Computer Programming for Social Scientists

Tom Paskhalis

tom.paskhal.is

- **Module Code:** POP77001
- **Module Website:** tom.paskhal.is/POP77001
- **ECTS Weighting:** 10
- **Semester/Term Taught:** Semester 1 (Michaelmas Term)
- **Contact Hours:**
  One 2-hour lecture
  One 2-hour tutorial
  per week (11 weeks)
- **Module Coordinator:** Dr Tom Paskhalis (tom.paskhalis@tcd.ie)
- **Teaching Fellows:**
  - Sara Cid (cidsb@tcd.ie)

Learning Aims

This module provides foundational knowledge of computer programming concepts and software engineering practices. It introduces students to major programming languages and workflows for data analysis, with a focus on social science questions and statistical techniques.

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;
- exhibit the ability to 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;

Module Content

Students will become familiar with R and Python, two principal programming languages used in data science and research. This course covers basic and intermediate programming concepts, such as objects, types, functions, control flow, debugging in both procedural and object-oriented paradigms. Particular emphasis will be made on data handling and analytical tasks with a focus on problems in social sciences. Homeworks will include hands-on coding exercises. In addition, students will apply their programming knowledge on a research project at the end of the module.

Software

In this module we will study the fundamentals of computer programming using R and Python. Both are free, open-source and interactive programming languages widely used for data analysis. R and Python are widely available for all major operating systems (Windows, Mac OS, Linux).

While there are a range of integrated development environments (IDEs) available for both R and Python (and which are very worth exploring further, more details below), we will use Jupyter Notebooks as the primary way of writing and executing code, and assignment submission.

To work with Jupyter Notebooks, you will need to install Jupyter Notebook on your local machine. I recommend installing JupyterLab Desktop, the cross-platform desktop application for working with Jupyter Notebooks. To use Jupyter Desktop with R, you will also need to install the IRkernel package. Check the instructions for further details on installation and setup.

Alternatively, you may want to try Kaggle Code, an online platform for working with, sharing and exploring data-science-focussed Jupyter Notebooks. Using Kaggle Code requires registration (you can also use your Google account if you have one). While this platform will provide sufficient functionality (and package availability) for completing all assignments for this module, I strongly advise to have a local installation of R, Python and Jupyter Notebook on your machine that you can use moving forward.

In addition to having a local installation of R, Python and JupyterLab Desktop, I advise to install a feature-rich text editor that will allow you to open and inspect (with syntax highlighting) a wide range of scripts and configuration files. Here are a couple of options to try:

- Visual Studio Code
- Sublime Text

Some IDEs for working in R and Python that you might like to try as well:

- RStudio - very popular IDE for R;
- Spyder - similar in appearance IDE for Python;
- PyCharm - development-focussed non-free IDE for Python.

Note that irrespective of your preferred IDE and tool chain all assignments have to be submitted as valid Jupyter Notebooks with all code cells executed prior to submission.

Recommended Reading List

In this module we will rely on a number of books that introduce R and Python with a particular focus on data analysis applications. All of the required readings are available either freely online or through the College Library. While it is not necessary, I strongly advise selecting one or two books (depending on their delivery style and your personal preferences) to purchase as reference texts.

- John Guttag. 2021. *Introduction to Computation and Programming Using Python: With Application to Computational Modeling and Understanding Data.* 3rd ed. Cambridge, MA: The MIT Press
- Norman Matloff. 2011. *The Art of R Programming: A Tour of Statistical Software Design.* San Francisco, CA: No Starch Press
- Wes McKinney. 2022. *Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter.* 3rd ed. Sebastopol, CA: O'Reilly Media. https://wesmckinney.com/book/
- Roger D. Peng. 2016. *R Programming for Data Science.* Leanpub. https://leanpub.com/rprogramming

- Hadley Wickham, Mine Çetinkaya-Rundel, and Garrett Grolemund. 2023. *R for Data Science.* 2nd ed. Sebastopol, CA: O'Reilly Media. https://r4ds.hadley.nz/
- Hadley Wickham. 2019. *Advanced R.* 2nd ed. Boca Raton, FL: Chapman and Hall/CRC. https://adv-r.hadley.nz/

In order to become a better programmer it is important to both develop a solid understanding of a specific language as well as learn about good coding practices more broadly. This book offers a very good and accessible coverage of these approaches:

- David Thomas and Andrew Hunt. 2019. *The Pragmatic Programmer: Your Journey to Master.* 2nd. Boston, MA: Addison-Wesley Professional. https://pragprog.com/titles/tpp20

While not focussed on computer programming *per se*, the following books provide a good background reading on general historical and technical (but accessible) details about binary systems and code, and how computers and related systems (networks, operating systems, etc.) work more broadly:

- Matthew Justice. 2020. *How Computers Really Work: A Hands-On Guide to the Inner Workings of the Machine.* No Starch Press
- Brian W. Kernighan. 2021. *Understanding the Digital World: What You Need to Know about Computers, the Internet, Privacy, and Security.* Princenton, NJ: Princenton University Press
- Charles Petzold. 2022. *Code: The Hidden Language of Computer Hardware and Software.* 2nd ed. Redmond, WA: Microsoft Press. https://www.codehiddenlanguage.com/

If you are looking for a book that provides examples of applying statistical analysis techniques using both R and Python see:

- Alan Agresti and Maria Kateri. 2021. *Foundations of Statistics for Data Scientists: With R and Python.* Boca Raton, FL: Chapman and Hall/CRC

Additional online resources:

- Git Book
- R Inferno
- An Introduction to R and Python For Data Analysis: A Side By Side Approach
- The Hitchhiker's Guide to Python
- Python For You and Me
- Python Wikibook

- Official documentation:
  - R Language Definition
  - Python Language Reference

Assessment Details

The final grade consists of the following parts (with corresponding weighting):

- Participation (tutorial attendance, 10% total)
- Programming exercises (30% total)
- Final project (60%)

All assignments should be submitted via Blackboard. Go to the "Assessment" section — you should be able to see all the assignments listed there. You will need to upload your assignments as Jupyter Notebook. Make sure to check that all cells in your notebook execute correctly and without error prior to submission.

Please make sure that you understand the submission procedure. Unexcused late submissions will be penalized in accordance with standard department policy. Five points per day will be subtracted until the Monday a week and a half after the deadline at which point the assignment is deemed to have failed.