Open-Source Technologies for Internal Audit

Ameya Abhyankar

FinQuest Institute

FinQuest Institute
Pursuing Excellence. Ensuring Success.

EKSPERT CONSULTING
RISK & ANALYTICS CONSULTANTS

Agenda

- Introduction
- What is Python
- Versions of Python
- Key Python Libraries
- Introduction to Anaconda
- Introduction to a few Python IDEs
- Introduction of IPython Notebook
- Growing role of Python in data analytics
- Brief on Analytics and big data
- Use cases for Internal Audit
- How we can help

Introduction – Ameya P Abhyankar

- **Current roles:**
  - **FinQuest Institute, Founder and Lead Trainer (started operations in Oct'20)**
    - ✓ Training programs for Finance, Quantitative Analytics and Programming
    - ✓ Training services for candidates appearing for the FRM examinations (GARP Institute, USA)
    - ✓ Career Counselling services for students and working professionals
  - **Ekspert Consulting, Founder (started operations in Dec'21)**
    - ✓ Risk Analytics
    - ✓ Model development/Model validation
    - ✓ Financial Products valuation
    - ✓ Process optimization
    - ✓ Technology implementations for industry applications
- **Previous work experience (12 + years)**
  - ✓ **Industry** : IndusInd Bank; **Consulting** : Nomura Research, Deloitte; **Software:** Wipro Technologies
- **Academic and professional qualifications**
  - ✓ Chartered Financial Analyst (CFA), CFA Institute, USA
  - ✓ Certificate in Quantitative Finance (CQF), CQF Institute, UK
  - ✓ Certificate in Machine Learning for Finance: MLI Institute, UK
  - ✓ MBA (Finance), University of Pune
  - ✓ Bachelors of Engineering, University of Pune

Introduction - Ameya P Abhyankar

- **Academic Associations:**
    - ✓ **Visiting Faculty (7+ years)** at GIPE, Pune; Welingkar Institute of Management , Meghnad Desai Academy and ISME in Mumbai
    - ✓ **Board Member** – Academic Board (Industry expert in Finance) @Atlas SkillTech University, Mumbai
    - ✓ Experienced in design/enhancement on academic curriculum to be in sync with industry requirements

- Publish articles regularly on various topics in Quantitative Finance, Capital Markets and Python for finance

What is Python

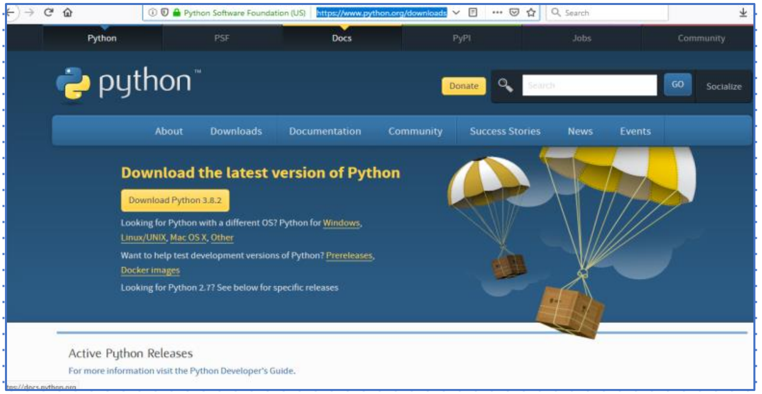
- Python is an **open-source programming language** which is extremely versatile, making it highly appealing to most industry domains, research as well as for academia.
- Python is supported by a very active user community and support forums which help all categories of users be it : beginner, intermediate or an expert
- A few key **features** of Python:
  - ✓ Open-source
  - ✓ Extensive set of libraries
  - ✓ Multipurpose programming tool
  - ✓ Operating System independent
  - ✓ Code that is easy to read and understand
  - ✓ A very supportive user community
  - ✓ New libraries get added very frequently to the Python ecosystem

What is Python (contd.)

- Python is widely used for: rapid application development, web development, supports scaling of applications from simpler applications to high end data intensive analytics tasks
- Globally organizations are investing heavily in technology with focus on building/enhancing their tech infrastructure with Python being in the forefront
- A few uses of Python in practical applications:
  - ✓ Quantitative Model development/Validations
  - ✓ Automation/streamlining of financial/ regulatory reporting
  - ✓ Forecasting/estimation of quantities
  - ✓ Classification of quantities
  - ✓ Big Data analytics
  - ✓ Trading
  - ✓ Risk Management
  - ✓ Parallel processing for advanced applications
  - ✓ Statistical testing

Versions of Python

- The first version of Python was released in 1991.
- Subsequent years saw further releases which were enhancements over the previous versions
- Most legacy systems are built using Python 2.x
- New application development in Python is done on Python 3.x
- Can be downloaded from: https://www.python.org/downloads/
- Compatibility among versions
- Slight changes in syntax for a few functions. For example:
  - ✓ Python 2.x: print "Hello World"
  - ✓ Python 3.x: print ("Hello World")



Download the latest version of Python

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

Active Python Releases

For more information visit the Python Developer's Guide.

Anaconda

- Anaconda is a package manager developed and supported by Continuum Analytics
- Its open source and free for use of the Python and R data sciences community
- Contains various libraries and packages for use in application development
- It allows us to launch applications provided in the Anaconda distribution and easily manage conda packages, environments and channels without the use of command-line commands
- It provides a working environment which is used for scientific computing, data science, statistical analysis and machine learning.
- Easily integrates with various python IDEs and supports seamless development process
- It's a memory intensive application, so lighter package managers like 'miniconda' are also used by some developers

Popular IDEs

Popular Python Integrated Development Environments (IDEs): A few are enlisted below

- PyCharm
- Spyder
- Jupyter Notebook
- SublimeText
- VS Code

PyCharm

The Python IDE
for Professional Developers

Selecting suitable IDE for development?

Key Python libraries

- **Numpy:** Short for Numerical python. This library supports multidimensional array object for handling data; provides functions/methods to operate on the array of data
- **SciPy:** Short for Scientific python. This library contains most of the functions required for scientific applications including numerical root finding, interpolation etc.
- **Pandas:** Powerful library for management and analysis of data frames, time series etc.
- **Matplotlib:** plotting and visualization library in python for visual analysis of data
- **Seaborn:** Enhanced data visualization and charting

Machine Learning libraries:

- **Scikit-learn:** Powerful library with functions/methods supporting advanced analytics and machine learning applications
- **TensorFlow/Keras:** Libraries for advanced analytics giving methods supporting the development of artificial neural networks

Growing role of Python in the industry

- Python caters to a huge spectrum of users: students, researchers, software engineers, data analysts etc.
- Its an easy to learn language and implement. Most of the times a complex looking equation can be directly written as a code.
- Organizations globally are increasing their investment in technology, Python is on the forefront of this investment.
- Many functionalities are readily available via libraries in Python. Therefore, majority of business use cases can be solved by calling Python libraries *rather than developing code from scratch*. This saves considerable time and development effort.
- In the era of Big Data, Python is a language of choice of most organizations given its rich set of libraries and a very supportive user community.
- There is a significant focus on *data aggregation & visualization*. Python has multiple libraries supporting such tasks.
- Python integrates very well with databases, so handling data becomes convenient

What is big data

- By definition, big data means all the potentially useful information that is generated in the economy/markets etc.

- Big data flows from of various sources including:

  ✓ **Traditional Sources:**

  a. Financial Markets

  b. Financial reports of companies

  c. Economic statistics published by authorities etc.

  ✓ **Non-traditional sources:**

  a. Individual actions like online social media posts, online reviews, emails, website hits

  b. *Corporate exhaust data* that is generated by various businesses like bank records, retail tag scanner data etc.

  c. *IoT data:* Sensors like RFID chips embedded on smart phones etc. also generate huge data. The broad network of such devices is called IoT.

Characteristics of big data

- **Velocity:**

Velocity is the rate at which data flows into the systems. They are classified as:

a. *Low Latency data* [eg: stock price feeds etc.]

b. *High Latency data* [eg: periodical but infrequent announcements etc.]

- **Variety:**

This property explains the myriad forms in which data flows.

a. Structured [eg: spreadsheets ; Databases]

b. Semi-structured [eg: photos , web page code]

c. Unstructured [eg: Video]

Use cases for Internal Audit

- Ability of open-source tools to work on **entire population** of data and **not just sample** of data
- The below examples have been demonstrated through simple hands-on exercises on Jupyter Notebook

- **Example 1:** A review of fixed fees charged by banks to clients.

o Banks define a fixed fee to be charged to clients based on the size of the client's business

o Internal Audit (IA) is expected to review if correct fees have been charged to clients as per the fees defined at an Organization wide level

o For this example:

a. Inputs:

✓ Fees Limit structure

✓ Data dump of the fees charged per client

b. Output:

✓ A report showing if incorrect fees have been applied for a certain client so that IA can highlight the same to business

Use cases for Internal Audit

- **Example 2:** Position reconciliation & Exposure reconciliation

- Back Office performs a daily operation of position reconciliation
- Risk Management is expected to perform a daily exposure monitoring for each counterparty
- For this example:

a. Inputs:

✓ Dump of trade-wise and counterparty-wise Exposure

b. Output:

✓ Report giving a recon of the number of trades and trade types per counterparty

✓ Reconciliation of Exposure numbers per counterparty

Use cases for Internal Audit

- **Example 3:** Reading tables given in a **.pdf** file and directly consuming it for analysis

- A lot of data is made available in the form of a .pdf file such as - organization policy defined counterparty limits; financial statements in annual reports; accounting/financial analysis tables done as an internal exercise etc.
- *NAV data for Security Receipts / VC investments etc.* are provided by Finance that risk management uses for capital charge and other applications. This is typically in .pdf format which exposes us to an **operational risk** that punching numbers from .pdf to Excel can cause mistakes which will impact subsequent computations
- Copying of such data into a readily useful format in Excel may not be always possible OR copied data is in a format that is not readable
- With Python we can read tables off .pdf file and write them in an Excel file in just a few lines of code
- For this example:

a. Inputs:

✓ .pdf file giving limits for a counterparty

b. Output:

✓ Limits converted into a table in Excel for ready usage by downstream systems

Use cases for Internal Audit

- **Example 4:** Review of forecasting models

○ Business frequently uses models for supporting estimation/forecasting of certain quantities (eg: inventories)

○ Most business attempt to use a JIT logic for inventory management

○ Normally, they either use a *Regression* OR *Time series analysis* for this purpose

○ Business may find it useful if IA could review such models for correctness

○ For this example:

a. Inputs:

✓ Sales data

b. Output:

✓ ARIMA Model for estimation of the sales in the next period

Use cases for Internal Audit

- **Example 5:** Review of reports created by Asset-Liability management (ALM) teams in banks
  - ALM teams are expected to calculate key regulatory reports like Liquidity Coverage Reports (LCR) , Net Stable Funding Ratio (NSFR) etc. on a daily basis
  - Incorrect/erroneous reports can have regulatory repercussions
  - Most of the time data for such reports comes from a myriad of systems for example, Calypso/Murex, Finacle/FlexCube, Back Office systems, risk/valuations team numbers etc.
  - There is a significant amount of operational risk owing to the distributed data being manually used.
  - IA may have a system in place to review key numbers that flow into the liquidity risk calculations.

- **Example 6:** Review of **RBS data points** to be submitted to the banking regulator each quarter

- **Example 7:** Tally the **collateral amount** as provided by the Back Office vis-à-vis those used by risk management for its reporting

Use cases for Internal Audit

- **Example 8:** Those firms that do not have an enterprise-wide system, and is currently relying on Excel for its entire computations, may transition to a Python - Enterprise Database model

o Assume a firm has legacy data on MS Excel, the same may be moved to open-source databases (eg: SQLite) and all subsequent analysis and reports can be generated via Python

o Moving data onto a database enables the firm to pursue advanced analytics moving further

o Centrally stored data makes it easy to retrieve the required information easily

In general, from the point of view of Internal Audit, the following items may be considered as an immensely value adding activity:

a. Optimization of activities done by Internal Audit

b. Automation of recurrent activities thereby eliminating chances of operational errors

c. Better visualization and reporting capabilities

d. Flexibility to design analytics / reports as required by the regulator OR industry best practices

How can we help?

- Personally, I have 7+ years of experience in hands-on Python programming and have used it for various applications in the past:
  - Model development (pricing, risk measurement, stress testing etc.)
  - Validation
  - Reporting
  - Automation etc.

- Having spent time in the banking and financial services industry and interacted with statutory, concurrent and internal auditors, we have a fair amount of knowledge of the regulatory landscape to support IA teams to streamline/optimize their processes by leveraging technology [We can also help **IA adopt AI**] ☺

- Open-source technologies like Python have gained traction and are here to stay. Firms are using these technologies for building a robust framework for core business processes and reporting