Homework #2

* Due date: October 1(M)
** Whenever necessary, capture the screen and submit the printed screen.

1. Input the following mathematical expressions into Python as arrays.

$$u = [1\ 1\ 2\ 3\ 5\ 8]$$

$$v = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 5 \\ 8 \end{bmatrix}$$

$$x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$z = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 \\ 1 & 2 & 1 & 2 \end{bmatrix}$$

$$w = \begin{bmatrix} x & x \\ y & y \end{bmatrix}$$

   Compute the values of $u + v'$, $v + u'$, vu, uv and xy (where the multiplication is as defined as linear algebra).

2. Construct each of the following sequences using linspace, arange and r_:
   0, 1,...,10
   4, 5, 6,..., 13
   0, .25, .5, .75, 1

3. Determine the differences between the rounding by applying round (or around), ceil and floor to
   y = [0, 0.5, 1.5, 2.5, 1.0, 1.0001, -0.5, -1, -1.5, -2.5]

4. Suppose x = [-42 -9 -8 10]. What is the difference between y = sort(x) and x.sort()?

5. Suppose y = [nan 2.2 3.9 4.6 nan 2.4 6.1 1.8] . How can nansum be used to compute the variance or the data? Note: sum(1- isnan(y)) will return the count of non-NaN values.

6. Produce two arrays, one containing all zeros and one containing only ones, of size 10×5.

7. Produce an identity matrix of size 5. Take the exponential of this matrix, element-by-element.

8. How could eye be replaced with diag and ones?

9. Let x = arange(12.0). Use both shape and reshape to produce $1{\times}12$, $2{\times}6$, $3{\times}4$, $4{\times}3$, $6{\times}2$ and $2{\times}2{\times}3$ versions or the array. Finally, return x to its original size.

10. How can a diagonal matrix containing the diagonal elements of

$$y = \begin{bmatrix} 2 & .5 \\ .5 & 4 \end{bmatrix}$$

be constructed using diag?

11. Using the y array from the previous problem, verify that the sum of the eigenvalues is the same as the trace, and the product of the eigenvalues is the determinant.

12. Generate an array of normal numbers of size 10. Find the locations of the elements of the vector having values greater than 0. Extract the elements from the generated vector that are greater than 0.

13. Using for and while, write Python codes that calculates $1/1 + 1/2 + ... + 1/100$.

14. Generate an array of normal numbers of size 1000. Using for, write Python codes that calculates the number of positive elements in the array.

15. Download data for the past 20 years for the S&P 500 from Yahoo!. Plot the price against dates, and ensure the date display is reasonable.