POP77001 — Computer Programming for Social Scientists

Module Information	
Module Code	POP77001
Module Title	Computer Programming for Social Scientists
Department	Political Science
Year Group	Postgraduate (Taught Masters)
Module Personnel	Dr Tom Paskhalis
Academic Year	2026–27
Credit Weighting	10 credits (ECTS)
Semester	Semester 1 (Michaelmas Term)
Contact Hours	One 2-hour lecture and one 2-hour tutorial per week, over 11 teaching weeks.

Module Content	
Module Description	This module provides a foundational introduction to computer programming and software engineering practices for social scientists. It introduces R and Python, two of the principal programming languages and workflows for data analysis, covering basic and intermediate concepts such as objects, types, functions, control flow and debugging in both procedural and object-oriented paradigms. Particular emphasis is placed on data handling and analytical tasks focused on problems in the social sciences. Through hands-on coding exercises, a mid-term exam and a final project, students learn to write, execute and debug scripts, perform data-wrangling tasks, and reason about the complexity and performance of their programs. Each week combines a two-hour lecture with a two-hour hands-on tutorial, with all work carried out in Jupyter Notebooks.
Module Learning Aims	To equip students with practical programming skills and an understanding of computational thinking relevant to social science research: to introduce the R and Python languages and their data-analysis workflows, and to develop students'

Module Content	
	ability to write, execute and debug scripts and to apply these skills to a research project.
Learning Outcomes	On successful completion of this module, students should be able to: ● describe fundamental computer programming concepts; ● demonstrate command of the R and Python programming languages; ● write, execute and debug scripts for data analysis; ● perform data-wrangling tasks using R and Python; ● analyse the complexity and assess the performance of computer programs.
Indicative Reading / Resources	Core texts (R and Python for data analysis): ● Guttag, J. V. (2021) Introduction to Computation and Programming Using Python (3rd edition). MIT Press. ● McKinney, W. (2022) Python for Data Analysis (3rd edition). O'Reilly. ● Matloff, N. (2011) The Art of R Programming. No Starch Press. ● Peng, R. D. (2016) R Programming for Data Science. Leanpub. ● Wickham, H., Çetinkaya-Rundel, M. & Grolemund, G. (2023) R for Data Science (2nd edition). O'Reilly. ● Wickham, H. (2019) Advanced R (2nd edition). Chapman & Hall/CRC. On good coding practice and how computers work: ● Thomas, D. & Hunt, A. (2019) The Pragmatic Programmer (20th Anniversary edition). Addison-Wesley. ● Kernighan, B. W. (2021) Understanding the Digital World (2nd edition). Princeton University Press. ● Petzold, C. (2022) Code: The Hidden Language of Computer Hardware and Software (2nd edition). Microsoft Press. For statistical analysis in R and Python: Agresti, A. & Kateri, M. (2021) Foundations of Statistics for Data Scientists: With R and Python. Chapman & Hall/CRC. A full reading list and online resources are provided on the module website.

Assessment	
Assessment Components	• Participation – tutorial attendance – 10% (coursework) • Assignment 2 – 5% (coursework) • Assignment 3 – 5% (coursework) • Mid-term exam – 20% • Final project – 60% (coursework)
Assessment Details	Assessment combines continuous assessment with a final project. Participation (10%) is based on tutorial attendance. Two graded programming assignments (5% each) are submitted as executable Jupyter Notebooks. A mid-term exam (20%) tests core programming concepts, and the final project (60%) requires students to apply their programming skills to a research problem of their own. All assignments are submitted via Blackboard as valid Jupyter Notebooks, with every code cell executed without error before submission. Reassessment is by examination (examined locally, 100%).

Additional Information	
Prerequisites	None. This is an introductory module and assumes no prior programming experience.
Co-requisites	None
Any other information	Software: the module uses R, Python and Jupyter Notebooks (JupyterLab Desktop is recommended), all of which are free and open-source; students should install them before the first tutorial. All assignments must be submitted as executable Jupyter Notebooks. Generative AI: limited fair use is permitted (for example, to explain concepts or code, to refactor code, and in the final project); using generative AI to generate solutions to assignments is strictly prohibited, and no private or personal data should be submitted to non-local AI services. Module website: https://tom.paskhal.is/POP77001