

Import Libraries

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
%matplotlib inline
```

Reading Dataset

```
In [2]: spds=pd.read_csv("C:\\Users\\Lenovo\\Downloads\\Advertising.csv")
```

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

```
Out[3]:
```

| | 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 [4]: spds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Unnamed: 0   200 non-null    int64
1   TV           200 non-null    float64
2   Radio        200 non-null    float64
3   Newspaper    200 non-null    float64
4   Sales        200 non-null    float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

```
In [5]: spds.describe()
```

```
Out[5]:
```

| | Unnamed: 0 | TV | Radio | Newspaper | Sales |
|-------|------------|------------|------------|------------|------------|
| count | 200.000000 | 200.000000 | 200.000000 | 200.000000 | 200.000000 |
| mean | 100.500000 | 147.042500 | 23.264000 | 30.554000 | 14.022500 |
| std | 57.879185 | 85.854236 | 14.846809 | 21.778621 | 5.217457 |
| min | 1.000000 | 0.700000 | 0.000000 | 0.300000 | 1.600000 |
| 25% | 50.750000 | 74.375000 | 9.975000 | 12.750000 | 10.375000 |
| 50% | 100.500000 | 149.750000 | 22.900000 | 25.750000 | 12.900000 |
| 75% | 150.250000 | 218.825000 | 36.525000 | 45.100000 | 17.400000 |
| max | 200.000000 | 296.400000 | 49.600000 | 114.000000 | 27.000000 |

Remove unnecessary column

```
In [6]: spds=spds.drop('Unnamed: 0',axis=1)
```

```
In [7]: spds.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 [8]: spds.isnull().sum()
```

```
Out[8]:
TV          0
Radio       0
Newspaper   0
Sales       0
dtype: int64
```

```
In [9]: spds.head()
```

Out[9]:

| | 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 [10]: spds.tail()
```

Out[10]:

| | 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 [11]: spds.columns
```

Out[11]: Index(['TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')

```
In [12]: spds.shape
```

Out[12]: (200, 4)

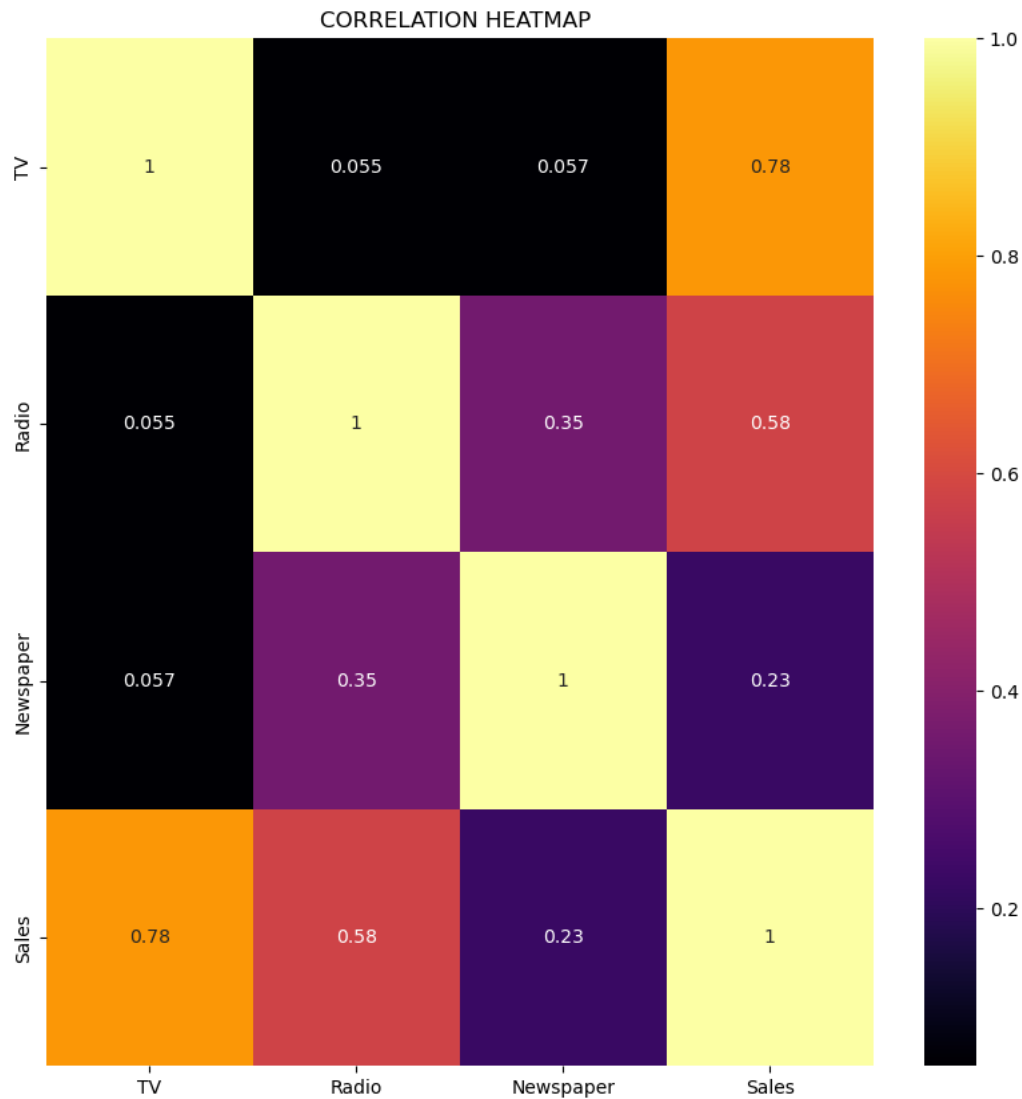
```
In [13]: spds.corr()
```

Out[13]:

| | TV | Radio | Newspaper | Sales |
|-----------|----------|----------|-----------|----------|
| TV | 1.000000 | 0.054809 | 0.056648 | 0.782224 |
| Radio | 0.054809 | 1.000000 | 0.354104 | 0.576223 |
| Newspaper | 0.056648 | 0.354104 | 1.000000 | 0.228299 |
| Sales | 0.782224 | 0.576223 | 0.228299 | 1.000000 |

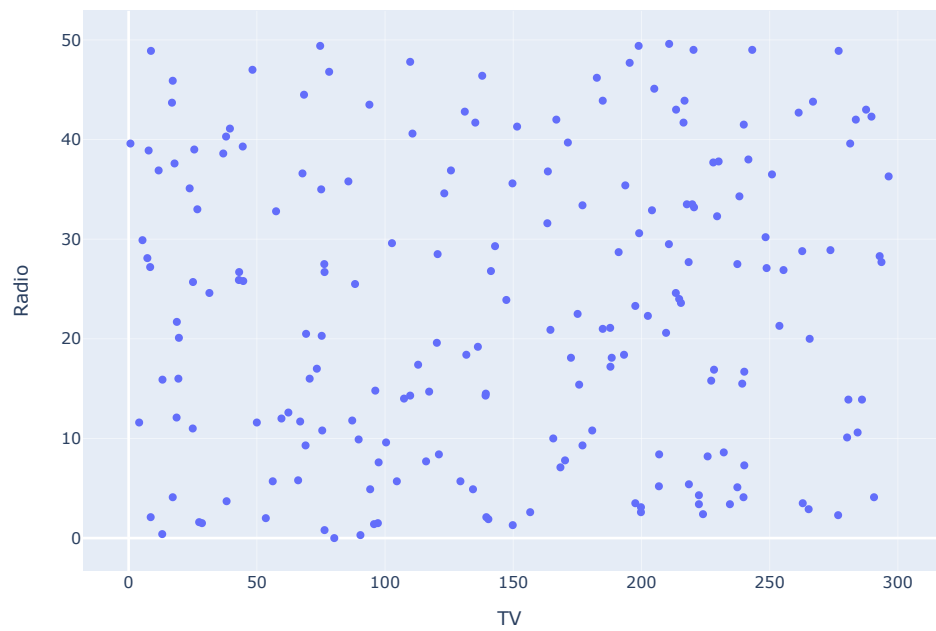
HEATMAP

```
In [14]: plt.figure(figsize=(10,10))
sns.heatmap(spds.corr(),cmap="inferno",annot=True)
plt.title("CORRELATION HEATMAP ")
plt.show()
```



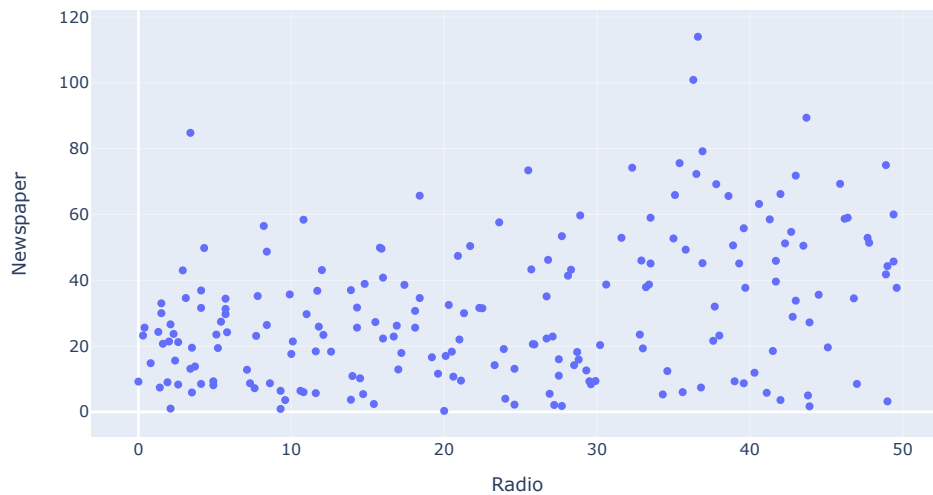
```
In [15]: px.scatter(spds,x="TV",y="Radio",title="TV VS RADIO",height=600,width=800)
```

TV VS RADIO



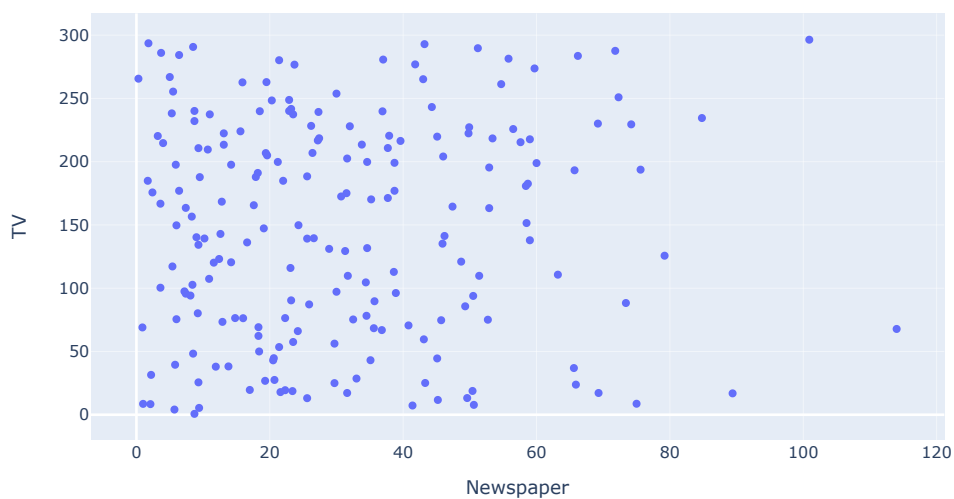
```
In [16]: px.scatter(spds,x="Radio",y="Newspaper",title="RADIO VS NEWSPAPER",height=500,width=800)
```

RADIO VS NEWSPAPER



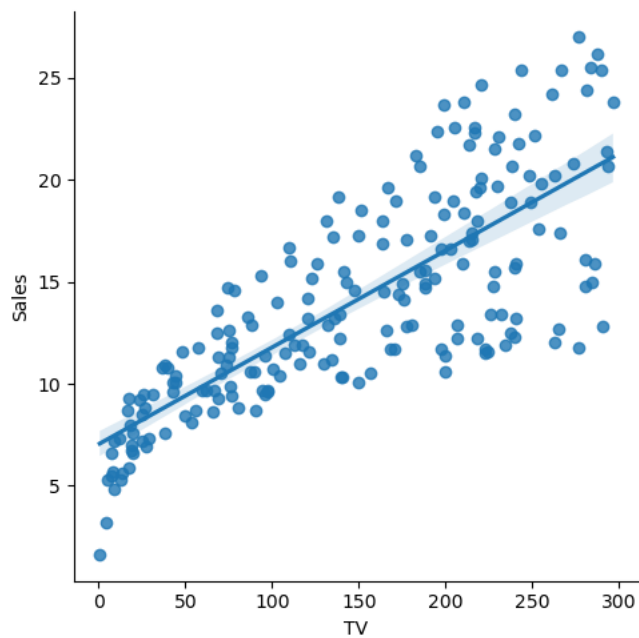
```
In [17]: px.scatter(spds,x="Newspaper",y="TV",title="NEWSPAPER VS TV",height=500,width=800)
```

NEWSPAPER VS TV



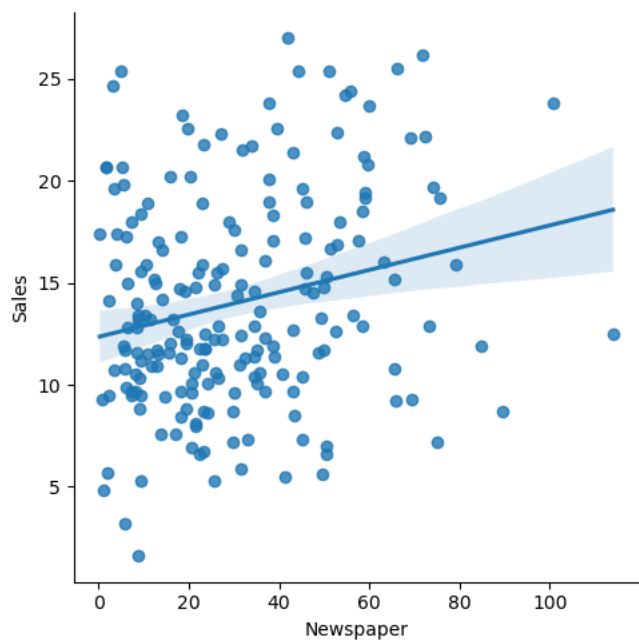
```
In [18]: sns.lmplot(x="TV",y="Sales",data=spds)
```

```
Out[18]: <seaborn.axisgrid.FacetGrid at 0x1f20a2ade50>
```



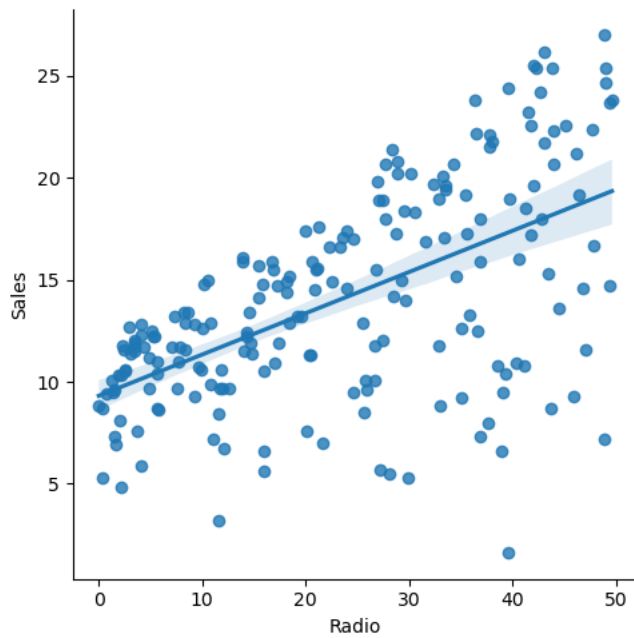
```
In [19]: sns.lmplot(x="Newspaper",y="Sales",data=spds)
```

```
Out[19]: <seaborn.axisgrid.FacetGrid at 0x1f20a2d4210>
```



```
In [20]: sns.lmplot(x="Radio",y="Sales",data=spds)
```

```
Out[20]: <seaborn.axisgrid.FacetGrid at 0x1f20a48c510>
```



TRAINING THE DATASET

In [21]: `spds`

Out[21]:

| | 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 |
| ... | ... | ... | ... | ... |
| 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 |

200 rows × 4 columns

In [22]: `x=spds.drop("Sales",axis=1)`
`y=spds[["Sales"]]`

In [23]: `x.shape`

Out[23]: `(200, 3)`

In [24]: `y.shape`

Out[24]: `(200, 1)`

In [25]: `x.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0    TV          200 non-null    float64
1    Radio       200 non-null    float64
2    Newspaper  200 non-null    float64
dtypes: float64(3)
memory usage: 4.8 KB
```

In [26]: `y.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype  
---  -
0    Sales    200 non-null       float64
dtypes: float64(1)
memory usage: 1.7 KB
```

```
In [27]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

```
In [28]: x_train.shape,y_train.shape,x_test.shape,y_test.shape
```

```
Out[28]: ((140, 3), (140, 1), (60, 3), (60, 1))
```

```
In [29]: from sklearn.linear_model import LinearRegression
model=LinearRegression()
```

```
In [30]: model.fit(x_train,y_train)
```

```
Out[30]: LinearRegression
LinearRegression()
```

```
In [31]: model.intercept_
```

```
Out[31]: array([2.88025529])
```

```
In [32]: model.coef_
```

```
Out[32]: array([[0.04391531, 0.20027962, 0.00184368]])
```

```
In [33]: predict=model.predict(x_test)
x_test
```

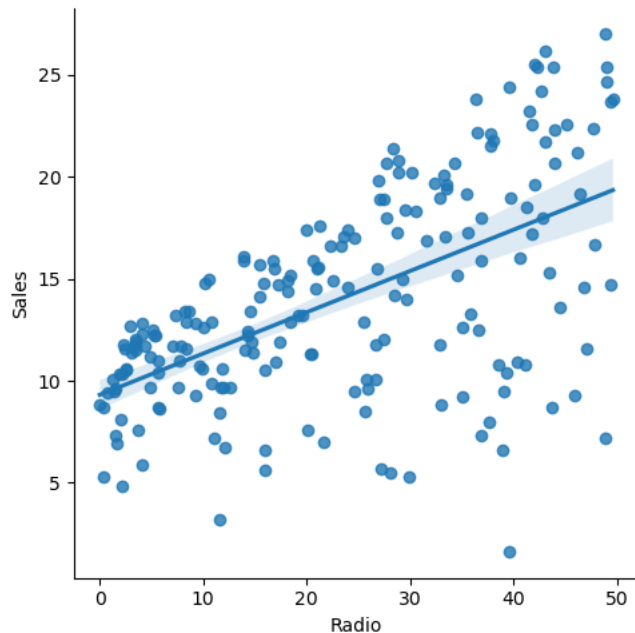
Out[33]:

| | TV | Radio | Newspaper |
|-----|-------|-------|-----------|
| 18 | 69.2 | 20.5 | 18.3 |
| 170 | 50.0 | 11.6 | 18.4 |
| 107 | 90.4 | 0.3 | 23.2 |
| 98 | 289.7 | 42.3 | 51.2 |
| 177 | 170.2 | 7.8 | 35.2 |
| 182 | 56.2 | 5.7 | 29.7 |
| 5 | 8.7 | 48.9 | 75.0 |
| 146 | 240.1 | 7.3 | 8.7 |
| 12 | 23.8 | 35.1 | 65.9 |
| 152 | 197.6 | 23.3 | 14.2 |
| 61 | 261.3 | 42.7 | 54.7 |
| 125 | 87.2 | 11.8 | 25.9 |
| 180 | 156.6 | 2.6 | 8.3 |
| 154 | 187.8 | 21.1 | 9.5 |
| 80 | 76.4 | 26.7 | 22.3 |
| 7 | 120.2 | 19.6 | 11.6 |
| 33 | 265.6 | 20.0 | 0.3 |
| 130 | 0.7 | 39.6 | 8.7 |
| 37 | 74.7 | 49.4 | 45.7 |
| 74 | 213.4 | 24.6 | 13.1 |
| 183 | 287.6 | 43.0 | 71.8 |
| 145 | 140.3 | 1.9 | 9.0 |
| 45 | 175.1 | 22.5 | 31.5 |
| 159 | 131.7 | 18.4 | 34.6 |
| 60 | 53.5 | 2.0 | 21.4 |
| 123 | 123.1 | 34.6 | 12.4 |
| 179 | 165.6 | 10.0 | 17.6 |
| 185 | 205.0 | 45.1 | 19.6 |
| 122 | 224.0 | 2.4 | 15.6 |
| 44 | 25.1 | 25.7 | 43.3 |
| 16 | 67.8 | 36.6 | 114.0 |
| 55 | 198.9 | 49.4 | 60.0 |
| 150 | 280.7 | 13.9 | 37.0 |
| 111 | 241.7 | 38.0 | 23.2 |
| 22 | 13.2 | 15.9 | 49.6 |
| 189 | 18.7 | 12.1 | 23.4 |
| 129 | 59.6 | 12.0 | 43.1 |
| 4 | 180.8 | 10.8 | 58.4 |
| 83 | 68.4 | 44.5 | 35.6 |
| 106 | 25.0 | 11.0 | 29.7 |
| 134 | 36.9 | 38.6 | 65.6 |
| 66 | 31.5 | 24.6 | 2.2 |
| 26 | 142.9 | 29.3 | 12.6 |
| 113 | 209.6 | 20.6 | 10.7 |
| 168 | 215.4 | 23.6 | 57.6 |
| 63 | 102.7 | 29.6 | 8.4 |
| 8 | 8.6 | 2.1 | 1.0 |
| 75 | 16.9 | 43.7 | 89.4 |
| 118 | 125.7 | 36.9 | 79.2 |
| 143 | 104.6 | 5.7 | 34.4 |
| 71 | 109.8 | 14.3 | 31.7 |
| 124 | 229.5 | 32.3 | 74.2 |
| 184 | 253.8 | 21.3 | 30.0 |
| 97 | 184.9 | 21.0 | 22.0 |
| 149 | 44.7 | 25.8 | 20.6 |

| | TV | Radio | Newspaper |
|-----|-------|-------|-----------|
| 24 | 62.3 | 12.6 | 18.3 |
| 30 | 292.9 | 28.3 | 43.2 |
| 160 | 172.5 | 18.1 | 30.7 |
| 40 | 202.5 | 22.3 | 31.6 |
| 56 | 7.3 | 28.1 | 41.4 |

In [34]: `sns.lmplot(x="Radio",y="Sales",data=spds)`

Out[34]: `<seaborn.axisgrid.FacetGrid at 0x1f20c3f4310>`



In [35]: `spds = pd.DataFrame({
 'Actual': y_test.values.flatten(),
 'Predict': predict.flatten()})

spds.head(20)`

Out[35]:

| | Actual | Predict |
|----|--------|-----------|
| 0 | 11.3 | 10.058667 |
| 1 | 8.4 | 7.433188 |
| 2 | 8.7 | 6.953057 |
| 3 | 25.4 | 24.168746 |
| 4 | 11.7 | 11.981720 |
| 5 | 8.7 | 6.544647 |
| 6 | 7.2 | 13.194268 |
| 7 | 13.2 | 14.902403 |
| 8 | 9.2 | 11.076753 |
| 9 | 16.6 | 16.250617 |
| 10 | 24.2 | 23.008116 |
| 11 | 10.6 | 9.120721 |
| 12 | 10.5 | 10.293423 |
| 13 | 15.6 | 15.370966 |
| 14 | 11.8 | 11.623965 |
| 15 | 13.2 | 12.105743 |
| 16 | 17.4 | 18.550308 |
| 17 | 1.6 | 10.858109 |
| 18 | 14.7 | 16.138799 |
| 19 | 17.0 | 17.202814 |

Accuracy

In [36]: `from sklearn.metrics import r2_score
r2_score(y_test,predict)`

```
Out[36]: 0.8649018906637791
```