Module Five Discussion: Correlation and Simple Linear Regression

This notebook contains the step-by-step directions for your Module Five 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.

Step 1: Generating cars dataset

This block of Python code will generate the sample data for you. You will not be generating the dataset using numpy module this week. Instead, the dataset 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 into a Python dataframe which you will use in later calculations.

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

In [1]:

```python
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 without replacement from mtcars dataset to make the data unique to you.
cars_df = cars_df_orig.sample(n=30, replace=False)

# print only the first five observations in the data set.
print("\nCars data frame (showing only the first five observations)")
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
9	Merc 280	19.2	6	167.6	123	3.92	3.44	18.30	1	0	4	4
17	Fiat 128	32.4	4	78.7	66	4.08	2.20	19.47	1	1	4	1
5	Valiant	18.1	6	225.0	105	2.76	3.46	20.22	1	0	3	1
12	Merc 450SL	17.3	8	275.8	180	3.07	3.73	17.60	0	0	3	3
0	Mazda RX4	21.0	6	160.0	110	3.90	2.62	16.46	0	1	4	4

Step 2: Scatterplot of miles per gallon against weight

The block of code below will create a scatterplot of 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.

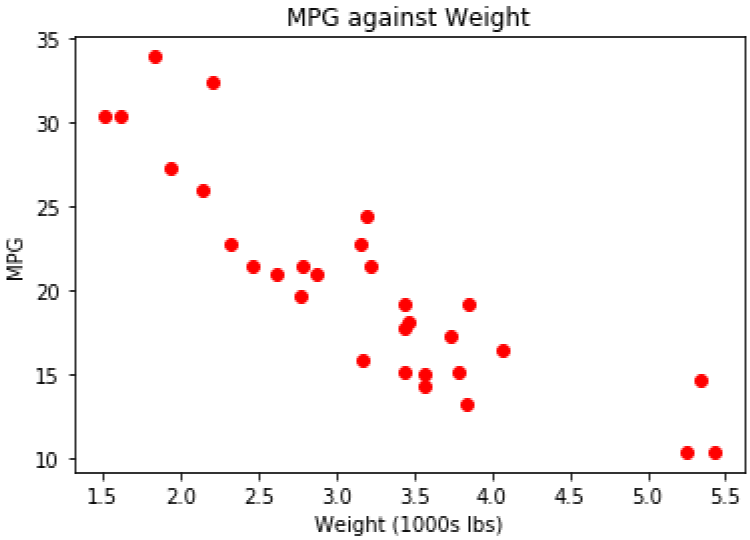
In [3]:

```
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: Correlation coefficient for miles per gallon and weight

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

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

In [4]:

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

```
          mpg        wt
mpg  1.000000 -0.869365
wt  -0.869365  1.000000
```

Step 4: Simple linear regression model to predict miles per gallon using weight

The block of code below produces a simple linear regression model using "miles per gallon" as the response variable and "weight" (of the car) as a predictor variable. The **ols** method in statsmodels.formula.api submodule returns all statistics for this simple linear regression model.

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

In [5]:

```
from statsmodels.formula.api import ols

# create the simple linear regression model with mpg as the response variable a
nd weight as the predictor variable
model = ols('mpg ~ wt', data=cars_df).fit()

#print the model summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.756
Model:                            OLS   Adj. R-squared:                  0.747
Method:                 Least Squares   F-statistic:                     86.66
Date:                Thu, 11 Apr 2024   Prob (F-statistic):           4.56e-10
Time:                        18:17:47   Log-Likelihood:                -75.475
No. Observations:                  30   AIC:                             155.0
Df Residuals:                      28   BIC:                             157.8
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     37.2825      1.911     19.508      0.000      33.368      41.197
wt            -5.3106      0.570     -9.309      0.000      -6.479      -4.142
==============================================================================
Omnibus:                        2.355   Durbin-Watson:                   1.863
Prob(Omnibus):                  0.308   Jarque-Bera (JB):                1.915
Skew:                           0.609   Prob(JB):                        0.384
Kurtosis:                       2.774   Cond. No.                         12.2
==============================================================================

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

End of initial post

Attach the HTML output to your initial post in the Module Five discussion. The HTML output can be downloaded by clicking **File**, then **Download as**, then **HTML**. Be sure to answer all questions about this activity in the Module Five discussion.