Week 4 Deliverables

Overview: In this week, you have studied additional Python language syntax including Arrays and Strings. In particular, you used the numpy, regular expressions, Panda and DataFrames libraries to help manipulate and store data. The Lab for this week demonstrates your knowledge of this additional Python functionality. Be sure to use the examples in the textbook reading along with the associate libraries, functions and processes when completing the assignments for this week.

Be sure to develop and test your Python code in the AWS Cloud9 IDE provided for the class.

You should continue to use the PEP Python Style guide mentioned in the book and found here:

https://www.python.org/dev/peps/pep-0008/

Some examples of Python Coding Style best practices include:

- Limit all lines to a maximum of 79 characters.
- Imports are always put at the top of the file, just after any module comments and before module globals and constants.
- Use 4 spaces for indentation.

Submission requirements for this project include 3 files. (Zipping them into one file is acceptable and encouraged):

- Python Matrix Math Application Code
- Python Data Munging Application Code
- Word, Excel or PDF file containing your test results

Python Applications for Lab4: (total 100 points):

This lab consists of two parts.

The first exercise **(40 points)** allows a user to enter the values of two, 3x3 matrices and then select from options including, addition, subtraction, matrix multiplication, and element by element multiplication. You should use numpy.matmul() for matrix multiplication (e.g. np.matmul(a, b)). The program should computer the appropriate results and return the results, the transpose of the results, the mean of the rows for the results, and the mean of the columns for the results.

When entering data you should check that each value is numeric for the matrices. The user interface should continue to run until the user indicates they are ready to exit.

A user interface might look similar to this:

```
********** Welcome to the Python Matrix Application**********
    Do you want to play the Matrix Game?
    Enter Y for Yes or N for No:
    Y
```

```
Enter your first 3x3 matrix:

1 2 4
4 2 1
3 8 9

Your first 3x3 matrix is:

1 2 4
4 2 1
3 8 9

Enter your second 3x3 matrix:

3 2 1
7 2 5
5 2 1

Your first 3x3 matrix is:

3 2 1
7 2 5
5 2 1

Select a Matrix Operation from the list below:

a. Addition
b. Subtraction
c. Matrix Multiplication
d. Element by element multiplication

a

You selected Addition. The results are:

4 4 5

11 4 6

8 10 10

The Transpose is:

4 11 8

4 4 10

5 6 10

The row and column mean values of the results are:

Row: 4.33, 7, 9.33

Column: 7.66, 6, 7
```

```
Do you want to play the Matrix Game?
Enter Y for Yes or N for No:
N
*************** Thanks for playing the Matrix Game **********
```

If an inappropriate entry is detected, the program should prompt for a correct value and continue to do so until a correct value is entered.

Hints:

1. Use numpy and associated functionality
2. Create and use functions as often as possible
3. Use comments to document your code
4. Both integers and float values are acceptable

For the second exercise **(40 points)** use Python Panda, Regular Expressions, .map() and other functions as appropriate to format existing address records and eliminate records with missing critical fields. Critical fields include FirstName, Lastname, Zipcode+4, and Phone number for customers. For this exercise, create an array to hold data with these 4 fields containing at least 25 records.

The Zipcode field should contain either traditional 5-digit Zipcode (e.g. 21801) or Zip+4 format (e.g 21801-1101). The phone numbers should contain 10-digit (e.g. 5555555555) or formatted 10-digit (e.g. 555-555-5555). Some records might be corrupt so the data needs to be munged. At this point, we assume only U.S data will be present therefor country code is not needed.

A few records, data might look like this:

```
Jim, Robertson, 21801,555-555-5555
John, Adams, 223211143, 4444444444
Helen, Cooper, edskd-2134,323232
,Franklin,234511, 323-333-2211
…
```

You python code should label each column, properly format the Zip code to be either 11111 or 11111-1111 formats, properly format the phone numbers to always be 111-111-1111 format, and replace incorrect values with a blank string.

For example, after processing the above file, the returned results would be:

Firstname	Lastname	Zipcode	Phone
Jim	Robertson	21801	555-555-5555
John	Adams	22321-1143	444-444-4444
Helen	Cooper		
	Franklin		323-333-2211

Notice invalid data was removed and formatting was applied as required. Commas can be left if desired. Default alignment with the column labels is acceptable.

Hints:

1. Hardwire the 25 records. (i.e. the user doesn't have to enter this)
2. Review the code in the textbook examples to simplify this work
3. Use comments to document your code

3. Document your results for each application within the AWS Cloud9 classroom environment. Provide screen captures and descriptions for each test cases you provide for each of your applications. Be sure to go through each possible user interface combination in your test cases. **(20 points)**

Any submissions that do not represent work originating from the student will be submitted to the Dean's office and evaluated for possible academic integrity violations and sanctions.