In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
df = pd.read_csv('C:\\Users\\acer\\Desktop\\csv_files\\Advertising.csv')
```

In [5]:
```
df.head()
```

Out[5]:

	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 [6]:
```
df.shape #checking the data shape
```

Out[6]:
```
(200, 5)
```

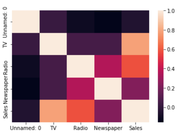
In [8]:
```
df.isnull().sum() #checking the null values
```

Out[8]:
```
Unnamed: 0    0
TV            0
Radio         0
Newspaper     0
Sales         0
dtype: int64
```

In [12]:
```
print(df.corr()) #correlation between the independent variables
sns.heatmap(df.corr())
```

```
             Unnamed: 0        TV     Radio  Newspaper     Sales
Unnamed: 0    1.000000  0.017715 -0.110680  -0.154944 -0.051616
TV            0.017715  1.000000  0.054809   0.056648  0.782224
Radio        -0.110680  0.054809  1.000000   0.354104  0.576223
Newspaper    -0.154944  0.056648  0.354104   1.000000  0.228299
Sales        -0.051616  0.782224  0.576223   0.228299  1.000000
```

Out[12]:
```
<AxesSubplot:>
```



In [13]:
```
X = df.iloc[:, 2:]
y = df.iloc[:,-1]
```

In [16]:
```
#splitting the data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=0)
```

In [17]:
```
#scaling the data

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)
X_train
```

Out[17]:
```
array([[ 1.0355176 ,  1.65941078, -0.67282365],
       [ 0.08249594, -1.30629738, -0.92876242],
       [ 0.40243892, -0.81990897,  0.15405545],
       [-0.18979597, -0.90868666,  0.33124383],
       [ 0.01442296,  1.28518893,  0.567495  ],
       [ 0.42280682, -1.01627544, -0.04282052],
       [-1.44914602, -1.36243065, -1.85407951],
       [ 1.38268979,  2.77272078, -1.08626452],
       [ 0.91979354,  2.29558792,  0.33124383],
       [-1.20408331,  0.19994556, -0.75157404],
       [-0.61865571,  0.07364569, -0.35782209],
       [ 0.60665785,  2.06169926,  1.07937254],
       [-0.14214488, -0.00587645,  0.66593299],
       [-0.16255678, -0.3800983 ,  0.25249344],
       [ 0.16418351, -0.44558712, -0.81063683],
       [-0.73437977, -0.55317591, -0.88938722],
       [ 0.33436595,  0.6115896 ,  1.41140617],
       [-0.35997841,  0.02686796,  0.03592987],
       [-0.07407191,  0.06896792,  0.46905701],
       [ 0.32075133,  0.52738968, -1.71626632],
       [-1.39468764,  0.60223405, -0.29875929],
       [-1.35384385, -1.13321977, -0.49563527],
       [-1.2245052 , -0.12749855, -0.39719728],
       [ 0.03484485, -0.51575372,  0.07530506],
       [ 1.11039787,  0.35431208,  0.94155936],
       [ 0.68834542,  1.35067775,  1.02030975],
       [ 0.36840244, -0.66544246,  1.17781053],
       [ 0.91298624, -1.06305318,  0.74468338],
       [-0.9385987 , -1.24080855, -0.69251125],
       [ 0.27990757, -0.66076469, -0.43657248],
       [ 1.39630437, -1.32968624,  1.27624852],
       [-1.25854169, -0.97417549, -0.59407326],
       [ 0.37520973,  1.38342216,  1.29593611],
       [ 0.87894975,  3.31066468,  1.88656404],
       [-0.58461923,  0.41044535, -0.55469806],
       [ 1.66178898,  0.99516699,  0.48874461],
       [ 0.23906378, -1.15193086,  1.09960114],
       [ 0.64750164,  0.74254725,  0.94155936],
       [-0.45528057, -0.33799834,  0.33124383],
       [ 0.81768608,  2.12713809,  0.98093455],
       [ 0.36159514, -0.55785368,  0.6068702 ],
       [-0.91817681,  0.26075661, -0.71219884],
       [ 0.17099081, -0.4502649 , -0.90907482],
       [-1.34022926, -0.7636757 , -1.30282677],
       [-1.56487008, -0.21163847, -1.75564152],
       [-0.53696814, -0.13217633,  0.29186863],
       [ 1.53245032,  1.83248838, -0.96813761],
       [ 0.41605352, -0.97417549,  0.82343377],
       [ 1.06274679, -0.97417549, -0.92876242],
       [-0.95902059, -1.10983091, -0.27907217 ],
       [-1.02028627, -0.17427628, -0.2593841 ],
       [-1.19727601, -0.27718729, -1.1095508 ],
       [ 0.83129867, -1.128542  ,  0.6068702 ],
       [-1.20408331,  0.0549346 , -0.63344845],
       [-1.25854169, -1.03030876, -0.88938722],
       [-1.43553142, -0.30057616, -0.47594767],
       [ 1.73666925,  0.54610077,  2.51656717],
       [ 1.05593949,  0.95774481, -1.49970275],
       [ 0.89256435,  1.97282157,  1.57156248],
       [-0.54377544, -1.29694183, -0.02313293],
       [ 0.91979354,  0.70514506, -1.36188957],
       [-0.83689113, -1.128542  , -0.85001203],
       [-1.38107304,  0.20930111, -0.55469806],
       [ 0.98105922,  1.82781061,  1.55187488],
       [-0.63907761, -0.89933112, -0.53501047],
       [-1.03390087,  1.23373342, -0.16094611],
       [ 0.55900677,  1.06533359,  0.52811981],
       [ 1.3214241 , -0.05733196,  0.74468338],
       [-1.23811979, -0.5017204 , -0.39719728],
       [ 0.68153812,  0.40108981,  0.567495  ],
       [ 1.39630437, -0.1368541 ,  1.59125008],
       [-0.79564545,  0.31222122, -0.88938722],
       [-0.51654625,  0.9250004 ,  0.11468026],
       [-0.33955651,  1.66408855,  0.19343065],
       [-1.07474465, -1.07240872, -0.88938722],
       [ 0.84491327,  0.89693376, -0.18063371],
       [-0.44146598, -0.1836183 ,  0.25249344],
       [-1.44914602, -0.16490274, -0.77126144],
       [ 1.6073306 , -1.01159767, -0.51532287],
       [ 0.29352216,  1.08872246,  0.74468338],
       [ 1.65498168,  1.06533359,  1.61093768],
       [ 0.44328271, -0.96949771, -1.75564152],
       [ 0.7428038 , -1.16128641,  1.27624852],
       [ 1.24654382,  0.44318977,  1.65031287],
       [-1.29938547,  0.92032262, -0.49563527],
       [-1.48318251, -0.44090935, -1.14063996],
       [ 1.21931463,  1.32728888,  0.84312137],
       [-0.61865571, -0.21169847, -0.39719728],
       [-0.59142652, -1.15660844, -0.45626007],
       [ 1.26696572,  1.68747741,  2.2212532 ],
       [ 0.64069434, -0.3099317 , -0.47594767],
       [ 0.27990757, -0.89445334,  0.92187176],
       [ 1.33503869,  0.17187893,  1.47312449],
       [-0.11491569,  0.94838926, -1.42095236],
       [-0.80245278, -1.14257532, -2.16908107],
       [-0.16937407,  0.80805607,  0.05561746],
       [ 1.36907518,  0.95306704,  0.21313182],
       [-1.4899898 ,  0.13445674, -1.36188957],
       [-1.00667168, -1.00224213, -0.16094611],
       [ 0.04165215, -1.22209746,  0.62455779],
       [-0.50293165, -0.36606498, -1.49970275],
       [-0.90456221, -0.40816494,  0.11468026],
       [-1.31300007, -1.01159767, -0.2790717 ],
       [-0.2850983 , -0.63269805, -0.20032131],
       [-0.95902059, -1.36710843, -0.96813761],
       [ 1.08316868,  0.70046729, -0.75157404],
       [ 0.23225648,  0.75192279,  0.25249344],
       [-0.35997841, -0.21169847,  0.13436785],
       [ 0.29352216, -1.32500847,  1.27624852],
       [ 1.36648681,  1.35067775,  0.98093455],
       [ 0.25948567, -1.31097515, -1.67689133],
       [-1.10878114, -0.81045343, -0.49563527],
       [ 1.10359057,  1.20098901,  2.00468963],
       [ 0.02154919,  0.23268998, -0.81063683],
       [ 0.68834542,  0.70046729,  1.05968494],
       [ 1.55287221,  1.33664443,  1.37768651 ],
       [-1.5036044 , -0.27250952, -0.81063683],
       [ 1.74347655, -1.25951965,  2.06375242],
       [-1.4967971 , -1.06305318, -0.92876242],
       [ 0.25267838, -0.33799834,  0.92187176],
       [ 1.593716  ,  0.20462334,  0.07530506],
       [-1.02028627,  0.86884712, -0.51532287],
       [-0.40762949,  0.39641204, -0.45626007],
       [ 0.96744462, -0.39880939, -1.22407638],
       [-1.59209927, -0.97885326, -1.06657561 ],
       [ 0.4637046 , -0.45962044,  1.17781053],
       [-1.4899898 , -0.00587645, -0.90907482],
       [ 0.66792353,  0.36366762,  1.15812293],
       [-0.87052572, -1.10983091,  0.15405545],
       [ 1.74347655,  0.6630451 ,  2.20015656 ],
       [-0.50293165,  0.49932304, -0.73188644],
       [ 1.24654382,  0.73788947,  0.5871826 ],
       [-0.21021786,  0.11106788, -0.57438566],
       [-1.06793735, -0.3286428 , -0.63344845],
       [ 0.79045489,  1.05597804, -0.31844689],
       [ 1.15124166, -0.85255339, -0.65313605],
       [ 1.26696572, -1.24080855,  1.05968494],
       [ 0.65430893, -0.50639818, -1.06657561 ],
       [ 0.34798054, -0.7449464 , -0.00344533],
       [-1.35384385, -0.49704263, -0.43657248],
       [-1.36065115,  2.55754321, -0.45626007],
       [-1.31300007,  0.31688989, -0.37750968],
       [-0.6458849 , -1.23613078,  0.33124383],
       [-1.36065115, -0.79642011, -0.53501047],
       [ 1.20570004, -1.13789755, -0.67282365],
       [ 0.97425192,  0.08767901,  1.4337693 ],
       [ 1.78432033,  0.35431208,  1.88656404],
       [-0.43485868, -0.80577565, -0.65313605],
       [ 0.14376162,  2.02427708, -0.2593841 ],
       [ 0.49093379,  0.40108981,  0.80374617],
       [ 1.17166355,  1.54714422,  0.35093143],
       [ 1.38949707, -1.17531973,  2.20015656 ],
       [-1.24492709, -0.3099317 , -0.33813449],
       [-1.41510953, -0.41752049, -0.71219884],
       [-0.42124408, -0.571887  ,  0.09499266],
       [-0.60504112, -0.93207553, -0.16094611],
       [-1.31300007,  0.06896792, -1.63751593],
       [-1.53764089, -0.71689797, -0.94845002],
       [ 1.23292923, -0.54382036,  1.76843846],
       [-0.22383245, -0.61398696, -1.30282677]])
```

In [18]:
```
#building the model
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
y_pred
```

Out[18]:
```
array([12.45867038,  9.83287585, 10.10450977, 25.2254645 , 12.82084894,
       10.10450977,  8.74634018, 14.17901853, 10.55723296, 17.25753626,
       24.13892883, 11.82485790, 11.73431327, 16.35208987, 12.91139358,
       14.17901853, 17.98189337,  3.67584039, 15.53718812, 17.61971482,
       25.94982162, 11.55322399, 15.71827739, 13.90738461,  9.56124193,
       15.98991131, 13.63575069, 22.69021461, 12.7303043 ,  9.92342049,
       13.54520605, 23.68620564, 16.80481306, 21.96585749,  7.29762595,
        8.29361698, 11.00995616, 13.90738461, 14.54119708,  8.74634018])
```

In [19]:
```
from sklearn.metrics import r2_score #accuracy of the model
r2_score(y_pred, y_test)
```

Out[19]:
```
0.9522485052659027
```

In [25]:
```
#building the SVR model
from sklearn.svm import SVR
svr = SVR()
svr.fit(X_train, y_train)
y_predd = svr.predict(X_test)
print('accuracy of the svr model',r2_score(y_predd, y_test))
```

accuracy of the svr model 0.7896349424098883

In [26]:
```
#lasso reg
from sklearn.linear_model import Lasso
l = Lasso()
l.fit(X_train, y_train)
y_predd = l.predict(X_test)
print('accuracy of the lasso model', r2_score(y_predd, y_test))
```

accuracy of the lasso model 0.8021262005842481

In [28]:
```
#ridge reg
from sklearn.linear_model import Ridge
r = Ridge()
r.fit(X_train, y_train)
r_predr = r.predict(X_test)
print('accuracy of the ridge model',r2_score(r_predr, y_test))
```

accuracy of the ridge model 0.9500946540839276

In [30]:
```
# from sklearn.model_selection import KFold
# cv = KFold(n_splits=10, random_state=1, shuffle=True)
```

In [ ]:
```
```