lesson 12

Algorithmic Trading

Python for Financial Analysis

Rajah Chacko

elvtr

Syllabus Review

1. Introduction to Python: Python in Finance
2. Python Basic Syntax: Importing Libraries
3. Working with Pandas
4. Pandas Underneath the Hood: Working with NumPy
5. Data Wrangling and Visualization
6. Extracting Financial Insights from Charts and Graphs
7. Financial Calculations with Python: Part 1
8. Financial Calculations with Python: Part 2
9. CAPM and Portfolio Management
10. Linear Regression
11. Time Series Analysis
12. **Algorithmic Trading**

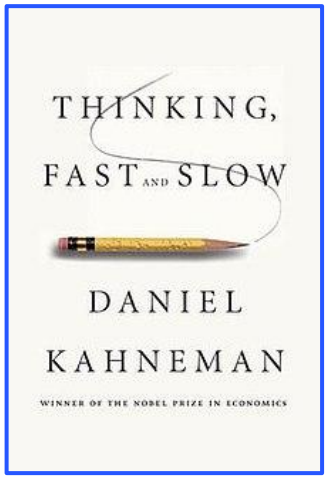
Bonus Class: Cryptocurrency Beyond the Basics with a Fintech Guest Speaker

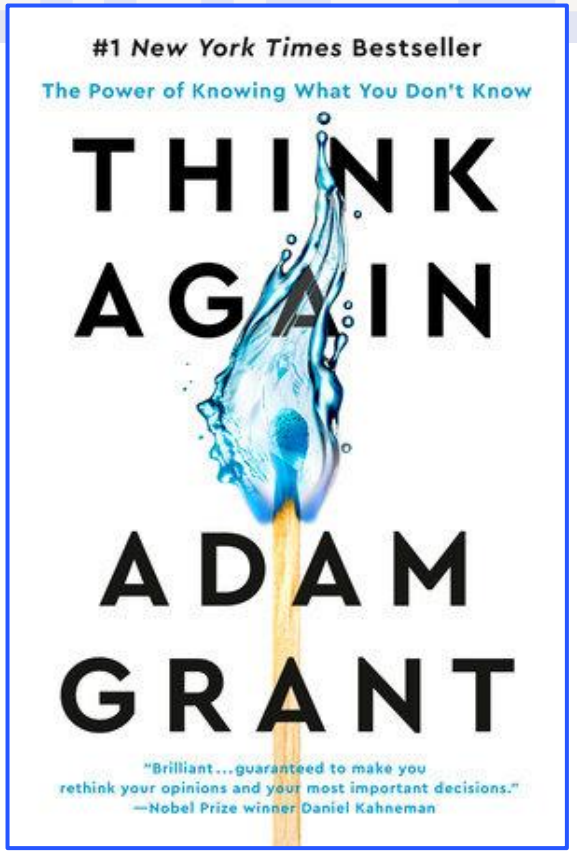
Class agenda

- Benefits of algorithmic trading
- Types of algorithmic trading strategies
- Risks and challenges of algorithmic trading
- Building an algorithmic trading strategy

Benefits of algorithmic trading

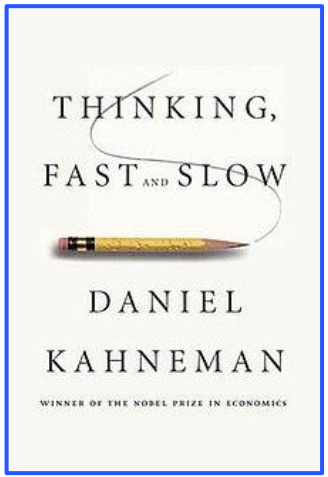
- What's the enemy of consistent, profitable trading?
  a. "We have met the enemy and he is us."
  b. Unfounded belief in my own abilities
  c. Why it's not as easy as "buy low, sell high."

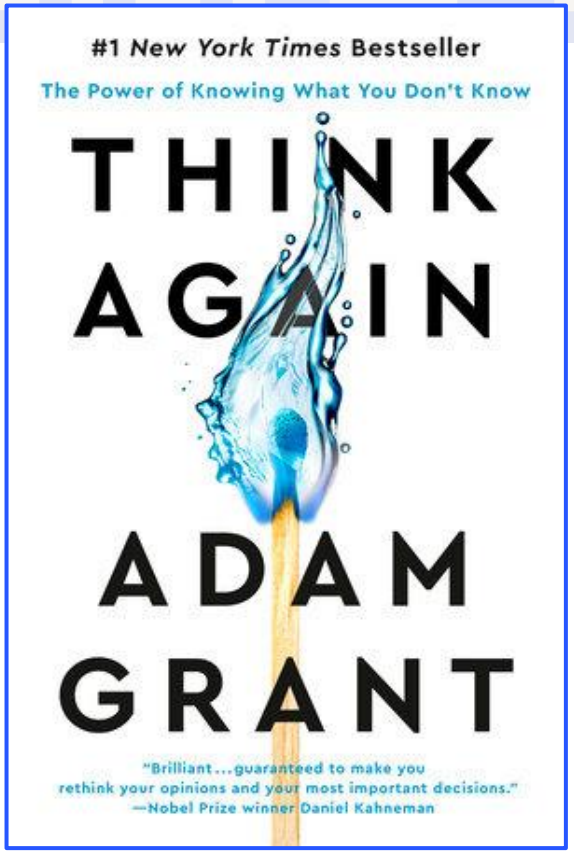




Be wrong. Learn something.

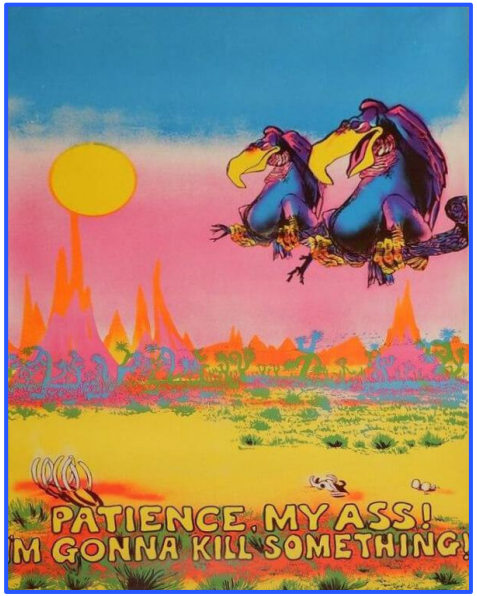
- ‘His (Daniel Kahneman’s) eyes lit up, and a huge grin appeared on his face. “That was wonderful,” he said. “I was wrong.” (p. 61)
- ‘As Danny told me, “Being wrong is the only way I feel sure I’ve learned anything.” (p. 62)





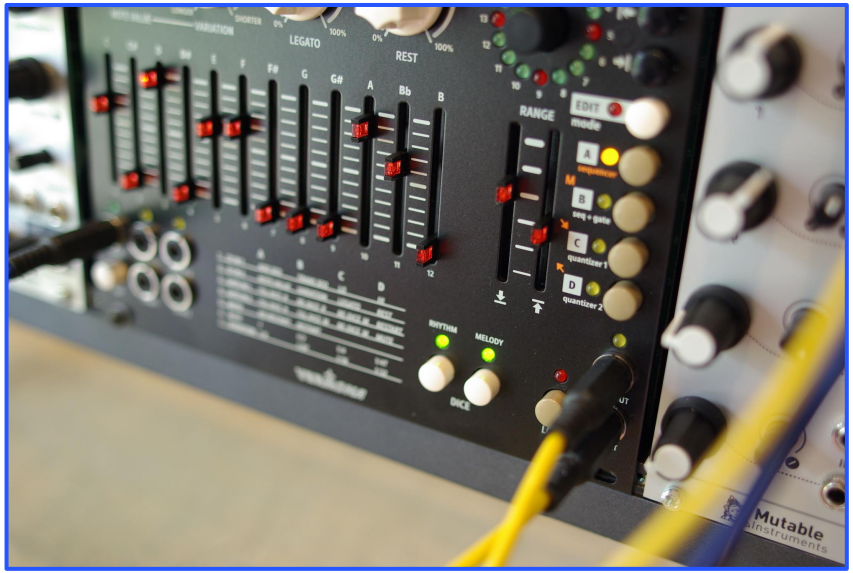
Risks and challenges of algorithmic trading

- Obsessing over bottom-line return
- Overriding the system
- Past performance is not a guarantee of future results
- You can only learn what you almost already know



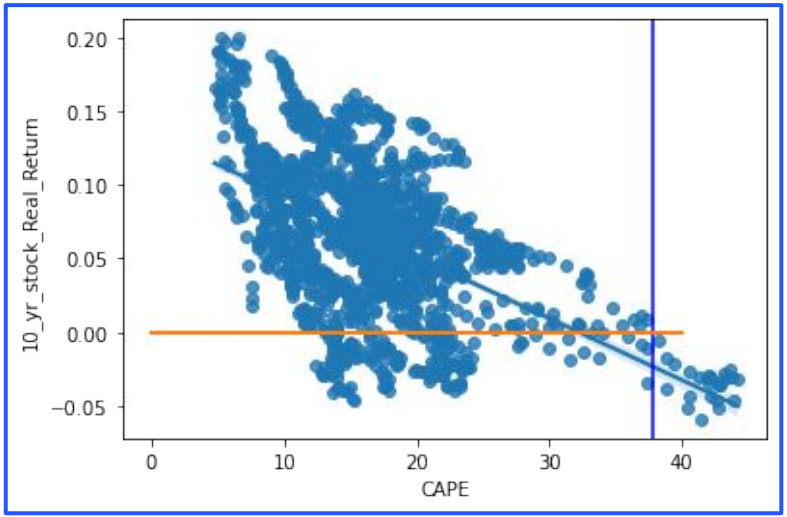
Building an algorithmic trading strategy

- Consistency over quantity
- Type 2 (Thinking slow) over Type 1 (Thinking fast)
- Problems with overfitting



Building an algorithmic trading strategy (cont.)

- Things to consider
  a. Where is the market going?
  b. Long or short?
  c. Does it trend? How often does it trend?



Building an algorithmic trading strategy (cont.)

- Trend-following tools
  a. MACD or RSI for: How strongly is it trending?
  b. SMA or EMA Golden cross for: When should I buy? When should I sell?
- (Reminder not to use ARIMA)
- Does it work for currencies? (It might work better!)
- System optimization
  a. Have enough data, corresponding to your trading patterns.
  b. Don't overfit
  c. Be wrong. Learn something.
  d. Compile many statistics:
    i. Average profit
    ii. Standard deviation on profits
    iii. Maximum loss
    iv. win/loss ratio
- Do performance monitoring on your system. Systems become obsolete.

Assignment #12

For this final project, you'll pull together all the Python skills you've developed in the last 6 weeks to build a basic algorithmic trading model. Find an existing algorithmic trading strategy, either online or in the libraries you've worked with in class. What will you optimize? Without overfitting, refine it to increase its return in an algorithmic model.

Resources: Books and Articles

- Thinking, Fast and Slow

  https://en.wikipedia.org/wiki/Thinking,_Fast_and_Slow

  Daniel Kahneman
- Think Again

  https://www.adamgrant.net/book/think-again/ (Take the quiz, if you like)

  Adam Grant
- Trading in the Zone

  Mark Douglas
- The Trouble with the Stock-Market Bubble

  Jason Zweig. Discusses the CAPE index

  https://www.wsj.com/articles/the-trouble-with-a-stock-market-bubble-11644595216

Resources

- Brokers with a Python API (from Mastering Python for Finance by James Ma Weiming, p. 265)

  (These are in case you want to have your laptop place buy & sell orders.)

  E*Trade

  Interactive Brokers

- Plotting the data

  https://medium.com/@jsteinb/python-build-a-program-to-retrieve-and-graph-live-stock-market-data-311d9ca1b7d3

- Backtrader

  https://medium.com/@danjrod/interactive-brokers-in-python-with-backtrader-23dea376b2fc

  https://github.com/mementum/backtrader

- Writing systems

  https://medium.com/@gk_/crypto-quant-programmatic-trading-of-btc-using-binance-and-backtrader-part-1-of-3-4ec95d6e8df8

Q&A