

I have several years of postgraduate teaching experience, primarily at the University of Oxford, but also the University of Birmingham. I am a certified Carpentries instructor, having been privileged to teach a wide variety of of courses in the past, including but not limited to: high-dimensional econometrics, open source statistical and general purpose software programming for social scientists, replication projects, demographic analysis, the basics of machine learning, an introduction to the command line, and life course research. Below I outline the short courses I am well positioned to offer in the future, either in person or remotely online.

1. Open and Reproducible (Social) Science: This course would comprehensively deal with all the issues associated with conducting Open, Transparent, and Reproducible Social Science. It would cover discussions related to the use of Free and Open Source Software (and the principles of version control), the on-going and forthcoming replication crisis in science, and outlines the tools and best practices for addressing researcher degrees of freedom. It also provides a quick practical introduction to these tools and best practices, in the form of: version control (Git), the command line (Bash), and typesetting (T_EX). **Length:** 4-6 Hours. **Programming:** Git, Bash, T_EX.

2. An Introduction to Typesetting in T_EX: This module gives an introduction to a popular typesetting language used within academic research (and quantitative research in particular). It roughly follows the ‘L^AT_EX in 127 Minutes’ book, and has interwoven exercises and interactive ‘live-programming’ elements. This course – with interactive elements and ‘live programming’ – would build on earlier courses taught for several years at the University of Birmingham, and then at the University of Oxford as part of the Oxford | Berlin Summer School in Open and Reproducible science. The objective is to get participants to a position where they can effortlessly craft their own academic documents using L^AT_EX. **Length:** 4-6 Hours. **Programming:** L^AT_EX.

3. The Foundations of Modern Demographic Analysis: This core course introduces the foundations of modern demography. It moves from the key concepts of age-standardized rates and exposure, through to Lexis Diagrams, and then through to cohort and period life table analysis. It optionally then advances to fertility and migration, before combining those two new topics with mortality to comprehensively create cohort component method based projections. Half of the class is lecture based, and half of it is practical, ‘lab-based’. **Length:** 6-18 Hours. **Programming:** R.

4. An Introduction to Python for the Social Sciences: This provides a comprehensive introduction to the Python programming language, useful for any interested in utilizing the power of a general purpose programming language for academic research. The course assumes no prior knowledge, moving from basic object types, to collections, to control statements and then functions. I propose four sessions of three hours each. At the end of each session will be a short (optional or compulsory) homework to be completed prior to the start of the next session. **Length:** 12 Hours. **Programming:** Python.

5. An Introduction to Social Data Science: This longer sequence of classes covers many of the topics listed above. It begins with an introduction to and review of the most commonly used language for social data science/computational social science: Python. It then talks about principles of version control and shell scripting for reproducible research in conjunction with Python as part of a holistic research pipeline. We then provide introductions to three key substantive areas of interest: machine learning, network science, and natural language processing (covering other submodules such as web-scraping and data visualisation along the way). The course will be provided as a sequence of jupyter notebooks, and involves a substantial element of live coding. Each session (between either two or four hours) will contain (optional or compulsory) homeworks which will be solved at the start of the following, subsequent class. The course can also conclude with an optional ‘Capstone’ type project which students can work towards and present together in groups. **Length:** 18-24 Hours. **Programming:** Python.