Computer Programming for Social Scientists

Trinity College Dublin 2022/23

Tom Paskhalis

tom.paskhal.is

- **Module Code:** POP77001
- **Module Website:** tinyurl.com/POP77001
- **ECTS Weighting:** 10
- **Semester/Term Taught:** Semester 1 (Michaelmas Term)
- **Contact Hours:**
  One 2-hour lecture:
  - Monday 14:00-16:00 in PX 201 (7-9 Leinster Street South)
  One 2-hour tutorial:
  - Thursday 09:00-11:00 in PX 201 (7-9 Leinster Street South)
  per week (11 weeks)
- **Module Coordinator:** Dr Tom Paskhalis (tom.paskhalis@tcd.ie)
- **Office Hours:** Thursday 11:00-13:00 in-person or online (booking required)
- **Tutorial Coordinator:** Dr Martyn Egan (eganm9@tcd.ie)

Learning Aims

This module provides foundational knowledge of computer programming concepts and software engineering practices. It introduces students to R and Python, two major programming languages used in data science and associated analytical workflows, with a focus on social science data and research questions.

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 be introduced to R and Python, two principal data science programming languages. This course covers basic and intermediate programming concepts, such as object types, functions, control flow, testing and debugging. Particular emphasis will be made on data handling and analytical tasks with a focus on problems in social sciences. Homeworks will include practical 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.

There are two main ways to install Jupyter Notebook on your local machine: pip and conda. Unless you have prior experience with Python, I recommend installing Anaconda distribution, which contains both R and Python as well as all the packages required for this course.

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 Jupyter Notebook, 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 few options to try:

- Visual Studio Code
- Atom
- Sublime Text

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

- RStudio - very popular (de-facto standard) 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 and Garrett Grolemund. 2017. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.* Sebastopol, CA: O'Reilly Media. http://r4ds.had.co.nz/
- Hadley Wickham. 2019. *Advanced R.* 2nd ed. Boca Raton, FL: Chapman and Hall/CRC. http://adv-r.had.co.nz/

While not explicitly covering either R or Python, this book is a great source of general historical and technical (but accessible) background on what computer code is (first edition is slightly dated, but would still work well):

- Charles Petzold. 2022. *Code: The Hidden Language of Computer Hardware and Software.* 2nd ed. Redmond, WA: Microsoft Press

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
- The Hitchhiker's Guide to Python
- Python For You and Me
- Python Wikibook
- Python 3 Documentation
- R Documentation
- R Inferno

Assessment Details

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

- Participation (tutorial attendance, 10% total)
- 4 Programming exercises (10% each, 40% total)
- Final project (50%)

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. Please, make sure to check that all cells in your notebook execute correctly and without error prior to submission.

Please make certain that you understand the submission procedure. Unexcused late submissions will be penalized in accordance with standard department policy.

All assignments are due by 12:00 Monday.

See module schedule summary below for the full list of due dates.

The final project will be due by 23:59 Friday, 16 December 2022.

Plagiarism

Plagiarism — defined by the College as the act of presenting the work of others as one's own work, without acknowledgement — is unacceptable under any circumstances. All submitted coursework must be **individual** and **original**. While the regulations you will find in the College policy on plagiarism largely describe assignments consisting of written text, note that similar guidelines apply to code submitted for assessment. Plagiarising computer code is as serious as plagiarising text and will have serious implications both within this class and in the real world (see Google LLC v. Oracle America, Inc. for one of the examples). While you may discuss general approaches to solutions with your peers, under no circumstances you are allowed to share and view each others code. Note that in case of an identified plagiarism all students whose code appears to come from the same source without giving due credit will be penalized. You can use online resources (e.g. Stack Overflow) but you need to give credit (and link) in the comments.

Module Schedule

Week 1: Introdution to Computation

Required Readings:

- Guttag Ch 1: Getting Started
- Jeannette M. Wing. 2006. Computational Thinking. *Communications of the ACM* 49 (3): 33–35. https://www.semanticscholar.org/paper/Computational-thinking-Wing/5cae1fd7937a1a14feb07b71f1b0e83d65ab9855

Additional Readings:

- Wickham & Grolemund Ch 1: Introduction
- McKinney Ch 1: Preliminaries

Week 2: R Basics

Required Readings:

- Wickham Chs 2: Data Structures, 3: Subsetting, 4: Vocabulary;
- Peng Chs 5: R Nuts and Bolts, 10: Subsetting R Objects, 11: Vectorized Operations;

Additional Readings:

- Matloff Chs 2: Vectors, 3: Matrices & Arrays, 4: Lists, 5: Data Frames, 6: Factors & Tables;

Week 3: Control Flow in R

Required Readings:

- Peng Ch 14: Control Structures;

Additional Readings:

- Matloff Ch 7: R Programming Structures;

Week 4: Functions in R

Required Readings:

- Peng Chs 15: Functions, 16: Scoping Rules;
- Wickham Ch 6: Functions;

Additional Readings:

- Wickham Chs 10: Functional Programming, 11: Functionals, 12: Function Operators;

Week 5: Debugging and Testing in R

Required Readings:

- Wickham Ch 9: Debugging & Exceptions;
- Peng Ch 20: Debugging;

Additional Readings:

- Matloff Ch 13: Debugging;

Week 6: Data Wrangling in R

Required Readings:

- Wickham & Grolemund Chs 7: Tibbles, 8: Data Import, 9: Tidy Data

Additional Readings:

- Peng Chs 13: Managing Data Frames, 18: Loop Functions

Week 8: Fundamentals of Python Programming I

Required Readings:

- Guttag Chs 2: Introduction to Python, 5: Structured Types and Mutability;

Additional Readings:

- Guttag Ch 3: Some Simple Numerical Programs;

Week 9: Fundamentals of Python Programming II

Required Readings:

- Guttag Chs 4: Functions, Scoping and Abstraction, 6: Recursion and Global Variables;

Additional Readings:

- Guttag Chs 7: Modules and Files, 8: Testing and Debugging, 9: Exceptions and Assertions;

Week 10: Data Wrangling in Python

Required Readings:

- McKinney Chs 4: NumPy Basics, 5: Getting Started with Pandas, 6: Data Loading, Storage and File Formats, 7: Data Cleaning and Preparation, 8: Data Wrangling: Join, Combine and Reshape;

Additional Readings:

- Guttag Ch 23: Exploring Data with Pandas;
- McKinney Ch 9: Plotting and Visualization;

Week 11: Classes and Object-oriented Programming

Required Readings:

- Guttag Chs 10: Classes and Object-oriented Programming;

Additional Readings:

- Bjarne Stroustrup. 1991. What is ”Object-Oriented Programming”? (1991 revised version). *Proceedings of the 1st European Software Festival,* https://stroustrup.com/whatis.pdf
- Wickham Ch 7: OO field guide;

Week 12: Performance and Complexity

Required Readings:

- Guttag Chs 11: A Simplistic Introduction to Algorithmic Complexity, 12: Some Simple Algorithms and Data Structures;

Additional Readings:

- Wickham Chs 16: Performance, 17: Optimising Code;

Table 1: **Module Schedule Summary**

Week	Date	Language	Topic	Due
1	12 September	-	Introduction to Computation	
2	19 September	R	R Basics	
3	26 September	R	Control Flow in R	
4	3 October	R	Functions in R	Assignment 1
5	10 October	R	Debugging and Testing in R	
6	17 October	R	Data Wrangling in R	
7	24 October	-	-	Assignment 2
8	31 October	Python	Fundamentals of Python Programming I	
9	7 November	Python	Fundamentals of Python Programming II	
10	14 November	Python	Data Wrangling in Python	Assignment 3
11	21 November	Python	Classes and Object-oriented Programming	
12	28 November	Python, R	Complexity and Performance	Assignment 4