In [1]:

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```python
df = pd.read_csv('advertising.csv')
```

In [3]:

```python
df.head(6)
```

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 | 12.0 |
| **3** | 151.5 | 41.3 | 58.5 | 16.5 |
| **4** | 180.8 | 10.8 | 58.4 | 17.9 |
| **5** | 8.7 | 48.9 | 75.0 | 7.2 |

In [9]:

```python
#checking of null values
pd.DataFrame(df.isnull().sum(),columns = ["Count of Null Values"]).T
```

Out[9]:

| | TV | Radio | Newspaper | Sales |
|---|---|---|---|---|
| **Count of Null Values** | 0 | 0 | 0 | 0 |

In [10]:

```python
df.describe(include = 'all')
```

Out[10]:

| | TV | Radio | Newspaper | Sales |
|---|---|---|---|---|
| **count** | 200.000000 | 200.000000 | 200.000000 | 200.000000 |
| **mean** | 147.042500 | 23.264000 | 30.554000 | 15.130500 |
| **std** | 85.854236 | 14.846809 | 21.778621 | 5.283892 |
| **min** | 0.700000 | 0.000000 | 0.300000 | 1.600000 |
| **25%** | 74.375000 | 9.975000 | 12.750000 | 11.000000 |
| **50%** | 149.750000 | 22.900000 | 25.750000 | 16.000000 |
| **75%** | 218.825000 | 36.525000 | 45.100000 | 19.050000 |
| **max** | 296.400000 | 49.600000 | 114.000000 | 27.000000 |

In [11]:

```python
df.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 [13]:

```python
#Data Analysis
a = df["TV"]
```

In [14]:

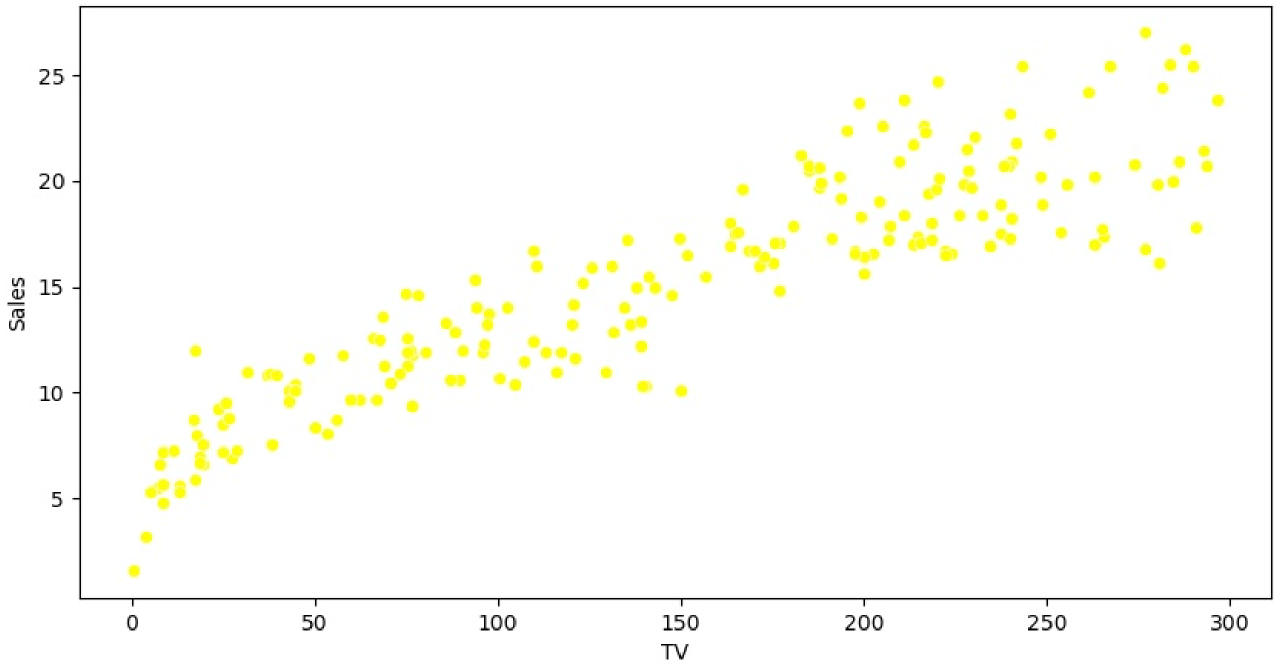
```python
b = df["Sales"]
```

In [15]:

```python
plt.figure(figsize = (10,5))
sns.scatterplot(a,b,color='yellow')
```

Out[15]:

```
<AxesSubplot:xlabel='TV', ylabel='Sales'>
```



In [16]:

```
a= df["Radio"]
```

In [17]:

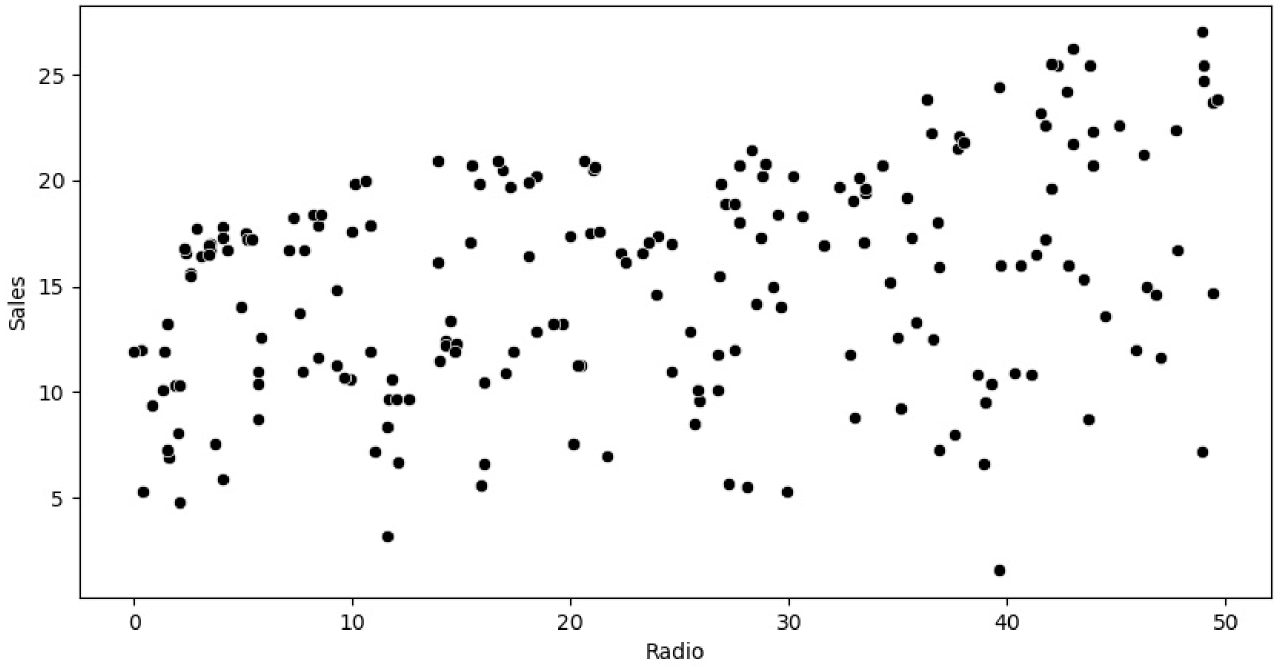
```
b = df["Sales"]
```

In [19]:

```
plt.figure(figsize = (10,5))
sns.scatterplot(a,b,color='black')
```

Out[19]:

```
<AxesSubplot:xlabel='Radio', ylabel='Sales'>
```



In [21]:

```
a = df["Newspaper"]
```

In [22]:

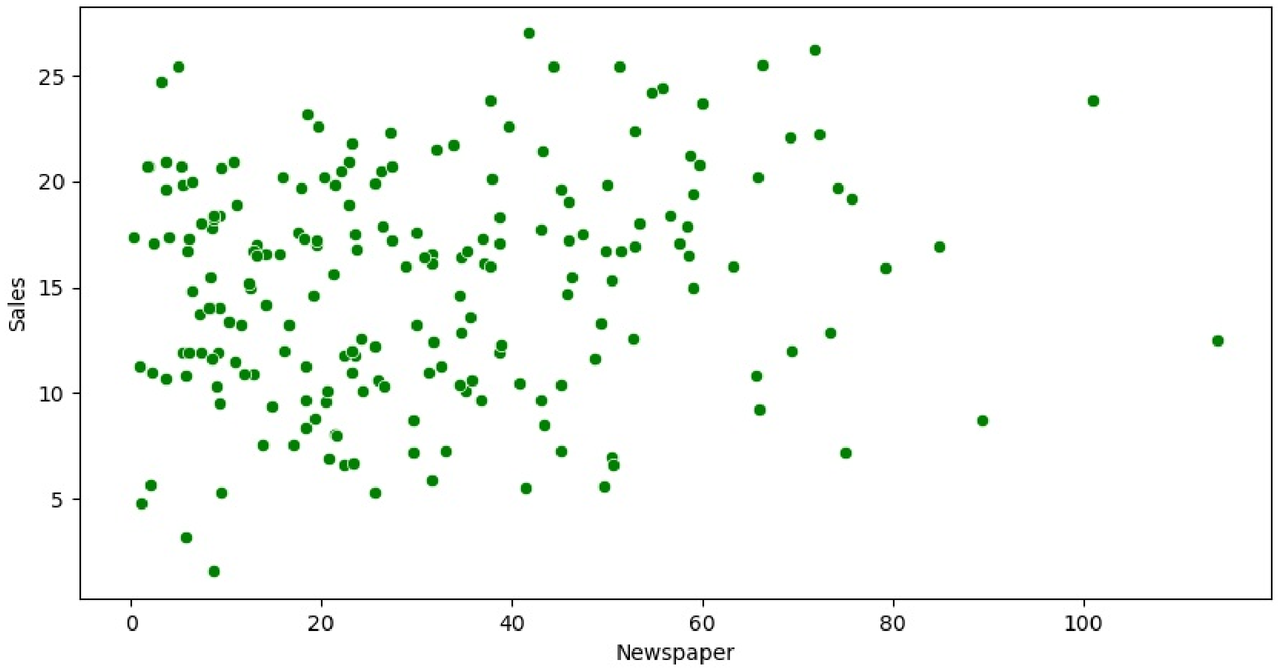
```
b = df["Sales"]
```

In [23]:

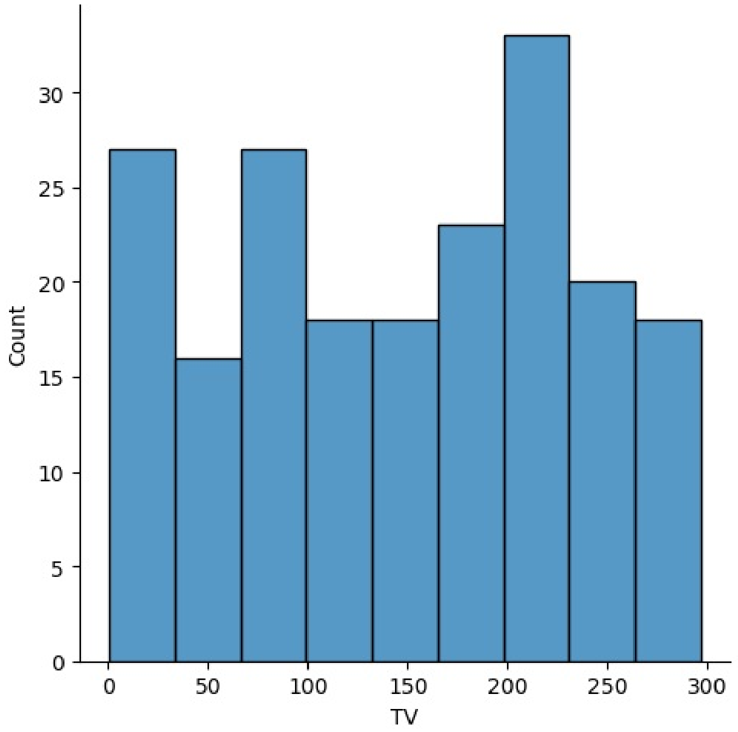
```
plt.figure(figsize = (10,5))
sns.scatterplot(a,b,color='Green')
```

Out[23]:

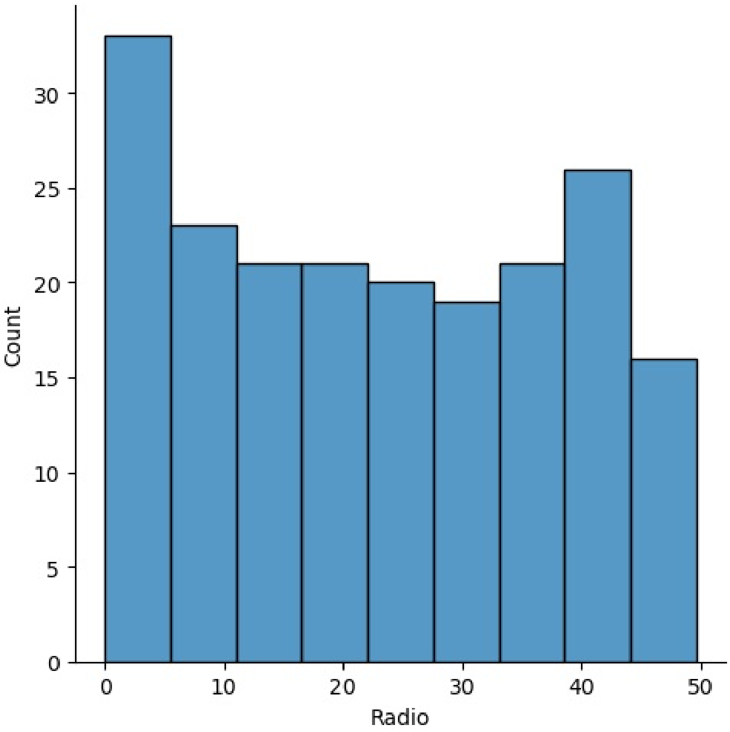
```
<AxesSubplot:xlabel='Newspaper', ylabel='Sales'>
```



```
In [24]: #Distplot
         plt.figure(figsize = (10,5))
         sns.displot(df['TV'])

Out[24]: <seaborn.axisgrid.FacetGrid at 0x1361fd51100>

         <Figure size 1000x500 with 0 Axes>
```

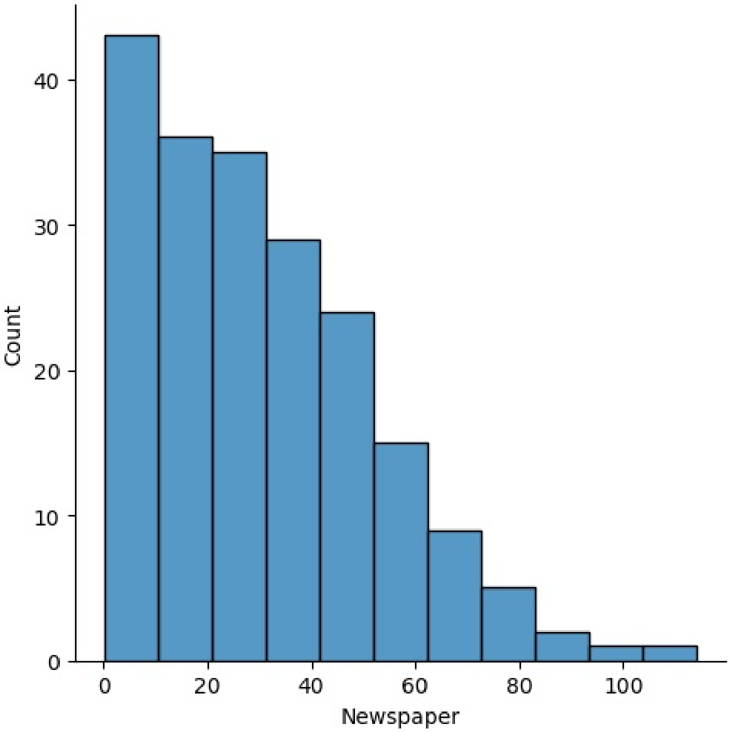


```
In [25]: plt.figure(figsize = (10,5))
         sns.displot(df['Radio'])

Out[25]: <seaborn.axisgrid.FacetGrid at 0x13620157b80>

         <Figure size 1000x500 with 0 Axes>
```



```
In [26]: plt.figure(figsize = (10,5))
         sns.displot(df['Newspaper'])
```

```
Out[26]: <seaborn.axisgrid.FacetGrid at 0x1362027edc0>

<Figure size 1000x500 with 0 Axes>
```

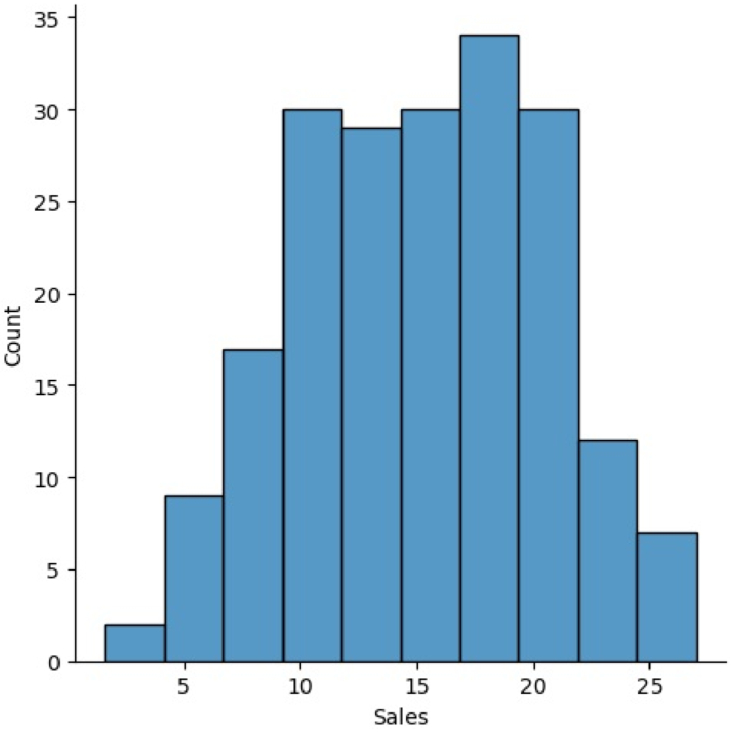


```
In [27]: plt.figure(figsize = (10,5))
         sns.displot(df['Sales'])
```

Out[27]:

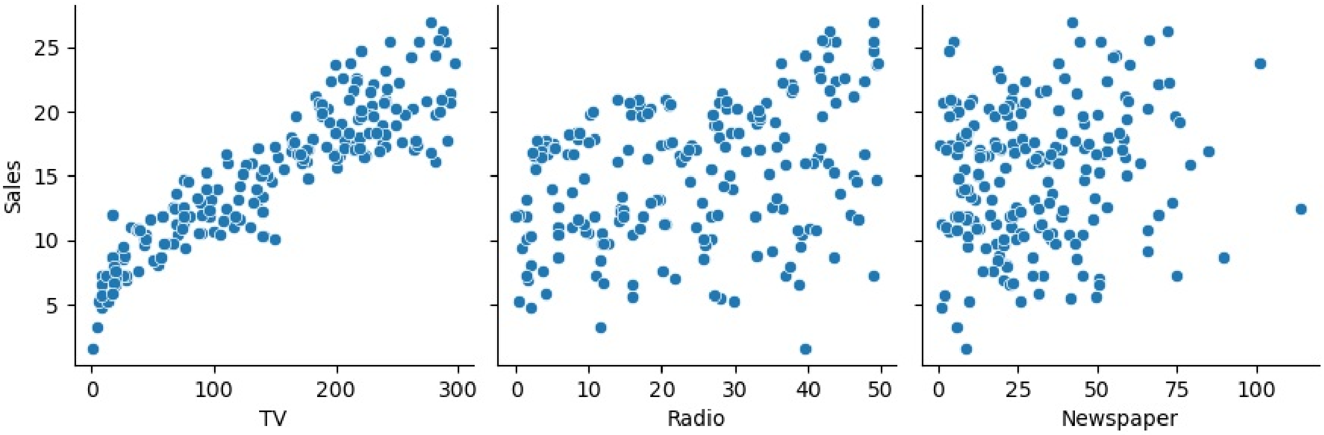
```
<seaborn.axisgrid.FacetGrid at 0x1362027ed90>
```

```
<Figure size 1000x500 with 0 Axes>
```



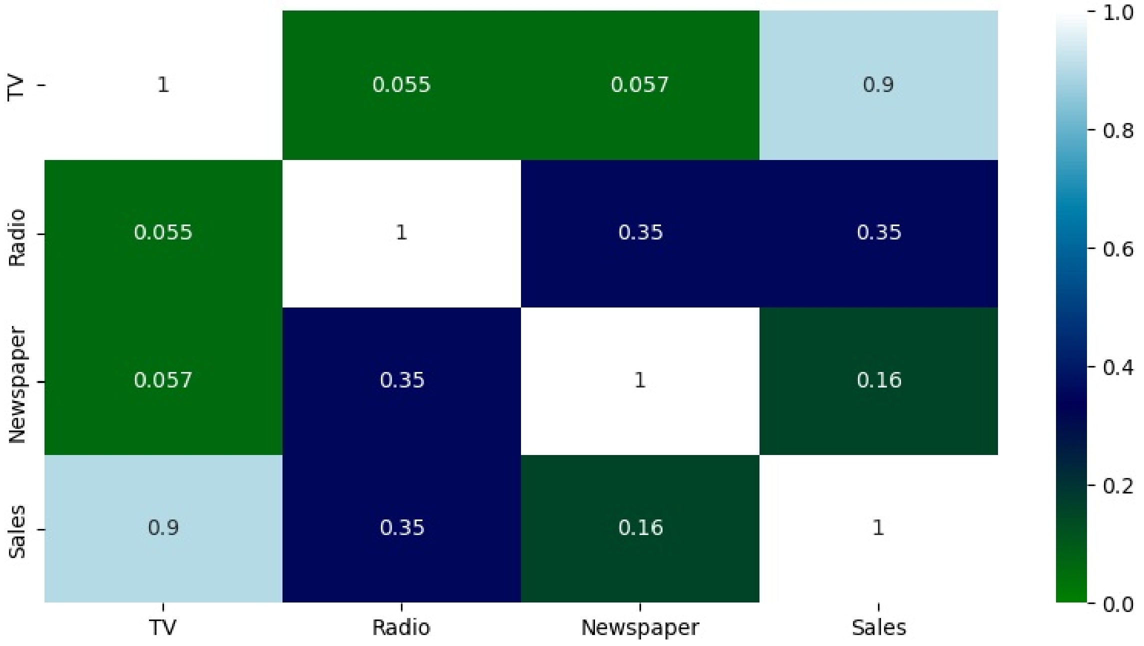
In [31]:

```python
import seaborn as sns
import matplotlib.pyplot as plt

sns.pairplot(df, x_vars=['TV', 'Radio', 'Newspaper'], y_vars='Sales', height=3, aspect=1)
plt.show()
```



In [37]:

```python
plt.figure(figsize=(10, 5))
sns.heatmap(df.corr(), annot=True, vmin=0, vmax=1, cmap='ocean')
plt.show()
```



In [39]:
```
#Staristical
df.std()
```

Out[39]:
```
TV           85.854236
Radio        14.846809
Newspaper    21.778621
Sales         5.283892
dtype: float64
```

In [40]:
```
#correlation
df.corr()
```

Out[40]:

| | TV | Radio | Newspaper | Sales |
|---|---|---|---|---|
| **TV** | 1.000000 | 0.054809 | 0.056648 | 0.901208 |
| **Radio** | 0.054809 | 1.000000 | 0.354104 | 0.349631 |
| **Newspaper** | 0.056648 | 0.354104 | 1.000000 | 0.157960 |
| **Sales** | 0.901208 | 0.349631 | 0.157960 | 1.000000 |

In [41]:
```
#variance
df.var()
```

Out[41]:
```
TV           7370.949893
Radio         220.427743
Newspaper     474.308326
Sales          27.919517
dtype: float64
```

In [42]:
```
#mean
df.mean()
```

Out[42]:
```
TV           147.0425
Radio         23.2640
Newspaper     30.5540
Sales         15.1305
dtype: float64
```

In [43]:
```
#median
df.median()
```

Out[43]:
```
TV           149.75
Radio         22.90
Newspaper     25.75
Sales         16.00
dtype: float64
```

In [60]:
```
#Linear regression
X = df[['TV']]
```

In [53]:
```
Y= df['Sales']
```

In [54]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,train_size=0.8,random_state=50)
```

In [56]:
```
from sklearn.linear_model import LinearRegression

# Create an instance of LinearRegression
lr = LinearRegression()
```

```
# Fit the model to your training data
lr.fit(X_train, Y_train)
```

Out[56]:
```
LinearRegression()
```

In [57]:
```
lr.intercept_
```

Out[57]:
```
6.889929307794299
```

In [59]:
```
lr.coef_
```

Out[59]:
```
array([0.05671244])
```

In [61]:
```
print("The Lr Model is Y = ",lr.intercept_, "+",lr.coef_,"Radio" )
```

```
The Lr Model is Y =  6.889929307794299 + [0.05671244] Radio
```

In [62]:
```
lr.score(X_train,Y_train)
```

Out[62]:
```
0.822322146620674
```

In [63]:
```
lr.score(X_test,Y_test)
```

Out[63]:
```
0.7281236097879917
```

In [64]:
```
Y_pred = lr.predict(X_test)
```

In [65]:
```
Y_pred #Linear Regression output for test and train data.
```

Out[65]:
```
array([16.85430492, 20.18899637,  8.23968537, 14.22851898, 12.34566599,
       19.27592609, 20.05288651,  7.37765629, 19.82036551, 11.89763771,
       20.48957229, 16.9280311 , 19.18518619, 16.67282512, 21.79962965,
       14.55745113, 13.75213448, 19.83737924,  9.32856421, 14.35895759,
       13.11695516, 11.160376  , 10.42311429, 19.50277585, 22.41212399,
       10.76906017, 12.01673384, 12.31730977, 17.37605937, 18.61806179,
       13.87123061, 15.38545274,  7.84836953, 18.09630735, 11.22275968,
        9.42497535, 12.71429684, 21.78828716, 11.83525403, 11.22275968])
```

In [66]:
```
#difference between actual data and predicted data
diff = pd.DataFrame({'Actual': Y_test,'Predicted':Y_pred})
```

In [67]:
```
diff.head(5)
```

Out[67]:

| | Actual | Predicted |
|---|---|---|
| **112** | 17.1 | 16.854305 |
| **165** | 16.9 | 20.188996 |
| **12** | 9.2 | 8.239685 |
| **73** | 11.0 | 14.228519 |
| **144** | 12.3 | 12.345666 |

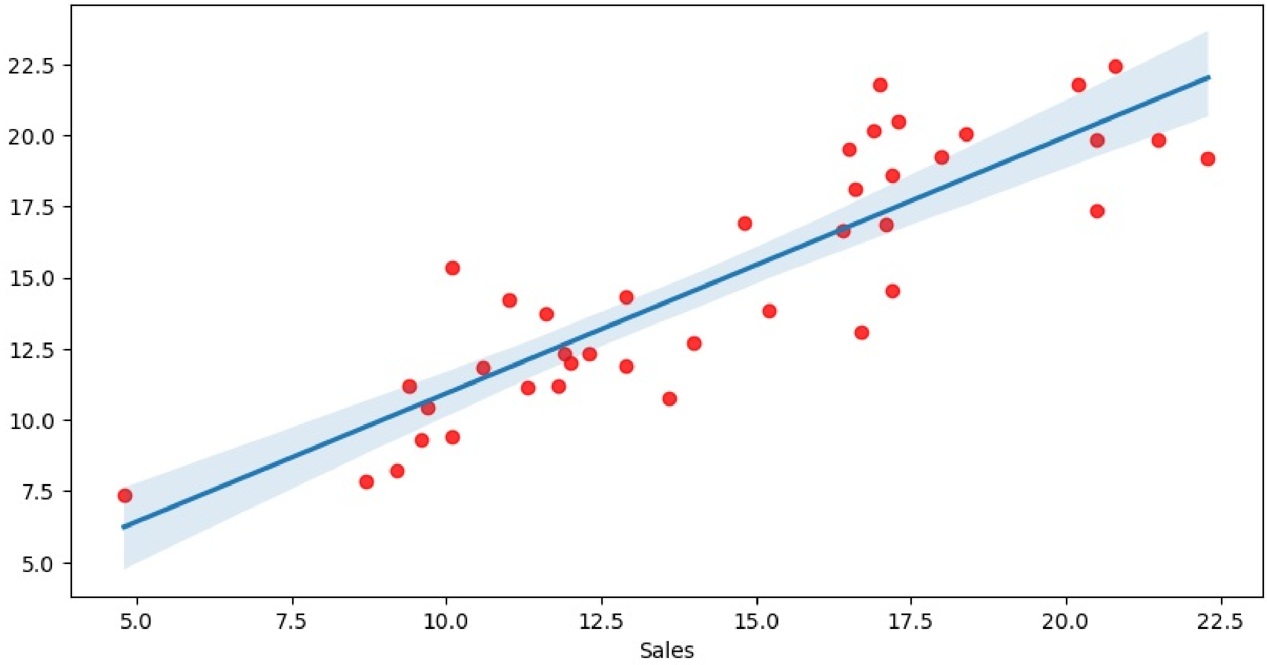
In [68]:
```
from sklearn import metrics
from sklearn.metrics import r2_score
```

In [69]:
```
R2=r2_score (Y_test,Y_pred)
mae = metrics.mean_absolute_error(Y_test,Y_pred)
mse = metrics.mean_squared_error(Y_test,Y_pred)
rmse = np.sqrt(metrics.mean_squared_error(Y_test,Y_pred))
```

In [71]:
```
print('Accurancy = ',R2.round(2)*100,'%')
print('mae = ',mae.round(2))
print('mse = ',mse.round(2))
print('rmse = ',rmse.round(2))
```

```
Accurancy =  73.0 %
mae =  1.74
mse =  4.66
rmse =  2.16
```

In [75]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 5))
sns.regplot(x=Y_test, y=Y_pred, scatter_kws={'color': 'red'})
plt.show()

#egression Graph
```



```
In [ ]:
```

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