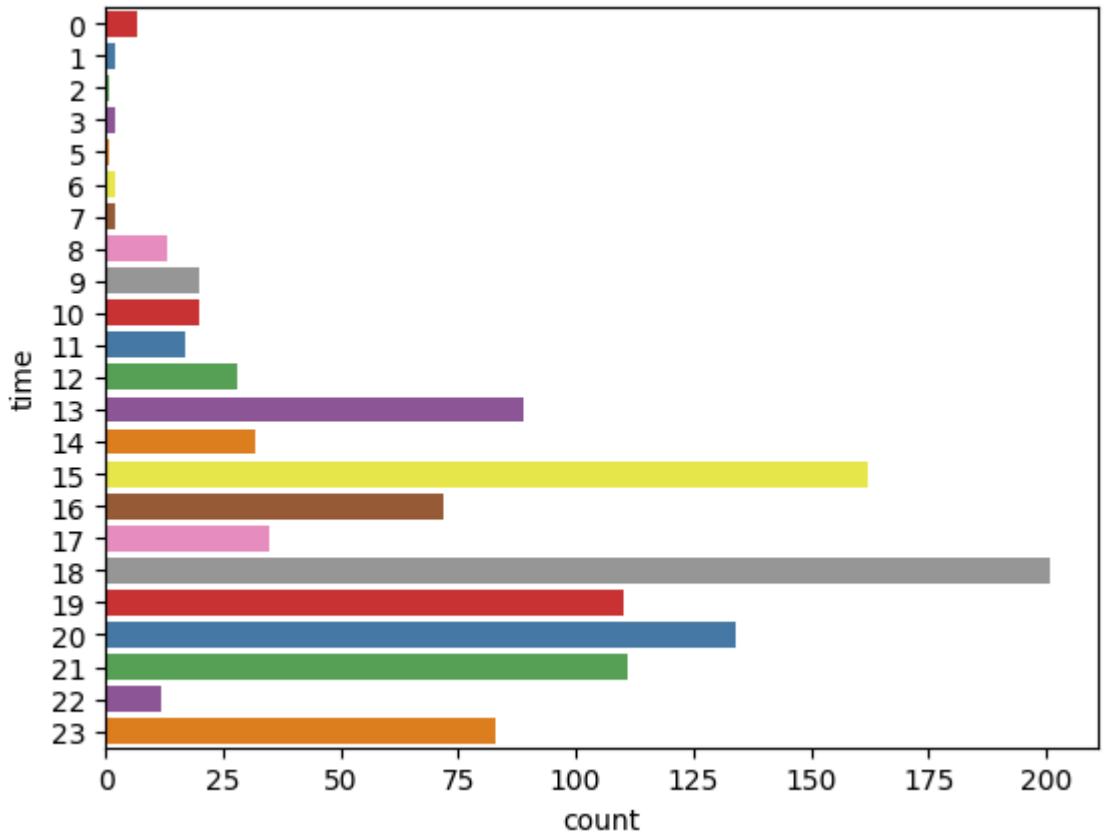
Out[34]: <Axes: xlabel='count', ylabel='PURPOSE'>



In [35]:
```
sns.countplot(y='time', data=df,palette="Set1")
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\592257224.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(y='time', data=df,palette="Set1")
```

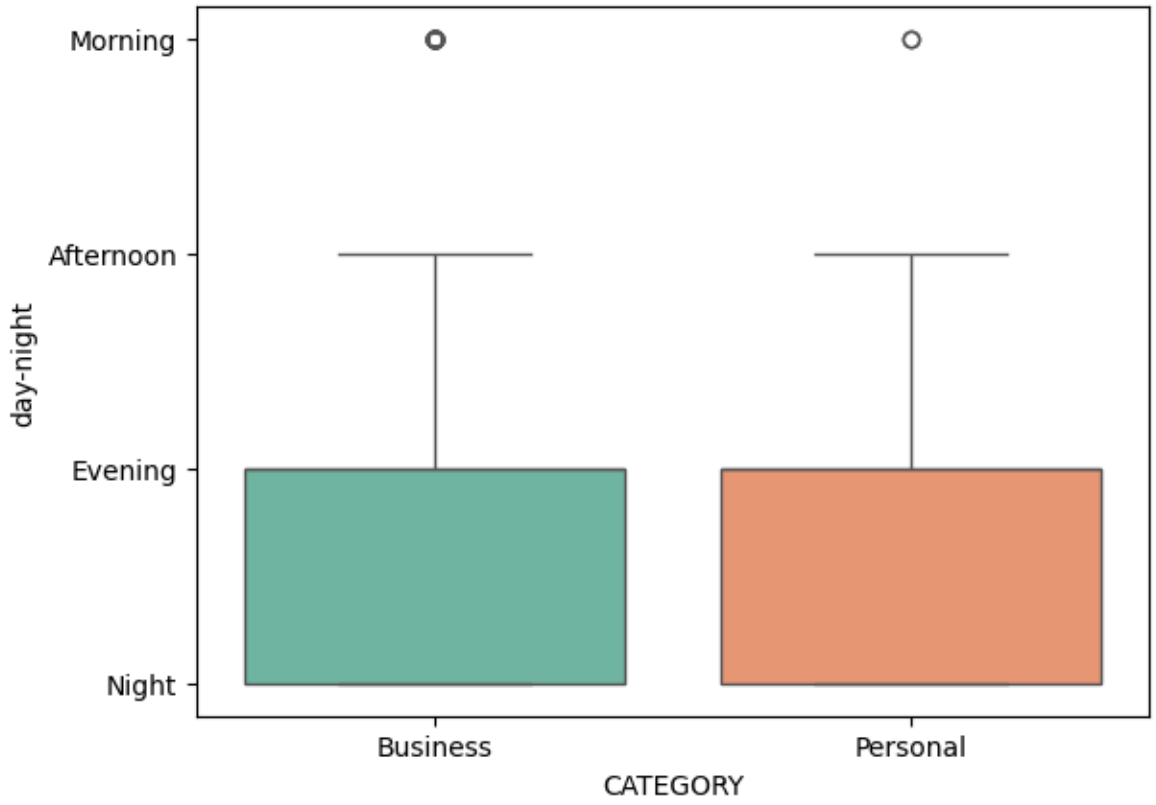
Out[35]: <Axes: xlabel='count', ylabel='time'>



In [36]:
```
sns.boxplot(x='CATEGORY',y='day-night',data=df,palette="Set2")
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\2202060117.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.boxplot(x='CATEGORY',y='day-night',data=df,palette="Set2")
```

Out[36]:
```
<Axes: xlabel='CATEGORY', ylabel='day-night'>
```



In [37]:
```
df['Month'] = pd.to_datetime(df['START_DATE']).dt.month_name()
```

In [38]:
```
df['Day'] = pd.to_datetime(df['START_DATE']).dt.day_name()
```

In [39]:
```
df
```

Out[39]:

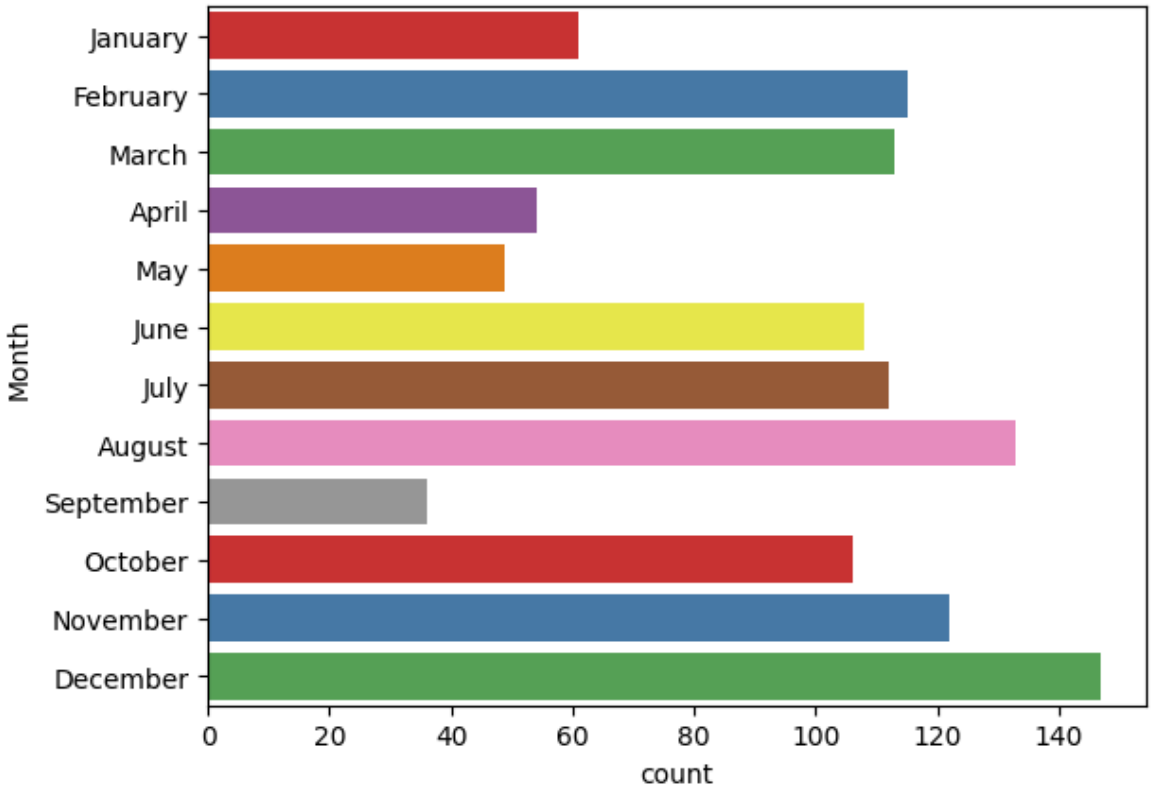
	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPO
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Enterta
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	N
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Suppl
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeti
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Vi
...	...	...	...	...	...	...	
1151	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Kar?chi	Unknown Location	3.9	Temporary S
1152	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Unknown Location	Unknown Location	16.2	Meeti
1153	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Katunayake	Gampaha	6.4	Temporary S
1154	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Gampaha	Ilukwatta	48.2	Temporary S
1155	2016-12-12 20:48:00	2016-12-12 20:57:00	Business	Gampaha	Ilukwatta	12204.7	N

1156 rows × 12 columns

In [40]:
```
sns.countplot(y='Month', data=df,palette="Set1")
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\3486702215.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(y='Month', data=df,palette="Set1")
```

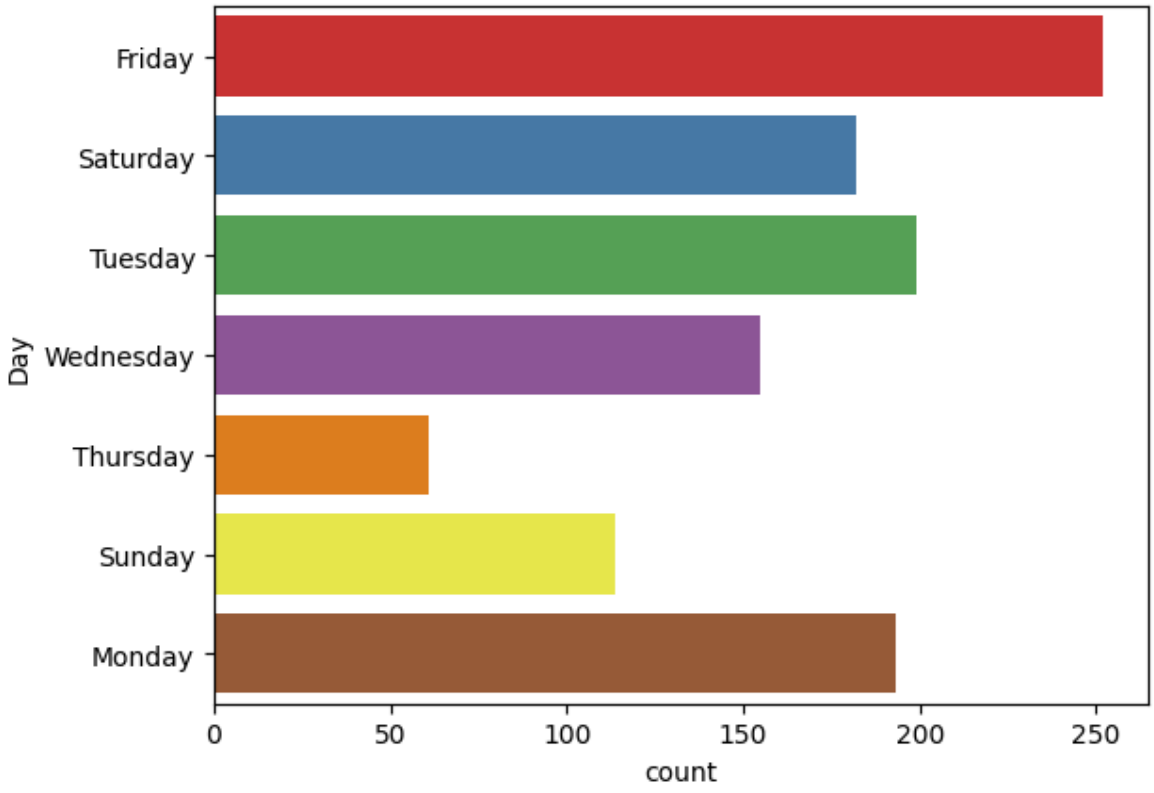
Out[40]: <Axes: xlabel='count', ylabel='Month'>



In [41]:
```
sns.countplot(y='Day', data=df,palette="Set1")
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\3198250369.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(y='Day', data=df,palette="Set1")
```

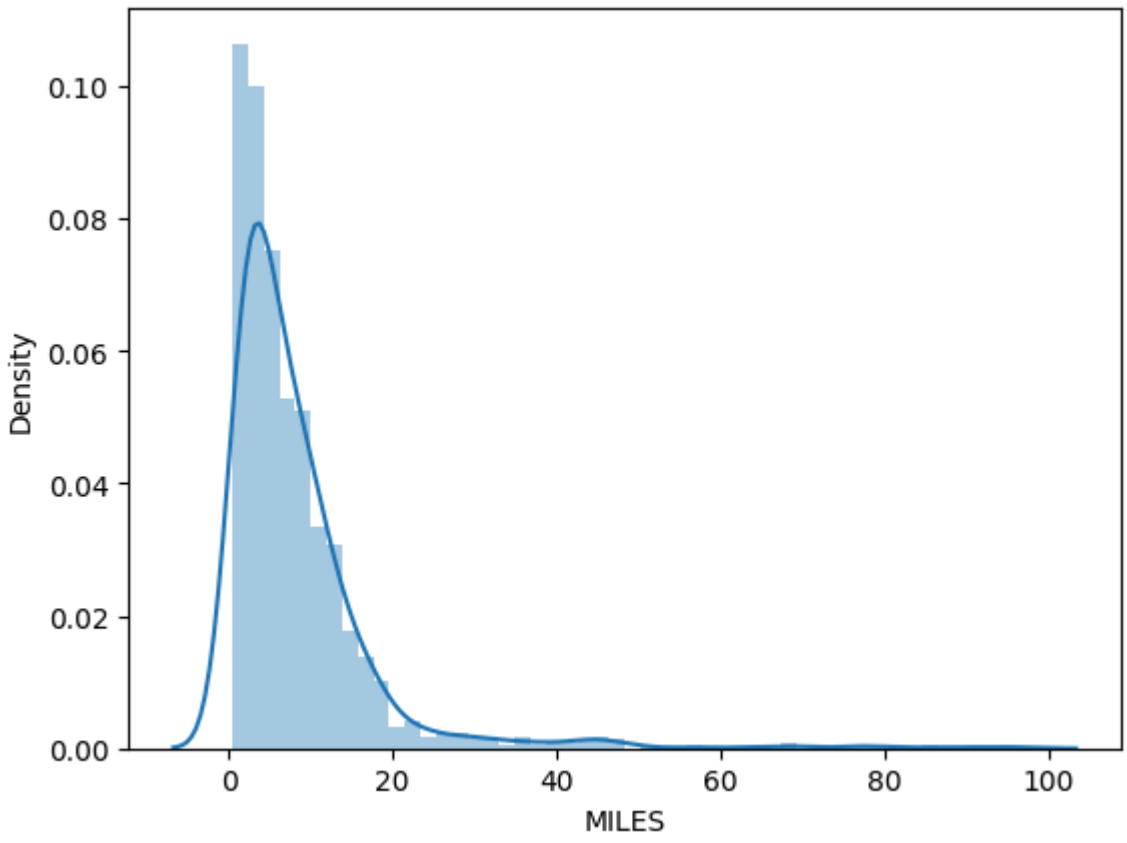
Out[41]: <Axes: xlabel='count', ylabel='Day'>



In [59]:
```
sns.distplot(df[df['MILES']<99]['MILES'])
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\1534027895.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(df[df['MILES']<99]['MILES'])
```

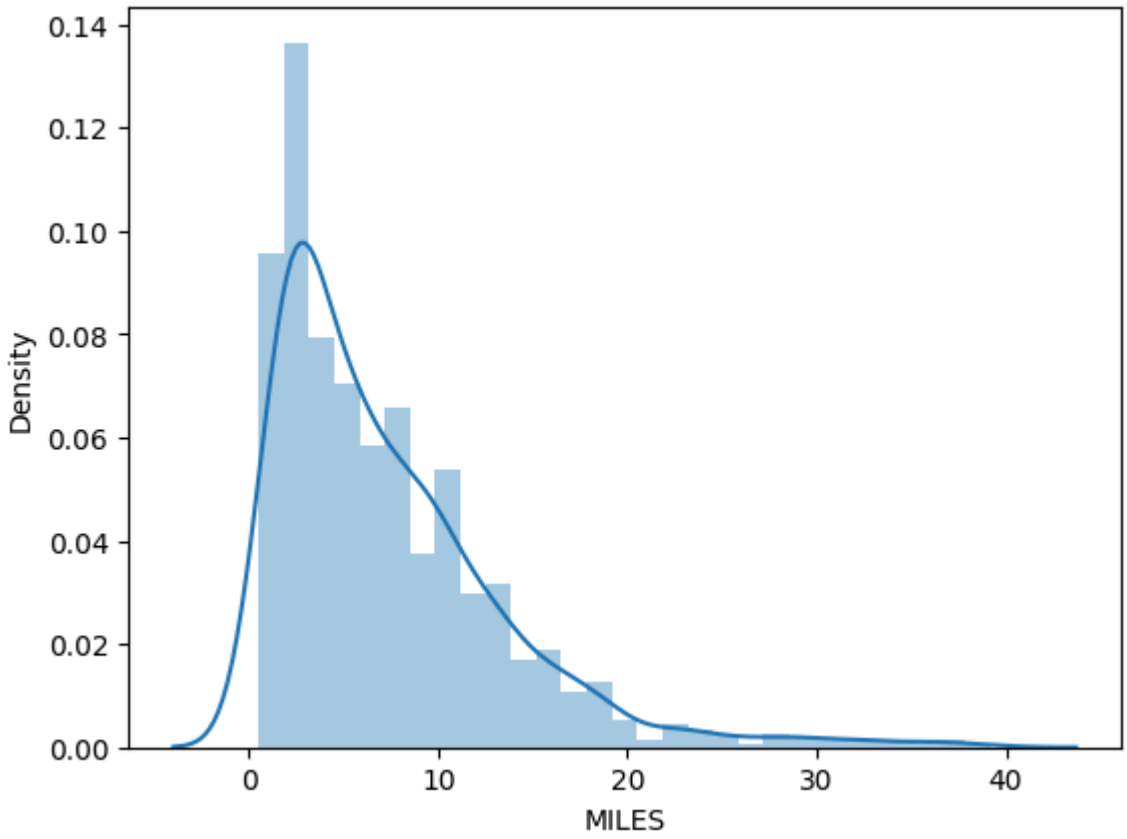
Out[59]: <Axes: xlabel='MILES', ylabel='Density'>



In [60]:
```
sns.distplot(df[df['MILES']<40]['MILES'])
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_11832\1171915261.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(df[df['MILES']<40]['MILES'])
```

Out[60]: <Axes: xlabel='MILES', ylabel='Density'>



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: