# STATISTICS DONE WRONG

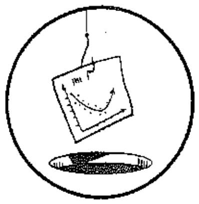
THE *WOEFULLY COMPLETE* GUIDE

ALEX REINHART

2015 no starch press

# 10

# EVERYBODY MAKES MISTAKES



Until now, I have presumed that scientists are capable of making statistical computations with perfect accuracy and err only in their choice of appropriate numbers to compute. Scientists may misuse the results of statistical tests or fail to make relevant computations, but they can at least calculate a *p* value, right?

Perhaps not.

Surveys of statistically significant results reported in medical and psychological trials suggest that many *p* values are wrong and some statistically insignificant results are actually significant when computed correctly.[1,2] Even the prestigious journal *Nature* isn't perfect, with roughly 38% of papers making typos and calculation errors in their *p* values.[3] Other reviews find examples of misclassified data, erroneous duplication of data, inclusion of the wrong dataset entirely, and other mix-ups, all concealed by papers that did not describe their analysis in enough detail for the errors to be easily noticed.[4]

These sorts of mistakes are to be expected. Scientists may be superhumanly caffeinated, but they're still human, and the constant pressure to publish means that thorough documentation and replication are ignored. There's no incentive for researchers to make their data and calculations available for inspection or to devote time to replicating other researchers' results.

As these problems have become more widely known, software tools have advanced to make analysis steps easier to record and share. Scientists have yet to widely adopt these tools, however, and without them, thoroughly checking work can be a painstaking process, as illustrated by a famous debacle in genetics.

## Irreproducible Genetics

The problems began in 2006, when a new genetic test promised to allow chemotherapy treatments to be carefully targeted to the patient's specific variant of cancer. Duke University researchers ran trials indicating that their technique could determine which drugs a tumor would be most sensitive to, sparing patients the side effects of ineffective treatments. Oncologists were excited at the prospect, and other researchers began their own studies. But first they asked two biostatisticians, Keith Baggerly and Kevin Coombes, to check the data.

This was more difficult than they expected. The original papers did not give sufficient detail to replicate the analysis, so Baggerly and Coombes corresponded with the Duke researchers to get raw data and more details. Soon they discovered problems. Some of the data was mislabeled—groups of cells that were resistant to a drug were marked as sensitive instead, and vice versa. Some samples were duplicated in the data, sometimes marked as both sensitive and resistant. A correction issued by the Duke researchers fixed some of these issues but introduced more duplicated data at the same time. Some data was accidentally shifted by one so that measurements from one set of cells were used when analyzing a different cell line. Genetic microarrays, which I discussed earlier in the context of pseudoreplication, varied significantly between batches, and the effect of the microarray equipment could not be separated from the true biological differences. Figures allegedly showing results for one drug actually contained the results for a different drug.

In short, the research was a mess.[5] Despite many of the errors being brought to the attention of the Duke researchers, several clinical trials using the genetic results began, funded by the National Cancer Institute. Baggerly and Coombes attempted to publish their responses to the research in the same academic journals that published the original research, but in several cases they were rejected—groundbreaking research is more interesting than tedious statistical detail. Nonetheless, the National Cancer Institute caught wind of the problems and asked Duke administrators to review the work. The university responded by creating an external review committee that had no access to Baggerly and Coombes' results. Unsurprisingly, they found no errors, and the trials continued.[6]

The errors attracted serious attention only later, some time after Baggerly and Coombes published their discoveries, when a trade magazine reported that the lead Duke researcher, Anil Potti, had falsified his résumé. Several of his papers were retracted, and Potti eventually resigned from Duke amid accusations of fraud. Several trials using the results were stopped, and a company set up to sell the technology closed.[7]

The Potti case illustrates two problems: the lack of reproducibility in much of modern science and the difficulty of publishing negative and contradictory results in academic journals. I'll save the latter issue for the next chapter. Reproducibility has become a popular buzzword, and you can probably see why: Baggerly and Coombes estimate they spent 2,000 hours figuring out what Potti had done and what went wrong. Few academics have that kind of spare time. If Potti's analysis software and data were openly available for inspection, skeptical colleagues would not be forced to painstakingly reconstruct every step of his work—they could simply read through the code and see where every chart and graph came from.

The problem was not just that Potti did not share his data readily. Scientists often do not record and document the steps they take converting raw data to results, except in the often-vague form of a scientific paper or whatever is written down in a lab notebook. Raw data has to be edited, converted to other formats, and linked with other datasets; statistical analysis has to be performed, sometimes with custom software; and plots and tables have to be created from the results. This is often done by hand, with bits of data copied and pasted into different data files and spreadsheets—a tremendously error-prone process. There is usually no definitive record of these steps apart from the overstressed memory of the graduate student responsible, though we would like to be able to examine and reproduce every step of the process years after the student has graduated.

## Making Reproducibility Easy

Ideally, these steps would be *reproducible*: fully automated, with the computer source code available for inspection as a definitive record of the work. Errors would be easy to spot and correct, and any scientist could download the dataset and code and produce exactly the same results. Even better, the code would be combined with a description of its purpose.

Statistical software has been advancing to make this possible. A tool called Sweave, for instance, makes it easy to embed statistical analyses performed using the popular R programming language inside papers written in LaTeX, a typesetting system commonly used for scientific and mathematical publications. The result looks just like any scientific paper, but another scientist reading the paper and curious about its methods can download the source code, which shows exactly how all the numbers and plots were calculated. But academic journals, which use complicated typesetting and publishing systems, do not yet accept Sweave publications, so its use is limited.

Similar tools are emerging for other programming languages. Data analysts using the Python programming language, for example, can record their progress using the IPython Notebook, which weaves together text descriptions, Python code, and plots and graphics generated by the Python code. An IPython Notebook can read like a narrative of the analysis process, explaining how data is read in, processed, filtered, analyzed, and plotted, with code accompanying the text. An error in any step can be corrected and the code rerun to obtain new results. And notebooks can be turned into web pages or LaTeX documents, so other researchers don't need to install IPython to read the code. Best of all, the IPython Notebook system has been extended to work with other languages, such as R.

Journals in heavily computational fields, such as computational biology and statistics, have begun adopting code-sharing policies encouraging public posting of analysis source code. These policies have not yet been as widely adopted as data-sharing policies, but they are becoming more common.[8] A more comprehensive strategy to ensure reproducibility and ease of error detection would follow the "Ten Simple Rules for Reproducible Computational Research," developed by a group of biomedical researchers.[9] These rules include automating data manipulation and reformatting, recording all changes to analysis software and custom programs using a software version control system, storing all raw data, and making all scripts and data available for public analysis. Every scientist has experienced the confusion of reading a paper and wondering, "How the hell did they get *that* number?", and these rules would make that question much easier to answer.

That's quite a lot of work, with little motivation for the scientist, who already knows how the analysis was done. Why spend so much time making code suitable for *other* people to benefit from, instead of doing more research? There are many advantages. Automated data analysis makes it easy to try software on new datasets or test that each piece functions correctly. Using a version control system means you have a record of every change, so you're never stuck wondering, "How could this code have worked last Tuesday but not now?" And a comprehensive record of calculations and code means you can always redo it later; I was once very embarrassed when I had to reformat figures in a paper for publication, only to realize that I didn't remember what data I had used to make them. My messy analysis cost me a day of panic as I tried to re-create the plots.

But even if they *have* fully automated their analysis, scientists are understandably reluctant to share their code. What if a competing scientist uses it to beat you to a discovery? Since they aren't required to disclose their code, they don't have to disclose that they used yours; they can get academic credit for a discovery based mostly on your work. What if the code is based on proprietary or commercial software that can't be shared? And some code is of such miserable quality that scientists find it embarrassing to share.

The Community Research and Academic Programming License (CRAPL), a copyright agreement drafted by Matt Might for use with academic software, includes in its "Definitions" section the following:

2. "The Program" refers to the medley of source code, shell scripts, executables, objects, libraries and build files supplied to You, or these files as modified by You.

   [Any appearance of design in the Program is purely coincidental and should not in any way be mistaken for evidence of thoughtful software construction.]
3. "You" refers to the person or persons brave and daft enough to use the Program.
4. "The Documentation" refers to the Program.
5. "The Author" probably refers to the caffeine-addled graduate student that got the Program to work moments before a submission deadline.

The CRAPL also stipulates that users must "agree to hold the Author free from shame, embarrassment, or ridicule for any hacks, kludges, or leaps of faith found within the Program." While the CRAPL may not be the most legally rigorous licensing agreement, it speaks to the problems faced by authors of academic code: writing software for public use takes a great deal more work than writing code for personal use, including documentation, testing, and cleanup of accumulated cruft from many nights of hacking. The extra work has little benefit for the programmer, who gets no academic credit even for important software that took months to write. And would scientists avail themselves of the opportunity to inspect code and find bugs? Nobody gets scientific glory by checking code for typos.

## Experiment, Rinse, Repeat

Another solution might be replication. If scientists carefully recreate the experiments of other scientists from scratch, collecting entirely new data, and validate their results—a painstaking and time-consuming process—it is much easier to rule out the possibility of a typo causing an errant result. Replication also weeds out fluke false positives, assuming the replication attempt has sufficient statistical power to detect the effect in question. Many scientists claim that experimental replication is the heart of science; no new idea is accepted until it is independently tested and retested around the world and found to hold water.

That's not entirely true. Replication is rarely performed for its own sake (except in certain fields—physicists love to make more and more precise measurements of physical constants). Since replicating a complicated result may take months, replication usually happens only when researchers need to use a previous result for their own work. Otherwise, replication is rarely considered publication worthy. Rare exceptions include the Reproducibility Project, born out of increasing concern among psychologists that many important results may not survive replication. Run by a large collaboration of psychologists, the project has been steadily retesting articles from prominent psychology journals. Preliminary results are promising, with most results reproduced in new trials, but there's a long way to go.

In another example, cancer researchers at the pharmaceutical company Amgen retested 53 landmark preclinical studies in cancer research. (By "preclinical" I mean the studies did not involve human patients, because they were testing new and unproven ideas.) Despite working in collaboration with the authors of the original papers, the Amgen researchers could reproduce only six of the studies.[10] Bayer researchers have reported similar difficulties when testing potential new drugs found in published papers.[11]

This is worrisome. Does the trend hold true for less speculative kinds of medical research? Apparently so. Of the top-cited research articles in medicine, a quarter have gone untested after their publication, and a third have been found to be exaggerated or wrong by later research.[12] That's not as extreme as the Amgen result, but it makes you wonder what major errors still lurk unnoticed in important research. Replication is not as prevalent as we would like it to be, and the results are not always favorable.

**TIPS**

- Automate your data analysis using a spreadsheet, analysis script, or program that can be tested against known input. If anyone suspects an error, you should be able to refer to your code to see exactly what you did.
- Corollary: Test all analysis programs against known input and ensure the results make sense. Ideally, use automated tests to check the code as you make changes, ensuring you don't introduce errors.
- When writing software, follow the best practices for scientific computing: *http://www.plosbiology.org/article/info:doi/10.1371/journal.pbio.1001745*.
- When using programs and scripts to analyze your data, follow the "Ten Simple Rules for Reproducible Computational Research."[9]
- Use a reproducible research tool like Sweave to automatically include data from your analysis in your paper.
- Make all data available when possible, through specialized databases such as GenBank and PDB or through generic data repositories such as Dryad and Figshare.
- Publish your software source code, spreadsheets, or analysis scripts. Many journals let you submit these as supplementary material with your paper, or you can deposit the files on Dryad or Figshare.