**Course Title: NumPy Fundamentals (Python Library for Data Science)**

***Note: This Documentation is for Python User!(All codes are written in python)***

**NumPy Overview**

NumPy is an open source library that was built on python for majorly mathematical operations and creating single or multidimensional arrays.

According to lesson 2(NumPy Overview) of the course, NumPy helps to increase the speed of python scripts by vectorizing the operation making it quicker.

1. **How to install NumPy**

The following lines of code below helps install the NumPy package

Enter the following into the terminal

*CODE: #pip3 install numpy*

**Note**: To install the package through ANACONDA

Enter the following into the anaconda prompt/shell environment.

*CODE: #conda install -c anaconda numpy*

**Note**: Other installation method can be using search engines like google, bing etc.

1. **Importing NumPy**

In order to use the NumPy package, on your console, it has **to** be imported first with the code below.

*CODE: #import numpy as np*

1. **Array Creation**

* ***Creating Single dimensional Array***

An array is like an ordered series of items.

*CODE: array\_a = np.array([1, 2, 3, 4])*

*print(array\_a) -OUTPUT: [1 2 3 4 ]*

* ***Creating Multidimensional Array***

A multidimensional array is and array with multiple rows and multiple columns

*CODE: seq\_a = [1, 2,3, 4];seq\_b = [5, 6,7, 8];seq\_c = [9, 10,11, 12]*

*array\_abc = np.array([seq\_a, seq\_b, seq\_c])*

*print(array\_abc)*

**Note:** NumPy arrays are homogeneous this means that it has one type of data in it. To preselect the data type of items in a particular array.

1. **NumPy Methods**

**Shape:** This produces a tuple with the first value representing the rows and the second value representing the columns I.e. (no of rows, no of cols)

*CODE: print(array\_a.shape)*

**Reshape**: This method takes in two arguments, the number of rows and the number of columns (whose multiple must be equal to the size of the matrix being reshaped.)

*CODE:* *array\_c = array\_b.reshape(2,6)*

**Transpose:** This switches the rows with columns.

*CODE: print(array\_c.T)*

**Indexing:** This is done by using integer values in square brackets. It is also referred to as slicing.

Negative indexing is using negative numbers to slice an array.

*CODE: print(array\_c[:, 1])#for all columns and row 1*

*print(array\_c[1:5,2:4])*

**Zeros:** This creates an array of zero values.

*CODE:* *array\_a = np.zeros((3,4))*

**Ones:** This creates on array of one values.

*CODE: print(np.ones((6,8)))*

**Full:** This creates an array of values which is predefined by the user.

*CODE: print(np.full((4,5), 7))*

**Random:** This method creates an array of random numbers.

*CODE: print(np.random.random((6,7)))*

1. **NumPy Math**

NumPy mathematics can be carried out on arrays and matrices provided they are of the same shape other wise a technique called broadcasting is used.

**Single array operations**

1. **Sum():**  This returns the sum of all the elements in the given array.

***Note:*** *Depending on the desired summation attribute the key word [axis = 1] or [axis = 0] can be used for rows and columns summation respectively)*

*CODE:* *arr\_a = np.array([[1, 3, 4],[2,5,6], [2,5,6]]); arr\_b = np.array([[7, 5, 4],[2,5,6],[5,5,9]])*

*print(arr\_a).sum())*

1. **Cumsum():** This returns the cumulative sum of elements in the array

*CODE: print(arr\_a.cumsum())*

1. **Prod():** This returns the products of elements in the array

*CODE: print(arr\_a.prod())*

1. **Cumprod():** This returns the cumulative products of elements in the array

*CODE: print(arr\_a.cumprod())*

**Multiple Array Math**

*CODE:*

* *print(arr\_a+arr\_b) ---- Array Addition*
* *print(arr\_a-arr\_b) -----Array Subtraction*
* *print(arr\_a\*arr\_b) -----Array multiplication*
* *print(arr\_a/arr\_b) -----Array Division*

1. **Broadcasting**

This array mathematics involves array of unequal sizes. It involves a smaller array operating on a bigger array.

***Note:*** *One**of the dimensions of the smaller array must match with one of the dimensions second(bigger) array*

*CODE: arr\_a = np.array([[2,5,6,7], [4,6,7,8],[6,4,3,7]] ----------------)#Shape of (3,4)*

*arr\_b = np.array([[2],[4],[6]]) ---------#shape of (1,3)*

*print(arr\_a+arr\_b)*

**Course Title: Data Science 101 Data Analytics Class Python Bootcamp NYC**

***Note: This Documentation is also for Python User!(All codes are written in python)***

**Pandas Overview**

Pandas is an open source software library built on python language used for data analysis and manipulating data. The name pandas is gotten from the term “Panel data”. This library contains functions used for manipulating data frames, numerical data, csv files, time series, sheets, etc.

1. **Installing Pandas**

The following lines of codes installs the pandas package. Open the terminal and Enter.

*CODE: pip install pandas*

**Note*:*** To install the package through anaconda platform.

*CODE: conda install pandas*

1. **Importing Pandas**

In order to use the Pandas package, on your console, it has **to** be imported first with the code below.

*CODE: #import pandas as pd*

**Note:** **The primary two components of pandas are the Series and Data Frame.**

1. **Creating a Data Frame**

A data frame is a labeled data structure with columns of potential different types

*CODE: data = {'state': ['Ujjain', 'Ujjain', 'Ujjain', 'Prayag', 'Prayag', 'Prayag'],*

*'year': [2000, 2001, 2002, 2001, 2002, 2003],*

*'pop': [1.5, 1.7, 3.6, 2.4, 2.9, 3.2]}*

*frame = pd.DataFrame(data)L*

1. **Creating a Series**

A series is a like a list of items.

The code below creates a pandas series

*CODE: obj = pd.Series(np.arange(4.), index=['a', 'b', 'c', 'd'])*

*print(obj)*

1. **Indexing or selecting columns or rows**

There are two ways of indexing a column using pandas. They are the implicit and explicit method which both refers to the iloc and loc method respectively.

1. **Iloc:** This involves using the index of the item to locate it.

*CODE:* *frame.iloc[[2]]*

1. **Loc:** This involves using the item directly to locate the item

*CODE: frame.loc[['three']]*

1. **Lambda functions**

Lambda functions are inline functions which defines and used a function on the same line.

*CODE:* *frame.apply(lambda x: x.max(), axis =1)*

1. **Pandas Methods**

**Groupby:** This function is used to split the data into groups based on some criteria.

*CODE: grouped = df['data1'].groupby(df['key1'])*

*print(grouped)*

**Read csv**: This method is used to read data from a csv file into a dataframe

*CODE:* *df = pd.read\_csv('Womens Clothing E-Commerce Reviews.csv')*

**Unstack:** This unstacks the index of a multiple index dataframe

*CODE: data.unstack()*

**Other methods.**

* Isnull(): Gives the number of null values in the dataframe
* Notnull():Gives the number of not null values in the dataframe
* Dropna(): removes all null values in the dataframe based on the selected axis(1, 0)
* Fillna(): fills missing values with values defined by user
* Melt(): this rearranges the dataframe depending on the columns selected.
* Pivot(): this rearranges the dataframe depending on the columns and syntax.
* Merge(): This combine two different dataframes together.

1. **Methods for Time Series.**

Time series are data frames involving data in the form of data time format.

1. **Creating a time series**

*CODE:* *from datetime import datetime, date, time*

*dt = datetime(2011, 10, 29, 20, 30, 21)*

1. **strftime():** This reads the time series data in the specified format
2. **Method under time series**

* Dt.day: This extract the day value in the time data
* Dt.second:This also extracts the second values in the time data

1. **Shift():** This rearrange the data frame by moving the time values by one step.

These are majority of the function covered during the course of the course. Please note that there are also several other methods for this library that has not been covered in this material due to word constraint

**REFERENCES**

1. Data Science 101 Data Analytics Class Python Pandas Bootcamp – Pandas Methods-<https://www.udemy.com/course/data-science-101-class-python-pandas-bootcamp-course-nyc-new-york/>
2. NumPy Fundamentals (Python Library for Data Science) – Numpy Methods - <https://www.udemy.com/course/python-numpy-fundamentals/>
3. Geeks for Geeks - General python method explained - <https://www.geeksforgeeks.org/reshape-a-pandas-dataframe-using-stackunstack-and-melt-method/>