

---

# BASE PYTHON AND NUMPY EXERCISES

## INTRODUCTION

This set of exercises leverages base python package functionality as well as Numpy. These exercises cover the following topics:

- Base python
  - strings, float, integers, lists, tuples, dicts, sets, functions, control flow, for loops, while loops, list comprehension and exception handling.
- Numpy

## DATA USED IN THIS EXERCISE:

There is no external data used in this exercise.

## QUESTIONS

- 1) Create two variables named first and last. Assign your first and last name to each variable.
  - a. Use the string format method to print the following string:
    - i. "My first name is \_\_\_\_, my last name is\_\_\_\_"
  - b. Concatenate these two variables together using the "+", include a space.
  - c. Use string methods to print this new variable all lower case, then all upper case, then in capitalized case.
  - d. What is the total length of the concatenated variable?
  - e. Slice the first letters from each string to create your initials with a period following each initial. Print this string.
  - f. Create a list with these two variables as elements of the list.
  - g. Use that list and the following string (" \_") and the .join method to print your name like so: First\_Last

2) Create a list representing the following matrix of data (list of lists):

4	6	7
3	5	7
2	5	4

- a. Subset the list down to the 1<sup>st</sup> row, 1<sup>st</sup> column
  - b. Add the 2<sup>nd</sup> row, 2<sup>nd</sup> column to the 3<sup>rd</sup> row, 3<sup>rd</sup> column
  - c. Using a for loop, get the average value for each row of data. Print a message in each iteration of the loop stating the average.
    - i. Hint: use a for loop, the enumerate function, the sum and len functions.
  - d. Using a for loop, get the average value for each column of data. Print the result each loop iteration.
    - i. Hint: this will require a double loop.
- 3) Use the datetime module in python to determine what weekday the first of each month falls on in 2016. Track your results in a list of tuples, where the first element of each tuple is the month and the second element of each tuple is the weekday. Finally, print your result using a string format.

Hints: Example String format code:

```
"Month is {0}, day is {1}".format("January", "Friday")
```

Example date creation:

```
Import datetime  
datetime.date(12,1,1)
```

- a. Can you do this for multiple years? Perhaps 2012 to 2016?
    - i. Add a loop to do this for the years 2012-2016. Track the year in your results list in addition to the month and weekday. Print the results accordingly.
- 4) Create a function that asks the users to input a number and returns the square root of the number. You can use the sqrt function from the math module. Put the following checks into the function:
- a. If the input fails to convert to a number, notify the user you cannot convert the number and request new input. (Hint- use try: except: and attempt to convert to a float)

- b. Check that the user input is a positive number; if the user inputs a negative number, request the user input a positive number
  - c. Upon success, print a message with the value and return the squared number from the function.
- 5) Using list comprehension and the range function, complete the following tasks:
  - a. Print a list of the numbers -10 through 10
  - b. For the numbers in (a) less than -5 or greater than 5, print their squared value
  - c. For the numbers in (a) greater than 0, return their square root.
  - d. Using the numbers in (a), return the string "NA" if they are less than 0, and return the number + 1 if it is 0 or higher.

## Numpy

- 6) Repeat Number 2 from above using Numpy. This should be much easier!
- 7) Create a 10x100 array drawing from the random normal with mean=10, variance=25.
  - a. First generate the data, and then reshape the array.
  - b. "center" this array by subtracting the observed mean.
  - c. "scale" this array by dividing by the observed variance.
  - d. Confirm your result by calculating the columns means and stdev of the resulting array. What do you expect the mean and stdev to be?
- 8) Create a series randomly generated Poisson distributions with lambda varying between 1 and 20. The size of each distribution draw should be 15 samples (n=15).
  - a. Calculate the variance and the mean for each series.
  - b. Plot the variance against the mean...is the relationship roughly linear?
    - i. Hint: Use the following code to make a basic x,y plot:
 

```
import matplotlib.pyplot as plt
plt.plot(x,y)
```
  - c. Repeat this exercises with n=50 in each group. How does this change your plot? Do it again with n = 100 and n = 1000. What is the relationship between the mean and variance in a Poisson distribution?