



Python for Data Munging ,Data Analysis and Data Visualization

- by Nitish Vishwakarma

Python libraries included in this course :

1. Pandas
2. Numpy
3. Matplotlib
4. Seaborn
5. IO
6. Time
7. OS

Pandas



This course will be delivered with following methods and approach:

1. **(EC)** - Exercise with Code (Code will be shared after every class on Google Docs/Github)
2. **(DR)** - Understanding of Concepts of Python through drawing explanations on online meeting tools (Gotomeeting /Zoom/MS-Paint).
3. **(XL)**- Excel Spreadsheet with operation performed(Mock/Actual) containing formulas (when required)
4. **(LINK)**- Links for understanding concepts in detail as homework on shared Google Drive/Sheet (when required)

Syllabus/Curriculum

1. What's Pandas/Numpy/Matplotlib/Seaborn for?
 - a. Comparison with SQL and other languages(SAS/R if required) and syntax comparison with code exercise
2. How does pandas fit into the data science toolkit?
 - a. Discussion on Data Science Primer
3. Pandas First Steps : Install and import
4. Core components of pandas:
 - a. Series
 - b. DataFrames
 - c. Date Range
 - d. Categories
 - e. Numpy
5. Creating DataFrames from scratch
 - a. Use of functions to create populate data and imputations:



- b. Linspace
 - c. Arrange
 - d. Random
 - e. Range
 - f. How to read in data (many files from computer)
6. Reading data from CSVs, Excel with delimiter options
7. Reading data from JSON with Web Scraping background
8. Reading data from a SQL database with Data pivots
9. Converting back to a CSV, JSON, or SQL, Excel with a small assignment.
10. Viewing your data Visuals and Deep-Dive summaries (with multiple ways of doing same operations)
 - a. Reading headers
 - b. Reading a particular column
 - c. Reading a particular column with limit of rows
 - d. Reading a multiple columns
 - e. Reading a specific row
 - f. Reading a multiple specific rows
 - g. Reading a specific location (row, column) using iloc, loc, iat
 - h. Iterating through the data frame using iterrows()
 - i. Iterating through the data frame and specifying columns
 - j. Filtering rows based on value of a column using loc
 - k. Describing the data frame for the count, mean, and other factors
 - l. Sorting values
 - m. Sorting multiple values with ascending and descending for each
11. Getting info about your data (in terms of EDA) :
 - a. Data types and composition
 - b. Filter Numeric, float and objects type variables
 - c. Null Values
 - d. Size
12. Handling duplicates (Techniques and Optimization comparison with SQL)
13. Column cleanup
14. How to work with missing values
 - a. Missing Value Imputation by mean, median
 - b. Detecting Missing value visualization through Charts
 - c. Row and Column imputation for missing Values
 - d. Interpolation for missing values
15. Removing null values
16. Understanding your variables (including numpy array)
 - a. DataFrame slicing, selecting, extracting
 - b. By column
 - c. By rows
17. Conditional selections for Insights
 - a. If else
 - b. Where



- c. Query Method
- 18. Applying functions (Lambda, Map etc)
- 19. Brief Plotting (Matplotlib , Dataframe plot, Seaborn library)
- 20. Joining DatFrames(merge, concat ,joins and their type)
- 21. Making Structural Changes in Data
 - a. a. Add a column
 - b. b. Dropping(Delete) a column
 - c. c. Equating a new column values of other columns using iloc
 - d. d. Rearranging Total column to nth column
- 22. Regex Filtering (filter based on textual patterns)

Above exercises (Practice with Code on Jupyter Notebook) are performed and taught in live session along with explanation of below concepts (explanation on Drawing tool such as Gotomeeting or MS Paint)

PYTHON Data Analyst Programming Course Structure(Concepts) :

1. A basic on Python Language and Install Python ([Jupyter Notebook](#))
2. Learn how to use functions to take an input and transform it into some output
3. Explore syntax error messages and troubleshoot basic Python code
4. Editing and starting Python scripts
5. Opening a text file (notepad file)
6. Reading text,CSV, JSON files
7. Raw (binary) data
8. Syntax of function definition([keyword arguments \(*args\)](#) and [positional arguments\(**kwargs\)](#))
9. OOPs Concepts , Inheritance in python
10. Creating single line & Multi Line Comments
11. Overview About Variables
12. Output/Print Variables ([ultimate guide to print\(\) object](#))
13. Python Numbers (**Numpy, Series**)
14. Python Variables
15. The range() function - ([Detailed with more arguments and taking best use of it](#))
16. Array types ([Multiple Dimensions](#))
17. Dictionaries ([Creation method Interview perspective](#))
18. Global versus local variables (DR)
19. Passing parameters and returning values (DR)
20. String Handling (EC)
21. String types: ([Normal and special values](#))
22. String operators and expressions
23. Math operators and expressions
24. Variables Type Conversion (to int ,float,str)
25. Python variable Casting



26. Python Strings
27. String Functions
28. String Formatting
29. Python Operators
 - a. Arithmetic operators
 - b. Assignment operators (for creating new arrays and series)
 - c. Comparison operators ([with Exercise Code and Notes](#))
 - d. Logical operators
 - e. Identity operators
 - f. Membership operators
 - g. Bitwise operators
30. Python Collections (Arrays)
31. Lists in Python
32. list operations
33. list methods
34. Module Tests
35. Tuple in Python
36. Tuple methods
37. Sets in Python
38. Sets Methods & functions
39. Functions and recursive functions
40. Conditional Statements (If, If-else, Nested If-else, And, OR etc)
41. Python Loops (while loops, for loops)
42. Break & Continue parameters

43. Python Functions
44. Passing a List as a Parameter
45. Default Parameters
46. Recursion (Basic)
47. File Handling ([Read & Write](#))
48. Exception Handling ([Drawing explanation](#))
49. Modules and Packages ([Regression Basics](#))
50. Lambda Expressions & Functions
51. List Comprehension ([Application on Exploring Data and Data Viz.](#))
52. Sequence or Collections in PYTHON ([Iterator and Iterables,Enumerate, Iterrows](#))
53. Python Dictionary([Dictionary Comprehension](#))
54. Python Functions using variables
55. Using the for statement for Dataframe,Plotting and Exploratory Data Analysis
56. The sorted() function
57. Dealing with syntax errors and notebook shortcuts
58. Handling exceptions with try/except
59. Cleaning up final code
60. Final Project on real world data & solution discussions



- 61. Alternate ways of handling day to day business queries
- 62. Interview Preparation & Grooming

Data Visualization with Python Data Visualization Library:

1. Matplotlib library
 - ❖ Intro to pyplot
 - ❖ Formatting the style of your plot
 - ❖ Plotting with keyword strings
 - ❖ Plotting with categorical variables
 - ❖ Controlling line properties
 - ❖ Working with multiple figures and axes
 - ❖ Annotating text

2. Seaborn library
 - ❖ Installing Seaborn.
 - ❖ Importing libraries and dataset.
 - ❖ Seaborn's plotting functions.
 - ❖ Scatter Plot
 - ❖ Customizing with Matplotlib.
 - ❖ The role of Pandas.
 - ❖ Box Plot
 - ❖ Seaborn themes.
 - ❖ Violin Plot
 - ❖ Color palettes.
 - ❖ Swarm Plot
 - ❖ Overlaying plots.
 - ❖ Heatmap
 - ❖ Histogram
 - ❖ Bar Plot
 - ❖ Factor Plot
 - ❖ Density Plot
 - ❖ Joint Distribution Plot