Question #1 a):

Create a linear regression object called "lm1".

[17]:
```
# Write your code below and press Shift+Enter to execute

lm1 = LinearRegression()
lm1
```

[17]:
```
LinearRegression()
```

Question #1 b):

Train the model using "engine-size" as the independent variable and "price" as the dependent variable?

[18]:
```
# Write your code below and press Shift+Enter to execute

lm1.fit(df[['engine-size']], df[['price']])
lm1
```

[18]:
```
LinearRegression()
```

Question #1 c):

Find the slope and intercept of the model.

Slope

[19]:
```
# Write your code below and press Shift+Enter to execute

lm1.coef_
```

[19]:
```
array([[166.86001569]])
```

▶ Click here for the solution

Intercept

[21]:
```
# Write your code below and press Shift+Enter to execute

lm1.intercept_
```

[21]:
```
array([-7963.33890628])
```

Question #1 d):

What is the equation of the predicted line? You can use x and yhat or "engine-size" or "price".

[22]:
```
# Write your code below and press Shift+Enter to execute
# using X and Y

Yhat=-7963.34 + 166.86*X

Price=-7963.34 + 166.86*df['engine-size']
```

Question #2 a): ¶

Create and train a Multiple Linear Regression model "lm2" where the response variable is "price", and the predictor variable is "normalized-losses" and "highway-mpg".

[28]:
```
# Write your code below and press Shift+Enter to execute

lm2 = LinearRegression()
lm2.fit(df[['normalized-losses','highway-mpg']],df['price'])
```

[28]:
```
LinearRegression()
```

▶ Click here for the solution

Question #2 b):

Find the coefficient of the model.

[29]:
```
# Write your code below and press Shift+Enter to execute

lm2.coef_
```

[29]:
```
array([   1.49789586, -820.45434016])
```

Question #3: ¶

Given the regression plots above, is "peak-rpm" or "highway-mpg" more strongly correlated with "price"? Use the method ".corr()" to verify your answer.

[33]:
```
# Write your code below and press Shift+Enter to execute

df[["peak-rpm","highway-mpg","price"]].corr()
```

[33]:

	peak-rpm	highway-mpg	price
peak-rpm	1.000000	-0.058598	-0.101616
highway-mpg	-0.058598	1.000000	-0.704692
price	-0.101616	-0.704692	1.000000

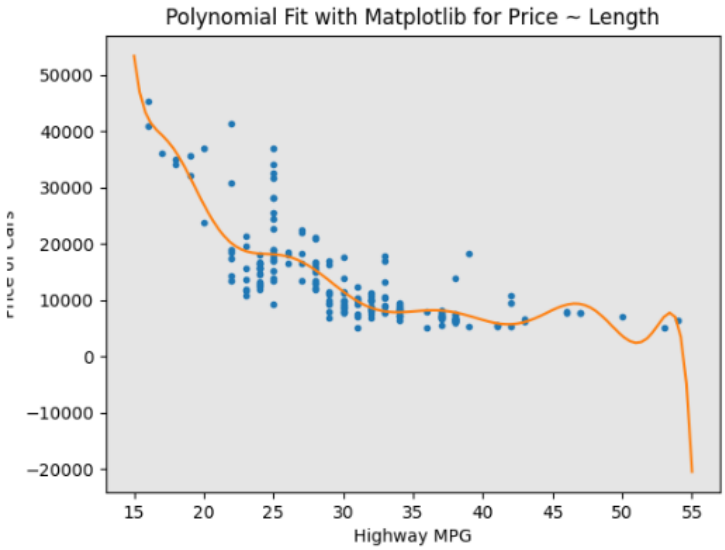
Question #4:

Create 11 order polynomial model with the variables x and y from above.

[42]:
```
# Write your code below and press Shift+Enter to execute

f1 = np.polyfit(x,y,11)
p1 = np.poly1d(f1)
print(p1)
PlotPolly(p1,x,y,'Highway MPG')
```

```
            11             10             9           8         7
-1.243e-08 x  + 4.722e-06 x  - 0.0008028 x + 0.08056 x - 5.297 x
          6        5             4             3             2
 + 239.5 x - 7588 x + 1.684e+05 x - 2.565e+06 x + 2.551e+07 x - 1.491e+08 x + 3.879e+08
```



Question #5:

Create a pipeline that standardizes the data, then produce a prediction using a linear regression model using the features Z and target y.

[53]:
```
# Write your code below and press Shift+Enter to execute

Input=[('scale',StandardScaler()),('model',LinearRegression())]

pipe=Pipeline(Input)

pipe.fit(Z,y)

ypipe=pipe.predict(Z)
ypipe[0:10]
```

[53]:
```
array([13699.11161184, 13699.11161184, 19051.65470233, 10620.36193015,
       15521.31420211, 13869.66673213, 15456.16196732, 15974.00907672,
       17612.35917161, 10722.32509097])
```