

Jon Reades¹

¹The Bartlett Centre for Advanced Spatial Analysis

Welcome

The *Foundations of Spatial Data Science* (FSDS) module is an optional element of CASA's MSc programmes and is intended provide an introduction to *doing* data science in Python for students who are new to programming or whose previous exposure to code is fairly limited. The module seeks to enable students to access, understand, use, analyse, and communicate data in a spatial context. FSDS is *not* about pushing buttons, but about using logic, programming, and your growing analytical skills to tackle real-world problems in a creative, reproducible, and open manner.

FSDS is *not* easy: in order to make the most of the module—and the foundation that it provides both for Term 2 modules on the MSc *and* for post-Masters employment—you will need to work hard. This does not mean cramming before each practical, it means *practicing* between practicals, doing the readings, and watching the videos. UCL expectations for a Masters-level module is 150 hours of study time: there are 4 hours of timetabled activity per week, and about an hour of videos to watch before each workshop, leaving up to 100 hours of 'self-study'. By implication, you should expect to spend about 1.25 hours/day 'studying' for this module: reading, coding, and (above all) practicing.

In exchange for your hard work, the design of the module is informed by the need for analysts, planners, and geographers able to *think* computationally using programming, analytics, and data manipulation skills that are anchored in the needs of policy-makers, businesses, and non-profits. There is a severe skills shortage in this domain across all sectors and, consequently, significant opportunity for those who can 'make sense' of data+code.

To Dos

- Shift away from .values
- Swap Weeks 8 (Visualising Data) and 9 (Dimensions in Data) + explain rationale to students.
- Shift focus of (new) Week 9 & 10 to review/discussing the final assessment. Content still available to view/practice but not required/delivered.
- Require more active use of GitHub/Git — a submission in which it's easy to automate checks that they've created a repo and populated it with the completed notebooks?

- Refresh Code Camp and make it more self-test oriented. Use the self-test to set expectations about level of effort/preparation required (and whether or not taking the module will be useful).
- Standardise delivery of practical content by TAs: should have consistent approach across practicals.
- Point to cross-module content/recaps.
- Joint reading list?
- Investigate CoCalc as an alternative. What is situation with JupyterHub at UCL.

Acknowledgements

While this module is indebted to both feedback from students and colleagues over the years, several people played a particularly outsize role in my thinking and deserve special acknowledgement:

1. Dani (for help with Docker, geopandas, and any number of other tools with which I've had to familiarise myself)
2. Andy (for somehow knowing about all kinds of new web apps that I could use to create 'books' to support the module)
3. The Geocomp team at King's College London, who supported my hare-brained scheme to teach Geography undergrads to code and offered all manner of useful feedback on what we could/could not feasibly cover.