In [9]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [10]:
```python
data = pd.read_csv("Advertising.csv")
print(data.head())
```

```
   Unnamed: 0     TV  Radio  Newspaper  Sales
0           1  230.1   37.8       69.2   22.1
1           2   44.5   39.3       45.1   10.4
2           3   17.2   45.9       69.3    9.3
3           4  151.5   41.3       58.5   18.5
4           5  180.8   10.8       58.4   12.9
```

In [11]:
```python
print(data.isnull().sum())
```

```
Unnamed: 0    0
TV            0
Radio         0
Newspaper     0
Sales         0
dtype: int64
```

In [15]:
```python
plt.style.use('seaborn-whitegrid')
plt.figure(figsize=(8, 6))
sns.heatmap(data.corr())
plt.show()
```

```
C:\Users\NAKSHI MILAN SHAH\AppData\Local\Temp\ipykernel_10796\3748562119.py:1: Matp
lotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated s
ince 3.6, as they no longer correspond to the styles shipped by seaborn. However, t
hey will remain available as 'seaborn-v0_8-<style>'. Alternatively, directly use th
e seaborn API instead.
  plt.style.use('seaborn-whitegrid')
```
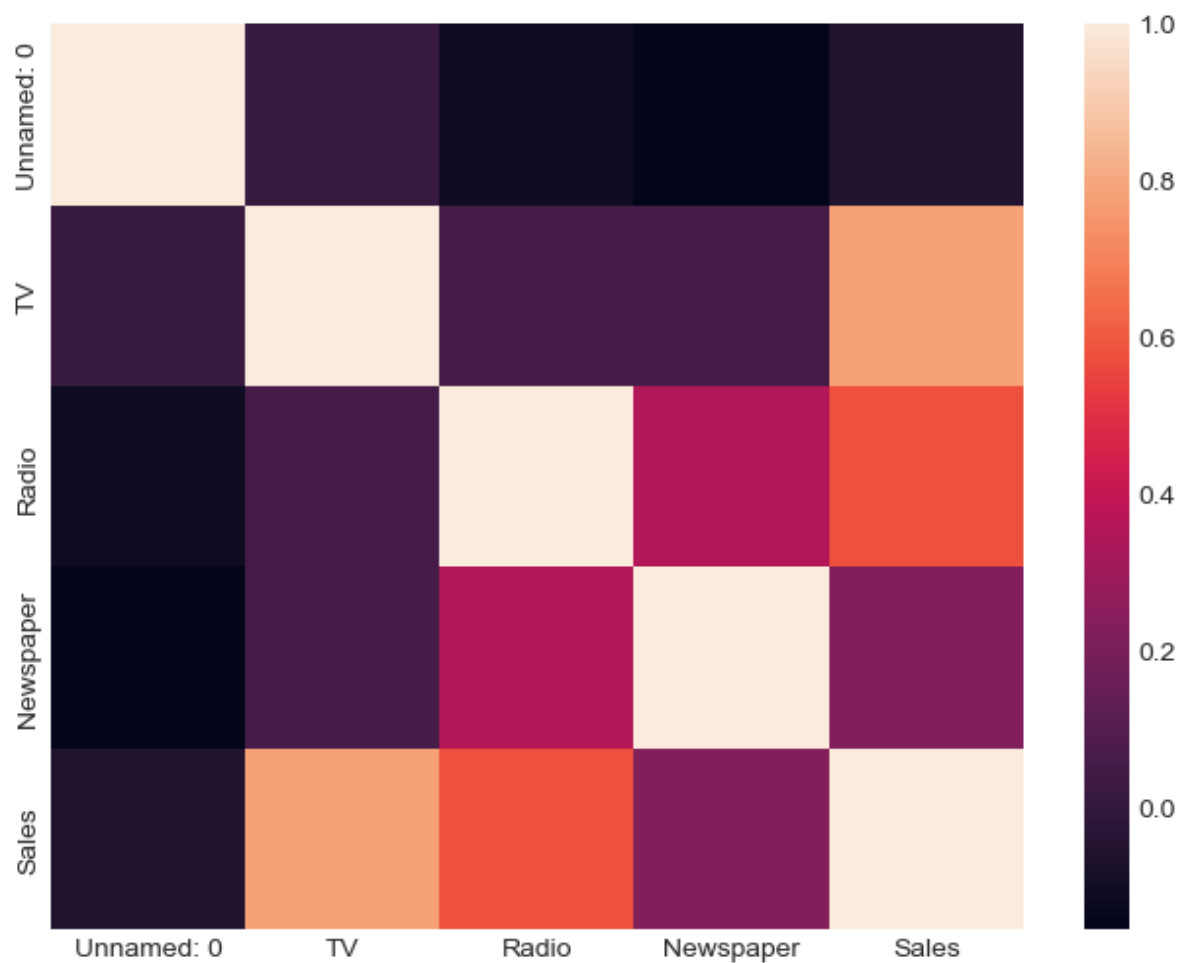
```
In [6]:  x = np.array(data.drop(["Sales"], axis=1))
         y = np.array(data["Sales"])
         xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=4
         model = LinearRegression()
         model.fit(xtrain, ytrain)
         ypred = model.predict(xtest)

         data = pd.DataFrame(data={"Predicted Sales": ypred.flatten()})
         print(data)
```

```
     Predicted Sales
0          16.412277
1          20.843193
2          21.511869
3          10.653100
4          22.124058
5          13.123835
6          21.037489
7           7.503854
8          13.647870
9          15.120409
10          9.020556
11          6.706666
12         14.401551
13          8.886090
14          9.743533
15         12.215520
16          8.713709
17         16.294145
18         10.268023
19         18.804982
20         19.588647
21         13.223183
22         12.364787
23         21.314736
24          7.797347
25          5.781900
26         20.749348
27         11.954767
28          9.199109
29          8.521787
30         12.404313
31          9.948652
32         21.368803
33         12.253748
34         18.299435
35         20.133474
36         14.074646
37         20.885947
38         11.052150
39          4.551687
```

In [ ]: