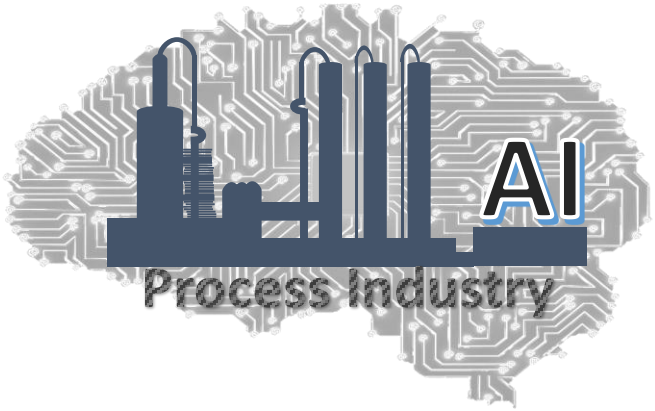
Statistical Techniques for Monitoring Industrial Processes



Lecture : Scientific Computing Package: NumPy

Module : Python Installation and Basics

Course TOC

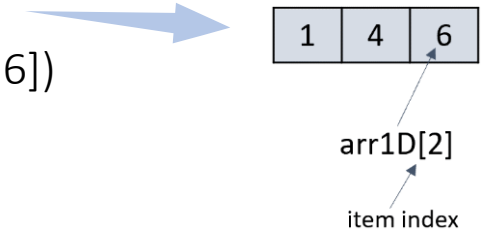
- ❑ Introduction to Statistical Process Monitoring (SPM)
- ❑ Python Installation and basics (optional)
  - ➢ Development environment, Scientific computing packages
- ❑ Univariate SPM & Control Charts
  - ➢ Shewhart Charts
  - ➢ CUSUM Charts
  - ➢ EWMA Charts
- ❑ Multivariate SPM
  - ➢ Fault detection using Principal Component Analysis (PCA)
  - ➢ Fault detection using Partial Least Squares (PLS) regression
  - ➢ Fault diagnosis using PCA/PLS contribution charts
  - ➢ Strategies for handling nonlinear, dynamic, multimode systems
- ❑ Deploying SPM solutions

NumPy for Grid-like / Tabular Data

- Industrial process data analysis entails working with data arranged in 2D or 3D tabular format
  - *Python list is not convenient for scientific computing with such data formats*
  - *NumPy facilitates efficient numerical operations on multidimensional arrays/matrices*

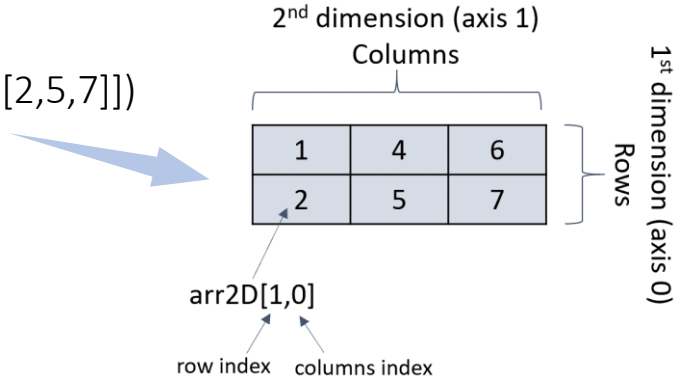
1D array

```
# create a 1D array
import numpy as np
arr1D = np.array([1,4,6])
```



2D array

```
# create a 2D NumPy array
import numpy as np
arr2D = np.array([[1,4,6],[2,5,7]])
```



Creating NumPy Arrays

- Converting list to array

```
arr = np.array([1,4,6])
```

- Creating sequence of numbers

```
arr = np.arange(3, 6) # same as Python range function; results in array([3,4,5])
```

- Creating special arrays

```
arr = np.ones((2,1)) # array of shape (2,1) with all items as 1
```

- Loading data from files

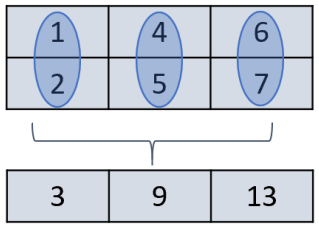
```
arr = np.loadtxt('file.csv, delimiter=",", skiprows=1) # skips the heading row
```

NumPy Functions

- NumPy provides several useful functions like *mean, sum, sort,* etc., to analyze arrays
  - *can specify dimension along which data needs to be analyzed*

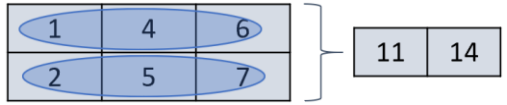
Summing up values along the rows of a 2D array

arr2D.sum(axis=0) # returns 1D array with 3 items



Summing up values along the columns of a 2D array

arr2D.sum(axis=1) # returns 1D array with 2 items



arr2D.sum() returns the sum over the whole array, i.e., 25

Logical Operations using NumPy Arrays

array1 ➡

5	4	3
2	1	0

array2 ➡

0	1	0
2	1	0

➤ `np.logical_or(array1>3, array2==1)` → array([[True, True, False], [False, False, True]])

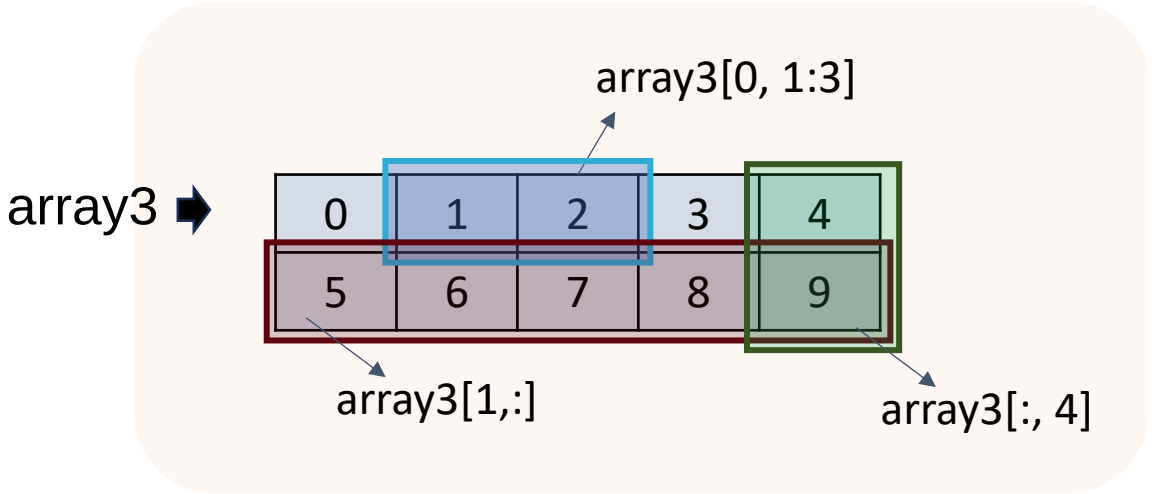
➤ `np.logical_and(array1>3, array2==1)` → array([[False, True, False], [False, False, False]])

➤ `np. any(array2 > 0)` → True

➤ `np. all(array2 > 0)` → False

Indexing and Slicing Arrays

- Works similar to what we saw for Python lists



Working with Views versus Copies

Changes made on array slices may change original array!

```
# extract a subarray from array3 and modify it

array3_sub = array3[:, :2] # columns 0 and 1 from all rows
array3_sub[1,1] = 1000

print(array3) # array3 gets modified as well!!

>>> [[   0    1    2    3    4]
       [   5 1000    7    8    9]]
```

Can lead to unexpected results!

use numpy.copy method for a separate copy of array or sliced array

Statistical Techniques for Monitoring Industrial Processes

Next Lecture : Scientific Computing Package: Pandas

Module : Python Installation and Basics