# Topics

* Pandas CSV Import
* Pandas Statistical Functions
* Pandas Mathematical Functions

# Background Information

we are going to make a program which will implement the features of the Pandas library.

Instructions

Important: Complete the below steps in the order they are given. Completing the steps out of order may complicate the assignment or result in an incorrect result.

## Creating a Python Script

1. Download and extract the provided Data Files ZIP file. It contains the following files for use in this assignment:
   1. Pokemon.csv – comma separated value file containing the names, types, health points, and various measured attributes for attack and defense about the Pokemon.

|  |  |  |  |
| --- | --- | --- | --- |
| Column Index | Column Name | Data Type | Description |
| 0 | id | Integer | ID or row of each Pokemon in the data set |
| 1 | Name | String | Name of the Pokemon |
| 2 | Type1 | String | First ability |
| 3 | Type2 | String | Second ability |
| 4 | Total | Integer | Sum of all power points |
| 5 | HP | Integer | Health Points |
| 6 | Attack | Integer | Attack points |
| 7 | Defense | Integer | Defense points |
| 8 | SpAtk | Integer | Speed points on attack |
| 9 | SpDef | Integer | Speed points on defense |
| 10 | Speed | Integer | Speed points |
| 11 | Generation | Integer | Number of generation from 1 to 6 |
| 12 | Legendary | Boolean | True whether it is a legendary character or not |

1. Create a new Python Script named lastname\_firstname\_pp23.py
2. At the beginning of your script, create a block comment to do the following:

# Firstname Lastname  
# Date  
# Computer Science 205 Section YY  
# Pandas Participation Project

1. Import the following libraries:
   1. **pandas as pd**

## Use the Pandas Library to Read the CSV File

1. Create a variable named df and assign the output of the read\_csv() function to df.
   1. The read\_csv() function is part of the Pandas library imported as **pd** during Step 4.
   2. The read\_csv() function takes the filename, column index, and many others as parameters. We will use the relative path of the **Pokemon.csv** file and set the index\_col = ‘id’.

**HINT**: Your df variable assignment should look like this:

df = pd.read\_csv(“path/to/file.csv”, index\_col = ‘id’)

Replace “path/to/file.csv” with the relative path to the **Pokemon.csv** file.

1. Print the data-frame head and tail.
   1. Use a print() statement to display the data frame head by calling the head() function on the df variable you loaded the csv in to.
      1. Output the entire header to the console with df.head()
      2. Now, add a 5 as a parameter to the head() method to display only the first five rows of the dataset in the console.
   2. Use a print() statement to display the data frame tail by calling the tail() function on the df variable you loaded the csv in to.
      1. Output the entire header to the console with df.head()
      2. Now, add a 5 as a parameter to the tail() method to display only the last five rows of the dataset in the console.
2. Create global x and y variables.
   1. Create a variable called x and assign the list of values from the data frame ‘price’ column to it.

**HINT**: It should look like this:

x = df[‘price’]

* 1. Create a variable called y and assign the list of values from the data frame ‘points’ column to it.

**HINT**: It should look like this:

y = df[‘points’]

1. Create a regression() function to do the following:
   1. Perform a multi-variable assignment by assigning slope, intercept, r\_value, p\_value, std\_err to the output of stats.linregress(x, y) using the global x and y variable defines in Step 7.

Hint: This will be used to store the array containing the pH values of each wine.

* 1. Create a list variable named results and assign the slope, intercept, r\_value, p\_value, std\_err variables to it.

Hint: It should look like this:

results = [slope, intercept, r\_value, p\_value, std\_err]

* 1. Return the results variable.

1. Do not modify the display\_stats() function.
2. Create a plot\_regression() function that takes in three variables as parameters to do the following:
   1. Create a variable inside the function called line\_y and set it equal to results[1] + results[0] \* x.
   2. Plot the wine data points.
      1. Use the **Matplotlib** import **plt** to call the plot().
      2. Take in the x and y global variables assigned in Step 7, ‘o’, and assign label to ‘Wines’ as parameters.

Hint: It should look like this:

plt.plot(x, y, 'o', label='Wines')

* 1. Plot the regression line.
     1. Use the **Matplotlib** import **plt** to call the plot().
     2. Take in the global x variable, line\_y, ‘r’, and assign label to ‘Fitted Line’ as parameters.

Hint: It should look like this:

plt.plot(x, line\_y, 'r', label='Fitted Line')

* 1. Add a legend using the legend() function.
  2. Set the title to ‘Wine Scores By Price’ using the title() function.
  3. Show the graph using the show() function.

1. Create a main() function and call the functions we created.
   1. Create a variable called res to store the list of regression values returned from calling the regression() function.
   2. Call the display\_stats() function taking in the res variable as its parameter.
   3. Call the plot\_regression() function taking in the global x and y variables, and local res variable as parameters.
2. Instantiate the main() function with the following if statement at the bottom:

if \_\_name\_\_ == "\_\_main\_\_":

main()

Grading Rubric

This assignment is worth 10 points. It will be graded by your instructor using this rubric:

|  |  |  |
| --- | --- | --- |
| Standard | Meets Requirements (10 points) | Does Not Meet Requirements (0 points) |
| Student made reasonable effort in correctly completing assignment. | Assignment is at least 70% complete and correct, or student contacted instructor for help on incorrect or incomplete items. | Assignment is less than 70% complete and correct, and student did not contact instructor for assistance on incorrect or incomplete items. |

This rubric will be used for peer evaluation of this assignment:

|  |  |  |  |
| --- | --- | --- | --- |
| Standard | Excellent | Satisfactory | Needs Improvement |
| Assignment is correct and complete. | Assignment is at least 90% complete and correct. | Assignment is 70%-89% complete and correct. | Assignment is less than 70% complete and correct. |