In [1]:
```python
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import skew, kurtosis
```

In [3]:
```python
# Load data
dataset = "https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data"
columns=["mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration", "model_year", "orig
df = pd.read_csv(dataset, delim_whitespace=True, names=columns)
```

In [5]:
```python
# Display the first 20 rows
df.head(20)
```

Out[5]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	car_name
0	18.0	8	307.0	130.0	3504.0	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693.0	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436.0	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433.0	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449.0	10.5	70	1	ford torino
5	15.0	8	429.0	198.0	4341.0	10.0	70	1	ford galaxie 500
6	14.0	8	454.0	220.0	4354.0	9.0	70	1	chevrolet impala
7	14.0	8	440.0	215.0	4312.0	8.5	70	1	plymouth fury iii
8	14.0	8	455.0	225.0	4425.0	10.0	70	1	pontiac catalina
9	15.0	8	390.0	190.0	3850.0	8.5	70	1	amc ambassador dpl
10	15.0	8	383.0	170.0	3563.0	10.0	70	1	dodge challenger se
11	14.0	8	340.0	160.0	3609.0	8.0	70	1	plymouth 'cuda 340
12	15.0	8	400.0	150.0	3761.0	9.5	70	1	chevrolet monte carlo
13	14.0	8	455.0	225.0	3086.0	10.0	70	1	buick estate wagon (sw)
14	24.0	4	113.0	95.00	2372.0	15.0	70	3	toyota corona mark ii
15	22.0	6	198.0	95.00	2833.0	15.5	70	1	plymouth duster
16	18.0	6	199.0	97.00	2774.0	15.5	70	1	amc hornet
17	21.0	6	200.0	85.00	2587.0	16.0	70	1	ford maverick
18	27.0	4	97.0	88.00	2130.0	14.5	70	3	datsun pl510
19	26.0	4	97.0	46.00	1835.0	20.5	70	2	volkswagen 1131 deluxe sedan

In [17]:
```python
# Check for missing values
df.replace("?",np.nan,inplace=True)

# Convert horsepower to numeric after handling '?'
df["horsepower"]=pd.to_numeric(df['horsepower'])  # Convert horsepower to numeric after handling '?'

missing_values = df.isnull().sum()
print("Missing Values:\n", missing_values)
```

```
Missing Values:
 mpg             0
cylinders       0
displacement    0
horsepower      6
weight          0
acceleration    0
model_year      0
origin          0
car_name        0
dtype: int64
```

In [19]:
```
# Basic description of data
df.describe()
```

Out[19]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
count	398.000000	398.000000	398.000000	392.000000	398.000000	398.000000	398.000000	398.000000
mean	23.514573	5.454774	193.425879	104.469388	2970.424623	15.568090	76.010050	1.572864
std	7.815984	1.701004	104.269838	38.491160	846.841774	2.757689	3.697627	0.802055
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	70.000000	1.000000
25%	17.500000	4.000000	104.250000	75.000000	2223.750000	13.825000	73.000000	1.000000
50%	23.000000	4.000000	148.500000	93.500000	2803.500000	15.500000	76.000000	1.000000
75%	29.000000	8.000000	262.000000	126.000000	3608.000000	17.175000	79.000000	2.000000
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000	82.000000	3.000000

In [21]:
```
# Calculate skewness and curtosis
numeric_col=df.select_dtypes(include=[np.number]).columns
skewness=df[numeric_col].skew()
kurtosis=df[numeric_col].kurtosis()
```

In [23]:
```
print(skewness)
```

```
mpg             0.457066
cylinders       0.526922
displacement    0.719645
horsepower      1.087326
weight          0.531063
acceleration    0.278777
model_year      0.011535
origin          0.923776
dtype: float64
```

In [25]:
```
print(kurtosis)
```

```
mpg            -0.510781
cylinders      -1.376662
displacement   -0.746597
horsepower      0.696947
weight         -0.785529
acceleration    0.419497
model_year     -1.181232
origin         -0.817597
dtype: float64
```

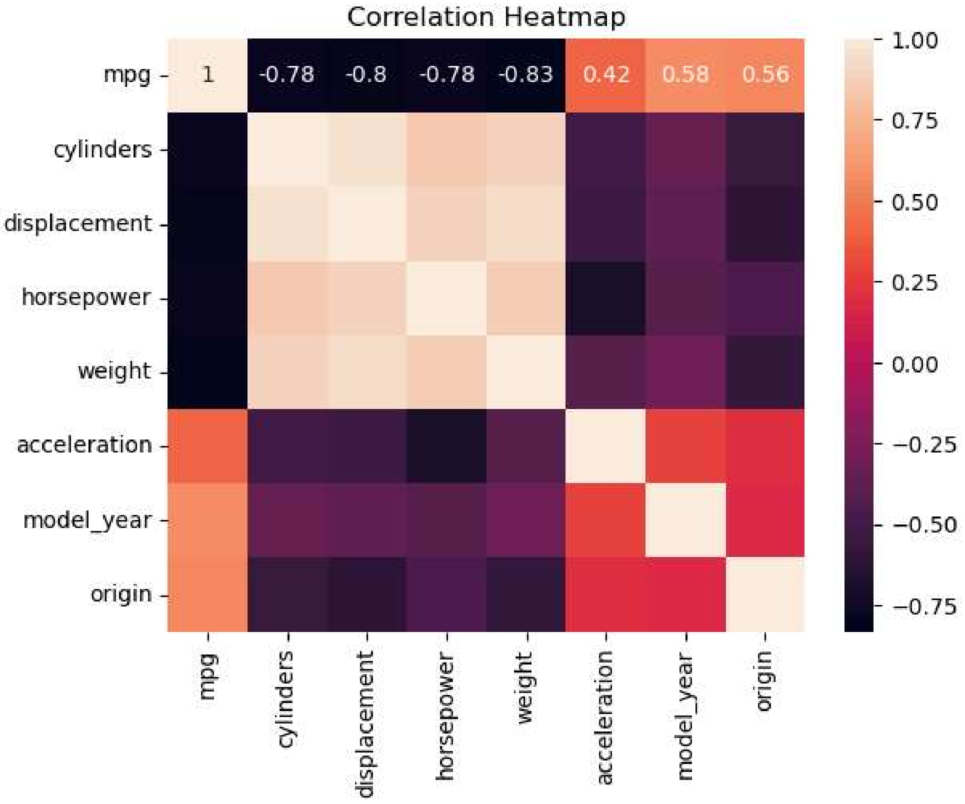
In [29]:
```
# Correlation of variables
correlation=df[numeric_col].corr()
correlation
```

Out[29]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
mpg	1.000000	-0.775396	-0.804203	-0.778427	-0.831741	0.420289	0.579267	0.563450
cylinders	-0.775396	1.000000	0.950721	0.842983	0.896017	-0.505419	-0.348746	-0.562543
displacement	-0.804203	0.950721	1.000000	0.897257	0.932824	-0.543684	-0.370164	-0.609409
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.689196	-0.416361	-0.455171
weight	-0.831741	0.896017	0.932824	0.864538	1.000000	-0.417457	-0.306564	-0.581024
acceleration	0.420289	-0.505419	-0.543684	-0.689196	-0.417457	1.000000	0.288137	0.205873
model_year	0.579267	-0.348746	-0.370164	-0.416361	-0.306564	0.288137	1.000000	0.180662
origin	0.563450	-0.562543	-0.609409	-0.455171	-0.581024	0.205873	0.180662	1.000000

In [31]:

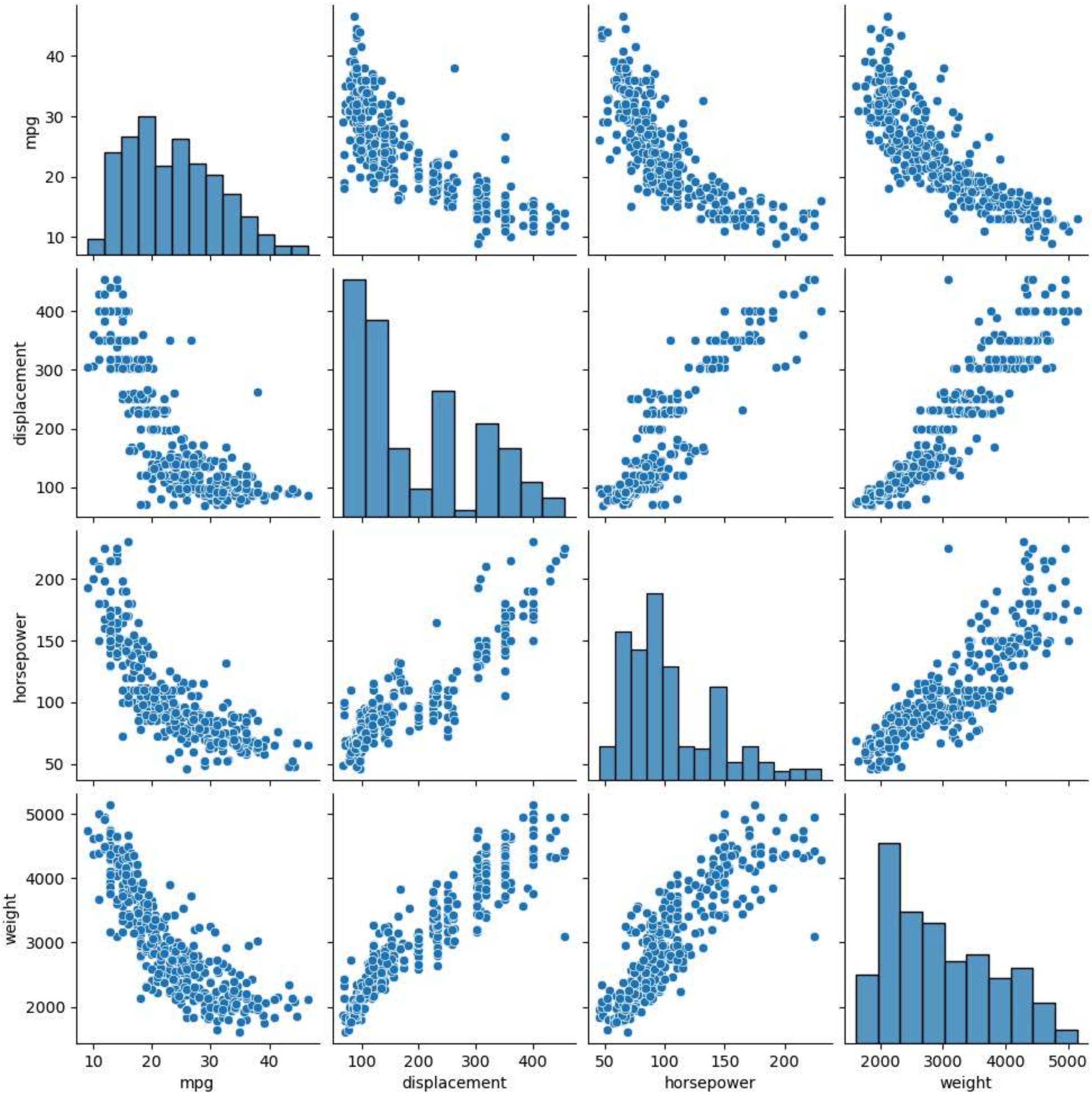
```
# Correlation Heat map
sns.heatmap(correlation, annot=True)
plt.title("Correlation Heatmap")
plt.show()
```



In [33]:

```
# Scatter plot for different parameters
sns.pairplot(df, vars=["mpg","displacement", "horsepower","weight"])
plt.show()
```

```
C:\Users\User\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option i
s deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\User\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option i
s deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\User\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option i
s deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\User\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option i
s deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



mpg
displacement
horsepower
weight
10
20
30
40
100
200
300
400
50
100
150
200
2000
3000
4000
5000