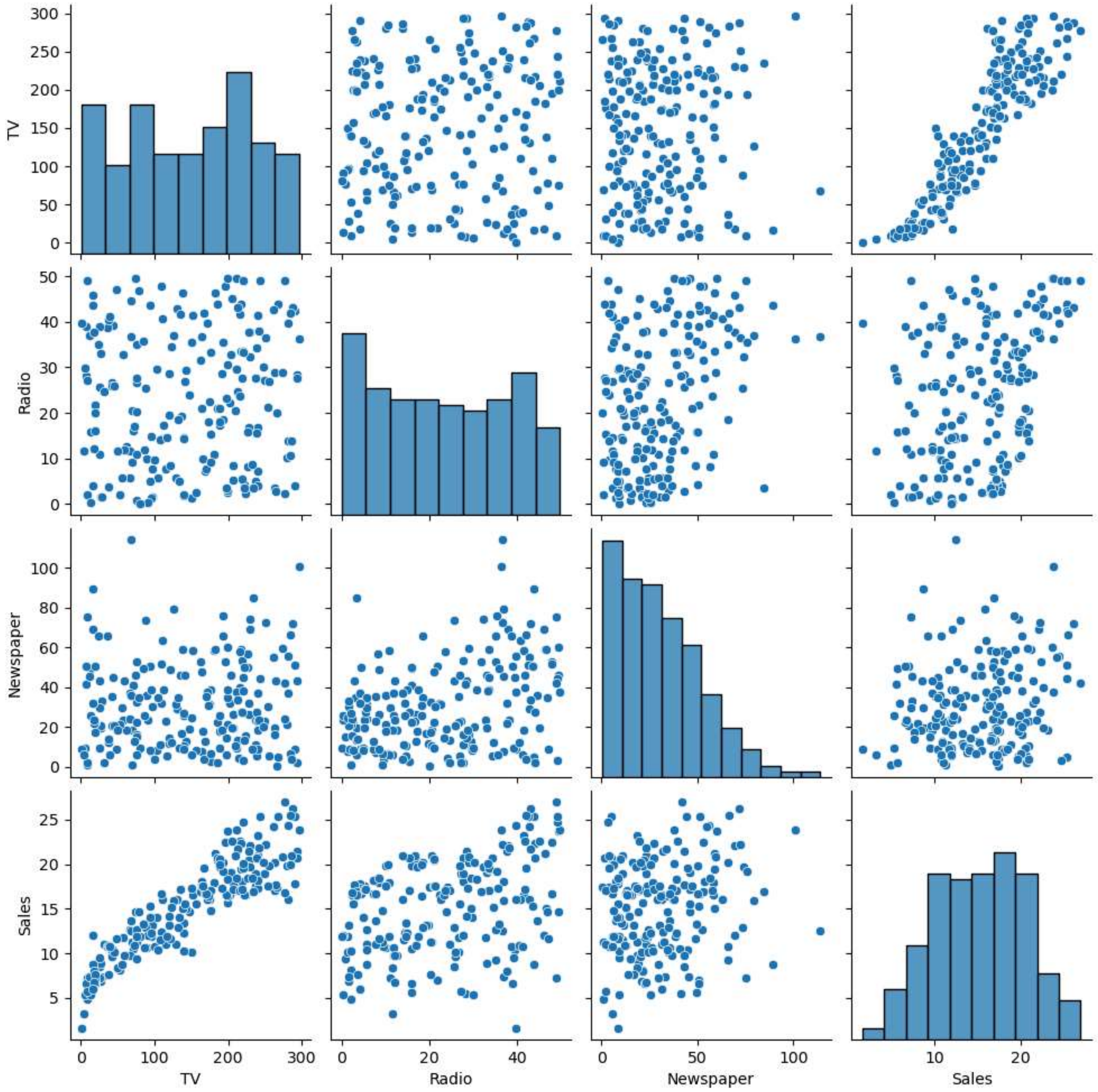
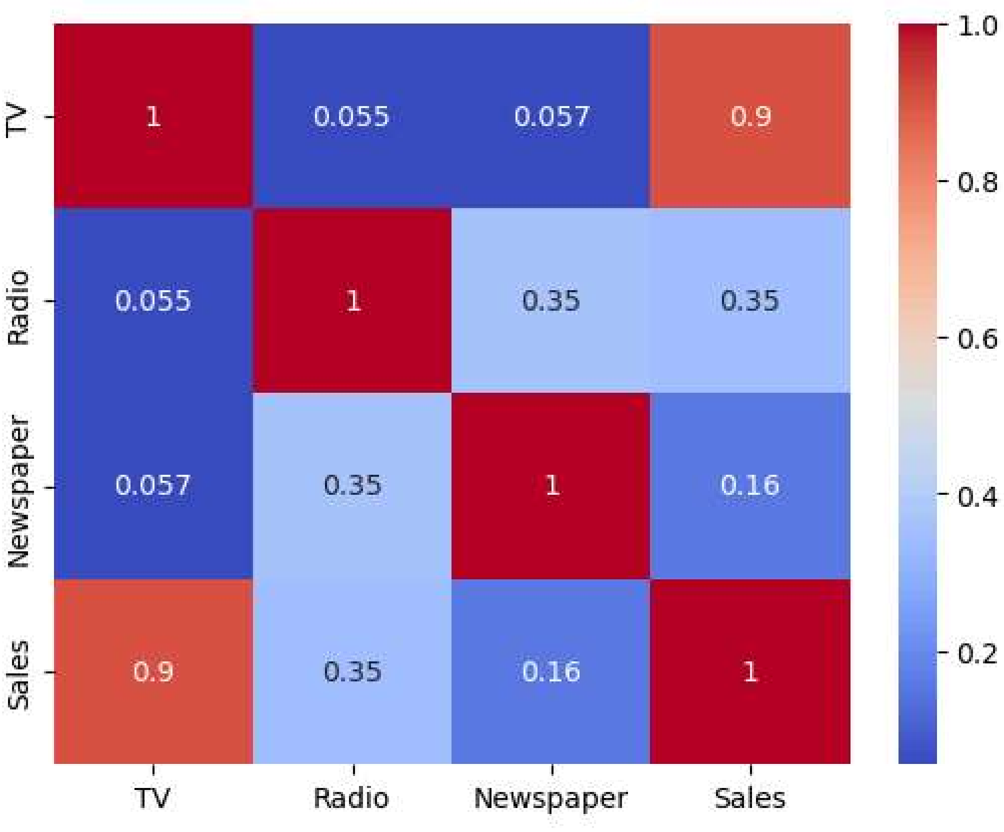
In [4]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
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

print("\033[94mCreated by Lettie Ngobeni\033[0m")


data = pd.read_csv('advertising.csv')
print("Dataset Preview:")
print(data.head())

print("\nSummary Statistics:")
print(data.describe())

#DATA CLEANING
print("\nMissing Values Check:")
print(data.isnull().sum())
print("\nData Visualizations:")

# Pairplot visualization
sns.pairplot(data)
plt.show()

#  heatmap visualization
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.show()

scaler = StandardScaler()
scaled_data = scaler.fit_transform(data.drop('Sales', axis=1))
scaled_data = pd.DataFrame(scaled_data, columns=data.columns[:-1])

scaled_data['Sales'] = data['Sales']

X = scaled_data.drop('Sales', axis=1)
y = scaled_data['Sales']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Model training
model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("\nModel Performance Metrics:")
print(f'Mean Absolute Error (MAE): {mae}')
print(f'Mean Squared Error (MSE): {mse}')
print(f'R-squared Score: {r2}')

new_data = [[230.1, 37.8, 69.2]]  # Sample new data
scaled_new_data = scaler.transform(new_data)
sales_prediction = model.predict(scaled_new_data)
print(f'\nPredicted Sales for the provided data: {sales_prediction[0]}')
```

```
Created by Lettie Ngobeni
Dataset Preview:
      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

Summary Statistics:
               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

Missing Values Check:
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64

Data Visualizations:
```

```
C:\Users\nhlav\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: T
he figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```


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


TV
Radio
Newspaper
Sales
1
0.055
0.057
0.9
0.055
1
0.35
0.35
0.057
0.35
1
0.16
0.9
0.35
0.16
1
TV
Radio
Newspaper
Sales
1.0
0.8
0.6
0.4
0.2

```
Model Performance Metrics:
Mean Absolute Error (MAE): 1.2748262109549349
Mean Squared Error (MSE): 2.9077569102710927
R-squared Score: 0.9059011844150825

Predicted Sales for the provided data: 21.372540280396883
```

```
C:\Users\nhlav\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X doe
s not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
C:\Users\nhlav\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X doe
s not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

In [ ]: