

```
In [11]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: data=pd.read_csv('Advertising.csv')
```

```
In [4]: data.head()
```

```
Out[4]:
```

	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

```
In [5]: data.shape
```

```
Out[5]: (200, 5)
```

```
In [6]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Unnamed: 0      200 non-null   int64
1   TV              200 non-null   float64
2   Radio           200 non-null   float64
3   Newspaper       200 non-null   float64
4   Sales           200 non-null   float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

```
In [7]: data=data.iloc[:,1:]
```

```
In [8]: data.tail()
```

```
Out[8]:
```

	TV	Radio	Newspaper	Sales
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

```
In [9]: data.isna().sum()
```

```
Out[9]: TV          0  
Radio         0  
Newspaper     0  
Sales         0  
dtype: int64
```

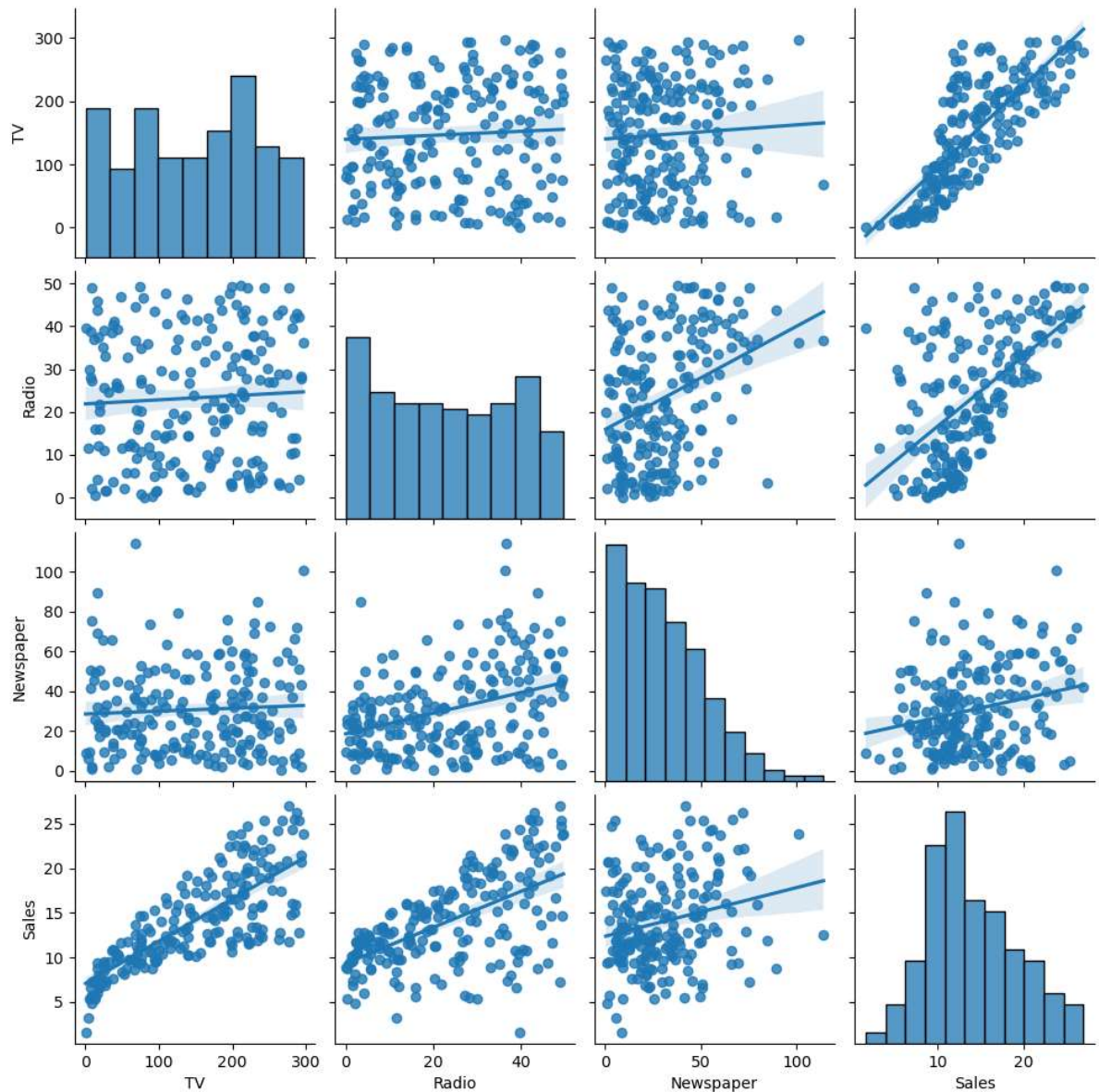
```
In [10]: data.describe()
```

```
Out[10]:
```

	TV	Radio	Newspaper	Sales
<b>count</b>	200.000000	200.000000	200.000000	200.000000
<b>mean</b>	147.042500	23.264000	30.554000	14.022500
<b>std</b>	85.854236	14.846809	21.778621	5.217457
<b>min</b>	0.700000	0.000000	0.300000	1.600000
<b>25%</b>	74.375000	9.975000	12.750000	10.375000
<b>50%</b>	149.750000	22.900000	25.750000	12.900000
<b>75%</b>	218.825000	36.525000	45.100000	17.400000
<b>max</b>	296.400000	49.600000	114.000000	27.000000

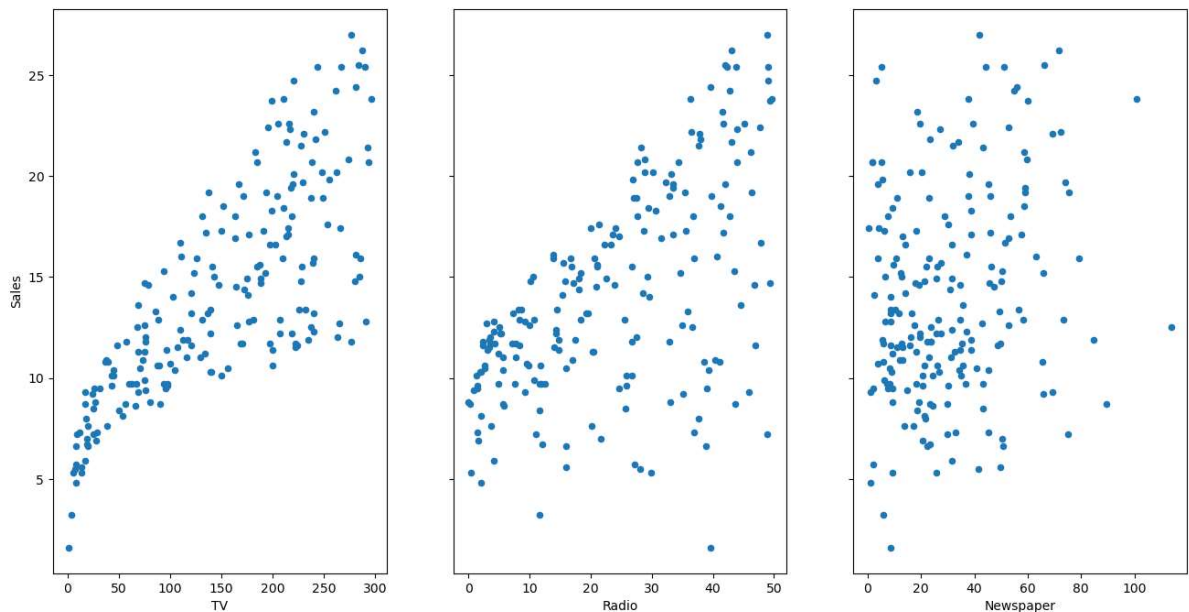
```
In [12]: sns.pairplot(data,kind="reg");
```

C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight  
self.\_figure.tight\_layout(\*args, \*\*kwargs)



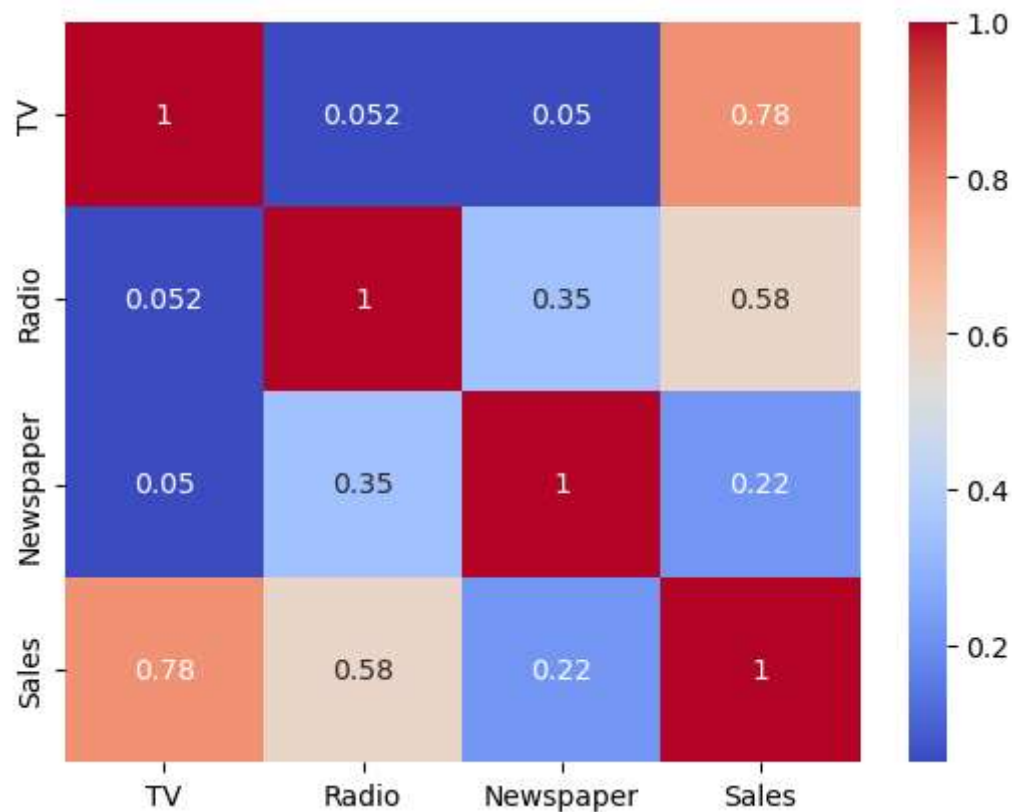
```
In [13]: fig,axs= plt.subplots(1,3,sharey=True)
data.plot(kind="scatter",x='TV',y='Sales',ax=axs[0],figsize=(16,8))
data.plot(kind="scatter",x='Radio',y='Sales',ax=axs[1],figsize=(16,8))
data.plot(kind="scatter",x='Newspaper',y='Sales',ax=axs[2],figsize=(16,8))
```

Out[13]: <Axes: xlabel='Newspaper', ylabel='Sales'>



```
In [29]: sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
```

Out[29]: <Axes: >



```
In [34]: sns.set(rc={"figure.figsize":(6,4)})  
sns.distplot(data["TV"], kde=True, color="orange", bins=10)
```

C:\Users\tejas\AppData\Local\Temp\ipykernel\_10940\2205456444.py:2: UserWarning:

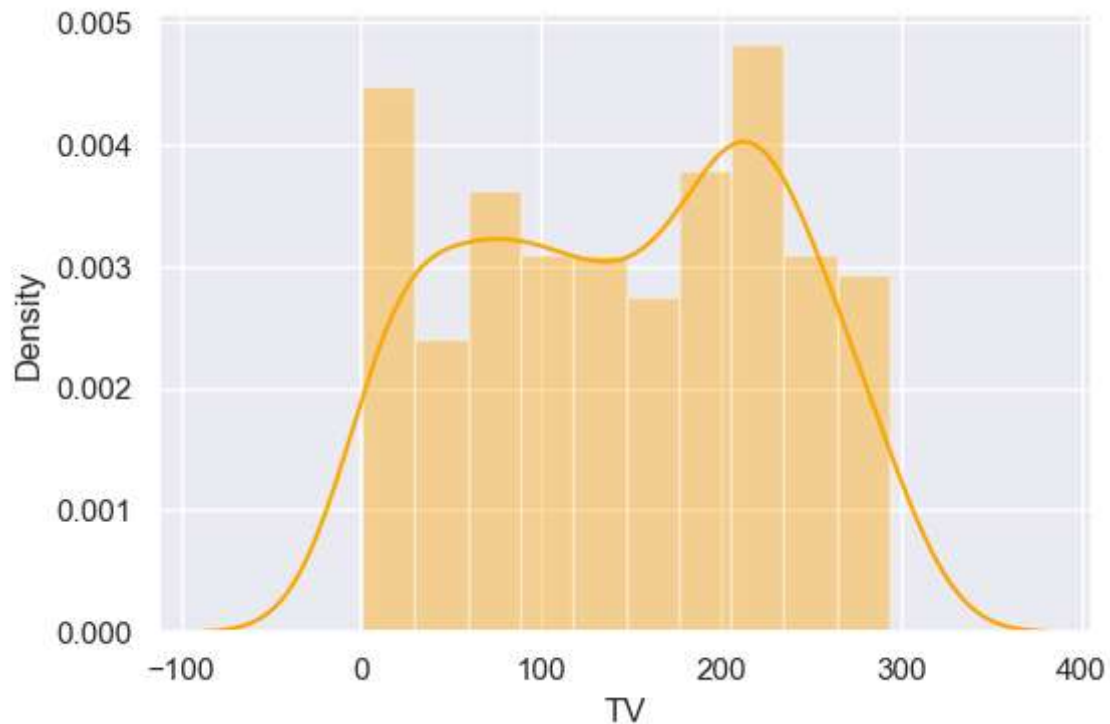
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

```
sns.distplot(data["TV"], kde=True, color="orange", bins=10)
```

Out[34]: <Axes: xlabel='TV', ylabel='Density'>



```
In [36]: sns.set(rc={"figure.figsize":(6,4)})
sns.distplot(data["Radio"], kde=True, color="orange", bins=10)
```

C:\Users\tejas\AppData\Local\Temp\ipykernel\_10940\555418184.py:2: UserWarning:

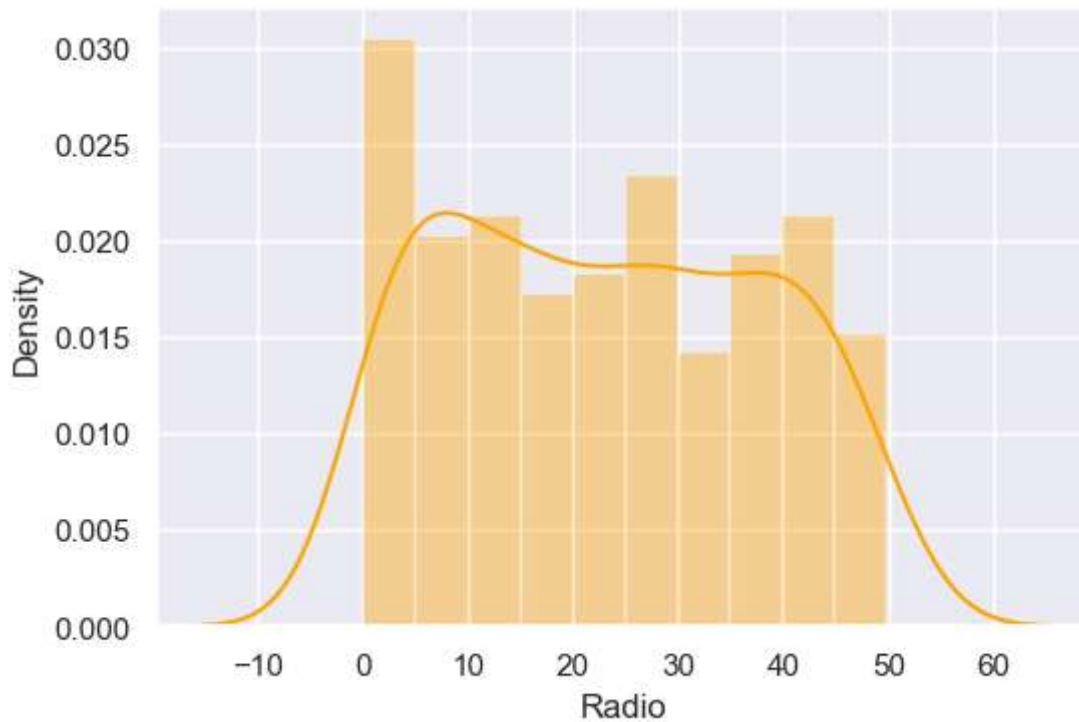
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

```
sns.distplot(data["Radio"], kde=True, color="orange", bins=10)
```

Out[36]: <Axes: xlabel='Radio', ylabel='Density'>



```
In [37]: sns.set(rc={"figure.figsize":(6,4)})  
sns.distplot(data["Newspaper"], kde=True, color="orange", bins=10)
```

C:\Users\tejas\AppData\Local\Temp\ipykernel\_10940\2773182218.py:2: UserWarning:

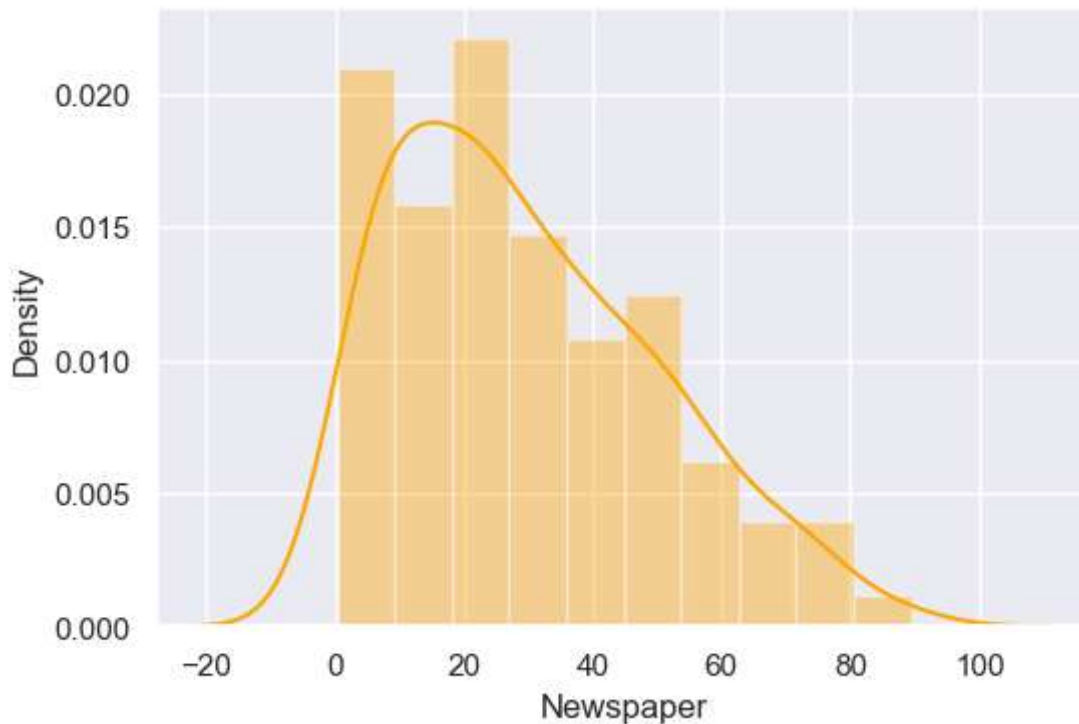
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

```
sns.distplot(data["Newspaper"], kde=True, color="orange", bins=10)
```

Out[37]: <Axes: xlabel='Newspaper', ylabel='Density'>



In [38]:

```
sns.set(rc={"figure.figsize":(6,4)})  
sns.distplot(data["Sales"], kde=True, color="orange", bins=10)
```

C:\Users\tejas\AppData\Local\Temp\ipykernel\_10940\631265641.py:2: UserWarning:

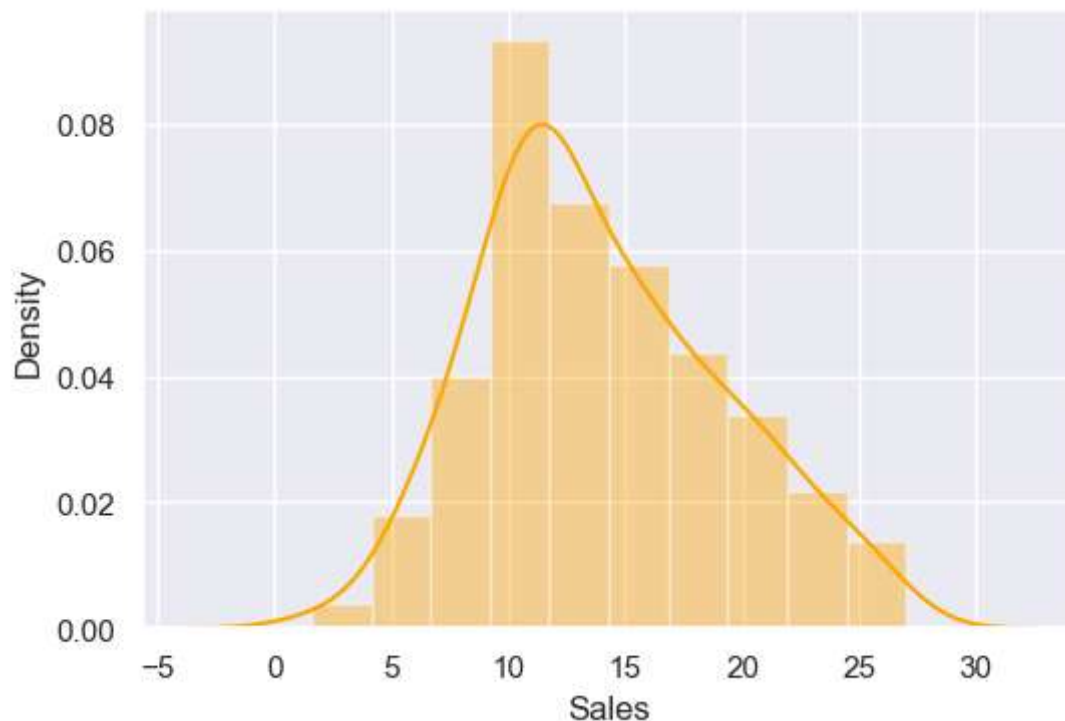
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

```
sns.distplot(data["Sales"], kde=True, color="orange", bins=10)
```

Out[38]: <Axes: xlabel='Sales', ylabel='Density'>



In [14]: *#moving the outlier from newspaper*

```
data=data[data['Newspaper']<=90]  
data.shape
```

Out[14]: (198, 4)



```
In [15]: data.corr()
```

```
Out[15]:
```

	TV	Radio	Newspaper	Sales
TV	1.000000	0.051978	0.049771	0.779121
Radio	0.051978	1.000000	0.346364	0.576748
Newspaper	0.049771	0.346364	1.000000	0.219555
Sales	0.779121	0.576748	0.219555	1.000000

```
In [16]: # Separating input and output data
```

```
x=data.drop(columns=['Sales'])
y=data['Sales']
```

```
In [17]: x.head()
```

```
Out[17]:
```

	TV	Radio	Newspaper
0	230.1	37.8	69.2
1	44.5	39.3	45.1
2	17.2	45.9	69.3
3	151.5	41.3	58.5
4	180.8	10.8	58.4

```
In [18]: y.head()
```

```
Out[18]: 0    22.1
1    10.4
2     9.3
3    18.5
4    12.9
Name: Sales, dtype: float64
```

```
In [19]: from sklearn.model_selection import train_test_split
```

```
In [20]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

```
In [21]: x_train.tail()
```

```
Out[21]:
```

	TV	Radio	Newspaper
47	239.9	41.5	18.5
26	142.9	29.3	12.6
14	204.1	32.9	46.0
32	97.2	1.5	30.0
111	241.7	38.0	23.2

```
In [22]: y_test.tail()
```

```
Out[22]: 74      17.0
        190     10.8
        119      6.6
        57     13.2
         5      7.2
        Name: Sales, dtype: float64
```

```
In [23]: from sklearn.preprocessing import OneHotEncoder, StandardScaler, OrdinalEncoder
        from sklearn.compose import make_column_transformer
        from sklearn.pipeline import make_pipeline
        from sklearn.metrics import r2_score
```

```
In [24]: column_trans=make_column_transformer((OneHotEncoder(sparse=False),[]),remainder=
        scaler=StandardScaler()
        oe=OrdinalEncoder())
```

```
In [25]: #Random Forest Regression Model
```

```
        from sklearn.ensemble import RandomForestRegressor
        r=RandomForestRegressor(n_estimators=10, random_state=0)
        pipe=make_pipeline(column_trans, scaler, r)
        pipe.fit(x_train, y_train)
        y_pred_r=pipe.predict(x_test)
        r2_score(y_test, y_pred_r)
```

```
Out[25]: 0.9852415053001466
```

```
In [26]: pipe.predict([[283.6, 42.0, 66.2]])
```

```
Out[26]: array([24.98])
```

```
In [27]: pipe.predict([[112.9, 17.4, 38.6]])
```

```
Out[27]: array([12.31])
```

```
In [ ]:
```