Advanced strategies for ecological data analysis

Dr. Christian Zang
Assistance Professorship for Land Surface-Atmosphere Interactions
Technical University Munich

Three weekly hours blocked on six wednesdays (27.4., 4.5., 11.5., 1.6., 8.6., 15.6.2016), 9am-5pm, Computer room S6 forestry building Weihenstephan

This hands-on course with six whole day installments explores strategies for efficiently analyzing ecological data sets, documenting your workflow and making it reproducible, speeding up your computations, and creating nice visualisations. The focus is on the R language and environment for statistical computing, but we will get to know some useful friends along our way.

Basic level understanding of R is required for this course (e.g. you should know how to write your own functions).

UNIX toolset, UNIX mindset

In the first installment we will explore why most data scientist choose a flavour of UNIX to get their work done. We will dive into the philosophy behind UNIX, get to know some of the most important UNIX standard tools and concepts and see how to leverage them for handling large data sets and complex data structures. Later, we will see how to translate these concepts into R and see why R is the way it is. We will see how to talk to external UNIX processes from inside R, and how to use R to write command line programs.

Tools covered: UNIX Shells, pipes, redirects, regular expressions, UNIX standard programs, magrittr

Get functional

Functional programming is an expressive and concise way of telling a computer what to do. This session builds the fundamentals of R and aims at introducing functional programming styles (as opposed to imperative styles, potentially leading to cleaner and more readable code.

Additionally, we look at object oriented programming, another strategy to increase readability, ease of maintainance and reuseability of your code.

*Tools covered: the *apply family, dplyr, method dispatch, S3/S4 classes*

The need for speed

Sometimes fast isn't fast enough. Starting with pure R solutions to speed up time-intensive computations, we will eventually explore foreign language interfaces, and will pick up some useful C++ along the way. We will also look at basic parallelisation.

Tools covered: data.table, byte-compiling, foreach, basic C++, Rcpp, RcppArmadillo

Play it again, Sam

Reproducability is an important aspect of a good data analysis strategy. We will look at options for combining code with documentation. Another important strategy is packaging for creating self-contained and shareable versions of your efforts. Furthermore, we will see how to travel in time and track our progress using a version control system.

Tools covered: Roxygen, R-Markdown, Sweave, R-Packages, packrat, Git

That looks nice!

At the end, we all want nice looking graphs. While plot() often does get us where we want, there are sleek and flexible alternatives. We will look at ggplot2 and its possibilities and limitations, and also check out the new web-targeted plotting engine ggvis, and the interactive shiny framework.

Tools covered: plotting fundamentals, ggplot2, ggvis, shiny

Hi, I'm Julia

The new and open-source Julia language for technical computation emerges as a fast and flexible alternative to R for large scale data science. In this session we will get to know Julia, and see how it compares to R.

Tools covered: basic Julia, statistical models in Julia, plotting with Julia