Module Six Discussion: Multiple Regression

This notebook contains the step-by-step directions for your Module Six discussion. It is very important to run through the steps in order. Some steps depend on the outputs of earlier steps. Once you have completed the steps in this notebook, be sure to answer the questions about this activity in the discussion for this module.

Reminder: If you have not already reviewed the discussion prompt, please do so before beginning this activity. That will give you an idea of the questions you will need to answer with the outputs of this script.

Initial post (due Thursday)

Step 1: Generating cars dataset

This block of Python code will generate the sample data for you. You will not be generating the data set using numpy module this week. Instead, the data set will be imported from a CSV file. To make the data unique to you, a random sample of size 30, without replacement, will be drawn from the data in the CSV file. The data set will be saved in a Python dataframe that will be used in later calculations.

Click the block of code below and hit the **Run** button above.

In [1]:
```
import pandas as pd
from IPython.display import display, HTML

# read data from mtcars.csv data set.
cars_df_orig = pd.read_csv("https://s3-us-west-2.amazonaws.com/data-analytics.
zybooks.com/mtcars.csv")

# randomly pick 30 observations from the data set to make the data set unique
to you.
cars_df = cars_df_orig.sample(n=30, replace=False)

# print only the first five observations in the dataset.
print("Cars data frame (showing only the first five observations)\n")
display(HTML(cars_df.head().to_html()))
```

```
Cars data frame (showing only the first five observations)
```

	Unnamed: 0	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
11	Merc 450SE	16.4	8	275.8	180	3.07	4.07	17.40	0	0	3	3
26	Porsche 914-2	26.0	4	120.3	91	4.43	2.14	16.70	0	1	5	2
8	Merc 230	22.8	4	140.8	95	3.92	3.15	22.90	1	0	4	2
30	Maserati Bora	15.0	8	301.0	335	3.54	3.57	14.60	0	1	5	8
21	Dodge Challenger	15.5	8	318.0	150	2.76	3.52	16.87	0	0	3	2

Step 2: Scatterplot of miles per gallon against weight

The block of code below will create a scatterplot of the variables "miles per gallon" (coded as mpg in the data set) and "weight" of the car (coded as wt).

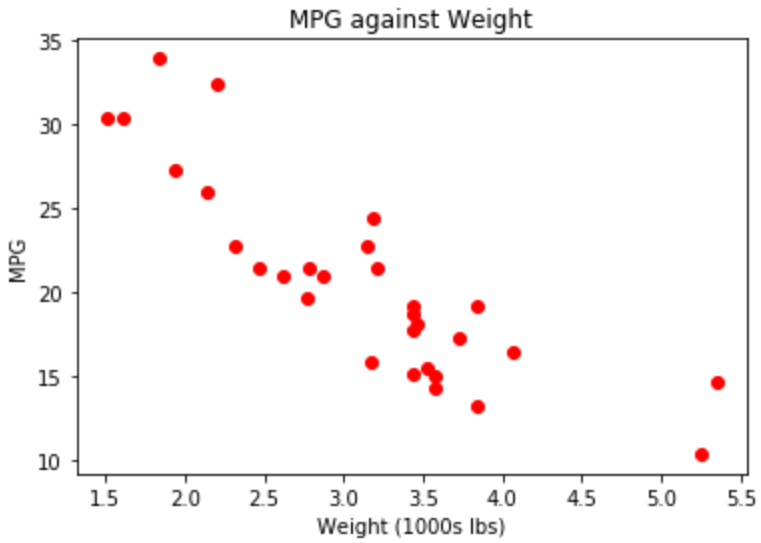
Click the block of code below and hit the **Run** button above.
NOTE: If the plot is not created, click the code section and hit the **Run** button again.

In [3]:
```
import matplotlib.pyplot as plt

# create scatterplot of variables mpg against wt.
plt.plot(cars_df["wt"], cars_df["mpg"], 'o', color='red')

# set a title for the plot, x-axis, and y-axis.
plt.title('MPG against Weight')
plt.xlabel('Weight (1000s lbs)')
plt.ylabel('MPG')

# show the plot.
plt.show()
```



Step 3: Scatterplot of miles per gallon against horsepower

The block of code below will create a scatterplot of the variables "miles per gallon" (coded as mpg in the data set) and "horsepower" of the car (coded as hp).

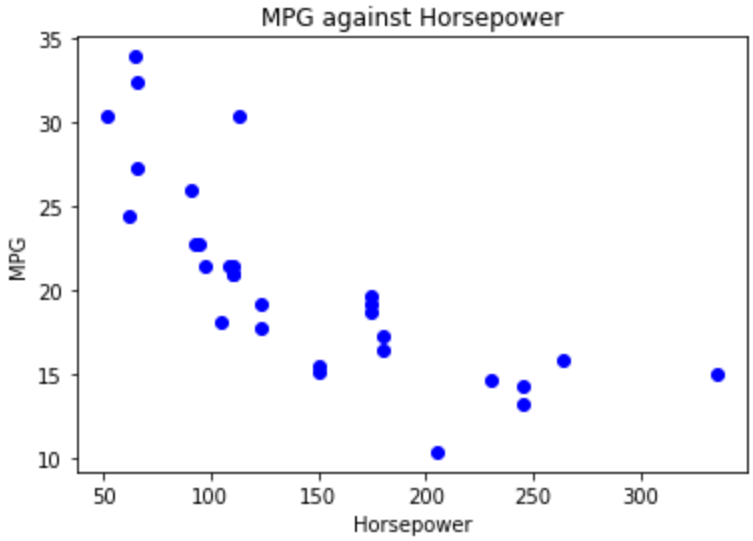
Click the block of code below and hit the **Run** button above.
NOTE: If the plot is not created, click the code section and hit the **Run** button again.

In [4]:
```
import matplotlib.pyplot as plt

# create scatterplot of variables mpg against hp.
plt.plot(cars_df["hp"], cars_df["mpg"], 'o', color='blue')

# set a title for the plot, x-axis, and y-axis.
plt.title('MPG against Horsepower')
plt.xlabel('Horsepower')
plt.ylabel('MPG')

# show the plot.
plt.show()
```



Step 4: Correlation matrix for miles per gallon, weight and horsepower

Now you will calculate the correlation coefficient between the variables "miles per gallon" and "weight". You will also calculate the correlation coefficient between the variables "miles per gallon" and "horsepower". The **corr** method of a dataframe returns the correlation matrix with the correlation coefficients between all variables in the dataframe. You will specify to only return the matrix for the three variables.

Click the block of code below and hit the **Run** button above.

In [5]:
```
# create correlation matrix for mpg, wt, and hp.
# The correlation coefficient between mpg and wt is contained in the cell for
mpg row and wt column (or wt row and mpg column).
# The correlation coefficient between mpg and hp is contained in the cell for
mpg row and hp column (or hp row and mpg column).
mpg_wt_corr = cars_df[['mpg','wt','hp']].corr()
print(mpg_wt_corr)
```

```
          mpg        wt        hp
mpg  1.000000 -0.854687 -0.767064
wt  -0.854687  1.000000  0.647527
hp  -0.767064  0.647527  1.000000
```

Step 5: Multiple regression model to predict miles per gallon using weight and horsepower

This block of code produces a multiple regression model with "miles per gallon" as the response variable, and "weight" and "horsepower" as predictor variables. The **ols** method in statsmodels.formula.api submodule returns all statistics for this multiple regression model.

Click the block of code below and hit the **Run** button above.

In [6]:
```
from statsmodels.formula.api import ols

# create the multiple regression model with mpg as the response variable; weig
ht and horsepower as predictor variables.
model = ols('mpg ~ wt+hp', data=cars_df).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                      mpg   R-squared:                       0.80
9
Model:                              OLS   Adj. R-squared:                  0.79
5
Method:                 Least Squares   F-statistic:                     57.2
1
Date:                Fri, 06 Dec 2024   Prob (F-statistic):           1.96e-1
0
Time:                        05:05:42   Log-Likelihood:                -70.34
5
No. Observations:                  30   AIC:                             146.
7
Df Residuals:                      27   BIC:                             150.
9
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
Intercept     37.4222      1.760     21.263      0.000      33.811      41.03
3
wt            -3.9621      0.709     -5.587      0.000      -5.417      -2.50
7
hp            -0.0312      0.009     -3.334      0.002      -0.050      -0.01
2
==============================================================================
=
Omnibus:                        4.769   Durbin-Watson:                   1.47
3
Prob(Omnibus):                  0.092   Jarque-Bera (JB):                3.59
0
Skew:                           0.840   Prob(JB):                        0.16
6
Kurtosis:                       3.224   Cond. No.                         60
4.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```