

International Professional Teacher Education

Practical Teacher Training in an Educational Institution

Learning Diary

Häme University of Applied Sciences



Hamed Ahmadinia

DOI: 10.5281/zenodo.15281939

© 2025 Hamed Ahmadinia / CC BY-NC-ND 4.0

First edition, 2025

Copyright © 2024–2025 Hamed Ahmadinia

This learning diary was prepared as part of the International Professional Teacher Education Programme at Häme University of Applied Sciences, Hämeenlinna, Finland.

All content in this publication is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

You are free to share, copy, and redistribute the material in any medium or format under the following terms: – **Attribution** must be given to the original author. – **No commercial use** is permitted. – **No modifications or derivatives** may be made.

Some images in this publication have been generated by Pyxl.pro. Text has been reviewed for language and grammar accuracy by Quillbot.com.

 **CC BY-NC-ND 4.0**

Published by Hamed Ahmadinia



Hamed Ahmadinia

Previous studies and degrees

PhD in Information Studies, 2024

MSc in Economics and Business Administration, 2017

MSc in Business Administration – Finance, 2010

BSc in Tax Accounting, 2007

Preface

This learning diary is a reflective synthesis of my journey through the Practical Teacher Training (PTT) module, part of the International Professional Teacher Education (IPTE) programme (60 ECTS) at Häme University of Applied Sciences (HAMK), Finland.

It documents not only the theoretical frameworks and institutional policies that shape the work of vocational teachers in Finland, but also the pedagogical planning, implementation, and reflection involved in delivering a real-world course—*Data Analytics & Statistics in Python*—at Metropolia University of Applied Sciences.

The diary is structured to follow the multi-phase requirements of the PTT module, beginning with an analysis of educational policies, national strategies, and institutional guidelines that influence teaching practices in the Finnish higher education context. Through both individual reflections and collaborative peer discussions, I have explored how these macro-level frameworks translate into the micro-level realities of teaching in a diverse, inclusive, and innovation-driven learning environment.

The second part of this diary is practice-oriented: it details the design and delivery of an eight-week data analytics course tailored for adult learners from multidisciplinary backgrounds. The sessions were guided by pedagogical principles such as constructivism, scaffolding, and learner-centred instruction. The course integrated tools like Jupyter Notebooks, Python libraries, and project-based learning to foster interactivity, applied understanding, and student engagement.

The included observation reports provide insight into the structures and strategies used in Finnish vocational and higher education. My self-reflections highlight the evolution of my teaching philosophy—one that values inclusion, active participation, and lifelong professional growth.

Ultimately, this diary serves not only as a formal record of my PTT training but also as a personal testament to the transformative power of reflective teaching practice. It emphasizes the alignment of educational design with institutional values, learner diversity, and real-world relevance—principles I am committed to upholding in my future career as an educator.

I would like to express my sincere gratitude to Vesa Parkkonen, Senior Lecturer at Häme University of Applied Sciences and my tutor teacher, and to Janne Salonen, Director School of ICT and Industrial Management at Metropolia University of Applied Sciences and my supervising teacher, for their guidance, support, and encouragement throughout this training period. Their professionalism, expertise, and mentorship have been truly invaluable.

Hamed Ahmadinia

Turku, Finland

April 2025

Contents

Preface	iii
1 Boundaries and Responsibilities of a Vocational Teacher	1
1.1 Phase 1: Observing Norms, Guidelines, and Institutional Policies in Education	1
1.2 Phase 2: Reflections and Key Perspectives in Teaching	3
1.3 Phase 3: Study Group Discussion Summary	4
1.4 References	6
2 Observation of Teaching	7
2.1 Session 1 (17.03.2025)	7
2.1.1 Artificial Intelligence with Python (3 ECTS) at Metropolia University of Applied Sciences	7
2.1.2 The teacher structured, planned, and scheduled the session:	7
2.1.3 Teacher Guiding Students: How?	8
2.1.4 How Did the Teacher Support Learners?	8
2.1.5 Overall Observations	8
2.2 Session 2 (24.03.2025)	9
2.2.1 Introduction to Artificial Intelligence (3 ECTS) at Metropolia University of Applied Sciences	9
2.2.2 The teacher structured, planned, and scheduled the session:	9
2.2.3 Teacher Guiding Students: How?	9
2.2.4 How Did the Teacher Support Learners?	10
2.2.5 Overall Observations	10
2.3 Session 3 (31.03.2025)	11
2.3.1 Python Programming (3 ECTS) at Metropolia University of Applied Sciences	11
2.3.2 The teacher structured, planned, and scheduled the session:	11
2.3.3 Teacher Guiding Students: How?	11
2.3.4 How Did the Teacher Support Learners?	12
2.3.5 Overall Observations	12
3 Practical Training	13
3.1 Introduction	13
3.1.1 Course Structure	14
3.1.2 Core Content	15

3.1.3	Complementary Knowledge	15
3.1.4	Specialist Knowledge	15
3.1.5	Course Outcomes	16
3.1.6	Course Assessment	16
3.1.7	Grading Scale	16
3.1.8	Mini-Project Instructions and Requirements	17
3.2	My Teaching Philosophy	18
3.3	Designing a Practice-Based Data Analytics Course	22
3.4	Lesson Diary	24
3.4.1	Designing Session 1: Course Kick-off & Python Recap	24
3.4.2	Designing Session 2: NumPy, Pandas & Time-Series Intro	24
3.4.3	Designing Session 3: Descriptive Statistics + Handling Missing Data	25
3.4.4	Designing Session 4: Probability, Distributions & Hypothesis Testing	26
3.4.5	Designing Session 5: Correlation & Regression	26
3.4.6	Designing Session 6: Data Visualisation + Visual Design Ethics	27
3.4.7	Designing Session 7: Mini-Project Walk-through & GitHub Workshop	27
3.4.8	Designing Session 8: Final Presentations, Feedback & Wrap-Up	28
3.5	Course Meeting	29
3.5.1	Contact Lesson 1 (5.3.2025)	29
3.5.2	Contact Lesson 2 (12.3.2025)	33
3.5.3	Contact Lesson 3 (19.3.2025)	34
3.5.4	Contact Lesson 4 (26.3.2025)	35
3.5.5	Contact Lesson 5 (2.4.2025)	36
3.5.6	Contact Lesson 6 (9.4.2025)	38
3.5.7	Contact Lesson 7 (16.4.2025)	40
3.5.8	Contact Lesson 8 (23.4.2025)	42
4	Teacher Training Outcomes and Feedback	44
4.1	Final Mini-Project Outcomes	44
4.2	Modes of Teaching Outcomes	45
4.3	Student's Feedback	47
4.4	My Counter Feedback	54
Appendix		56
	Appendix I: Student Background Survey	56
	Appendix II: Course Evaluation Survey	58
	Appendix III: Final Mini-Project Presentation Schedule	60
	Appendix IV: Peer Feedback via Zoom Polls	62

List of Figures

1	Course slide used to introduce the constructivist, hands-on learning approach.	18
2	Jupyter Notebook used to demonstrate Python coding and data analysis.	19
3	Covers of recommended books used to support learning.	19
4	In-notebook activities and coding exercises to reinforce theory.	20
5	Kahoot quiz to reinforce Boolean operations in Python.	20
6	Custom cheat sheet provided to help learners recall core Python operations.	21
7	Education Level of Participants	29
8	Field of Study or Work	30
9	Previous Programming Experience	30
10	Familiarity with Python Concepts	30
11	Experience with Jupyter Notebook	31
12	Data Analysis Experience	31
13	Completed Data-Related Tasks	31
14	Need for Extra Resources	32
15	Word cloud of keywords from student mini-projects	36
16	Distribution of Final Grades	44
17	Distribution of Total Mini-Project Scores	45
18	Average Scores per Mini-Project Component	45
19	Participants' Highest Educational Level	47
20	Familiarity with Python Before the Course	47
21	Overall Course Content Rating	48
22	Achievement of Learning Objectives	48
23	Organization and Clarity of Course Content	48
24	Relevance of Topics to Career Goals	48
25	Most Valuable Topics Selected by Students	49
26	Instructor's Practical Explanation and Application of Topics	49
27	Instructor's Accessibility and Responsiveness	49
28	Timeliness and Helpfulness of Feedback Provided	49
29	Support Channels Utilized by Students	50
30	Student Rating of Course Structure	50
31	Perceived Relevance of Hands-On Exercises	50
32	Challenge Level of the Course	50
33	Effectiveness of Online Learning Format	51
34	Technical Difficulties Experienced	51
35	Student Engagement During Live Sessions	51

36	Accessibility and Usefulness of Online Resources	51
37	Clarity of Mini-Project Instructions	52
38	Final Presentations and Reinforcement of Learning	52
39	Likelihood of Recommending the Course	52
40	Overall Satisfaction with the Course	52
41	Suggestions for Course Improvements	53
42	Word Cloud of Free-Text Feedback	53

List of Tables

1	Course Implementation Summary	13
2	Course Schedule	14
3	Summary of Mini-Project Requirements	17
4	Comparison of Original vs. Enhanced Course Versions	23

Chapter 1

Boundaries and Responsibilities of a Vocational Teacher

1.1 Phase 1: Observing Norms, Guidelines, and Institutional Policies in Education

The policies within the Finnish educational context are guiding structures by which it is ensured that quality education is integrative and under constant development. This reflection will mirror how those features shape teaching activities and the operational practices of educational institutions through a case study at Metropolia University of Applied Sciences. Observation of legislation guidelines regarding teacher employment terms, strategic goals, and funding models shows how setting such standards supports personal and institutional growth within education.

Legislative Framework and Key Educational Guidelines

The Finnish system of education legislates equal opportunities, inclusivity, and excellence. These national standards projected to set an agenda for Metropolia University are very instrumental in influencing educational policies to focus on quality and equality from primary to higher education. The Finnish Education Law stipulates that lifelong learning and societal values, as founding elements of Finland's integrative and democratic system, need to be emphasised (Paksuniemi & Keskitalo, 2019). For instance, Metropolia applies an inclusive education standard as set by Finland by incorporating support for the different backgrounds and abilities of students, which automatically means that teachers provide learning environments that support a range of learning needs (Metropolia University of Applied Sciences, 2024a).

In the equity policy dimension, Finland has worked out clear policies that ensure that everybody accesses education, addressing all socio-economic, linguistic, and regional disparities. Statistics support this commitment: over recent years, Finnish educational outcomes have retained their high performance, with almost 90% of students completing upper secondary education or equivalent by age 25. This completion rate testifies to Fin-

land's commitment toward the idea of education for all—an idea to which Metropolia is no exception in its inclusive educational practice (Paksuniemi & Keskitalo, 2019). Policies that support a common school culture, integral to the mission of Metropolia, further ensure such an inclusive approach within all practices (Metropolia University of Applied Sciences, 2024c). It therefore follows that educational practice and guidelines for teachers at Metropolia reflect wider priorities within Finnish education and provide students with supportive and engaging adaptive learning.

Teacher Employment Terms and Professional Orientation

Recruiting teachers in Finland is based on providing strong support for them through different programs and initiatives right from the beginning of the career into professional life. According to Metropolia University of Applied Sciences, a systematic orientation process is underlined as being of particular importance for new teachers since it eases their adaptation into institutional culture and practices without reducing teaching effectiveness (Metropolia University of Applied Sciences, 2024b). These terms comply with Finnish employment policy for teachers and set the framework for professional development opportunities, supplemented by continuous pedagogical support.

Institutional values and expectations are inculcated in Metropolia through orientation programs, reflecting national norms of encouraging teacher reflectivity and adapting practice. Quality teaching is enhanced with an enabling environment promoting understanding by teachers of the need for adaptability in teaching diverse student groupings. Continuous professional development in Finland ensures that quality teaching will be not only maintained at the highest level but also responsive to changing educational needs. Indeed, according to research findings, this approach is effective: about 97% of Finnish teachers update their skills every year, which reflects in high teaching standards (Metropolia University of Applied Sciences, 2024d). The terms of employment at Metropolia also nurture the desire and inspire teachers to embark on professional development activities aimed at the enhancement of their skillsets and ensuring that their skills match the institutional requirements as well as the national standards of education.

Government Projects and Strategic Educational Initiatives

Nationally, Finnish educational strategies emphasise innovation and adaptability, aligning closely with Metropolia's strategic vision for future education. Finnish government projects, particularly those focused on digital transformation and sustainability, influence institutional strategies by guiding resource allocation and setting benchmarks for quality and innovation (Paksuniemi & Keskitalo, 2019). For example, Finland's national digitalisation initiatives aim to enhance learning outcomes by integrating technology into educational practices. Metropolia responds to this initiative by continually developing its digital learning resources, ensuring accessibility and promoting learning efficiency.

Furthermore, Metropolia's strategic plans reflect these broader educational trends by setting goals for technological advancement and adopting research-driven teaching methodologies. The EFQM (European Foundation for Quality Management) model—a widely

adopted quality management system in Europe—provides Metropolia with a structured approach to assess and enhance its teaching and administrative processes. This model ensures that Metropolia’s educational practices are aligned with the national emphasis on quality and accountability (Metropolia University of Applied Sciences, 2024e). Finland’s educational institutions have a robust tradition of quality assurance, and EFQM helps maintain this by encouraging institutions like Metropolia to adopt continuous improvement strategies that benefit both teachers and students.

Funding Models and Institutional Resources

Finland’s educational institutions rely on a public funding model that prioritises high-quality, accessible education. Metropolia University, like other Finnish universities, receives public funding, which allows it to offer superior educational services while maintaining low or no tuition fees for EU citizens (Metropolia University of Applied Sciences, 2024c). Publicly funded resources enable Metropolia to maintain inclusivity and provide access to advanced educational tools and technologies, supporting a diverse student body.

In addition to public funding, project-based grants and strategic initiatives further the institution’s goals. Specific grants focused on improving digital infrastructure and support services enable Metropolia to address emerging educational needs while staying competitive in an evolving educational landscape (Metropolia University of Applied Sciences, 2024a). Funding models that blend public funding with targeted grants allow Metropolia and similar institutions to focus on inclusivity and innovation without budgetary constraints, thereby encouraging a consistent standard of excellence in educational practices.

Data from recent educational surveys also highlight the efficiency of Finland’s funding model. The Finnish government allocates approximately 5.5% of its GDP to education, which supports both academic and vocational training throughout the country. This funding commitment contributes to Finland’s status as one of the world leaders in education, reflected in Finnish students’ strong performance in international assessments (Metropolia University of Applied Sciences, 2024b).

1.2 Phase 2: Reflections and Key Perspectives in Teaching

Based on my reflections related to learning from the observed norms, guidelines, and policies at Metropolia, I am equally interested in how such frameworks identify what role a teacher would play in this educational institution. Inclusiveness, adaptability, and quality are basic values of Finnish education and substantial cornerstones of my pedagogical way of thinking. Applying Metropolia’s guidelines for teaching methods will mean flexibility toward the students, with a view to establishing an interactive and supportive learning environment, which, in turn, may provide opportunities for reflection in line with reflective and inclusive practice. I will try to introduce feedback loops; this is promoted both by the institution and generally by Finnish educational norms.

Continuous professional development is also very strongly put across in Metropolia's orientation and training programs. Because one believes in lifelong learning, this will be an arena where the author will express personal commitment. I will read current journals on educational advancements and government initiatives in order to maintain a sense of how new ideas should be implemented and how different types of students can be better served. For example, through the EFQM model, systematic quality assurance in teaching was intimated; this appealed, as it resulted in orderliness in refining teaching practices.

Further, Finland's commitment to inclusiveness finds reflection in both Metropolia strategic goals and resource allocation concerning equitable access to quality education. An understanding of how the funding model and the institutional policies ensure this inclusivity further serves to inform my approach to teaching, underscoring the necessity for an interactive, adaptive classroom environment. This is corroborated by statistics: for instance, a recent survey shows that approximately 90 percent of Finnish educators are of the opinion that their institution effectively promotes equality (Metropolia University of Applied Sciences, 2024a).

Equality in education goes hand-in-glove with my values and aims as far as education and my choice of an educational institution are concerned. This therefore means that institutional and national policies underline the need for inclusion, flexibility, as well as entrepreneurship in education. Indeed, these considerations will not only inform my pedagogy but also go further to ensure that the classroom climate I strive to establish is one in which all the students, irrespective of their background, are taken into consideration and actively involved in it. At Metropolia, I would like to draw not only on the resources available at this institution for continued professional advancement but also seek to contribute to positive change through enhancing the learning experiences for my students.

1.3 Phase 3: Study Group Discussion Summary

On November 29, 2024, I, Hamed, had the pleasure of participating in a lively online meeting with Isaac and Mark from 10 AM to 12 PM. Our discussion revolved around some significant topics, specifically how legislation, funding, and policies shape vocational and higher education. We touched on several key areas: Legislation, Norms, and Policies; Funding Models; Educational Policy and Future Directions; and Institutional Preparedness. Although Jessica couldn't join us due to a scheduling conflict, she shared her thoughtful reflections later in the evening at 8 PM. Here's a breakdown of our discussion and insights.

Legislation, Norms, and Policies:

The Finnish education system is built on the principles of equality and lifelong learning, and Metropolia University of Applied Sciences is a shining example of this ethos. They champion inclusivity through tailored support for diverse learners and comprehensive teacher induction programs. Finland's national commitment to professional development is evident,

with 97 percent of teachers participating in annual skill updates, ensuring high standards across the board.

Mark provided insights into Ateneo de Manila University (ADMU) in the Philippines, where compliance with laws like the Anti-Hazing Act and Data Privacy Act ensures student safety and data protection. ADMU also stands out for its Ateneo Laudato Si' Plan, which integrates sustainability into academics and aims for carbon neutrality by 2030—a bold step toward fostering ecological responsibility. Isaac highlighted the Finnish vocational education system's alignment with EU standards, which emphasizes inclusivity and quality assurance. Haaga-Helia University of Applied Sciences exemplifies these values by adhering to national legislation and offering vocational programs that are industry-aligned and inclusive.

Funding Models:

Public investment in education - the driver for inclusiveness and quality in Finnish education. Since the government funding is attached to the number of outgoing students, Metropolia also uses competitive funding instrument to develop digital tools, enhance support services, and also ensure equal opportunities for student participation. It keeps our university innovative and responsive when it comes to the ongoing changes in demands for higher educations. Mark explained that ADDM diversified its funding via tuition fees, donations, and partnered institutions. These resources underpin scholarships for needy students, developments in infrastructure, and other initiatives such as the Ateneo Center for English Language Teaching. Alumni donations and partnerships play a critical role in maintaining ADMU's high standards. Isaac discussed Haaga-Helia's funding structure, which combines government support, grants, and business partnerships. This mix enables the university to meet legislative requirements while prioritizing equity and access for its students.

Educational Policy and Future Directions:

Metropolia demonstrates Finland's commitment to digitalization and sustainability by offering free courses through its Open University of Applied Sciences. These initiatives remove financial barriers, making lifelong learning accessible to a diverse range of students, including working professionals and international learners. Mark highlighted ADMU's forward-looking approach, including its transition to co-education by 2029 to foster inclusivity and equality. Programs like its Artificial Intelligence Competition integrate liberal arts with technology, preparing students to tackle complex societal and technological challenges. ADMU's alignment with the UN Sustainable Development Goals underscores its focus on addressing global issues. Isaac emphasized Haaga-Helia's strategic alignment with industry trends. The university prioritizes sustainable development, digitalization, and competency-sharing, equipping students with the skills they need to meet future workforce demands and societal shifts.

Institutional Preparedness:

Metropolia is outstanding in adapting to rapid educational changes, focusing on digital learning, and inclusivity to meet the future of both teachers and students. Mark explained how ADMU was ready for the COVID-19 pandemic: The university implemented Emergency Remote Teaching, or ERT, supported by faculty training, with strong cybersecurity measures in place. Its Campus Emergency Management Plan further showcases its preparedness to address natural disasters and other crises and ensure community welfare. Isaac said, "Haaga-Helia collaborates on an international level quite extensively, there is an exchange too. Its continuing education programs support people in updating their professional and work-related knowledge and to compete in the rapid change faced in the employment market."

Group member's reflections:

Later in the evening, one of the group member gave her views on the discussion. She admired how Metropolia worked with national policies to make education accessible and inclusive. She was very impressed with ADMU's innovative funding strategies that strike a balance between inclusivity with infrastructure development and academic excellence. Lastly, she commended Haaga-Helia's internationalization and how the education aligns with industry needs to make sure students are prepared for global challenges.

1.4 References

- Metropolia University of Applied Sciences. (2024a). Academic information. Retrieved October 28, 2024, from <https://www.metropolia.fi/en/academics/general-information-on-studies/manage-your-studies/academic-information>
- Metropolia University of Applied Sciences. (2024b). Rules and regulations. Retrieved October 25, 2024, from <https://www.metropolia.fi/en/academics/applying/rules-and-regulations>
- Metropolia University of Applied Sciences. (2024c). Homepage. Retrieved October 26, 2024, from <https://www.metropolia.fi/en>
- Metropolia University of Applied Sciences. (2024d). Good study practices. Retrieved October 26, 2024, from <https://www.metropolia.fi/en/academics/general-information-on-studies/good-study-practices>
- Metropolia University of Applied Sciences. (2024e). Language guidelines. Retrieved October 26, 2024, from <https://www.metropolia.fi/en/about-us/language-guidelines>
- Paksuniemi, M., & Keskitalo, P. (Eds.). (2019). *Introduction to the Finnish educational system*. Brill. <https://doi.org/10.1163/9789004394278>

Chapter 2

Observation of Teaching

2.1 Session 1 (17.03.2025)

2.1.1 Artificial Intelligence with Python (3 ECTS) at Metropolia University of Applied Sciences

View course description

This is my observation report for a three ECTS credits course, asynchronous online in artificial intelligence with Python at Metropolia University of Applied Sciences. Delivered via the VIPELEARN platform, Janne Salonen planned and oversaw this non-stop, self-contained course. The self-paced course allows students to finish learning modules on their own, within a large and flexible period, therefore allowing different schedules and learning rates.

2.1.2 The teacher structured, planned, and scheduled the session:

Janne Salonen, the instructor, divided the course into eight basic modules, each advancing from simple Python programming towards increasingly sophisticated artificial intelligence ideas, including regression, categorisation, and algorithmic reasoning. Emphasising fundamental Python programming, the session covered data types (e.g., strings, lists, sets, and tuples); control structures (e.g., if, else, for, and while); user-defined functions; and basic input/output operations. Every subject was presented as a well-crafted, text-based course augmented with hands-on coding projects and tests. This scaffolded approach lets students reinforce and absorb important programming ideas before moving into more abstract artificial intelligence subjects, therefore promoting slow cognitive development. The workshop materials followed a clear learning path: they start with basic Python knowledge and logically advance towards building the algorithmic thinking needed for machine learning uses.

2.1.3 Teacher Guiding Students: How?

The teacher presented a well-defined and encouraging framework despite the asynchronous style. Students must complete at least 80 percent of the course activities and turn in a learning journal (maximum two A4 pages) to qualify for course credit. This is a welcome note at the start of the course outlining expectations and evaluation criteria. A specific course support email (viopesupport@metropolia.fi) was created to help students who might struggle or lag behind. Starting from basic operations (e.g., $1 + 2$, `'Hello' * 3`) and progressively introducing more complicated constructions, including nested loops, custom function definitions, exception handling, and modular programming, the advice was buried within the learning path through scaffolded examples. Students were urged to put their knowledge to use creating interactive projects requiring control structures, user input, and list manipulations—such as a grocery list manager and a supermarket checkout simulator. These assignments were formative tests as well as teaching aids.

2.1.4 How Did the Teacher Support Learners?

Many gamified components—a points-based system, graphic progress bars, and a public leaderboard—promoted student participation. These tools let students track their degree of task completion and view their relative performance to their peers, therefore fostering a healthy competition and natural inspiration. Every workout was learner-centred, contextual, and pragmatic. One could find examples like:

- a method of string filtering depending on their length
- a tailored bubble sort system
- Split and join activities carried out without using built-in Python tools
- Exercises in object-orientated programming combining method overloading and inheritance

These assignments improved students' logical reasoning, debugging techniques, problem-solving ability, and core Python knowledge. The emphasis on practical programming situations inspired even more innovation and experimentation.

2.1.5 Overall Observations

Emphasising active learning, structured autonomy, and real-world applicability, the observed course session showed great pedagogical planning. Thoughtfully designed to empower independent learners, the asynchronous arrangement provided modular materials, clear goals, and several means of feedback and reinforcement. For students who rely on aural or visual modalities, the lack of video lectures may present difficulties; nevertheless, the well-organised written resources, scaffolded activities, and offered support systems provide strong basis for success. The general learning environment promotes student agency and profound knowledge of programming and artificial intelligence foundations.

2.2 Session 2 (24.03.2025)

2.2.1 Introduction to Artificial Intelligence (3 ECTS) at Metropolia University of Applied Sciences

View course description

This is my observation report for a recorded, self-paced online course called Introduction to Artificial Intelligence (NonStop), provided at Metropolia University of Applied Sciences. Aimed at offering a conceptual and application-orientated overview of artificial intelligence and machine learning for beginners, this ongoing course requires neither prior programming nor mathematics skills. Especially for non-technical students, the course materials are designed to progressively deepen fundamental knowledge while making artificial intelligence understandable.

2.2.2 The teacher structured, planned, and scheduled the session:

From the definition and history of artificial intelligence to the present relevance of artificial intelligence, machine learning concepts, and practical applications, the session I saw offered basic material spanning several related topics. Beginning with simple, daily examples, the material was arranged logically into a series of chapters that progressively introduced core artificial intelligence ideas—beginning with intuitive, everyday examples and later moving into key concepts including types of machine learning (supervised, unsupervised, reinforcement) and an introduction to adversarial machine learning.

The learning materials were created by the teacher so that students could reflectively study on their own. Supported by visual metaphors (such as the often-used nested-circle model of AI, ML, and DL), images, and optional links to external readings or tutorials, most of the material was provided in written form. Particularly for students with less technological backgrounds, the course design was customised to fit different learning preferences and obviously followed the ideas of asynchronous learning.

2.2.3 Teacher Guiding Students: How?

Neither direct teacher-student contact nor real-time interaction defined the course. Rather, it offered an indirect but powerful teaching presence using structured material blocks, short reflection prompts, and multiple-choice tests. The teaching method stressed clarity over technical complexity, accessibility, and practical relevance.

The course regularly employed analogies—for example, contrasting how babies learn from trial and error to how algorithms learn patterns from data—to help students understand fundamental ideas. Using diagrams like the one shown in Figure 1 to highlight their hierarchical relationship, the differences between artificial intelligence, machine learning, and deep learning were precisely both graphically and textually reinforced.

Key pedagogical strategies included clearly defined learning objectives at the start of each module; text-based instruction enhanced with simple, relevant visuals; spreadsheets-

based interactive examples in place of coding; short quizzes following each section to reinforce conceptual understanding

The training encouraged critical and moral thought as well. Real-world vulnerabilities—such as how little image changes can trick advanced artificial intelligence systems; therefore, increasing awareness of the limits and hazards inherent in present artificial intelligence technologies—were shown using adversarial machine learning.

2.2.4 How Did the Teacher Support Learners?

Regular embedded tests spanning significant thematic portions (e.g., AI vs. ML vs. DL, ML kinds, Watson’s history) helped to encourage student drive. These tests were formative assessments as well as means of reinforcement for knowledge. Everyday examples—such as Netflix suggestions, spam filters, and AI-powered customer care bots—helped keep the material interesting and familiar.

Students also received optional links to outside sites, including the UCI Machine Learning Repository or Kaggle. Supporting alternative learning paths, these links inspired more advanced students to investigate datasets and engage in practical tasks. These instruments gave value for people trying to enhance their knowledge, even though they were not required.

The course deliberately worked to lower the entrance requirements. Originally point-and-click substitutes for conventional coding environments, tools like Azure ML Studio help to demystify artificial intelligence. The inclusive format and examples guaranteed accessibility for students from non-STEM backgrounds or without prior computer science knowledge.

2.2.5 Overall Observations

The course was a deliberately inclusive and well-chosen introduction to artificial intelligence and machine learning. Combining ethical issues, intellectual explanations, and pragmatic examples, it presented a fair view. The course’s design and pedagogy clearly promoted autonomous, self-paced learning even without a live instructor.

Visual tools, analogies, spreadsheets, and tests enabled simple scaffolding of challenging concepts. By including subjects like bias, data ethics, and adversarial artificial intelligence, students were pushed to consider the societal effects of artificial intelligence in addition to how it operates. The asynchronous style combined with easily available resources and learning methodologies made the course the perfect starting point for anyone from many disciplinary and professional backgrounds interested in artificial intelligence.

2.3 Session 3 (31.03.2025)

2.3.1 Python Programming (3 ECTS) at Metropolia University of Applied Sciences

View course description

Below is my observation report for a recorded and self-paced online course titled Python Programming (NonStop), offered through Metropolia University of Applied Sciences. This course introduces the fundamentals of Python 3 programming and serves as a foundational entry point for students across multiple study backgrounds, particularly those new to Python or transitioning from earlier programming languages such as Python 2. The course aims to demystify Python programming through structured, accessible content and low-threshold exercises suitable even for learners with only minimal prior programming exposure.

2.3.2 The teacher structured, planned, and scheduled the session:

The course is clearly designed with beginner and transitioning learners in mind. The introductory session begins with a detailed overview of Python's evolution from version 2 to version 3, focusing on key syntax changes and usability improvements, such as the adoption of the `print()` function, more consistent input handling, and clarification of string formatting and division behaviour. The session was structured into coherent thematic segments, presented in a logical, digestible order: 1. Introduction to Python and its major updates 2. Comparison with other programming languages 3. Applications of Python in real-world settings 4. Installing and using the Python interpreter 5. Getting to know the interactive shell 6. Writing and running the first Python program 7. Programming exercises (text output and syntax awareness) 8. Multiple-choice conceptual checks The content sequence allowed students to first build conceptual knowledge—understanding why certain elements work as they do—before moving to practical implementation. This pedagogical flow supports meaningful learning by anchoring syntax and tools in broader programming logic and purpose.

2.3.3 Teacher Guiding Students: How?

Although the course is delivered asynchronously, it provided clear, accessible instruction and step-by-step guidance. Explanations were presented in a conversational tone and enhanced with example code, making key actions—such as generating output, managing indentation, handling errors, and defining simple GUIs with Tkinter—approachable and engaging. The instructor guided learners through:

- Basic syntax and commands (`print`, variable assignment, `input`, etc.)
- Interactive shell usage and experimentation (`>>>`)
- Use of beginner-friendly integrated development environments like IDLE or PyCharm
- Programming tasks with structured feedback and visible expected outputs

For instance, in the palindrome checker exercise, the slicing technique `[::-1]` was introduced in a way that

showcased Python’s expressive power and simplicity. The code was explained with clarity, emphasizing Python’s strengths in string manipulation and readability.

2.3.4 How Did the Teacher Support Learners?

Student engagement was fostered through incremental, confidence-building exercises such as:

- Creating a simple “Hello World!” program
- Testing system setup with `import sys`
- Resolving common syntax and indentation errors
- Multiple-choice quizzes to reinforce understanding

Learners were encouraged to experiment early on by editing and running their own Python scripts directly in the Viope environment. The teacher emphasized Python’s low entry barrier, clean syntax, and community-driven philosophy. Introducing the Zen of Python added a motivational and philosophical element, framing the learning journey within broader programming values like readability, simplicity, and elegance. Importantly, the course also helped manage learner anxiety by normalizing common issues—like garbled output characters—as cosmetic rather than critical. This supportive tone helped reduce the fear of failure often felt by first-time coders.

2.3.5 Overall Observations

This introductory session demonstrated effective pedagogical planning and thoughtful support for novice learners. The session provided hands-on experience with Python tools while ensuring that students could not only read but also write and execute simple programs. Real-world applications such as GUI development and text reversal gave meaningful context to abstract syntax rules. Although the course is text-heavy, its step-by-step approach, annotated examples, and integrated exercises made the content accessible and engaging. By blending practical tasks with clear explanations and immediate feedback, the course succeeded in building both competence and confidence in Python programming—an essential outcome for any beginner-level technical course.

Chapter 3

Practical Training

3.1 Introduction

Data Analytics & Statistics in Python course was implemented as part of my International Professional Teacher Education (IPTE) education at Häme University of Applied Sciences (HAMK). As part of this training, I, Hamed Ahmadinia, designed and delivered this course at Metropolia University of Applied Sciences during spring 2025.

The course introduced students to core concepts in Python-based data analytics, emphasising statistical reasoning and real-world applications. Topics included data wrangling, visualisation, hypothesis testing, predictive modelling, and cryptocurrency market analysis. The teaching period consisted of eight live online sessions, held from 05.03.2025 to 23.04.2025, and was supervised by Janne Salonen, Head of Competence Area at Metropolia UAS. A summary of the course information is provided in Table 1.

Table 1: Course Implementation Summary

Course Code	DATA.STATS.101 (internal code)
Course Name	Data Analytics & Statistics in Python
Institution	Metropolia University of Applied Sciences
Implementation Year	2024–2025
Contact Period	05.03.2025 – 23.04.2025
Number of Credits	3 ECTS
Number of Students	39 registered students
Supervising Teacher	Janne Salonen
Moodle Page	Metropolia OMA Workspace
Repository	Course GitHub

3.1.1 Course Structure

The *Data Analytics & Statistics in Python* course spanned eight weeks, comprising live, two-hour online sessions held every Wednesday from 16:00 to 18:00 (EET). Designed with working professionals and students in mind, the course followed a flexible yet intensive format that balanced theoretical foundations with practical, hands-on learning in Python. Each session combined:

- Short theoretical lectures to introduce core concepts
- Interactive exercises using real-world datasets
- Live coding in Jupyter Notebooks to reinforce the application of Python and statistical methods

Participants gradually progressed from a refresher on Python fundamentals to more advanced topics such as statistical modelling, data visualization, and time series forecasting. The course culminated in a capstone mini-project, where students independently explored a dataset of their choice (e.g., cryptocurrency, healthcare, or environmental data) and applied the full data analytics pipeline—from preprocessing to interpretation and presentation. Pedagogically, the course employed blended learning, scaffolding, and active learning techniques to foster engagement and deeper understanding. Tools and libraries such as NumPy, pandas, Matplotlib, and Seaborn were introduced contextually and reinforced through project-based learning. The structure of each session was aligned with both the Finnish higher education learning objectives and practical industry needs. The detailed weekly breakdown of session topics and learning goals is provided in Table 2.

Table 2: Course Schedule

Date	Topic	Learning Goal
05.03.2025	Python Recap	Master basic Python concepts and syntax
12.03.2025	Matrix and Data Frames	Manipulate data using NumPy and Pandas
19.03.2025	Statistics: Theory and Application	Understand key statistical metrics
26.03.2025	Probability and Variability	Learn probability distributions and significance tests
02.04.2025	Relationships Between Variables	Explore regression and correlation concepts
09.04.2025	Data Visualization	Create effective visualizations with Matplotlib and Seaborn
16.04.2025	Advanced Topics and Integration	Apply various analytics methods to real-world problems
23.04.2025	Student Mini-Project Presentation	Present and discuss projects to demonstrate learning

3.1.2 Core Content

Key areas covered during the course included:

- **Python Programming Basics** – with NumPy, Pandas, Matplotlib, and Seaborn.
- **Descriptive and Inferential Statistics** – including measures of central tendency, variability, and hypothesis testing.
- **Data Visualization** – using best practices to communicate data-driven insights.
- **Probability Theory** – exploring binomial, Poisson, and normal distributions.
- **Regression and Correlation Analysis** – modelling and evaluating relationships between variables.
- **Cryptocurrency Data Analytics** – analysing trends and forecasting based on historical price data.

Weekly exercises enabled immediate practice and reinforcement of the concepts taught during each session.

3.1.3 Complementary Knowledge

To enhance the learning experience, the course incorporated:

- **Real-world data applications** – datasets included cryptocurrency markets, environmental data, health surveys, and global happiness indices.
- **Problem-based learning** – guiding students to investigate questions relevant to their interests.
- **GitHub integration** – fostering professional collaboration and version control skills.
- **Imputation and handling of missing data** – from basic methods to advanced KNN-based techniques.

These elements supported a diverse group of learners in connecting theoretical knowledge with practical tools and scenarios.

3.1.4 Specialist Knowledge

Through structured activities and guided exploration, students gained advanced skills in:

- Time-series data manipulation and forecasting
- Multivariate regression and model evaluation
- Z-score computation, t-tests, Mann–Whitney U, and Chi-Square tests
- Residual analysis and model diagnostics
- Creative communication of analytical findings through visualizations

These advanced analytics techniques were especially emphasized in the final mini-projects, where learners worked independently or in teams on domain-specific problems.

3.1.5 Course Outcomes

Upon completing the course, students were able to:

- Analyse datasets using Python libraries such as Pandas, NumPy, Seaborn, and Matplotlib
- Interpret descriptive and inferential statistical results
- Create meaningful data visualizations and regression models
- Apply statistical reasoning to real-world questions
- Present a full mini-project involving exploration, analysis, and visualization using GitHub and Jupyter Notebook

3.1.6 Course Assessment

The course evaluation was based on both formative and summative assessment methods. The assessment structure was aligned with authentic learning principles and aimed at developing practical data analytics competence:

- **Hands-on Exercises (30%)** – Weekly assignments were designed to reinforce the concepts introduced in each session. These exercises focused on real-world datasets and allowed students to apply their skills incrementally.
- **Final Mini-Project (70%)** – As the capstone of the course, students completed a mini-project where they analysed a dataset of their choice (e.g., cryptocurrency, health, or environmental data). This project required integration of all course elements—from data preparation to visualization and statistical modelling—and was presented during the final session.

3.1.7 Grading Scale

The final grade was determined according to the following scale:

- **90–100** = Grade 5
- **80–89** = Grade 4
- **70–79** = Grade 3
- **60–69** = Grade 2
- **41–59** = Grade 1

The assessment included weekly hands-on exercises ($6 \times 5\% = 30\%$) and a final mini-project (**70%**), aligned with authentic learning principles and practical competence development criteria.

3.1.8 Mini-Project Instructions and Requirements

As the capstone of the course, students completed a mini-project either individually or in groups of up to three members. The project aimed to integrate the full range of analytics techniques covered throughout the course—from data loading and statistical analysis to visualization and (optionally) predictive modelling. Students were required to register their project by submitting a proposal via a designated form by **9.4.2025 at 12:00 PM**. The project instructions emphasised flexibility, real-world relevance, and alignment with students' academic or career interests. To support this, I provided a categorised PDF list to help my course participants select a dataset aligned with their personal or professional focus (e.g., finance, healthcare, marketing, environment). Each dataset had to be unique unless students were working together as a group.

To complete the project, my course participants were instructed to begin with dataset loading and exploration, examining the dataset's structure, data types, and any missing values. This was followed by descriptive statistics, including the calculation of measures such as the mean, median, and standard deviation, and identifying potential outliers. The analysis was to include time-based insights, identifying trends and patterns over time, along with data visualisation using Matplotlib and Seaborn. For students who had missed more than two exercises, an optional predictive modelling task—typically regression-based forecasting—was required as a compensation activity. All code had to be uploaded to a personal or group GitHub repository, and the submission link shared via the Metropolia portal by 22.4.2025 at 12:00 PM. Final presentations took place in our last session on 23.4.2025, where each student or group delivered a 5–10 minute presentation, either as a live code demo or a visual summary using 5–7 slides. A summary of these mini-project requirements is provided in Table 3.

Table 3: Summary of Mini-Project Requirements

Team Size	1–3 students
Registration Deadline	9.4.2025 at 12:00 PM
Submission Format	GitHub repository (.py or .ipynb), submitted via form
Submission Deadline	22.4.2025 at 12:00 PM
Presentation Date	23.4.2025
Presentation Length	5–10 minutes (code demo or 5–7 slides)
Dataset Requirement	Aligned with interest; unique per student/group
Compensation Option	Predictive modeling required if more than 2 exercises missed

3.2 My Teaching Philosophy

“Education is not to be viewed as something like filling a vessel with water but, rather, assisting a flower to grow in its own way.” — Bertrand Russell

I approach teaching as an invitation to discover, create, and grow. My goal is to cultivate a learning environment that is both supportive and intellectually stimulating, where learners are not merely passive recipients of information but active participants in the knowledge-building process.

In my *Data Analytics & Statistics in Python* course, I foster a hands-on, exploratory mindset, where students are encouraged to ask questions, test ideas, and learn through practice. My teaching style is anchored in constructivist pedagogy, emphasizing active problem-solving and incremental learning.

Data Analytics & Statistics in Python Session 1: Course Introduction



Learning data-driven decision-making with Python

Instructor: Hamed Ahmadiania, Ph.D.

Email: hamed.ahmadiania@metropolia.fi

1

Figure 1: Course slide used to introduce the constructivist, hands-on learning approach.

Rather than lecturing through theory alone, I focus on applying concepts through interactive coding, visual exercises, and practical interpretation of data stories. A typical class begins with a concise concept overview, followed by real-time coding in Jupyter Notebooks.

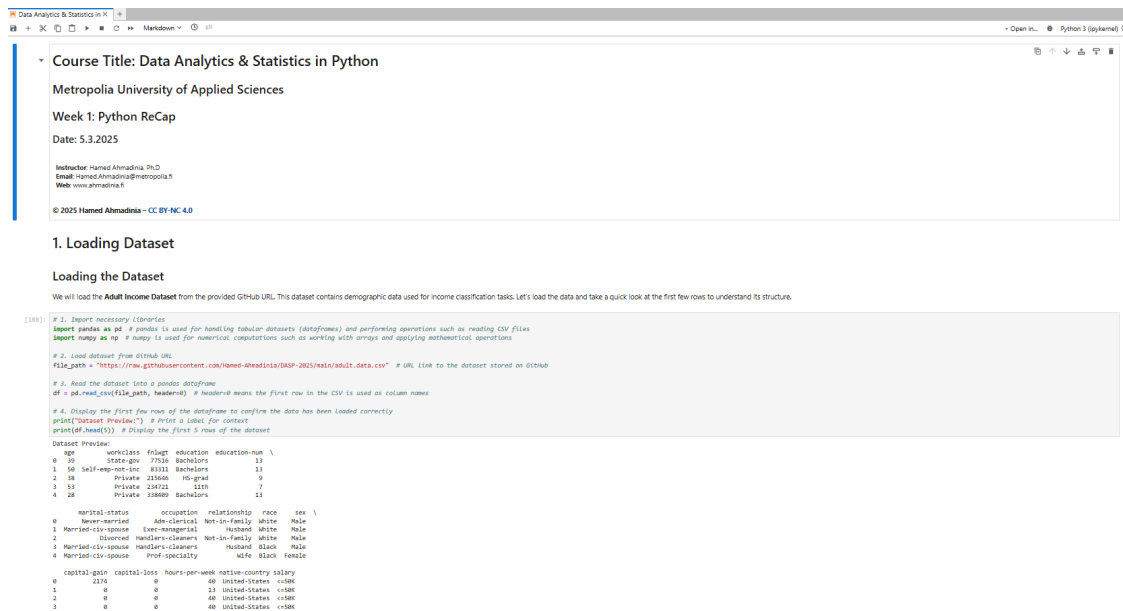


Figure 2: Jupyter Notebook used to demonstrate Python coding and data analysis.

To support different learning preferences and build autonomy, I integrate a carefully curated selection of recommended materials:

Books:

- Python Data Science Handbook BY Jake VanderPlas
- Practical Statistics for Data Scientists BY Andrew Bruce & Peter Bruce
- Python for Everybody Exploring Data Using Python 3 BY Charles Severance

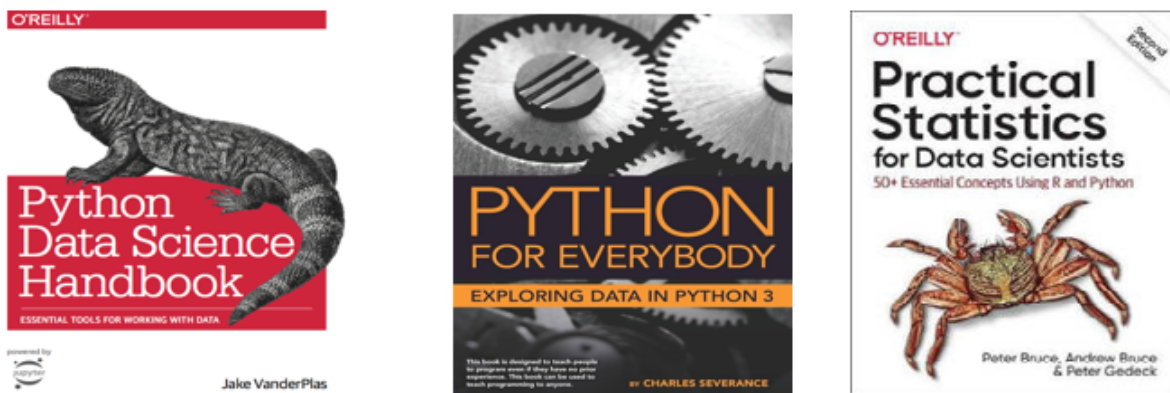


Figure 3: Covers of recommended books used to support learning.

Documentation & Platforms:

- Pandas.pydata.org, matplotlib.org
- Kaggle and GitHub for self-paced exploration and collaboration

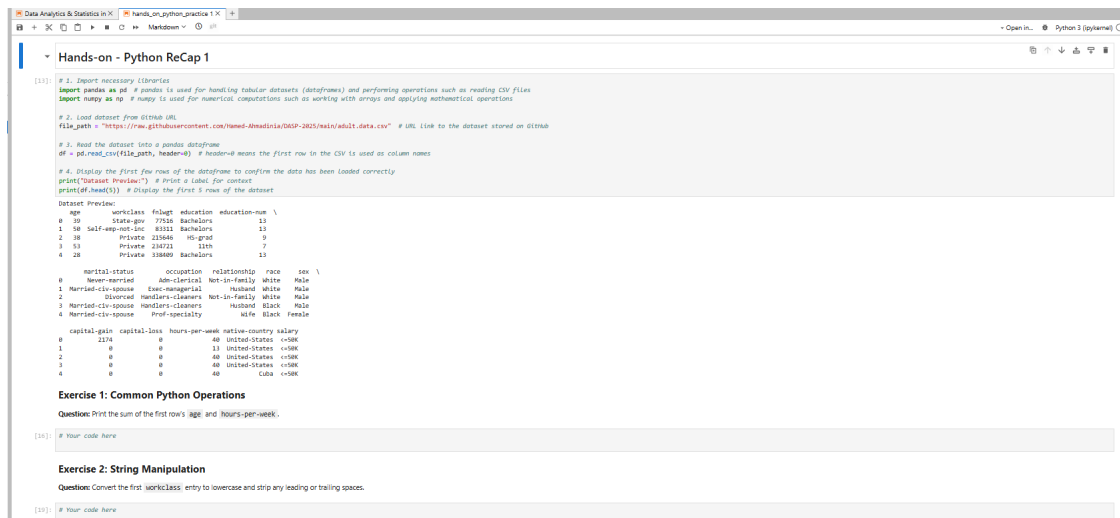


Figure 4: In-notebook activities and coding exercises to reinforce theory.

In class, I combine Jupyter Notebooks with interactive slides to demonstrate coding in real time. Students receive immediate feedback as they experiment with Python syntax, statistical tools, and visualizations. I also use quizzes on platforms like Kahoot to boost engagement and reinforce key concepts.

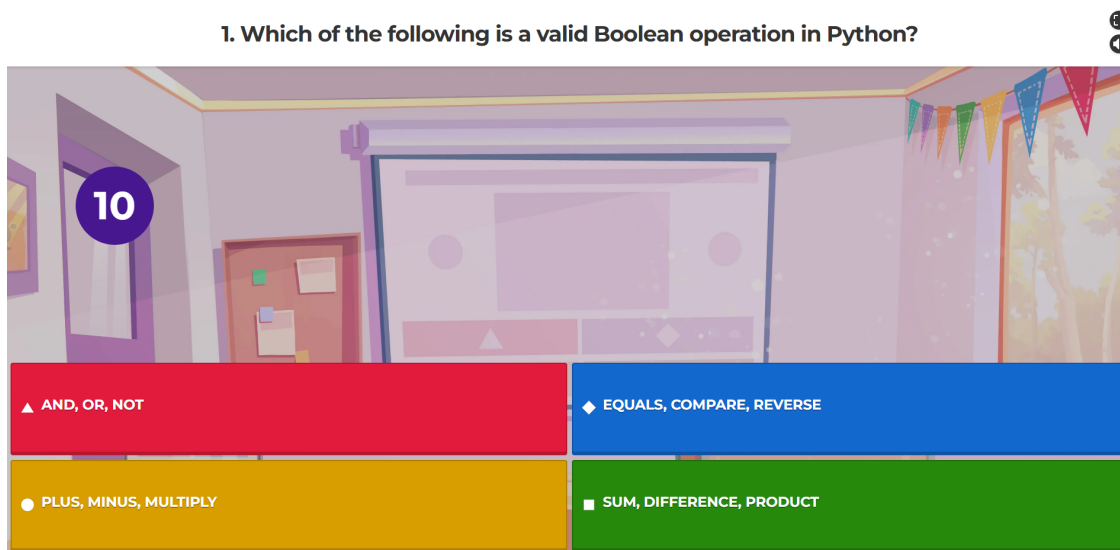


Figure 5: Kahoot quiz to reinforce Boolean operations in Python.

My instruction is shaped by storytelling—not as entertainment, but as a pedagogical tool to contextualize data. When discussing forecasting, we don't just analyze numbers—we explore cryptocurrency price movements or health behavior trends, anchoring abstract techniques in meaningful narratives.

Recap of Python Operations Cheat Sheet

Hamed Ahmadinia, Ph.D.

Hamed.Ahmadinia@metropolia.fi

Core Operations

Arithmetic:

- + `df['age'] + 5` → Add 5 to all ages
- - `df['capital_gain'] - 1000` → Deduct 1000
- * `df['hours'] * 2` → Double hours
- / `df['age'] / 2` → Float division
- // `df['qty'] // 10` → Integer division
- ** `df['score'] ** 2` → Square values
- % `df['index'] % 7` → Modulo operation

Comparisons:

- == `df['income'] == '>50K'` → High earners
- != `df['edu'] != 'Bachelors'` → Non-Bachelors
- > `df['age'] > 40` → Senior filter
- < `df['hours'] < 30` → Part-time
- >= `df['exp'] >= 5` → 5+ yrs experience
- <= `df['score'] <= 100` → Valid scores

Assignment:

- += `age += 5` → Increment age
- -= `balance -= 100` → Deduct balance
- *= `factor *= 1.1` → 10
- /= `total /= 2` → Halve total
- //= `items //= 5` → Batch grouping
- **= `base **= 3` → Cube base
- %= `index %= 10` → Cycle index

Dictionaries

Methods:

- `.keys()` → List all keys
- `.values()` → List all values
- `.get()` → `dict.get('key', default)`
- `.update()` → Merge dicts

Example:

```
employee = {  
    'name': 'Alice',  
    'dept': 'Data Science',  
    'projects': 5  
}
```

Control Flow

Conditionals:

```
if temp > 30:  
    category = "Hot"  
elif temp > 20:  
    category = "Mild"  
else:  
    category = "Cold"
```

Loops:

Figure 6: Custom cheat sheet provided to help learners recall core Python operations.

Above all, I see teaching as a co-learning journey. My role is to guide, support, and challenge learners as they transform curiosity into competence. Every class becomes a space where students build real-world skills with confidence, and where I, as an educator, continue to evolve through reflection, feedback, and shared discovery.

3.3 Designing a Practice-Based Data Analytics Course

When designing the *Python for Data Analytics & Statistics* course at Metropolia University of Applied Sciences, I began by drawing inspiration from the self-paced “non-stop” version created by Janne Salonen. His original syllabus provided a solid foundation, covering Python programming basics, essential statistical concepts, and data visualization techniques, so I kept these core elements as the backbone of my course. The structured sequence of topics—starting with a Python refresher, moving through statistics, and concluding with basic visualizations—also remained largely unchanged to ensure we met the expected learning outcomes. However, I approached the redesign with a student-centred mindset, aiming to transform it from a solitary self-study module into a vibrant, interactive learning experience that emphasized practical applications.

While I preserved the course’s main content areas (Python basics, data handling, statistics, and visualizations), I significantly enhanced the instructional design and delivery methods. My goal was to help students not only learn the theory but also actively engage with real datasets and develop confidence in applying what they learned. To achieve this, I introduced several key innovations and improvements:

Curated Real-World Data & Case Studies: I spent over eight hours selecting a variety of datasets from fields like healthcare, finance, and environmental science. Each week, students worked with authentic data—such as patient health metrics, stock market trends, or climate records—and tackled short case studies. This approach tied the abstract concepts of statistics and Python coding to tangible, real-world scenarios, making the learning experience more relevant and engaging.

Active Student Engagement: Every session was designed to be interactive. I used live polls and Kahoot! quizzes to gauge understanding in real time and energize the (online) classroom. Students also worked on hands-on coding challenges—like cleaning a dataset or computing a statistical measure—in small groups, and then we discussed the solutions together. This shifted learners from passive listeners to active problem-solvers and created a fun, collaborative atmosphere that encouraged participation.

Expanded Content & New Modules: Building on Salonen’s framework, I added modules to address key gaps and modernize the curriculum. In Session 3, for example, I introduced techniques for handling missing data—a crucial topic not fully covered in the original syllabus. In Session 6, we went beyond basic charting and examined examples of good vs. bad visualizations, teaching students design principles and how to avoid common pitfalls. By Session 7, we moved on to a collaborative GitHub project walk-through, where students learned about version control and how to present a data analysis project—practical skills that mirror real-world data science workflows.

Innovative Learning Resources: To accommodate different learning styles, I developed additional materials and used a design tool called Noppe to plan engaging session content. I created concise cheat sheets summarizing key Python/pandas commands and statistical formulas, and recorded short video explainers to clarify complex topics (e.g., interpreting outliers or understanding p-values). Students could revisit these resources at their own pace. I also gathered anonymous feedback after each session via online forms, allowing me to refine the course continuously based on their input.

Asynchronous Support & Guidance: Recognizing that some students might review the content on their own schedules, I made sure to offer support outside the live sessions. I posted detailed weekly announcements recapping what we covered and previewing the upcoming topics. These updates reinforced the week’s key takeaways, linked extra resources for practice, and set clear expectations for the next session. This inclusive approach ensured that students who couldn’t attend live or needed to catch up always felt supported and informed.

Overall, my aim was to retain the strong structure of the original course while injecting a more interactive, application-driven focus. Each week was carefully organized so that theory was immediately followed by practice, with an emphasis on student engagement—fostering questions, discussion, and collaboration. By the end of the course, students weren’t just completing tasks to reach 80% progress; they were building a mini-project, sharing their code on GitHub, and working with real datasets in meaningful ways. This evolution from a content-based course to a dynamic, student-centred experience stands out as one of the highlights of my teaching practice. Table 4 summarises the key similarities and differences in various aspects of the course.

Table 4: Comparison of Original vs. Enhanced Course Versions

Area	Original Course	Enhanced Version
Format	Self-paced online, no classes. Students progressed at own pace.	Weekly live sessions over 8 weeks with guided interaction.
Topics	Python basics, data structures, stats, distributions, visualization.	Same topics plus missing data handling, visual design, GitHub.
Real-World Data	Mostly theoretical or static examples.	Weekly case studies with real datasets (health, finance, climate).
Engagement	No interactive elements or group work.	Kahoot quizzes, group coding tasks, feedback on students codes.
Resources	Basic Viope materials only.	Cheat sheets, videos, Noppe session plans, GitHub + Jupyter.
Assessment	80% task completion + 2-page learning diary. Pass/Fail.	GitHub mini-project + weekly activities. Grading 1 to 5.
Teaching Style	Static, content-focused. Minimal instructor contact.	Active, feedback-rich, student-centred and practical.

This comprehensive redesign not only maintained the strengths of the original course (a solid grounding in Python and data analysis) but also enriched it with a pedagogically robust, student-centred experience. By blending the existing framework with new interactive elements, I created a learning journey that kept students motivated, supported, and closely connected to real-world data analytics. This introductory section of my teaching portfolio demonstrates how thoughtful course design can spark innovation, leading to more profound learning and higher student satisfaction.

3.4 Lesson Diary

3.4.1 Designing Session 1: Course Kick-off & Python Recap

I devoted significant time to preparing a welcoming, well-structured first session. I began by spending about three hours reviewing and updating the syllabus, weaving in modern learning goals and aligning each session with both essential theory and practical applications. I also hand-picked various datasets that could be referenced throughout the course, making sure each one matched the related weekly topics.

Challenge: Balancing the depth of Python fundamentals for beginners with more advanced data needs.

Solution: I created a “mixed-level” approach by introducing coding concepts at a measured pace and providing advanced resources for quick learners.

I then spent 1.5 hours creating a clear, engaging presentation that introduced the course structure, assessment model, and the primary learning platforms (Anaconda, VS Code, and Noppe), while clarifying expectations for participation and collaboration. I recognized students might have different levels of technical experience, so I demonstrated tool installation and setup live to ensure accessibility for everyone.

I followed this with 2.5 hours of work on a Jupyter Notebook that guided students through core Python concepts—variables, data types, conditionals, loops, and file handling—using live coding. I wanted to establish a “theory-to-practice” rhythm early on, so students would know what to expect in future sessions.

I also designed a 20-question Kahoot quiz (2 hours) to review the core Python refresher material, and I created a hands-on exercise (2 hours to design, plus 1 hour for solution walk-throughs) so students could immediately apply their skills. This blend of content, interaction, and guided practice helped everyone start the course with confidence and motivation.

3.4.2 Designing Session 2: NumPy, Pandas & Time-Series Intro

I dedicated three hours to creating a detailed slide deck that covered the foundational concepts of NumPy arrays—such as reshaping, indexing, and performing vectorized operations—along with the essentials of managing data in Pandas. During this preparation, I also designed a concise cheat sheet to help students quickly grasp key functions and commands. By emphasizing both array transformations (like reshape, stack, and concatenate) and the basics of merging Data Frames, I aimed to give learners a solid toolkit for handling real-world datasets..

Challenge: Ensuring students who are new to NumPy and Pandas weren’t overwhelmed by the syntax differences between these libraries and vanilla Python.

Solution: I provided multiple small, focused examples (e.g., array creation, Data Frame merging) before introducing more complex time-series operations. This incremental build

helped students adjust to new concepts gradually.

I then spent another three hours crafting a comprehensive Jupyter Notebook, focusing on how to group, merge, and resample time-series data. My goal was to show students the seamless integration between NumPy's numerical capabilities and Pandas' more advanced features, such as groupby for aggregation and merge for combining datasets. I introduced examples involving daily sales data, demonstrating how to resample by month or week, and I walked them through cleaning steps like handling missing timestamps.

I devoted two hours to designing hands-on tasks that reinforced these techniques—everything from filtering large arrays to merging multiple Data Frames based on common keys. After completing those tasks, I spent an hour creating thorough solution files so students could check their work. To wrap up my preparations, I spent two hours writing Kahoot questions that tested key ideas from the session. I also wrote three structured weekly announcements reminding everyone of upcoming deadlines, sharing troubleshooting tips for installation issues, and addressing frequently asked questions about time-series handling.

3.4.3 Designing Session 3: Descriptive Statistics + Handling Missing Data

I kicked off my Session 3 preparations by analysing the feedback from the first two sessions, noticing that many students struggled with incomplete or noisy datasets. In response, I spent about 1.5 hours designing a dedicated slide deck on handling missing data, which I presented alongside my standard two-hour session on core descriptive statistics. This extra material gave students a more holistic view of data cleaning and quality checks—both crucial for accurate analysis.

Challenge: Making sure students could grasp multiple imputation methods (mean, forward-fill, KNN) without confusion.

Solution: I showed side-by-side comparisons of different techniques using a single realistic dataset. This “one dataset, many methods” approach highlighted each method's pros and cons, preventing information overload.

I then dedicated three hours to building a Jupyter Notebook full of practical code snippets. First, I reviewed descriptive measures—like mean, median, and standard deviation—and then moved on to illustrate various imputation techniques: forward-fill, backward-fill, mean or median replacement, and even KNN-based methods for more advanced scenarios. Each method was paired with a realistic example (e.g., patient health metrics) to demonstrate when, and why, a certain approach might be more appropriate.

I used another two hours to design interactive exercises, allocating an additional hour to write clear solutions. My Kahoot quiz, which took about two hours to develop, zeroed in on the new concepts of missing data treatment and fundamental descriptive statistics. By the end of the session, students were better equipped to identify and correct issues in their datasets, ensuring cleaner inputs for subsequent analyses.

3.4.4 Designing Session 4: Probability, Distributions & Hypothesis Testing

I began Session 4 by designing two separate slide decks—a total of three hours of work—to address the dual focus on probability distributions (binomial, normal, and Poisson) and the fundamentals of hypothesis testing. One deck covered the theoretical side of distributions and their real-world use cases, such as predicting the likelihood of equipment failures. The other deck broke down the logic behind hypothesis testing into accessible language, making it easier for students to grasp the underlying assumptions and steps.

Challenge: Getting students comfortable with formal statistical methods while keeping the content approachable for those not majoring in statistics.

Solution: I drew on concrete examples (e.g., cryptocurrency data) and everyday analogies (like coin flips and dice rolls) to demystify abstract concepts such as p-values and significance levels.

I spent another three hours creating a Jupyter Notebook that showcased essential computations, from calculating expected values and Z-scores to performing t-tests and chi-square tests. To drive home the applicability, I included a “real-world” component where students analysed, for example, cryptocurrency volatility or monthly sales data—testing whether differences were statistically significant or mere random fluctuations.

I rounded out this session by preparing a Kahoot quiz (two hours) to reinforce the week’s material, and I took three hours to design a hands-on assignment that challenged learners to apply various statistical tests on sample datasets. These tasks included everything from verifying normality assumptions to interpreting p-values in practical contexts. The class emerged with a more robust understanding of when—and how—to apply statistical methods to their own projects.

3.4.5 Designing Session 5: Correlation & Regression

I devoted two hours to crafting slides that introduced both the conceptual foundation and real-world applications of correlation (Pearson and Spearman) and linear regression. I highlighted the importance of examining scatter-plots visually before jumping into correlation coefficients, showing how outliers or non-linear trends can skew results.

Challenge: Encouraging students to look beyond the raw correlation coefficient and consider data distribution, outliers, and confounding variables.

Solution: In the presentation, I used multiple datasets—some with obvious outliers, some with non-linear patterns—so students could see how correlation alone can be misleading without visual inspection and domain context.

I then spent three hours building a visually oriented Jupyter Notebook. Students walked through step-by-step examples of computing correlation coefficients, fitting a linear regression model, and analysing residuals. I placed a strong emphasis on interpreting results—such as distinguishing between correlation and causation—so that learners could better

assess the validity of their conclusions.

I also curated two short explainer videos on regression diagnostics, which I embedded in Moodle for asynchronous review. Creating the Kahoot quiz (two hours) helped me distil the essential takeaways: how to interpret r-values, p-values, and the slope in a regression equation. Finally, I took three more hours to design hands-on tasks and draft thorough solution guides, ensuring that each student left the session capable of performing and critiquing basic regression analyses.

3.4.6 Designing Session 6: Data Visualisation + Visual Design Ethics

I initiated Session 6 by setting aside 2.5 hours to build a slide deck exclusively focused on “good vs. bad” visualization practices. Drawing on real examples—some with distorted axes or misleading colour schemes—I illustrated how seemingly small design choices can drastically alter the audience’s interpretation of data. I also explained how to align chart types (bar, line, scatter, boxplot) with different data narratives.

Challenge: Students sometimes underestimate the impact of minor design flaws—like incorrect axis scales—on data interpretation.

Solution: I included side-by-side “before and after” slides, highlighting how a small tweak (e.g., fixing an axis range) can completely transform the takeaway message. This hands-on comparison made the lessons more memorable.

I then spent another three hours scripting a Jupyter Notebook to demonstrate practical tips for using Matplotlib and Seaborn effectively. Students learned how to identify common pitfalls, like cutting off zero on the y-axis or overcrowding a legend. Through guided redesign exercises, they practiced correcting these issues in real time, improving clarity and honesty in visual presentations.

I concluded my preparation by creating a Kahoot quiz and Zoom poll questions (two hours) that focused on ethical considerations—like how to responsibly highlight outliers—and by finalizing a hands-on exercise set (three hours total). These practice tasks walked students through a series of “fix this chart” scenarios, ensuring they could spot and rectify problematic visuals before sharing insights with stakeholders.

3.4.7 Designing Session 7: Mini-Project Walk-through & GitHub Workshop

I started by dedicating two hours to designing a structured, step-by-step project guide, aimed at helping students build a mini-analysis from scratch. Alongside it, I prepared a sample Jupyter Notebook that walked through data cleaning, exploration, and basic visualization. This resource served as a reference point so students could compare their own progress and see best practices in action.

Challenge: I initially developed a mini-project based on a Spotify dataset and designed my entire presentation around music-streaming data. However, after consulting with my supervising teacher at Metropolia, I was asked to pivot the core topic to a cryptocurrency

dataset—a shift that required me to quickly redesign the materials and integrate a fresh dataset into my presentation flow.

Solution: I reworked the project scope by mapping the original learning goals onto the crypto dataset, showing how time-series exploration, correlation checks, and regression can be applied just as effectively to crypto market data. This enabled me to keep my existing learning outcomes intact while giving students a valuable demonstration of how to adapt to real-world changes midstream.

I spent 2.5 hours developing a mini-lecture on effective documentation—highlighting how to annotate code cells, comment on crucial steps, and summarize findings. I also integrated a practical GitHub demo into this lecture, guiding the class through setting up a repository, committing changes, and sharing project links. These skills not only helped them collaborate in real time but also showcased industry-relevant workflows for version control.

To round out the session, I created a final checklist PDF (two hours) to ensure no student overlooked critical parts of the assignment, from explaining assumptions to properly labelling plots. I then prepared a few additional Markdown templates for README files, which took about an hour, and wrapped everything up with a short Kahoot quiz on Git and version control fundamentals (1.5 hours). By the end, students had the confidence to not just complete a project but to share and maintain it professionally.

3.4.8 Designing Session 8: Final Presentations, Feedback & Wrap-Up

I allocated 1.5 hours to designing a structured wrap-up presentation that guided the flow of the final session. It featured a summary of everything we had covered—from Python basics in Session 1 to advanced visualization strategies in Session 6—reinforcing how each piece fit into the broader learning journey. I also added a brief Q&A section so students could address last-minute questions before presenting their final projects.

I then created a 20-question Kahoot quiz (two hours) that spanned all the key concepts introduced throughout the course, from data cleaning and statistical measures to regression diagnostics and GitHub collaboration. This served as a fun, interactive review of the knowledge they had gained and a way to highlight areas that might need a quick refresher.

Finally, I developed a feedback rubric (1.5 hours) to guide my evaluation of each group's mini-project presentations. The rubric focused on clarity, correctness, creativity, and collaboration, helping me provide targeted, constructive comments. After each group shared its work, I facilitated a short peer-review segment, encouraging learners to celebrate each other's progress and exchange practical tips for real-world data analysis. By concluding the course this way, I aimed to leave everyone motivated and well-prepared to continue their journeys in Python and data analytics.

3.5 Course Meeting

3.5.1 Contact Lesson 1 (5.3.2025)

Lesson Summary: I opened the course, *Python for Data Analytics & Statistics*, by introducing the syllabus, the 8-week structure, the necessary tools (Anaconda, VS Code, Noppe), and our main datasets (income classification, medical records, forum analytics, cryptocurrency pricing). Prior to the session, I had students complete a background survey so I could gauge their experience with Python, programming, and data analytics.

We started with a quick Python recap, covering variables, operations, control flows, functions, and file handling—illustrated through a live Jupyter Notebook demonstration. Students also participated in a Kahoot quiz to test their comprehension.

Self-Reflection:

1. I presented about 30 slides, blending theoretical explanations with hands-on activities to keep the class engaged.

2. Participation was high; students asked questions via Zoom chat and took part in the Kahoot quiz. A few had issues installing Anaconda, which I helped resolve during the break.

3. I reminded them to work in groups of two or three for Hands-On Exercise #1 and submit it through Moodle. I also directed them to the Python Recap Cheat Sheet for independent review.

Student Background Survey Results: The survey shows that 56.7% of participants hold a Bachelor's degree, 36.7% finished high school, and a small number have Master's or PhDs. Their fields of study and work lean heavily toward IT, Data Science, or Computer Science—80% in total—while the remaining students come from Business, Engineering, or Math.

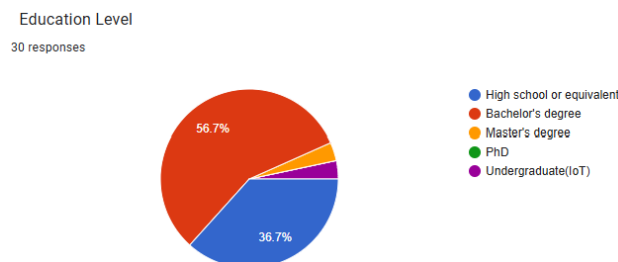


Figure 7: Education Level of Participants

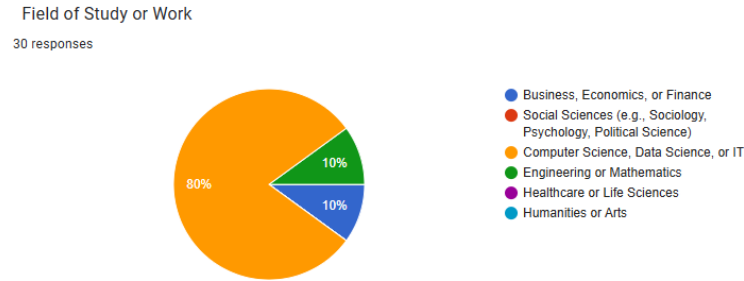


Figure 8: Field of Study or Work

In terms of programming experience, 86.7% have used Python before, 10% know other languages, and only one student is entirely new to coding. When asked about Python familiarity, 97% reported being comfortable with basic syntax, 76% with loops, 73% with conditionals, and 40% with file handling—while about 26.7% have worked with Pandas.

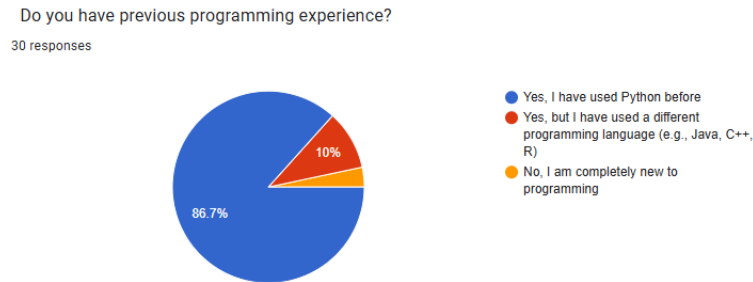


Figure 9: Previous Programming Experience

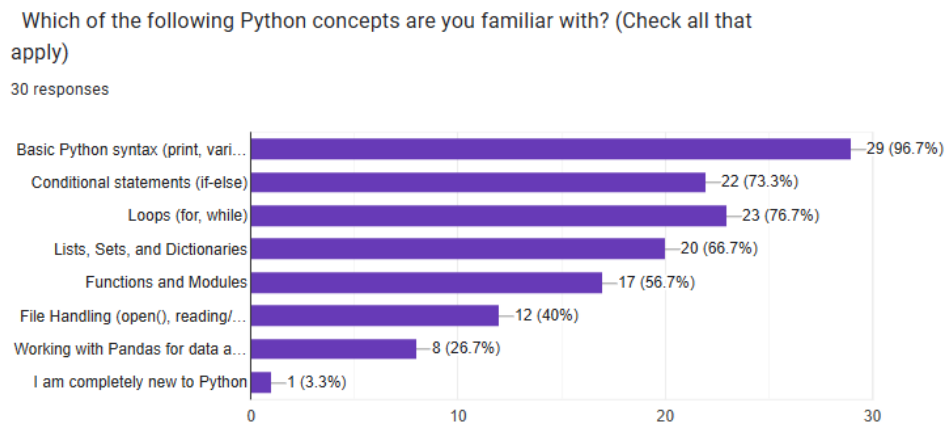


Figure 10: Familiarity with Python Concepts

As for hands-on skills and support, 36.7% feel confident with Jupyter Notebook, 40% need more practice, and 23.3% are new to it. Approximately 13.3% have practical data analysis experience, 36.7% need additional practice, and 50% are new—though many have tried summary statistics (53.3%), visualization (50%), and data exploration (40%).

Finally, 63.3% requested extra Python refresher materials, which I will make available on Moodle.

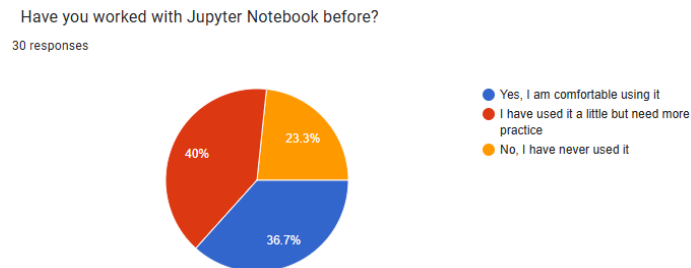


Figure 11: Experience with Jupyter Notebook

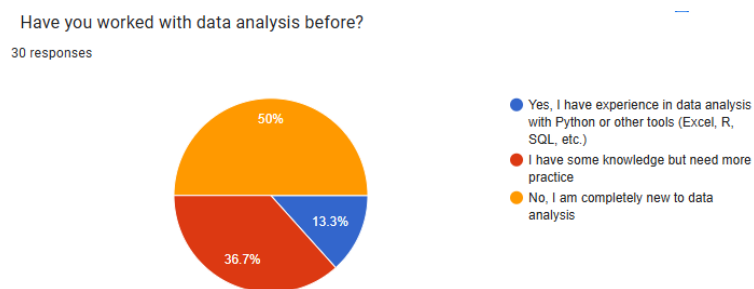


Figure 12: Data Analysis Experience

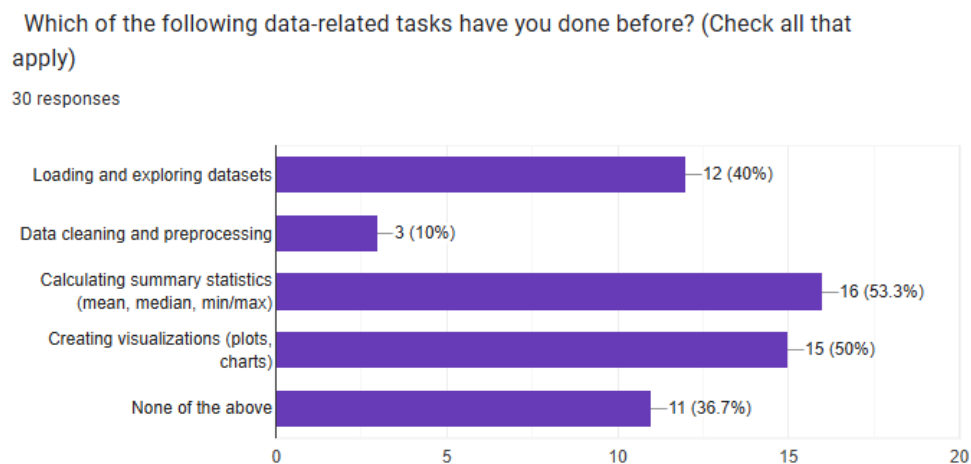


Figure 13: Completed Data-Related Tasks

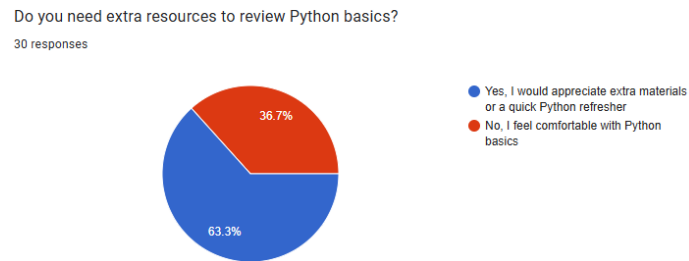


Figure 14: Need for Extra Resources

I also explain to student how they can get Support During the Course. I offered the three channels including Moodle Discussion for formal academic inquiries, Slack Channel for informal discussion, collaboration, and quick help, and Email Support for personal questions or direct assistance. I emphasised that learning together will enhance everyone's experience.

Attachments

- Slides 1: Session 1 Recap of Python operations.pdf
- Cheat Sheet 1: Python Recap Cheat Sheet.pdf
- Course Notebook 1: Week 1 Notebook.ipynb
- Assignment 1: Hands On Python Practice 1.ipynb
- Quiz: Kahoot – Python ReCap

3.5.2 Contact Lesson 2 (12.3.2025)

Lesson Summary: In this session, we explored foundational tools for data analysis, focusing on NumPy arrays and Pandas DataFrames. I demonstrated how these structures are used to store, manipulate, and analyze structured data efficiently. We covered reshaping arrays, filtering and editing DataFrames, and working with time-series data using datetime formatting and resampling techniques.

Students followed along in a Jupyter Notebook, completed a hands-on exercise, and participated in a 20-question Kahoot quiz to reinforce learning. The session materials, including the cheat sheet and notebook, are available on the course page.

Self-Reflection:

1. Students were highly engaged during the live coding and quiz segments. Several raised questions about array operations and data filtering.

2. I reminded them to complete Hands-On Exercise #2 and noted that 28 out of 39 students submitted their first assignment.

3. I also reintroduced our support channels: the Moodle discussion forum, the Slack group, and direct email support.

I also introduced the Mini-Project, which will allow students to apply skills gained throughout the course. Projects may be completed individually or in groups of up to three. All participants must submit the Project Registration Form by April 9, 2025, at 12:00 PM.

I explained that each group must select a unique dataset aligned with their interests (e.g., finance, healthcare, or social sciences). A list of recommended datasets is available in Moodle.

I elaborated that final projects are to be submitted via GitHub by April 22, 2025, at 12:00 PM. Presentations will take place on April 23, and each group will have 5–10 minutes to present.

I reminded students to create a GitHub account if they haven't already, and to follow the setup guide posted in the course workspace.

Attachments

- Slides 2: Session_2_Slides.pdf
- Cheat Sheet 2: Data Frame and Matrices Cheat.pdf
- Course Notebook 2: Week 2 Notebook.ipynb
- Assignment 2: Hands On Python Practice 2.ipynb
- Quiz: Kahoot – DataFrames and Matrices

3.5.3 Contact Lesson 3 (19.3.2025)

Lesson Summary: In this session, I shifted our focus from data structures to foundational statistical concepts in Python. I explained to students how to compute and interpret minimum and maximum values and different types of means (arithmetic, geometric, harmonic, and weighted), as well as the median and mode for central tendency analysis. I also explored quantiles and the interquartile range (IQR) for understanding data spread and detecting outliers.

I guided students through a Jupyter Notebook with real coding examples, complemented by a cheat sheet and a 20-question Kahoot quiz on descriptive statistics. In addition, I introduced bonus material on handling missing data, covering techniques from basic imputation to advanced machine learning-based approaches.

Self-Reflection:

1. The session was content-rich and interactive. Students were engaged with the statistical theory and its implementation using NumPy, Pandas, and SciPy.

2. I introduced the bonus missing data strategies, which sparked thoughtful discussion, especially on KNN and time-series imputation.

3. Out of 39 registered students, 26 submitted their second hands-on exercise by the deadline. I reminded the rest to complete and submit it as soon as possible.

I reminded all participants to complete the Mini-Project Registration Form by April 9, 2025, at 12:00 PM. Students may work individually or in groups of up to three, but each student must submit the form even if in a group. Some students had login issues, which were resolved by using their Metropolia accounts.

Attachments

- Slides 3: Session 3 Statistics in Python.pdf
- Bonus Slides 3: Handling Missing Data.pdf
- Cheat Sheet 3: Statistics in Python Cheat.pdf
- Course Notebook 3: Week 3 Notebook.ipynb
- Bonus Notebook 3: Handling Missing Data Notebook.ipynb
- Hands-On Assignment 3: Hands On Python Practice 3.ipynb
- Quiz: Kahoot – Statistics (theory and implementation)

3.5.4 Contact Lesson 4 (26.3.2025)

Lesson Summary: In this session, I focused on probability theory, variability, and hypothesis testing, which are essential for statistical reasoning in data analytics. I explored both discrete and continuous probability distributions, including binomial, Poisson, and normal distributions, as well as key concepts such as expected value, variance, standard deviation, and Z-scores. I also discussed statistical tests—such as t-tests, Mann-Whitney U, and Chi-squared—to help students evaluate differences and relationships in data.

To reinforce learning, I reviewed these concepts through Python code examples in Jupyter Notebook and supplemented them with a visual, beginner-friendly explanation of hypothesis testing. The session concluded with a Kahoot quiz covering 20 key questions from the lecture.

Self-Reflection:

1. Students engaged well with today's material, especially the hands-on Z-score examples and live statistical testing walk-through.
2. We discussed real-world scenarios like cryptocurrency prices and user behaviour trends.
3. Out of 39 registered students, 29 submitted their third hands-on assignment on time, which reflects an increase in participation.

I reminded students about their mini-project registration and offered optional feedback sessions for early review.

Attachments

- Slides 4: Session 4 Probability and Variability.pdf
- Bonus Slides 4: Hypothesis Testing in Simple Words.pdf
- Cheat Sheet 4: Probability and Variability Cheat Sheet.pdf
- Course Notebook 4: Week 4 Notebook.ipynb
- Bonus Notebook 4: Hypothesis Testing.ipynb
- Assignment 4: Hands On Python Practice 4.ipynb
- Quiz 4: Kahoot – Probability and Variability

3.5.5 Contact Lesson 5 (2.4.2025)

Lesson Summary: In this session, I guided students through exploring relationships between variables using Python—an essential step in understanding how different data points influence one another. We began with correlation and covariance, discussing how they help us measure interaction between variables. I explained the differences between positive, negative, and no correlation, and showed how to compute Pearson, Spearman, and Kendall's Tau using real-world datasets.

After setting a solid foundation, we moved on to linear regression—starting with simple regression before diving into multivariate models. I walked students through how to evaluate model performance using R-squared, Mean Squared Error (MSE), and residual plots. We also talked about common challenges like under fitting and over fitting, and how to spot them.

Students followed along in a live Jupyter Notebook session, supported by a visual cheat sheet and two short explainer videos I created to help demystify covariance and regression. We wrapped things up with a 20-question Kahoot quiz, which kept the energy high and helped reinforce key takeaways in a fun, interactive way.

To illustrate the variety of student project topics at this stage of the course, I shared the following word cloud built from the submitted project descriptions (see Figure 15):



Figure 15: Word cloud of keywords from student mini-projects

Self-Reflection:

1. Students were highly engaged throughout the session, asking questions during both the correlation section and regression demo.

2. The visual videos explaining abstract concepts helped clarify student doubts and sparked productive discussions.

3. Out of 39 registered students, 28 submitted their Week 4 hands-on exercise before the deadline. Others were reminded to submit via Moodle or email before the late cut-off.

I also gave a mid-course update on our Mini-Project, which serves as the capstone for this course. So far, 24 students have registered their project topics. Four are working independently, while the rest have formed teams of 2–3. I reminded everyone to finalize their group and register their project via the form on Moodle by April 9, 2025, at 12:00

PM. I also emphasized that students shouldn't choose duplicate datasets unless they're working in the same group.

The range of topics submitted has been impressive, spanning:

- 1. Public health:** (e.g., Obesity Prediction, CDC Nutrition & Physical Activity Data)
- 2. Cybersecurity:** (e.g., Threat detection, vulnerability data)
- 3. Energy and environment:** (e.g., Household Load & Solar Generation, Heating Demand)
- 4. Finance and economics:** (e.g., S&P 500 trends, Penn World Table, Economic Performance)
- 5. Behavioural analytics:** (e.g., Spotify usage patterns, Titanic survival, user trend datasets)

I'm happy to see students aligning their project topics with real-world interests and professional goals. The diversity in themes is really encouraging—it reflects the interdisciplinary nature of data analytics.

Attachments

- Slides 5: Session 5 Relationships Between Variables.pdf
- Cheat Sheet 5: Relationships Between Variables Cheat Sheet.pdf
- Course Notebook 5: Week 5 Notebook.ipynb
- Assignment 5: Hands On Python Practice 5.ipynb
- Extra Video 1: Linear Regression in 2 minutes
- Extra Video 2: The Covariance Explained in One Minute: Definition, Formula and Examples
- Mini-Project Instruction: Mini-Project Instruction.pdf
- Mini-Project Dataset List: Mini-Project Dataset List.pdf
- Quiz 5: Kahoot – Relationships Between Variables

3.5.6 Contact Lesson 6 (9.4.2025)

Lesson Summary: This session, I went a little deeper into the world of data visualization and shared principles for effective data visualization in Python, building on the groundwork I laid in our previous lessons. I reviewed the hallmarks of solid charting—emphasizing clarity, proper scaling, and accurate labelling—and compared these with examples that were cluttered or potentially misleading. I showed how students can create clean and professional looking visuals that communicate data effectively using examples from Matplotlib and Seaborn. Along the way, I highlighted some common pitfalls to avoid — such as choosing the wrong type of chart, not taking into account how that data should be ordered, or jamming too many things into a plot so that the important insights get lost.

After I illustrated the notions of “good” and “bad” visualizations, I guided the class through hands-on exercises in a Jupyter Notebook. Together, we explored how to create subplots, scatter plots, histograms, box plots, heatmaps, and residual plots, taking the time to see how each approach can clarify our data storytelling. To keep it interactive, I then led a Kahoot quiz with twenty questions about data visualization best practices, offering students a fun way to test their knowledge and reinforce the day’s main lessons.

Lastly, I posted an updated list of group and individual mini-project participants along with a feedback schedule, giving everyone—whether solo or in a team—an opportunity to refine their dataset choices, preliminary analyses, and overall project direction based on direct, real-time feedback.

Self-Reflection:

1. Students had a good grasp of why clear labelling, consistent scales, and honest visual representation are important, in the context of data visualization. A handful went further to explore how small design decisions can influence how information is interpreted.

2. Discussing the “good” vs. “bad” visualization examples produced great conversation. The hands-on approach really appealed to students and helped them to identify and avoid the common pitfalls in their own work.

3. The feedback trainings went very well. The early drafts of their project notebooks — an individual and two group — were shared with me and I offered suggestions for their code, analysis and visualizations that would assist them in improving these components of their work.

I also mentioned the Mini-Project Registration one last time. This is the capstone project for this course. I gave them the deadline of tonight, April 9, 2025, at 24:00 (midnight), and told any students who hadn’t yet submitted their forms to do so without delay. I passed on the list of registered participants and selected datasets and reminded everyone to double-check their submission details—especially make sure they aren’t submitting duplicate datasets (unless you’re in the same group working on the same one). For everyone struggling with the form: please remember to login with your Metropolia account.

In addition, I offered optional feedback sessions for anyone seeking early feedback on their mini-project notebooks or draft presentation slides. The next available session is on April 16, 2025, from 16:30 to 18:00, and students can book a 5–10 minute slot via the

Feedback Session Sign-Up Form.

By the end of the day, a total of 31 students had registered their mini-projects, including 11 working individually and the rest organized into 8 group projects.

Attachments

- Slides 6: Session 6 Data Visualisation.pdf
- Bonus Slides 6: Good & Bad Visualisation.pdf
- Cheat Sheet 6: Visualisation Cheat Sheet.pdf
- Course Notebook 6: Week 6 Notebook.ipynb
- Bonus Notebook 6: Good & Bad Visualisation.ipynb
- Assignment 6: Hands On Python Practice 6.ipynb
- Extra Video: The Value of Data Visualization The Power of Visual Storytelling
- Quiz 6: Kahoot – Data Visualisation

3.5.7 Contact Lesson 7 (16.4.2025)

Lesson Summary: This session was all about integration and application. I began by recapping the major topics we've explored so far—from descriptive statistics and data cleaning to regression, visualization, and hypothesis testing. Then I shifted the focus entirely to the capstone **Cryptocurrency Mini-Project**, where students applied everything they've learned to a real-world dataset.

I led a walkthrough of the Jupyter Notebook we've been building, focusing on techniques for dataset exploration, summary statistics, and visual pattern discovery. Students followed along as we:

- Loaded and cleaned historical crypto market data (2015–2025)
- Explored prices, volume, and market cap across tokens and years
- Created histograms, box plots, scatter plots, and heatmaps
- Identified outliers using Z-scores and visual diagnostics
- Applied hypothesis testing and ran basic regression models
- Generated predictive insights and selected tokens to recommend

The goal was not only to strengthen their technical skills but also to emphasize storytelling—how to clearly communicate insights using well-designed visuals and logical structure. I reminded them that both the notebook and the accompanying presentation slides should tell a compelling, data-driven story.

To keep the mood light and interactive, I wrapped the session with a 20-question **Kahoot quiz** that reviewed Python functions, chart types, data wrangling methods, and best practices in analysis and visual reporting.

Self-Reflection:

1. Students were visibly more confident connecting multiple techniques—like combining descriptive stats with hypothesis tests or creating regression-informed plots. Their questions were more strategic, focusing on refining their analysis for insight generation.

2. Many of them asked about fine-tuning their plots and regression models. A few even went beyond the required work to explore time-based groupings or token volatility. This kind of curiosity and initiative was exciting to see at this point in the course.

3. I reminded them that their Mini-Project (Notebook + Presentation) is due soon and that optional feedback sessions are available. The next one is on April 16, 2025, from 16:30 to 18:00. Students can reserve a slot using the Feedback Session Sign-Up Form on Moodle.

4. Based on our session poll, most groups are now in the final analysis and visualization stages. I reassured them that clarity is more important than complexity—don't try to over-model; just tell the best story the data supports.

Attachments

- Slides 7: Session 7 – Recap & Crypto Mini-Project Overview
- Jupyter Notebook 7: Week 7 Notebook.ipynb
- Project Presentation Slides: Crypto Market Analysis (2015–2025).html
- Extra Video: Making a better research poster
- Quiz 7: Kahoot – Advanced Topics & Integration

3.5.8 Contact Lesson 8 (23.4.2025)

Lesson Summary: This final session marked the culmination of everything we've explored throughout the course—from Python fundamentals to applied analytics. I had the privilege of watching students present their Mini-Projects, each one a unique showcase of their creativity, technical skill, and ability to tell compelling stories with data.

Before the session, I selected a Teaching Assistant to help manage technical aspects in Zoom—such as participant mics, screen shares, and troubleshooting. We conducted a one-hour test and correspondence session the day before to make sure everything ran smoothly.

The session began with a warm welcome and a quick overview of the presentation format. Students were reminded that each group or individual had 5–6 minutes to present, followed by a 1-minute peer feedback vote using Zoom polls. I emphasized the importance of structure, clarity, and insight, rather than overcomplication.

Throughout the two and a half hour session, students showcased a wide array of real-world data applications, including:

- Cryptocurrency trends and token predictions
- Spotify user behavior and genre analytics
- Global health and nutrition indicators
- Solar energy demand and smart homes
- Economic growth and employment data
- Cybersecurity threat patterns and vulnerability detection

All presenters submitted their final Jupyter Notebooks and slide decks via GitHub prior to class. Their projects included everything from data import and cleaning to hypothesis testing, visualization, and in some cases, predictive modeling using regression or classification techniques.

Among all registered mini-projects, one student could not attend the live session and submitted a 5-minute pre-recorded presentation. I shared it during their assigned slot. Additionally, three other students were unable to attend due to personal conflicts. All others participated actively—either by presenting or through peer feedback.

We took a short break around 17:10 and wrapped up around 18:30 with some final thoughts. I reminded students of the project grading scheme (70% final project, 30% hands-on exercises), the optional +10% bonus for predictive components, and encouraged everyone to vote thoughtfully and fairly in the peer polls.

Self-Reflection:

1. I was genuinely impressed with the quality of work presented. Several projects were not only technically sound but delivered with confidence and clarity. Some even included advanced elements like time-series forecasting and domain-specific analytics.

2. Having a TA for technical support significantly reduced my cognitive load, allowing me to focus on timekeeping, feedback, and facilitating discussion. This support was crucial in keeping the session smooth and professional.

3. Peer voting worked well as a feedback tool and added a democratic, participatory spirit to the session. It also encouraged students to actively listen and learn from each other.

4. I closed the session and course by thanking all participants and inviting them to complete a short feedback survey. I also shared some encouraging words in a follow-up email, reflecting on the journey we've taken together over these eight weeks.

This course was part of my Practical Teacher Training (IPTE) at HAMK, conducted in collaboration with Metropolia University of Applied Sciences. Teaching it has been an immensely rewarding experience, made even more meaningful by the students' engagement and enthusiasm.

Attachments

- Presentation Schedule (Published on Moodle): Presentation Schedule
- Feedback Form: Feedback Form

Chapter 4

Teacher Training Outcomes and Feedback

4.1 Final Mini-Project Outcomes

Summary: The final session of the course featured the live presentations of the Mini-Projects—a cumulative task combining all skills and knowledge acquired during the Python for Data Analytics & Statistics course. Each project reflected real-world datasets and demonstrated students’ abilities to explore, analyse, and visualize data effectively.

Students presented individually or in small teams via Zoom, with each group given 5–6 minutes followed by peer voting using Zoom polls. To ensure a seamless session, I arranged for a Teaching Assistant to support technical tasks during the session. We completed a 1-hour testing session the day before the final class.

The presentations were diverse in both topic and approach. Projects included time series forecasting, regression-based predictions, clustering, and insightful data storytelling. One student submitted a recorded video ahead of time (due to scheduling conflicts), and three students were absent but had submitted their materials.

Visual Summary of Outcomes:

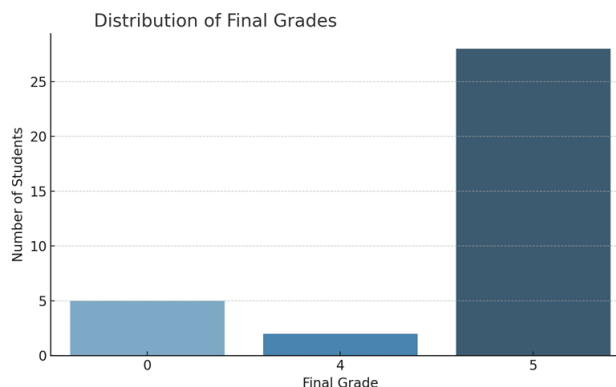


Figure 16: Distribution of Final Grades

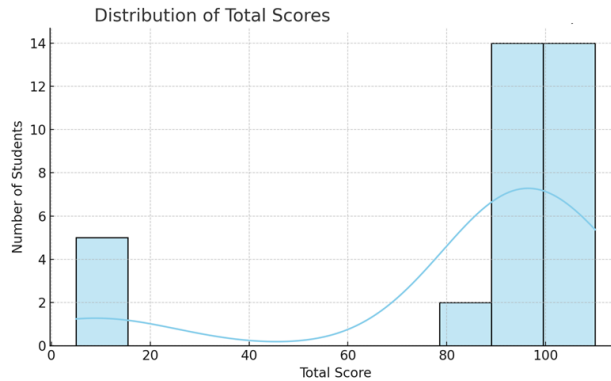


Figure 17: Distribution of Total Mini-Project Scores

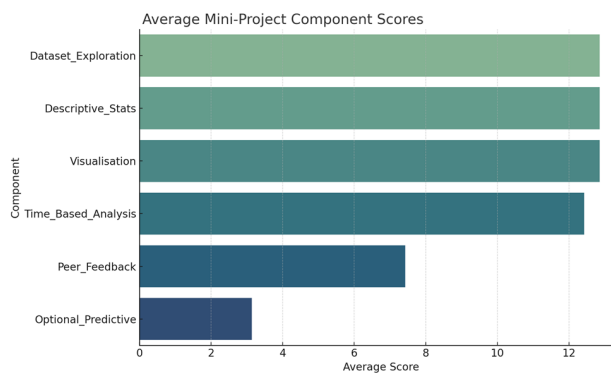


Figure 18: Average Scores per Mini-Project Component

The strongest performance areas were *Visualisation* and *Descriptive Summary*, indicating students' strengths in storytelling and collaborative engagement. Optional predictive analysis was submitted by several students, who received up to +10% bonus credit on their final score.

4.2 Modes of Teaching Outcomes

Looking back on the course delivery and final student performance in *Python for Data Analytics & Statistics*, I can confidently say that the interactive, student-centred approach yielded meaningful learning outcomes. The structure of the course—from foundational Python programming to advanced data visualization and statistical modelling—was designed to support progressive skill-building. The outcomes not only met but, in many cases, exceeded expectations.

One of the most gratifying indicators of success was the high engagement with the hands-on exercises: by the end of the course, the vast majority of students had completed all six weekly tasks, demonstrating consistency and commitment. A closer review of the assignment scores shows that over two-thirds of students earned full points (5/5) for each

weekly hands-on activity, with only minor variation across weeks. This indicates that the learning scaffolding, cheat sheets, and guided notebooks effectively supported learners of varying backgrounds.

Moreover, the final mini-project presentations served as a comprehensive benchmark of learning. Students applied a range of techniques—from exploratory analysis and visualization to hypothesis testing and regression modelling—on diverse, real-world datasets. According to the grading breakdown, the most consistent strengths were in descriptive exploration (DE), statistical summary (DS), and visualization (VIS), with many students achieving full marks in these areas. This aligns closely with the feedback received during and after the course, where students highlighted the data visualization session and mini-project as the most valuable and enjoyable components.

One standout observation from the final grades is that 28 out of 40 students achieved the top final grade of 5, reflecting an exceptional level of performance. Several students also earned bonus points for including predictive modelling elements in their projects, demonstrating initiative beyond the course requirements. Peer voting during the final session confirmed these strengths, with students recognizing clarity, creativity, and real-world relevance in each other's work.

Despite the overwhelmingly positive outcomes, the student feedback also surfaced constructive areas for growth. While many praised the well-structured content and availability of support, some requested deeper case-based analysis and more opportunities for small group interaction. Additionally, a few learners noted that exercises were occasionally too guided, prompting me to plan more open-ended tasks and collaborative breakouts in future iterations.

In terms of inclusivity and teaching adaptability, I observed notable improvement in participation among students who had initially self-identified as beginners. Their ability to complete all required tasks and confidently present their mini-projects suggests that differentiated instruction—such as extra resources, feedback sessions, and optional scaffolding—was effective.

Overall, the course fulfilled its intended learning goals while also fostering a collaborative and exploratory spirit. The combination of structured instruction, real-world datasets, timely feedback, and a culminating project created an environment where students could thrive. I leave this course not only with a sense of accomplishment but with actionable insights to further elevate the learning experience in future runs.

4.3 Student's Feedback

Overview: Course evaluation was excellent among a group of mostly B.S. and M.S. level participants. Students also ranged in their proficiency in Python, with some being completely new to Python and others holding previous advanced knowledge. Overall, over 60 percent of learners ranked the content of the course as “excellent”, and the learning objectives were universally felt to be well defined as well as met. Course Format and Organisation Course Format and Organisation received a lot of good feedback for being clear, whereas Data Visualisation and the Mini-Project were highlighted as areas of excellence. Several students expressed their high appreciation for the relevance of the course to their career aspirations, and many appreciated the active assignments as good for the creation of new neural pathways. The last presentation was also recognised as having an important role in the process of synthesising knowledge.

Instructor quality and availability were rated highly, with the majority of students reporting timely and supportive feedback. Most often used and valued were support channels, including emails and feedback sessions. Most of the students perceived the course to be at the right challenge level and effective; indeed, the online learning environment suited most of the students. More than 75 percent of the individuals indicated they would have recommended the course to someone else, and the majority of the participants were overall satisfied with it. Recommendations for enhancing the curriculum included increased use of case-based examples and more small group sessions to allow for more peer interaction and practical application.

1. What is your highest educational level?

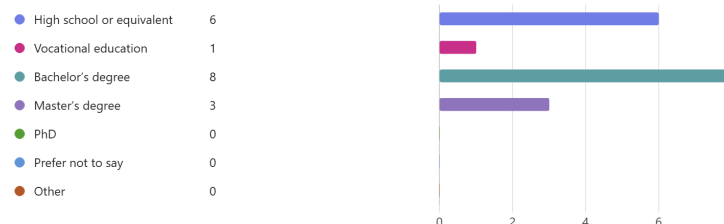


Figure 19: Participants' Highest Educational Level

2. How familiar were you with Python programming before this course?



Figure 20: Familiarity with Python Before the Course

3. How would you rate the overall content of the course?

Excellent	11
Good	6
Fair	1
Poor	0
Very poor	0

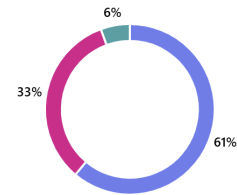


Figure 21: Overall Course Content Rating

4. Were the course learning objectives (such as Python skills and statistical analysis) clearly stated and met?

Fully stated and fully met	10
Clearly stated, mostly met	7
Somewhat clear and partially met	1
Unclear or unmet	0
Not sure	0

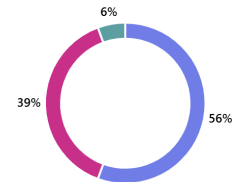


Figure 22: Achievement of Learning Objectives

5. Was the course content presented in a clear and organized manner?

Very clearly and well organized	14
Mostly clear and organized	3
Somewhat clear, could be improved	1
Unclear or disorganized	0

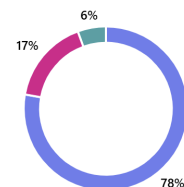


Figure 23: Organization and Clarity of Course Content

6. How relevant were the course topics to your learning or career goals?

Extremely relevant	7
Very relevant	7
Moderately relevant	2
Slightly relevant	2
Not relevant	0

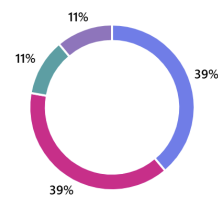


Figure 24: Relevance of Topics to Career Goals

7. Which topics did you find most valuable? (Select up to 3)

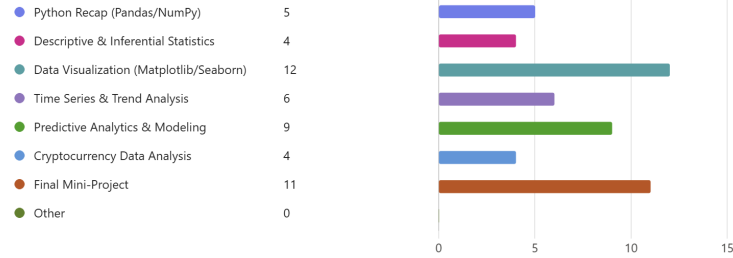


Figure 25: Most Valuable Topics Selected by Students

8. How well did the instructor explain and apply the course topics in practice?



Figure 26: Instructor's Practical Explanation and Application of Topics

9. How would you rate the instructor's accessibility and responsiveness?



Figure 27: Instructor's Accessibility and Responsiveness

10. How would you rate the feedback provided by the instructor?



Figure 28: Timeliness and Helpfulness of Feedback Provided

11. Which support channels did you use? (Select all that apply)

Course Discussion Page	4
Slack Channel	1
Email Support	13
Feedback Sessions	7
Other	3

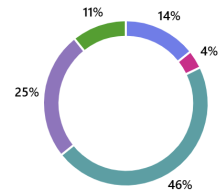


Figure 29: Support Channels Utilized by Students

12. How would you rate the course structure (e.g., pacing, workload, assignments)?

Excellent	11
Good	5
Fair	2
Poor	0
Very poor	0

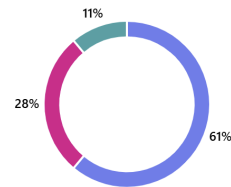


Figure 30: Student Rating of Course Structure

13. How relevant and helpful were the hands-on exercises and assignments?

Very relevant and helpful	14
Mostly relevant	4
Somewhat relevant	0
Not very relevant	0
Not at all relevant	0

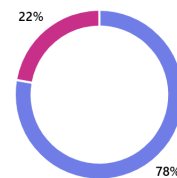


Figure 31: Perceived Relevance of Hands-On Exercises

14. How challenging was the course overall?

Too easy	2
Just right	8
Challenging but manageable	8
Very difficult	0

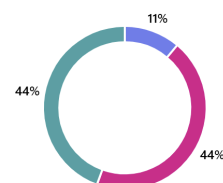


Figure 32: Challenge Level of the Course

15. How effective was the online learning format for your learning style?

Very effective	8
Effective	5
Neutral	3
Ineffective	1
Very ineffective	1

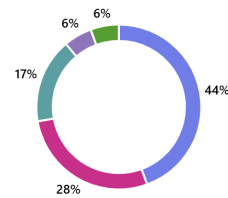


Figure 33: Effectiveness of Online Learning Format

16. Did you experience any technical difficulties with the tools/platforms used (e.g., Zoom, GitHub)?

No issues at all	15
Minor issues, but manageable	0
Occasional issues that disrupted learning	3
Frequent issues	0
I could not participate fully due to technical problems	0

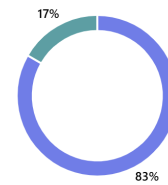


Figure 34: Technical Difficulties Experienced

17. How engaged did you feel during live online sessions?

Very engaged	7
Engaged	6
Neutral	4
Disengaged	0
Very disengaged	1

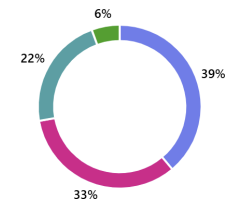


Figure 35: Student Engagement During Live Sessions

18. How would you rate the accessibility and usefulness of the online learning resources (e.g., Jupyter Notebooks, datasets)?

Very accessible and extremely useful	12
Accessible and useful	6
Somewhat accessible or useful	0
Difficult to access or not useful	0

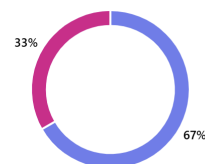


Figure 36: Accessibility and Usefulness of Online Resources

19. How clear were the mini-project instructions?

Very clear	12
Clear	5
Unclear	1
Very unclear	0

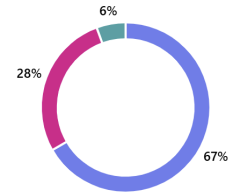


Figure 37: Clarity of Mini-Project Instructions

20. To what extent did the final presentation session help reinforce your learning?

Greatly reinforced my learning	10
Somewhat reinforced my learning	6
Neutral	1
Slightly helpful	1
Not helpful at all	0

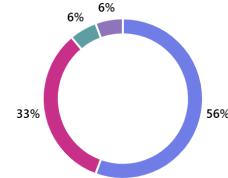


Figure 38: Final Presentations and Reinforcement of Learning

21. How likely are you to recommend this course to others?

Very likely	14
Likely	2
Neutral	0
Unlikely	1
Very unlikely	1

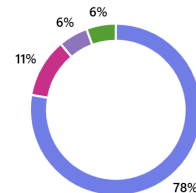


Figure 39: Likelihood of Recommending the Course

22. What is your overall satisfaction with the course?

Very satisfied	13
Satisfied	3
Neutral	1
Dissatisfied	1
Very dissatisfied	0

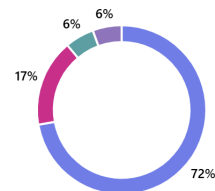


Figure 40: Overall Satisfaction with the Course

23. What improvements would you suggest for future iterations? (Select all that apply)

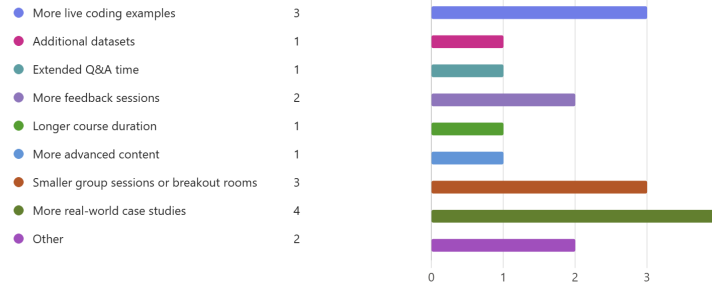


Figure 41: Suggestions for Course Improvements



Figure 42: Word Cloud of Free-Text Feedback

Open Comments:

- “Excellent course overall, one of the best courses at Metropolia. Inspirational and well-organized.”
- “The teacher was very clear and helpful. The professionalism and passion showed throughout the course.”
- “Kahoot quizzes worked well to keep the interest up. Some exercises were too easy due to complete examples in notebooks.”
- “Would be great to improve the audio setup (e.g., use a mic) and include more small interactive questions during class.”
- “Overall, a great course that I will be recommending to others.”

4.4 My Counter Feedback

Reflection on Student Responses:

Based on the survey results, I am pleased that over 60% of students rated the course content as “Excellent,” and that the vast majority found the learning objectives clear and the visualizations and mini-project highly valuable. Several areas of consistent praise emerged:

- *Clarity and Organization:* 78%+ found the course well structured and clearly presented.
- *Instructor Engagement:* Over 80% rated my accessibility and responsiveness as “Extremely” or “Very” positive.
- *Practical Exercises:* Hands-on assignments and the final mini-project were highlighted as the most relevant components.

Addressing Constructive Criticism:

A few recurring suggestions and concerns stood out:

- *Audio Quality:* Some students found the audio difficult to hear. I will invest in a higher-quality headset and test sound levels before each session.
- *Exercise Difficulty:* A handful noted that certain homework tasks felt too guided. I will introduce one or two “open-ended” challenges each week, requiring students to design parts of the solution themselves.
- *Interactive Checkpoints:* Students requested more bite-sized polls or discussions mid-lecture. Going forward, I will embed 2–3 live polls per session via Zoom reactions or integrated quizzes.
- *Case Studies & Breakouts:* There was interest in smaller group work. I plan to pilot 10-minute breakout rooms around Session 6–7 for peer collaboration on mini-problems.

Action Plan for Future Iterations:

1. **Upgrade AV Setup:** Acquire and test a dedicated USB microphone before the next course run.
2. **Revise Assignments:** Add one “design-your-own” coding challenge per week, with minimal scaffolding.
3. **Embed Live Polls:** Use Zoom polls or Mentimeter at key concept checkpoints in every session.
4. **Introduce Breakout Tasks:** After major theory segments, send students to 3–4 person Zoom rooms for short data puzzles.

5. **Expand Case Studies:** Curate 2–3 new real-world datasets (e.g. healthcare, climate) for optional mini-projects.

Looking Ahead:

I'm excited to incorporate this feedback into the next offering of "Python for Data Analytics & Statistics." My goal is to maintain the high standards you already value—clarity, interactivity, and practical relevance—while enhancing the student experience through improved audiovisuals, richer challenges, and more peer collaboration.

Hamed Ahmadinia

Turku, 25.4.2025

Appendix

Appendix I: Student Background Survey

Python for Data Analytics & Statistics – Student Background Survey

This short survey is just to get a sense of your programming background and experience.

Section 1: General Information

1. Education Level

- ☐ High school
- ☐ Bachelor's
- ☐ Master's
- ☐ PhD
- ☐ Other: _____

2. Field of Study or Work

- ☐ Business ☐ Social Sciences ☐ CS/IT
- ☐ Engineering ☐ Healthcare ☐ Humanities ☐ Other

Section 2: Programming Experience

3. Do you have programming experience?

- ☐ Yes, in Python ☐ Yes, in another language ☐ No experience

4. Which Python concepts do you know?

- ☐ Basic syntax ☐ If-else ☐ Loops ☐ Data structures
- ☐ Functions ☐ File handling ☐ Pandas ☐ I am new

5. Experience with Jupyter Notebook?

- ☐ Comfortable ☐ Tried it ☐ Never used

Section 3: Data & Analytics Background

6. Experience with data analysis?

- ☐ Experienced ☐ Some knowledge ☐ New to it

7. Data-related tasks done before?

- ☐ Load data ☐ Clean/preprocess ☐ Stats
- ☐ Visuals ☐ None

Section 4: Learning Needs

8. Need refresher materials?

☐ Yes ☐ No

9. Specific help needed?

Appendix II: Course Evaluation Survey

Thank you for taking part in the course. Your feedback helps improve content and teaching quality.

Section 1: Participant Background

1. Highest education?
☐ High school ☐ Bachelor's ☐ Master's ☐ PhD ☐ Other
2. Python knowledge before course?
☐ None ☐ Basic ☐ Intermediate ☐ Advanced

Section 2: Course Content & Objectives

3. Overall content?
☐ Excellent ☐ Good ☐ Fair ☐ Poor
4. Objectives met?
☐ Fully ☐ Mostly ☐ Partially ☐ Not sure
5. Presentation clarity?
☐ Very clear ☐ Clear ☐ Unclear
6. Relevance?
☐ Extremely ☐ Very ☐ Moderate ☐ Slight ☐ Not relevant
7. Most valuable topics?
☐ Python ☐ Statistics ☐ Viz ☐ Time Series
☐ Predictive ☐ Crypto ☐ Mini-Project ☐ Other: _____

Section 3: Instructor & Support

8. Instructor effectiveness?
☐ Excellent ☐ Good ☐ Fair ☐ Poor
9. Accessibility?
☐ Very accessible ☐ Sometimes ☐ Rare
10. Feedback helpful?
☐ Very helpful ☐ Somewhat ☐ Not helpful ☐ No feedback
11. Support channels used?
☐ Discussion ☐ Slack ☐ Email ☐ Feedback ☐ None

Section 4: Assignments and Structure

12. Course structure?
☐ Excellent ☐ Good ☐ Fair
13. Assignment relevance?
☐ Very relevant ☐ Somewhat ☐ Not relevant
14. Difficulty level?
☐ Too easy ☐ Just right ☐ Challenging

Section 5: Online Learning Experience

15. Online format effectiveness?
☐ Very effective ☐ Somewhat ☐ Not effective
16. Technical issues?
☐ None ☐ Minor ☐ Major
17. Engagement level?
☐ Very engaged ☐ Neutral ☐ Disengaged
18. Learning resources?
☐ Excellent ☐ Good ☐ Poor

Section 6: Mini-Project Feedback

19. Instructions clarity?
☐ Clear ☐ Neutral ☐ Unclear
20. Final presentation helpfulness?
☐ Reinforced learning ☐ Neutral ☐ Not helpful

Section 7: Overall Experience

21. Recommend course?
☐ Yes ☐ Maybe ☐ No
22. Overall satisfaction?
☐ Very satisfied ☐ Satisfied ☐ Neutral ☐ Dissatisfied

Section 8: Open Feedback

23. Suggested improvements?
☐ More live coding ☐ Extra datasets ☐ Breakout rooms
☐ Longer course ☐ Case studies ☐ Other: _____
24. Other comments?

Appendix III: Final Mini-Project Presentation Schedule

Thank you for your active participation in the course. Below is the full presentation and feedback schedule for the final mini-projects. Note: names are anonymized for publishing purposes.

Session Overview

- **5–6 minutes** per presentation + **1-minute peer voting (Zoom poll)**.
- **10% of the grade** comes from peer feedback.
- **Full attendance required** (16:00–18:40).
- **Break** scheduled from 17:10–17:20.

Presentation Schedule (Anonymized)

Each row includes **presentation time, anonymized topic title, and 1-minute peer voting**.

Geography, Energy and Global Trends

Presentation Slot	Mini Project Topic	Voting Slot
16:00–16:06	GeoNames Worldwide	16:06–16:07
16:07–16:13	Penn World Table	16:13–16:14
16:14–16:20	Renewable Energy + Battery Storage	16:20–16:21
16:21–16:27	Household Load & Solar Generation	16:27–16:28
16:28–16:34	European Heating Demand	16:34–16:35

Social, Health and Human-Centric Data

Presentation Slot	Mini Project Topic	Voting Slot
16:35–16:41	UCI Adult (Income Prediction)	16:41–16:42
16:42–16:48	Titanic Survival Data	16:48–16:49
16:49–16:55	Obesity Prediction	16:55–16:56
16:56–17:02	Subjective General Health (Europe)	17:02–17:03
17:03–17:09	Covid-19 Variants Survival	17:09–17:10

Break: 17:10–17:20

Cybersecurity and Software Risk

Presentation Slot	Mini Project Topic	Voting Slot
17:20–17:26	Software Vulnerabilities (CVE)	17:26–17:27
17:27–17:33	Cyber Defense Traffic/Logs	17:33–17:34
17:34–17:40	Global Cybersecurity Threats	17:40–17:41

Markets, Finance and Trends

Presentation Slot	Mini Project Topic	Voting Slot
17:41–17:47	S&P 500 Analysis	17:47–17:48
17:48–17:54	Wolt's Dataset	17:54–17:55

Entertainment, Gaming and Media

Presentation Slot	Mini Project Topic	Voting Slot
17:55–18:01	Steam Game Data	18:01–18:02
18:02–18:08	FIFA-2021 Player Stats	18:08–18:09
18:09–18:15	Netflix Production Trends	18:15–18:16
18:16–18:22	Spotify User Historical Data	18:22–18:23

Wrap-Up and Final Discussion

Time Slot	Activity
18:23–18:40	Final remarks, Q&A, discussion, session closing

Final Reminders:

- Be ready at 15:55 — we start exactly at 16:00.
- Make sure your slides or notebooks are prepared for screen sharing.
- If you have work or commitments, please arrange time off now to stay until 18:30.
- Zoom Polls will follow each presentation, so your feedback is essential.
- If you are unable to attend the live presentation session, you have the alternative of recording your 5-minute mini-project presentation and coordinating with a fellow course participant to play your recorded video during your allocated slot. Please note, it is your responsibility to ensure that this arrangement is coordinated.

Appendix IV: Peer Feedback via Zoom Polls

Dear Students,

During our final mini-project presentation session, we will use Zoom Polls to collect anonymous peer feedback after each presentation.

How it works:

After each presentation, a quick 3-question Zoom Poll will pop up on your screen. You will rate the presentation based on:

1. **Clarity** – Was it well-structured and easy to follow?
2. **Creativity** – Was the idea engaging and well presented?
3. **Data Usefulness** – Was the dataset relevant and applied meaningfully?

For each question, choose 1–5 stars (*1 = Poor, 5 = Excellent*).

Presentation x : x

3 questions

1. How clear and well-structured was this presentation? (Single choice) *

- ☐ ★
- ☐ ★ ★
- ☐ ★ ★ ★
- ☐ ★ ★ ★ ★
- ☐ ★ ★ ★ ★ ★

2. How original and engaging was the project idea or presentation? (Single choice) *

- ☐ ★
- ☐ ★ ★
- ☐ ★ ★ ★
- ☐ ★ ★ ★ ★
- ☐ ★ ★ ★ ★ ★

3. How relevant and well-applied was the dataset to the problem? (Single choice) *

- ☐ ★
- ☐ ★ ★
- ☐ ★ ★ ★
- ☐ ★ ★ ★ ★
- ☐ ★ ★ ★ ★ ★

Figure 43: Example of the Zoom Poll students will see during peer evaluation.

Your ratings will contribute 10% to your peer's mini-project grade, so please provide thoughtful and respectful feedback.

Tips for Giving Feedback:

- Be honest but supportive.
- Focus on content and delivery, not language proficiency.
- Stay present and participate in every poll – your feedback matters!

Let's work together to create a positive, fair, and constructive final session for everyone.

If you have any technical issues with Zoom Polls during the session, please send a direct message in Zoom chat.



LOREM
IPSUM DOLOR
SIT AMER

CODEDOLF

LOREM
IPSUM
DOLOR
SIT AMET



DOI: 10.5281/zenodo.15254795