1. Cars Dataset

```
import pandas as pd
```

```
cars = pd.read_csv("https://raw.githubusercontent.com/AmenaNajeeb/Data/master/Cars.csv")
```

```
cars.head(10)
```

	HP	MPG	VOL	SP	WT
0	49	53.700681	89	104.185353	28.762059
1	55	50.013401	92	105.461264	30.466833
2	55	50.013401	92	105.461264	30.193597
3	70	45.696322	92	113.461264	30.632114
4	53	50.504232	92	104.461264	29.889149
5	70	45.696322	89	113.185353	29.591768
6	55	50.013401	92	105.461264	30.308480
7	62	46.716554	50	102.598513	15.847758
8	62	46.716554	50	102.598513	16.359484
9	80	42.299078	94	115.645204	30.920154

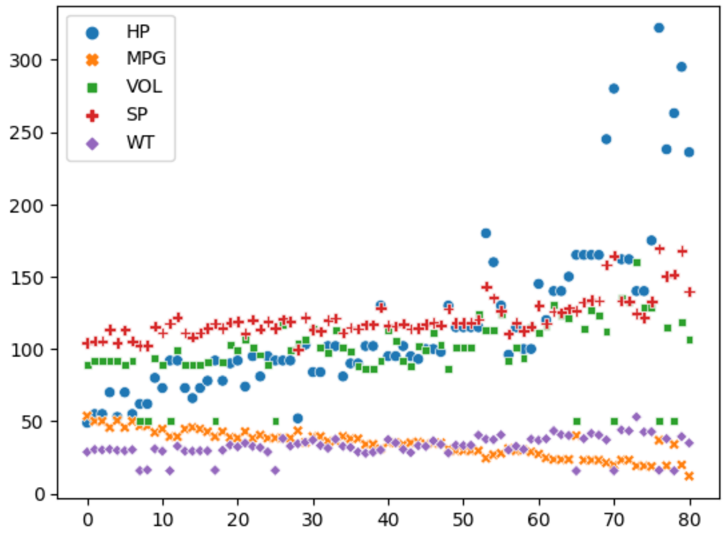
```
cars.shape
```

```
(81, 5)
```

```
import seaborn as sns
```

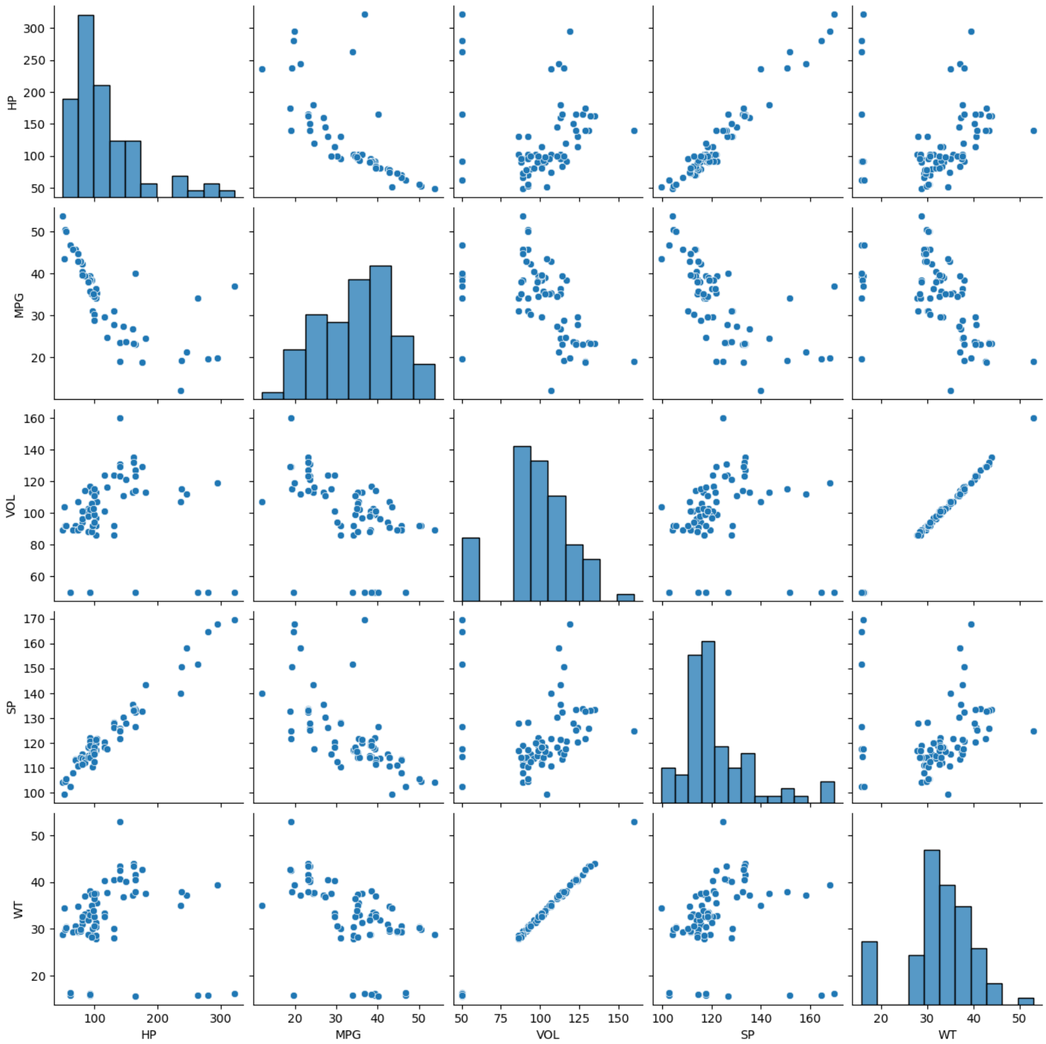
```
sns.scatterplot(cars)
```

```
<Axes: >
```



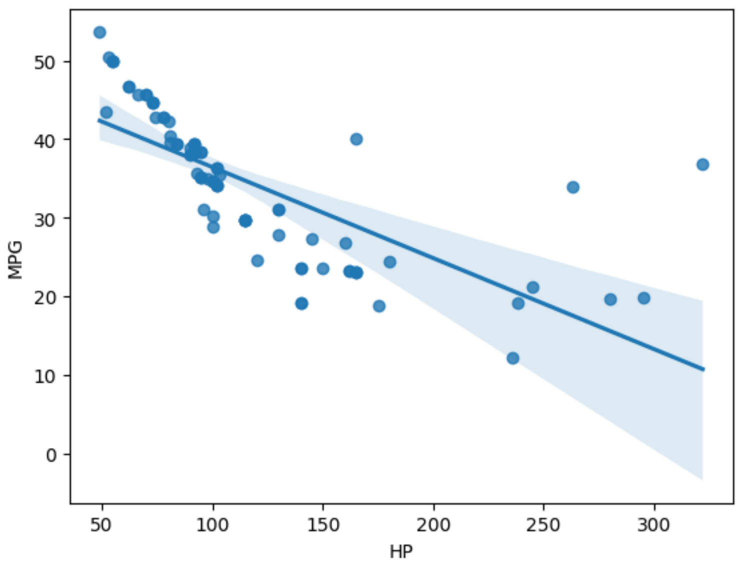

```
sns.pairplot(cars)
```

<seaborn.axisgrid.PairGrid at 0x7f4484066d70>



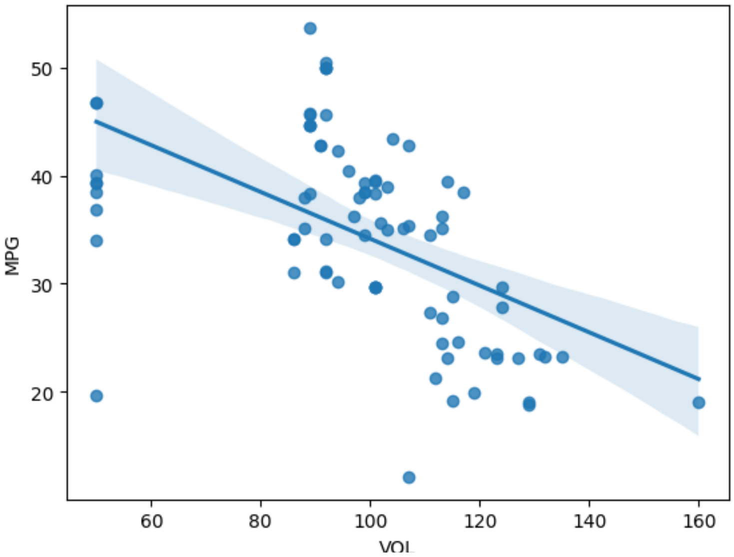

```
sns.regplot(x=cars["HP"],y=cars["MPG"])
```

<Axes: xlabel='HP', ylabel='MPG'>



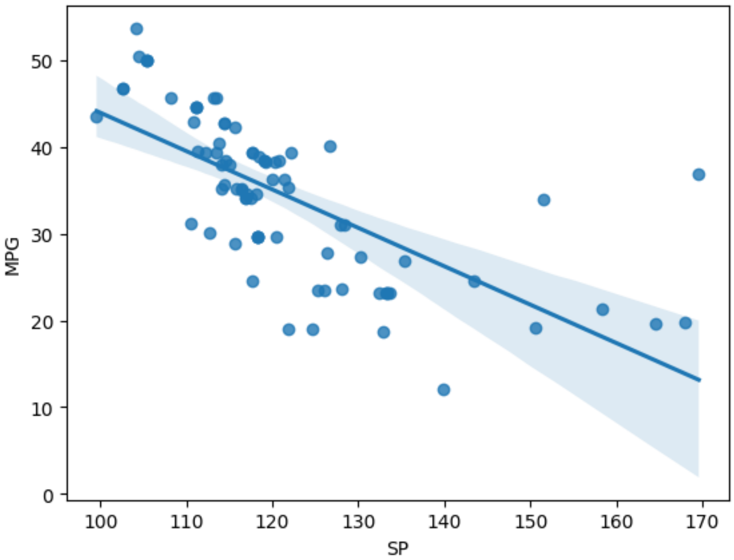

```
sns.regplot(x=cars["VOL"],y=cars["MPG"])
```

```
<Axes: xlabel='VOL', ylabel='MPG'>
```



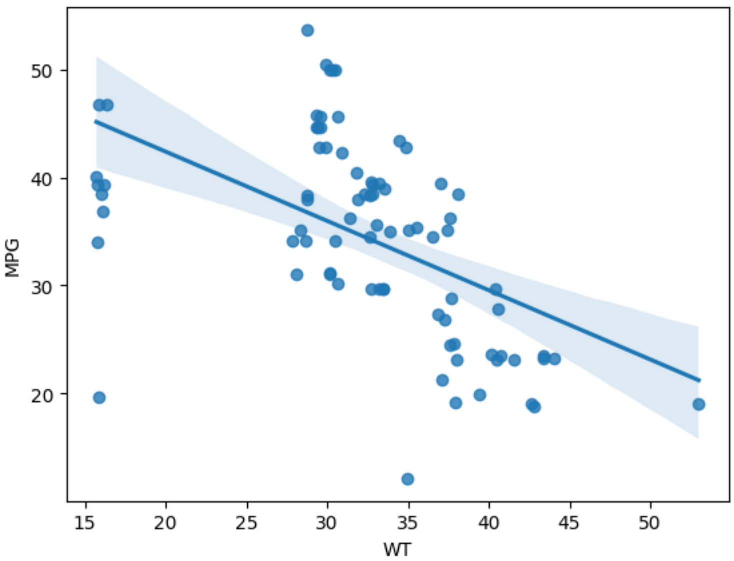

```
sns.regplot(x=cars["SP"],y=cars["MPG"])
```

```
<Axes: xlabel='SP', ylabel='MPG'>
```




```
sns.regplot(x=cars["WT"],y=cars["MPG"])
```

```
<Axes: xlabel='WT', ylabel='MPG'>
```




```
#statsmodel
import statsmodels.formula.api as smf
```

```
model=smf.ols('MPG~HP+VOL+SP+WT',data=cars).fit()
```

```
model.params
```

```
Intercept    30.677336
HP           -0.205444
VOL          -0.336051
SP            0.395627
WT            0.400574
dtype: float64
```

```
x = [[55,92,105,30]]
df2 = pd.DataFrame(x,columns=["HP","VOL","SP","WT"])
df2
```

	HP	VOL	SP	WT
0	55	92	105	30

```
model.predict(df2)
```

```
0    42.019303
dtype: float64
```

```
model.summary()
```

OLS Regression Results

Dep. Variable:	MPG	**R-squared:**	0.771
Model:	OLS	**Adj. R-squared:**	0.758
Method:	Least Squares	**F-statistic:**	63.80
Date:	Thu, 18 May 2023	**Prob (F-statistic):**	1.54e-23
Time:	10:10:06	**Log-Likelihood:**	-233.96
No. Observations:	81	**AIC:**	477.9
Df Residuals:	76	**BIC:**	489.9
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	30.6773	14.900	2.059	0.043	1.001	60.354
HP	-0.2054	0.039	-5.239	0.000	-0.284	-0.127
VOL	-0.3361	0.569	-0.591	0.556	-1.469	0.796
SP	0.3956	0.158	2.500	0.015	0.080	0.711
WT	0.4006	1.693	0.237	0.814	-2.972	3.773

Omnibus:	10.780	**Durbin-Watson:**	1.403
Prob(Omnibus):	0.005	**Jarque-Bera (JB):**	11.722
Skew:	0.707	**Prob(JB):**	0.00285
Kurtosis:	4.215	**Cond. No.**	6.09e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.09e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

```
x=cars[["HP","VOL","SP","WT"]]
df3=pd.DataFrame(x,columns=["HP","VOL","SP","WT"])
data_pred=model.predict(df3)
from sklearn import metrics
mse=metrics.mean_squared_error(cars["MPG"],data_pred)
from math import sqrt
rmse=sqrt(mse)
print(rmse)
```

```
4.347084212704317
```

2. WC_AT Dataset

```
import pandas as pd
```

```
wc_at = pd.read_csv("https://raw.githubusercontent.com/AmenaNajeeb/Data/master/WC_AT.csv")
```

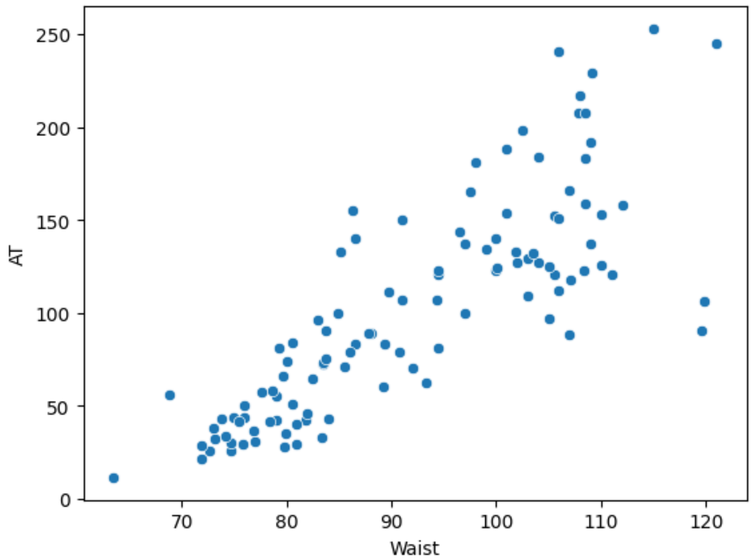
```
wc_at.head(10)
```

	Waist	AT
0	74.75	25.72
1	72.60	25.89
2	81.80	42.60
3	83.95	42.80
4	74.65	29.84
5	71.85	21.68
6	80.90	29.08
7	83.40	32.98
8	63.50	11.44

```
import seaborn as sns
```

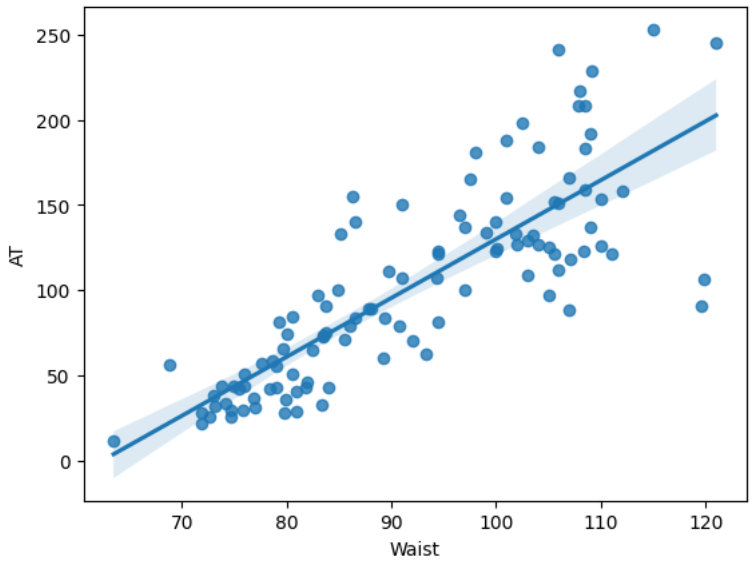
```
sns.scatterplot(x=wc_at["Waist"],y=wc_at["AT"])
```

```
<Axes: xlabel='Waist', ylabel='AT'>
```




```
sns.regplot(x=wc_at["Waist"],y=wc_at["AT"])
```

```
<Axes: xlabel='Waist', ylabel='AT'>
```




```
#statsmodel
import statsmodels.formula.api as smf
```

```
model = smf.ols("AT~Waist",data=wc_at).fit()
```

```
model.params
```

```
Intercept   -215.981488
Waist          3.458859
dtype: float64
```

```
x = [75] #waist
df2 = pd.DataFrame(x,columns=["Waist"])
df2
```

	Waist
0	75

```
model.predict(df2)
```

```
0    43.432966
dtype: float64
```

```
model.summary()
```

OLS Regression Results

Dep. Variable:	AT	**R-squared:**	0.670
Model:	OLS	**Adj. R-squared:**	0.667
Method:	Least Squares	**F-statistic:**	217.3
Date:	Thu, 18 May 2023	**Prob (F-statistic):**	1.62e-27
Time:	10:10:50	**Log-Likelihood:**	-534.99
No. Observations:	109	**AIC:**	1074.
Df Residuals:	107	**BIC:**	1079.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-215.9815	21.796	-9.909	0.000	-259.190	-172.773
Waist	3.4589	0.235	14.740	0.000	2.994	3.924

Omnibus:	3.960	**Durbin-Watson:**	1.560
Prob(Omnibus):	0.138	**Jarque-Bera (JB):**	4.596
Skew:	0.104	**Prob(JB):**	0.100
Kurtosis:	3.984	**Cond. No.**	639.

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

```
x=wc_at["Waist"]
df3=pd.DataFrame(x,columns=["Waist"])
data_pred=model.predict(df3)
from sklearn import metrics
mse=metrics.mean_squared_error(wc_at["AT"],data_pred)
from math import sqrt
rmse=sqrt(mse)
print(rmse)
```

```
32.760177495755144
```

3. Toyota Corolla Dataset

```
import pandas as pd
```

```
df = pd.read_csv("https://raw.githubusercontent.com/AmenaNajeeb/Data/master/Toyoto_Corrola.csv")
```

```
df.head(10)
```

	Id	Model	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	46986	90	3	4	5	1165

```
df.corr()
```

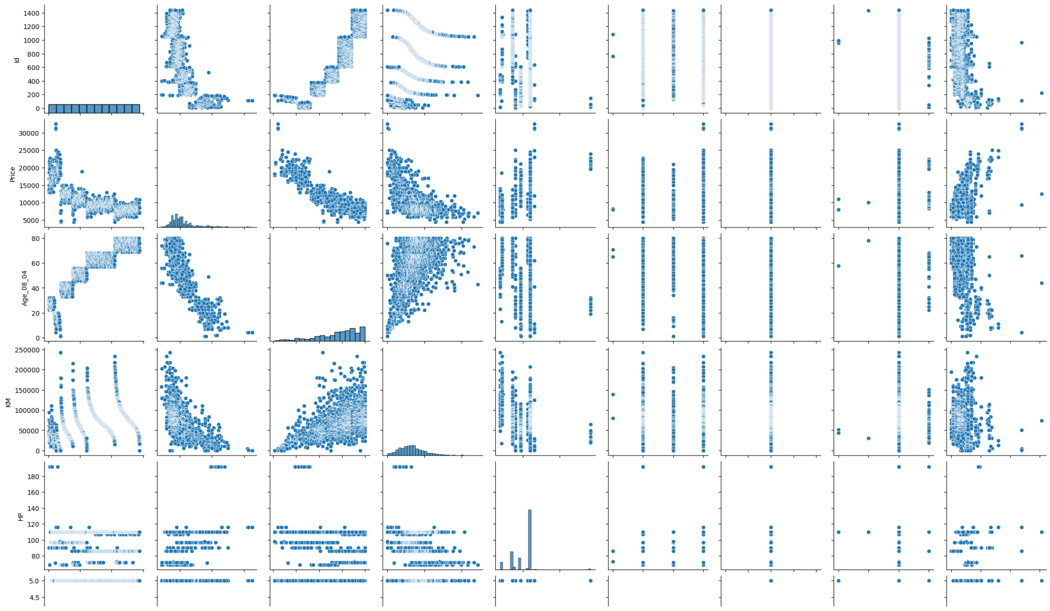
```
<ipython-input-107-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr
  df.corr()
```

	Id	Price	Age_08_04	KM	HP	Doors	Cylinders	Gears	Weight
Id	1.000000	-0.738250	0.906132	0.273298	-0.109375	-0.130207	NaN	-0.043343	-0.414500
Price	-0.738250	1.000000	-0.876590	-0.569960	0.314990	0.185326	NaN	0.063104	0.581198
Age_08_04	0.906132	-0.876590	1.000000	0.505672	-0.156622	-0.148359	NaN	-0.005364	-0.470253
KM	0.273298	-0.569960	0.505672	1.000000	-0.333538	-0.036197	NaN	0.015023	-0.028598
HP	-0.109375	0.314990	-0.156622	-0.333538	1.000000	0.092424	NaN	0.209477	0.089614
Doors	-0.130207	0.185326	-0.148359	-0.036197	0.092424	1.000000	NaN	-0.160141	0.302618
Cylinders	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gears	-0.043343	0.063104	-0.005364	0.015023	0.209477	-0.160141	NaN	1.000000	0.020613
Weight	-0.414500	0.581198	-0.470253	-0.028598	0.089614	0.302618	NaN	0.020613	1.000000

```
import seaborn as sns
```

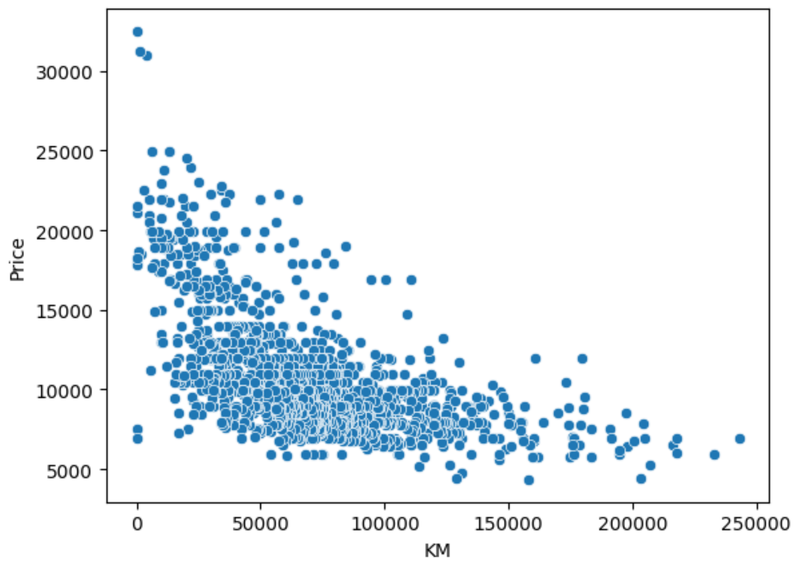
```
sns.pairplot(df)
```

<seaborn.axisgrid.PairGrid at 0x7f4486fd4130>



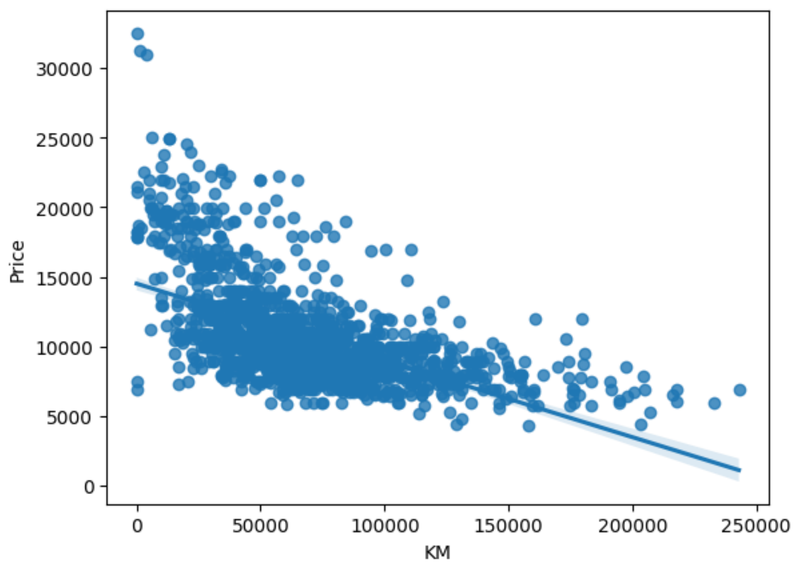
```
sns.scatterplot(x=df["KM"],y=df["Price"])
```

<Axes: xlabel='KM', ylabel='Price'>



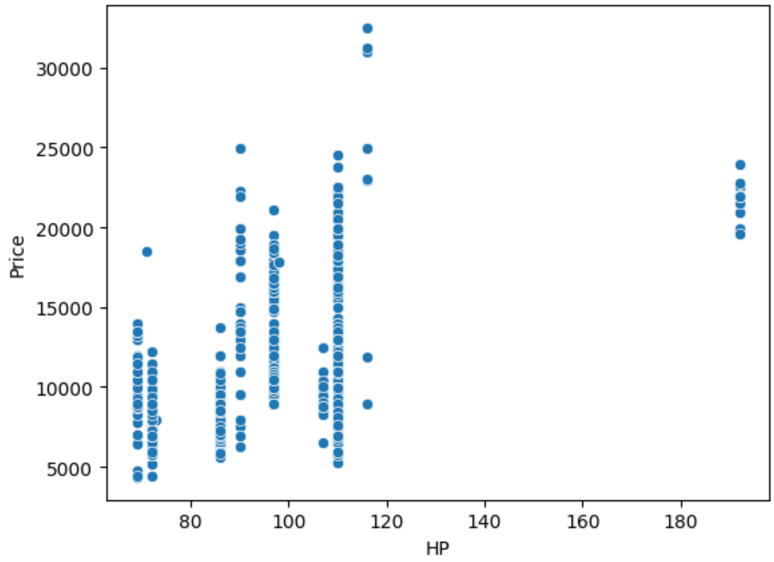
```
sns.regplot(x=df["KM"],y=df["Price"])
```

<Axes: xlabel='KM', ylabel='Price'>



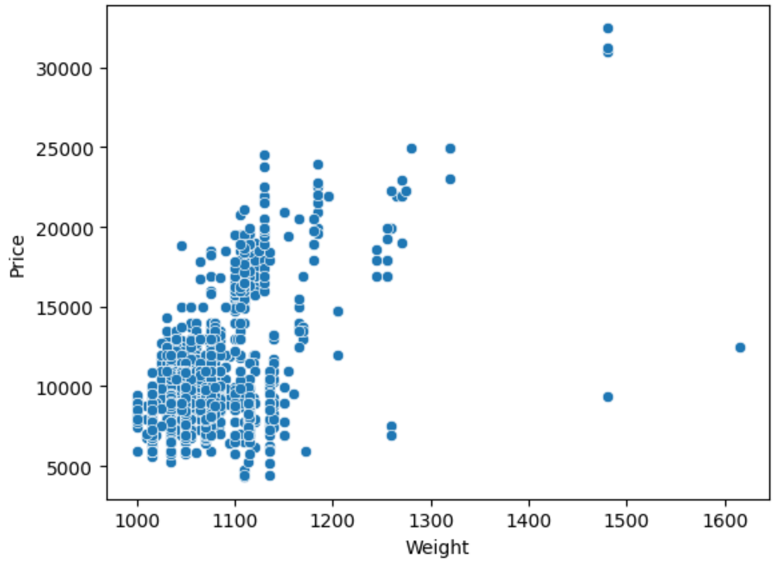
```
sns.scatterplot(x=df["HP"],y=df["Price"])
```

<Axes: xlabel='HP', ylabel='Price'>



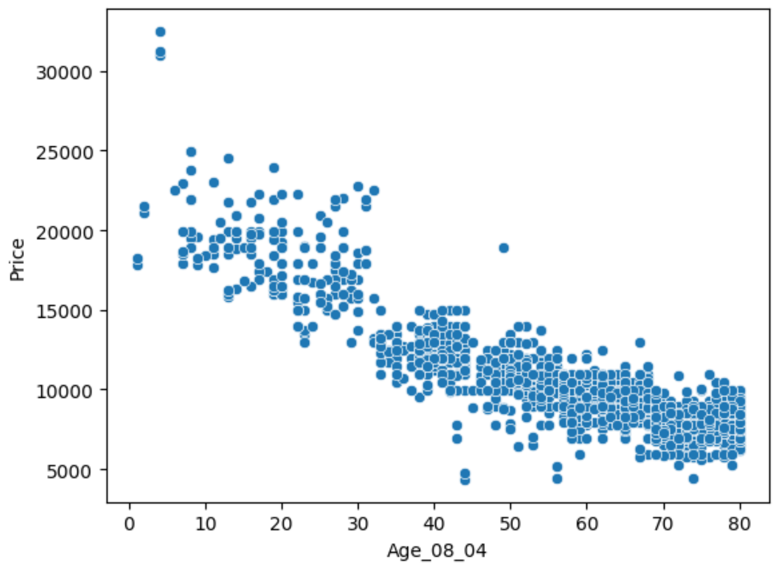

```
sns.scatterplot(x=df["Weight"],y=df["Price"])
```

<Axes: xlabel='Weight', ylabel='Price'>



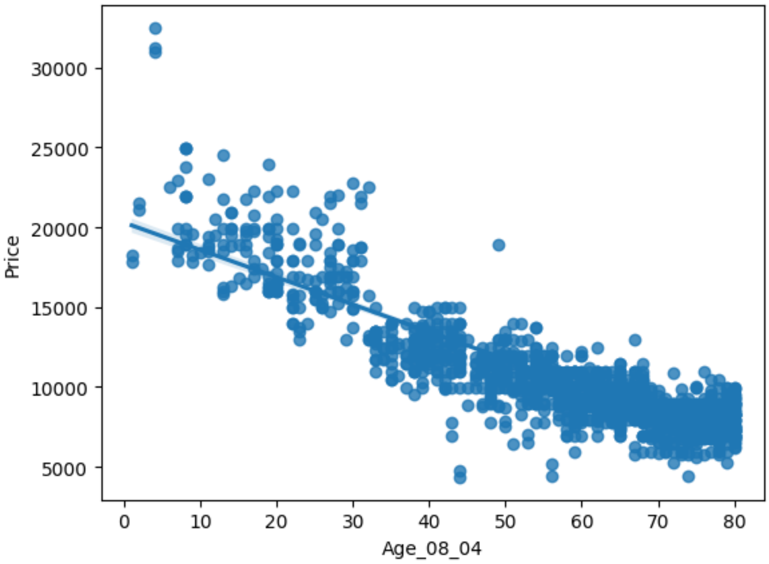

```
sns.scatterplot(x=df["Age_08_04"],y=df["Price"])
```

<Axes: xlabel='Age_08_04', ylabel='Price'>




```
sns.regplot(x=df["Age_08_04"],y=df["Price"])
```

<Axes: xlabel='Age_08_04', ylabel='Price'>




```
#statsmodel
import statsmodels.formula.api as smf
```

```
model=smf.ols('Price~Age_08_04+KM+HP+Weight',data=df).fit()
```

```
model.params
```

```
Intercept   -4014.641772
Age_08_04    -122.424469
KM             -0.019647
HP             30.211927
Weight         18.531868
dtype: float64
```

```
x = [[24,41711,90,1165]]
df2 = pd.DataFrame(x,columns=["Age_08_04","KM","HP","Weight"])
df2
```

	Age_08_04	KM	HP	Weight
0	24	41711	90	1165

```
model.predict(df2)
```

```
0    16536.360171
dtype: float64
```

```
model.summary()
```

OLS Regression Results

Dep. Variable:	Price	**R-squared:**	0.862

```
x=df[["Age_08_04","KM","HP","Weight"]]
df3=pd.DataFrame(x,columns=["Age_08_04","KM","HP","Weight"])
data_pred=model.predict(df3)
from sklearn import metrics
mse=metrics.mean_squared_error(df["Price"],data_pred)
from math import sqrt
rmse=sqrt(mse)
print(rmse)
```

```
1347.9816495722343
```

Intercept -4014.0418 936.044 -4.289 0.000 -5850.808 -2178.476

```
# Results of three models:

# R-squared value of cars model: 0.771
# R-squared value of waist-circumference model: 0.670
# R-squared value of toyota-corolla model: 0.862
```

Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1498.712
Skew:	-0.384	**Prob(JB):**	0.00
Kurtosis:	7.946	**Cond. No.**	2.05e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.05e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

✓ 0s completed at 3:51 PM