

SALES PREDICTION USING PYTHON:

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import linear_model
```

```
In [2]: data_set=pd.read_csv(r"C:\Users\HP\OneDrive\Documents\mohankumar\Advertising1.csv")
```

```
In [3]: data_set.head()
```

```
Out[3]:
```

	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	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

```
In [4]: data_set.tail()
```

```
Out[4]:
```

	TV	Radio	Newspaper	Sales
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

```
In [5]: data_set.shape
```

```
Out[5]: (200, 4)
```

```
In [6]: data_set.isnull().sum() #checking missing values.
```

```
Out[6]: TV          0
Radio          0
Newspaper      0
Sales          0
dtype: int64
```

```
In [7]: data_set.describe()
```

```
Out[7]:
```

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	14.022500
std	85.854236	14.846809	21.778621	5.217457
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	10.375000
50%	149.750000	22.900000	25.750000	12.900000
75%	218.825000	36.525000	45.100000	17.400000
max	296.400000	49.600000	114.000000	27.000000

```
In [8]: data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0    TV          200 non-null    float64
1    Radio       200 non-null    float64
2    Newspaper   200 non-null    float64
3    Sales       200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

```
In [9]: data_set.columns
```

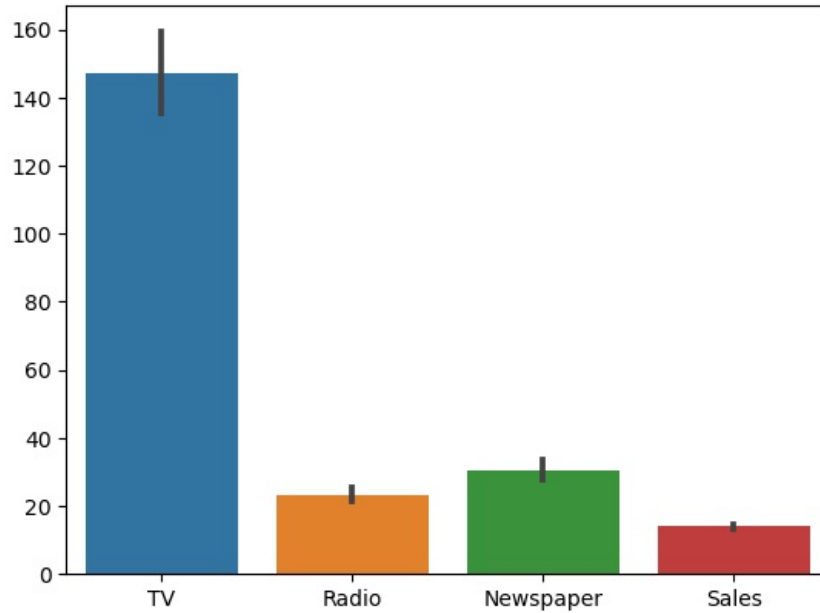
```
Out[9]: Index(['TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')
```

```
In [ ]:
```

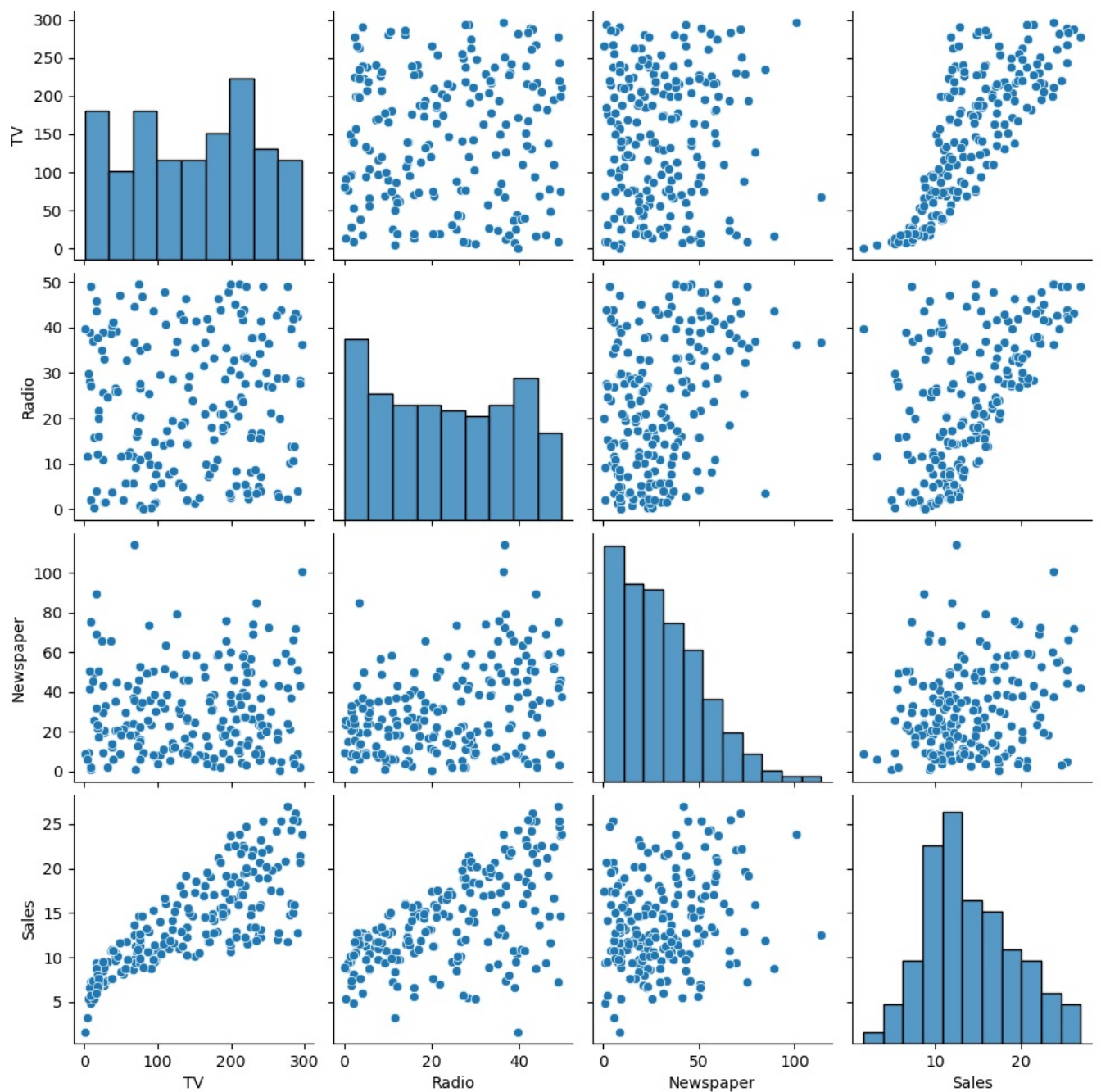
```
In [ ]:
```

Data visualization

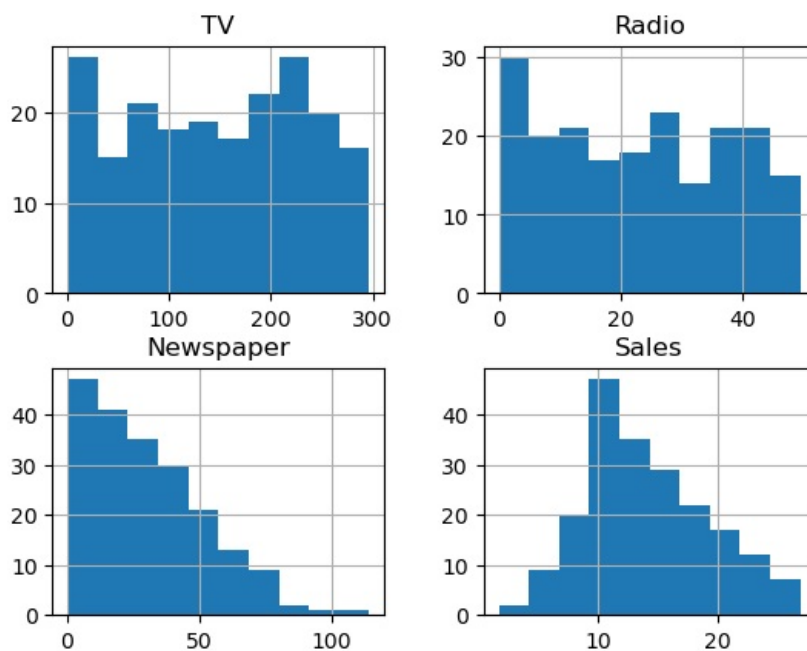
```
In [22]: sns.barplot(data_set)  
plt.show()
```



```
In [11]: sns.pairplot(data_set)  
plt.show()
```

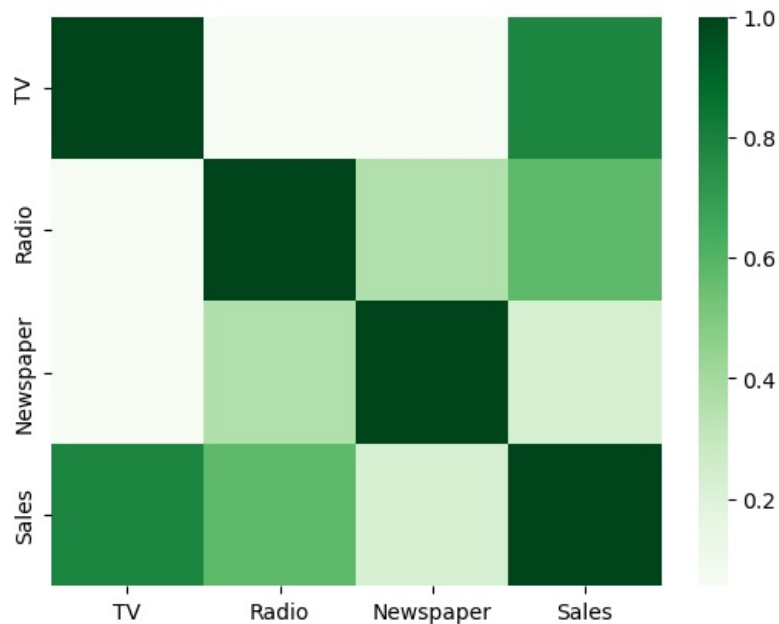


```
In [12]: data_set.hist()
plt.show()
```



```
In [13]: sns.heatmap(data_set.corr(), cmap='Greens')
```

Out[13]: <Axes: >



In []:

In []:

Training the model

In [15]: `inputs=data_set.drop(['Sales'],axis='columns')`
`inputs`

Out[15]:

	TV	Radio	Newspaper
0	230.1	37.8	69.2
1	44.5	39.3	45.1
2	17.2	45.9	69.3
3	151.5	41.3	58.5
4	180.8	10.8	58.4
...
195	38.2	3.7	13.8
196	94.2	4.9	8.1
197	177.0	9.3	6.4
198	283.6	42.0	66.2
199	232.1	8.6	8.7

200 rows × 3 columns

In [16]: `Target=data_set.Sales`
`Target`

Out[16]:

0	22.1
1	10.4
2	9.3
3	18.5
4	12.9
...	...
195	7.6
196	9.7
197	12.8
198	25.5
199	13.4

Name: Sales, Length: 200, dtype: float64

In [17]: `model=linear_model.LinearRegression()`
`inputs=inputs.values`

In [18]: `model.fit(inputs,Target)`

Out[18]:

▼ LinearRegression

LinearRegression()

In []:

In []:

Final Result of Sales prediction using python:

In [27]:

prediction=model.predict([[3.4,5.2,7.5]])
print("our sales prediction is",prediction)

#input: (T.V, Radio, Newspaper.)

our sales prediction is [4.06706405]

In []:

In [20]: *# Thanking you...*

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js