In [61]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [62]:
```
df=pd.read_csv('https://raw.githubusercontent.com/dsrscientist/DSData/master/Advertising.csv')
```

In [63]:
```
df.head()
```

Out[63]:

	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 [64]:
```
df.shape
```

Out[64]:
```
(200, 5)
```

In [65]:
```
df.drop(['Unnamed: 0'],axis=1,inplace=True)
```

In [66]:
```
df
```

Out[66]:

	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	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...	...	...	...	...
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

200 rows × 4 columns

In [67]:
```
df.shape
```

Out[67]:
```
(200, 4)
```

In [68]:
```
df.describe()
```

Out[68]:

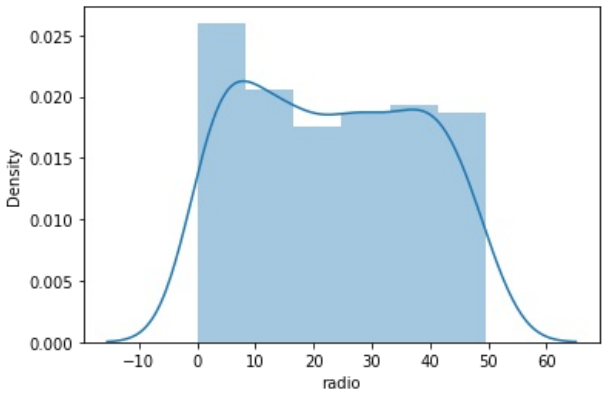
	TV	radio	newspaper	sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	14.022500
std	85.854236	14.846809	21.778621	5.217457
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	10.375000
50%	149.750000	22.900000	25.750000	12.900000

75%	218.825000	36.525000	45.100000	17.400000
max	296.400000	49.600000	114.000000	27.000000

In [69]:
```
sns.distplot(df['radio'])
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

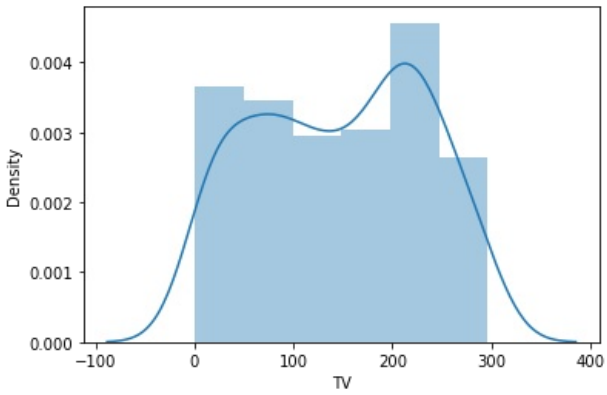
Out[69]: <AxesSubplot:xlabel='radio', ylabel='Density'>



In [70]:
```
sns.distplot(df['TV'])
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

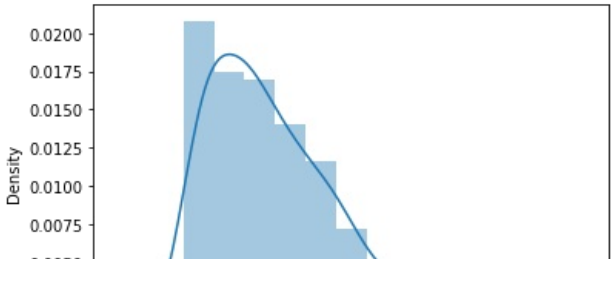
Out[70]: <AxesSubplot:xlabel='TV', ylabel='Density'>

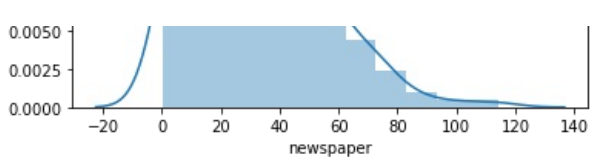


In [71]:
```
sns.distplot(df['newspaper'])
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a fig
ure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[71]: <AxesSubplot:xlabel='newspaper', ylabel='Density'>





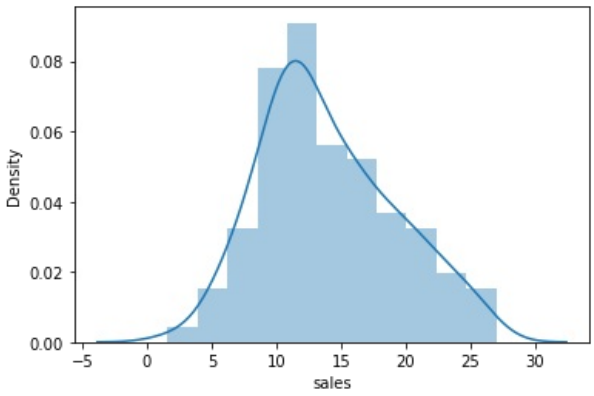
In [72]:
```
df.skew()
```

Out[72]:
```
TV          -0.069853
radio        0.094175
newspaper    0.894720
sales        0.407571
dtype: float64
```

In [73]:
```
sns.distplot(df['sales'])
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[73]:
```
<AxesSubplot:xlabel='sales', ylabel='Density'>
```

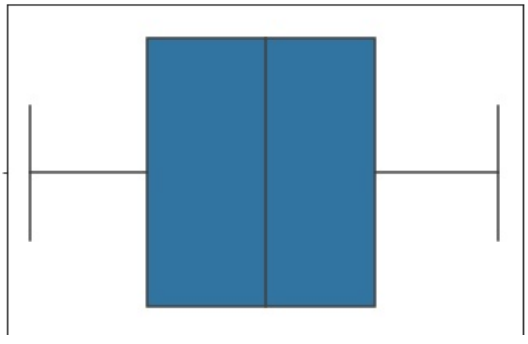


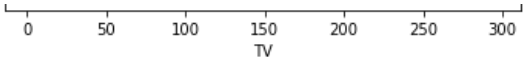
In [74]:
```
df.isnull().sum()
```

Out[74]:
```
TV           0
radio        0
newspaper    0
sales        0
dtype: int64
```

In [75]:
```
sns.boxplot(df['TV'])
plt.show()
```

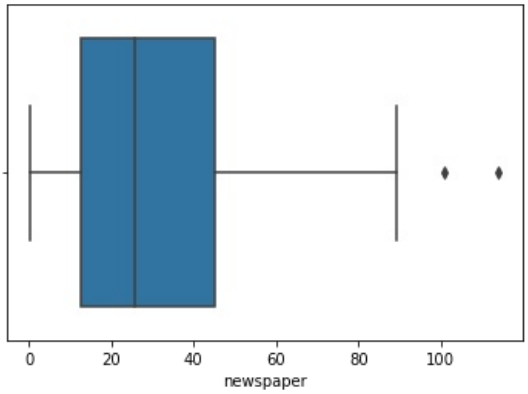
```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```





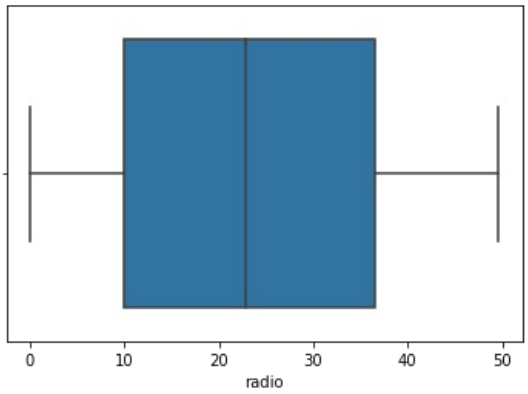
In [76]:
```
sns.boxplot(df['newspaper'])
plt.show()
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [77]:
```
sns.boxplot(df['radio'])
plt.show()
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following va
riable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [78]:
```
## only news paper have outliers
```

In [79]:
```
q1=df.quantile(0.25)
q3=df.quantile(0.75)
print(q1,q3)
```

```
TV           74.375
radio         9.975
newspaper    12.750
sales        10.375
Name: 0.25, dtype: float64 TV           218.825
radio          36.525
newspaper      45.100
sales          17.400
Name: 0.75, dtype: float64
```

In [80]:
```
iqr=q3-q1
print(iqr)
```

```
TV           144.450
radio         26.550
newspaper     32.350
sales          7.025
dtype: float64
```

In [81]:
```
news_paper=(q3.newspaper+(1.5*iqr.newspaper))
```

In [82]:
```
print(news_paper)
```

```
93.625
```

In [83]:
```
index=np.where(df['newspaper']>news_paper)
print(index)
```

```
(array([ 16, 101], dtype=int64),)
```

In [84]:
```
df=df.drop(df.index[index])
```

In [85]:
```
df.shape
```

Out[85]:
```
(198, 4)
```

In [86]:
```
df.reset_index()
```

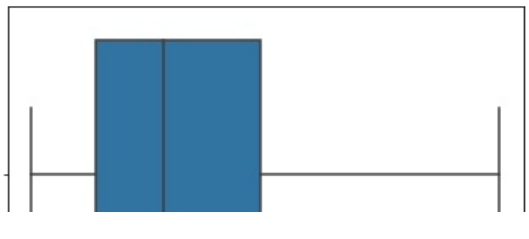
Out[86]:

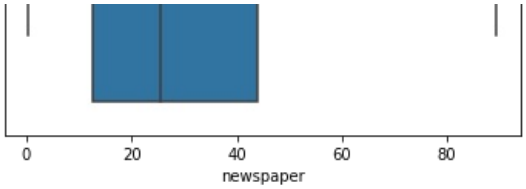
	index	TV	radio	newspaper	sales
0	0	230.1	37.8	69.2	22.1
1	1	44.5	39.3	45.1	10.4
2	2	17.2	45.9	69.3	9.3
3	3	151.5	41.3	58.5	18.5
4	4	180.8	10.8	58.4	12.9
...	...	...	...	...	...
193	195	38.2	3.7	13.8	7.6
194	196	94.2	4.9	8.1	9.7
195	197	177.0	9.3	6.4	12.8
196	198	283.6	42.0	66.2	25.5
197	199	232.1	8.6	8.7	13.4

198 rows × 5 columns

In [87]:
```
sns.boxplot(df['newspaper'])
plt.show()
```

```
C:\Users\Rakesh Lodem\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```





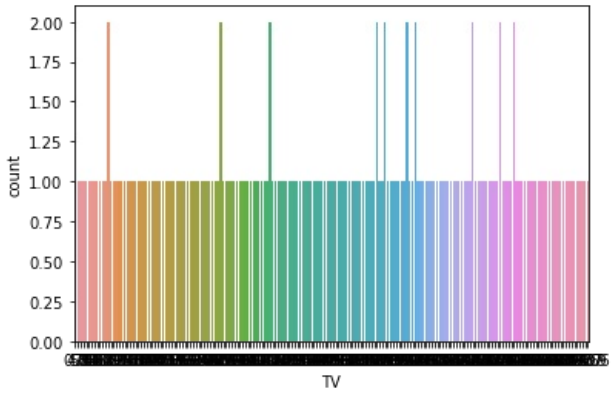
In [88]:
```
df.describe()
```

Out[88]:

	TV	radio	newspaper	sales
count	198.000000	198.000000	198.000000	198.000000
mean	146.688384	23.130808	29.777273	13.980808
std	85.443221	14.862111	20.446303	5.196097
min	0.700000	0.000000	0.300000	1.600000
25%	74.800000	9.925000	12.650000	10.325000
50%	149.750000	22.400000	25.600000	12.900000
75%	218.475000	36.325000	44.050000	17.375000
max	293.600000	49.600000	89.400000	27.000000

In [89]:
```
sns.countplot(x=df['TV'])
```

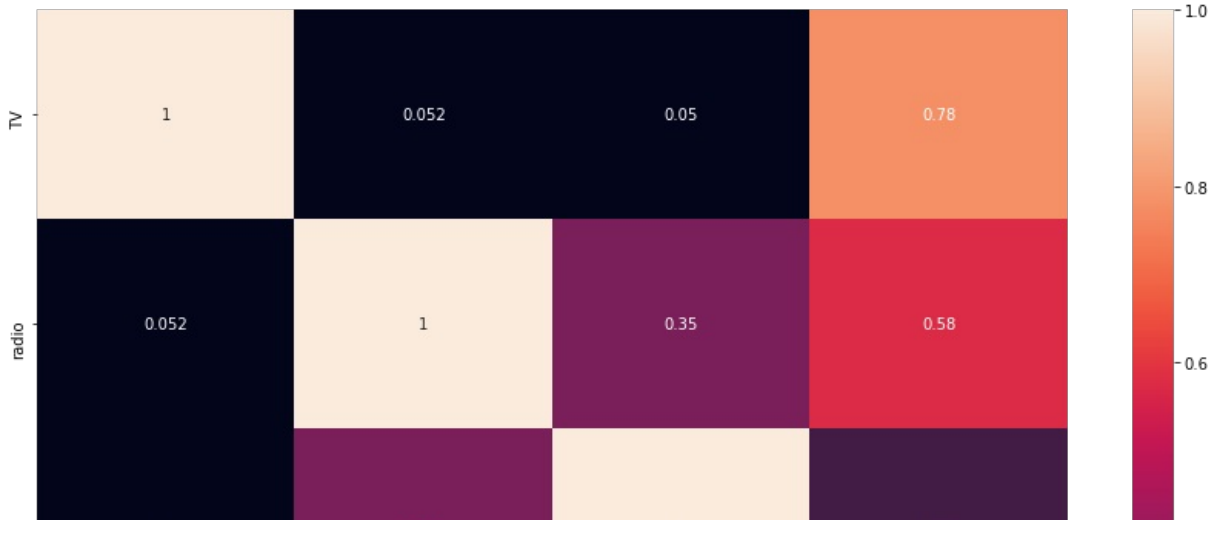
Out[89]:
```
<AxesSubplot:xlabel='TV', ylabel='count'>
```

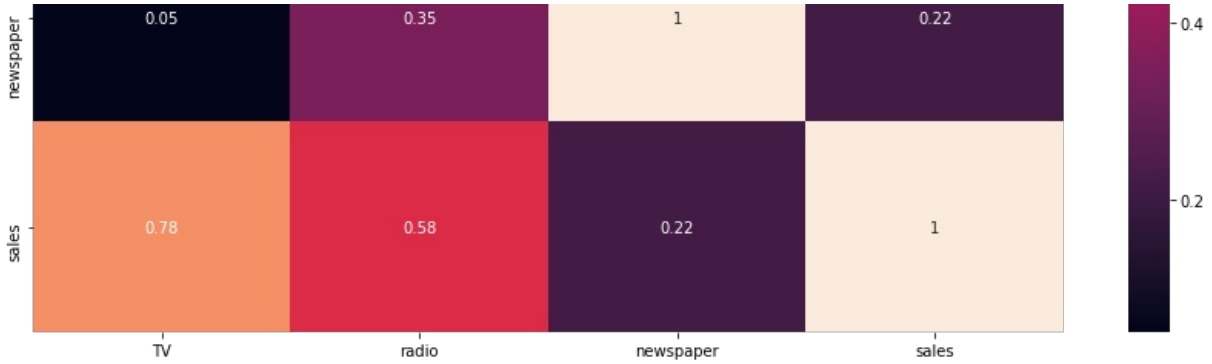


In [90]:
```
df_new=df.corr()
```

In [91]:
```
plt.figure(figsize=(15,10))
sns.heatmap(df_new,annot=True)
```

Out[91]:
```
<AxesSubplot:>
```





In [92]:
```
df_new
```

Out[92]:

	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 [93]:
```
df.shape
```

Out[93]:
```
(198, 4)
```

In [96]:
```
x=df.iloc[:,0:3]
x
```

Out[96]:

	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
...	...	...	...
195	38.2	3.7	13.8
196	94.2	4.9	8.1
197	177.0	9.3	6.4
198	283.6	42.0	66.2
199	232.1	8.6	8.7

198 rows × 3 columns

In [97]:
```
y=df.iloc[:,3]
y
```

Out[97]:
```
0      22.1
1      10.4
2       9.3
3      18.5
4      12.9
       ...
195     7.6
196     9.7
197    12.8
198    25.5
199    13.4
Name: sales, Length: 198, dtype: float64
```

In [98]:
```
from sklearn.preprocessing import StandardScaler
```

In [99]:
```
sc=StandardScaler()
```

In [100…
```
sc.fit_transform(x)
```

Out[100…
```
array([[ 0.97869734,  0.98952135,  1.93299778],
       [-1.19901165,  1.09070498,  0.75131275],
       [-1.51933199,  1.53591293,  1.93790103],
       [ 0.05645636,  1.22561648,  1.40834924],
       [ 0.40024339, -0.83178391,  1.40344598],
       [-1.61906543,  1.73828018,  2.2173867 ],
       [-1.04647815,  0.6522426 , -0.30779084],
       [-0.31079737, -0.23817331, -0.89127846],
       [-1.62023876, -1.41864895, -1.41102374],
       [ 0.62317696, -1.38492107, -0.42056576],
       [-0.94557138, -1.16906267, -0.27346804],
       [ 0.79800381,  0.05863199, -1.26392602],
       [-1.44189191,  0.80739083,  1.77119028],
       [-0.57714432, -1.04764232, -1.10702179],
       [ 0.67363035,  0.65898817,  0.79544207],
       [ 0.57155024,  1.65733328,  1.13376683],
       [ 1.58061798,  1.1109417 ,  1.27596129],
       [-0.90919801, -0.17746313, -0.56276022],
       [ 0.0071763 ,  0.05188642, -0.52353416],
       [ 0.84141719,  0.30821827,  1.15828311],
       [ 1.06435076, -1.21628169, -0.30779084],
       [-1.56626537, -0.48775959,  0.97195933],
       [ 0.95757732, -0.42030384, -0.17540289],
       [-0.99015809, -0.71036356, -0.56276022],
       [ 1.36355108, -1.32421089, -0.50392113],
       [-0.04445042,  0.41614747, -0.84224589],
       [ 1.0960308 , -0.43379499, -0.33721038],
       [ 1.19811091,  0.26774482, -0.33721038],
       [-0.89277132, -0.48101401,  0.54047269],
       [ 1.71555146,  0.34869172,  0.65815086],
       [-0.39645079, -0.38657596,  0.43260102],
       [-0.58066432, -1.4591224 ,  0.01092089],
       [ 1.39523112, -0.21119101, -1.44534654],
       [-0.59826434, -1.46586797, -1.09721527],
       [ 1.6897381 , -1.28373744, -1.04327944],
       [ 1.41048447,  1.39425586, -1.21489345],
       [-0.8446646 ,  1.77200806,  0.7807323 ],
       [-1.21543833,  0.24076252,  0.26098702],
       [ 0.95405731,  0.98277578,  0.10898604],
       [ 0.65485699, -0.05604278,  0.08937301],
       [ 0.35565668,  0.69271605,  0.43750428],
       [ 1.7237648 ,  0.30821827, -1.37179768],
       [ 0.70648372, -0.99367772, -0.16559637],
       [-1.42663856,  0.17330677,  0.66305412],
       [ 0.33336332, -0.04255163,  0.08446975],
       [-0.66866442, -0.89249409,  0.29040656],
       [ 1.09368413,  1.23910763, -0.5529537 ],
       [ 0.94467064, -0.49450516,  0.98666911],
       [-0.9361847 , -0.77107374,  0.34434239],
       [ 0.62317696, -1.35119319,  0.23647073],
       [-0.54311762, -0.91273082, -1.28353905],
       [ 0.8179505 ,  1.25259878,  0.4816336 ],
       [ 0.42136341,  1.55614966,  1.41815575],
       [ 1.36120441,  0.3824196 , -0.6804384 ],
       [ 0.61261695,  1.77200806,  1.4818981 ],
       [-1.63549211,  0.33520057,  0.56989223],
       [-0.12306383, -0.26515561, -0.64611559],
       [ 0.75224376,  1.78549921,  0.38847171],
       [ 0.75107043,  0.42963862, -1.00405338],
       [-1.09341154, -1.42539452, -0.41075924],
       [ 1.34477773,  1.32005453,  1.22202546],
       [ 1.08664412, -0.51474189, -0.12146706],
       [-0.51613092,  0.4363842 , -1.0481827 ],
       [-0.1829039 ,  1.32680011, -0.04301494],
       [-0.91154467, -0.93296754, -1.415927  ],
       [-1.35154514,  0.09910544, -1.35218465],
       [-0.08669046, -0.58219764, -0.95992407],
       [ 1.06435076,  0.29472712, -0.92069801],
       [ 0.82264384,  1.40100143, -0.12637031],
       [ 0.61496362,  0.50383995,  0.43750428],
       [-0.43282416, -0.59568879,  0.09427627],
       [-1.40669187,  0.66573375, -0.51372765],
       [-0.20285059, -1.17580824,  0.07466324],
       [ 0.78275046,  0.09910544, -0.8177296 ],
       [-1.52285199,  1.38751028,  2.92345576],
```

```
[-1.39847853, -1.45237682, -0.44508204],
[-0.30727736,  0.36218287, -0.76379377],
[-1.65778547,  0.45662092, -0.99915012],
[-0.36007742, -1.04089674, -0.32740387],
[-0.82471792,  0.24076252, -0.36662993],
[ 1.09251079, -1.28373744,  0.34924565],
[-0.8376246 , -0.19095428,  0.13350233],
[-0.91858468,  1.44147488,  0.2855033 ],
[ 0.7839238 ,  1.34029126,  0.19724467],
[ 0.54573688, -0.31912021,  1.76138377],
[-0.82589125,  0.29472712, -0.67553514],
[-0.42226415,  1.17839745,  1.63880233],
[-0.6850911 ,  0.15981562,  2.13893458],
[-0.43282416,  1.66407886,  1.06021797],
[-0.14535719, -1.22977284, -1.00405338],
[-1.38557185, -1.4591224 ,  0.15801861],
[ 0.83320385,  0.69946162,  1.43286552],
[ 1.22275093,  0.90182888,  2.08499875],
[-0.46098419, -0.61592551, -0.92560126],
[ 0.19490984,  0.5712957 ,  1.13376683],
[ 0.5973636 , -1.32421089, -1.17076413],
[ 0.44835011, -0.14373526, -0.3813397 ],
[ 1.67800475,  1.29307223,  1.05041145],
[-0.13479718,  1.25259878,  0.79053881],
[ 0.88835058, -1.27024629,  0.98176585],
[ 1.56653796, -0.87900294, -0.41075924],
[ 0.48355014, -0.40006711, -0.58237325],
[ 1.07373744,  0.75342622, -1.20018368],
[-0.10311715,  1.56964081,  1.43286552],
[-1.42781189, -0.81829276, -0.00378888],
[-0.66045107, -1.5400693 , -0.32250061],
[-1.56743871, -1.53332372, -0.20482243],
[ 1.27555099,  0.25425367, -1.19037716],
[ 0.92824395, -1.00716887,  1.31028409],
[ 1.11480415,  1.0030125 , -0.32250061],
[ 0.34040333, -0.52148746, -1.34237814],
[ 0.73816375, -0.17071756, -0.93540778],
[-0.80359789,  1.59662311,  0.23156747],
[-0.83997127,  0.80064525,  1.12396031],
[-0.0878638 , -0.59568879, -0.20482243],
[-0.82471792, -1.50634142, -0.73437423],
[-0.24626397,  0.92881118,  2.42332351],
[-1.49351863, -0.48101401, -0.36662993],
[-0.06322377,  0.2475081 ,  0.80524858],
[-1.50055864, -0.09651623,  1.01118539],
[ 0.90712393, -1.39841222, -0.69514817],
[-0.27677067,  0.77366295, -0.8520524 ],
[ 0.97165733,  0.61851472,  2.17816064],
[-0.69799778, -0.76432816, -0.19011266],
[-1.62962544,  1.06372268,  1.02099191],
[-0.7801312 , -1.56030602, -1.00895664],
[ 0.86371055,  1.74502576, -1.30315208],
[-1.02183813, -0.75083701,  0.65324761],
[-1.7129322 ,  1.1109417 , -1.03347293],
[ 1.39053778, -1.36468434,  0.64834435],
[-1.62258543,  0.2744904 , -1.35708791],
[ 0.85784388,  0.69946162,  0.75131275],
[-1.28818508,  1.04348595,  1.75648051],
[-1.15442493,  1.61011426, -1.04327944],
[-1.42077188,  1.07046825, -1.00405338],
[ 1.49027122,  0.38916517,  1.46718833],
[-1.21661167,  0.18679792, -0.45488856],
[ 0.44835011,  1.40100143, -1.37670094],
[-0.85991795, -0.41355826, -0.82753612],
[ 0.55160355,  0.82762755,  2.24680625],
[ 0.86605722,  0.6792249 ,  0.39827822],
[-0.49383756, -1.17580824,  0.22666422],
[-0.59239767, -0.56196091,  0.4473108 ],
[-0.07495712, -1.4321401 , -1.01876315],
[ 1.0960308 , -1.06787904, -1.03347293],
[ 1.13240417,  1.74502576,  0.71208669],
[-1.2752784 ,  1.15816073, -0.87656869],
[-1.19666498,  0.18005234, -0.4499853 ],
[ 1.57240464, -0.62267109,  0.35414891],
[-0.30141069, -0.99367772,  0.92783002],
[ 0.5973636 ,  0.01141297, -0.76379377],
[ 0.2887766 ,  1.11768728,  0.38847171],
[ 0.48237681, -0.13698968, -0.99424687],
[-1.67303882, -0.77781931, -1.18057065],
[-0.61938436,  1.37401913,  1.01608865],
[ 0.03650967, -1.47261355, -0.26856478],
[-1.58386539,  0.92881118,  0.75621601],
[-0.17586389, -0.31912021,  0.23647073],
```

```
       [ 0.30285662, -0.33935694,  0.04524369],
       [-0.7155978 ,  0.85460985,  0.95724956],
       [ 0.48941682, -0.33935694, -0.20482243],
       [ 0.19725651,  0.9220656 , -1.09721527],
       [-0.34599741, -0.56870649, -1.19528042],
       [ 1.03032406, -1.33095647,  2.69790592],
       [-1.51111865,  0.9760302 , -0.40095273],
       [ 0.70531038, -1.20953612, -0.50882439],
       [ 0.80621716,  0.03164969,  1.36421992],
       [ 1.61464468, -0.84527507, -1.14624784],
       [-1.13447825, -0.77781931, -0.55785696],
       [ 0.20898985, -0.15048083,  0.86408767],
       [-1.49117196, -0.20444543, -0.62650256],
       [ 0.2547499 , -1.08137019, -0.83243937],
       [ 0.88835058, -1.33095647, -0.8177296 ],
       [ 1.52781792,  1.73828018,  0.58950526],
       [ 1.19341757,  0.47685765, -0.46469507],
       [ 0.27586992, -1.03415117,  0.26589027],
       [ 1.52547125, -1.4051578 , -0.29798432],
       [ 0.22189653, -0.88574852, -0.59708302],
       [ 0.11629642, -1.38492107, -1.05308596],
       [ 0.84259053, -1.19604497, -0.1165638 ],
       [-1.0617315 , -1.17580824, -0.00378888],
       [ 1.65336472,  1.34029126,  2.06048247],
       [ 1.25677764, -0.12349853,  0.01092089],
       [ 0.68419036,  1.48194833, -0.49901787],
       [-0.08434379, -1.41864895, -0.15578986],
       [ 0.52109685,  0.37567402, -0.56766348],
       [ 1.63459137, -0.62267109, -1.27863579],
       [-1.50173197, -0.74409144, -0.31269409],
       [-1.25767838,  1.21212533, -1.17566739],
       [-0.83527793, -0.83178391, -1.16586087],
       [-1.51933199, -1.28373744,  0.08937301],
       [ 0.23597655,  1.2728355 , -1.28353905],
       [ 0.03533633,  0.8411187 , -1.16586087],
       [-1.27293173, -1.31071974, -0.7834068 ],
       [-0.61586436, -1.22977284, -1.06289247],
       [ 0.35565668, -0.93296754, -1.14624784],
       [ 1.60643134,  1.2728355 ,  1.78590005],
       [ 1.00216403, -0.98018657, -1.03347293]])
```

In [107…
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,mean_squared_error,mean_absolute_error
```

In [108…
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [109…
```
from sklearn.linear_model import LinearRegression
```

In [110…
```
lr=LinearRegression()
```

In [111…
```
lr.fit(x_train,y_train)
```

Out[111…
```
LinearRegression()
```

In [112…
```
y_pred=lr.predict(x_test)
```

In [113…
```
y_test
```

Out[113…
```
19     14.6
170     8.4
64     18.0
177    11.7
72      8.8
87     16.0
5       7.2
120    15.5
12      9.2
152    16.6
61     24.2
76      6.9
165    11.9
```

```
97     15.5
115    12.6
7      13.2
34      9.5
136     9.5
38     10.1
168    17.1
111    21.8
145    10.3
46     10.6
159    12.9
140    10.9
113    15.9
179    12.6
185    22.6
93     22.2
45     14.9
17     24.4
137    20.8
138     9.6
98     25.4
23     15.5
196     9.7
128    24.7
4      12.9
67     13.4
125    10.6
133    19.6
175    27.0
27     15.9
62     15.7
147    25.4
84     21.7
8       4.8
143    10.4
75      8.7
189     6.7
Name: sales, dtype: float64
```

In [114…
```
y_pred
```

Out[114…
```
array([13.94883141,  7.34343318, 16.85024368, 12.11985798, 10.31363072,
       15.87389799, 13.02218046, 14.48960609, 11.02939854, 16.0151057 ,
       22.84708206,  4.50657856, 14.63225056, 15.10092361, 13.15168238,
       11.87450944,  7.35720088, 11.28058613, 10.01596085, 17.28458841,
       20.79469039,  9.43997415,  8.9533758 , 12.38548614,  9.33146006,
       16.00868346, 12.15359972, 20.45828315, 21.40402617, 15.03994291,
       23.16973211, 20.87313577,  9.71920857, 23.99511936, 16.29883021,
        7.94994889, 21.70134219, 13.37568538, 11.75565084,  9.10067399,
       19.20167218, 24.56638869, 16.75142364, 16.5355503 , 23.11806603,
       20.58242625,  3.57001665,  8.8173416 , 12.55728611,  6.10040077])
```

In [117…
```
mean_absolute_error(y_pred,y_test)
```

Out[117…
```
1.2704174584059549
```

In [118…
```
mean_squared_error(y_pred,y_test)
```

Out[118…
```
2.7246533749571156
```

In [121…
```
##
```

In [122…
```
from sklearn.ensemble import RandomForestRegressor
```

In [123…
```
rf=RandomForestRegressor()
```

In [124…
```
rf.fit(x_train,y_train)
```

Out[124… RandomForestRegressor()

In [125…
```
b=rf.score(x_train,y_train)
```

In [126…
```
print(b)
```

```
0.9959871003851948
```

In [127…
```
y_pred=rf.predict(x_test)
```

In [128…
```
y_pred
```

Out[128…
```
array([15.515,  9.385, 16.581, 12.544,  8.399, 15.275,  7.238, 15.54 ,
        8.358, 17.083, 24.632,  7.405, 12.114, 15.567, 13.187, 12.688,
        9.493,  8.302, 10.388, 17.571, 21.429, 10.407, 10.178, 12.846,
       11.09 , 15.89 , 13.842, 22.614, 21.071, 16.081, 22.684, 20.266,
       10.186, 24.852, 14.956,  9.614, 23.132, 14.003, 12.686, 10.198,
       19.8  , 24.749, 15.052, 15.07 , 23.568, 22.939,  5.086, 10.509,
        8.388,  6.869])
```

In [131…
```
y_test
```

Out[131…
```
19     14.6
170     8.4
64     18.0
177    11.7
72      8.8
87     16.0
5       7.2
120    15.5
12      9.2
152    16.6
61     24.2
76      6.9
165    11.9
97     15.5
115    12.6
7      13.2
34      9.5
136     9.5
38     10.1
168    17.1
111    21.8
145    10.3
46     10.6
159    12.9
140    10.9
113    15.9
179    12.6
185    22.6
93     22.2
45     14.9
17     24.4
137    20.8
138     9.6
98     25.4
23     15.5
196     9.7
128    24.7
4      12.9
67     13.4
125    10.6
133    19.6
175    27.0
27     15.9
62     15.7
147    25.4
84     21.7
8       4.8
143    10.4
75      8.7
```

```
189     6.7
Name: sales, dtype: float64
```

In [142…

```
from sklearn.linear_model import Ridge
r=Ridge(alpha=0.05,solver='cholesky')
r.fit(x_train,y_train)
predict_r=r.predict(x_test)
mse=mean_squared_error(y_test,predict_r)
mse
r_score=np.sqrt(mse)
r_score
```

Out[142… 1.6506527662941695

In [134…

```
from sklearn.ensemble import GradientBoostingRegressor
gbr=GradientBoostingRegressor()
gbr.fit(x_train,y_train)
p=gbr.predict(x_test)
gb_score=mean_squared_error(y_test,p)
gb_score=np.sqrt(gb_score)
gb_score
```

Out[134… 0.7428414859109423

In [135…

```
import numpy as np
a=np.array(y_test)
predicted=np.array(rf.predict(x_test))
df_con=pd.DataFrame({'true':a,'predicted':predicted},index=range(len(a)))
df_con
```

Out[135…

	true	predicted
0	14.6	15.515
1	8.4	9.385
2	18.0	16.581
3	11.7	12.544
4	8.8	8.399
5	16.0	15.275
6	7.2	7.238
7	15.5	15.540
8	9.2	8.358
9	16.6	17.083
10	24.2	24.632
11	6.9	7.405
12	11.9	12.114
13	15.5	15.567
14	12.6	13.187
15	13.2	12.688
16	9.5	9.493
17	9.5	8.302
18	10.1	10.388
19	17.1	17.571
20	21.8	21.429
21	10.3	10.407
22	10.6	10.178
23	12.9	12.846
24	10.9	11.090
25	15.9	15.890
26	12.6	13.842
27	22.6	22.614

28	22.2	21.071
29	14.9	16.081
30	24.4	22.684
31	20.8	20.266
32	9.6	10.186
33	25.4	24.852
34	15.5	14.956
35	9.7	9.614
36	24.7	23.132
37	12.9	14.003
38	13.4	12.686
39	10.6	10.198
40	19.6	19.800
41	27.0	24.749
42	15.9	15.052
43	15.7	15.070
44	25.4	23.568
45	21.7	22.939
46	4.8	5.086
47	10.4	10.509
48	8.7	8.388
49	6.7	6.869

In [136…

```
import numpy as np
a=np.array(y_test)
predicted=np.array(lr.predict(x_test))
df_con=pd.DataFrame({'true':a,'predicted':predicted},index=range(len(a)))
df_con
```

Out[136…

	true	predicted
0	14.6	13.948831
1	8.4	7.343433
2	18.0	16.850244
3	11.7	12.119858
4	8.8	10.313631
5	16.0	15.873898
6	7.2	13.022180
7	15.5	14.489606
8	9.2	11.029399
9	16.6	16.015106
10	24.2	22.847082
11	6.9	4.506579
12	11.9	14.632251
13	15.5	15.100924
14	12.6	13.151682
15	13.2	11.874509
16	9.5	7.357201
17	9.5	11.280586
18	10.1	10.015961
19	17.1	17.284588
20	21.8	20.794690
21	10.3	9.439974
22	10.6	8.953376
23	12.9	12.385486
24	10.9	9.331460
25	15.9	16.008683

26	12.6	12.153600
27	22.6	20.458283
28	22.2	21.404026
29	14.9	15.039943
30	24.4	23.169732
31	20.8	20.873136
32	9.6	9.719209
33	25.4	23.995119
34	15.5	16.298830
35	9.7	7.949949
36	24.7	21.701342
37	12.9	13.375685
38	13.4	11.755651
39	10.6	9.100674
40	19.6	19.201672
41	27.0	24.566389
42	15.9	16.751424
43	15.7	16.535550
44	25.4	23.118066
45	21.7	20.582426
46	4.8	3.570017
47	10.4	8.817342
48	8.7	12.557286
49	6.7	6.100401

In [137…

```
import numpy as np
a=np.array(y_test)
predicted=np.array(gbr.predict(x_test))
df_con=pd.DataFrame({'true':a,'predicted':predicted},index=range(len(a)))
df_con
```

Out[137…

	true	predicted
0	14.6	14.846956
1	8.4	8.913325
2	18.0	16.369102
3	11.7	12.301679
4	8.8	8.560825
5	16.0	15.677984
6	7.2	8.681306
7	15.5	15.181145
8	9.2	8.114174
9	16.6	16.793422
10	24.2	24.650158
11	6.9	7.016469
12	11.9	12.624875
13	15.5	15.254235
14	12.6	13.179129
15	13.2	12.070850
16	9.5	9.309880
17	9.5	8.277391
18	10.1	10.090564
19	17.1	17.218332
20	21.8	22.125686
21	10.3	10.326069
22	10.6	10.446757

23	12.9	12.358582
24	10.9	10.768312
25	15.9	15.864652
26	12.6	13.932289
27	22.6	22.375519
28	22.2	20.936724
29	14.9	15.787048
30	24.4	24.192463
31	20.8	21.305115
32	9.6	10.059444
33	25.4	25.729549
34	15.5	15.284627
35	9.7	10.064867
36	24.7	23.665672
37	12.9	14.903186
38	13.4	12.766300
39	10.6	10.457595
40	19.6	19.758423
41	27.0	25.834425
42	15.9	15.532820
43	15.7	15.301929
44	25.4	23.960985
45	21.7	22.838379
46	4.8	4.979050
47	10.4	10.462531
48	8.7	9.461049
49	6.7	6.754194

In [138…

```
import pickle
```

In [139…

```
filename='ADVERTISING CHANNEL PREDICTION.pkl'
pickle.dump(lr,open(filename,'wb'))
```

In [ ]: