Advanced Financial Modeling

A Road-map to Learning Advanced Financial Modeling Topics

Nick DeRobertis[1]

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

November 19, 2020

Table of Contents

What we Covered and What is Left

- Throughout this course, we have covered Python and Excel basics
- We have also covered financial modeling specifics, such as how to structure a financial model in both Python and Excel, cash flow and probability modeling, sensitivity analysis, scenario analysis, and Monte Carlo simulations
- As far as types of financial models, we covered a retirement model, a capital budgeting model, a lender profitiability model, and the discounted cash flow (DCF) valuation of a stock
- This could have been a two-semester course. There is a lot I didn't cover. Let's do a quick overview of it today.

Table of Contents

Financial Models 1

- Portfolio valuation and optimization: Find the returns, value, and risk of a portfolio and select the best asset allocation for the portfolio
- Additional Funds Needed (AFN): Budgeting model which uses forecasted financial statements to determine how much capital should be raised
- Lease or Own: Guides the decision of whether to rent or buy an asset
- Event Studies: Tries to determine the impact of an event
- Merger and Aquisition (M&A): A DCF valuation of a target company with operations being combined with the parent at a merger date. Detetermines M&A price.

Financial Models 2

- Leveraged Buyout (LBO): A specialized M&A model used for when large amounts of debt are being used to purchase the target firm.
- Derivatives Valuation: Value options, swaps, forwards, etc.
- Debt models: Immunization models are about having the right amount of cash in the future, term structure models estimate the rates for different maturity bonds, and default-adjusted return models factor in default in determining debt returns
- Value at Risk (VaR): Determine how much an investment might lose with a certain probability in a certain time span

Financial Model Resources

- Here are some links with free resources to learn more about types of models
- The following examples will be using Excel only
- Macabacus
- Corporate Finance Institute
- Financial Modeling Institute
- And I found a couple Python resources, though the coding standards are not great:
- Finance and Python
- Build a Trading Algorithm in Python

Table of Contents

- Data pipelines are about getting the input data into your model in a standardized way
- Within data pipelines, there are two main steps: data collection and data cleaning.
- Either step can be automated, ideally both would be, but it always comes down to a tradeoff with modeler time

Data Collection

- Data can come from many sources, but most typically it originates from the Internet
- If you can manually download the data, you can automate it via web scraping
- You can also extract different types of data with web scraping, even if it is not structured as a dataset
- Some data comes from an API, where you need to send requests to download the data
- `selenium` uses Python code to drive a browser such as Chrome while `requests` is a more lightweight, text based way to make web requests (good for APIs).

Data Cleaning

- We have already covered the best general-purpose tool for cleaning data: `pandas`
- We did not cover it in enough detail to cover all data cleaning cases
- The main material left is selecting, merging, grouping, and reshaping
- Regular expressions are a way of matching any possible string and extracting parts of it, which useful for messy data

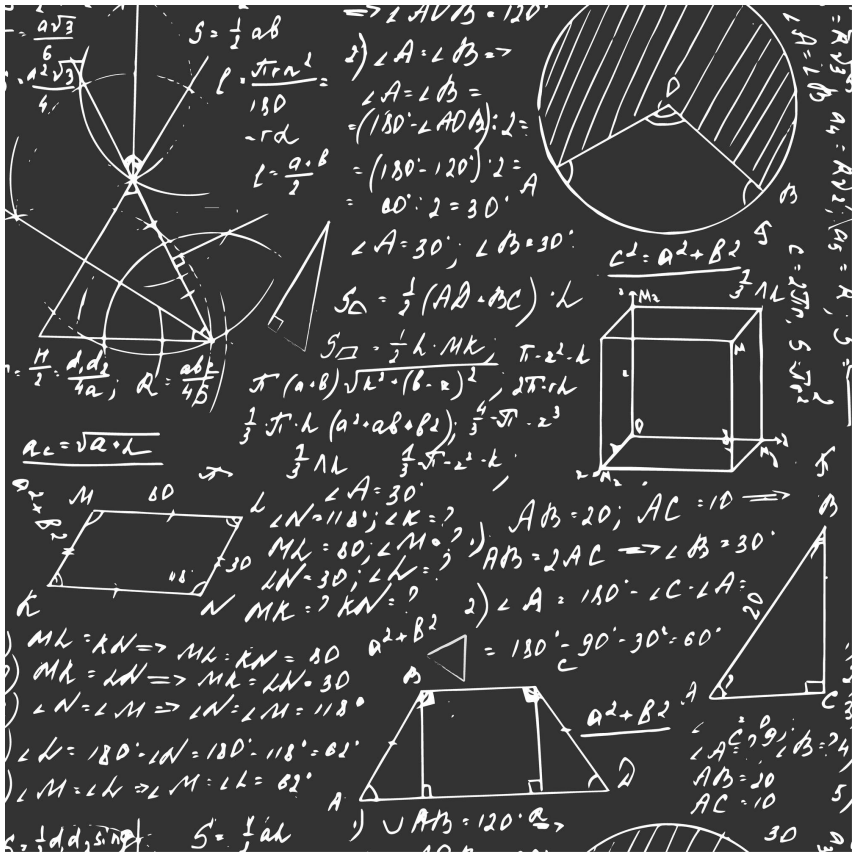
Data Pipelines Resources

- Get Started Browser-Based Web Scraping with Selenium
- Get Started Text-Based Web Scraping with Requests
- Pandas Intro, Covers Basics of Needed Topics
- Advanced Pandas
- Intro to Regular Expressions
- Regular Expression Reference

Table of Contents

What Math we Covered

- We covered basic mathematical tools including basic probability theory, algebra, variance/standard deviation, averages, and basic regressions
- Most financial modeling does not take very advanced math, with the exception of some derivatives models
- But there are a few more useful tools we didn't have time to cover



General Math Tools

- Often we want to maximize or minimize something, e.g. maximize returns or NPV, minimize risk. We can do this in general with a mathematical technique called optimization
- If you have a complicated custom model, such that the algebra is getting too difficult to do by hand, you can use a computer algebra system
- Levenshtein (edit) distance can be used to say how similar two strings are, which is useful for data cleaning and more.

Statistics Tools

- We covered Ordinary Least Squares (OLS) regressions, which is what anyone means if they just say regression
- There are many more kinds of regressions, we can't even mention them all here. But most likely to be useful include logistic regression for when probabilities are dependent variables and panel regression + fixed effects for when you are dealing with multiple instruments over time
- There are many time-series models to cover, as was mentioned in the forecasting lecture
- Machine learning/AI can be used to make classifications or predictions

Mathematical Tools Resources

- Optimization with SciPy
- Computer Algebra with SymPy
- Levenshtein Distance using fuzzywuzzy
- Intro to Logistic Regression in Python
- Intro to Panel Regression in Python
- Statistical Models in statsmodels
- More Statistical Models in linearmodels
- More Statistical Tools in Scipy
- Getting Started with Machine Learning in Python
- General Introduction to Machine Learning
- Deep Learning (AI) in Python using Keras

Table of Contents

Presenting Model Results

- We didn't have very much time to cover presenting the model in Python, other than formatting text
- For Excel, the model is already presented so just structure the workbook well
- A well structured Jupyter notebook or Python script is important for another modeler to pick up where you left off. But it still is not an ideal format for non-technical consumers of your model

Regional Visitor Experience

Hawaii Tourism 2017

U.S. East

Total Visitors: 2.31 Million

Vistors by Month

Places Visited

O'ahu Maui Kaua'i Kona Hilo Moloka'i Lāna'i

Likelihood to Return

For U.S. East respondents, 44.7 percent of first-time visitors said they would very likely return, compared to 40 percent in 2015. However, a slightly lower percentage of repeat Visitors (-1.7 points to 68.3%) said that they would very likely revisit.

Negative Experiences

Among this group of respondents, travel problems (12.1%) was their main issue, followed by nuisances (10.1%) and unfriendly people/poor service (7.9%).

Positive Experiences

A third of U.S. East respondents said the one detail that made their Hawai'i trip memorable was the events/celebrations with friends/family. A quarter of the respondents gave positive evaluation to their Hawai'i trip overall, while some mentioned Hawai'i's natural beauty and activities/attractions.

Experience Rating by Island

Experience Rating	HILO	KAUA'I	KONA	LĀNA'I	MAUI	MOLOKA'I	O'AHU	Overall
Excellent	80%	93%	88%	80%	91%	84%	78%	90%
Above Average	16%	6%	10%	13%	8%	11%	20%	9%
Below Average	3%	1%	2%	2%	1%	2%	2%	1%
Poor	1%	0%	1%	5%	0%	2%	0%	0%

Report Generated 8/23/2018 12:27:52 AM

Page 1 of 5

Present with Basic Jupyter

- One easy way that doesn't require anything we haven't learned is separating model logic and presentation
- Develop as you will while building the model. But when it's time to share the results, separate the main model logic (classes and functions) into separate Python file(s)
- Then the Jupyter notebook will just show high-level calls to your model and the results

Create Reports

- Less technical consumers of the model may just want the model conclusions and not to actually use the model themselves
- In this case, it is useful to generate reports containing the model results
- There are three major ways to make reports: HTML, LaTeX, and direct PDF solutions
- If all you need is PDF output, a direct solution is fine, but HTML and LaTeX (which can be converted to HTML) can both be converted to PDF and have the added advantage of being viewable as a web page
- Templating is also easier with HTML and LaTeX

Publish Reports and Models

- There is another way for non-technical consumers of your model to interact with it: via an app
- It is possible to build apps right in a Jupyter notebook using widgets
- A web app could also be created with a web framework
- With an app, a non-technical consumer of your model can adjust the inputs and see the outputs, without having any knowledge of Python or instructions

Advanced Plotting

- We covered very basic plots using `pandas`
- `matplotlib` provides all the plotting functionality to `pandas` and any `pandas` plots can be adjusted using `matplotlib` options
- Several libraries exist for interactive plots, which allow for dropdowns, sliders, selecting points, zooming, tooltips, and more
- Other plotting styles are becoming more popular, such as `holoviews`, in which you just describe the data and it can generate interactive plots for you

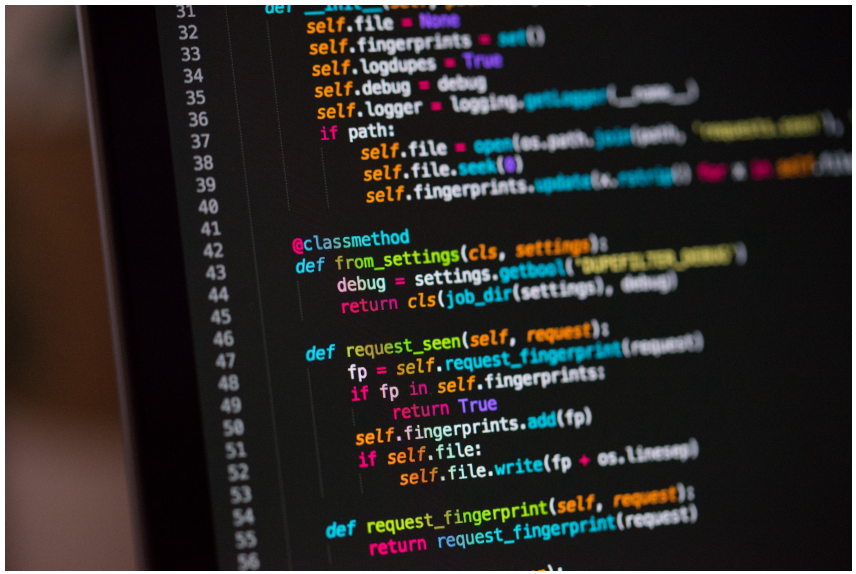
Presentation Resources

- Intro to Creating HTML Reports in Python
- Create HTML Reports and Output to PDF
- Templating HTML and LaTeX Using Jinja
- Direct PDF Output with Reportlab
- Direct to LaTeX Using pyexlatex
- Matplotlib Plotting Tutorial
- Get started with Holoviews
- Use ipywidgets to Build an App in Jupyter
- Convert a Jupyter Notebook to a Web App using Viola
- Examples of Viola Web Apps
- Get Started Building Web Apps with Flask
- Convert a Jupyter Notebook to a Web App using Anvil
- Full Tutorials for Python Web Development

Table of Contents

Your Programming Future

- We were only able to cover basic Python, and certainly not a lot of practices that would typically be used in programming
- Python is very flexible in that you can be productive in it with very little understanding or mastery of it
- But if you do gain a greater knowledge of it, it can unlock new potential.
- Further, there is more we can learn from the software engineering world in how to improve our programming



General Programming

- We can learn a few things from the software engineers that are programming all day every day
- Version control or source control is a way of tracking changes to code over time. You can get the full history of the project, and multiple people can work on the same project at the same time and later merge the changes together. `git` is the gold standard for this.
- Share `git` repositories with the world via Github (or others), allowing anyone to use and improve your project
- Automated testing of your code allows you to make changes ensuring it won't break anything
- Input validation is useful for when others interact with your model directly
- Use an integrated development environment (IDE) such as PyCharm or VS Code for better code completion and linking
- Automatically run tests, style your code, deploy Python packages or web apps, and more with continuous integration/continuous deployment (CI/CD)

Python Programming

- Python can work with the files on your computer through the `open` command and the `os` `pathlib` and `shutil` modules
- Get unique items, unions, intersections using sets
- Override how objects work by overriding double-underscore (dunder) class methods
- Modify existing functions using decorators
- Store Python objects by picking them
- Check yourself and make your code easier to understand using type annotations
- Structure projects using packages and modules
- Isolate project requirements using virtual environments

Programming Resources 1

- Get Started with Git and Github
- Github Desktop is an App that Makes Git Easy
- Get Started with Automated Testing in Python
- Intro to Input Validation in Python
- PyCharm IDE
- Visual Studio Code (VS Code) IDE
- Intro to CI/CD
- CI/CD Using Github Actions
- Intro to Reading and Writing Files with Python
- Beyond the Basics of Working with Files in Python

Programming Resources 2

- Work with Sets in Python
- Get Started with Dunder Methods in Python
- Advanced Dataclasses
- Rich Object Display in Jupyter
- Get Started with Python Decorators
- Store Python Objects with pickle and dill
- Intro to Type Annotations in Python
- Intro to Project Structure in Python
- Introduction to Virtual Environments in Python
- Virtual Environments Made Easy with pipenv

Table of Contents

General Resources

- The Hickhiker's Guide to Python
- Real Python
- Automate the Boring Stuff with Python
- Practical Business Python

Lecture Resources

Lecture Resources

1. Slides - Advanced Financial Modeling
2. Lecture Notes - Advanced Financial Modeling