# Data Analysis With Python – Module 13 – Case Studies

### Case Study #1: Linear Regression of Height & Weight

This case study will use data science and machine learning to predict mass when height is known. The dataset is a sample of women aged 30-39, derived from here:
https://en.wikipedia.org/wiki/Simple_linear_regression#Numerical_example

Please perform the following steps to complete this case study:

1. Create a new empty Jupyter Notebook.
2. Import all the modules required for:
    - numpy
    - pandas
    - matplotlib
    - seaborn
    - LinearRegression
3. Read the height_mass.csv file into a Pandas DataSet called: people
    - Use the pandas read_csv method.
4. Use a Seaborn histplot to show the distribution for Mass.
    - https://seaborn.pydata.org/generated/seaborn.histplot.html
    - Experiment with different bin #'s and find which one gives you the clearest information about the data.
    - What does the plot tell you about the data? Be specific. Don't focus on the numbers as much as the visual attributes.
    - Insert a markdown cell and note your observations. Superficial answers will lose marks.
5. Use a Seaborn histplot to show the distribution for Height.
    - Experiment with different bin #'s and find which one gives you the clearest information about the data.
    - What does the plot tell you about the data? Be specific. Don't focus on the numbers as much as the visual attributes.
    - Insert a markdown cell and note your observations. Superficial answers will lose marks.
6. Use a Seaborn jointplot to plot x=Height, y=Mass
    - If you get warnings, use named arguments like:
        - jointplot(x="x axis column name", y="y axis column name")
    - Does this plot confirm what the histplot showed?
    - Insert a markdown cell and note your observations. Superficial answers will lose marks.
7. Split the data into training and testing data, using appropriate variable names.
    - Prepare your x and y:
        - x: Drop the Mass column.
        - y: Specify the Mass column.
    - Use sklearn train_test_split to split the data.

8. Create the model and fit it to the training data.
    - Create a sklearn LinearRegression model.
    - Use the fit method to fit it to the training data.
9. Predict values based on testing data.
    - Use the predict method to predict values with the x testing data.
10. Print out error metrics:
    - Mean Absolute Error (MAE)
    - Mean Squared Error (MSE)
    - Root Mean Squared Error (RMSE)
11. Predict some specific mass. Choose any height directly from the data, predict the weight for that height, and see whether the prediction is close to reality.
    - Use the predict method and feed it a 2d array like: [[1.70]]
    - Add a markdown cell and explain how well the prediction matched reality, with specific attention to the RMSE error. Be specific and compare the numbers. Superficial answers will lose marks.
12. Use seaborn to display an lmplot with the linear regression line shown (fit_reg=True).
    - If you get warnings, use named arguments like:
        - lmplot(x=”x axis column name”, y=”y axis column name”)
    - Does this plot support your observations from task #6?
    - Insert a markdown cell and note your observations. Superficial answers will lose marks.