sales-prediction

September 27, 2023

1 Task - 5 Sales Prediction

[2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

[3]:

```
df = pd.read_csv("Advertising.csv")
```

[4]:

```
df
```

[4]:

```
     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
..          …    …    …        …    …
195         196   38.2    3.7       13.8    7.6
196         197   94.2    4.9        8.1    9.7
197         198  177.0    9.3        6.4   12.8
198         199  283.6   42.0       66.2   25.5
199         200  232.1    8.6        8.7   13.4

[200 rows x 5 columns]
```

[5]:

```
df.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
```

[6]:
```
df.drop("Unnamed: 0",axis=1,inplace=True)
```

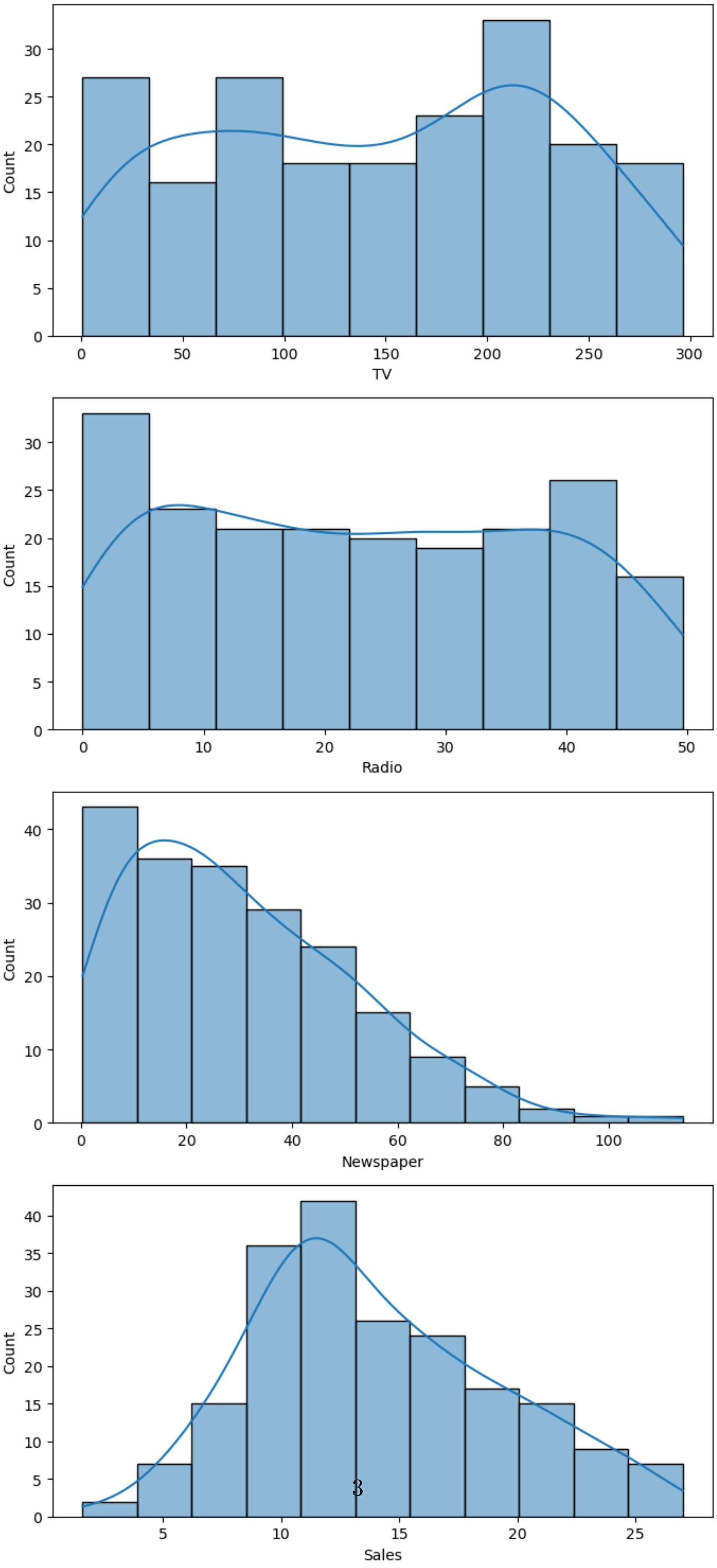
[7]:
```
df.isnull().sum()
```

[7]:
```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

2 Performing EDA

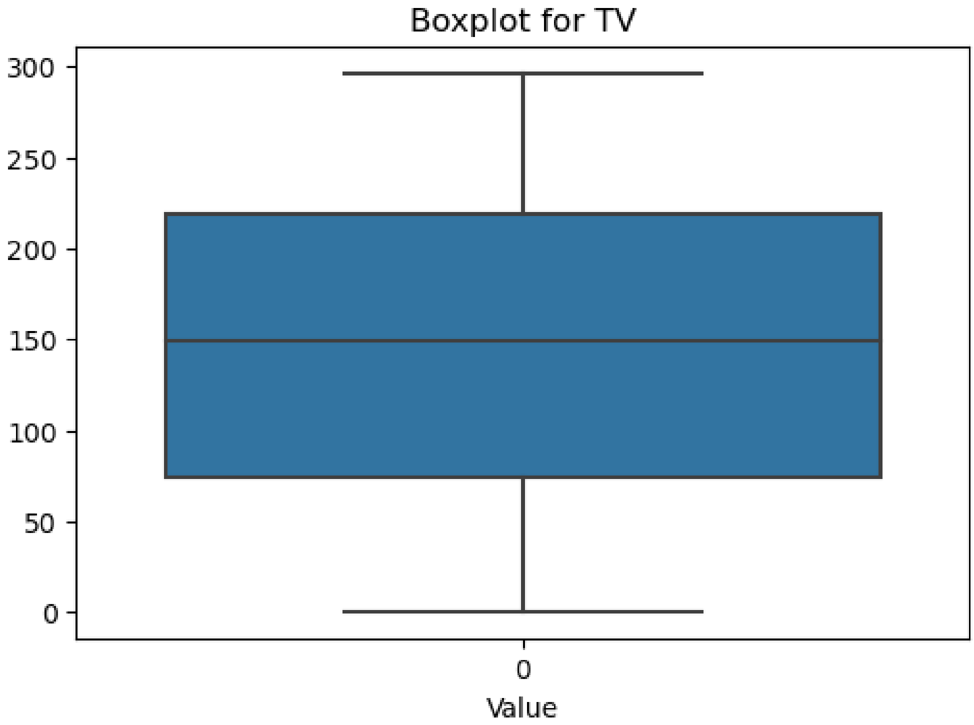
Univarite Analysis

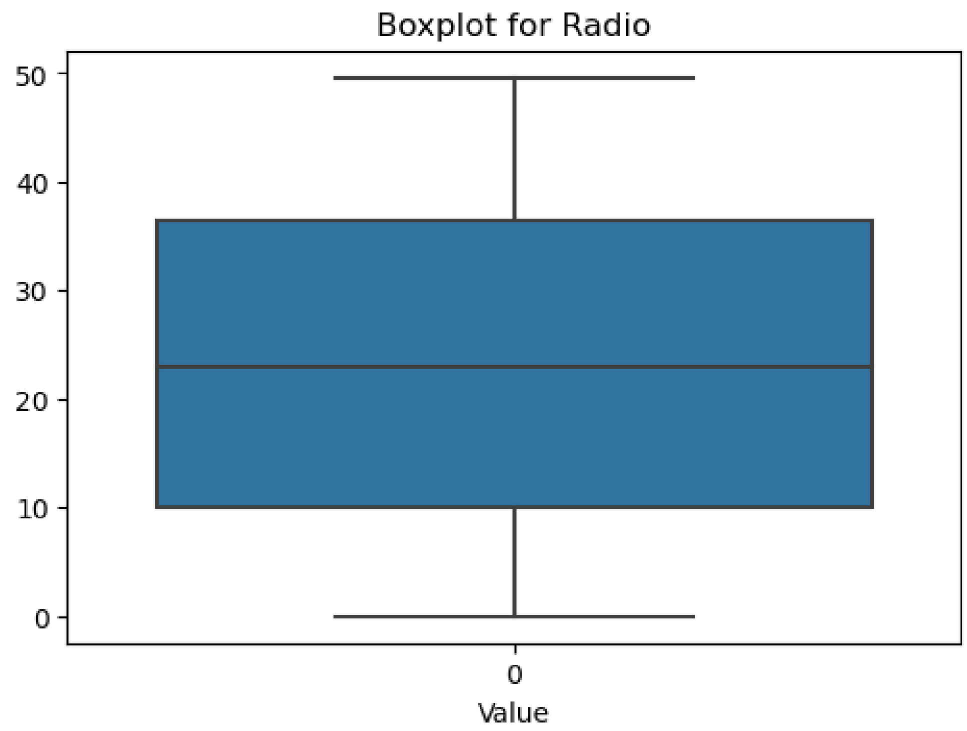
[8]:
```
#creating subplot
columns=['TV','Radio','Newspaper','Sales']
plt.figure(figsize=(7,15))  # Adjust the figure size as needed
for i in columns:
    plt.subplot(4,1, columns.index(i) + 1)
    sns.histplot(df[i],kde=True)
plt.tight_layout()
```

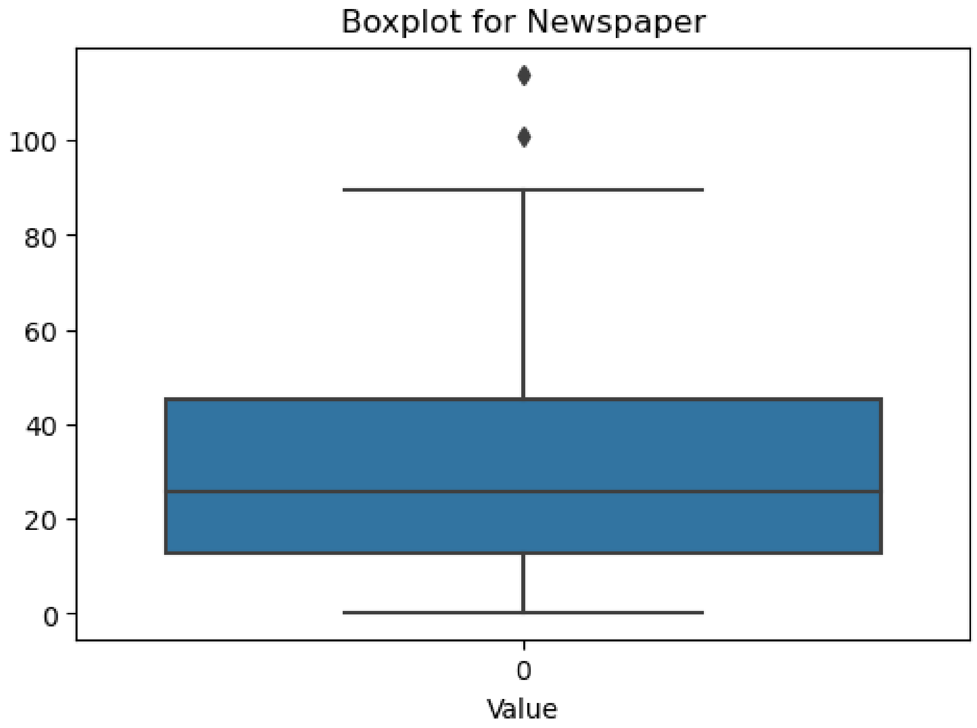

Count
0
5
10
15
20
25
30
0
50
100
150
200
250
300
TV
Count
0
5
10
15
20
25
30
0
10
20
30
40
50
Radio
Count
0
10
20
30
40
0
20
40
60
80
100
Newspaper
Count
0
5
10
15
20
25
30
35
40
5
10
15
20
25
Sales

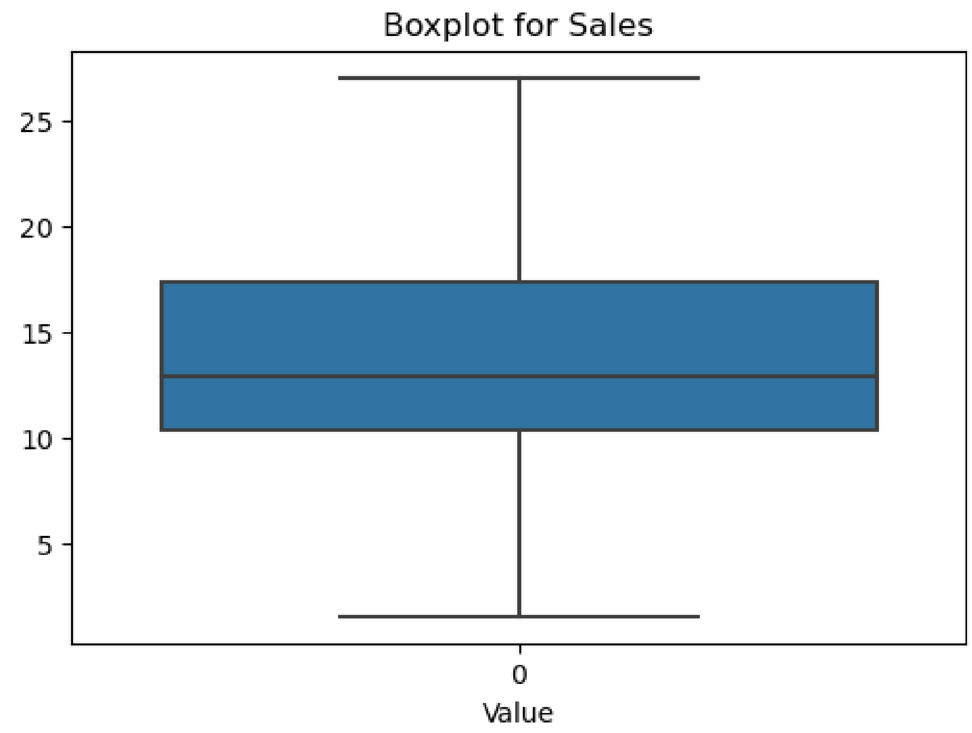
[25]:

```
#Creating Box-Plot
for column in df.columns:
    plt.figure(figsize=(6, 4))  # Adjust the figure size as needed
    sns.boxplot(data=df[column])
    plt.title(f'Boxplot for {column}')
    plt.xlabel('Value')
    plt.show()
```




Boxplot for Radio
50
40
30
20
10
0
0
Value


Boxplot for Newspaper
100
80
60
40
20
0
0
Value

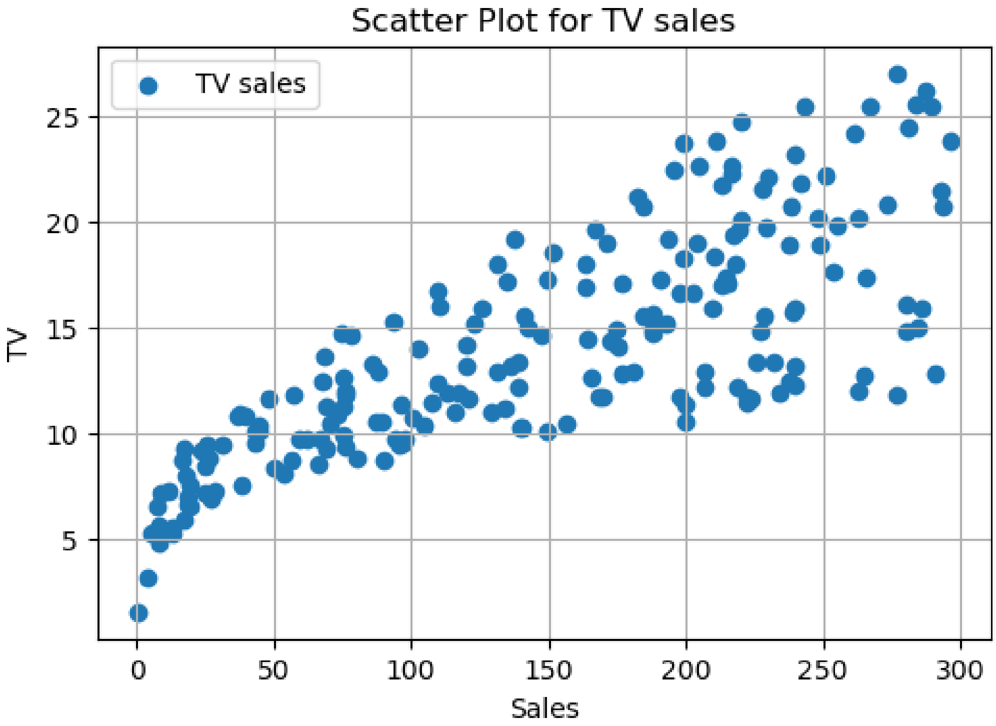


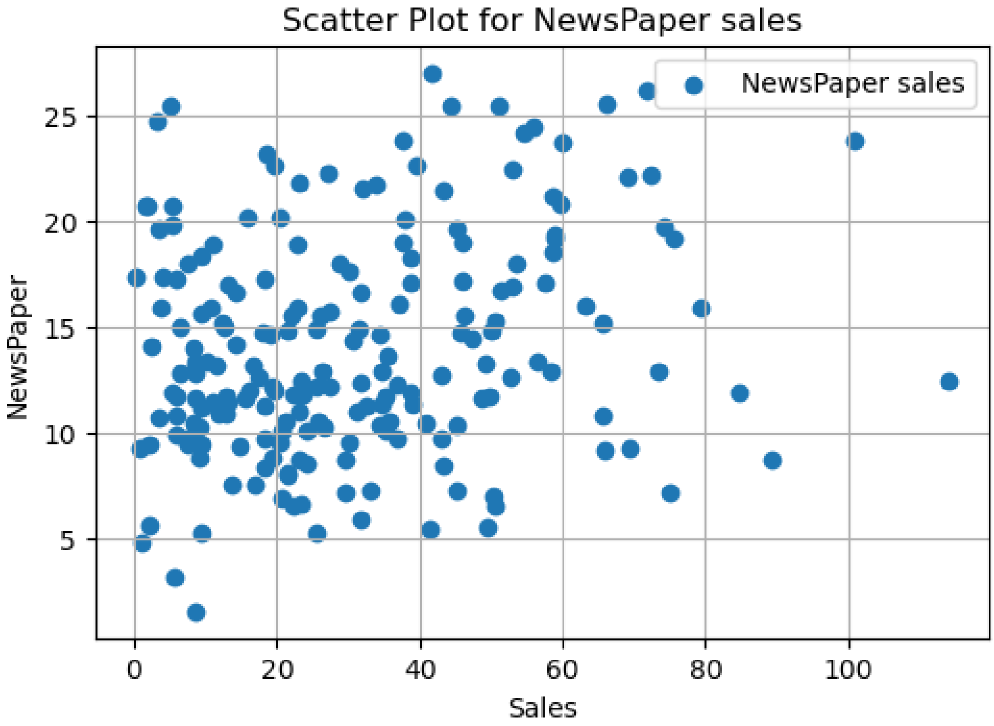
3 Bivarite Analysis

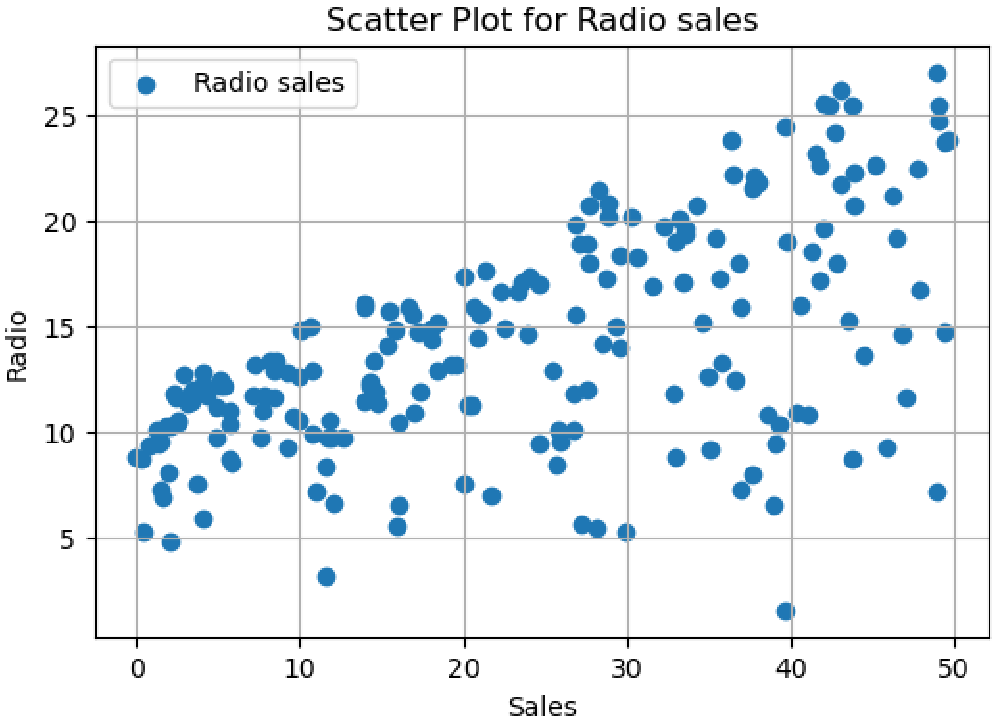
[12]:

```
#Creating Scatter Plot

data_pairs = [
    (df["TV"], df["Sales"],'TV sales','Sales','TV'),
    (df["Newspaper"],df["Sales"],'NewsPaper sales','Sales','NewsPaper'),
    (df["Radio"],df["Sales"],'Radio sales','Sales','Radio')
]

# Create scatter plots using a for loop
for i, (x, y,label,x_label,y_label) in enumerate(data_pairs):
    plt.figure(figsize=(6, 4))  # Adjust figure size as needed
    plt.scatter(x, y, label=label)
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.title(f'Scatter Plot for {label}')
    plt.legend()
    plt.grid(True)
    plt.show()
```

```
# Create sample data (replace this with your own data)
```




Scatter Plot for NewsPaper sales
NewsPaper sales
NewsPaper
Sales
5
10
15
20
25
0
20
40
60
80
100

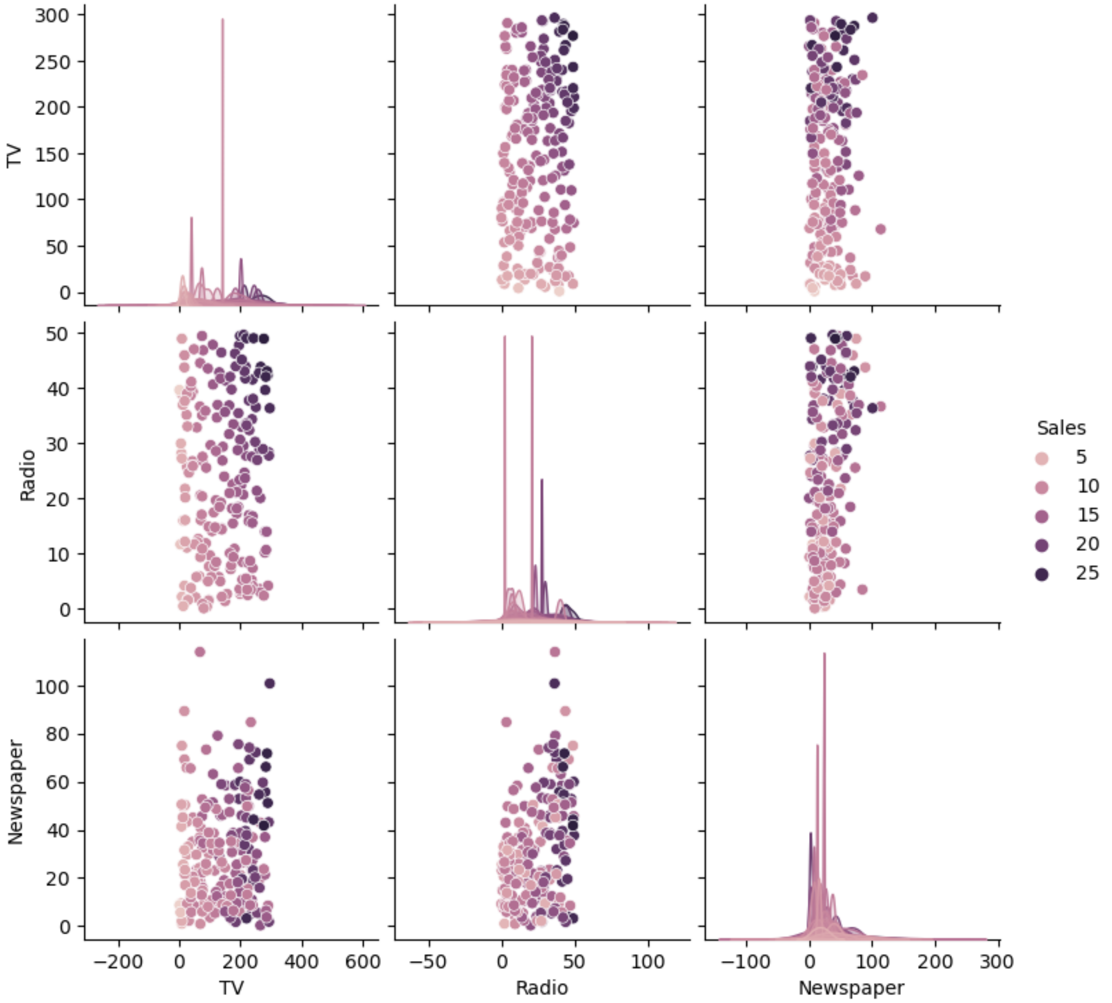

Scatter Plot for Radio sales
Radio sales
Radio
Sales
5
10
15
20
25
0
10
20
30
40
50

[14]:

```
#Create PairPlot

sns.pairplot(df,hue='Sales')
```

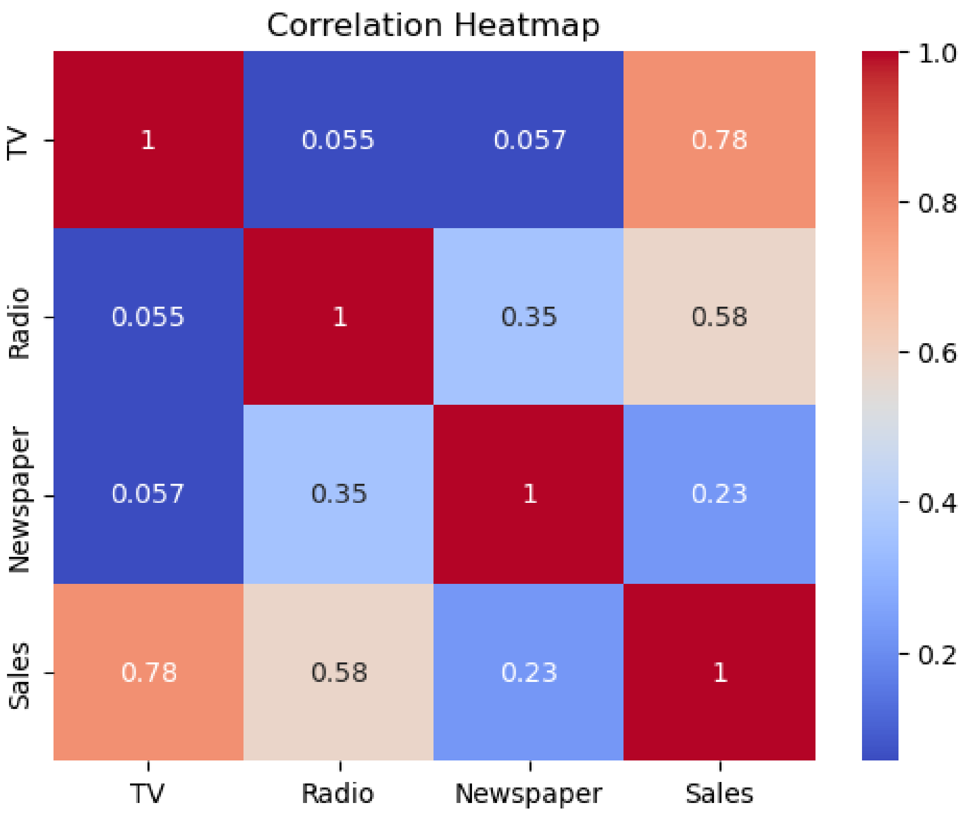
[14]:

```
<seaborn.axisgrid.PairGrid at 0x1e1dfb5b050>
```



[15]:

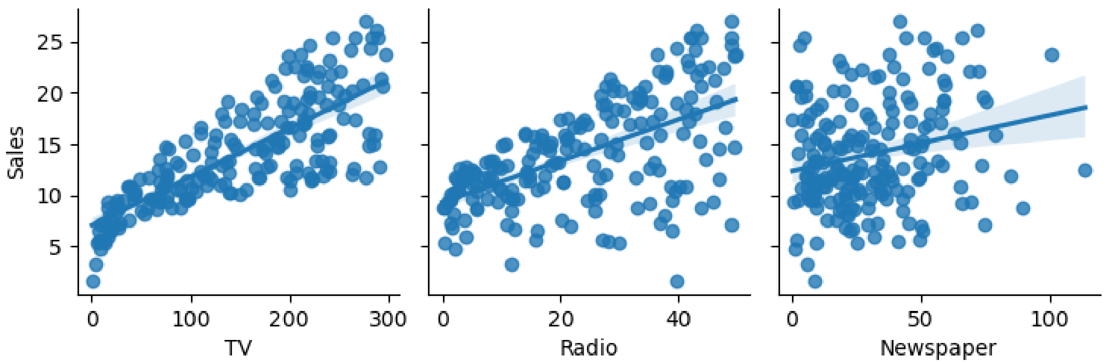
```
# Calculate the correlation matrix
correlation_matrix = df.corr()

# Create a heatmap of the correlation matrix
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



```
[16]: df.corr()*100
```

```
[16]:                   TV       Radio   Newspaper       Sales
      TV         100.000000    5.480866    5.664787   78.222442
      Radio        5.480866  100.000000   35.410375   57.622257
      Newspaper    5.664787   35.410375  100.000000   22.829903
      Sales       78.222442   57.622257   22.829903  100.000000
```

```
[17]: sns.pairplot(data = df,x_vars=["TV", "Radio", "Newspaper"], y_vars="Sales",␣
       ↪kind="reg",palette="pastel")
      plt.show()
```



4 Splitting Data Into Train and Test

[18]:
```
x=df[['TV','Radio','Newspaper']]
y=df['Sales']
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7,␣
  ↪random_state=10)
```

5 Applying Model for Prediction

[19]:
```
model=LinearRegression()
```

[20]:
```
model.fit(x_train,y_train)

y_predict=model.predict(x_test)
```

[21]:
```
print(model.intercept_)
print(model.coef_)
```

```
3.410641588611826
[ 0.04303172  0.19352212 -0.00386729]
```

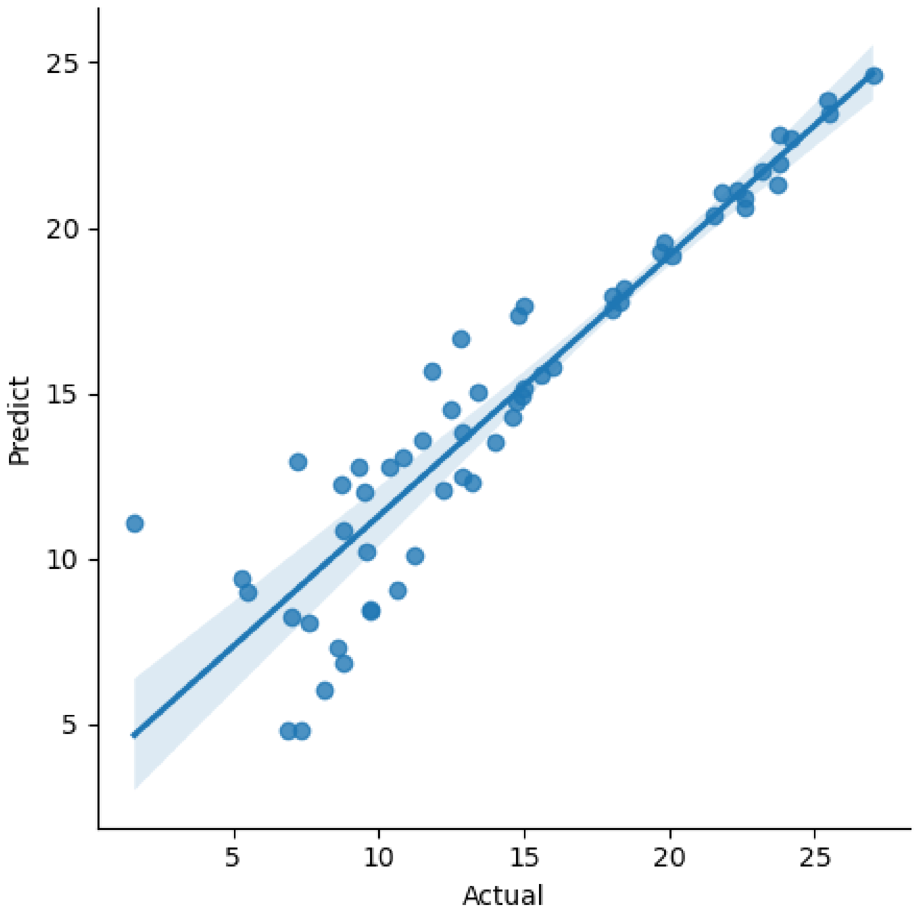
[22]:
```
act_predict=pd.DataFrame({
    'Actual':y_test.values.flatten(),
    'Predict':y_predict.flatten()
})
act_predict.sample(20)
```

[22]:
```
    Actual    Predict
51    12.9  13.837394
35    11.8  15.670965
0     18.4  18.150362
```

```
28    23.2  21.693574
55     8.1   6.017123
45     8.8  10.875483
24    14.8  17.339943
5     14.6  14.300527
47    13.4  15.028949
8     22.3  21.130349
48     5.5   9.002639
41    13.2  12.331227
2     18.0  17.962819
42    15.0  15.181345
31    21.5  20.393904
15    14.0  13.525769
7     23.7  21.297606
44    23.8  21.934629
17     5.3   9.392972
58    11.2  10.102094
```

[23]:
```
sns.lmplot(data=act_predict,x='Actual',y="Predict")
```

[23]:
```
<seaborn.axisgrid.FacetGrid at 0x1e1e07f0110>
```



```
[24]: print("Mean_absolute_error:",mean_absolute_error(y_test,y_predict))
      print("Mean_squared_error:",mean_squared_error(y_test,y_predict))
      print("Squre_Mean_absolute_error:",np.
       ↪sqrt(mean_absolute_error(y_test,y_predict)))
      print("r2_score:",r2_score(y_test,y_predict)*100)
```

```
Mean_absolute_error: 1.7091210770106202
Mean_squared_error: 5.374051588095294
Squre_Mean_absolute_error: 1.3073335752632609
r2_score: 85.67790678044409
```

6 ThankYou

```
[ ]:
```