

```
In [17]: ► import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

-----  
**ModuleNotFoundError** Traceback (most recent call last)

Cell In[17], line 8

```
    6 from sklearn.linear_model import LinearRegression
    7 from sklearn.metrics import mean_squared_error, r2_score
----> 8 from sklearn.model.predict import (new_advertising_spending)
```

**ModuleNotFoundError**: No module named 'sklearn.model'

```
In [3]: ▶ # Load the dataset
data = pd.read_csv('Advertising.csv')

# Explore the dataset to understand its structure and missing values
print(data.head())
print(data.info())
```

```
   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
<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
None
```

```
In [4]: ▶ # Check for missing values
print(data.isnull().sum())

# If there are missing values, handle them appropriately, e.g., drop rows with missing values
data.dropna(inplace=True)
```

```
Unnamed: 0    0
TV            0
Radio         0
Newspaper     0
Sales         0
dtype: int64
```

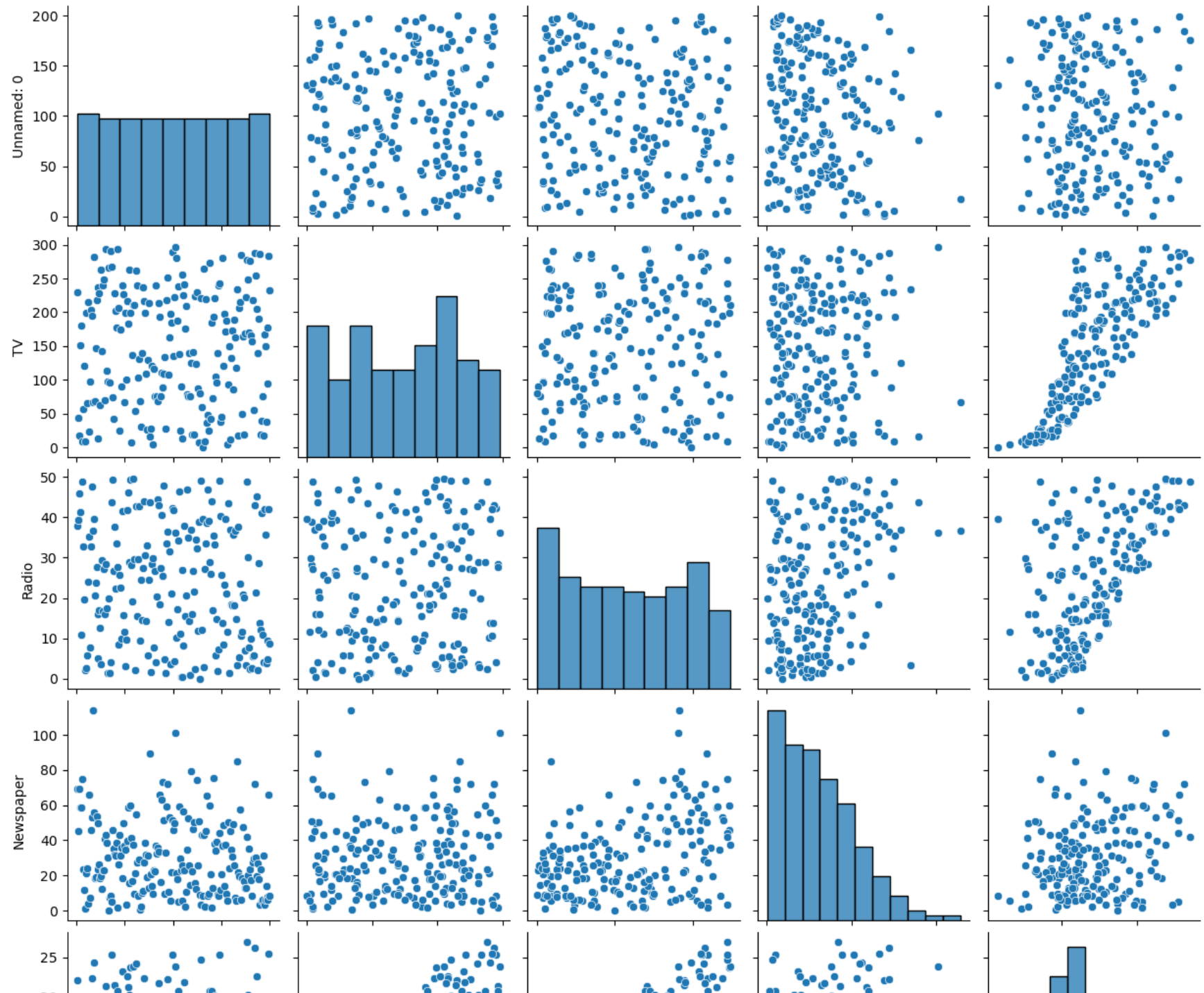
In [7]: ▶

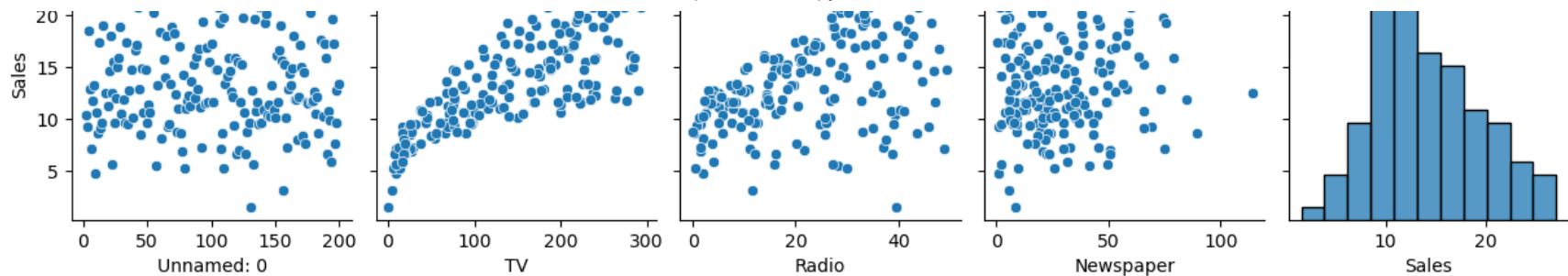
```
-----  
NameError                                Traceback (most recent call last)  
Cell In[7], line 1  
----> 1 car_dat.shape  
  
NameError: name 'car_dat' is not defined
```

```
In [8]: ▶ # You can plot and visualize the relationships between features and the target variable 'Sales'  
sns.pairplot(data)  
plt.show()
```

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







```
In [9]: ▶ # Split the data into features (X) and target variable (y)
X = data[['TV', 'Radio', 'Newspaper']]
y = data['Sales']

# Split the data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [10]: ▶ # Create a linear regression model and fit it on the training data
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Squared Error: 3.1740973539761046
R-squared: 0.899438024100912
```

```
In [18]: ► new_advertising_spending = np.array([[200, 20, 40]]) # Example values for TV, Radio, Newspaper spending
predicted_sales = model.predict(new_advertising_spending)

print("Predicted Sales:", predicted_sales[0])
```

Predicted Sales: 15.819316490208111

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names  
warnings.warn(

In [ ]: ►