Python for Computer Science and Data Science 1 (CSE 3651)

MINOR ASSIGNMENT-6: NUMPY AND PANDAS

1. Create a 2-by-3 array with ones, a 3-by-3 array with zeros and a 2-by-5 array with sevens.
2. Use arange to create a 2-by-2 array containing the numbers 0–3. Use broadcasting to perform each of the following operations on the original array:
   (a) Cube every element of the array.
   (b) Add 7 to every element of the array.
   (c) Multiply every element of the array by 2.
3. Create a 3-by-3 array containing the even integers from 2 through 18. Create a second 3-by-3 array containing the integers from 9 down to 1, then multiply the first array by the second.
4. Create a 2D array and swap the first two rows and two columns.
5. Create a 2-by-5 array from an argument which is a list of the two five-element lists $[2, 3, 5, 7, 11]$ and $[13, 17, 19, 23, 29]$.
6. Create a 2-by-3 array containing the first six powers of 2 beginning with 2^0. Flatten the array first with method flatten, then with ravel. In each case, display the result then display the original array to show that it was unmodified.
7. Find the most frequent values in an array of positive integers. The original array is [6 9 5 1 7 5 1 0 1 5 5 0 8 9 0 7 0 7 6 5 1 1 9 5 3 8 7 9 6 3 4 5 9 7 2 7 0 2 2 6].
8. (needs to change) Use linspace and reshape to create a 2-by-3 array with the values $1.1, 2.2, \ldots, 6.6$. Then use astype to convert the array to an array of integers.
9. Write a function `format_2d_array(arr)` that takes a two-dimensional array of positive integers (represented as nested lists) and returns a formatted string that mimics NumPy's column-based format. In this format, each element in the array should be right-aligned, and the width of each column should be determined by the number of characters required to display the largest element in the array.
10. Create an array containing the values 1–15, reshape it into a 3-by-5 array, then use indexing and slicing techniques to perform each of the following operations:
    (a) Select row 2.
    (b) Select column 5.
    (c) Select rows 0 and 1.
    (d) Select columns 2–4.
    (e) Select the element that is in row 1 and column 4.
    (f) Select all elements from rows 1 and 2 that are in columns 0, 2 and 4.
11. Given the following two-dimensional arrays in NumPy:

```
array1 = np.array([[0, 1], [2, 3]])
array2 = np.array([[4, 5], [6, 7]])
```

    Perform the following tasks:

(a) Use vertical stacking to create a 4-by-2 array named `array3`, with `array1` stacked on top of `array2`.

(b) Use horizontal stacking to create a 2-by-4 array named `array4`, with `array2` to the right of `array1`.

(c) Use vertical stacking with two copies of `array4` to create a 4-by-4 array named `array5`.

(d) Use horizontal stacking with two copies of `array3` to create a 4-by-4 array named `array6`.

12. Use NumPy's concatenate Function to reimplement the previous problem.

13. Use NumPy's tile function to create a checkerboard pattern of dashes and asterisks

14. Use the NumPy bincount function to count the number of occurrences of each non-negative integer in a 5-by-5 array of random integers in the range $0 - 99$.

15. Write functions median and mode that use existing NumPy capabilities to determine the median (middle) and mode (most frequent) of the values in an array. Your functions should determine the median and mode regardless of the array's shape. Test your function on three arrays of different shapes.

16. Write a NumPy program to create a 9*9*2 array with random values and extract any array of shape (5,5,2) from the said array.

17. Write a code to create a 4*4 array with random values and sort each column.

18. Write a Pandas program to convert the first column of a DataFrame as a Series.

19. Convert s1=[1,2,3,4,2] and s2=[3,4,5,6] to two series objects. Find elements in s1, which are not present in s2.

20. Write a Pandas program to find the index of the first occurrence of the smallest and largest value of a given series. If the input is [1,1, 3, 7,88, 12, 88, 23, 3, 1, 9, 0], the output should be 0 and 4.

21. Convert L=['Cry', 'Apple', 'Orange', 'Sky', 'Banana'] to a pandas series. Create a new series with the elements which has a vowel. Create another series which starts with a vowel.

22. Perform the following tasks using the pandas `Series` object:

(a) Create a Series from the list `[7, 11, 13, 17]`.

(b) Create a Series with five elements where each element is `100.0`.

(c) Create a Series with 20 elements that are all random numbers in the range 0 to 100. Use the `describe` method to produce the Series' basic descriptive statistics.

(d) Create a Series called `temperatures` with the following floating-point values: `98.6`, `98.9`, `100.2`, and `97.9`. Use the `index` keyword argument to specify the custom indices 'Julie', 'Charlie', 'Sam', and 'Andrea'.

(e) Form a dictionary from the names and values in Part (d), then use it to initialize a Series.

23. Perform the following tasks using the pandas `DataFrame` object:

(a) Create a DataFrame named `temperatures` from a dictionary of three temperature readings each for 'Maxine', 'James', and 'Amanda'.

(b) Recreate the DataFrame `temperatures` in Part (a) with custom indices using the `index` keyword argument and a list containing 'Morning', 'Afternoon', and 'Evening'.

(c) Select from `temperatures` the column of temperature readings for 'Maxine'.

(d) Select from `temperatures` the row of 'Morning' temperature readings.

(e) Select from `temperatures` the rows for 'Morning' and 'Evening' temperature readings.

(f) Select from `temperatures` the columns of temperature readings for 'Amanda' and 'Maxine'.

(g) Select from `temperatures` the elements for 'Amanda' and 'Maxine' in the 'Morning' and 'Afternoon'.

(h) Use the `describe` method to produce `temperatures`' descriptive statistics.

(i) Transpose `temperatures`.

(j) Sort `temperatures` so that its column names are in alphabetical order.