

Programming for Economists

Fall 2016

Syllabus

August 25, 2016

1 Instructor

Tyler Abbot
Email: tyler.abbot@sciencespo.fr
Office Hours: by appointment.

2 Course Description

This class will aim to provide first year graduate students with the tools necessary to perform basic computation. Given the broad range of available tools, this course will seek to provide in depth instruction in *programming* instead of a programming language. That being said, we will work mainly in R and introduce Python if time permits.

By the end of the course students should know how to design pseudo-code to outline a problem solution, read and write basic programs in R and Python, know how to run and debug basic programs in R and Python, should have R and Python installed on their computers, and most importantly should feel comfortable tackling a computational problem from start to finish, from algorithm design to choosing the most effective tools to sharing the results.

Teaching Objectives:

- Introduce students to the basic workings of a computer: logic, loops, memory, processing, etc.
- Provide elementary practice in the programming process:
 1. Problem definition.
 2. Algorithm design.
 3. Pseudo-code generation.

4. Language selection.
 5. Coding.
 6. Optimization.
 7. Results presentation.
- Help students to gain a basic knowledge of the use of R and Python.

3 Course Requirements

Little to no prior programming knowledge is required. All texts and resources will be readily available online. Students should come to class with a laptop capable of running open source software and with plenty (at least 5 gigabytes) of free disk space before installing any of the required packages. If you do not have a laptop, you can always work using internet based resources (e.g. PythonAnywhere) on Sciences Po computers, looking on with others in class.

Suggested Reading: There is no required text, but the following are useful resources:

- Chambers, John. "Software for Data Analysis: Programming with R".
- Braun, W. John and Murdoch, Duncan J. "A First Course in Statistical Programming with R".
- Dalgaard, Peter. "Introductory Statistics with R".
- quant-econ.net
- Press, William H., et. al. "Numerical Recipes: The Art of Scientific Computing".

4 Grading

The course will be graded on a pass/fail basis. **To pass a student must submit a strong majority (at least 6) of the problem sets and both the project and take home final.**

Weekly problem sets will be given on a completion basis. Copying is easy, but your instructors next semester will expect you to know these topics, so it is in your interest to at least try.

There will be a semester project to be completed in groups or alone.

There will be a take home final exam to be returned after the final course.

5 Proposed Schedule

- **Week 1: Introduction to programming.** How a computer "thinks", how you communicate with a computer (interpreted versus compiled), introduction to open source and GitHub, introduction to R.
- **Week 2: Basic problem solving in R.** Example based and exercise based practice in native R.
- **Week 3: Functions in R.** Functions, arguments and keyword arguments, nested functions, possibly dynamic programming.
- **Week 4: Linear algebra in R.** Basic matrix and vector operations.
- **Week 5: Creating your own packages.** An introduction to devtools, code documentation, and testing.
- **Week 6: Dealing with data.** An introduction to the R dataframe.
- **Week 7: Fitting linear models in R.** Using S syntax to fit linear models using R dataframes.
- **Week 8: Convex optimization.** An introduction to gradient based optimization and built in optimization tools in R.
- **Week 9: An introduction to Python.** How is Python different from R and how to use it.
- **Week 10: Python packages.** Introduction to NumPy, SciPy, Pandas, and matplotlib.
- **Week 11: Workshop or multiprocessing.** Either in class work to finish project or parallel processing in Python.
- **Week 12: Project presentation or multiprocessing.** Either project presentations or parallel processing in Python.

6 Academic Honesty

From the Plagiarism Charter and Règlement de Scolarité Sciences Po:

The definition of the word plagiarism Sciences Po submits to students is the one that is cited in the Règlement de Scolarité (School Education Rules): Plagiarism occurs when the student presents a paper in which his own thought can't be separated from other authors' thought. This happens when the student doesn't quote a group of consecutive words (starting from 5), a paraphrase or a translation, or a copied abstract (article 12 Intellectual honesty).

When a case of plagiarism is detected, the student will receive a grade of 0/20 for the module.

”Les sanctions sont fixées par un décret en Conseil d’Etat. Elles peuvent aller de l’avertissement à l’exclusion définitive de tout établissement public d’enseignement supérieur.”