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 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 [1]:

```
# 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)
```

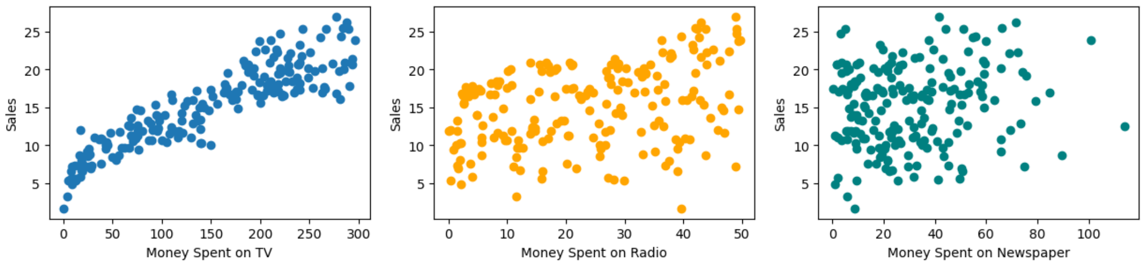
Plotting Advertising Media vs Sales

In [2]:

```
# 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[2]:

```
Text(0, 0.5, 'Sales')
```



In [3]:

```
#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,r2_score
import numpy as np
```

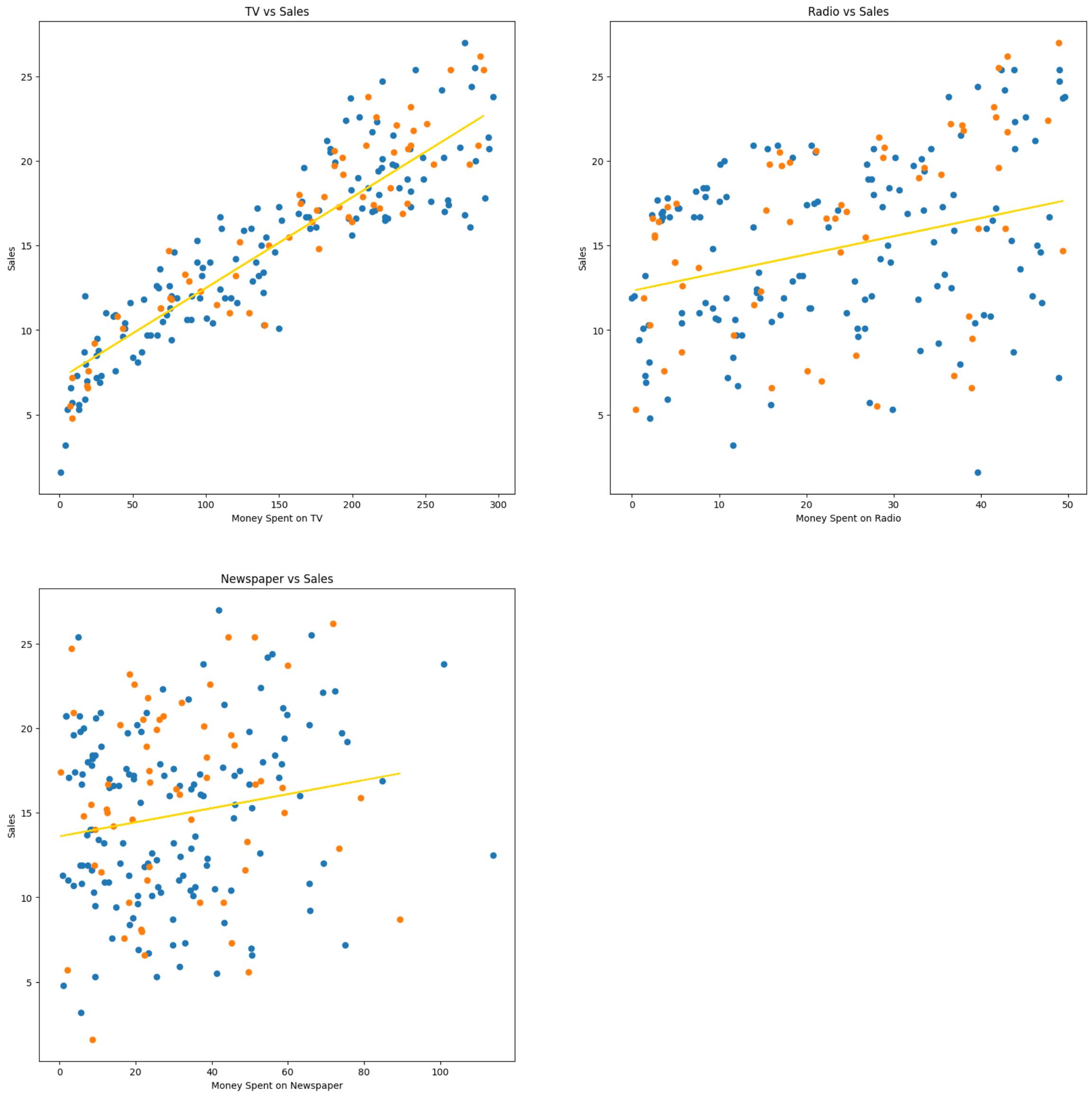
In [4]:

```
# 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 RMS error
    rmse=np.sqrt(mean_squared_error(y_test,pred))
    print("Root mean sqr err=",rmse)
    #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 [5]:

```
#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= 0.05364227554716221
Intercept= 7.121968091653801
The linear model of TV versus Sales is: Y = 7.12 + 0.054X
Root mean sqr err= 2.0756567785234425
=============Radio vs Sales=============
Coeff= 0.10747586783041801
Intercept= 12.320758763149406
The linear model of Radio versus Sales is: Y = 12.3 + 0.11X
Root mean sqr err= 5.144649759730167
===========Newspaper vs Sales===========
Coeff= 0.04155357962411486
Intercept= 13.606195606179064
The linear model of Newspaper versus Sales is: Y = 13.6 + 0.042X
Root mean sqr err= 5.642399667463073
```



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.