SALES PREDICTION USING PYTHON

Sales prediction means predicting how much of a product people will buy based on factors such as the amount you spend to advertise your product, the segment of people you advertise for, or the platform you are advertising on about your product.

In [10]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from statsmodels.formula.api import ols
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```
sales = pd.read_csv('C:/Users/Gayatri/Downloads/Advertising.csv',index_col=0)

sales.head()
```

Out[2]:

	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 [3]:
```
# Exploratory Data Analysis
sales.shape
```

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

In [4]:
```
sales.info
```

Out[4]:
```
<bound method DataFrame.info of         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
..     ...    ...        ...    ...
196   38.2    3.7       13.8    7.6
197   94.2    4.9        8.1    9.7
198  177.0    9.3        6.4   12.8
199  283.6   42.0       66.2   25.5
200  232.1    8.6        8.7   13.4

[200 rows x 4 columns]>
```

In [5]:
```
sales.describe
```

Out[5]:
```
<bound method NDFrame.describe of         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
..      ...     ...         ...     ...
196    38.2     3.7        13.8     7.6
197    94.2     4.9         8.1     9.7
198   177.0     9.3         6.4    12.8
199   283.6    42.0        66.2    25.5
200   232.1     8.6         8.7    13.4

[200 rows x 4 columns]>
```

In [6]:
```
sales.isnull().sum()
```

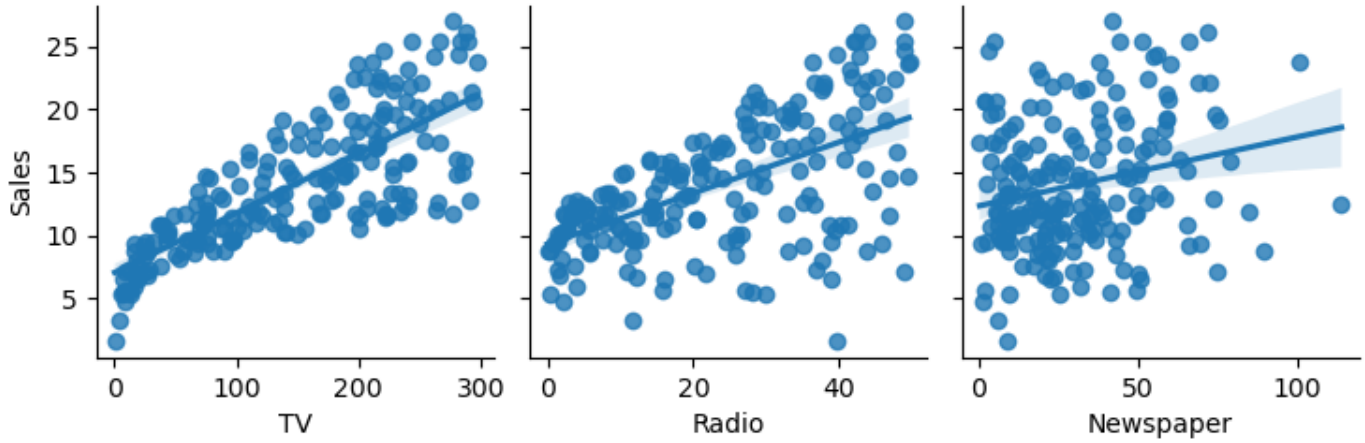
Out[6]:
```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

In [7]:
```
# Check the main assupmtions of linear Regression linearity , normality,mutlicolinearity

#1- linearity: which mean the relation between predictor(x) and outcome(y) must be linea

sns.pairplot(sales,x_vars=['TV','Radio','Newspaper'],y_vars='Sales',kind='reg')
```

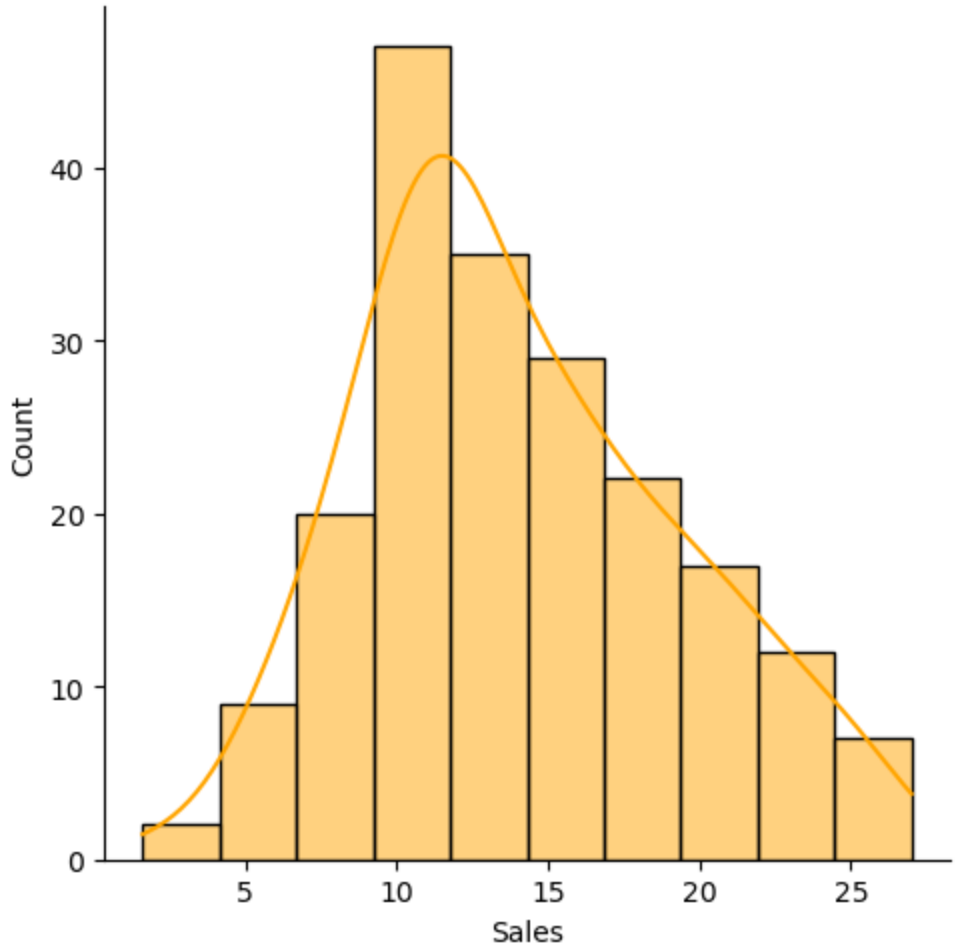
Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x2b903ddc790>
```



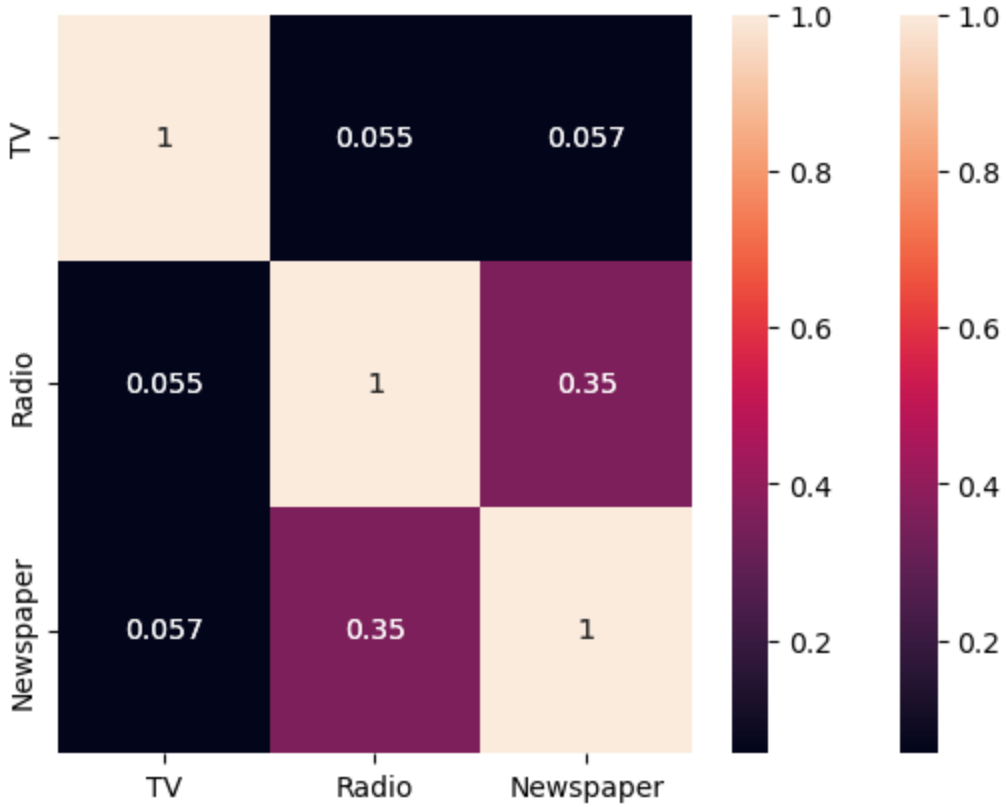
as we can see the relation between newspaper is not linear this will affect our model so may be we will remove this feature

2-Normality: we need the outcome variable(y) to be normally distributed

In [12]:
```
sns.displot(sales.Sales,bins=10,color='orange',kde=True)
plt.show()
```



In [14]:
```
sns.heatmap(sales.drop('Sales',axis=1).corr(),annot=True)
plt.show()
```



In [15]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
r = sales[["TV", "Radio","Newspaper"]].values
vif_df = pd.DataFrame()
vif_df["VIF"] = [variance_inflation_factor(r, i) for i in range(3)]
```

```
vif_df["feature"] = ["TV", "Radio","Newspaper"]
vif_df
```

Out[15]:

	VIF	feature
0	2.486772	TV
1	3.285462	Radio
2	3.055245	Newspaper

In [16]:
```
X = sales.drop(['Sales','Newspaper'], axis=1)
y = sales[["Sales"]]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=4


models = [('LinearRegression', LinearRegression())]
```

In [17]:
```
lin_model = ols(formula="Sales ~ TV + Radio ",data=sales).fit()

print(lin_model.params,"\n")
print(lin_model.summary())
```

```
Intercept    2.921100
TV           0.045755
Radio        0.187994
dtype: float64

                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.897
Model:                            OLS   Adj. R-squared:                  0.896
Method:                 Least Squares   F-statistic:                     859.6
Date:                Mon, 26 Jun 2023   Prob (F-statistic):           4.83e-98
Time:                        14:37:56   Log-Likelihood:                -386.20
No. Observations:                 200   AIC:                             778.4
Df Residuals:                     197   BIC:                             788.3
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.9211      0.294      9.919      0.000       2.340       3.502
TV             0.0458      0.001     32.909      0.000       0.043       0.048
Radio          0.1880      0.008     23.382      0.000       0.172       0.204
==============================================================================
Omnibus:                       60.022   Durbin-Watson:                   2.081
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              148.679
Skew:                          -1.323   Prob(JB):                     5.19e-33
Kurtosis:                       6.292   Cond. No.                         425.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
ed.
```

In [18]:
```
# this model with Newspaper feature added
X = sales.drop(['Sales'], axis=1)
y = sales[["Sales"]]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=4


models = [('LinearRegression', LinearRegression())]
```

In [19]:
```
lin_model = ols(formula="Sales ~ TV + Radio + Newspaper ",data=sales).fit()

print(lin_model.params,"\n")
print(lin_model.summary())
```

```
Intercept    2.938889
TV           0.045765
Radio        0.188530
Newspaper   -0.001037
dtype: float64

                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.897
Model:                            OLS   Adj. R-squared:                  0.896
Method:                 Least Squares   F-statistic:                     570.3
Date:                Mon, 26 Jun 2023   Prob (F-statistic):           1.58e-96
Time:                        14:39:01   Log-Likelihood:                -386.18
No. Observations:                 200   AIC:                             780.4
Df Residuals:                     196   BIC:                             793.6
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.9389      0.312      9.422      0.000       2.324       3.554
TV             0.0458      0.001     32.809      0.000       0.043       0.049
Radio          0.1885      0.009     21.893      0.000       0.172       0.206
Newspaper     -0.0010      0.006     -0.177      0.860      -0.013       0.011
==============================================================================
Omnibus:                       60.414   Durbin-Watson:                   2.084
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              151.241
Skew:                          -1.327   Prob(JB):                     1.44e-33
Kurtosis:                       6.332   Cond. No.                         454.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
ed.
```

Adding the news paper or remove it not affect the model too much

In [ ]:
```
```