

```
import seaborn as sns
import pandas as pd

# Include Any Additional Packages you need to import

# Load the "mpg" dataset from seaborn's GitHub repository
df = sns.load_dataset('mpg')

# Alternatively, you can load the "mpg" dataset directly from the UCI Machine Learning Repository:
# df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data', delim_whitespace=

# Print the first few rows of the dataset to verify that it's loaded correctly
df.head()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino

```
df.info()

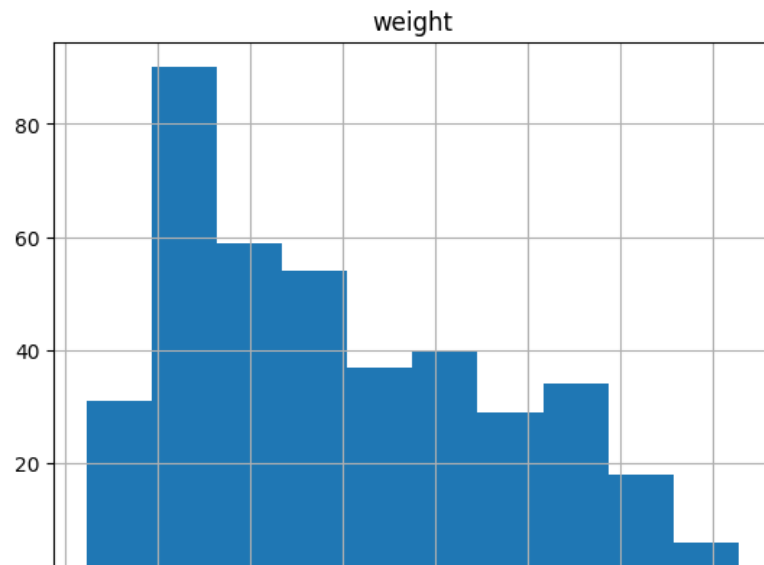
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0    mpg             398 non-null    float64
1    cylinders        398 non-null    int64
2    displacement     398 non-null    float64
3    horsepower       392 non-null    float64
4    weight           398 non-null    int64
5    acceleration     398 non-null    float64
6    model_year       398 non-null    int64
7    origin           398 non-null    object
8    name            398 non-null    object
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB
```

```
df.describe()
```

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

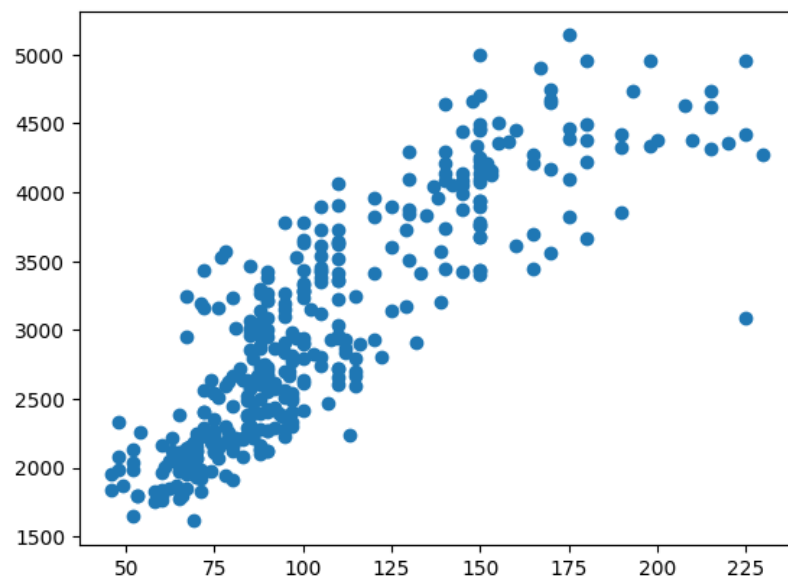
```
import matplotlib.pyplot as plt
df.hist("weight")
```

```
array([[<Axes: title={'center': 'weight'}>]], dtype=object)
```

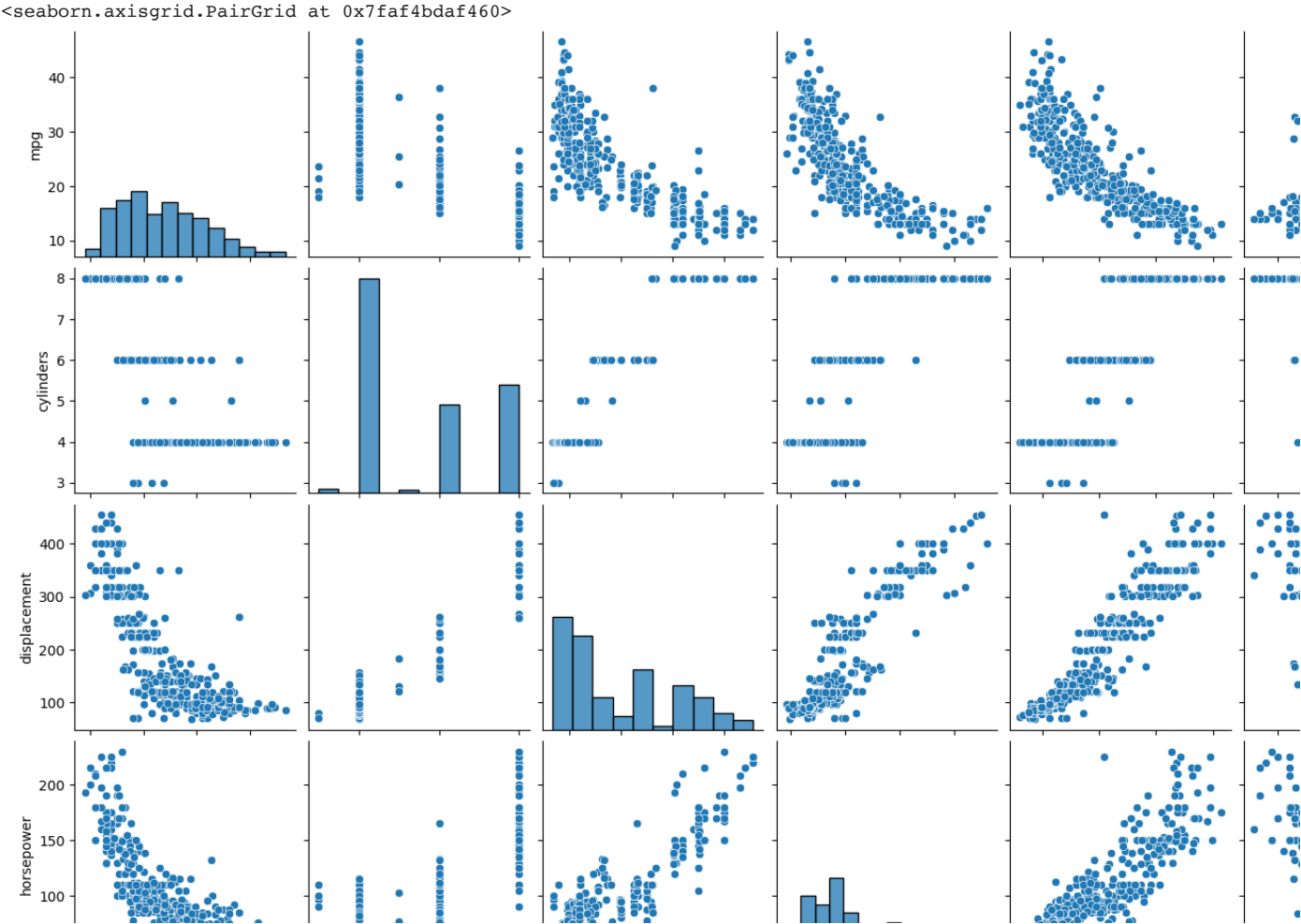


```
plt.scatter(x=df.horsepower,y=df.weight)
```

```
<matplotlib.collections.PathCollection at 0x7faf4bef38b0>
```



```
import seaborn as sns
sns.pairplot(df)
```



Based on the calculations done, mpg and weight are the best candidates for the linear regression.

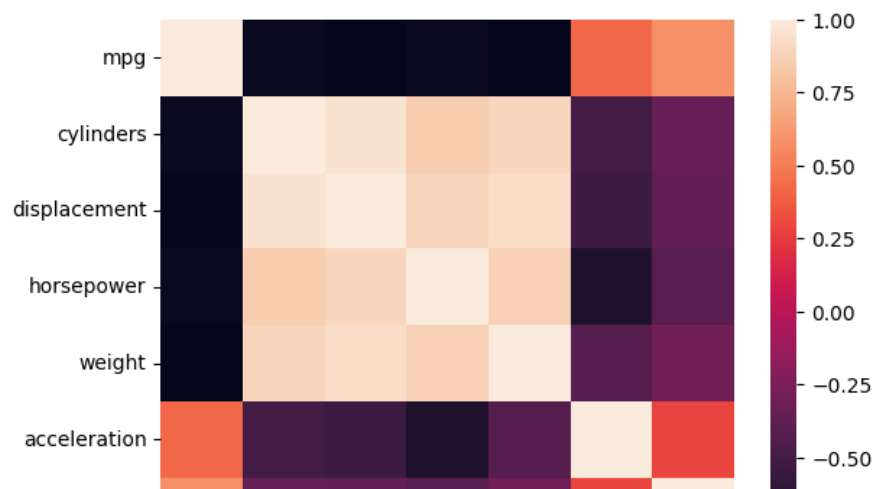
```
df.corr()
```

<ipython-input-17-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year
mpg	1.000000	-0.775396	-0.804203	-0.778427	-0.831741	0.420289	0.579267
cylinders	-0.775396	1.000000	0.950721	0.842983	0.896017	-0.505419	-0.348746
displacement	-0.804203	0.950721	1.000000	0.897257	0.932824	-0.543684	-0.370164
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.689196	-0.416361
weight	-0.831741	0.896017	0.932824	0.864538	1.000000	-0.417457	-0.306564
acceleration	0.420289	-0.505419	-0.543684	-0.689196	-0.417457	1.000000	0.288137
model_year	0.579267	-0.348746	-0.370164	-0.416361	-0.306564	0.288137	1.000000

```
sns.heatmap(df.corr())
```

```
<ipython-input-18-aa4f4450a243>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated
sns.heatmap(df.corr())
<Axes: >
```



There are different linear relations between MPG and weight, for example and also some other variables like acceleration and mpg, acceleration and model_year.

```

n      dk      ne      w      fig      iti      ye
X=df.weight
y=df.mpg

c

from sklearn.linear_model import LinearRegression
```

```
model=LinearRegression()
```

```
import statsmodels.api as sm
model=sm.OLS(y,X).fit()
model.predict(X)
```

```

0      23.641070
1      24.916231
2      23.182282
3      23.162042
4      23.269992
...
393    18.823797
394    14.370856
395    15.484091
396    17.710562
397    18.351516
Length: 398, dtype: float64
```

✓ 0 s completado a las 8:05

×