Week 10 Assignment- Portfolio Reflection for

Master of Science

Information and Communications Technology

Peyton Holdbrooks

University of Denver University College

May 13, 2025

Faculty: Guido Gioberto, MS
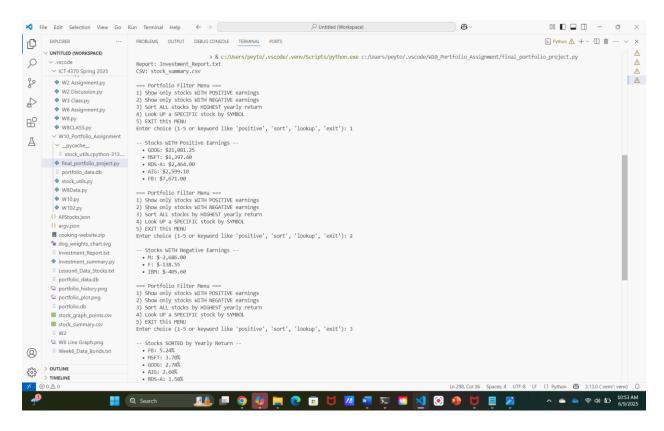
Dean: Michael J. McGuire, MLS
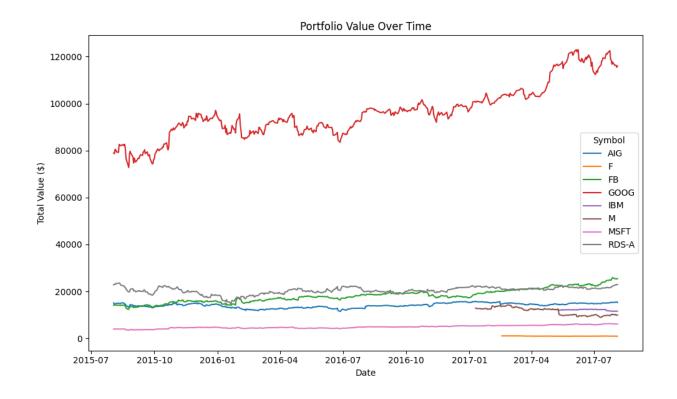
Contents

<u>Reflection</u>

Over the course of this Python programming course, I built a multi-faceted stock and bond analysis tool, evolving from simple list-based calculations to a comprehensive, object-oriented, data-driven application. Early on in the course, I focused on computing earnings and losses using lists, and then I migrated those calculations into reusable functions and classes, improving long-term maintainability. Storing and querying my final portfolio in an SQLite database added persistence while demonstrating practical database integration. Updates made to my final code (largely based on instructor feedback) included modularizing code into clearly separated functions and adding user interaction elements. The biggest feedback-driven enhancement was the interactive command-line filter, which allows you to display only positive or negative performers, sort by yearly return, or look up individual symbols. This made the tool more engaging and user-friendly, while making my application functional.
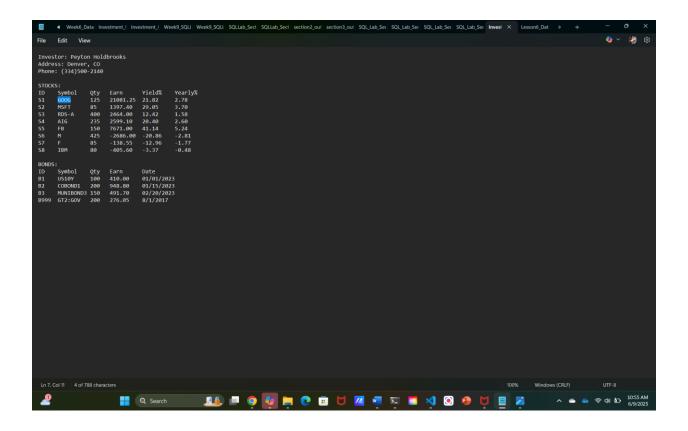
While pondering areas for further improvement, I considered building a minimal GUI—perhaps with Tkinter—to avoid command-line prompts or integrating a real-time API for live price updates. I'd also like to add more advanced analytics using Pandas, as this course's data analytics discussion inspired me to explore richer charting/graphing beyond simple line graphs. Implementing the filter menu was straightforward logically but required careful parsing of user input; consulting Python's string methods and trial-and-error in the REPL helped me refine a keyword-friendly parser. The documentation for input() and list comprehensions was easy to navigate, though piecing together edge-case handling such as invalid choices took some experimentation. The JSON-based historical visualization was more challenging: matching

symbols between my portfolio and the JSON file required diagnosing mismatches. I added a

dedicated portfolio_json mapping, following my Week 8 script patterns, and built diagnostic

printouts to guide symbol corrections. The Matplotlib docs were clear on date formatting and

legend handling, though I had to suppress warnings for unlabeled artists.

Documentation and Screenshots

Portfolio Value Over Time

Investor: Peyton Holdbrooks
Address: Denver, CO
Phone: (334)500-2140

STOCKS:

ID	Symbol	Qty	Earn	Yield%	Yearly%
S1	GOOG	125	21081.25	21.82	2.78
S2	MSFT	85	1397.40	29.05	3.70
S3	RDS-A	400	2464.00	12.42	1.58
S4	AIG	235	2599.10	20.40	2.60
S5	FB	150	7671.00	41.14	5.24
S6	M	425	-2686.00	-20.86	-2.81
S7	F	85	-138.55	-12.96	-1.77
S8	IBM	80	-405.60	-3.37	-0.48

BONDS:

ID	Symbol	Qty	Earn	Date
B1	US10Y	100	410.00	01/01/2023
B2	COBOND1	200	948.80	01/15/2023
B3	MUNIBOND3	150	491.70	02/20/2023
B999	GT2:GOV	200	276.05	8/1/2017

<u>References</u>

Matthes, Eric. 2023. Python Crash Course: a Hands-on, Project-Based Introduction to

Programming. 3rd ed. San Francisco: No Starch Press.