In [1]:
```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import matplotlib.pyplot as plt
```

In [2]:
```
# Load your dataset (replace 'your_dataset.csv' with your actual dataset)
# The dataset should have columns like 'advertising_expenses', 'target_audience', 'sal
df = pd.read_csv(r"C:\Users\warul\Downloads\archive (4).zip")

# Explore the dataset
print(df.head())
```

```
   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 [3]:
```
X = df[['TV', 'Radio', 'Newspaper']]  # Features
y = df['Sales']
```

In [4]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

In [5]:
```
model = LinearRegression()
```

In [6]:
```
model.fit(X_train, y_train)
```

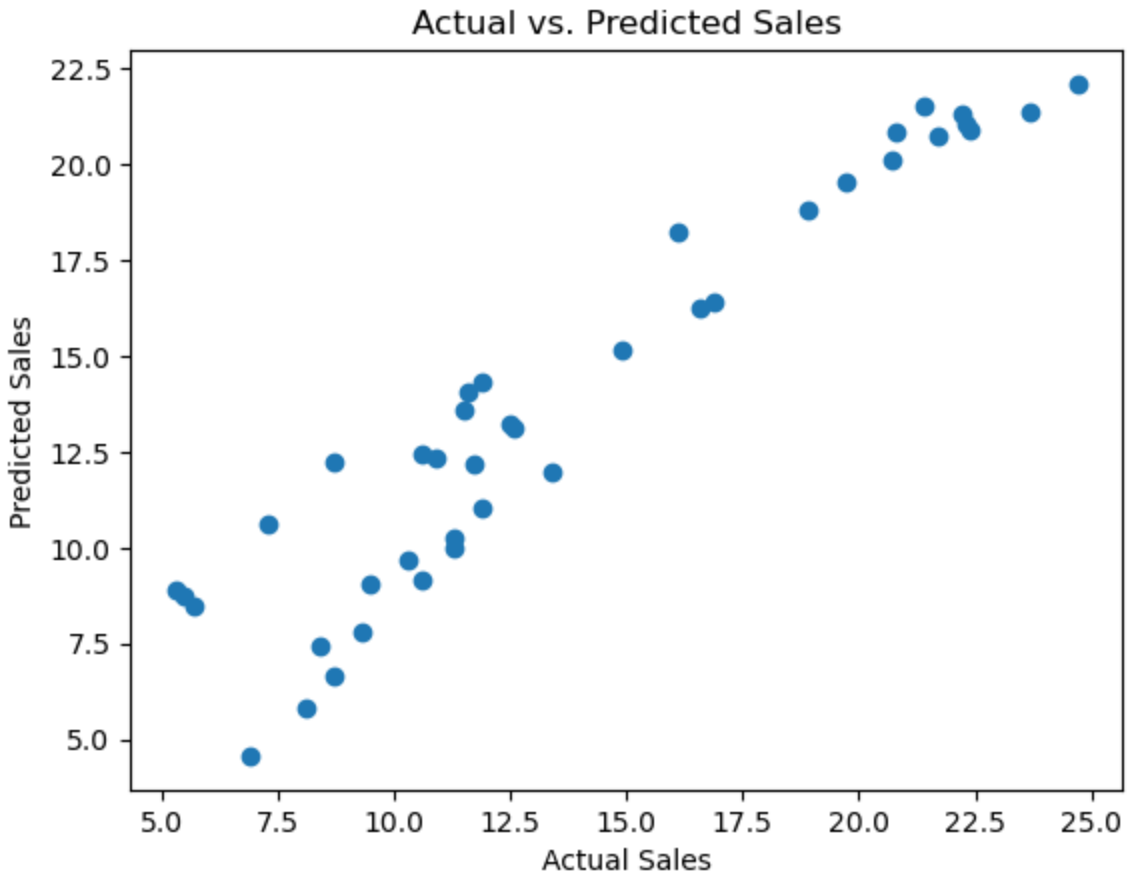
Out[6]:
▾ LinearRegression

LinearRegression()

In [9]:
```
y_pred = model.predict(X_test)
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred, squared=F
```

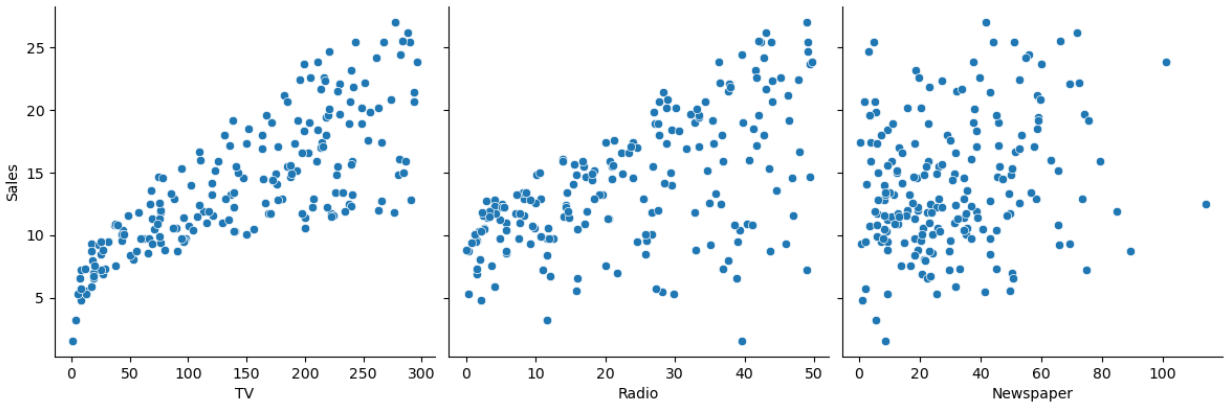
```
Mean Absolute Error: 1.4607567168117606
Mean Squared Error: 3.1740973539761046
Root Mean Squared Error: 1.7815996615334502
```

In [10]:
```
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Sales')
plt.ylabel('Predicted Sales')
plt.title('Actual vs. Predicted Sales')
plt.show()
```



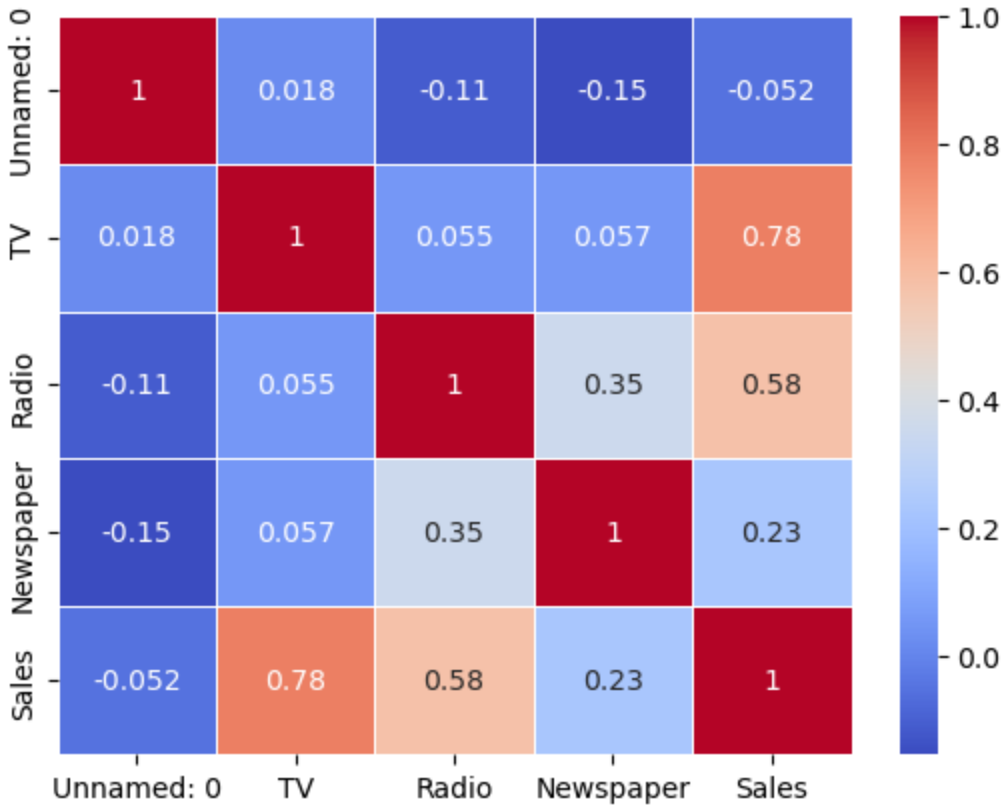
In [11]:
```
import seaborn as sns

# Pair plot
sns.pairplot(df, x_vars=['TV', 'Radio', 'Newspaper'], y_vars='Sales', height=4, aspect
plt.show()
```



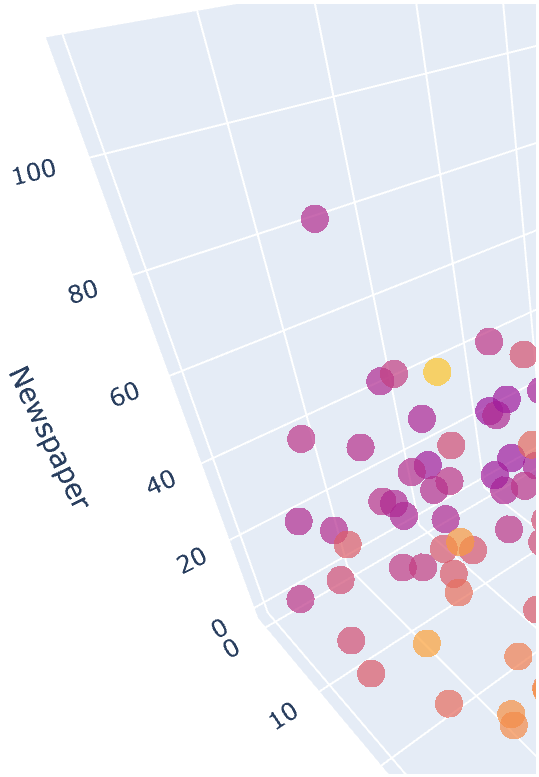
In [12]:
```
# Correlation matrix
corr = df.corr()

# Heatmap
sns.heatmap(corr, annot=True, cmap='coolwarm', linewidths=.5)
plt.show()
```

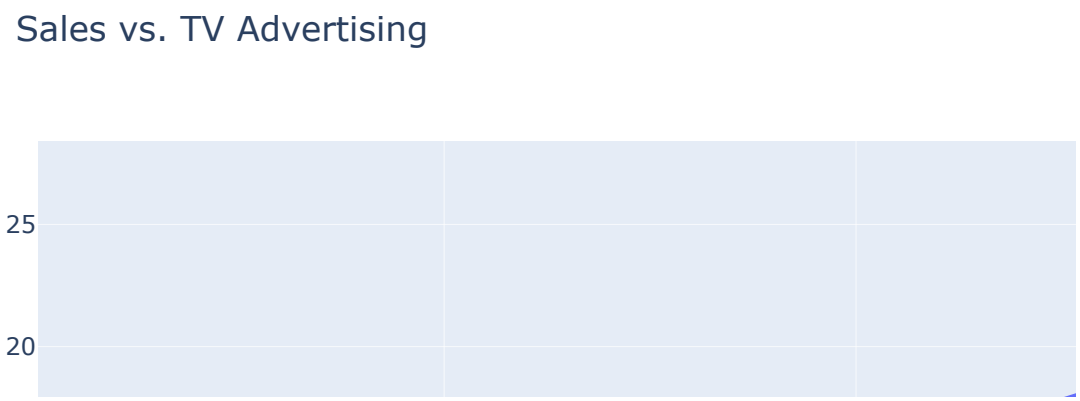


In [13]:
```
import plotly.express as px

fig = px.scatter_3d(df, x='TV', y='Radio', z='Newspaper', color='Sales', opacity=0.7)
fig.show()
```



In [14]:
```
fig = px.line(df, x='TV', y='Sales', title='Sales vs. TV Advertising')
fig.show()
```



In [ ]: