Geophysical Research Abstracts
Vol. 19, EGU2017-14375, 2017
EGU General Assembly 2017

Writing your own scientific R packages increases reproducibility, research quality and output

Koen Hufkens and Andrew D. Richardson
Harvard University, Richardson Lab, Cambridge, United States (koen.hufkens@gmail.com)

The use of R as a statistics and general purpose computing platform exploded over the past 10 years. This democratization of easy processing tools is remarkable as it should increase the level of transparency and reproducibility of science. However, reproducibility is not yet common practice for many researchers. Most of them are unaware of simple workflows which one can leverage to process (open) data effectively and transparently.

We draw from experience in writing mostly small R packages and graphical user interfaces to manage, download, visualize and analyze open data within the context of (phenology) research. These R packages include the amerifluxr package, facilitating Ameriflux data retrieval (http://ameriflux.lbl.gov/), visualization and basic summary statistics; the daymetr package interfacing with the ORNL DAAC for Daymet data retrieval (https://daymet.ornl.gov/) for e.g. phenology modelling; and the snotelr package which retrieves snow water equivalent measurements from the SNOTEL network (https://www.wcc.nrcs.usda.gov/snow/) and calculates basic snow phenology metrics.

All these packages represent small incremental work, but serve us and others on a daily basis. Unlike unbundled scripts in personal directories these packages are not static as they are exposed to the scientific community and will evolve or facilitate other people in their research.

Here, we advocate for: the minimum viable R package concept (publish fast and often), version control (e.g. git) and continuous integration to keep track of changes and collaborate, being open in your coding practices in order to accelerate your own and other research as well as track errors to ensure repeatability. Although the additional effort required in formalizing R code will pay off in terms of efficiency and collaborations we further argue for maximizing a package's impact by publishing it in software journals (e.g. Journal of Open Source Software) or making releases citable (e.g. Zenodo).