Financial Modeling with Python and Excel
An Introduction

Nick DeRobertis[1]

[1]University of Florida
Department of Finance, Insurance, and Real Estate

September 1, 2020

Table of Contents

What is this Class?

- This is a skills-based course focused on teaching financial modeling techniques in Python and Excel

- The focus is not a lot of specific models, but rather general model-building techniques

- The focus will be simple models, but extending them in powerful ways

What is a Model?

Inputs $\longrightarrow$ Model $\longrightarrow$ Outputs

A Retirement Problem

A General Structure

Inputs → Process → Outputs

Real-world Problem

Wages, Savings → Investment → Cash in the bank, person retires

A Model of the Problem

Cash Flows, Savings Rate, Interest Rates → Model → FV of CF, time until retirement

Valuing a Company (DCF Model)

A General Structure

Inputs → Process → Outputs

Real-world Problem

Microsoft creates software → Sells the software → Generates cash for investors, stock reaches a price

A Model of the Problem

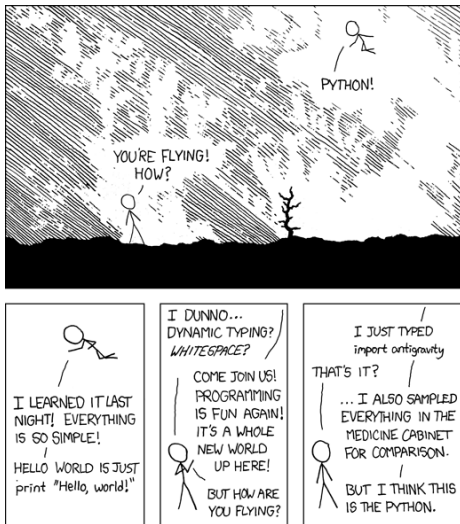Revenue, COGS, SG&A, growth rates, costs of capital, etc. → Model → Stock price, stock returns

Table of Contents

1 Introduction to Financial Modeling

2 Introduction to the Modeling Toolset
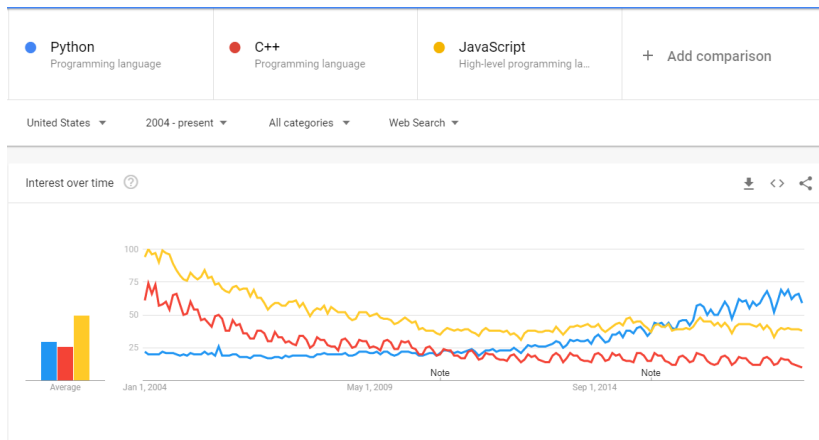
Why Python?

Python is a Good Choice for Finance

- The easiest to learn mainstream programming language

- Heavily used in the financial industry

Python is the Most Flexible of the Top Languages

- Modeling
- Data science
- Algorithmic Trading
- Scripting

- Devices
- Web Development
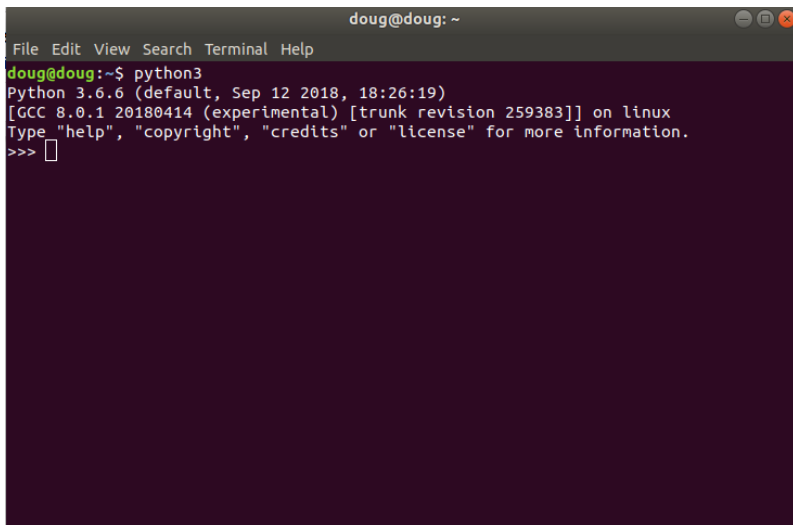- Web Scraping
- Even these slides

Python is the Fastest Growing Programming Lanugage

More Python Advantages

- Open source - completely free and open

- Focus on readability - almost pseudo-code

- Take as much as you need. Easy for beginners, many features for experts.

- Deep integrations with Excel - VBA replacement, run Python in Excel, run Excel from Python

Why Not use Python?

Python Disadvantages

- No graphical interface (by default)

- Hard to get others working with it if they don't know Python

- Can take more work to get started on a project

Why Not use VBA?


```
(General)

Sub Range_Find_Method_Row()
'Finds the last non-blank row on a sheet/range.

Dim lRow As Long

    On Error Resume Next
    lRow = Cells.Find(What:="*", _
                    After:=Range("A1"), _
                    LookAt:=xlPart, _
                    LookIn:=xlFormulas, _
                    SearchOrder:=xlByRows, _
                    SearchDirection:=xlPrevious, _
                    MatchCase:=False).Row
    On Error GoTo 0

    MsgBox "Last Row: " & lRow

End Sub
```

What does all this code mean???

VBA is Old-School

- Code is not as readable as Python

- Power is limited to working within Microsoft Office

- Python has a complete VBA API built into a package - Python can do VBA and more

We Can't Ditch Excel Yet

- Excel is everywhere. Most of the world's data is in Excel spreadsheets

- (Nearly) everyone knows how use it

- You can see what you're doing (without effort)

- Easy introspection into a particular value

Escaping Excel Hell

The Pains of Excel

- Code and view are mixed together, code is hidden

- Both cell formulas and VBA macros - What is going on?

- Easy to make mistakes (one cell different)

- Some tasks which are very simple in Python are very complex in Excel

Let's Get Python Set Up on your System

Install Steps

1. Go to https://www.anaconda.com/products/individual to download Python 3.8

2. Follow the steps in the installer

3. You will hit "Advanced Installation Options". It is very important that you select "Add Anaconda to my PATH environment variable". It says it is not recommended, and will highlight it in red when checked, but we will need it later in the course.

4. Open CMD (windows key, search cmd)

5. Type python and hit enter. You should see Python 3.8 and a >>> come up.

Make sure you have selected Python 3.8 and not 2.7