An Introduction to Python

Pritam Dalal

Beyond Spreadsheets

- 99% of data analysis in finance occurs in Excel.
- pros:
  - visual and tactile
  - it's nice to see the data
  - easy to spot check calculations
  - great for small-scale problems
- cons:
  - doesn't scale well
  - can introduce errors easily
  - lacks reproducibility
- Python (and R) remedy a lot of this - but spreadsheets are still useful.

Python In Brief

- Developed in the late 1980s as language simple enough to teach children about programming.
  - It turns out adults appreciate simplicity too.
- Python is free and open source.
- It's functionality has been greatly extended by *packages*
  - e.g. `pandas`, `numpy`
- During the 2000s, packages that are extremely useful for data analysis were developed: `IPython`, `numpy`, `pandas`.
- Today, Python is one of the de facto standards for computing in finance (and many other fields).

Python vs Competitors

- R and Matlab were designed explicitly for scientific computing and data analysis.
- Python was designed as a general purpose programming language and data analysis is just a small subset of its functionality.
- For example, you wouldn't build a website using R orv, but lot's of people build websites with Python.
- Python is more object-oriented (instances of classes, methods, etc) in nature. R is more functional in nature - pretty much everything is either a function or data.

What is a Distribution?

- In practice, any functioning Python setup consists of the base language, plus a collection of packages.
- One approach to setting up Python your machine is to install the language and then all the packages you want separately.
- This would be time consuming, and because of the open-source nature of Python, the packages might not play nice together.
- This is where a distribution comes in handy:
  - A *distribution* is the language -plus- a curated collection of packages.

The Anaconda Distribution

- The company Continuum Analytics bundles together Python and all the major science related packages into a distribution called *Anaconda*.
  - Free to use, but Continuum charges for support (*freemium*)
- In this class we will be using the *Anaconda*.
- The Anaconda packages are curated in two ways:
  - they are relevant to scientific computing
  - and they are all ensured to work together

Python 2 vs Python 3

- Bottom line: don't worry about this too much.
- A while back there was a major revision of the language (from 2 to 3).
- So now there is a rift in the Python world, but nowadays most packages are available in Python 3.
- In this course we will use Python 3, and I would recommend that you stick to 3 moving forward.

SciPy

- SciPy: a collection of packages related to scientific computing and data analysis.
  - NumPy: vector and matrix computations
  - SciPy: also a package, optimization
  - IPython: interactive wrapper around Python
  - Pandas: dataframes and timeseries
  - Matplotlib: data visualization
  - Jupyter: a notebook interface for IPython
- SciPy turns Python into a scientific computing framework much like R and Matlab.
- In this class, we are mainly going to use the SciPy ecosystem of packages for the purposes of financial data analysis.

Jupyter Notebook and Other IDEs

- In this class, we'll mainly be writing code in Jupyter Notebooks.
- Jupyter Notebook is the predominant IDE for data analysis in Python
- Other IDEs include PyCharm and VS Code (used in FM 5091).
- There are lots of different alternative IDEs for Python. As compared to RStudio for R, none of the Python IDEs are as dominant.