Getting Started with Python and Excel

Building a Basic Model in Both Excel and Python

Nick DeRobertis[1]

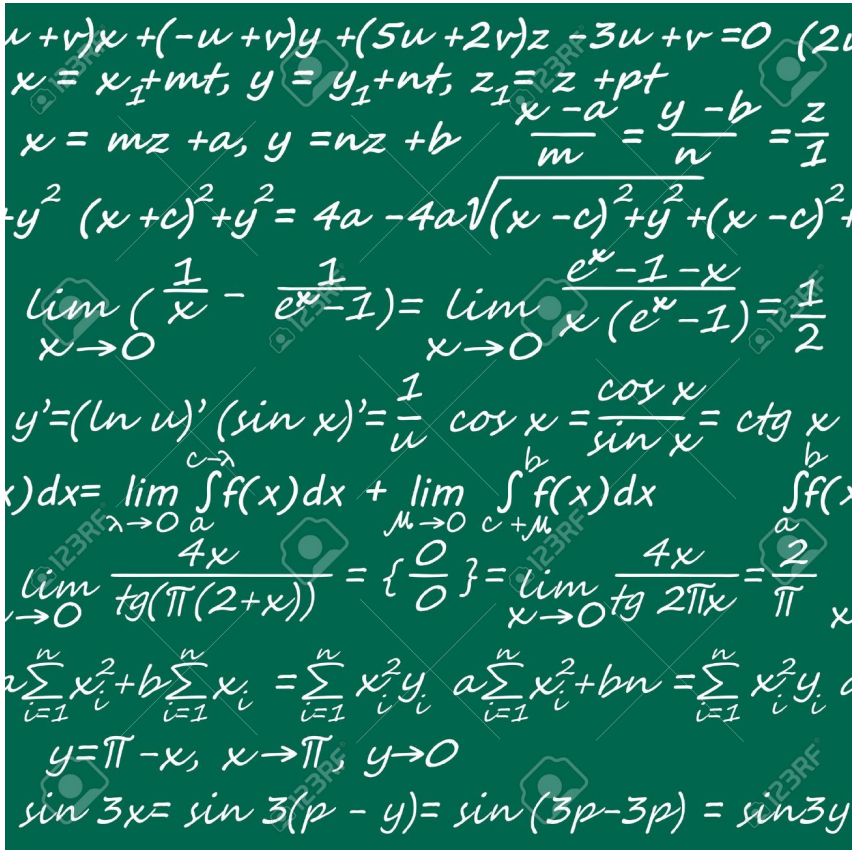
[1]University of Florida
Department of Finance, Insurance, and Real Estate

September 1, 2020

Table of Contents

Approaching a Problem with Two Tools

- The focus today is to get familiar working in both Excel and Python
- We will approach this by building a simple model with both tools
- In later lectures, we will move to combining the tools



A Simple Retirement Problem

- Let's take what is perhaps the simplest finance problem, which everyone should understand
- While you may have approached such a problem with a calculator before, we will build models for it instead
- Martha is saving for retirement. She earns $60,000 per year and is able to save 25% of that. If she invests her savings, earning 5% per year, and she needs $1,500,000 to retire, how soon can she retire?

Breaking Down the Retirement Problem

A General Structure

Real-world Problem

A Model of the Problem

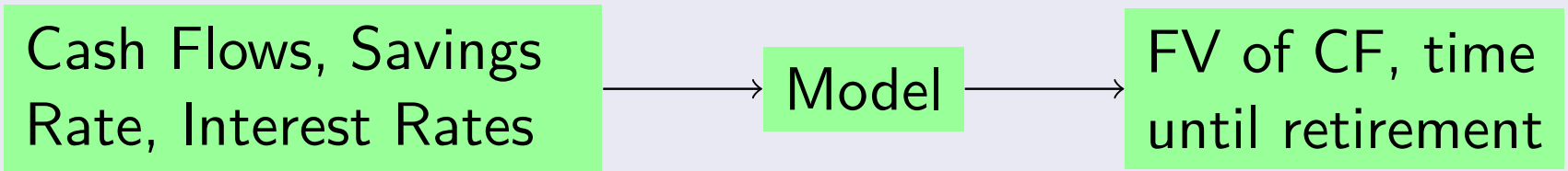


Table of Contents

Solving the Problem in Excel

- It is easy to use Excel as a calculator and just type the math in directly. But we want to build a model.
- Changing inputs should result in a change to outputs. The way to do this in Excel is cell references
- Fixed references become important when trying to drag formulas, e.g. `$A$2` (fully fixed), `$A2` (fixed on column), or `A$2` (fixed on row).

Simple Retirement Problem in Excel

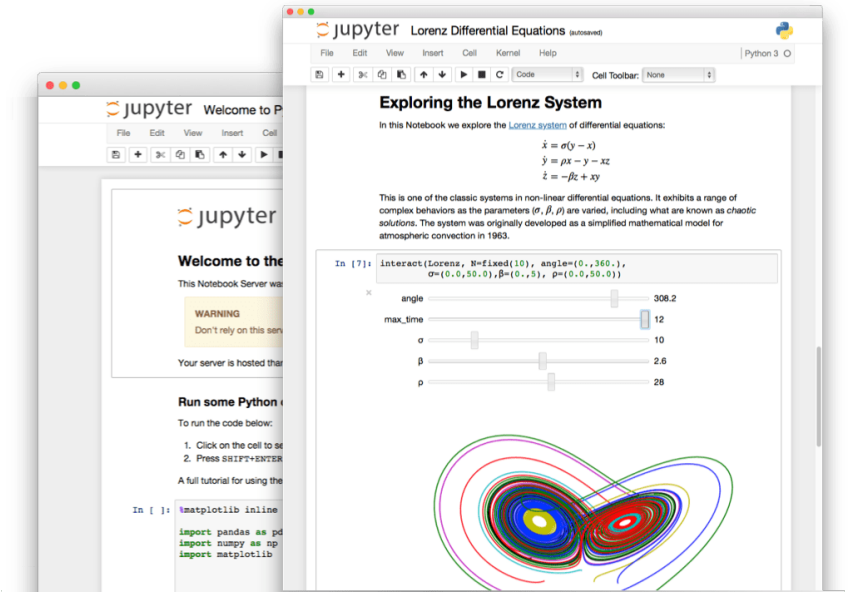
Intro Excel Exercise

- Go to Canvas and download Simple Retirement Model.xlsx from Examples > Intro > Excel
- Follow along as I recreate the simple model.

Table of Contents

How We'll Work in Python

- Using Python in the terminal is kind of a pain. And so, tools were born.
- Jupyter is a graphical interface we can use for Python. It also supports over 40 other languages such as R, SAS, Julia, and Scala
- You can use `jupyter notebook` or `jupyter lab`. The latter has a lot more features outside of the notebook



Let's Get Set up with Jupyter

Launch Jupyter Notebook

1. Launch Anaconda Navigator
2. Find Jupyter Notebook on the main screen, and click launch
3. You should see a list of folders and files. Click New and then Python 3
4. Now you should see a code cell with `In [ ]:` next to it

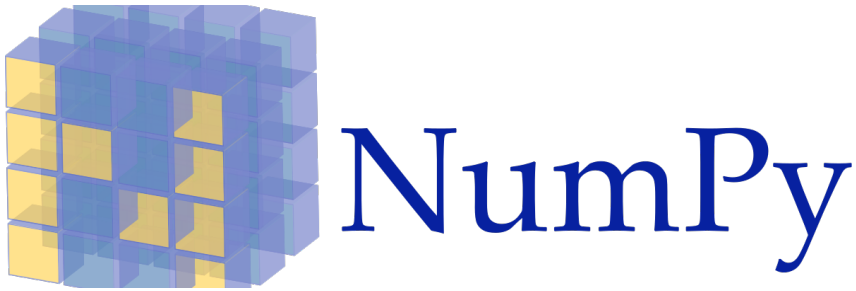
If you don't have Anaconda Navigator, just open a terminal (search `cmd` on Windows, `terminal` on Mac). Then in the terminal, type `jupyter notebook` and enter. Then continue with the third step.

Some Python Basics

- In Excel, the basic unit is a cell. In Python, the basic unit is an object.
- In Excel, content in a cell is either a number (123) or a string (ABC)
- In Python, all objects have types. They might also be a number or a string, or something else.
- Rather than using a cell reference like `$A$2`, we assign names to objects in Python
-

```
my_number = 6
my_string = 'ABC'
```

Doing Some Math in Python



- Basic operations in Python are straightforward
- `2 + 5 = 7`
- `6 - 2 = 4`
- `2 * 3 = 6`
- `5 / 2 = 2.5`
- A lot more is available using the `numpy` package
- `np.pv`, `np.nper`, `np.fv`, `np.pmt`
- All numpy financial functions

Simple Retirement Problem in Python

Intro Python Exercise

- Go to Canvas and download Simple Retirement Model.ipynb from Examples > Intro > Python
- In Jupyter, then navigate to your Downloads folder (or wherever you saved it)
- You should then see Simple Retirement Model.ipynb come up in the list of files in Jupyter. Click it to open it and follow along.

Table of Contents

Extending the Model - Multiple Interest Rates

- Now we've got basic models to determine how long it will take Martha to retire.
- We've got a few assumptions built into the model. One is that Martha will earn 5% on her investments
- Rates of return are volatile, so we want to see how long it would take her to retire if her return was different

Programming Fundamentals - Iteration

- In programming, for model building or otherwise, you often need to repeat the same process for multiple different things
- In Excel, you would do this by dragging formulas.
- In Python, as in most other programming languages, we would use a `for` loop
- This says, do something, for each value I pass into the loop

Iteration - Python vs. Excel

Python Iteration

```
inputs = [5, 10, 15]
for item in inputs:
    new_value = item + 2
    print(new_value)

7
12
17
```

Excel Iteration

Input	Output	Function
5	7	=B4+2
10	12	=B5+2
15	17	=B6+2

Explaining Python Iteration

- There's a few things to unpack here
- Here's another type of object: not a number or a string, but a `list`
- A `list` holds multiple objects, and you can add or remove items from `lists`

Python Iteration

```
inputs = [5, 10, 15]
for item in inputs:
    new_value = item + 2
    print(new_value)
```

```
7
12
17
```

Explaining Python Iteration (pt. 2)

- Here we define a list of three numbers as inputs
- Then we use a `for` loop to get each input out of the list, and add 2 to it to create the new value
- Finally we print each value as it is generated

Python Iteration

```
inputs = [5, 10, 15]
for item in inputs:
    new_value = item + 2
    print(new_value)
```

```
7
12
17
```

Iterating the Existing Model

Expanding on Python and Excel

- I will now expand the existing Excel and Python models to examine multiple interest rates
- Continue viewing the same previously downloaded files.

Putting it All Together - A Basic Model with Iteration

Let's Vary the Savings Rate, Too

- Now we want to see the effect of savings rate on time until retirement, in addition to interest rate
- In both Excel and Python, calculate the years to retirement for savings rates of 10%, 25%, and 40%, and each of these cases with each of the interest rate cases, 4%, 5%, and 6%
- Be sure that you drag formulas in Excel and use `for` loops in Python to accomplish this

Note

In total you should have 9 calculated years to retirement numbers, in each of the two models.