Geocomputation with open-source software under Linux: Hands-on training for computational thinking and skills

Giuseppe Amatulli*
giuseppe.amatulli@yale.edu
School of Forestry & Environmental Studies, Center for Research Computing - Yale University
New Haven, CT, USA

Tushar Sethi*
t.sethi@spatial-ecology.net
Spatial Ecology
Redruth, Cornwall, UK

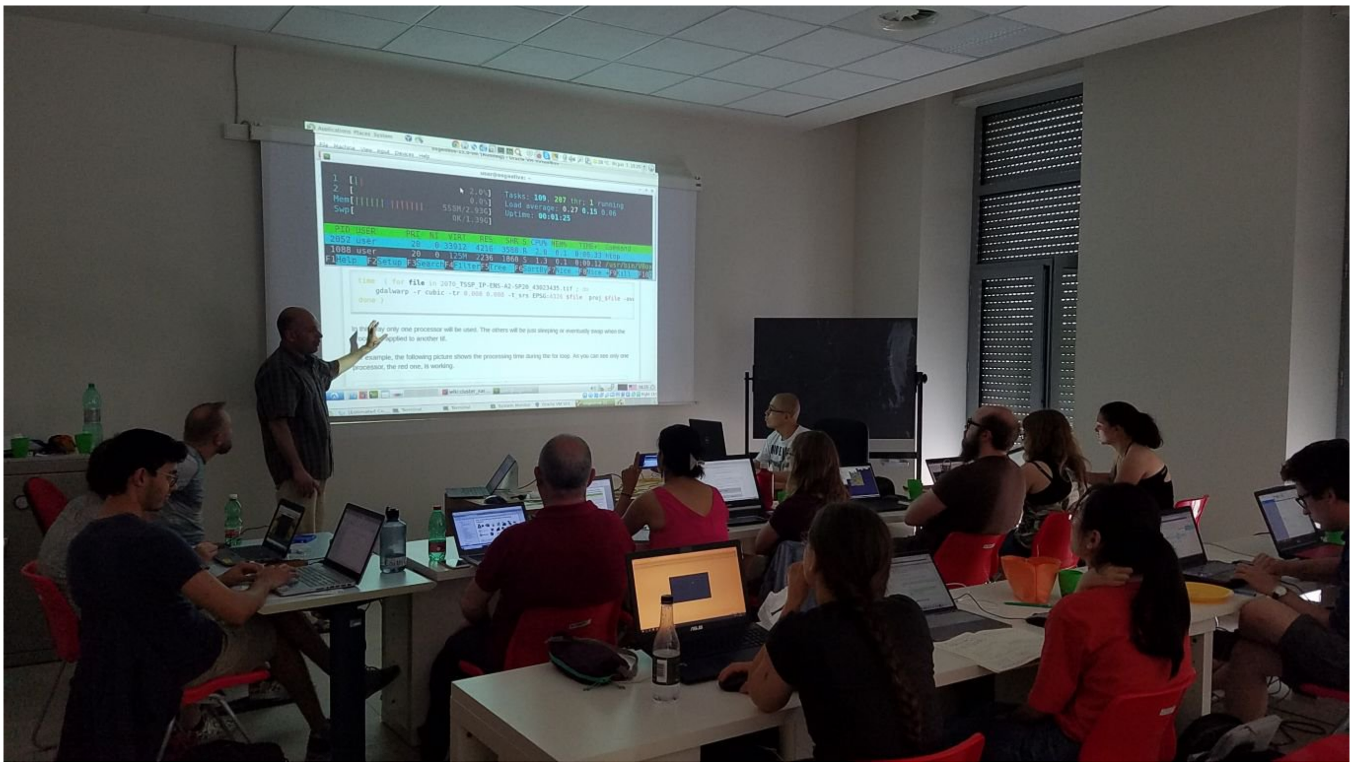

Figure 1: International Summer School in "Geocomputation Using Free Open Source Software" 3rd - 7th June 2019, Matera, Italy, organized by www.spatial-ecology.net

KEYWORDS

geocomputation courses, learning programming, computational thinking

*Both authors contributed equally to this research.

GeoEd'19, November 5, 2019, Chicago, IL, USA

ACM ISBN 978-1-4503-6952-7/19/11…$15.00
https://doi.org/10.1145/3356393.3365371

ACM Reference Format:
Giuseppe Amatulli and Tushar Sethi. 2019. Geocomputation with open-source software under Linux: Hands-on training for computational thinking and skills. In *1st ACM SIGSPATIAL International Workshop on Geocomputational Thinking in Education (GeoEd'19), November 5, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3356393.3365371

1 INTRODUCTION

In recent years, there has been an explosion of geo-datasets derived from an increasing number of remote sensors, field instruments, sensor networks, and other GPS-equipped "smart" devices. Processing "Big GeoData" of this kind requires flexible tools that combine efficient programming, on either personal or supercomputers. Open Source geo-software such as GRASS, GDAL/OGR, PKTOOLS, CDO and Orfeo ToolBox allow for the fast and efficient processing of geo-datasets found as rasters and vectors. These programming languages can be integrated into complex workflows using a BASH or Python interface.

Mastering skills for the analysis of spatio-temporal data is fundamental for most environmental and engineering disciplines. However, academic curricula often do not equip students for utilising these new data streams and programming languages. People trained in scientific fields are not normally endowed with the level of programming proficiency required for high performance computing but they can reap tremendous benefits from geodata processing, once such skills are acquired.

GIS and Remote Sensing courses that deal with geodata are typically based on proprietary and/or graphical user interface-based software. These platforms are limited in their scope of application and cannot be user-modified to meet sophisticated data needs. As a result, cross-disciplinary researchers and analysts are seldom able to design and implement complex geospatial workflows. In the absence of suitable courses, acquiring programming skills often requires long-term independent learning and strong personal motivation.

2 METHOD

To address the aforementioned educational need, the team at www.spatial-ecology.net has been organising intensive geospatial training workshops for the past 15 years. The curriculum is based on open source programming under a standardised Linux OS, with training delivered through pre-installed software, and readily available tutorials and exercises. The courses comprise of lectures, personalised tutorials and exercises, and round-table discussions. To aid the learning process, all course documentation and exercises are made accessible online (through wikis at www.spatial-ecology.net). These tutorials are compiled by researchers and experts who use open source tools in their respective professions.

Our approach to teaching data analysis is unique, as it integrates multiple programming languages such as AWK, BASH, PYTHON, GRASS, GDAL/OGR, PKTOOLS, CDO, Orfeo ToolBox and other software to build workflows. Nevertheless, our teaching methods complement the work of existing communities, such as OSGeo, which also provide specific tutorials and tools for the analysis of geographic data. From the outset, basic programming concepts that use command-line utilities to process large data sets are taught on our course. With simple scripts, we demonstrate how to automate essential tasks and modify programs to solve specific problems, all while achieving optimal performance and dependable results. Despite first impressions as a computer programming course, the curriculum is, in fact, focused on the application of scientific concepts: participants are guided towards solving discipline-specific problems with the help of customised computer scripts.

Rather than focusing on one programming language or software package, we introduce several tools and languages, and teach how to combine the most useful functions from each. Most of the scripts taught are based on powerful geo-tools that act as modular building blocks for the data processing tasks at hand. Each language/tool is taught under the following structure:

- Language syntax, including details on various flags and options
- Accessing help manuals and understanding available documentation
- Explaining common syntax problems and procedures to solve them (debugging)
- Structuring a script to connect various tools and/or languages
- Working with computed outputs: how to visualise and interpret data
- Assessing and validating results based on a comprehensive sketch of the analysis

Nowadays, with the advent of cloud computing, large amounts of data can be processed and stored in remote servers. Moreover, multicore computing allows several processes to run simultaneously. To enable course participants to access the most advanced computing technologies, we also offer an introduction to cluster computing tools towards the end of each course. Training includes topics such as the transformation of a simple *for loop* into *multicore for loops*, which allow the simultaneous processing of massive datasets. Specific R libraries (*foreach, doParallel*) and BASH commands (*xargs, parallel*) are covered in the material. The teaching of this advanced module is contingent on the participants' skill levels.

3 CONCLUSION

Learning programming languages enhances problem-solving skills, and can inspire new research ideas by stimulating critical, analytical and lateral thinking. It also enables users to run preliminary tests on the viability of analytical objectives. We believe that the computational thinking process can be a natural outcome of learning code, and the former cannot be initiated without mastering the latter. This concept is akin to learning a foreign language, where a person cannot think or dream in that language without first learning its structural rules.

Overall, as trainers, the diversity of our backgrounds adds additional value to the learning experience. We come from various professions, such as geoinformatics, quantum chemistry, neuroscience, ecology and agriculture, and have taught our course globally. This rich experience uniquely positions us to understand different perspectives and learning needs. Despite our disparate backgrounds, we are equipped to teach a practical and uniform approach to data analytics, and offer our students a truly well-rounded learning opportunity. Information for new courses can be found at www.spatial-ecology.net or by registering to the Spatial-Ecology mailing list at https://lists.osgeo.org/mailman/listinfo/spatial-ecology.