# Get all imports

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

# Use Pandas to read data and separate into input features and target

### **Data Description**: We are going to predict Miles per Gallon (MPG) consumed by cars based on the following features

1. Cylinders - (Real) Number of cylinders used in the car's engine
2. Displacement - (Real) A measure of the cylinder volume swept by all of the pistons
3. Horsepower - (Real) The power of the engine
4. Weight - (Real) The weight of the entire car
5. Acceleration - (Real) The acceleration of the car
6. Mode Year - (Real) The year that the vehicle was built

In [ ]:

```
df = pd.read_csv("auto-mpg.data", delimiter=",")
# Drop the origin and car name columns
df = df.drop(["origin", "car name"], axis=1)
display(df.head())

X, Y = df.drop(["mpg"], axis=1).values, df["mpg"].values
```

| | mpg | cylinders | displacement | horsepower | weight | acceleration | model year |
|---|---|---|---|---|---|---|---|
| **0** | 18.0 | 8 | 307.0 | 130.0 | 3504 | 12.0 | 70 |
| **1** | 15.0 | 8 | 350.0 | 165.0 | 3693 | 11.5 | 70 |
| **2** | 18.0 | 8 | 318.0 | 150.0 | 3436 | 11.0 | 70 |
| **3** | 16.0 | 8 | 304.0 | 150.0 | 3433 | 12.0 | 70 |
| **4** | 17.0 | 8 | 302.0 | 140.0 | 3449 | 10.5 | 70 |

# Define the linear regression model using Pseudo Inverse.

In [ ]:

```
class LinearRegression():
    # Constructor of this class
    def __init__(self):
        self.coeff = list()

    # This function is used to find the coefficients for the line of best fit
    def fit(self, A, Y):
        # Add Bias
        A = np.concatenate((np.ones((len(A), 1)), A), axis=1)
        # Find Pseudo Inverse
        pseudo_inv = np.matmul(np.linalg.inv(np.matmul(np.transpose(A), A)), np.transpos
        # Finally get the coefficients
        self.coeff = np.matmul(pseudo_inv, np.reshape(Y, (-1, 1)))

    # This function uses the found coefficients to deliver predictions
    def predict(self, A):
        A = np.concatenate((np.ones((len(A), 1)), A), axis=1)
        print("Model Coefficients:", self.coeff)
        return np.squeeze(np.matmul(A, self.coeff))
```

# Applying Linear Regression to Predict Miles Per Gallon

In [ ]:

```
model = LinearRegression()
model.fit(X, Y)
preds = model.predict(X)
print("Mean Squared Error:", (np.sum((Y-preds)**2))/len(Y))
```

```
Model Coefficients: [[-1.45352505e+01]
 [-3.29859089e-01]
 [ 7.67843024e-03]
 [-3.91355574e-04]
 [-6.79461791e-03]
 [ 8.52732469e-02]
 [ 7.53367180e-01]]
Mean Squared Error: 11.590170981415225
```

In [ ]: