

# Python Data Science

**Course Duration: 2-day; Instructor-led**

Time Schedule: 9am-5pm

Lunch: 1:00pm- 2:00pm

Morning Tea-break: 10:30am-10:45am

Afternoon Tea-break: 3:30pm -3:45pm

**WHAT YOU WILL LEARN**

This course includes the fundamental python programming techniques such as lambdas, reading and manipulating csv files, and the numpy library. The course also introduces data manipulation and cleaning techniques using the popular python pandas data science library and introduce the abstraction of the Series and Data Frame as the central data structures for data analysis, along with tutorials on how to use functions such as group by, merge, and pivot tables effectively. By the end of this course, participants will be able to take tabular data, clean it, manipulate it, and run basic inferential statistical analyses.

**AUDIENCE**

This course is designed for those already using Korn shell scripting, Perl, or C programming languages to manipulate files or control processes

**PREREQUISITES**

There are no prerequisites for this course but python knowledge with a little programming background is preferred.

**METHODOLOGY**

This program will be conducted through Instructor-led (classroom)

**COURSE OBJECTIVES**

After completing this course, you should be able to:

- Explore Python fundamentals, including basic syntax, variables, and types
- Create and manipulate regular Python lists
- Use functions and import packages
- Build Numpy arrays, and perform interesting calculations
- Create and customize plots on real data
- Supercharge with control flow, and get to know the Pandas DataFrame
- Use Python to read and write files
- Illustrate Supervised Learning Algorithms
- Identify and recognize machine learning algorithms around us

## COURSE OUTLINE

### Module 1: Introduction to NumPy

- Understanding Data Types in Python
- Creating Arrays from Python Lists
- Creating Arrays from Scratch
- NumPy Standard Data Types
- The Basics of NumPy Arrays
- NumPy Array Attributes
- Array Indexing: Accessing Single Elements
- Array Slicing: Accessing Subarrays
- Reshaping of Arrays
- Array Concatenation and Splitting

### Module 2: Data Manipulation with Pandas

- Installing and Using Pandas
- Introducing Pandas Objects
- The Pandas Series Object
- The Pandas DataFrame Object
- The Pandas Index Object
- Data Indexing and Selection
- Data Selection in Series
- Data Selection in DataFrame
- Operating on Data in Pandas
- Ufuncs: Index Preservation
- UFuncs: Index Alignment
- Ufuncs: Operations Between DataFrame and Series
- Handling Missing Data
- Trade-Offs in Missing Data Conventions
- Missing Data in Pandas
- Operating on Null Values
- Hierarchical Indexing
- A Multiply Indexed Series
- Methods of MultiIndex Creation
- Indexing and Slicing a MultiIndex
- Rearranging Multi-Indices
- Data Aggregations on Multi-Indices
- Combining Datasets: Concat and Append
- Recall: Concatenation of NumPy Arrays
- Simple Concatenation with `pd.concat`
- Combining Datasets: Merge and Join
- Relational Algebra
- Categories of Joins
- Specification of the Merge Key
- Specifying Set Arithmetic for Joins
- Overlapping Column Names: The `suffixes` Keyword

- Example: US States Data
- Aggregation and Grouping
- Planets Data
- Simple Aggregation in Pandas
- GroupBy: Split, Apply, Combine
- Pivot Tables
- Motivating Pivot Tables
- Pivot Tables by Hand
- Pivot Table Syntax
- Example: Birthrate Data
- Vectorized String Operations
- Introducing Pandas String Operations
- Tables of Pandas String Methods
- Example: Recipe Database
- Working with Time Series
- Dates and Times in Python
- Pandas Time Series: Indexing by Time
- Pandas Time Series Data Structures
- Frequencies and Offsets
- Resampling, Shifting, and Windowing
- Where to Learn More
- Example: Visualizing Seattle Bicycle Counts
- DataFrame.query() Method

### **Visualization with Matplotlib**

- Simple Line Plots
- Saving Figures to File
- Adjusting the Plot: Line Colors and Styles
- Adjusting the Plot: Axes Limits
- Labeling Plots
- Simple Scatter Plots
- Scatter
- Pie
- Histograms
- Multiple Subplots
- plt.axes: Subplots by Hand
- plt.subplot: Simple Grids of Subplots
- Geographic Data with Basemap