```
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


data=pd.read_csv("/content/advertising.xls")
data
```

Loading...

```
        TV  Radio  Newspaper  Sales
  0  230.1   37.8       69.2   22.1
  1   44.5   39.3       45.1   10.4
  2   17.2   45.9       69.3   12.0
  3  151.5   41.3       58.5   16.5
  4  180.8   10.8       58.4   17.9
...    ...    ...        ...    ...
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1   14.0
197  177.0    9.3        6.4   14.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   18.4

200 rows × 4 columns
```

Next steps: Generate code with data View recommended plots

Files ✕

..
- sample_data
  - README.md
  - anscombe.json
  - california_housing_test.csv
  - california_housing_train.csv
  - mnist_test.csv
  - mnist_train_small.csv
- advertising.xls

Next steps: Generate code with data View recommended plots

```
data.head() #first five rows of the dataset
```

Loading...

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

Next steps: Generate code with data View recommended plots

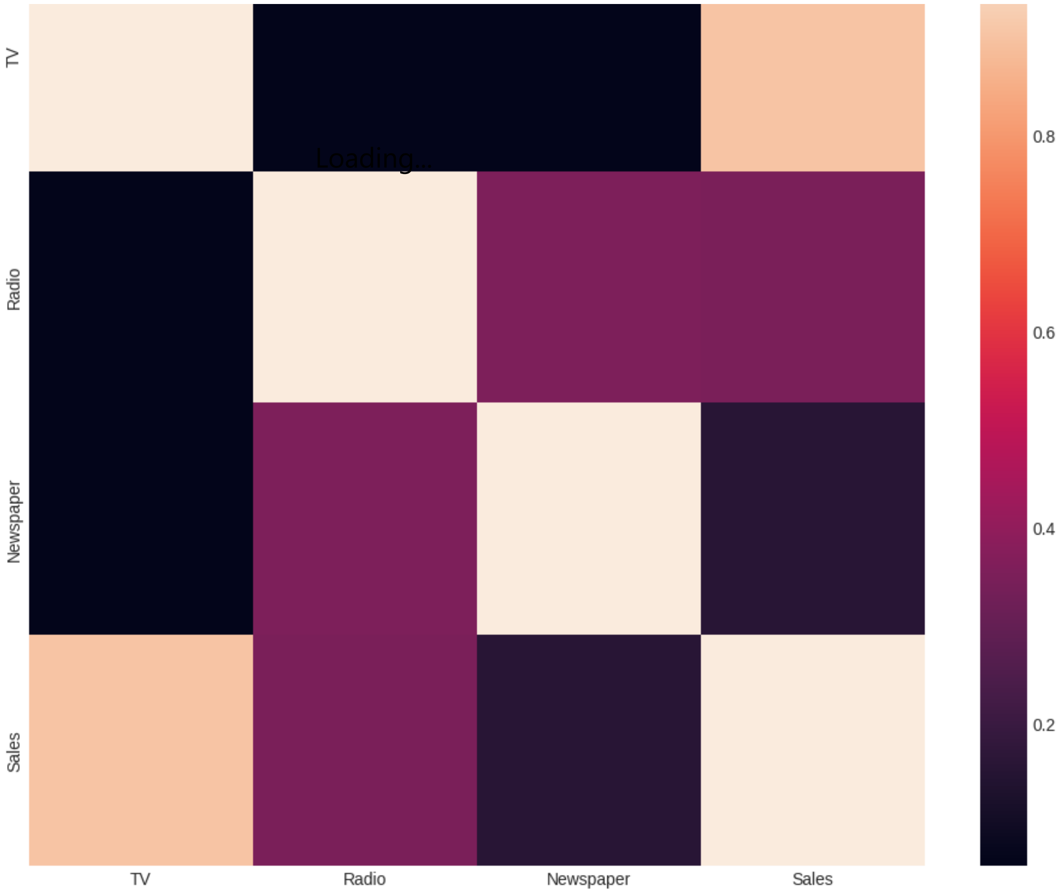
```
print(data.isnull().sum())
```

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

this means that the dataset does not contain any null values.

```
#correlation between features
plt.style.use('seaborn-whitegrid')
plt.figure(figsize=(12,10))
sns.heatmap(data.corr())
plt.show()
```

```
<ipython-input-19-4f75ecb7c7a6>:2: MatplotlibDeprecationWarning: The s
  plt.style.use('seaborn-whitegrid')
```


TV
Radio
Newspaper
Sales
Loading
TV
Radio
Newspaper
Sales
0.8
0.6
0.4
0.2

```
#training model using Linear Regression

x= np.array(data.drop(["Sales"],axis=1)) #use axis=1 to indicate column wise
y= np.array(data["Sales"])Loading...

xtrain, xtest, ytrain, ytest= train_test_split(x, y, test_size=0.2, random_s

model=LinearRegression()
model.fit(xtrain, ytrain)
ypred= model.predict(xtest)

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

```
    Predicted Sales
0         17.034772
1         20.409740
2         23.723989
3          9.272785
4         21.682719
5         12.569402
6         21.081195
7          8.690350
8         17.237013
9         16.666575
10         8.923965
11         8.481734
12        18.207512
```

```
13     8.067507
14    12.645510
15    14.931628  Loading...
16     8.128146
17    17.898766
18    11.008806
19    20.478328
20    20.806318
21    12.598833
22    10.905183
23    22.388548
24     9.417961
25     7.925067
26    20.839085
27    13.815209
28    10.770809
29     7.926825
30    15.959474
31    10.634909
32    20.802920
33    10.434342
34    21.578475
35    21.183645
```

36	12.128218
37	22.809533
38	12.609928
39	6.464413

Loading...

Next steps: Generate code with `data_predicted` View recommended plots

Disk 80.66 GB available