

# Course Overview

MGMT 675: AI-Assisted Financial Analysis



# Meet your Prof

- At Rice since 2009, in Jones and in Econ Dept.
  - teach core finance, quantitative investments, investments theory, and python for business research
  - to PhD and Masters in Data Science students
- Previously at Northwestern, Indiana, Washington Univ. in St. Louis, and Texas A&M. Associate Dean at Wash U.
- Former and current editor and associate editor of several journals. Two textbooks (derivatives and PhD asset pricing theory).
- Course materials and info at [kerryback.com](http://kerryback.com)

# Course Overview

---

# Motivation for this course

- How will AI affect finance jobs?
- Basic template for finance chatbot
  - Python code for financial analysis
  - App that encapsulates python code
  - App makes API calls to LLMs to process user's (natural language) input and pass structured version to python code
- Random web cites: 40+% of all new code written by AI, 80+% of developers use AI assistance

# Learning objectives

- Learn to use AI to write python code to perform financial analyses
- Get additional practice in financial analyses
- Obtain a basic understanding of
  - Python
  - Making API calls to LLMs
  - Building simple apps

- Julius.ai to write and execute python code
- Google Colab as alternative python environment (free)
- VS Code (free) and Cursor as alternative local environments
- Streamlit Cloud for hosting apps (free)
- OpenAI for API calls
- HuggingFace as alternative for API calls (free)

- Julius.ai to write and execute python code
- Google Colab as alternative python environment (free)
- VS Code (free) and Cursor as alternative local environments
- Streamlit Cloud for hosting apps (free)
- OpenAI for API calls
- HuggingFace as alternative for API calls (free)

# Assignments

- 6 group assignments, due by midnight on Thursdays beginning this week
- Form groups on canvas of no more than six students
- We'll break out into groups on most days
- Each assignment is to provide a link to a Julius workflow that will accomplish a specific financial analysis



**Julius**

---

## Get a Julius account

- Julius.ai provides a 50% academic discount. Sign up for a free account, then send an email using your Rice email account to team@julius.ai and ask for the academic discount. They will respond with a promo code to use.
- The Lite account (\$8 per month after discount) allows 250 messages per month and may be ok.
- Standard account (\$18 per month after discount) allows unlimited messages.

## Warm-Up

Click "+ New" to start a chat (thread) with Julius. Ask it to do the following (you could start a new chat for each topic):

- Ask Julius to use yfinance 0.2.54 to get AAPL's closing price from Yahoo Finance, plot it, and save as a jpeg.
- Ask Julius to use pandas datareader to get the 10-year Treasury yield from FRED, plot it, and save as a jpeg.
- Ask Julius to plot the payoff diagram of a call option with a strike of 100.
- Ask Julius what data it needs to compute the Black-Scholes value of a call option. Supply the data and ask for the value.

# Julius Workflows

- Scroll down the "Workspace" menu to find and select "Workflows."
- Move to the "Sandbox" tab and select "New Sandbox."
- The "Prompt" menu on the right includes a "User Prompt" option and a "Prompt" (tell Julius to run an analysis) option.
- Each workflow should start with a user prompt, asking the user to supply data and maybe asking the user what output the user wants.
- Specify in a Julius prompt the steps Julius should take with the information provided by the user.
- Follow up with another user prompt if needed, etc.

## Example Workflow

Create a workflow including the following:

- User prompt: Ask the user for a ticker and a start date.
- Julius prompt: Use yfinance to get the closing price of the stock from Yahoo Finance beginning at the start date. Plot the price. Output the plot to a jpeg file and show it to the user.

# Effective Prompting

- We are writing prompts to Julius that will supplement user input.
- Be explicit in what you want. Break down into steps if possible.
- Provide examples to Julius (code that has worked before or spreadsheets).
- Ask Julius to explain each step as it accomplishes it.
- Ask Julius to explain what it has done as it accomplishes each step.

# Excel, Python, and AI

---

- Python output → Excel data is easy
- Data in Excel → python is easy
- Formulas in Excel → python is possible via AI (ask Julius to read the formulas not the data).



# Building and saving models in python with AI

- Building:
  - Read and mimic Excel model, or
  - Ask AI to build a model from scratch
  - Both are often iterative processes. Need to correct AI mistakes.
- Saving:
  - Add working code to a Julius workflow: Copy working code and paste it into a "Prompt" cell as an example when creating a workflow.
  - Save working code elsewhere:
    - Copy and paste into a Jupyter notebook on your hard drive or Google Drive (for Colab).
    - Create a chatbot app that runs the code.