# Python for Citi Quantitative Analysts – 2.5 Days

### This course is designed for analysts from a technical or mathematics background who will used Python to manage a wide range of datasets and to implement & test trading strategies. The course covers the following topics

### A review of Python’s data analysis and data manipulation features

### Citi’s DataFlame environment and Python best practices

### Importing and charting large datasets

### Working with timeseries data in Python

### Python’s programming tools and building prediction algorithms

### Back-testing and tuning strategies

Each topic combines lecture, exercise and a worked reference solution using financial market data.

The course also includes a team-based **Citi Data Challenge**, an open-ended “mini-projects” where the analysts apply their new skills to complete real-world data analysis and programming tasks.

### Pre-Requisites

* Programming experience, preferably with Python, or with a language such as R or C++.
* Experience of working with large datasets and relational databases.

### Objectives

**At the end of this course, attendees will be able to:**

* Use Citi’s Dataflame environment and be familiar with Python Configuration Management in Citi
* Understand where and when to use NumPy and Pandas
* Visualize data with clearly formatted charts
* Work with multiple data sources including historical tick-data, daily closing prices across multiple asset classes, corporate actions, sentiment analysis datasets and real-time data-streams
* Implement, test and tune a trading strategy in Python

### Content - Day 1

**Introduction - Data Science, Data Analysis & Your Role in Citi**

* Data Science Ice-Breaker
* What do we mean by data science and data analysis
* Working with the volume, complexity and range of data in Citi
* Citi’s DataFlame Environment
* Course Overview

**Module 1 – Overview of Python & Review**

* Overview of the main python language features that will be used throughout the course
* Functions, Lists, Dictionaries, Strings
* Unit Testing and debugging
* The main Python Modules for numerical programming – NumPy, Pandas, matplotlib and quandl

**Lab 1 – Python Scripting**

* A simple step-by-step programming task to introduce the development environment.

**Module 2 – Python Best Practices & Source Code Configuration**

* Build and deployment.
* Modules
* Command Line arguments
* Debugging tips
* Testing existing python applications

**Lab 2 – Configuration and Debugging**

* Creating your own workspace with a set of pre-prepared python utilities
* Adding a new python module.
* Build, test and debug the entire suite.
* One of the modules will be seeded with a defect which needs to be debugged.

### CONTENT - Day 2

**Module 3 – Importing data, data visualization and Timeseries Data in Python**

* Importing Data
* from Text Files.
* from Spreadsheets
* from Databases
* Data Visualization
* Line properties
* Working with Multiple axes
* Adding Text to a chart
* Logarithmic and nonlinear axes
* Line graphs, block graphs, pie charts
* Timeseries Data using quandl

**Lab 3 – NASDAQ Market Data Presentation**

Create a python application to import a security dataset based with combination of

* symbol
* date range
* opening & closing prices
* intra-day high and low prices
* volume

The application should size and shape the data to produce a professional rendering of the information in standard graphs and charts.

**Module 4 – Programming tools and techniques for prediction algorithms**

* The Template Design Pattern
* The Strategy Design Pattern
* Moving Averages
* Volatility
* Market Signals
* Prediction algorithms for categorical and regression data

**Lab 4 – Trading Strategy Metrics**

Participants will be provided with several datasets and asked to create a Python application to

* Import the datasets
* Generate a series of measures and metrics used to assess the performance of trading algorithms

**Module 5 – Back-testing, Validating and Tuning strategies**

* The 4 components of a Back-test
* Strategy
* Portfolio
* Data Handler
* Execution Handler
* Evaluating a Strategy
* Standard Measurements
* Sharpe Ratio
* Drawdowns
* Cumulative Returns
* Compound annual growth rate (CAGR)

**Lab 5 – Back-test a Simple Trading Strategy**

Participants will be given a simple trading strategy written in Python along with a sample dataset in CSV format.

They will be asked to

1. Write a Python application which back-tests the strategy
2. Produce metrics which demonstrate the performance of the strategy

### Day 3 – Citi Data Challenges

In this day-long programming challenge participants will design and develop their own trading strategy.

They will need to prepare a sales pitch and demonstration for their trading strategy covering

1. The market signals the strategy is designed to take advantage off
2. Key design elements of the strategy – prediction model and back-testing approach
3. How to assess its performance
4. Its actual performance.