**Answers:**

1. 1) Numpy and its function:

Numpy is an open source add-on modules/library to python, to provide python interface mathematical and numerical routines, fast functions though the routines were built in C, C++. These are highly mature packages that provide numerical functionality that meets, or perhaps exceeds, that associated with commercial software like MatLab. Numpy package provides basic routines for manipulating large arrays and matrices of numeric data.

Numpy has many functions for many different job, for example:

* Import and Export data:

1. np.loadtxt(‘file.txt’) : to load data from a local file.
2. np.genfromtxt('file.csv',delimiter=',') : loading data from a csv file
3. np.savetxt('file.txt',arr,delimiter=' ') : Writes to a text file
4. np.savetxt('file.csv',arr,delimiter=',') - Writes to a CSV file

* Creating arrays:

1. np.array([1,2,3]) - One dimensional array
2. np.array([(1,2,3),(4,5,6)]) - Two dimensional array
3. np.zeros(3) - 1D array of length 3 all values 0
4. np.ones((3,4)) - 3x4 array with all values 1
5. np.eye(5) - 5x5 array of 0 with 1 on diagonal (Identity matrix)
6. np.linspace(0,100,6) - Array of 6 evenly divided values from 0 to 100
7. np.arange(0,10,3) - Array of values from 0 to less than 10 with step 3 (eg [0,3,6,9])
8. np.full((2,3),8) - 2x3 array with all values 8
9. np.random.rand(4,5) - 4x5 array of random floats between 0-1
10. np.random.rand(6,7)\*100 - 6x7 array of random floats between 0-100
11. np.random.randint(5,size=(2,3)) - 2x3 array with random ints between 0-4

* Inspecting properties:

1. arr.size - Returns number of elements in arr
2. arr.shape - Returns dimensions of arr (rows, columns)
3. arr.dtype - Returns type of elements in arr
4. arr.astype(dtype) - Convert arr elements to type dtype
5. arr.tolist() - Convert arr to a Python list np.info(np.eye) - View documentation for np.eye

* Copying/Sorting/Reshaping:

1. np.copy(arr) - Copies arr to new memory
2. arr.view(dtype) - Creates view of arr elements with type dtype
3. arr.sort() - Sorts arr arr.sort(axis=0) - Sorts specific axis of arr
4. two\_d\_arr.flatten() - Flattens 2D array two\_d\_arr to 1D
5. arr.T - Transposes arr (rows become columns and vice versa)
6. arr.reshape(3,4) - Reshapes arr to 3 rows, 4 columns without changing data
7. arr.resize((5,6)) - Changes arr shape to 5x6 and fills new values with 0

* Adding/Removing elements:

1. np.append(arr,values) - Appends values to end of arr
2. np.insert(arr,2,values) - Inserts values into arr before index 2
3. np.delete(arr,3,axis=0) - Deletes row on index 3 of arr
4. np.delete(arr,4,axis=1) - Deletes column on index 4 of arr

* Combining/Splitting:

1. np.concatenate((arr1,arr2),axis=0) - Adds arr2 as rows to the end of arr1
2. np.concatenate((arr1,arr2),axis=1) - Adds arr2 as columns to end of arr1
3. np.split(arr,3) - Splits arr into 3 sub-arrays
4. np.hsplit(arr,5) - Splits arr horizontally on the 5th index

* Indexing/Slicing/Subsettting:

1. arr[5] - Returns the element at index 5
2. arr[2,5] - Returns the 2D array element on index [2][5]
3. arr[1]=4 - Assigns array element on index 1 the value 4
4. arr[1,3]=10 - Assigns array element on index [1][3] the value 10
5. arr[0:3] - Returns the elements at indices 0,1,2 (On a 2D array: returns rows 0,1,2)
6. arr[0:3,4] - Returns the elements on rows 0,1,2 at column 4
7. arr[:2] - Returns the elements at indices 0,1 (On a 2D array: returns rows 0,1)
8. arr[:,1] - Returns the elements at index 1 on all rows arr5) - Returns an array with boolean values
9. ~arr - Inverts a boolean array
10. arr[arr<5] : Returns array elements smaller than 5

* Scalar MATH:

1. np.add(arr,1) - Add 1 to each array element
2. np.subtract(arr,2) - Subtract 2 from each array element
3. np.multiply(arr,3) - Multiply each array element by 3
4. np.divide(arr,4) - Divide each array element by 4 (returns np.nan for division by zero)
5. np.power(arr,5) - Raise each array element to the 5th power

* VECTOR MATH

1. np.add(arr1,arr2) - Elementwise add arr2 to arr1
2. np.subtract(arr1,arr2) - Elementwise subtract arr2 from arr1
3. np.multiply(arr1,arr2) - Elementwise multiply arr1 by arr2
4. np.divide(arr1,arr2) - Elementwise divide arr1 by arr2
5. np.power(arr1,arr2) - Elementwise raise arr1 raised to the power of arr2
6. np.array\_equal(arr1,arr2) - Returns True if the arrays have the same elements and shape
7. np.sqrt(arr) - Square root of each element in the array
8. np.sin(arr) - Sine of each element in the array
9. np.log(arr) - Natural log of each element in the array
10. np.abs(arr) - Absolute value of each element in the array
11. np.ceil(arr) - Rounds up to the nearest int
12. np.floor(arr) - Rounds down to the nearest int
13. np.round(arr) - Rounds to the nearest int

* STATISTICS

1. np.mean(arr,axis=0) - Returns mean along specific axis
2. arr.sum() - Returns sum of arr
3. arr.min() - Returns minimum value of arr
4. arr.max(axis=0) - Returns maximum value of specific axis
5. np.var(arr) - Returns the variance of array
6. np.std(arr,axis=1) - Returns the standard deviation of specific axis
7. arr.corrcoef() - Returns correlation coefficient of array