Question 1 - Part B

Supervised Learning Algorithms - Simple Linear Regression (Univariant): Consider any dataset from UCI repository. Create Simple Linear Regression models using the training data set. Predict the scores on the test data and find the error in prediction (E.g. RMSE, MAE, LSE). Include appropriate code snippets to visualize the model. Use Sub-Plots Interpret the result. Write the Inference.

Kaggle Dataset link
Dataset which mam gave

- In dataset given by mam code remains same only before doing anything on the dataframe we have to run df.drop(['Unnamed: 0'], axis=1),to drop the extra column.

Dataset Description

In this Program the data set used is *'**advertising.csv**'* It shows the money spent on **TV**, **Radio** and **Newspaper** Ads and the ***Sales*** Income generated. The Dataset is 200 rows and 4 columns. (TV, Radio,Newspaper and Sales).

In [22]:

```
# Importing dataset
import pandas as pd
df=pd.read_csv('advertising.csv')
print(df.head())
print("Dataframe shape is = ",df.shape)
```

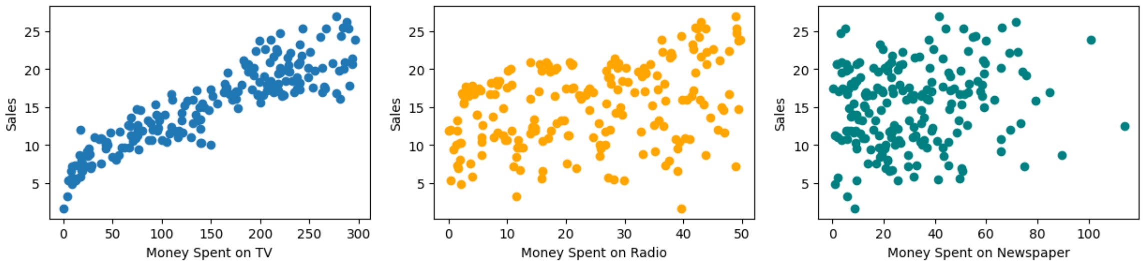
```
      TV  Radio  Newspaper  Sales
0  230.1   37.8       69.2   22.1
1   44.5   39.3       45.1   10.4
2   17.2   45.9       69.3   12.0
3  151.5   41.3       58.5   16.5
4  180.8   10.8       58.4   17.9
Dataframe shape is =  (200, 4)
```

Plotting Advertising Media vs Sales

In [23]:

```
# Plotting advetising media vs sales for each
import matplotlib.pyplot as plt
graphSheet = plt.figure(figsize=(15,10))
graphSheet.add_subplot(3,3,1)
plt.scatter(df['TV'],df['Sales'])
plt.xlabel("Money Spent on TV")
plt.ylabel("Sales")
graphSheet.add_subplot(3,3,2)
plt.scatter(df['Radio'],df['Sales'],c='orange')
plt.xlabel("Money Spent on Radio")
plt.ylabel("Sales")
graphSheet.add_subplot(3,3,3)
plt.scatter(df['Newspaper'],df['Sales'],c='teal')
plt.xlabel("Money Spent on Newspaper")
plt.ylabel("Sales")
```

```
Out[23]: Text(0, 0.5, 'Sales')
```



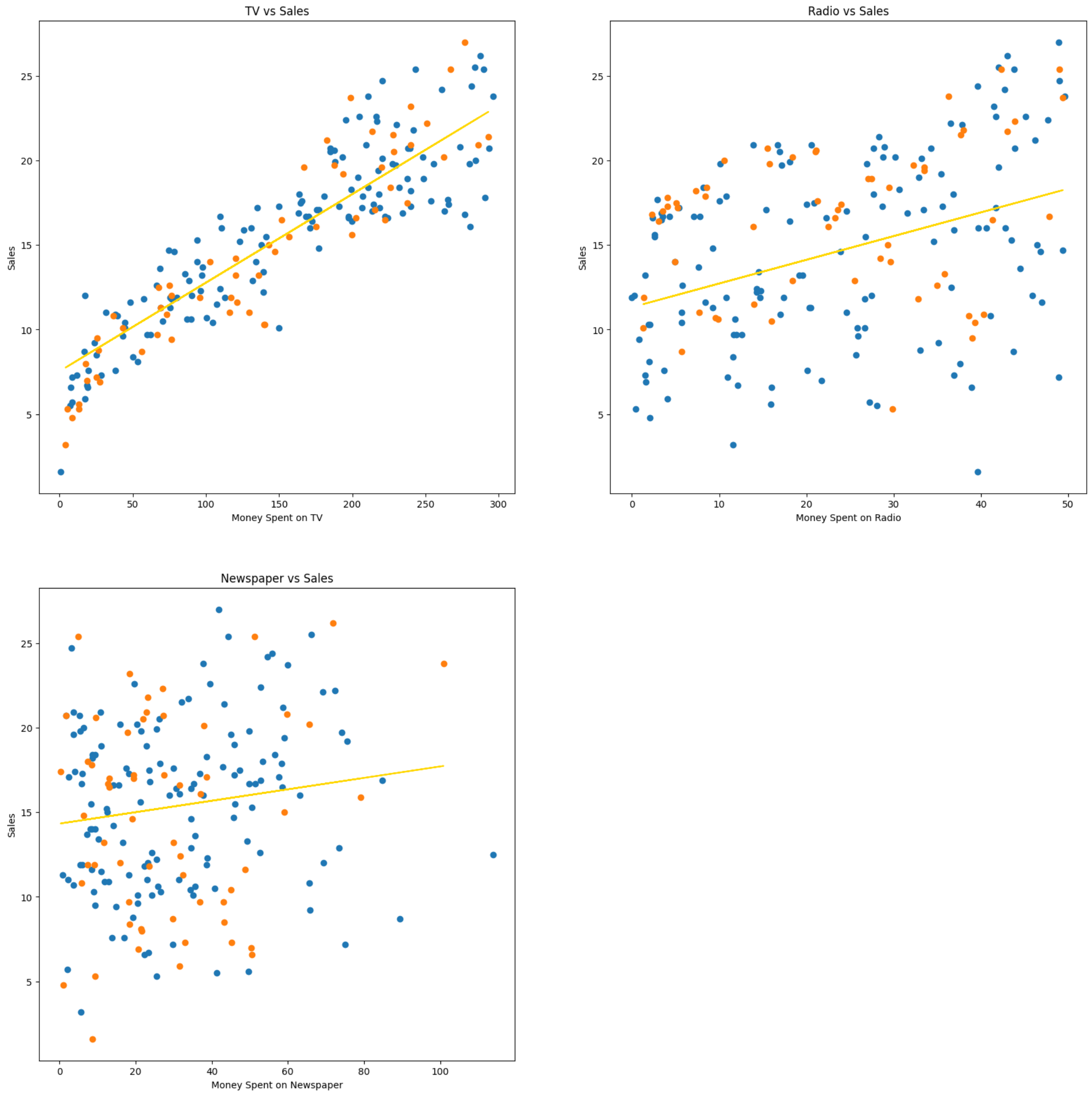
```
In [24]: #Importing and fitting model
         from sklearn.linear_model import LinearRegression
         from sklearn.model_selection import train_test_split
         from sklearn.metrics import mean_squared_error,mean_absolute_error
         import numpy as np
```

```
In [25]: # Defining a function to do the work
         def linReg(x,y):
             print((x+" vs " + y).center(40,'='))
             # train_test_split with 70-30 ratio
             x_train,x_test,y_train,y_test=train_test_split(df[x],df[y],test_size=0.3)
             x_train = x_train.to_numpy().reshape(-1,1)
             x_test = x_test.to_numpy().reshape(-1,1)
             y_train = y_train.to_numpy().reshape(-1,1)
             y_test = y_test.to_numpy().reshape(-1,1)
             #Initializing and fitting LinearRegression model
             lr = LinearRegression()
             lr.fit(x_train,y_train)
             #Printing Coeff and Intercept values
             print("Coeff=",lr.coef_[0][0],"\nIntercept=",lr.intercept_[0])
             pred = lr.predict(x_test)
             #Prining line of regression
             print("The linear model of {} versus {} is: Y = {:.3} + {:.2}X".format(x,y,l
             #Finding Root Mean Sqaured Error
             rmse=np.sqrt(mean_squared_error(y_test,pred))
             #Finiding Mean Absolute Error
             mae = mean_absolute_error(y_test,pred)
             #Finding Mean Squared Error
             mse = mean_squared_error(y_test,pred)
             #Printing all error matrics
             print("Root Mean Square Srror= {}\nMean Absolute Error = {}\nMean Squared Er
             #Plotting the final Graph
             plt.scatter(x_train,y_train)
             plt.scatter(x_test,y_test)
             plt.xlabel("Money Spent on "+x)
             plt.ylabel(y)
             plt.title(x+" vs "+y)
             plt.plot(x_test,pred,c='gold')
```

```
In [26]: #Creating a shhet where we`ll print all graphs
         #Calling the function with column names only
         sheet = plt.figure(figsize=(20,20))
         sheet.add_subplot(2,2,1)
         linReg('TV','Sales')
         sheet.add_subplot(2,2,2)
         linReg('Radio','Sales')
```

```
sheet.add_subplot(2,2,3)
linReg('Newspaper','Sales')
```

```
==============TV vs Sales===============
Coeff= 0.05233041215402095
Intercept= 7.540181071215254
The linear model of TV versus Sales is: Y = 7.54 + 0.052X
Root Mean Square Srror= 2.3039551948568016
Mean Absolute Error = 1.8745358406118053
Mean Squared Error = 5.308209539907643
=============Radio vs Sales============
Coeff= 0.14013949825892588
Intercept= 11.321246203110555
The linear model of Radio versus Sales is: Y = 11.3 + 0.14X
Root Mean Square Srror= 4.70270176743546
Mean Absolute Error = 4.156484025205724
Mean Squared Error = 22.115403913440602
===========Newspaper vs Sales==========
Coeff= 0.033793279855684394
Intercept= 14.327389023061361
The linear model of Newspaper versus Sales is: Y = 14.3 + 0.034X
Root Mean Square Srror= 5.891909594049261
Mean Absolute Error = 5.07299028558451
Mean Squared Error = 34.71459866444973
```



Inference

- Newspaper RMSE = 5.514231114677423
- Radio RMSE = 4.773872035217551
- TV RMSE = 2.396676218281216

The RMSE Value for TV is the least.
This means that the money spent on TV. Ads has the highest possiblity of a reliable sales income prediction.