A protocol for data exploration to avoid common statistical problems

Alain F. Zuur, Elena N. Ieno and Chris S. Elphick

Summary:

In this paper Zuur et al. sets out to call out, and offer solutions to some common mistakes perpetrated by graduate students in ecology. Although the specifically mentioned target is graduate students in ecology the message given throughout the paper is applicable to students in any discipline who need brushing up on their statistics.

The authors state that they commonly have noticed that graduate students stumble their way through data analysis, getting tracked into a mindset and not taking a step back to see if what they are doing to their data (e.g. axis transformations or specific tests) are actually responsible given the data. In response to this problem the authors lay out a system for "data exploration," or the act of getting a feel for the data before making any transformations or doing any tests.

Among the recommended methods of data exploration mentioned are box plots and cleveland box plots for attempting to get a picture of the shape of the data, histograms to search for normality to multi-framed scatter plots of different variables to try and further tease out dependence. The authors state that if these steps are taken, which ones depending on the data, and combined with the responsible knowledge of the researcher higher quality quantitative analysis will result.

Ten Simple Rules for Reproducible Computational Research

Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, Eivind Hovig

This paper deals with the issue of reproducibility in data driven publications. Starting with the quote "replication is the cornerstone of a cumulative science" provides a blunt explanation of the motivation behinds efforts to make your research reproducible. The authors acknowledge the fact that we live in a world where pressures are mounting on researchers to publish more and faster, but they make the claim that the opportunity cost of making an effort to thoroughly document your research and make it entirely transparent how you arrived at your result is worth it. They repeatedly raise the point that the struggles a researcher will most likely encounter trying to simply reproduce their own research will more than make the effort worth it, let alone in the increased confidence it gives reviewers and citers in your results.

The authors list ten rules to follow in order to maximise your works reproducibility. These range from the obvious "for every result, keep track of how it was produced" to the perhaps less often considered "archive exact versions of all external programs used." Inside each of these steps the motivation is stated and best practices laid out. E.g. Motivation: trying to reproduce scripts written for Python 2.7 on a machine with 3.1 won't work too well (darn print foo vs. print(foo)), best practice: note what program versions used or alternatively archive a complete virtual machine at the state the original script was run.

Ultimately the listed rules comprise a very thorough list of actions to perform, or at the very least keep at the top of your mind, when engaging in computational research. If the rules are respected it will only go to further the credibility of the researcher, get them published and increase their h-index.

How I can use this

Both of these papers resonated very deeply with me (something pretty darn rare in scientific literature). The first paper very much so because of the fact that a large amount of the research I have conducted thus far has been in ecology. I even went to the length of sending the paper to the director of my lab here at UVM saying he should distribute it to his graduate students. I absolutely see students wielding statistics in ways that are irresponsible. I have many times had a graduate student call me over to look at the results of an ANOVA test that they ran, only to not have an answer to the question “why did you run an ANOVA?”

I am myself (even though I am a statistics major) also guilty of this statistical ignorance, but I think on the other end. Some famous guy once said “the more I learn the more I realize how much I don’t know”* and I tend to follow that. The more I learn about statistics the more hesitant I am to haphazardly through it on every single set of data or data-like object I see. Being entirely honest about this, I feel like it is a result of fear. After seeing so many poorly informed wield-ings of statistics, I am scared that I am myself going to be guilty of failing to account for confounding factors, or more frequently, simply not remembering what the heck an ANOVA test is. As a result, I often skimp on statistics, the question is then, which is the greater evil?

The protocol for data exploration paper was really valuable to me because it lays out (in a way I probably should have figured out for myself) a set of steps to orient yourself with data. Having a solid grasp of the shape of your data is so very valuable in my opinion. I would absolutely be more bold with my statistical analyses if I completely understood the entity of my data. I will, without a doubt, refer back to this paper countless times in my future data driven exploits.

The reproducibility paper also hit close to home. As I was reading it I could not help but think about the crutches that I have so recently been attempting to wean myself off of, R and iPython notebooks. Both of these tools allow you to segment your code, run parts at a time and see everything instantly, but what they also allow you to do is run a section of your code, get an error, fix it by running something through a loop and in the process accidentally delete a crucial line. When you then start fresh and run section by section (or line by line for R) you can’t get back to where you started. This tends to lead to me huddling at the end of my iPython notebook making changes and petrified to re-run prior cells in fear of losing the magically concocted dataframe or plot I have in my most recent cell.

How irresponsible of me is it to be scared of my own scripts? Imagine how quickly a reviewer would tear apart my bipolar iPython notebooks. While that is only one of the 10 bullet points I think the same concepts permeated through the whole paper: there is no room for

crossed fingers in computational research, write code you would be confident in showing to anyone, and make sure that it is like that at all stages.

I will be constantly referring back to these papers throughout the rest of this class, but also throughout my career as a computational researcher. While they do bring to the surface some of my biggest weaknesses in my coding/general analysis, they also provide clear structured methods of fixing those weaknesses.