Coding exercises for Lecture 3: Mean Squared Error

CMSE 381 - Spring 2024

This notebook has some code to go along with Lecture 2 on Mean Squared Error.

In [1]:
```
# As always, we start with our favorite standard imports.

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

Info about the data set

From https://rdrr.io/cran/ISLR/man/Auto.html

Auto: Auto Data Set

Description

Gas mileage, horsepower, and other information for 392 vehicles. Usage

Format

A data frame with 392 observations on the following 9 variables.

- `mpg` : miles per gallon
- `cylinders` : Number of cylinders between 4 and 8
- `displacement` : Engine displacement (cu. inches)
- `horsepower` : Engine horsepower
- `weight` : Vehicle weight (lbs.)
- `acceleration` : Time to accelerate from 0 to 60 mph (sec.)
- `year` : Model year (modulo 100)
- `origin` : Origin of car (1. American, 2. European, 3. Japanese)
- `name` : Vehicle name

The orginal data contained 408 observations but 16 observations with missing values were removed.

Source

This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the 1983 American Statistical Association Exposition.

In [2]:
```
# First, we're going to do all the data loading and cleanup we figured out last time
auto = pd.read_csv('../../DataSets/Auto.csv')
```

```
auto = auto.replace('?', np.nan)
auto = auto.dropna()
auto.horsepower = auto.horsepower.astype('int')
auto.shape
```

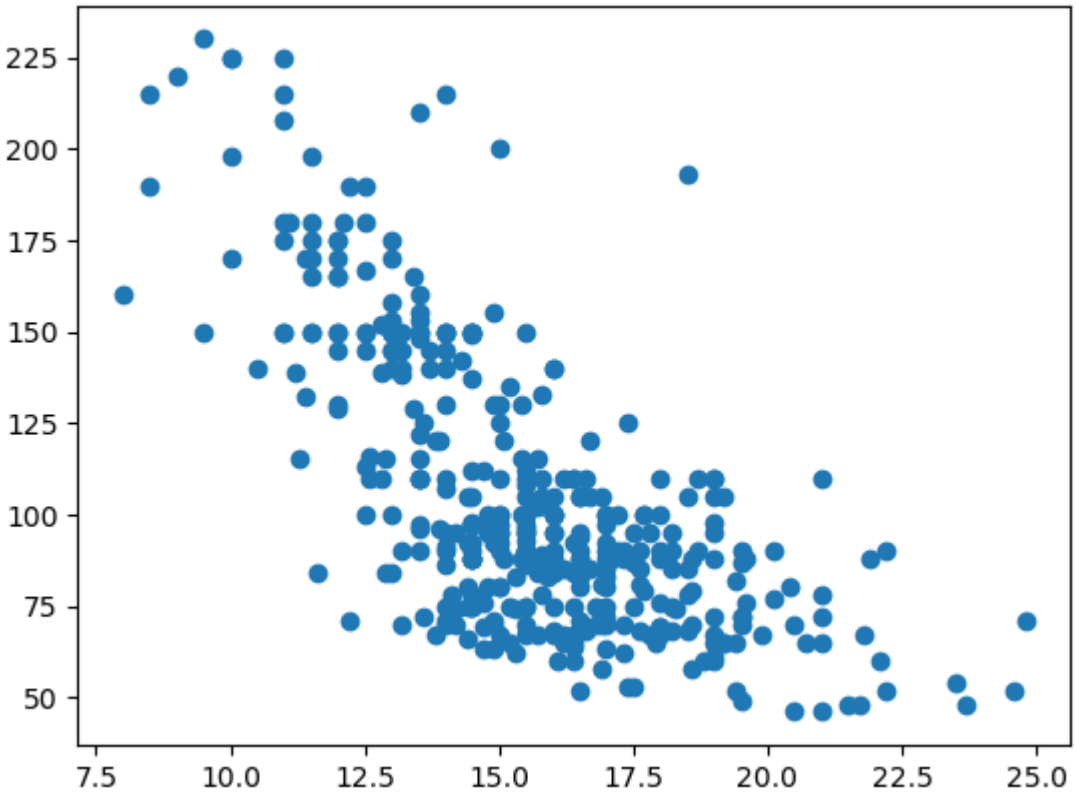
Out[2]:
```
(392, 9)
```

I want to just predict acceleration using horsepower.

✅ **Do this:** Make a scatter plot of acceleration (the output varible) vs horsepower (the input variable). Does it look like there's a relationship between the two variables?

In [3]:
```
# Your code here.

plt.scatter(auto['acceleration'], auto['horsepower'])
plt.show()
```



I've decided to use the model

$$\hat{f}\left(\texttt{horsepower}\right) = 23 - 0.05 \cdot \texttt{horsepower}$$

✅ **Do this:** Make a panda Series with entries $\hat{f}\left(\texttt{horsepower}\right)$ for each entry in `auto.horsepower`.

In [5]:
```
horsepower_data = auto.horsepower

def f(horsepower):
    return 23 - 0.05 * horsepower

f_horsepower_series = pd.Series([f(hp) for hp in horsepower_data])
f_horsepower_series
```

Out[5]:
```
0      16.50
1      14.75
2      15.50
3      15.50
4      16.00
       ...
387    18.70
388    20.40
389    18.80
390    19.05
391    18.90
Length: 392, dtype: float64
```

✅ **Do this:** Using the series you just built, calculated the mean squared error,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

In [6]:
```
# Your code here

mse = ((auto['mpg'] - f_horsepower_series) ** 2).mean()
mse
```

Out[6]:
```
77.31824289405684
```

Have some spare time? Can you mess around with the coefficients in your model to decrease the MSE?

In [ ]:
```
# Your code here
```

Congratulations, we're done!

Written by Dr. Liz Munch, Michigan State University 

In [ ]:
```

```