In [6]:
```
import numpy as pd
import pandas as pd
numpy.__version__
```

Out[6]: '1.26.4'

In [50]:
```
data = pd.read_csv(r'D:\Pycharm_Projects\Projects\pythonProject\data.csv')
data.head()
data.tail()
data.info
data.describe()
```

Out[50]:

	Volume	Weight	CO2	Unnamed: 5
count	36.000000	36.000000	36.000000	0.0
mean	1611.111111	1292.277778	102.027778	NaN
std	388.975047	242.123889	7.454571	NaN
min	900.000000	790.000000	90.000000	NaN
25%	1475.000000	1117.250000	97.750000	NaN
50%	1600.000000	1329.000000	99.000000	NaN
75%	2000.000000	1418.250000	105.000000	NaN
max	2500.000000	1746.000000	120.000000	NaN

In [54]:
```
data.fillna(0)
```

Out[54]:

	Car	Model	Volume	Weight	CO2	Unnamed: 5
0	Toyota	Aygo	1000	790	99	0.0
1	Mitsubishi	Space Star	1200	1160	95	0.0
2	Skoda	Citigo	1000	929	95	0.0
3	Fiat	500	900	865	90	0.0
4	Mini	Cooper	1500	1140	105	0.0
5	VW	Up!	1000	929	105	0.0
6	Skoda	Fabia	1400	1109	90	0.0
7	Mercedes	A-Class	1500	1365	92	0.0
8	Ford	Fiesta	1500	1112	98	0.0
9	Audi	A1	1600	1150	99	0.0
10	Hyundai	I20	1100	980	99	0.0
11	Suzuki	Swift	1300	990	101	0.0
12	Ford	Fiesta	1000	1112	99	0.0
13	Honda	Civic	1600	1252	94	0.0
14	Hundai	I30	1600	1326	97	0.0
15	Opel	Astra	1600	1330	97	0.0
16	BMW	1	1600	1365	99	0.0
17	Mazda	3	2200	1280	104	0.0
18	Skoda	Rapid	1600	1119	104	0.0
19	Ford	Focus	2000	1328	105	0.0
20	Ford	Mondeo	1600	1584	94	0.0
21	Opel	Insignia	2000	1428	99	0.0
22	Mercedes	C-Class	2100	1365	99	0.0
23	Skoda	Octavia	1600	1415	99	0.0
24	Volvo	S60	2000	1415	99	0.0
25	Mercedes	CLA	1500	1465	102	0.0
26	Audi	A4	2000	1490	104	0.0
27	Audi	A6	2000	1725	114	0.0
28	Volvo	V70	1600	1523	109	0.0
29	BMW	5	2000	1705	114	0.0

	Car	Model	Volume	Weight	CO2	Unnamed: 5
30	Mercedes	E-Class	2100	1605	115	0.0
31	Volvo	XC70	2000	1746	117	0.0
32	Ford	B-Max	1600	1235	104	0.0
33	BMW	2	1600	1390	108	0.0
34	Opel	Zafira	1600	1405	109	0.0
35	Mercedes	SLK	2500	1395	120	0.0

In [56]:
```
data= data.dropna(axis=1)
data
```

Out[56]:

	Car	Model	Volume	Weight	CO2
0	Toyota	Aygo	1000	790	99
1	Mitsubishi	Space Star	1200	1160	95
2	Skoda	Citigo	1000	929	95
3	Fiat	500	900	865	90
4	Mini	Cooper	1500	1140	105
5	VW	Up!	1000	929	105
6	Skoda	Fabia	1400	1109	90
7	Mercedes	A-Class	1500	1365	92
8	Ford	Fiesta	1500	1112	98
9	Audi	A1	1600	1150	99
10	Hyundai	I20	1100	980	99
11	Suzuki	Swift	1300	990	101
12	Ford	Fiesta	1000	1112	99
13	Honda	Civic	1600	1252	94
14	Hundai	I30	1600	1326	97
15	Opel	Astra	1600	1330	97
16	BMW	1	1600	1365	99
17	Mazda	3	2200	1280	104
18	Skoda	Rapid	1600	1119	104
19	Ford	Focus	2000	1328	105
20	Ford	Mondeo	1600	1584	94
21	Opel	Insignia	2000	1428	99
22	Mercedes	C-Class	2100	1365	99
23	Skoda	Octavia	1600	1415	99
24	Volvo	S60	2000	1415	99
25	Mercedes	CLA	1500	1465	102
26	Audi	A4	2000	1490	104
27	Audi	A6	2000	1725	114
28	Volvo	V70	1600	1523	109
29	BMW	5	2000	1705	114

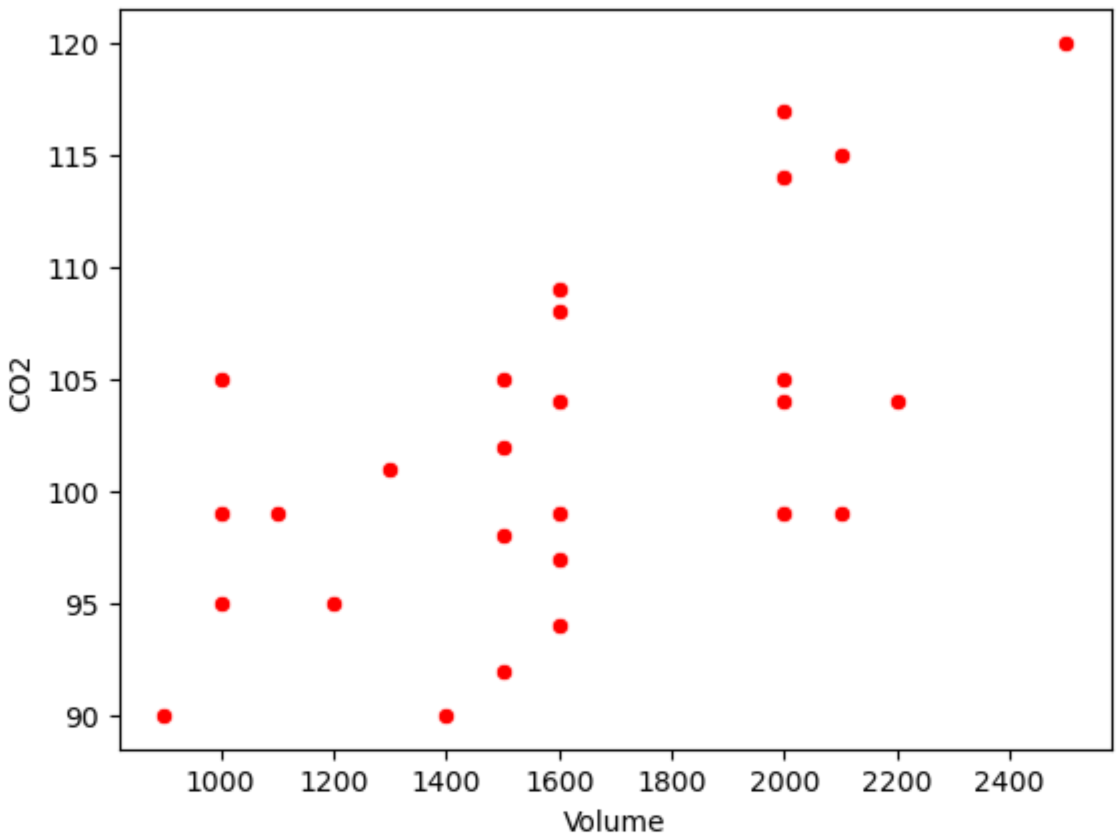
	Car	Model	Volume	Weight	CO2
30	Mercedes	E-Class	2100	1605	115
31	Volvo	XC70	2000	1746	117
32	Ford	B-Max	1600	1235	104
33	BMW	2	1600	1390	108
34	Opel	Zafira	1600	1405	109
35	Mercedes	SLK	2500	1395	120

In [57]:
```
data.shape
```
Out[57]:
```
(36, 5)
```

In [58]:
```
data.isna().sum()
```
Out[58]:
```
Car       0
Model     0
Volume    0
Weight    0
CO2       0
dtype: int64
```

In [59]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [62]:
```
sns.scatterplot(x='Volume',y='CO2', data=data, color='r')
```
Out[62]:
```
<Axes: xlabel='Volume', ylabel='CO2'>
```



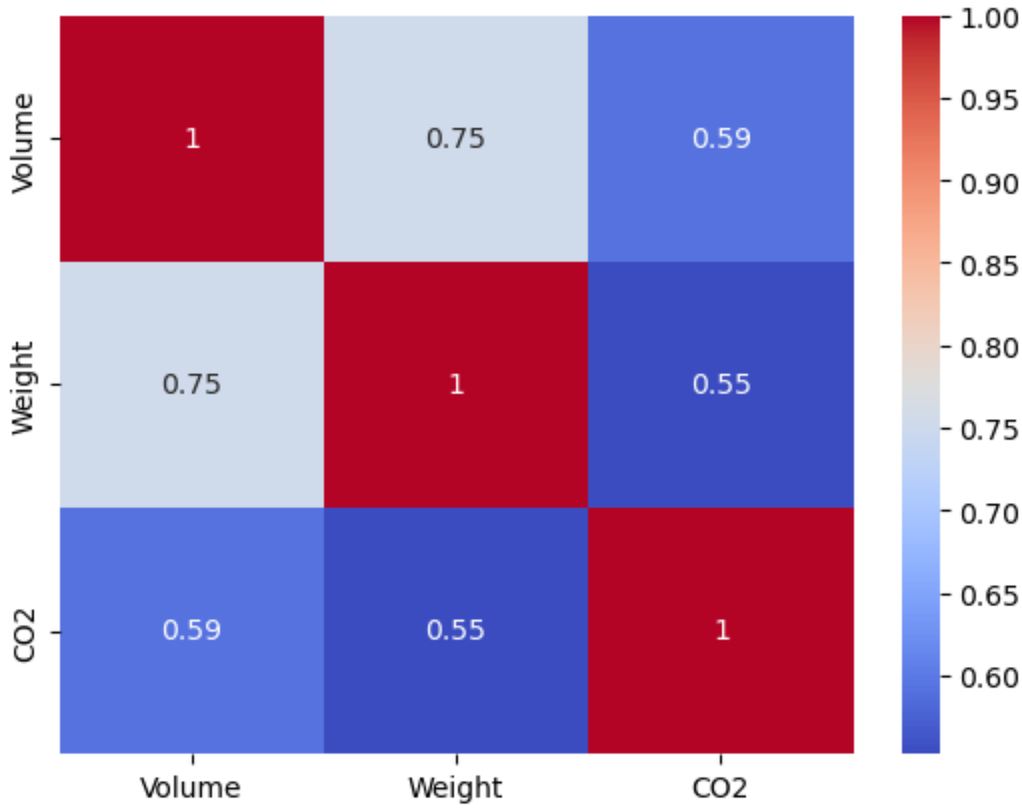
In [63]:
```
# The above scatter plot shows that the volume is directly related to the CO2 emiss
# Corellation Analysis
data.corr(numeric_only=True)
```

Out[63]:

	Volume	Weight	CO2
Volume	1.000000	0.753537	0.592082
Weight	0.753537	1.000000	0.552150
CO2	0.592082	0.552150	1.000000

In [64]:
```
# To visualize the heatmap
sns.heatmap(data.corr(numeric_only=True), annot=True, cmap='coolwarm')
```

Out[64]: <Axes: >



In [69]:
```
# Using ML ideas i.e. linear Regression
from sklearn.linear_model import LinearRegression
car_independent = data[['Weight','Volume']]
car_dependent = data['CO2']
regr = linear_model.LinearRegression()
regr.fit(car_independent,car_dependent)
```

Out[69]:
```
▾ LinearRegression  i ?

LinearRegression()
```

In [77]:
```
new_data= pd.DataFrame([[939,1000],[2200,1280],[1600,1415],[1300,990]], columns=['W
predictedCO2= regr.predict(new_data)
print(predictedCO2)
```

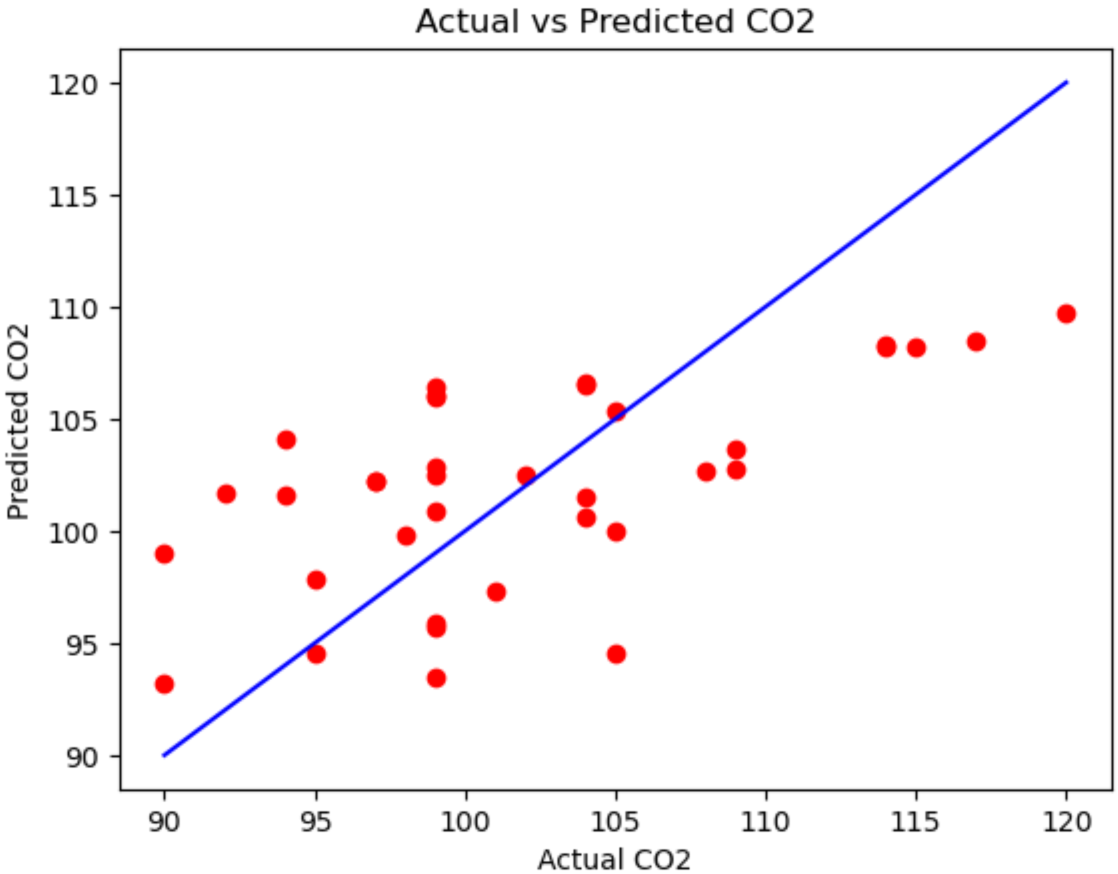
```
[ 94.59031631 106.29753292 102.82067433  97.2381557 ]
```

In [79]:
```
#checking the model accuracy
from sklearn.metrics import r2_score
predicted= regr.predict(car_independent)
print("R2 score:",r2_score(car_dependent,predicted))
```

```
R2 score: 0.37655640436199866
```

In [88]:
```
# To plot the visualization actual vs predicetd co2 values
plt.scatter(car_dependent,predicted, color='red')
plt.xlabel("Actual CO2")
plt.ylabel("Predicted CO2")
plt.title("Actual vs Predicted CO2")
plt.plot([car_dependent.min(),car_dependent.max()],[car_dependent.min(),car_depende
```

```
plt.show()
```


Actual vs Predicted CO2
Predicted CO2
Actual CO2
90
95
100
105
110
115
120

In [ ]: