Course Syllabus: FINM 32900, Winter 2024

Contents

FINM 32900, Data Science Tools for Finance

Summary

Course Description “Data Science for Finance” is a hands-on course centered on key data science tools in quantitative finance. Acknowledging the field’s wide scope, the course focuses on a common skill set across various data science subfields. That is, this course examines elements of the analytical pipeline, from data extraction and cleaning to exploratory analysis, visualization, and modeling, and finally, publication and deployment. It does so with the aim of teaching the tools and principles behind creating reproducible and scalable workflows, including build automation, dependency management, unit testing, the command-line environment, shell scripting, Git for version control, and GitHub for team collaboration. These skills are taught through case studies, each of which will additionally give students practical experience with key financial data sets and sources such as CRSP and Compustat for pricing and financials, macroeconomic data from FRED and the BEA, bond transactions from FINRA TRACE, Treasury auction data from TreasuryDirect, textual data from EDGAR, and high-frequency trade and quote data from NYSE. Prior experience at an intermediate level with Python and the PyData stack is assumed.

- **Class:** Mondays, 6 - 9 PM, in-person at the Stevanovich Center building, Room #112. (5727 S. University Ave.)
- **Lecturer:** Jeremy Bejarano, jbejarano@uchicago.edu
- **Instructor Office Hours:** Fridays, 3 - 4 pm, on Zoom only. Link: Zoom link is available in the calendar on Canvas.
- **Teaching Assistants:**
    - Tobias Rodriguez del Pozo, tobiasdelpozo@uchicago.edu
    - Younghun Lee, hun@uchicago.edu
    - Note: Please include both TAs on all emails. However, students are strongly encouraged to post questions on the discussion page of the class GitHub repository here: Zoom link is available in the calendar on Canvas.
- **TA Office Hours:** Saturdays, 10-11 am ET, on Zoom only. Zoom link is available in the calendar on Canvas.
- **Website:** Canvas will be used for grades and for publishing Zoom links only. Homework and notes will be posted on the course GitHub repo: jmbejara/finm-32900-data-science. Questions and other class-related discussions should be posted here as well.
- **Textbook:** The text for the course will be published incrementally here: https://finm-32900.github.io/

NOTE: Due to the holiday on January 15, a makeup class on Zoom with be held on Saturday, Jan 13.

Assignments

- Assignments must be submitted via GitHub before 3 pm on Mondays. Each assignment will be distributed on a Monday, and will be due the following Monday. Assignments are automatically graded via the autograder on GitHub Classroom and solutions will be released shortly after. This means that the due date is strict. Late assignments will not be accepted.
- Each student is to individually submit their assignment (unless otherwise specified). Students may work in groups, but students are not allowed to copy each other's code. Each student must write their own solutions individually.
- After assignments are graded, solutions will be posted in separate GitHub repos, found here: finm-32900

Final Project

In lieu of a final exam, students will be organized into groups of 4 and will each complete a course project. Each group will present their completed project to the instructor at the end of the course. These presentations will be scheduled individually.

Assessment

Grades will be based on 7 coding assignments (70%), a final group project (25%), and participation (5%).

- Assignments will be submitted individually and will be graded using GitHub's automated testing tools.
- The final project will be completed in groups. Students will choose the project from among a few options provided at the beginning of the quarter. The project will be graded not only on how well it accomplishes the assigned data cleaning and analysis task, but will be primarily graded on whether (1) the steps to reproduce it are fully automated and well documented, (2) the code is written in a clean and reusable fashion, and (3) the results are presented clearly and presented in a way that convinces the reader that the results are correct. A more

specific rubric will be provided in class.

- The participation grade will depend on the positive impacts that a student has on the class. These include participating in in-class discussions and/or answering questions on the class GitHub page (or on Canvas). Students are in no way penalized for giving wrong answers in these in-class discussions nor is there any penalty for asking for help—asking for help is often the best way to learn!

Schedule

The schedule will follow the ordering of the chapters listed in the GitHub book found here: https://finm-32900.github.io/. Each week is it’s own chapter and the agenda is listed in the first sub-section of the chapter.

HW Due Dates

- HW 0: Ungraded. Due ASAP, preferably before the first class
- HW 1: Due Monday, Jan 15 at 3 pm
- HW 2: Due Monday, Jan 22 at 3 pm

References

I will provide the lecture notes that we will use in class here: https://finm-32900.github.io/. As a prerequiste, you should have some prior familiarity with Python and the PyData stack (e.g., Numpy, Scipy, Pandas, Matplotlib). The following references may serve as useful refreshers:

- Python for Data Analysis, 3rd Edition, by Wes McKinney
- Python Data Science Handbook, by Jake VanderPlas
- Python Programming for Economics and Finance, by Thomas J. Sargent and John Stachurski

A significant portion of this course is inspired by "The Missing Semester of Your CS Education", a short course taught in the Computer Science department at MIT. I'll rely on the material shown there for portions of this course.

Software to be used in class

Lectures will feature live programming exercises in class, so students should have a WiFi-enabled laptop to bring to class.

Before the first class, please make sure to install the required software and sign up for the required services. Students will need to install the following software on their laptop. Each of these pieces of software are free:

- Anaconda distribution of Python (Individual Edition)
- Visual Studio Code (NOT Visual Studio. Visual Studio Code is different from Visual Studio)
- Git
- GitKraken You will need to use GitKraken Client Pro, which is available for free for students.
- TeX Live
- PuTTY
- WinSCP For those using a Mac, you may need to find a software alternatives for WinSCP.

Students should also sign up for an account with the following websites. We will use free versions of each of these services:

- GitHub
- IPUMS CPS
- Wharton Research Data Services (WRDS) Apply for access through the University of Chicago, using the registration form here. For any issues that may arise, please contact the WRDS representative for UChicago's Mathematics department, John

Zekos, zekos@math.uchicago.edu.

Instructions to Run Code in this Repository

- To compile the book, run this from the repository's root directory

```
jupyter-book build -W ./
```

The option `-W` will treat warnings as errors.

Dependencies and Virtual Environments

The following is additional helpful information to run the code used in the lectures.

Working with `pip` requirements

`conda` allows for a lot of flexibility, but can often be slow. `pip`, however, is fast for what it does. You can install the requirements for this project using the `requirements.txt` file specified here. Do this with the following command:

```
pip install -r requirements.txt
```

Working with `conda` environments

The dependencies used in this environment (along with many other environments commonly used in data science) are stored in the conda environment called `blank` which is saved in the file called `environment.yml`. To create the environment from the file (as a prerequisite to loading the environment), use the following command:

```
conda env create -f environment.yml
```

Now, to load the environment, use

```
conda activate blank
```

Note that an environment file can be created with the following command:

```
conda env export > environment.yml
```

However, it's often preferable to create an environment file manually, as was done with the file in this project.

Also, these dependencies are also saved in `requirements.txt` for those that would rather use pip. Also, GitHub actions work better with pip, so it's nice to also have the dependencies listed here. This file is created with the following command:

```
pip freeze > requirements.txt
```

Other helpful `conda` commands

- Create conda environment from file: `mamba env create -f environment.yml`
- Activate environment for this project: `mamba activate blank`
- Remove conda environment: `mamba remove --name finm --all`
- Create blank conda environment: `mamba create --name myenv --no-default-packages`
- Create blank conda environment with different version of Python: `mamba create --name myenv --no-default-packages python` Note that the addition of "python" will install the most up-to-date version of Python. Without this, it may use the system version of Python, which will likely have some packages installed already.

`mamba` and `conda` performance issues

Since `conda` has so many performance issues, it's recommended to use `mamba` instead. I recommend installing the `miniforge` distribution. See here: conda-forge/miniforge

‹ Previous
Data Science Tools for Finance

Next ›
Homework 0: Setting up your computing environment