Advanced Financial Modeling

Nick DeRobertis

June 9, 2021

1 Introduction to Advanced Financial Modeling

- This lecture gives an overview of what was covered in the Financial Modeling with Python and Excel course and the types of topics that will be covered in the Advanced Financial Modeling with Python course
- If you have not completed the Financial Modeling with Python and Excel course, I would recommend doing that first, or at least looking through the topics to make sure you know them
- The Advanced Financial Modeling with Python course will assume knowledge of the topics in the basic course so that we can move more quickly
- It will take me a while to cover all the topics mentioned here, so I have provided external resources you can use to learn the topics in the meantime
- Please subscribe on YouTube to get notified about future course updates

2 Additional Types of Financial Models

- This lecture does not seek to give a comprehensive list of financial models, but instead highlights a selection of common models
- In future lectures, we will look at implementing some of these in Python
- Resources are listed for both Excel and Python models. The free Python resources I have been able to find outside this course do not have very good coding standards, though, so bring what you have learned from the Financial Modeling with Python and Excel course when looking at these

3 Data Pipelines for Financial Modeling

- Data pipelines are a very common application for automation in a business setting
- Models need data, and most models get updated over time with new data. Data pipelines are all about streamlining getting the data into your model
- First you must gather the data (data collection), then get it into the format ready to be consumed by your model (data cleaning/wrangling)
- In deciding whether to build an automated data pipeline or to manually complete the steps, consider how often the data will need to be updated and how long it takes you to complete the manual process
- There are additional advantages beyond time savings: elimination of mistakes and allowing new opportunities (e.g. historically the report is run weekly, now that it is automated it is run every hour)
- Web scraping, API access, and SQL databases will be where you get most of your data beyond Excel spreadsheets

- Data wrangling means cleaning the data of errors and unneeded values as well as reformatting it to be used in your model, such as combining multiple sources, aggregating, resampling, etc.
- Requests is typically used for basic HTML scraping, while Selenium is typically used for for web pages that use JavaScript to display the necessary data
- Helium is an easier way to use Selenium

4 Advanced Mathematical Tools for Financial Modeling

- We have already covered one library typically used for high-performance computation: numpy
- SciPy is the gold-standard for optimization in Python
- Sympy lets you do algebra and calculus with symbolic math, so it is great for working with lots of or very complex equations. You can convert the symbolic formula into a function directly
- fuzzywuzzy is a library that makes working Levenshtein distance very simple
- There are a lot more statistical models you can access through statsmodels beyond just OLS. There is also linearmodels for working with panel data and SciPy has some additional tools
- Scikit-Learn is the standard for high-level machine learning in Python, though its performance is limited
- Keras helps with building deep learning models and can be more performant than Scikit-Learn

5 Better Presentation of Python Financial Models

- Presentation from a Python model is a more complex story than doing the same with an Excel model. But you also have many more possibilities
- The easiest presentation method is to use the Jupyter notebook itself. You can offload code into separate .py files and just import it so that there is not much code in the notebook
- You can create reports from your Python model. You could output to Excel, HTML/CSS, LaTeX, or PDF to create the report.
- If you go with HTML/CSS, Jinja will be useful for templating. HTML/CSS can also be converted into a PDF with pdfkit
- If you go with LaTeX, pyexlatex allows you to create Python objects which compile into LaTeX and PDFs
- You can go direct to PDF with reportlab
- You can have the most polished presentation of your model by building an app, which is not as hard as it sounds. Voila can directly convert your Jupyter notebook into an app. Panel will let you create a more customized app, by adding onto your existing Jupyter notebook
- You can learn about building full web applications from scratch with Flask, but that is more complicated than the other approaches
- Interactive plots allow the consumer of the model to view more information in a single plot by adding some dynamic behavior to it. There are many libraries for this such as Bokeh, Altair, Plotly, and Holoviews

6 Programming Skills for Advanced Financial Models

- If you are going to take one suggestion from the general programming practices, make it version control. This is an essential skill in programming to be able to work with others. It will also open up options for yourself, giving greater freedom to tear apart the code and not worry about losing anything
- Automated testing also allows greater freedom as you are not worried about breaking existing functionality when adding new features of the model. pytest is a useful package to help
- Once you are offloading your code into separate .py files, it is useful to work with an IDE such as PyCharm or VS Code. In general, I recommend PyCharm if you are working only or primarily in Python and VS Code if you need a multi-language editor
- CI/CD allows you to automate the lifecycle of your code and remove all the manual actions you take to maintain it besides writing the code itself
- If you do go to working in an IDE, it is useful to start using type annotations as it will give super powers to your IDE to know exactly what you can and cannot do in the code, giving you suggestions and highlighting issues
- Experienced Python programmers use virtual environments religiously because they have been bitten in the past by package conflicts across projects. Once you are managing multiple Python projects, it becomes all but a necessity. They will allow you to keep the dependencies of each project isolated

7 Extra Resources for Python Financial Modeling

- Hitchhiker's Guide to Python goes through many general Python topics and also has sections dedicated to specific applications of Python
- Real Python regularly posts articles containing in-depth explanations and tutorials. It is generally targeted at intermediate Python programmers, and will give lots of detail
- Automate the Boring Stuff is a book that goes through general practical tasks you might want to automate and shows how to do it in Python along with explanations of the underlying code
- Practical Business Python is a similar article structure to Real Python but focuses more on common business tasks and is generally targeted at less-experienced programmers