General Points

- Use the course material located at:
  - Python Data Science Handbook
- Assignment 11 can be completed using previously covered material and content from the following chapters:
  - 00.00-Preface *through* 03.07-Merge and Join
- After completing requirements, test to ensure all cells run correctly in the .ipynb file.
- Include appropriate markdown cells to identify the requirements below by number. See this example.
- Produce an .html file that shows the .ipynb after a *successful test run*.
  - by File | Download as | HTML (.html) .
- Test the .html file by opening it in a browser and ensure the content is produced correctly from the run in Jupyter Notebook.
- Submit BOTH the .ipynb and .html files to the appropriate link in Blackboard | Assignments. Submit the files individually (via a multi-select). However, if your browser posts an error for the .html file, submit it as a .zip.
- Submit any additional files required to complete the assignment.

Requirements

(*Ensure that all Requirements are complete*)

1. Using Jupyter Notebook (or similar tool), create a file named:
   - assignment-11.ipynb
2. Add an H1 markdown: "This is Assignment 11 - <yournamehere>"
3. In your own words, describe the commonalities and differences between NumPy arrays and Pandas series.

4. Create a Pandas series named bill_names with American currency bill denominations as indices (keys) and President last names as values. For example: 1 – Washington, 2 – Jefferson, etc.
5. Create a Pandas series-as-dictionary beverages_dict. Make the indices beverage names and comments about the beverage as values. For example: 'Aquafina': 'This is my favorite bottled water!'. Demonstrate accessing:
    - individual values via the keys
    - multiple values via slicing
6. Obtain the real *city proper* population data for the following cities: Chongquing, Shanghai, Tokyo, Moscow, Mexico City, London, & New York. Create a series-as-dictionary named population based on this information.
7. Create a Pandas series-as-dictionary named city_country with the cities in population and the countries for each city.
8. Create a Pandas dataframe object named city_dataframe from a dictionary of series objects using population and city_country. Show:
    - the .index property
    - the .columns property
    - the .keys() method
9. Create a Pandas series object named my_pd_series from a collection of string keys and collection of numeric values of your choosing. Demonstrate:
    - modifying a value based on key
    - slicing by explicit index
    - slicing by implicit integer index
    - masking
    - fancy indexing
    - loc[]
    - iloc[]
10. Using city_dataframe, demonstrate:
    - access column via dictionary-style indexing of the column name
    - access column via column names that are strings

- add a new column to the city_dataframe named altitude. Hint: See this example for adding columns:

```
area = pd.Series({'California': 423967, 'Texas': 695662,
                  'New York': 141297, 'Florida': 170312,
                  'Illinois': 149995})
pop = pd.Series({'California': 38332521, 'Texas': 26448193,
                 'New York': 19651127, 'Florida': 19552860,
                 'Illinois': 12882135})
data = pd.DataFrame({'area':area, 'pop':pop})
data
```

11. Create two Pandas series that when added using '+' produce some NaN entries. Use the .add() method and a fill_value to replace the NaN entries.
12. Create two Pandas dataframes that when added using '+' produce some NaN entries. Use the .add() method and a fill_value of the mean of one of the dataframes to replace the NaN entries.
13. Create a two-dimensional NumPy array using:

    A = rng.randint(5, 10, size=(4, 4))

    Demonstrate: subtracting row 0 of A from A

14. Create a Pandas dataframe using:

    df = pd.DataFrame(A, columns=list('QRST'))

    Demonstrate: subtracting row 1 of df from df using df.iloc[1]

15. Compare and contrast the two sentinel values Pandas uses to represent missing data.
16. Demonstrate the %timeit difference between operations using Python objects and Python integers.
17. Create a Pandas series containing null data. Use .isnull() to identify the entries that are null.
18. Create a Pandas dataframe containing null values. Demonstrate:

- drop all rows containing a null value
- drop all columns containing a null value
- drop only rows that contain all null values
- drop only columns that contain all null values
- replacing null with 0
- forward fill
- backward fill

19. Demonstrate Pandas MultiIndex techniques (can use book examples):
    - .from_tuples()
    - .reindex()
    - .unstack()
    - .stack()
    - indexing and slicing
20. Demonstrate concatenating two Pandas series. One series contains automobile data. The other series contains motorcycle data.
21. Demonstrate concatenating two Pandas dataframe using the .append() method.
22. Using Pandas dataframes, demonstrate the following joins:
    - one-to-one
    - many-to-one
    - many-to-many
23. Using Pandas dataframes, demonstrate merge with the following keywords:
    - on
    - left_on and right_on
    - left_index and right_index
24. Use markdown to include a statement at the end of assignment-11.ipynb explaining your experiences with Assignment 11. Make this authentic (minimum of 2-3 sentences).

TEST – TEST – TEST your .ipynb file to ensure all requirements are met. Produce an .html file from a *successful test run* of the .ipynb file. Ensure that the .html is produced correctly by opening it in a browser and inspecting the output of each cell.

- Use the list above as a confirmation checklist.
- Not meeting all requirements = 0 points for the assignment.