Name : SINU S MARIAM

Designation : Data Science Intern

Organization : OASIS INFOBYTE

#Task 5 - SALES PREDICTION USING PYTHON

Problem Statement:

- As a Data Scientist in a product / Service based company , Try to predict the future sales of the product considering the budget the company spent on different Advertisement Tools
- Use Machine Learning Techniques for Sales Prediction using Python Programming

In [29]:
```
#importing necessary libraries
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
#importing libraries for visualisation
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns
```

In [2]:
```
#importing Data
data_frame = pd.read_csv('C:/Users/sinun/OneDrive/Documents/oasis infobyte/Advertisi
```

Performing descriptive analysis. Understand the variables and their corresponding values.

In [3]:
```
# Understanding the dimensions of data
data_frame.shape
```

Out[3]:
```
(200, 4)
```

In [4]:
```
# Understanding the Data Variables
data_frame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 200 entries, 1 to 200
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 7.8 KB
```

In [5]:
```
data_frame.columns
```

Out[5]:
```
Index(['TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')
```

* The company spent their budget for differnt products on 3 advertising medias such as TV, Radio, Newspaper and the corresponding sales for each product

In [6]:
```
# Show the top 5 Rows of data
data_frame.head()
```

Out[6]:

	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

In [7]:
```
# Performing Descriptive Analysis
data_frame.describe().T
```

Out[7]:

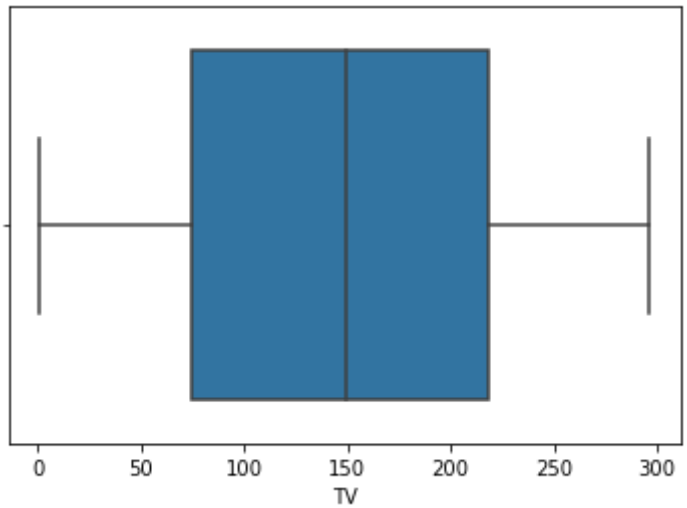
	count	mean	std	min	25%	50%	75%	max
TV	200.0	147.0425	85.854236	0.7	74.375	149.75	218.825	296.4
Radio	200.0	23.2640	14.846809	0.0	9.975	22.90	36.525	49.6
Newspaper	200.0	30.5540	21.778621	0.3	12.750	25.75	45.100	114.0
Sales	200.0	14.0225	5.217457	1.6	10.375	12.90	17.400	27.0

In [8]:
```
# Check for Duplicated Entries
data_frame.duplicated().sum()
```

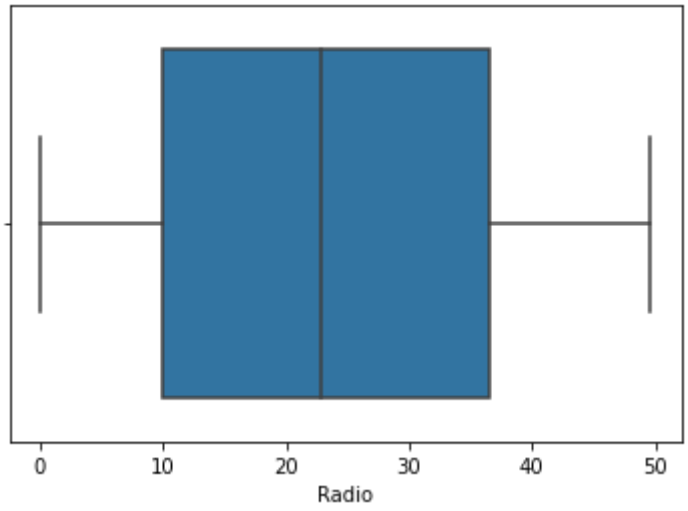
Out[8]: 0

Outlier Analysis

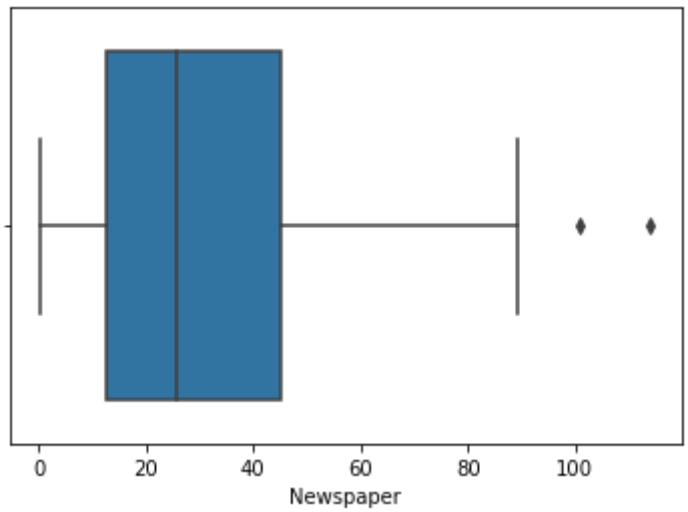
In [9]:
```
fig,axs=plt.subplots(1,1)
plt1=sns.boxplot(data_frame['TV'],ax=axs)
```



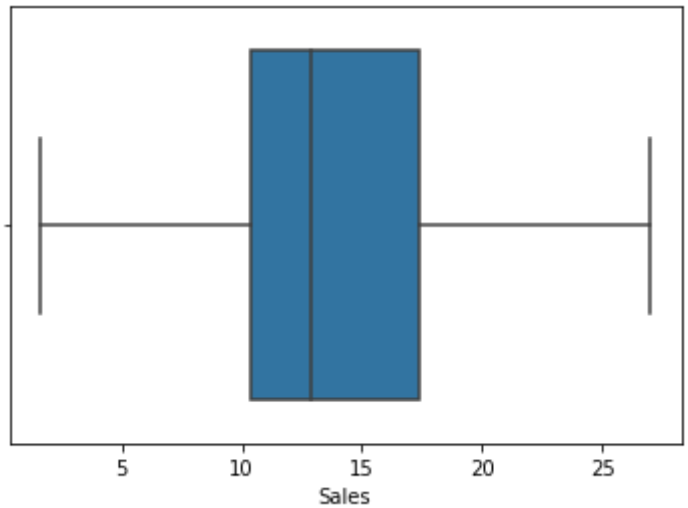
In [10]:
```
fig,axs=plt.subplots(1,1)
plt1=sns.boxplot(data_frame['Radio'],ax=axs)
```



In [11]:
```
fig,axs=plt.subplots(1,1)
plt1=sns.boxplot(data_frame['Newspaper'],ax=axs)
```



In [12]:
```
fig,axs=plt.subplots(1,1)
plt1=sns.boxplot(data_frame['Sales'],ax=axs)
```

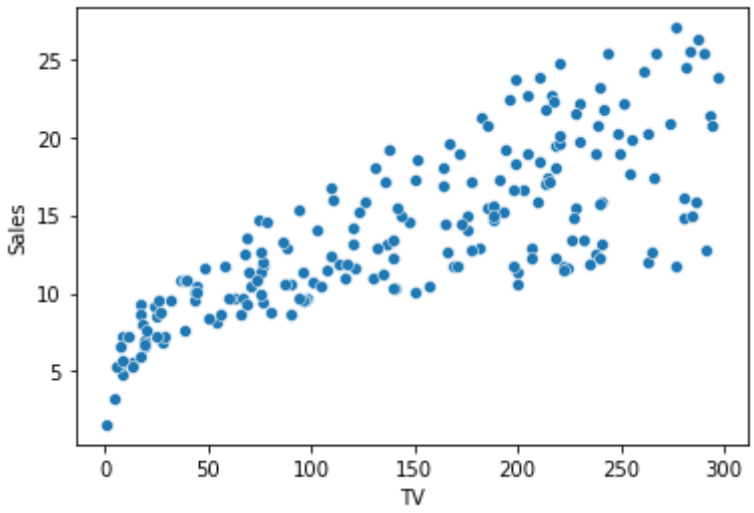


Data Visualization

* Data Visualization helps to show how the budget spent on each advertising media affect the sales of products

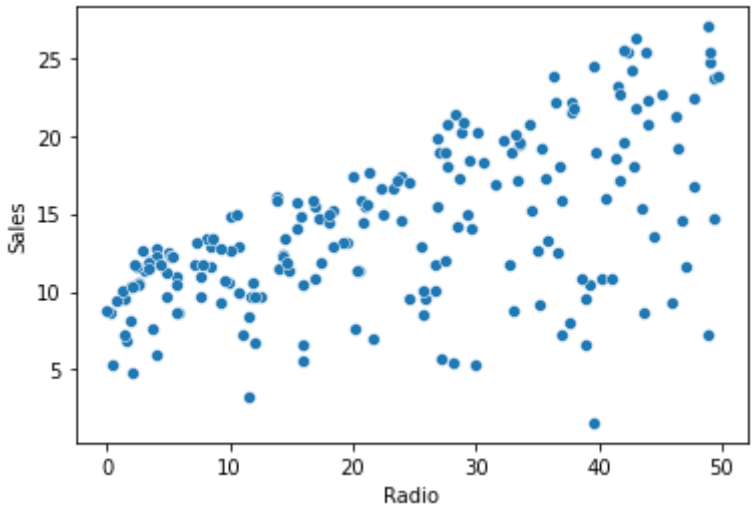
In [13]:
```
#Scatter plot is used find the distribution of effects of each advertising media aga
plt.figure(figsize=(6,4))
sns.scatterplot(data=data_frame,x=data_frame['TV'],y=data_frame['Sales'])
```

Out[13]: <AxesSubplot:xlabel='TV', ylabel='Sales'>



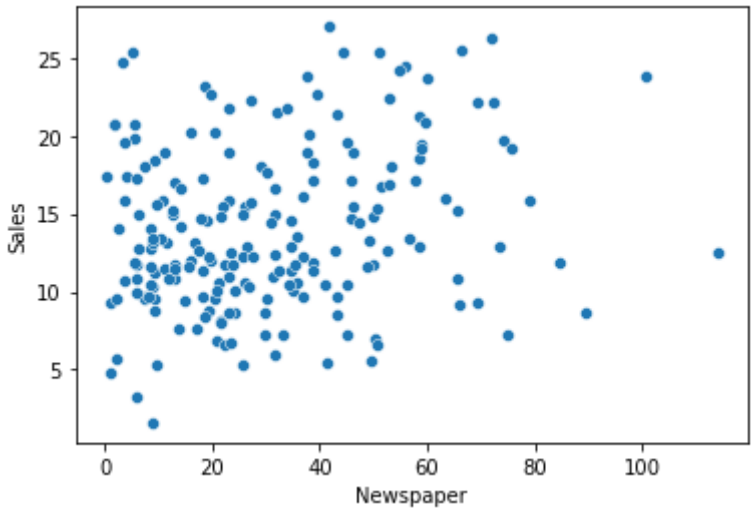
In [14]:
```
plt.figure(figsize=(6,4))
sns.scatterplot(data=data_frame,x=data_frame['Radio'],y=data_frame['Sales'])
```

Out[14]: <AxesSubplot:xlabel='Radio', ylabel='Sales'>



In [15]:
```
plt.figure(figsize=(6,4))
sns.scatterplot(data=data_frame,x=data_frame['Newspaper'],y=data_frame['Sales'])
```

Out[15]: <AxesSubplot:xlabel='Newspaper', ylabel='Sales'>

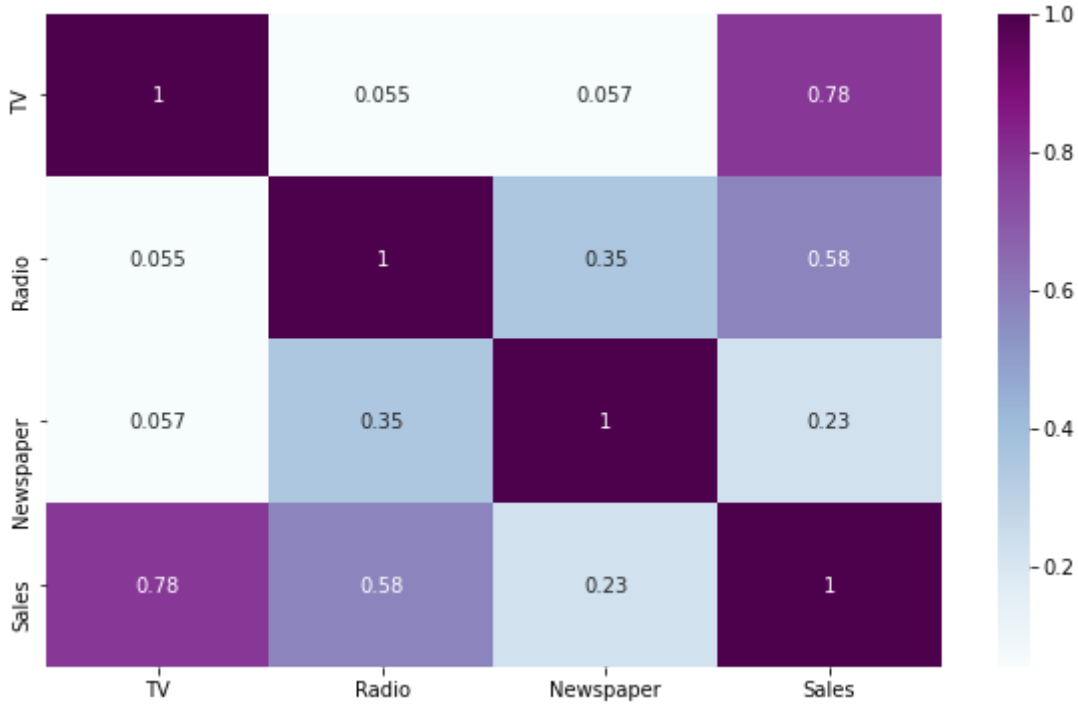


* It is seen that TV data set is more linear as compared to other 2 variables .

Heat Map

In [16]:
```
# find correlation between variables in data set for plotting heatmap
df_corr=data_frame.corr()
```

In [17]:
```
plt.figure(figsize=(10,6))
sns.heatmap(df_corr,annot=True,cmap="BuPu")
plt.show()
```



* We can see that TV variable has highest correlation value with the target Sales variable

Building the Forecasting Model

In [18]:
```
from sklearn.model_selection import train_test_split
```

In [19]:
```
#First step in building the forecasting model is to identify the Feature(Input) vari
features = data_frame[['TV', 'Radio', 'Newspaper']]
target = data_frame[['Sales']]
```

* Splitting data for training and testing the model

In [20]:
```
# Splitting data for training the model and testing the model
# train size taken as 0.8
X_train, X_test, y_train, y_test = train_test_split(features, target, train_size = .
# Dimensions of Train and Test Data sets
print('Train set of features: ', X_train.shape)
print('Test set of features: ', X_test.shape)
print('Target for train: ', y_train.shape)
print('Target for test: ', y_test.shape)
```

```
Train set of features:  (160, 3)
Test set of features:  (40, 3)
Target for train:  (160, 1)
Target for test:  (40, 1)
```

Learn the model on train data

In [21]:
```
from sklearn.linear_model import LinearRegression
```

In [22]:
```
# Linear Regression Model ( a Supervised Machine learning Algorithm)
# LR models impose a linear function between predictor and response variables
my_model = LinearRegression()
```

In [23]:
```
# Fitting the model in train data set ie the Linear Regression Model learned from th
my_model.fit(X_train, y_train)
```

Out[23]:
```
LinearRegression()
```

Predicting the Sales

In [24]:
```
# Predicting the sales from Feature Test values
y_pred = my_model.predict(X_test)
y_pred
```

Out[24]:
```
array([[11.58310885],
       [17.93587708],
       [21.01084252],
       [14.83055643],
       [ 4.44748623],
       [22.66459738],
       [13.32231326],
       [20.90987386],
       [ 9.61644771],
       [15.60268276],
       [13.99517498],
       [23.92820759],
       [ 8.06853038],
       [ 3.63975544],
       [15.10225533],
       [ 6.52337431],
       [22.9798927 ],
       [12.98092833],
       [19.07905645],
       [17.95713955],
       [ 9.93706319],
       [ 8.3044354 ],
       [11.91575511],
```

```
[24.51383377],
[ 4.39627089],
[17.0716927 ],
[ 9.66437931],
[17.09224071],
[13.91558757],
[21.05423316],
[18.19180844],
[ 9.02162474],
[12.16559185],
[ 8.79231074],
[15.34609319],
[21.93046173],
[ 7.42286054],
[17.60122685],
[17.11894979],
[20.07699477]])
```

Test the model

In [25]:
```
from sklearn.metrics import mean_squared_error
```

Mean Squared Error

In [26]:
```
# Compare the predicted values with the true values
mean_squared_error(y_pred, y_test)
```

Out[26]: 2.4729855051485083

Coefficient of Determination or R Squared Value (r2)

In [27]:
```
from sklearn.metrics import r2_score
```

In [28]:
```
# find Coefficient of Determination or R Squared Value (r2)
r2_score(y_test,y_pred)
```

Out[28]: 0.9290521316359344