Assignment and Assessment Information

COMP 4433

Assignment 1 (due by midnight MST the day prior to Live Session 2)

- Assignment 1, Part 1: Construct a CSV file, called elements.csv, with the first eight elements of the periodic table. Include columns for name, symbol, and atomic_number. Create a function, called atomic_elements, which takes in a file name as an argument. The function should read in the file and create a pandas DataFrame out of it. The function should also add a ninth and 10th element. It should also add a column with the atomic weights rounded to the nearest integer. Return the DataFrame.

- Assignment 1, Part 2: Write a function, called arr_to_df, which takes no arguments. The function will make a list of strings for nine Greek letters, for example 'alpha', the list should not be in alphabetic order. Then make two 9-element numpy arrays of random floating-point numbers with an estimated mean of 10 and a standard deviation of 1.5. Then make another array of nine elements ranging from zero to two times pi, name it 'angle'. Then make another array holding the cosine of that 'angle' array. Construct a dictionary from all of the above. Create a DataFrame from that dictionary, and print it out. Sort the DataFrame ascending on the Greek letters, drop two columns of your choice, drop one of the rows, and return the DataFrame.

- Assignment 1, Part 3: Write function, called fib, which takes no arguments. It should create a list of the first 12 Fibonacci numbers starting with 1, not zero. Then iterate over the last five numbers to build another list with the ratio of each number to its predecessor. Return the list with the ratios. What do you observe about this latter list?

- Assignment 1, Part 4: Write a function, called k_to_r, which takes a temperature in Kelvin (float) as an argument. The function will convert the temperature in Kelvin to Rankine, return the Rankine temperature value. Make a list of five Kelvin temperatures, use the function to make the conversion and print out the values after each conversion. Repeat using a lambda function.

Assignment 2 (due by midnight MST the day prior to Live Session 4)

- Assignment 2, Part 1: Write a class, called AtomicWeights, which takes no arguments. Create a class variable containing a list of the atomic weights of the first six elements of the periodic table, each rounded to the nearest integer.

Define a method which takes two elements, strings, as arguments. The method will create two pie charts using the elements in the class variable created above. Use the following format for the pie charts: (1) each slice should be annotated with a percentage of the whole and (2) each slice annotated with its atomic weight. Using the two elements passed in as arguments, explode the first element passed in for the first chart and explode the second element passed in for the second chart.

- Assignment 2, Part 2: Write a function, called my_charts, which takes a file name as an argument. Read into a DataFrame the file py_ide2.csv, and provide both a horizontal bar chart and a vertical bar chart, complete with all labels. Be sure to rotate the IDE names so that they are readable.

- Assignment 2, Part 3: Write a function, called values_dates, which takes no arguments. Construct a list of eight strings that represent days evenly spread out. Drawing from the random uniform distribution, make an array of eight floats ranging from 100 to 200 in value. Create a DataFrame from that list and that array, convert the dates to pandas datetime objects, and set them to the index. Make two charts in the same window or canvas as follows: (1) a line plot of the values vs. dates and (2) a bar chart of the same.

- Assignment 2, Part 4: Create a class, called YahooFinance. The class will nee to have the following methods:

  - __init__: contractor which will take a ticker symbol and a duration as arguments. The “duration” argument should have a default value of “1mo”. The constructor will create an attribute called ticker with the ticker argument passed in. The constructor must then pull data from Yahoo! Finance using the load_data method described below.

  - load_data: this method takes no arguments. Using the ticker attribute, retrieve the trading data for the appropriate company and store the closing prices and volumes of the stock of your choice over the trading days of one month (potentially more or less if “duration” is larger or smaller than 1 month). Save the data using a pandas DataFrame in an attribute called company_df.

  - plot_data: this method takes no arguments. It should plot the prices and volumes on a canvas in two separate panels, one above the other, with the dates aligned.

Hint: to pull data from Yahoo ! Finance you will need to install the yfinance module and use the Ticker method.

Assignment 3 (due by midnight MST the day prior to Live Session 6)

- Assignment 3, Part 1: Write a function, called autompg, which takes no arguments. Using the built-in Seaborn dataset mpg, provide a heatmap of the correlation of all the numeric columns and provide a pairplot of the same.

- Assignment 3, Part 2: Write a function, called my_precious, which takes two strings as arguments. These strings will represent the color grades for diamonds so they must be upper case letters. Using the built-in Seaborn dataset diamonds, establish a FacetGrid based on 'cut' and 'color'. Eliminate the two colors passed into the function as well as the cut 'Fair'. Within that grid, plot the scatterplot for 'price' vs. 'carat'.

- Assignment 3, Part 3: Write a function called, crashes, which takes no arguments. Using the built-in Seaborn dataset car_crashes, prepare plots with a scattergram with the linear model for both the total vs. speeding and the total vs. alcohol.

- Assignment 3, Part 4: Write a function, called iris, which takes no arguments. Using the built-in Seaborn dataset iris, provide a plot with four subplots wherein the distribution of each of the numeric columns is presented as a set of boxplots, one for each 'species'.

Assignment 4 (due by midnight MST the day prior to Live Session 8)

- Write a function, called the_48, which takes no arguments. Using free and open source tools, provide a set of choropleth visualizations for each of the columns containing dates such that the resulting visualizations (48 states only) tell the story by conveying through color, texture, or both the time lines of achievement of each milestone/column in the provided dataset. Missing data are of particular interest in that when a state has never achieved a given milestone, that should be indicated in a standout manner such as cross-hatching. Consider that the publication may be grayscale. Provide a solution for that as well. Provide the titles, labels, and legends necessary for clarification. File support is given as follows: SturmCodebook has the explanation. SturmData is the data CSV.

Assignment 5 (due by midnight MST the day prior to Live Session 10)

- Assignment 5, Part 1: Write a function, called capitals, which takes no arguments. Using the file capitals_lat_lon.csv, form a DataFrame, and write out an HTML file that when rendered in a browser displays a marker for each capital city. Hint: Use folium.

- Assignment 5, Part 2: Write a function, called triangulation, which takes 3 arguments. Each of these arguments represent a longitude and latitude for a point on the globe. All 3 arguments will have a default value of None. If the function is called without any values, i.e. the arguments are None, you must ask the user for input where they estimate the latitude and longitude for three points of a triangle that would just cover Africa or come close. (Hint: Rabat, Cape Town, and Mogadishu, for example.) Use folium to demonstrate where the points are on the map so that a person could see at a glance that they do nearly cover Africa. Use Shapely to define a polygon from those three points. Compute the area and perimeter of the triangle assuming flat earth and allowing each degree of latitude and each degree of longitude to be considered one unit of length.

- Assignment 5, Part 3: Write a function, called states, which takes two arguments. The first argument will represent the the boundary for Kansas represented with 4 lat-lon points and the second argument will be the boundary for Nebraska represented with 6 lat-lon points. The function must then construct a GeoJSON file from that and save it with the name "kansas_nebraska". Then write another function, called plot_states, which takes a file name as an argument which will read a geojson file, form a dictionary, and plot the result.

- Assignment 5, Part 4: Write a function, called if_youre_happy_and_you_know_it, which takes no arguments. It will generate at random the "happiness index" for each state in the United States. Provide a choropleth of the United States with that data represented there on.

- Assignment 5, Part 5: Write a function, called random_hist, which takes an int, n, as an argument. Generate n points from the random exponential distribution. Choose your own parameters for that. Use Plotly to plot a histogram of that distribution such that there are at least 15 bins with counts greater than zero.