Part 1

Frist, using the csv library, open the 'data.csv' file and read the file. And append what I read using the csv.reader() to array named dataset. Because dataset is consisted of float number, stores it as a float data type.
Then declare class named 'BasicStatistics'. The constructor get dataset as an argument and store it as a member variable to use it at other function in the class. The 'mean()' is function that calculates mean of data in dataset and it don't need any argument. Using the sum function, calculate the sum of data and divide by number of data.

$$\bar{x} = \frac{\sum x_i}{n}$$

The 'median()' is function that calculates median of data in dataset and it don't need any argument too. To calculate the median, first sort the dataset using 'sorted()' function and store it to array named sorted_dataset. If number of data is odd, the median should be '(length of sorted dataset-1)/2'th data in sorted dataset. If it is even, it should be mean of 'length of sorted dataset/2'th data and 'length of sorted dataset/2 -1'th data.

The 'variance()' is function that calculates variance of data. First to calculate the square of the deviation of all data and store it to array named sum_square_dev. Then calculate the sum of sum_square_dev and divide by length of dataset-1. For calculate variance of sample, the denominator should be 'n-1' and when for the variance of population, the denominator should be 'n'. I thought that for the calculation of variance of this dataset, the denominator should be 'n-1' because it is sample.

$$v = \frac{\sum(x_i - \bar{x})^2}{n-1}$$

The 'standard_deviation()' is function that calculates variance of data and it can be easily calculated by rooting the variance.

$$\sigma = \sqrt{v}$$

The 'standardize()' is function that standardize the data. First make array named standardize store standardized data. And this is how to do standardize.

$$z = \frac{x_i - \bar{x}}{\sigma}$$

The 'plot()' is function that show the data as a plot. For histogram plot I colored as blue and label mean, median, standard deviation. And for original dataset plot and standardized dataset plot also colored as blue and label mean and standard deviation.

Part 2

Import a 'numpy' library and create a 10*50 with random integer numbers between -99 and 99 using the numpy function 'numpy.random.randint()'. Since the range and type of the number of matrix components are irrelevant, I just arbitrarily set it as a number between -99 and 99. Print the shape of matrix, ask the user for a row number and print the row given, and then do the same for a column. I didn't work for an error message when the input value was out of range.
To create a matrix, I used 'numpy.array()' function. And to calculate dot product of matrix, I used 'numpy.dot()' function. Because the dot product of a 1D array with itself is simply the sum of the squares of its elements, I just used 'numpy.inner()' function for matrix [1, 2, 3]. And to calculate multiplication, also use 'numpy.dot()' function. Calculate A*B is easy but for A*C because A is a 2*2 matrix and C is a 1*2 matrix, I need to transpose C to make it 2*1 matrix. So, A*C is a 2*1 matrix. To calculate power, determinant, rank, inverse, pseudo inverse, eigenvalues and eigenvectors, I just used functions that is declared in 'numpy' library, and it is identified in the code. And I also create a 4*4 identity matrix using the 'eye()' function and 4*4 matrix using 'ones()' function.