Programming Practices for Research in Economics

Lachlan Deer Julian Langer

Fall 2021

E-mail: pp4rs.contact@gmail.com
Dates: 1st September - 17th September
Class Days: daily, Monday - Friday

Web: pp4rs.github.io/2021-uzh
Class Hours: 9:30am-12:30pm, 2pm - 5pm
Class Room: To Be Announced

Motivation

Much of a researcher's time in modern economics and business research is spent in front of a computer 'doing' some form of computational analysis – be it to analyze data or to simulate economic models.

Until recently, there has been little emphasis on teaching early career researchers how to perform their computational tasks and manage the resulting projects in a structured and efficient way. Class exposure to programming languages is most often limited to the simple use of Stata and Matlab to solve 'toy' examples designed to illustrate a theoretical result or implement a method with known properties and ex-ante known results. These skills do not scale up in a straightforward manner to handle complex projects that make up research papers, PhD theses or typical work in government or private business settings. As a result, young economics and business researchers spend too much time wrestling with software and too little time doing research – where our comparative advantage lies.

This course is designed to improve learner's programming abilities and acquire skills to decrease their time wrestling with software. It is aimed at PhD students who expect to write their theses in a field that requires modest to heavy use of computation and analyzing data. Examples include applied microeconomics, econometrics, macroeconomics, quantitative marketing, quantitative finance and other fields that either involve real-world data or do not generally lead to analytical models with closed-form solutions.

The course introduces students to a new set of tools and programming methods that aim to reduce time spent programming while at the same time making programs more dependable and results reproducible. It draws extensively on techniques and tools that are the backbone of modern software development and large scale data science. Students gain insight into the the usefulness of these techniques and how to use them by means of hands-on examples from a wide variety of applications in economics and business research.

Target Audience

This course is intended for PhD students in economics and business who are transitioning from coursework to research. Next to your economics/business background, we will only assume that you have written small pieces of code before, like Stata .do-files or Matlab .m-files for problem sets in your Masters degree or first-year PhD classes. Knowledge of a specific programming language is not required.

A large part of this course is really about tool choice. We will take care in pointing out which language is most appropriate for which problem, and provide you with introductions to two popular choices for data- and computationally intensive computing. We also introduce a tool kit designed to improve the replicability of your code. The programming languages and tools introduced in the course are not the only choices available but some knowledge of these languages and best practices will make picking up others on your own relatively easy by providing a solid basic training.

Course Objectives

By the end of this course, students will be able to:

1. Use a computer′s command line to provide text instructions that can navigate around a computer′s file-system, copy and move files and edit new/existing files.
2. Explain different variable types and their advantages and disadvantages in Python and R.
3. Construct scripts that load, manipulate and visualize data in Python and R.
4. Implement statistical and economic models in Python and R.
5. Run Python and R inside a GUI and from the command line (including passing across complex arguments).
6. Explain the advantages of using a Version Control Software such as Git as opposed to 'manual' version control.
7. Manage a version controlled project using Git.
8. Explain what a workflow management system is and their advantages for economics and business research.
9. Manage a research project using the workflow management software Snakemake.

Learning objectives for specific modules will be provided within the Course Notes.

Evaluation

The course is evaluated on a pass/fail basis. There will be a final assignment that is due four weeks after the course concludes that utilizes the tools students learned to use in class. This assignment will count 100%. More information will be provided in the first class.

Rules of the Game

The class is designed to be 'hands-on' in the sense that you will be programming a lot of things *during the class*. We strongly believe the only way to learn programming is to do programming. Please bring your laptop with you to each session and install the required software before the course begins. Try to complete each activity we do in class and be prepared to ask and answer questions during class. Slides or notes will be made available at the beginning of each day, codes that solve exercises will be posted during or after the session.

Office Hours

Due to the intensive nature of the course, we have decided to not schedule office hours. Feel free to talk to us before and after each session throughout the course and ask many questions during each session.

Times and Locations

- Dates: Daily from 1st September until 17th September (excluding weekends)
- Morning Session: 9.30 - 12.30
- Afternoon Session: 14.00 - 17.00
- Location: TBA

Preliminary Program

The following is a preliminary program. It may be updated prior to the beginning of the course, an updated schedule will be forwarded before the course begins.

	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1:					
AM			Installation Help	Terminal	Basic Python
PM			Installation Help	Basic Python	Basic Python
Week 2:					
AM	Python: Numpy	Python: Plotting	Python: SciPy	Adv. Python	Version Control
PM	Python: Pandas	Python: Metrics	Webscraping	Version Control	R: Basics
Week 3:					
AM	R: Basics	R: Plotting	R: Econometrics	Build Tools	Project Kickstarter
PM	R: Data Analy.	R: Econometrics	Advanced R	Build Tools	Project Kickstarter

Students are expected to have completed the Installation Guide and successfully installed all required software for the course prior to the first day, Thursday 2nd September. The instructors will hold an "Installation Help" session where students can drop in and get additional help configuring their computers if needed.

Brief Topic Outlines

Terminal

The Unix shell emphasizes providing text-based instructions to computers. It's power lies in the ability to do complex and repetitive tasks with a few keystrokes. Use of the shell is the fundamental starting point into using a wide range of other powerful tools and computing resources (including "high-performance computing" and cloud computing resources).

Python Programming Language

Python is one of the most popular and fastest growing programming languages because of its ease of use and power. It has become the most used statistical software in recent years due to its extensive machine learning libraries. These modules have two distinct goals: (1) introduce basic programming syntax, and (2) introduce specific packages that are useful for computational analysis and data-driven research. We introduce basic programming syntax using the Python language because of its simplicity to get started for novice users. The introductions to specific packages are designed to highlight how to solve problems that are typically encountered by economics and business researchers – such as solving models using computational techniques, the automated collection of data from websites and applied econometric modeling. We emphasize best practice techniques as we progress through the material.

Version Control

Version control is the lab notebook of the digital world: it's what software development teams, and increasingly applied researchers, use to keep track of what they've done and to collaborate. The idea of a formal version control software is to take systematic 'snapshots' of our work and store the changes so one can easily track the development of their work and, when necessary go to a previous version of a file. We introduce the basics of version control using Git, and show how to use it with online repositories such as GitHub to share our work with others. We then look at more advanced version control techniques such as maintaining multiple parallel versions of our files so that we can test out changes, and then either revert back to an old state or integrate the new updates into our stable code base.

The R Programming Language

The goal of this lesson is to teach novice programmers to write modular code and best practices for using R for data analysis. R is free and has a wide array of third-party packages for almost all imaginable statistical applications. The emphasis of these materials is to give attendees a strong foundation in the fundamentals of R and to teach best practices for scientific computing: breaking down analyses into modular units, task automation, and encapsulation. This workshop will focus on teaching the fundamentals of the programming language R, data wrangling, data visualization and regression techniques common to applied researchers.

Build Tools

Build Tools can run commands to read files, process these files in some way, and write out the processed files in a specified order. For example, we can:

- Run analysis scripts on raw data files to get data files that summarize the raw data;
- Run visualization scripts on data files to produce plots and statistical tables; and
- Parse and combine text files, tables and plots to create papers.

Most importantly, Build Tools track the dependencies between the files they create and the files used to create them. This allows build tools to only recompute necessary steps after a data or code file has been changed – without the need to run the whole project from the beginning.