



# **Artificial Intelligence (Machine Learning & Deep Learning) [Course]**

**Week 2 – Numpy & Pandas**

**[See examples / code in GitHub code repository]**

**It is not about Theory, it is 20% Theory and 80% Practical –  
Technical/Development/Programming [Mostly Python based]**

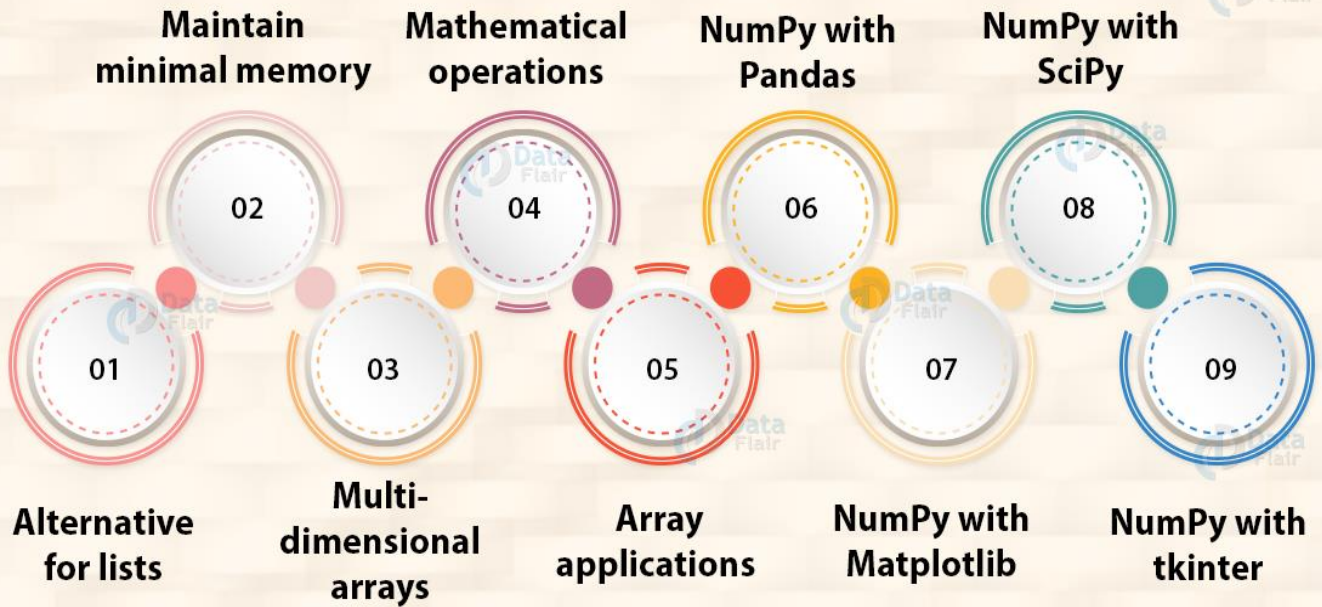
# Numpy - Overview



NumPy is a Python library.  
NumPy is used for working with arrays.  
NumPy is short for "Numerical Python".



## Applications of NumPy



25



# Numpy – Overview 2



Application-specific

cesium    PyChrono    MDAnalysis    eht-imaging    iris  
khmer    PsychoPy    Qiime2    FiPy    deepchem  
nibabel    mne-python    yellowbrick    scikit-HEP  
PyWavelets    librosa    SunPy    QuTIP    yt

Domain-specific

Astropy    Biopython    NLTK  
Astronomy    Biology    Linguistics  
QuantEcon    cantera    simpeg  
Economics    Chemistry    Geophysics

Technique-specific

scikit-learn    scikit-image  
Machine learning    Image processing  
pandas, statsmodels    NetworkX  
Statistics    Network analysis

Foundation

SciPy    Matplotlib  
Algorithms    Plots

Python    NumPy    IPython / Jupyter  
Language    Arrays    Interactive environments

New array implementations

NumPy API — Array Protocols - - -



# python

## PYTHON FOR DATA SCIENCE CHEAT SHEET

### Python NumPy

#### What Is NumPy?

A library consisting of multidimensional array objects and a collection of routines for processing those arrays.

#### Why NumPy?

Mathematical and logical operations on arrays can be performed. Also provides high performance.

#### Import Convention

import numpy as np - **import numpy**

#### ND Array

Space efficient multi-dimensional array, which provides vectorized arithmetic operations.

#### Creating Array

- `a=np.array([1,2,3])`
- `b=np.array([[1,2,3,4],[7,8,9,10]],dtype=int)`

### Saving and Loading

On disk:

- `np.save("new_array",x)`
- `np.load("new_array.npy")`

Text/CSV files:

- `np.loadtxt('New_file.txt')` - From a text file
- `np.genfromtxt('New_file.csv',delimiter=',')` - From a CSV file
- `np.savetxt('New_file.txt',arr,delimiter=',')` - Writes to a text file
- `np.savetxt('New_file.csv',arr,delimiter=',')` - Writes to a CSV file

Properties:

- `array.size` - Returns number of elements in array
- `array.shape` - Returns dimensions of array(rows, columns)
- `array.dtype` - Returns type of elements in array

### Operations

Copying:

- `np.copy(array)` - Copies array to new memory array.
- `view(dtype)` - Creates view of array elements with type dtype

Sorting:

- `array.sort()` - Sorts array
- `array.sort(axis=0)` - Sorts specific axis of array
- `array.reshape(2,3)` - Reshapes array to 2 rows, 3 columns without changing data.

### Array Mathematics

Arithmetic Operations:

- **Addition:** `np.add(a,b)`
- **Subtraction:** `np.subtract(a,b)`
- **Multiplication:** `np.multiply(a,b)`
- **Division:** `np.divide(a,b)`
- **Exponentiation:** `np.exp(a)`
- **Square Root:** `np.sqrt(b)`

Comparison:

- **Element-wise:** `a==b`
- **Array-wise:** `np.array_equal(a,b)`

### Functions

- **Array-wise Sum:** `a.sum()`
- **Array-wise min value:** `a.min()`
- **Array row max value:** `a.max(axis=0)`
- **Mean:** `a.mean()`
- **Median:** `a.median()`

- Learn from industry experts and be sought-after by the industry!

- Learn any technology, show exemplary skills and have an unmatched career!

## Python NumPy Cheat Sheet

<https://intellipaat.com/blog/tutorial/python-tutorial/numpy-cheat-sheet/>

<https://intellipaat.com/blog/wp-content/uploads/2022/10/Python-Numpy-Cheat-Sheet-2022.pdf>

<https://www.datacamp.com/cheat-sheet/numpy-cheat-sheet-data-analysis-in-python>





See code here: <https://github.com/ShahzadSarwar10/FULLSTACK-WITH-AI-BOOTCAMP-B1-MonToFri-2.5Month-Explorer/blob/main/Week2/Case2-1-NumPy-Zameencom-property-data-By-Kaggle.py>

You should be able to analyze – each code statement, you should be able to see trace information – at each step of debugging. “DEBUGGING IS BEST STRATEGY TO LEARN A LANGUAGE.” So debug code files, line by line, analyze the values of variable – changing at each code statement. BEST STRATEGY TO LEARN DEEP.

Let's put best efforts.

Thanks.

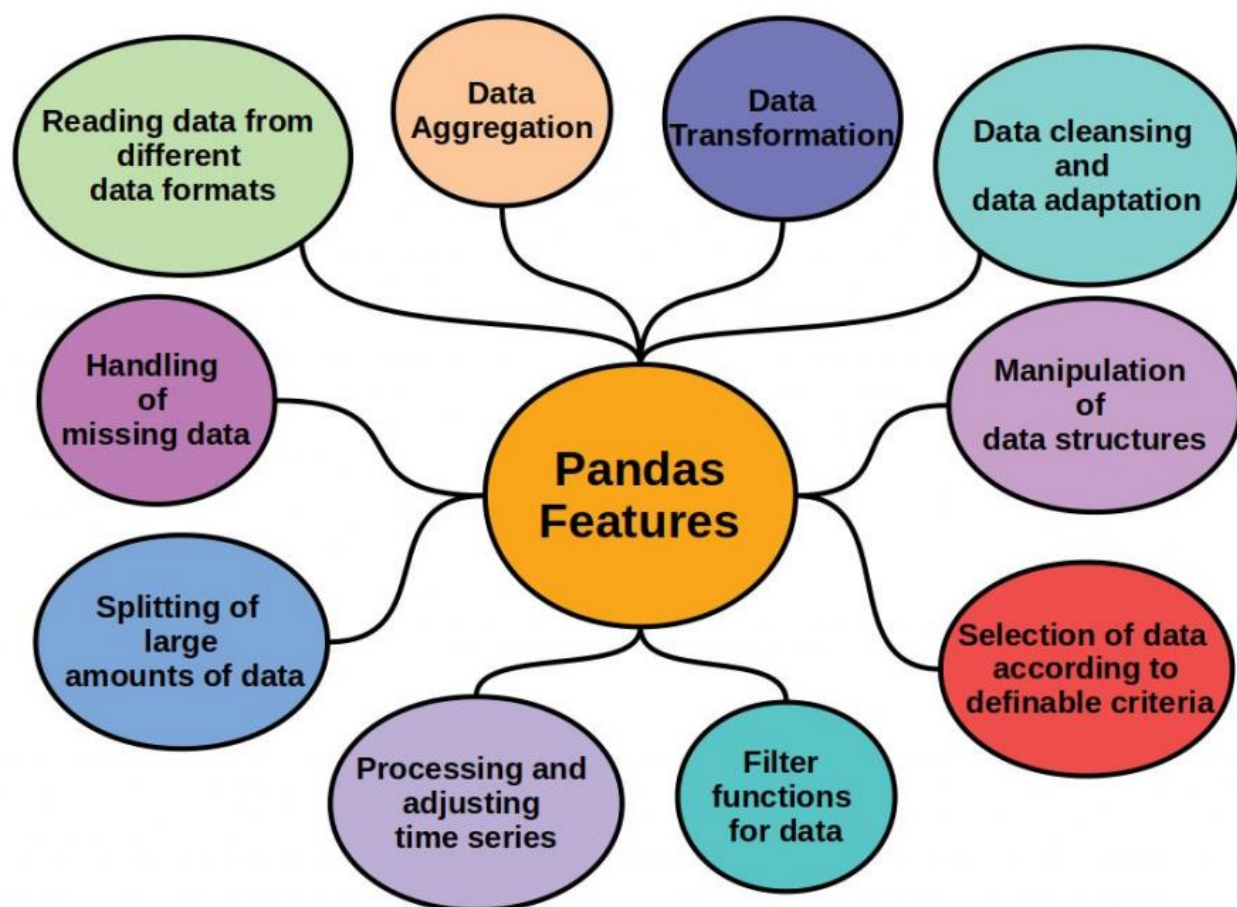
Shahzad – Your AI – ML Instructor

25

## Exercises



- ❑ Pandas is a Python library used for working with data sets.
- ❑ It has functions for analyzing, cleaning, exploring, and manipulating data.
- ❑ The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.



# python



## Applications of Pandas



### Reference:

<https://data-flair.training/blogs/applications-of-pandas/>

25

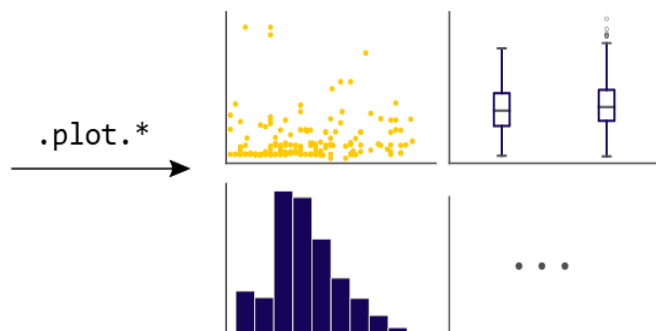
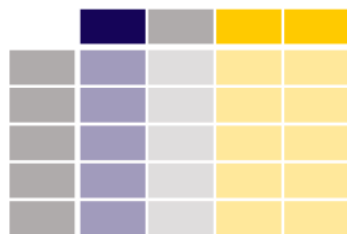
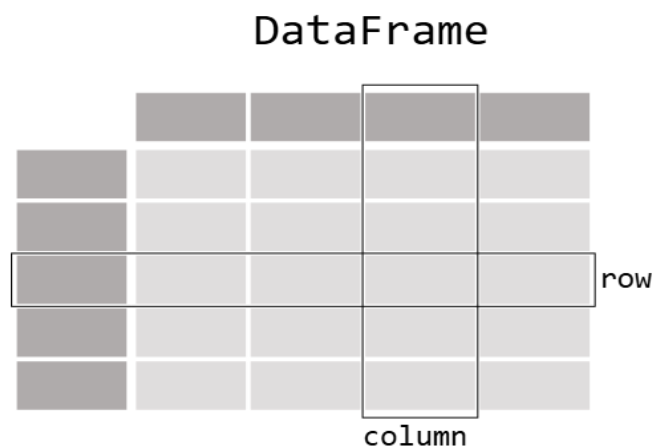


# Pandas – Key Concepts



❑ A data table is called a DataFrame.

❑ Pandas provides plotting your data out of the box, using the power of Matplotlib.



**DATA TYPE**

**STRENGTHS**

**AMOUNT OF DATA  
BEING ANALYZED**

**DATA TYPES**

**MEMORY USAGE**

**SPEED**

pandas

Tabular

Data frame, Series

>500K rows

Can contain dissimilar  
data types

More

Slower

NumPy

Numerical

Arrays

<50K rows

Homogenous  
data types

Less

Faster





# Pandas – Key Concepts

dataframe

| x    | y    |
|------|------|
| 12.3 | ace  |
| 3    | tea  |
| 5.01 | oil  |
| 2.3  | tree |

matrix

|      |     |
|------|-----|
| 12.3 | 0.1 |
| 3.0  | 5.2 |
| 5.01 | 3.0 |
| 2.3  | 0.1 |

list

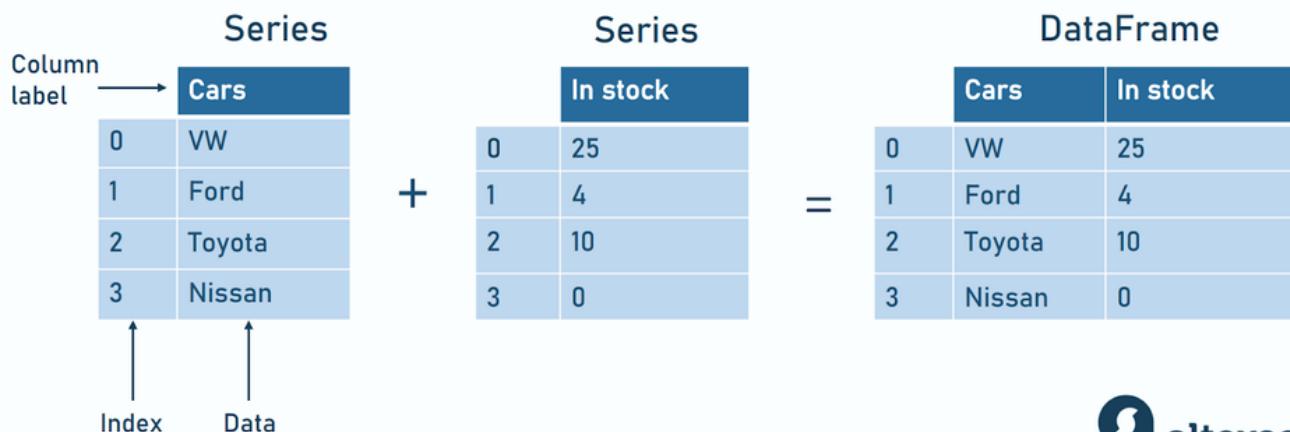
| x              | y    |
|----------------|------|
| 12.3           | ace  |
| 3              | tea  |
| 5.01           | oil  |
| 2.3            | tree |
| 3              |      |
| $Y \sim x - 1$ |      |
| some text      |      |

| Characteristics | NumPy Array   | Pandas Dataframe   |
|-----------------|---|--|
| Homogeneity     | Arrays consist of only homogeneous elements (elements of same data type)                | Dataframes have heterogeneous elements.                                  |
| Mutability      | Arrays are mutable  | Dataframes are mutable   |
| Access          | Array elements can be accessed using integer positions.                                 | Dataframes can be accessed using both integer position as well as index. |
| Flexibility     | Arrays do not have flexibility to deal with dynamic data sequence and mixed data types. | Dataframes have that flexibility.  |
| Data type       | Array deals with numerical data.  | Dataframes deal with tabular data.                                       |



| Pandas dtype  | Python type  | NumPy type   | Usage  |
|---------------|--------------|--|--|
| object        | str or mixed | string_, unicode_, mixed types                                 | Text or mixed numeric and non-numeric values |
| int64         | int          | int_, int8, int16, int32, int64, uint8, uint16, uint32, uint64 | Integer numbers                              |
| float64       | float        | float_, float16, float32, float64                              | Floating point numbers                       |
| bool          | bool         | bool_  | True/False values                            |
| datetime64    | NA           | datetime64[ns]   | Date and time values                         |
| timedelta[ns] | NA           | NA   | Differences between two datetimes            |
| category      | NA           | NA   | Finite list of text values                   |

## DATA STRUCTURES IN PANDAS



## Creating DataFrames

|   | a | b | c  |
|---|---|---|----|
| 1 | 4 | 7 | 10 |
| 2 | 5 | 8 | 11 |
| 3 | 6 | 9 | 12 |

```
df = pd.DataFrame(  
    {"a" : [4, 5, 6],  
     "b" : [7, 8, 9],  
     "c" : [10, 11, 12]},  
    index = [1, 2, 3])  
Specify values for each column.
```

```
df = pd.DataFrame(  
    [[4, 7, 10],  
     [5, 8, 11],  
     [6, 9, 12]],  
    index=[1, 2, 3],  
    columns=['a', 'b', 'c'])  
Specify values for each row.
```

|   | a | b | c |
|---|---|---|---|
| N | v |   |   |

## Reshaping Data – Change layout, sorting, reindexing, renaming



`pd.melt(df)`  
Gather columns into rows.



`df.pivot(columns='var', values='val')`  
Spread rows into columns.



`pd.concat([df1, df2])`  
Append rows of DataFrames



`pd.concat([df1, df2], axis=1)`  
Append columns of DataFrames

`df.sort_values('mpg')`  
Order rows by values of a column (low to high).

`df.sort_values('mpg', ascending=False)`  
Order rows by values of a column (high to low).

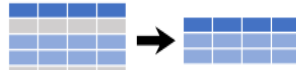
`df.rename(columns = {'y':'year'})`  
Rename the columns of a DataFrame

`df.sort_index()`  
Sort the index of a DataFrame

`df.reset_index()`  
Reset index of DataFrame to row numbers, moving index to columns.

`df.drop(columns=['Length', 'Height'])`  
Drop columns from DataFrame

### Subset Observations - rows



### Subset Variables - columns



### Subsets - rows and columns

Use `df.loc[]` and `df.iloc[]` to select only rows, only columns or both.

## Python Pandas Cheat Sheet

<https://www.datasciencecentral.com/data-science-in-python-pandas-cheat-sheet/>

[https://pandas.pydata.org/Pandas\\_Cheat\\_Sheet.pdf](https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf)

<https://www.datacamp.com/cheat-sheet/pandas-cheat-sheet-data-wrangling-in-python>



See code here: <https://github.com/ShahzadSarwar10/FULLSTACK-WITH-AI-BOOTCAMP-B1-MonToFri-2.5Month-Explorer/blob/main/Week2/Case2-2-Pandas-Zameencom-property-data-By-Kaggle.py>

You should be able to analyze – each code statement, you should be able to see trace information – at each step of debugging. “DEBUGGING IS BEST STRATEGY TO LEARN A LANGUAGE.” So debug code files, line by line, analyze the values of variable – changing at each code statement. BEST STRATEGY TO LEARN DEEP.

Let's put best efforts.

Thanks.

Shahzad – Your AI – ML Instructor

25

## Exercises







Thank you - for listening and participating

- ☐ Questions / Queries
- ☐ Suggestions/Recommendation
- ☐ Ideas.....?

Shahzad Sarwar  
Cognitive Convergence

<https://cognitiveconvergence.com>  
[shahzad@cognitiveconvergence.com](mailto:shahzad@cognitiveconvergence.com)

voice: +1 4242530744 (USA) +92-3004762901 (Pak)