Advertisement Dataset

```
# Import libraries
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 pandas.core.common import random_state
from sklearn.linear_model import LinearRegression
```

```
# Get dataset
df_adv = pd.read_csv('/content/Advertising.csv')
df_adv.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

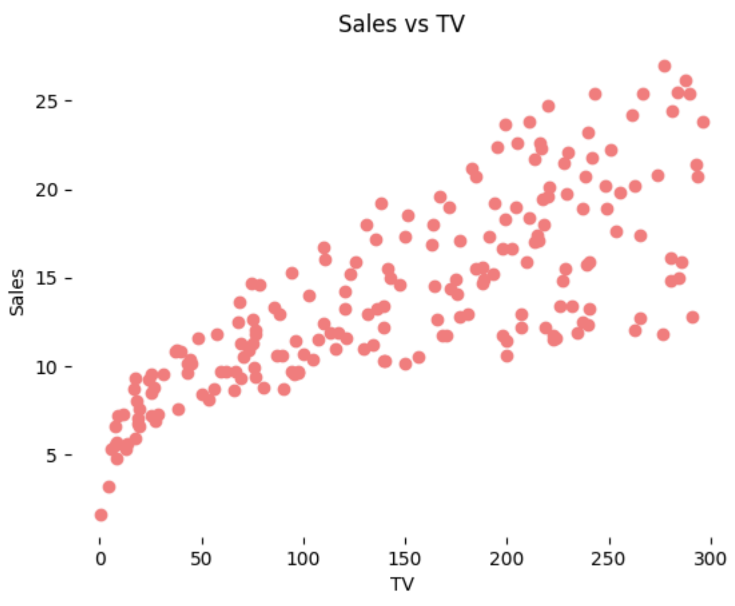
Next steps: Generate code with df_adv View recommended plots

'Sales' is the target variable that needs to be predicted. Now, based on this data, our objective is to create a predictive model, that predicts sales based on the money spent on different platforms for marketing.

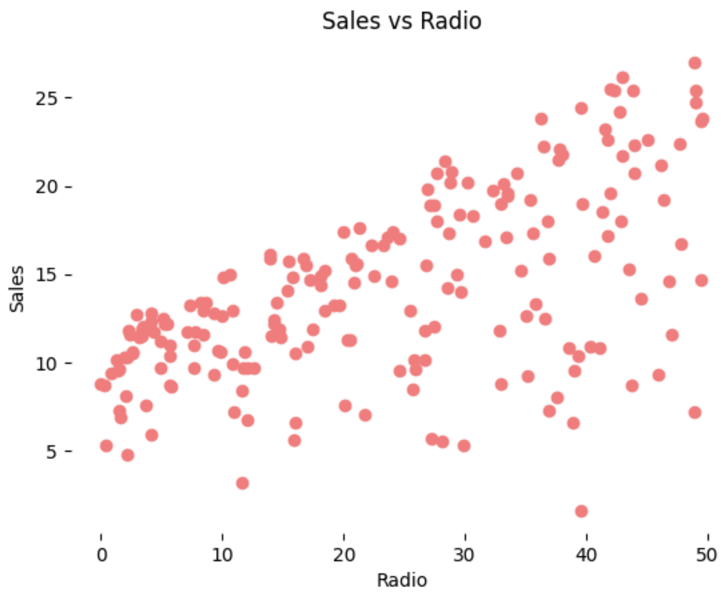
** Data Visualization**

Let us plot the scatter plot for target variable vs. predictor variables

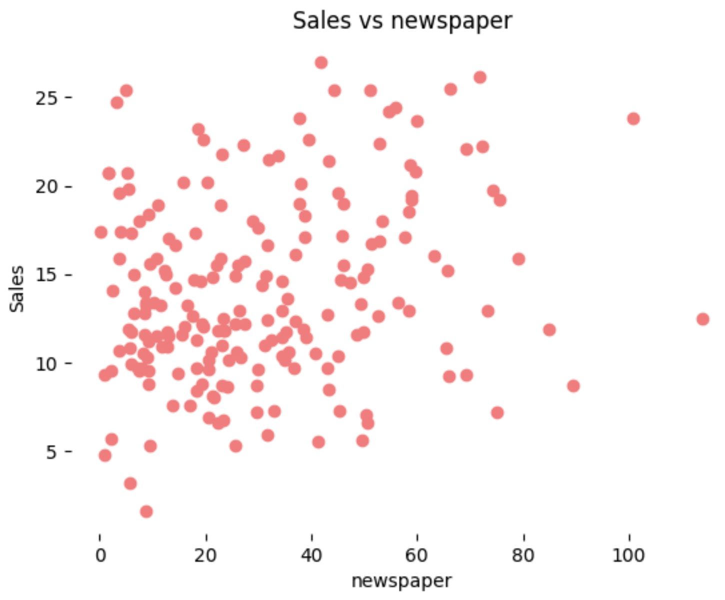
```
# Relationship between Sales and TV
plt.scatter(df_adv['TV'], df_adv['sales'], color='lightcoral')
plt.title('Sales vs TV')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.box(False)
plt.show()
```



```
# Relationship between Sales and Radio
plt.scatter(df_adv['radio'], df_adv['sales'], color='lightcoral')
plt.title('Sales vs Radio')
plt.xlabel('Radio')
plt.ylabel('Sales')
plt.box(False)
plt.show()
```

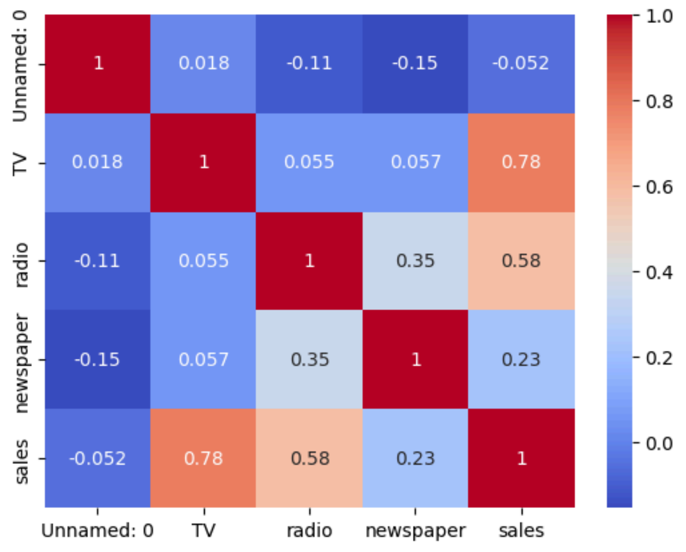


```
# Relationship between Sales and Newspaper
plt.scatter(df_adv['newspaper'], df_adv['sales'], color='lightcoral')
plt.title('Sales vs newspaper')
plt.xlabel('newspaper')
plt.ylabel('Sales')
plt.box(False)
plt.show()
```



Plotting a heatmap for all the variables

```
sns.heatmap(df_adv.corr(), cmap='coolwarm', annot=True)
plt.show()
```



From the scatterplot and the heatmap, we can observe that ‘Sales’ and ‘TV’ have a higher correlation as compared to others because it shows a linear pattern in the scatterplot as well as giving 0.78 correlation.

Performing Simple Linear Regression

As the TV and Sales have a higher correlation we will perform the simple linear regression for these variables.

First assign the feature variable, `TV`, during this case, to the variable `X` and the response variable, `Sales`, to the variable `y`.

```
X = df_adv[ 'TV' ]
y = df_adv[ 'sales' ]
```

Split our variable into training and testing sets. Peforming this by keeping 70% of the data in train dataset and the rest 30% in test dataset.

```
X_train, X_test, y_train, y_test = train_test_split( X, y, train_size = 0.7, test_size = 0.3, random_state = 100 )
```

Check the shapes of train and test sets

```
print( X_train.shape )
print( X_test.shape )
print( y_train.shape )
print( y_test.shape )
```

```
(140,)
(60,)
(140,)
(60,)
```

Train the regression model

Pass the X_train and y_train data into the regressor model by regressor.fit to train the model with our training data

```
regressor.fit(X_train.values.reshape(-1, 1), y_train)
```

```
▾ LinearRegression
LinearRegression()
```

Predict the result

we will predict any value of y (sales) dependent on X (TV) with the trained model using regressor.predict

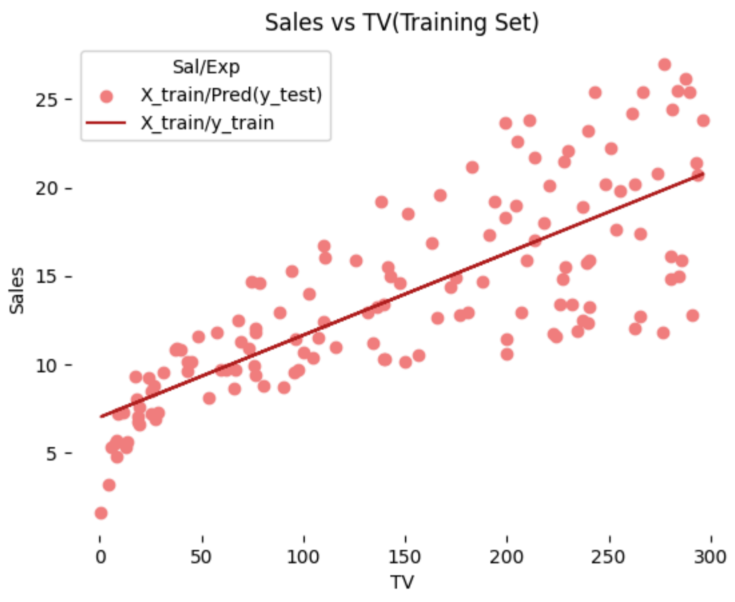
```
# Reshape X_test and X_train into 2D arrays
X_test_2d = X_test.values.reshape(-1, 1)
X_train_2d = X_train.values.reshape(-1, 1)

# Prediction result
y_pred_test = regressor.predict(X_test_2d)     # predicted value of y_test
y_pred_train = regressor.predict(X_train_2d)   # predicted value of y_train
```

Plot the training and test results

Plot training set data vs predictions

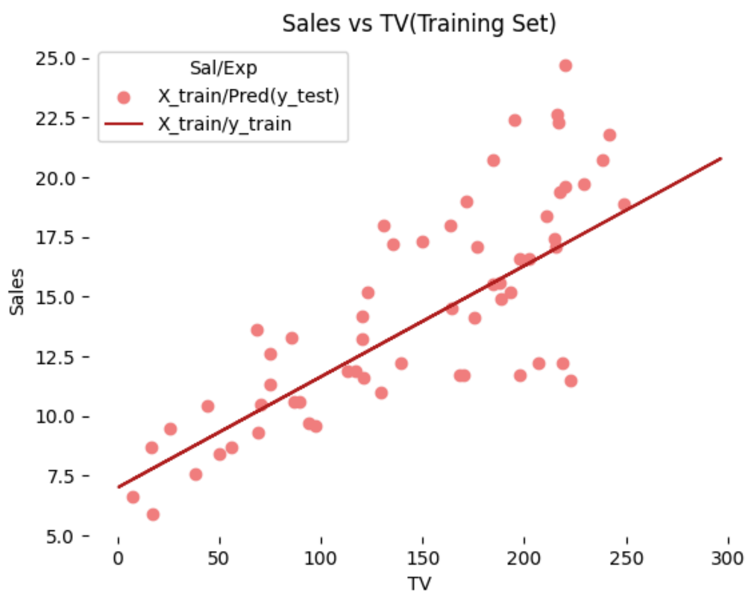
We will plot the result of training sets (X_train, y_train) with X_train and predicted value of y_train (regressor.predict(X_train))

```
# Prediction on training set
plt.scatter(X_train, y_train, color = 'lightcoral')
plt.plot(X_train, y_pred_train, color = 'firebrick')
plt.title('Sales vs TV(Training Set)')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.legend(['X_train/Pred(y_test)', 'X_train/y_train'], title = 'Sal/Exp', loc='best', facecolor='white')
plt.box(False)
plt.show()
```



Plot test set data vs predictions

```
# Prediction on test set
plt.scatter(X_test, y_test, color = 'lightcoral')
plt.plot(X_train, y_pred_train, color = 'firebrick')
plt.title('Sales vs TV(Training Set)')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.legend(['X_train/Pred(y_test)', 'X_train/y_train'], title = 'Sal/Exp', loc='best', facecolor='white')
plt.box(False)
plt.show()
```



Regressor coefficients and intercept

```
# Regressor coefficients and intercept
print(f'Coefficient: {regressor.coef_}')
```