

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Accountancy

School of Accountancy

12-2022

The effectiveness of using Python programming approach in teaching financial analytics

Clarence GOH

Singapore Management University, clarecegoh@smu.edu.sg

Yuanto KUSNADI

Singapore Management University, yuantok@smu.edu.sg

Gary PAN

Singapore Management University, garypan@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/soa_research



Part of the [Accounting Commons](#), [Higher Education Commons](#), and the [Programming Languages and Compilers Commons](#)

Citation

GOH, Clarence; KUSNADI, Yuanto; and PAN, Gary. The effectiveness of using Python programming approach in teaching financial analytics. (2022). 1-32.

Available at: https://ink.library.smu.edu.sg/soa_research/2038

This Working Paper is brought to you for free and open access by the School of Accountancy at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Accountancy by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylids@smu.edu.sg.

The Effectiveness of Using Python Programming Approach in Teaching Financial Analytics

Clarence Goh, Yuanto Kusnadi, Gary Pan
Singapore Management University

Abstract

This paper presents the learning method and challenges regarding implementing a Python-programming approach in teaching financial analytics to graduate accounting students. The advent of Big Data as well as related applications and technologies have largely changed the process and practice of accounting. These have led to essential changes in the construction and teaching contents of accounting education. While there have been several studies examining how data analytics is embedded in accounting curriculum, majority of the teaching cases in accounting focus on analysis and communication with Excel as the principal tool, with very few of them covering the necessary steps prior to data analysis such as extraction and data preparation. Therefore, there is an opportunity for academics to create new teaching cases to provide students with comprehensive experiences throughout the data analytics cycle, perhaps with other much needed tools, such as Python. In this learning exercise, we describe our experience in developing and implementing a coding approach towards learning financial analytics for graduate accounting students as part of a data analytics curriculum. We share techniques on how to teach coding to accounting students with little or no coding experience, and some of the practical challenges in the classroom. Results from implementing this module in a graduate accounting program showed that using Python programming to teach financial analytics to accounting students was effective and innovative.

Keywords: Python programming approach, financial analytics, teaching innovation

I. INTRODUCTION

This paper demonstrates an effective approach to teaching graduate accounting students financial analytics through practical application of Python programming. The advent of Big Data as well as related applications and technologies have largely changed the process and practice of accounting and led to essential changes in the construction and teaching contents and practice of accounting education. As a result, we saw a need to embed financial analytics in our accounting curriculum, establishing students' programming skill and applying it to enhance financial analysis. Our experience showed that using Python codes helped accounting students to better understand valuation of financial instruments and portfolio analysis, despite our students' lack of substantial background in statistical programming.

Financial analytics refers to the interpretation and use of financial data to garner a robust understanding of business trends and give insight that allows management to make strategic, operational and investment decisions in a timely manner. For instance, companies may use market trends, company financial forecasts, and other economic indicators to set investment strategies.

In our course, students focus on learning financial analytics through Python programming to make investment decisions. Our students identify what trading instruments are available, what investment objectives are, and when is the best time to make an investment move. From an accountancy perspective, Python is most useful when working with data. It can essentially read any type, both structured and unstructured. It has powerful capabilities in data importation and manipulation – tasks like merging and recoding – as well as handling large amounts. Python can be used for efficient automation. For example, if one needs to run the same analysis on a new set of sales data each week, doing this in Excel would require opening a different file manually each week and re-entering formulas and other elements needed for the analysis, or creating complex automation with Power Query. But one could do that same

analysis automatically in a language such as Python, writing a simple script that imports the new data and runs the same analysis each week, outputting the results in whatever format one would like.

The benefits of financial analytics include setting clear investment direction; making timely investment decisions; enhancing market forecasting and comparison-based modelling (Andriosopoulos et al., 2019). According to a financial analytics market research study¹, the global financial analytics market was valued at \$7.6 billion in 2020 and is projected to reach \$19.8 billion by 2030. As such, it is essential for accountants to master financial analytics knowledge and skill. This is important as there is a growing expectation that accountants will be in the driver's seat for the transition companies must make as they take the critical role of analytics to become a data-driven organization. Moreover in 2018, the AACSB added a requirement that students master data analytics and that technology be incorporated across the entire accounting curriculum. Data analytics was also added to the CPA Exam in 2019.

To keep pace with digital revolution, the accounting curriculum offered in universities are being revamped to allow students who complete a bachelor's degree, to become competent in programming with Python upon graduation (Polimeni and Burke, 2021). Qasim and Kharbat (2020) also highlight data analytics ought to be incorporated in the accounting curriculum so as to prepare accounting graduates for the workplace and enhance their employability.

While there is little doubt that data analytics knowledge and skills are important to an accounting education, the challenge for most accounting faculty, however, is how to deliver a financial analytics course with Python programming. Accounting faculty may not be used to teaching a programming course, which involves knowing what to code, executing their code

¹ <https://www.alliedmarketresearch.com/financial-analytics-market>.

and resolving compiler errors, and debugging the code, knowledge and skills varied significantly from traditional accounting courses.

This paper aims to plug a gap in the accounting education literature by proposing a coding approach toward learning financial analytics. Such approach goes beyond case study or excel-based exercises. The technology and data analytics developments affecting the accounting profession have a profound effect on accounting curricula. Therefore, it is essential for academics to integrate current and emerging technologies into accounting curriculums in order to provide students with the appropriate knowledge and skill sets necessary to succeed in the accounting profession (Zhan et al., 2018; Polimeni, 2021; Dow et al., 2021). To date, there have been a number of studies in accounting education that have examined how data analytics are taught in the classroom. For instance, a common theme is teaching blockchain to accounting students by using a combination of storytelling and scaffolding approach (Stratopoulos, 2020); offering teaching syllabus and resources (Stern and Reinstein, 2021); and through coding with R. Another theme includes teaching Robotic Process Automation (RPA) to accounting students (Zhang and Vasarhelvi, 2022) and in an audit class (Bellinga et al., 2022). Other themes include teaching Python programming to management accounting students (Schoute, 2019) and introducing textual analysis into the accounting classroom (Fisher et al., 2022).

While there have been several studies examining how data analytics is embedded in accounting curriculum, majority of the teaching cases in accounting focus on analysis and communication with Excel as the principal tool, with few of them cover necessary steps prior to data analysis such as extraction and data preparation (Raschke and Charron, 2021). Therefore, there is a need for academics to create new teaching cases to provide students with comprehensive experiences throughout the data analytics cycle, perhaps with other much needed tools. As company's systems become more automated, it will also be important for accountants to understand the language that builds these programs and how information can be

extracted from it. Understanding coding languages like Python will enable accountants to understand how data were obtained and review code to ensure the population is error-free and complete. From performing data reconciliation and document review to being able to understand the output from a programme that does these tasks for them will become required skills for accounting graduates as they are entering the profession. To our knowledge, our study is one of the first teaching cases to implement a Python coding approach in a financial analytics course. Python skills have become increasingly important to support data analytics work. It helps accountants with the data extraction and wrangling work that can be so time-consuming, as well as supporting more statistical analysis. Where accountants are working extensively with large amounts of data, coding is increasingly likely to become part of the job.

To this end, we describe our experience in developing and implementing a coding approach towards learning financial analytics for graduate accounting students. We share techniques on how to teach coding to accounting students with little or no coding experience and some of the practical challenges in the classroom.

The remainder of the paper is organized as follows. In the next section, we discuss the motivations and challenges in teaching financial analytics with Python coding to graduate accounting students. We then discuss our approach for teaching the financial analytics module by highlighting the challenges and including suggestions for course implementation. Finally, we acknowledge limitations of our learning experience.

II. MOTIVATION AND CHALLENGES IN PYTHON EDUCATION IN ACCOUNTING

Growing Demand for Python Skills in Accounting

Python is an important programming tool and platform technology that has many practical applications in the accounting setting. In addition to being able to handle large

volumes of data, Python is also capable of reading both structured and unstructured data types, and has powerful capabilities in data importation, data manipulation, and statistical modelling. These characteristics have allowed Python to play a key role in tasks such as automation, model creation, and data visualization (Gillon 2021). Accordingly, Python has been widely adopted across a range of accounting and finance functions, including in the areas of financial accounting (York 2018), management accounting (Schoute 2019), forensic accounting (Miller and Bryce 2019), audit (AICPA 2019; ICAEW 2021), and financial valuation (Yan 2017). Organizations which have been reported to have adopted Python include accounting firms, financial institutions, fintech firms, and hedge funds (Amen 2019; Protasiewicz 2020; Taylor 2020; Hilpisch 2022).

Recognizing the important role that Python can play in their organizations, employers in the accounting and finance industry increasingly require their employees to possess knowledge of python programming. For example, the Developer Skills Report highlights that 26.4 per cent of employers in the finance services industry surveyed require employees to possess Python programming skills (HackerRank 2018). In addition, a recent study by the jobs portal eFinancialCareers finds that Python is the most commonly requested programming language by employers in the accounting and finance industry, with 1,497 job postings in 2020 (Butcher 2020). Citigroup, the financial services institution, has also been reported to have started requiring its analysts to attend Python classes as part of their continuing education programme. In explaining Citigroup's motivation for providing such classes for its analysts, Lee Waite, Citigroup's head of markets and securities services in North America was quoted as saying that the bank was "moving more quickly into this world" because "an understanding of coding seems to be valuable (Surane 2018)."

Despite the growing demand for Python skills from the industry, there is evidence of a relative lack of job-seekers in the accounting and finance industry who possess the relevant

skillsets. For example, the previously cited study by eFinancialCareers also finds that for every banking job requiring Python skills that is posted on the job portal, there are only 14 candidates with the relevant skillsets listed on their resumes (i.e. there are only 14 candidates with Python skills vying for each job posted). In contrast, for every banking job requiring skillsets such as VBA (PHP), there are 44 (77) candidates with the relevant skillsets listed on their resumes (Butcher 2020). Speaking at the launch of a study examining job postings for finance and accountancy roles over the past five years, Andrew Bushfield, a Director at recruitment firm Robert Half, highlighted the growing demand for employees with technology skills, including in Python, by saying that “demand for specialist digital skills like Python, SAP, ERP, Structured Query Language (SQL) and Scrum have been growing for more than five years, but the COVID-19 pandemic caused a real crunch..the talent to fill this demand is in short supply, which is pushing employers to increase salary offers for skilled professionals by 10 to 20 per cent (Williamson 2021).”

Given the demand for skillsets in Python in the accounting and finance sector, universities have a key role to play in developing talent who possess these skillsets. In 2018, the Association to Advance Collegiate Schools of Business (AACSB) formally adopted new accreditation standards for business school accounting programs, and included standard A5, which specifically requires accounting programs to integrate the teaching of current and emerging technologies across the curriculum (AACSB 2018). In explaining the motivation for the implementation of standard A5, Stephanie Bryant, the Chief Accreditation Officer for AACSB, highlighted that AACBS intends “for the accounting curriculum to be current, relevant, and forward-looking to prepare students to be successful as they enter the accounting profession (Vien 2018).” Therefore, to the extent that Python programming skills fulfil a key demand that employers in the accounting and finance industry have (Williamson 2021),

incorporating Python training in the accounting curriculum would play a critical role in enabling accounting programs to meet standard A5 set by AACSB.

Challenges of Teaching Python to Accounting Students

Accounting instructors face several key challenges in incorporating Python in the accounting curriculum. First, the use of Python in accounting-related courses is a relatively recent development in many accounting programs. For example, the course which we examine in this paper was first introduced only in 2018. Accordingly, instructors often face the challenging task of having to not only design the courses, but to also familiarize themselves with the course material, without structured support from their institutions (Gainey, Gunderman, Slayton, and Splenda 2021). Second, while courses in accounting may employ Python, they are typically not designed as coding classes aimed primarily at imparting coding skills. Instead, these courses often focus on the application of Python to solve problems within the accounting/finance context. In this regard, a key challenge facing instructors relates to how they should approach and integrate both the Python programming and accounting-application aspects of the course. Our study addresses these two challenges by providing guidance on the implementation of a course that uses Python in a specific aspect of accounting/finance application (i.e. the analysis of financial instruments).

Third, given that accounting students' academic backgrounds, including in the area of computer literacy, can differ (Larres, Ballantine, and Whittington 2003; Byrne and Flood 2008), instructors often find themselves having to teach a class where students' experience with Python can vary greatly. For example, one survey respondent in a study examining the experiences of university instructors who teach data-related courses conducted by Gainey et al. (2021) highlights the even the coding ability of incoming first-year students can differ according to the high school they attended. In this regard, one challenge that instructors face

relates to the need to design their courses to accommodate students with varying levels of prior exposure to Python programming. Our study addresses this challenge by evaluating the effectiveness of a Python course where there is substantial variation in students' prior knowledge of Python programming.

III. TEACHING PYTHON TO ACCOUNTING STUDENTS

The most important research question that this paper is attempting to answer is what is the effectiveness of using Python in teaching various concepts related to financial analytics in one of the core modules titled “*Analytics for Financial Instruments*” that postgraduate students of the Singapore Management University learn in the Master of Science in Accounting (Data Analytics) program, offered by the School of Accountancy. We also describe the implementation of the Python programming lessons in the classroom setting.

The Master of Science in Accounting Program

The Master of Science in Accounting (Data Analytics) or MSA in short was launched in the Academic Year 2018/19. It is the first Master's in Accounting degree program in Asia, designed for professionals who are interested in developing expertise in applying data technology into their domain knowledge to solve business problems, especially in the finance and accounting sectors. The availability of massive amounts of data has created a high demand for skilled data analytics professionals who can analyse such data and obtain the insight needed for informed decision-making. This is extremely vital to accounting and financial services firms as they have been dealing with large amount of information produced by their clients. Armed with machine learning and data analytics skills, our MSA graduates excel in using data to streamline processes, optimize resources and forecast future revenues.

The courses offered in the program include: Programming with Data (using Python), Data Management (using SQL), Applied Statistics for Data Analytics (using R), Visutal

Analytics for Accounting (using Tableau and PowerBI), Forecasting and Forensic Analytics (using R), and Analytics for Financial Instruments and Value Investing (both using Python).

Analytics for Financial Instruments

“*Analytics for Financial Instruments*” is one of the core modules that postgraduate students enrolled in the MSA program have to clear and it is usually offered in the last semester (Term 3) of the Academic Year (from middle of April to end of June). Python programming language is used to teach the various concepts (as elaborated below) and students need to pass one basic Python course offered in Term 1 of the Academic Year (middle of August to end of October in the previous year) as one of the pre-requisites. It is also one of the two courses on Python taken in the last semester which students will learn to apply the Python programming language in accounting and finance setting (the other course that uses Python is titled “*Analytics for Value Investing*”).

Classroom Implementation

In this course, students will be introduced to the various financial instruments that can be traded in the capital markets. In particular, the course will focus on stocks, bonds, and derivatives (such as options and futures) and how statistical tools can be applied to value and analyze these important instruments. As the course emphasizes on the practical application of Python programming language, the instructor will usually upload the Python codes on eLearn (SMU’s learning management system) that students can download and study, at least one week before each seminar. During the seminar, the instructor will explain the important codes and execute the codes to illustrate the various concepts. Jupyter Notebook is used as the platform to modify and execute the Python codes.

For example, in the first and second seminars, students will learn how to extract basic financial data from public sources (such as Yahoo! Finance) and how these data can be used by managers to plot candlestick charts, histogram and return graphs, as well as perform basic statistics (such as mean, standard deviation, median, skewness, kurtosis, and correlation) make informative investment decisions for their firms. Please refer to the teaching notes (Stage 1 to 3) for the Python codes that are used in the first two seminars.

More importantly, in the subsequent seminars, students will also be taught the application of financial theories such as modern portfolio optimization, fixed income analytics, Black Scholes option pricing model, Monte Carlo simulation and machine learning for forecasting of financial instruments as well as techniques on hedging. Please refer to the teaching notes (Stage 4) for the Python codes that are used in one application (i.e. Monte Carlo Simulation).

This course is particularly suitable for students who wish to work for financial institutions as it exposes them use the various analytical tools in Python to generate important inferences and help managers make value-increasing investment decisions. Therefore, students will be organized into teams at the start of the course, working collaboratively together to work on a portfolio optimization project. In the middle of the semester, there is also a guest lecture arranged where a portfolio manager offers various learning tips to students and illustrates the real life application of Python in executing modern portfolio strategies.

At the end of the course, students are expected to present their project findings and write an investment proposal report to recommend the appropriate portfolio diversification strategy for an investor. The codes that they submit will be verified for correctness by the instructor, to ensure that each project group has delivered presentation, report as well as Python codes that are of high quality.

Effectiveness of Python as a Teaching Tool

In order to test the effectiveness of Python as a teaching tool in the course, MSA students enrolled in “*Analytics of Financial Instruments*” offered in Term 3 of AY 2021/22 were asked to complete a pre-course and post-course survey questionnaires. From a total of 46 students, 40 students (representing a very high response rate of 87%) completed both survey forms, which were distributed via Google Forms.² Responses were collected and statistical analysis were performed on the responses to test the research question of this paper.

Table 1 presents the basic demographics of the survey respondents, which they filled up in the Pre-Course survey form. 25 (62.5%) of the respondents were female students and the remaining 15 (27.%) were male students. The age distribution revealed that a majority (80%) of the students enrolled in the course was in the age group of 18 to 30. Moreover, 30 (75%) out of the 40 respondents were full-time students, with the remaining 10 (25%) being part-time students.

[Insert Table 1 here]

In terms of background of undergraduate degree, an overwhelming 78% of the respondents was from accounting, with another 10% from business. 82.5% of the respondents had working experience of up to 10 years and only 5% of the respondents were matured, experienced students with more than 15 years of work experience. It is also not surprising to see that 78% of the respondents were in either audit or accounting and finance. Only 10% of the respondents did not have any work experience.

In terms of prior Python programming course taken during undergraduate, only 33% of the respondents had taken at least one course. Considering that the respondents’ undergraduate background were mostly from accounting and business, only 1 student reported that he/she had not taken any accounting and finance course before.

² To incentivize students to participate in the surveys, a \$10 Starbucks voucher was given to every student who had completed both the Pre and Post-Course surveys.

Five questions in the Pre-Course survey form were related to respondents' proficiency level in using Python programming language, before formally taking the course. Similar questions were asked in the Post-Course survey form, after the respondents had completed the course. The survey questions were designed using a seven-point Likert scale ranging from very low (1) to very high (7), in terms of proficiency. The responses were tracked so as to enable a pre and post course analysis, which we present in Panels A and B Table 2.

The first question asked is related to the students' proficiency in Python programming language in general. Since all of them had taken a basic Python course in Term 1 of the AY 2021/22 before taking course, 50% reported that their proficiency was high (mean = 4.48, std dev = 1.3). This value increased significantly to 83% (mean = 5.48, std dev = 1.11), after taking the course. Panel C of Table 2 also reports the difference in the mean values pre and post course, which was highly significant (t -statistic = 4.79) for the first question. Hence, through the Python codes taught in the various topics, the course had resulted in an increase in students' proficiency in Python programming language.

[Insert Table 2 here]

More relevant for the course was the various concepts being taught, namely: obtaining stock market data from Yahoo! Finance, portfolio analysis, valuation of financial instruments and forecasting using Python. Before students took the course, majority of them reported low proficiency in all the concepts mentioned above, ranging from 73% for forecasting to 88% for valuation of financial instruments. The mean scores were low as well, ranging from 1.98 (for valuation of financial instruments) to 2.25 (obtaining stock market data from Yahoo! Finance). These findings from the Pre-Course survey gave us the confidence that the students' proficiency in the main learning objectives of the course was in general low.

After taking the course, the Post-Course survey revealed a starkly different picture. Now, students mostly report high proficiency in the concepts taught in class using Python.

Forecasting had the lowest mean scores of 5.43 (std dev of 1) and 85% of the students reporting high proficiency level, Meanwhile obtaining stock market data from Yahoo! Finance had the highest mean scores of 6.28 (std dev of 1.11) with all students reporting high proficiency level. The pre and post analysis in Panel C reports consistent findings as the difference in the mean levels for all the questions being positive and statistically significant at least at the 1% level. Overall, there is a strong evidence to suggest that the use of Python programming language is effective in helping students understand and apply the concepts being taught to achieve the learning outcomes of the course.

Apart from the analysis in Table 2, we further ask seven questions in the Post-Course survey to understand if students agree that the instructions and Python codes taught in class help them to understand the various concepts taught, if the group project was relevant in applying the knowledge learnt on the topic of portfolio analysis, and if students would recommend using Python in the course and other accounting and finance courses. Similarly, a seven-point Likert scale questionnaire design was used, ranging from 1 (Strongly Disagree) to 7 (Strongly Agree). The results are presented in Table 3.

We observe that the mean scores for all questions in Table 3 are above 6, ranging from 6.08 (writing Python codes helped me understand valuation of financial instruments) to 6.68 (I would recommend using Python in finance and accounting courses). More relevantly for the course, an overwhelming 95% of the students recommend continuing using Python in the course (mean score = 6.68, std dev = 0.66) the group project was relevant in applying students' knowledge learnt on the topic of portfolio analysis (mean score of 6.55, std dev = 0.93).

[Insert Table 3 here]

The findings from Table 3 complement that of Table 2, which may also explain why students were able to acquire high proficiency level in the various concepts learnt using Python.

The main reasons were due to the detailed instructions and codes provided earlier to the students, before the instructor covered the various topics (obtaining stock market data, portfolio analysis, valuation of financial instruments, and forecasting). The experience in the group project also allowed students to apply the Python codes learnt in the seminars to recommend portfolio diversification strategy using real financial data. Finally, the experience in the course also led students to recommend continuing using Python in this course as well as other finance and accounting courses. Therefore, the objective of using Python programming language in teaching financial analytics is achieved in that students are now able to acquire high proficiency in not only Python, but also the various concepts being taught, so that they can be prepared to take on more challenges after their graduation.

IV. CONCLUSION

The accounting profession is rapidly evolving due to technological innovations such as data analytics. The role of accountants will need to develop programming skills as data analytics continues to enhance and drive changing business models. At the same time, academic accounting departments in universities are revamping their curriculum to ensure that their programs equip accounting students with relevant programming and data analytics skills that will prepare them for the workplace of the future. In this learning exercise, we describe our experience in developing and implementing a coding approach towards learning financial analytics for graduate accounting students as part of a data analytics curriculum. We share techniques on how to teach coding to accounting students with little or no coding experience and some of the practical challenges in the classroom.

We acknowledge several limitations of this study. Specifically, while Python programming is introduced in a number of different courses in the curriculum, the survey of teaching effectiveness was only conducted in one course (Analytics of Financial Instruments).

This can be expanded to the other courses so that a more holistic picture on the effectiveness of using Python programming language can be formed. In addition, the survey respondents were limited to only 40. We aim to increase the respondents size to a more sizeable number in future runs of the course.

REFERENCES

Amen, S. 2019. Which version of Python should you learn for finance jobs?

eFinancialCareers. Available at:

<https://www.efinancialcareers.sg/news/2019/08/learning-python-for-banking-jobs>

American Institute of Certified Public Accountants (AICPA). 2019. *Upgrade the financial statement audit using audit data analytics*. Available at:

<https://us.aicpa.org/content/dam/aicpa/interestareas/frc/assuranceadvisoryservices/downloadabledocuments/ads-instructional-paper-python.pdf>

Andriosopoulos, D., Doumpos, M., Pardalos, P., and C. Zopounidis. 2019. Computational approaches and data analytics in financial services: A literature review. *Journal of the Operational Research Society* 70 (10): 1581-1599.

Association to Advance Collegiate Schools of Business (AACSB). 2018. *2018 Eligibility procedures and accreditation standards for accounting accreditation*. Available at:

<https://www.aacsb.edu/-/media/documents/accreditation/accounting/standards-and-tables/2018-accounting-standards.pdf?la=en&hash=8DCDA6CE3B0CEF6AB82D39CBF53995DA96111196>

Bellinga, J., Bosman, T., Höcük, S., Janssen, W., Khzam, A. 2022. Robotic Process Automation for the Extraction of Audit Information: A Use Case. *Current Issues in Auditing* 16 (1): A1-A8.

Bryne, M, and B. Flood. 2008. Examining the relationships among background variables and academic performance of first year accounting students at an Irish University. *Journal of Accounting Education* 26 (4): 202-212.

Butcher, S. 2020. The coding languages that will get you a job in banking, ranked.

eFinancialCareers. Available at:

<https://www.efinancialcareers.co.uk/news/finance/the-six-hottest-programming-languages-to-know-in-banking-technology>

Dow, K., Jacknis, N., Watson, M. 2021. A Framework and Resources to Create a Data Analytics-Infused Accounting Curriculum. *Issues in Accounting Education* 36 (4): 183-205.

Fisher, I., Hughes, M., Janvrin, D. 2022. Put your Best Text Forward: Introducing Textual Analysis into the Accounting Classroom. *Issues in Accounting Education* 37 (1): 141-195.

Gainey, M., H. Gunderman, E. Slayton, and R. Splenda. 2021. Understanding the practices and challenges of teaching with data in undergraduate social science courses at Carnegie Mellon University. Available at:
https://kithub.cmu.edu/articles/report/Understanding_the_Practices_and_Challenges_of_Teaching_with_Data_in_Undergraduate_Social_Science_Courses_at_Carnegie_Mellon_University/16693072

Gillon, K. 2021. Why you should learn to code. *Institute of Chartered Accountants of England and Wales (ICAEW)*. Available at:
<https://www.icaew.com/technical/business/webinars-and-publications/business-and-management-magazine/BAM-May-2021/Why-you-should-learn-to-code>

HackerRank. 2018. *2018 Developer skills report*. Available at:
<https://www.hackerrank.com/research/developer-skills/2018/>

Hilpisch, Y. 2022. *Financial theory with Python*. Sebastopol, C. A.: O'Reilly Media Inc.

Institute of Chartered Accountants of England and Wales (ICAEW). 2021. *How Eurobank got to grips with data analytics in internal audit*. Available at:
<https://www.icaew.com/insights/viewpoints-on-the-news/2021/dec-2021/how-eurobank-got-to-grips-with-data-analytics-in-internal-audit>

- Kaden, S., Lingwall, J., and Shonhiwa, T. 2021. Teaching Blockchain through Coding: Educating the Future Accounting Professional. *Issues in Accounting Education* 36 (4): 281-290.
- Larres, P., J. Ballantine, and M. Whittington. 2003. Evaluating the validity of self-assessment: Measuring computer literacy among entry-level undergraduates within accounting degree programmes at two UK universities. *Accounting Education* 12 (2): 97-112.
- Miller, P., and C. Bryce. 2019. *Learning Python for forensics*. Birmingham, U.K.: Packt Publishing Ltd.
- Polimeni, R. 2021. Integrating Emerging Accounting Digital Technologies and Analytics into an Undergraduate Accounting Curriculum – A Case Study. *Journal of Emerging Technologies in Accounting* 18 (1): 159-173.
- Protasiewicz, J. 2020. How Python is used in finance and fintech. *Netguru*. Available at: <https://www.netguru.com/blog/python-in-finance>
- Raschke, R., and Charron, K. 2021. Review of Data Analytic Teaching Cases, have we Covered Enough? *Journal of Emerging Technologies in Accounting* 18 (2): 247-255.
- Schoute, M. 2019. Teaching Python to management accounting students: An illustration using support department cost-allocation methods. *The Accounting Educators' Journal* 29 (1): 137-161.
- Stern, M., and Reinstein, A. 2021. A Blockchain Course for Accounting and Other Business Students. *Journal of Accounting Education* 56: 100742.
- Stratopoulos, T. 2020. Teaching Blockchain to Accounting Students. *Journal of Emerging Technologies in Accounting* 17 (2): 63-74.

Surane, J. 2018. Citi wants analysts to add Python to list of languages on resume. *Bloomberg*.

Available at: <https://www.bloomberg.com/news/articles/2018-06-14/citi-wants-analysts-to-add-python-to-list-of-languages-on-resume#xj4y7vzkg>

Taylor, C. 2020. For workers, 'digital upskilling' puts tech trends on fast-forward. *Reuters*.

Available at: <https://www.reuters.com/article/us-world-work-digitalupskilling-idUSKBN2771C3>

Vien, C. 2018. What to know about AACSB accounting standard A5. *Journal of*

Accountancy. Available at: <https://www.journalofaccountancy.com/newsletters/extra-credit/aacsb-accounting-standard-a5.html>

Williamson, R. 2021. Surprising tech skills in demand for accounting and finance. *CPA*

Australia. Available at: <https://intheblack.cpaaustralia.com.au/careers-and-workplace/tech-skills-in-demand-for-accounting-and-finance>

Yan, Y. 2017. *Python for finance*. Birmingham, U.K.: Packt Publishing Ltd.

York, D. 2018. Accounting-system: A GAAP compliant basic accounting system in Python

and SQLite. Available at: <https://medmatix.github.io/Accounting-System/>

Zhan, J., Her, YW., Hu, T., and Du, C. 2018. Integrating Data Analytics into the

Undergraduate Accounting Curriculum. *Business Education Innovation Journal* 10 (2): 169-178.

Zhang, C., and Vasarhelvi, M. 2022. How to Teach a 14-Week Robotic Process Automation

(RPA) Course for Accounting Students? *Issues in Accounting Education* 10.2308.

TABLE 1
Demographics of Graduate Students Enrolled in
Analytics for Financial Instruments

Characteristics of Respondents	Number of Respondents	Percentage of Respondents
Gender		
Female	25	62.50%
Male	15	37.50%
Age		
18 to 25	17	42.50%
26 to 30	15	37.50%
31 to 35	4	10.00%
36 to 40	2	5.00%
Above 40	2	5.00%
Student Status		
Full time	30	75.00%
Part time	10	25.00%
Undergraduate Degree Background		
Accounting	31	77.50%
Business	4	10.00%
Others	5	12.50%
Working Experience		
0 to 5 Years	27	67.50%
6 to 10 Years	6	15.00%
11 to 15 Years	4	10.00%
More than 15 Years	2	5.00%
Field of Work		
I do not have any work experience	4	10.00%
Audit	13	32.50%
Accounting & Finance	18	45.00%
Others	5	12.50%
Python Programming course(s) taken during Undergraduate		
Have not taken any course	27	67.50%
At least one course	13	32.50%
Accounting & Finance course(s) taken during Undergraduate		
Have not taken	1	2.50%
1 to 2	1	2.50%
3 to 5	4	10.00%
More than 5	34	85.00%

TABLE 2
Proficiency Level in using Python to Achieve the Learning Outcomes, before and after taking Analytics for Financial Instruments

<i>Panel A: Before taking the course</i>								
Please rate your proficiency on the following:	Mean	Median	Std Dev	Min	Max	% Low	% Neutral	% High
1. Python programming language in general	4.48	4.50	1.30	2.00	7.00	22.50%	27.50%	50.00%
2. Obtaining stock market data from Yahoo! Finance using Python	2.25	1.00	1.56	1.00	6.00	77.50%	12.50%	10.00%
3. Portfolio analysis using Python	2.05	1.50	1.26	1.00	5.00	85.00%	10.00%	5.00%
4. Valuation of financial instruments Python	1.98	1.00	1.23	1.00	5.00	87.50%	7.50%	5.00%
5. Forecasting using Python	2.40	1.50	1.68	1.00	7.00	72.50%	12.50%	15.00%
<i>Panel B: After taking the course</i>								
Please rate your proficiency on the following:	Mean	Median	Std Dev	Min	Max	% Low	% Neutral	% High
1. Python programming language in general	5.48	5.50	1.11	3.00	7.00	5.00%	12.50%	82.50%
2. Obtaining stock market data from Yahoo! Finance using Python	6.28	6.00	0.78	5.00	7.00	0.00%	0.00%	100.00%
3. Portfolio analysis using Python	6.10	6.00	0.87	4.00	7.00	0.00%	5.00%	100.00%
4. Valuation of financial instruments Python	5.68	6.00	1.00	4.00	7.00	0.00%	10.00%	100.00%
5. Forecasting using Python	5.43	5.00	1.15	3.00	7.00	5.00%	10.00%	85.00%
<i>Panel C: Difference in means test</i>								
Please rate your proficiency on the following:	Difference in Mean (After – Before)		t-statistics			p-value		
1. Python programming language in general	1.00		4.79			(0.00)		
2. Obtaining stock market data from Yahoo! Finance using Python	4.03		14.70			(0.00)		
3. Portfolio analysis using Python	4.05		17.89			(0.00)		
4. Valuation of financial instruments Python	3.70		18.21			(0.00)		
5. Forecasting using Python	3.05		11.49			(0.00)		

A seven-point Likert scale survey is used where 1 = Very Low, 2 = Low, 3 = Somewhat Low, 4 = Neutral, 5 = Somewhat High, 6 = High, and 7 = Very High

TABLE 3
Descriptive Statistics on Usage of Python in Analytics for Financial Instruments (n = 40)

Questions	Mean	Median	Std Dev	Min	Max	% of Agree	% Neutral	% Disagr
Please rate your agreement with the following statements:								
1. Instructions and codes given in class helped me understand portfolio analysis	6.43	7.00	0.75	4.00	7.00	97.50%	2.50%	0%
2. Writing Python codes helped me understand portfolio analysis	6.25	6.00	0.84	4.00	7.00	97.50%	2.50%	0%
3. Instructions and codes given in class helped me understand valuation of financial instruments	6.21	6.00	0.77	4.00	7.00	97.50%	2.50%	0%
4. Writing Python codes helped me understand valuation of financial instruments	6.08	6.00	0.86	4.00	7.00	97.50%	2.50%	0%
5. I would recommend continuing using Python in this course	6.35	7.00	0.89	4.00	7.00	95.00%	5.00%	0%
6. I would recommend including Python in finance and accounting courses	6.68	7.00	0.66	4.00	7.00	97.50%	2.50%	0%
7. The group project was relevant in applying your knowledge learnt on the topic of portfolio analysis:	6.55	7.00	0.93	3.00	7.00	95.00%	2.50%	2.5%

A seven-point Likert scale survey is used where 1 = Strongly Disagree, 2 = Disagree, 3 = Somewhat Disagree, 4 = Neutral, 5 = Somewhat Agree, 6 = Agree, and 7 = Strongly Agree

APPENDIX 1: TEACHING NOTES

The teaching notes below provides the detailed implementation to learning financial analytics through programming in Python.

Stage 1: Understanding the Python programming language

In the Master of Science in Accounting (Data Analytics) program, students are expected to have completed a course titled “Programming with Data”, which includes learning the basics of data analytics in Term 1 of the program. The skill-sets that they are expected to acquire in that course includes: learning programming fundamental concepts and constructs, the building blocks of information systems, through the Python programming language.

Stage 2: Obtaining financial data using Python and visualizing the data

The first skill-set that students will learn in the course titled “Analytics of Financial Instruments” is to retrieve financial data from online public data sources such as <https://finance.yahoo.com>, using Python. One important requirement of accomplishing the task is to apply the knowledge they have learnt earlier in Stage 1 on “pandas” , which is a software library written for Python, that is most commonly used for data manipulation and analysis as well as machine-learning tasks. Pandas are intended to work with many other data science packages inside the Python ecosystem and is typically included in every Python distribution, such as Anaconda. It is particularly useful for time series and tables operations. As financial data is often organized in time-series format, pandas provide a convenient and powerful way to retrieve the data (such as daily stock prices) and then to manipulate the data to obtain information such as mean and standard deviation of the stock returns.

The following codes below illustrates an exercise on how to obtain the stock prices of Microsoft (MSFT), Apple (AAPL), and Amazon (AMZN) from 1 January 2022 to 31

December 2022, to store them into a DataFrame (a two-dimensional, size-mutable, and potentially heterogenous tabular data), and to print a graph containing the movement of the three stocks between the periods outlined.

The three tasks can be achieved through importing the relevant packages such as pandas, pandas_datareader, yfinance, numpy, and matplotlib (as the codes below show).

Exercise 1: Obtain financial data and print the graph of stock prices

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
from pandas_datareader import data as pdr
import yfinance as yf

yf.pdr_override()

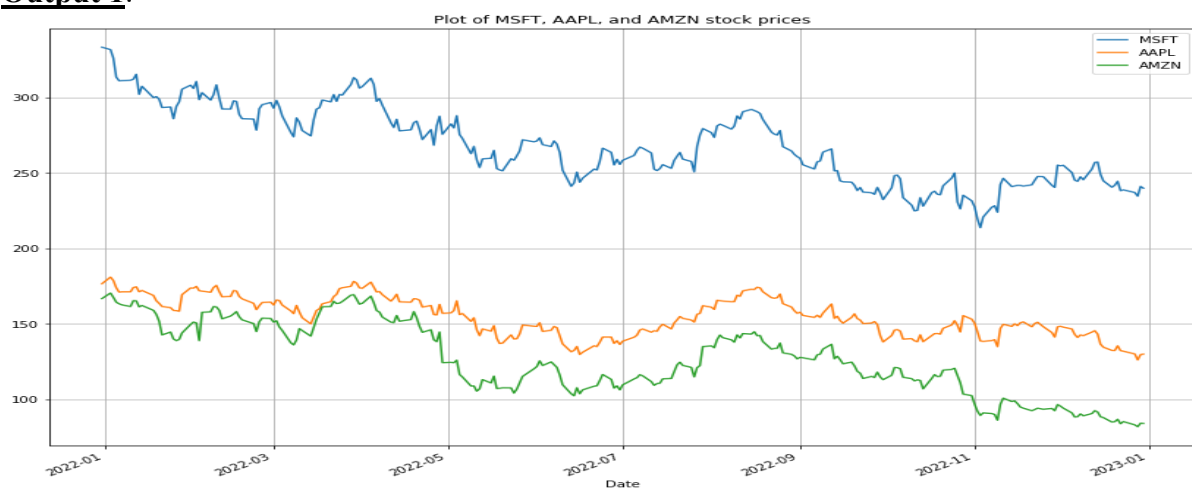
# Specify the starting and ending dates for the time series
start = datetime(2022, 1, 1)
end = datetime(2022, 12, 31)

# The first argument is the series we want, second and third are the starting and ending dates
msft = pdr.get_data_yahoo('MSFT', start, end)
aapl = pdr.get_data_yahoo('AAPL', start, end)
amzn = pdr.get_data_yahoo('AMZN', start, end)

# Select only the adjusted closing price data and convert the data into a DataFrame
stocks = pd.DataFrame({'MSFT': msft['Adj Close'],
                      'AAPL': aapl['Adj Close'],
                      'AMZN': amzn['Adj Close']})

# Plot the respective graphs
stocks.plot(title='Plot of MSFT, AAPL, and AMZN stock prices', figsize=(14, 10), grid=True)
```

Output 1:

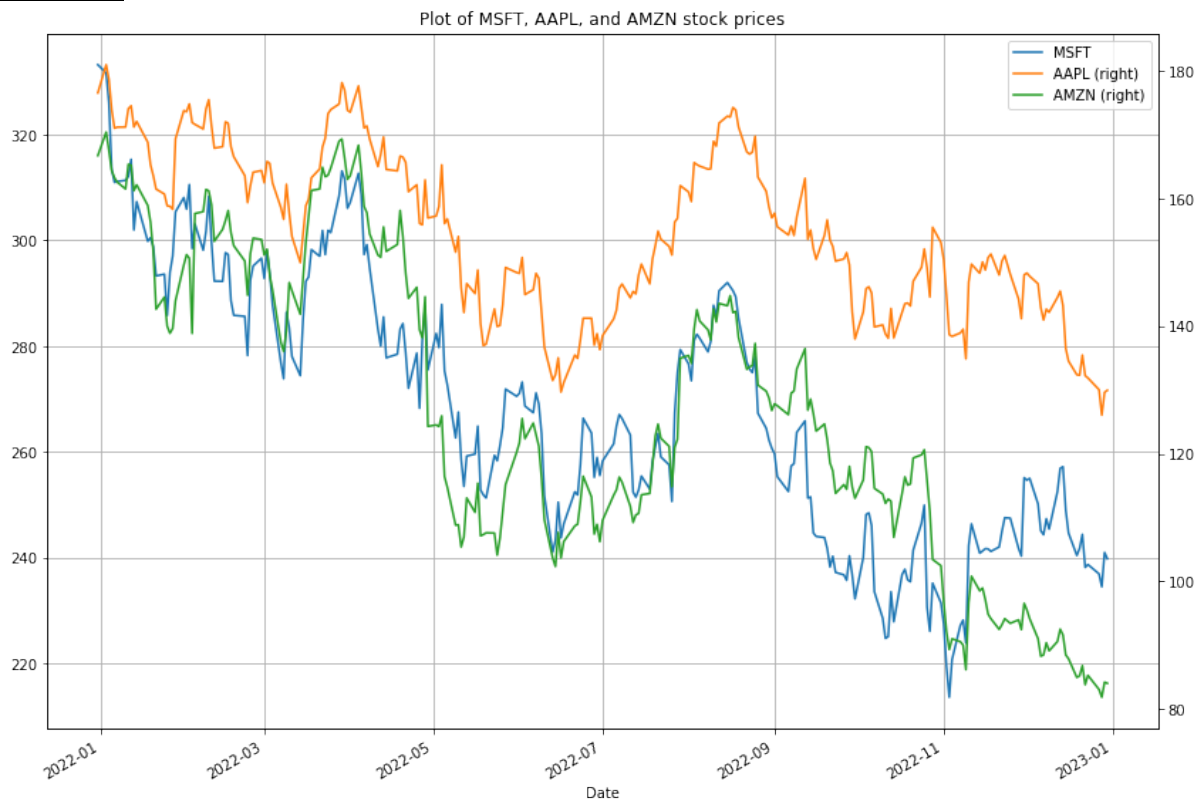


The next exercise for students will be to observe the output above and to point out how to improve the presentation of graphs. The solution is to use two different scales when plotting the data: one scale used for AMZN and AAPL, and another scale used for MSFT. The codes would have been taught to them in the earlier course on “Programming with Data”

Exercise 2: Better presentation of output using different scale

```
stocks.plot(secondary_y = ['MSFT', 'AAPL'], title = 'Plot of MSFT, AAPL and AMZN stock prices', grid = True)
```

Output 2:



Stage 3: Financial analytics using Python

Having learnt how to obtain stock price data and visualize them, Stage 3 is for students to learn how to perform financial analytics on the data that has been obtained using Python. In Exercise 2 below, students will learn how to present the relevant summary statistics for the 3 stocks extracted from Yahoo! Finance earlier. They are also expected to separately present the mean, standard deviation skewness and kurtosis of the distribution of the 3 stocks, rounding the output to 4 decimal places. The tasks can be achieved by important scipy and using the various methods such as describe(), aggregate(), kurtosis(), and skew() to present the relevant statistics required.

Exercise 3: Present summary statistics of the three stocks

```
import scipy
from scipy.stats import skew, kurtosis

# Presenting the summary statistics, rounding to 4 decimal places
stats = round(stocks.describe(), 4)
print(stats)
print('\n')

# Presenting the mean and standard deviation only
aggregate = round(stocks.aggregate([np.mean, np.std]), 4)
print(aggregate)
print('\n')

# Presenting the skewness and kurtosis
kurtosis = round(stocks.kurtosis(),4)
skew = round(stocks.skew(),4)
print('Skewness of the stocks:\n', skew)
print('\n')
print('Kurtosis of the stocks:\n', kurtosis)
```

Output 3:

```
count      MSFT      AAPL      AMZN
mean      267.8030    154.4699    126.2600
std       25.3036     12.8957     23.9935
min       213.6479    126.0400     81.8200
25%      246.4271    144.4749    108.9047
50%      264.7404    153.8273    122.8850
75%      287.7927    165.2807    146.2599
max       333.1979    180.9597    170.4045
```

```
mean      MSFT      AAPL      AMZN
std       25.3036     12.8957     23.9935
```

```
Skewness of the stocks:
MSFT      0.2254
AAPL      0.0130
AMZN      0.0466
dtype: float64
```

```
Kurtosis of the stocks:
MSFT     -0.8385
AAPL     -1.0143
AMZN     -1.0784
dtype: float64
```

Next, in Exercise 4 below, students learnt how to compute monthly returns for the 3 stocks obtained earlier, using the `resample()` and `pct_change()` functions, and to bar charts of the monthly returns.

Exercise 4: Compute simple monthly returns and present the bar charts

```
# Computing percentage change (simple return) and round to 4 d.p.
daily_return = stocks.pct_change()
monthly_return = stocks.resample('M').ffill().pct_change()

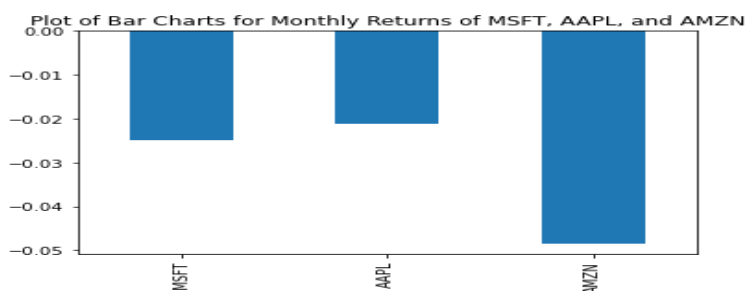
# Presenting the mean and std dev of daily and monthly returns
daily_stats = round(daily_return.aggregate([np.mean, np.std]), 4)
print(daily_stats)
print("\n")

monthly_stats = round(monthly_return.aggregate([np.mean, np.std]), 4)
print(monthly_stats)

# Plotting Bar Chart for monthly returns all 3 stocks
monthly_return.mean().plot(kind = 'bar', title = 'Plot of Bar Charts for Monthly Returns of MSFT, AAPL, and AMZN');
```

Output 4:

```
mean      MSFT      AAPL      AMZN
std       0.0695     0.0950     0.1263
```



In Exercise 5 below, students further learn how to compute the cumulative logarithmic returns and the correlations of the logarithmic returns between the 3 stocks as well as plot the histogram as the scatterplot diagrams. The functions used are the cumprod(), corr(), and plotting.scatter_matrix().

Exercise 5: Compute the cumulative logarithmic returns and correlation among the stocks and present the cumulative compounded log returns, histogram and scatterplot diagrams.

```
# Computing the logarithmic return
log_return = np.log(stocks/stocks.shift(1))
log_return = log_return.replace(np.nan, 0)

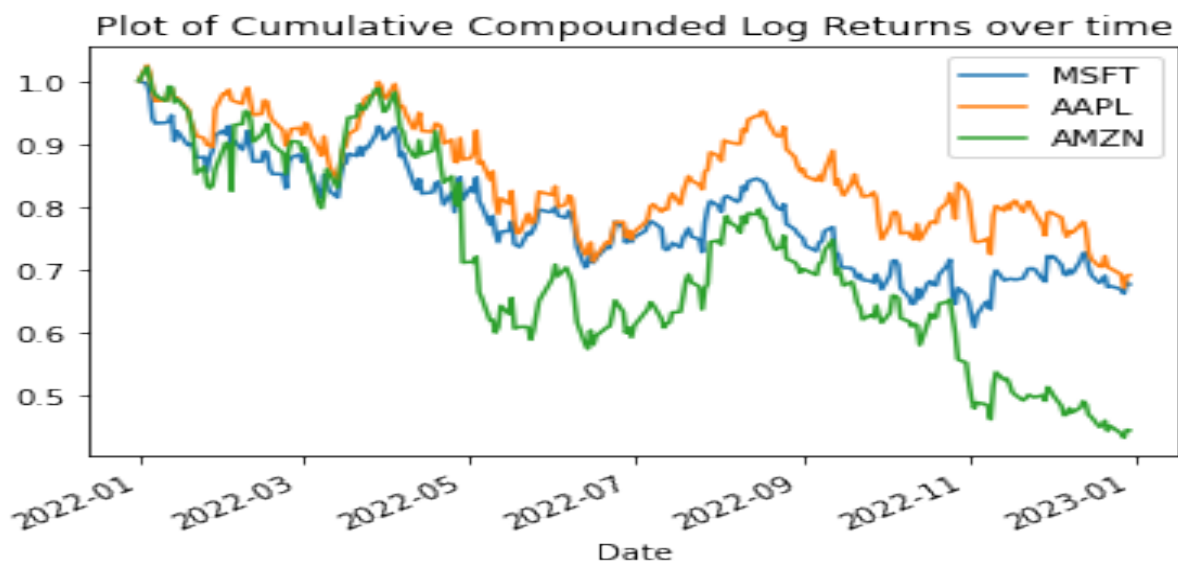
# Plotting the cumulative logarithmic returns over time; use the cumprod() method to compute
cum_comp_log_return = (1 + log_return).cumprod()

# Plotting the cumulative returns of all stocks
cum_comp_log_return.plot(title = 'Plot of Cumulative Compounded Log Returns over time')
plt.show();

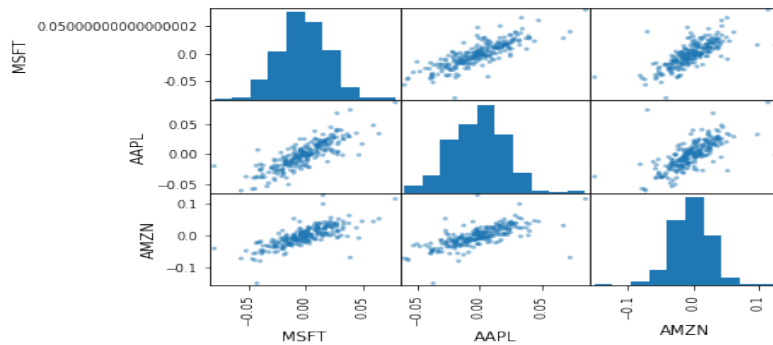
# Plotting the Scatterplot Diagram and Compute and print the correlation matrix, and round up to 4 decimal places
pd.plotting.scatter_matrix(log_return, diagonal = 'hist')

log_return.corr()
```

Output 5:



Histogram and Scatterplot Diagram



Correlation of the 3 stocks:

	MSFT	AAPL	AMZN
MSFT	1.000000	0.821637	0.741830
AAPL	0.821637	1.000000	0.699181
AMZN	0.741830	0.699181	1.000000

Stage 4: Monte Carlo Simulation using Python

Stage 4 requires applications of financial analytics into advance topics, such as Monte Carlo Simulation. In Exercise 6 which is illustrated below, an example using the APPL (Apple, Inc) stock is used whereby we will use the stock prices from January 1 to December 31, 2022 to predict the one year ahead stock prices. The Monte Carlo simulator generates theoretical future 'r' values. Because rate of return is a random number, to model the movement and determine possible future values, we must use a formula that models the random movement – called the **Brownian motion** (applied in the Black-Scholes formula also) Annual changes in the stock price (i.e. return) = expected growth rate over time + the effect of constant volatility of people randomly buying and selling the stock over time has on that expected growth. Therefore, we can model return as follows:

log return = fixed drift rate+random stochastic value

$$\ln\left(\frac{S_t}{S_{t-1}}\right) = \alpha + \sigma * Z_t$$

$$r = \exp(\text{drift} + \text{stochastic term})$$

Exercise 6: Monte Carlo Simulation of AAPL stock price

```
import pandas as pd
import numpy as np
import datetime as dt
import matplotlib.pyplot as plt
from datetime import datetime
from matplotlib import style
from scipy.stats import norm

from pandas_datareader import data as pdr
import yfinance as yf
yf.pdr_override()

# Specify the starting and ending dates for the time series
start = datetime(2022, 1, 1)
end = datetime(2022, 12, 31)
data = pd.DataFrame()
data['AAPL'] = pdr.get_data_yahoo('AAPL', start, end)['Adj Close']

# Computing logarithmic returns of AAPL
log_returns = np.log(1 + data.pct_change())

# Computing the drift component
u = log_returns.mean()
var = log_returns.var()
drift = u - (0.5 * var)

# Measure the dispersion component
stdev = log_returns.std()

# Creating the random component
Z = norm.ppf(np.random.rand(10,2))

# How far into the future we want to go (365 days) and number of simulations (10)
t_intervals = 365
iterations = 10

# Generating a series of random future returns
daily_returns = np.exp(drift.values + stdev.values * norm.ppf(np.random.rand(t_intervals, iterations)))

# Getting the last traded price of AAPL
S0 = data.iloc[-1]

# Create an array of zeroes that follow the same array as the daily returns
price_list = np.zeros_like(daily_returns)
price_list[0] = S0

# Forecasting daily prices for 1 year ahead
for t in range(1, t_intervals):
    price_list[t] = price_list[t - 1] * daily_returns[t]

# Plotting the Predicted Stock Prices of AAPL over the next 1 year
plt.figure(figsize=(10,6))
plt.title("Plot of AAPL Predicted Stock Prices")
plt.ylabel("Price (P)")
plt.xlabel("Time (Days)")
plt.plot(price_list)
plt.show()
```


Output 6:

