Programming with Data Syllabus

Module description

This module will show you how to work with data: getting data from a variety of sources, visualising data in compelling, informative ways, processing data to make it useful and shareable, and reasoning with data to test assumptions and explore the complexity of rich datasets. The module will also introduce you to a new language and programming environment that is well-adapted for these applications – Python.

This module explores a multitude of ways to work with data of different types. Students will take an activity-centric approach to learning, exploring some of the complexities around data through a variety of lenses and a range of different challenges. Each topic covers a different set of skills that are useful in handling, managing and working with data from a variety of sources. Through the identification and exploration of data that is constructed, stored and retrieved in different ways students will gain practical experience in handling a variety of technical challenges.

1. Setting up your programming environment
2. Variables, control flow and functions
3. Data structures
4. Reading and writing data on the filesystem
5. Retrieving data from the web
6. Retrieving data from databases using query languages
7. Cleaning and restructuring data, part one
8. Cleaning and restructuring data, part two
9. Data plotting
10. Version control systems

Module goals and objectives

The module goals are to introduce you to a variety of topics around data programming. Each topic explores both the theory and practice behind using certain techniques, tools and technologies for a given application. As part of the work you will produce a coursework assignment which focuses on a few of these core topics and enables you to explore a research agenda of your choosing.

Upon successful completion of this module, you will be able to:

1. Set up and work within a Python environment for writing data-driven software
2. Write simple programs in Python that perform data analysis and visualisation
3. Explain how to retrieve data from various sources and be able to load it into a program
4. Prepare and summarize data
5. Reason about data to make predictions and classifications

Textbook and Readings

Specific essential readings for this module will be taken from the following text book and online resource:

Wes McKinney, Python for Data Analysis, 2nd Edition, O'Reilly Media, Inc, ISBN: 9781491957660.

https://docs.python.org/3/.

The specific pages for the reading activities will be given in the platform, and there is no need to read beyond to recommended pages.

In addition to the text book, there are additional activities written by the course author, some of which involve coding exercises.

There will also be discussion prompts asking you to do some independent research using online sources.

Module outline

The module consists of 10 topics, each of which spans two weeks.

Topic 1. Setting up your programming environment	**Key concepts:** • development environments • accessing and using the University-provided development environment • installing and using local development environments **Learning outcomes:** • 1.1 use a pre-configured Jupyter notebook system to create, edit and run Python code • 1.2 install and use a Jupyter notebook system on a computer running Windows, MacOS or GNU/Linux • 1.3 write and explain simple Python programmes using variables and mathematical operators.
Topic 2. Variables, control flow and functions	**Key concepts:** • using logic to control the branching of a program execution • using iteration to work with arrays of data • importing modules and using the built-in functions **Learning outcomes:** • identify and use correct syntax and explain the purpose of built-in variable types int, float and list • use logic and iteration to fill arrays with data, sum array elements and locate array elements with certain characteristics • import Python numpy and scipy modules and use them to compute basic statistics
Topic 3. Data structures	**Key concepts:** • how data can be represented in different

	structures • how lists and dicts are built in data structures and numpy arrays are provided by the numpy library • the possibility to convert between data types and how this is the start of the data processing pipeline **Learning outcomes:** • explain the difference between lists, dicts and numpy arrays and select appropriate data structures for particular examples of data • write Python programs that can process and analyse text data in lists, dicts and numpy arrays • implement linguistic analysis algorithms that can handle natural language processing tasks
Topic 4. Reading and writing data on the filesystem	**Key concepts:** • reading, processing and writing flat data in CSV format • reading, processing and writing structured data in JSON format • using data cleaning techniques to identify and remove unwanted data. **Learning outcomes:** • describe different types of data files and evaluate their appropriateness for storing different types of data • write Python programs that can read and write files in CSV and JSON formats • handling and preparing data for further processing
Topic 5. Retrieving data from the web	**Key concepts:**

	• exploring how data on the internet is provided by HTTP servers and can be accessed using HTTP clients • identifying how data retrieved from the internet commonly needs to be processed prior to analysis to extract the relevant content • handling data using requests **Learning outcomes:** • explain what HTTP is and how the client server model makes it possible to access data on the internet • implement an HTTP client in Python and use it to retrieve data from an HTTP server in HTML and JSON format • handle data using requests or APIs
Topic 6. Retrieving data from databases using query languages	**Key concepts:** • relational databases and SQL • how to connect to and issue queries to a database • how to work with database tables in Python **Learning outcomes:** • describe the structural elements of a relational database such as tables, columns and relations • write simple SQL queries to read and write data from a relational database into Python using an SQL library • select and use appropriate data structures to store data obtained from relational databases.
Topic 7. Cleaning and restructuring data. Part 1	**Key concepts:** • problems with real world data • techniques for cleaning dirty data • techniques for writing more robust data

	processing code. **Learning outcomes:** • explain the problems that can occur in particular data processing scenarios if data has not been properly cleaned • apply data cleaning techniques such as interpolation to cope with missing and corrupted data • use exception handling and data verification techniques to write more robust data processing code.
Topic 8. Cleaning and restructuring data. Part 2	**Key concepts:** • data processing pipelines • decomposition of complex processes into simple steps • unit testing applied to data processing pipelines. **Learning outcomes:** • explain what data restructuring is and define data processing pipelines to get from one form to another • implement data processing pipelines that have been broken into simple steps • explain what unit tests are and write unit tests that can test the steps in a data processing pipeline.
Topic 9. Data plotting	**Key concepts:** • data visualisation principles and different types of plots • writing data visualisation code • adding more elements such as labels and error bars to graphs.

	Learning outcomes: - select and critically evaluate data visualisations for different types of data, justifying their choice in terms of data communication principles - create data visualisations such as graphs, histograms - add error bars, labels and other elements to data visualisations and justify their use.
Topic 10. Version control systems	**Key concepts:** - existing open source projects use version control - version control for backup and snapshots - branching and merging. **Learning outcomes:** - access pre-existing git repositories using command line tools - create a new git repository and add files to it - explain the purpose of version control, including why and how version control is used

Activities of this module

The module is comprised of the following elements:

- Lecture videos. In each topic, you will find a sequence of videos in which the example programs for the course are coded up. Further videos review the key programming techniques seen in the coding videos.
- Readings. Each topic may include several suggested readings. These are a core part of your learning, and, together with the videos, will cover all of the concepts you need for this module.

- Practice Quizzes. Each topic will include practice quizzes, intended for you to assess your understanding of the topics. You will be allowed unlimited attempts at each practice quiz. There is no time limit on how long you take to complete each attempt at the quiz. These quizzes do not contribute toward your final score in the class.
- Programming Activities. Each topic includes programming activity work to enable you to practice implementing your skills. These build on the concepts from the lectures and often provide code excerpts. They also contain challenging activities where you have the opportunity to explore more complex constructs.
- Discussion Prompts. Each topic includes discussion prompts. You will see the discussion prompt alongside other items in the lesson. Each prompt provides a space for you to respond. After responding, you can see and comment on your peers' responses. All prompts and responses are also accessible from the general discussion forum and the module discussion forum. This is a good opportunity to learn from your peers, explore different types of challenges and identify where techniques might vary for given problem spaces or datasets.
- Assessed coursework. There is one assessed coursework, in the middle of the module. You will have the opportunity to explore an area of your choosing and write a report about your approach, communicating both scientific and technical skills.
- Examination. There will be an examination at the end of the session to assess your understanding of the core concepts explored in the course.

How to pass this module

The module has two major assessments each worth 50% of your grade:

- Midterm coursework. This consists of a research activity where you will acquire, explore and analyse a dataset of your choosing.
- End of term examination. This consists of a series of multiple choice and open-ended questions where you will be asked about different concepts pertaining to the ten topics covered.

Activity	Required?	Deadline week	Estimated time per course	% of final grade
Midterm	Yes	1-12	25 hours	50%

coursework				
End of term examination	Yes	22	25 hours	50%