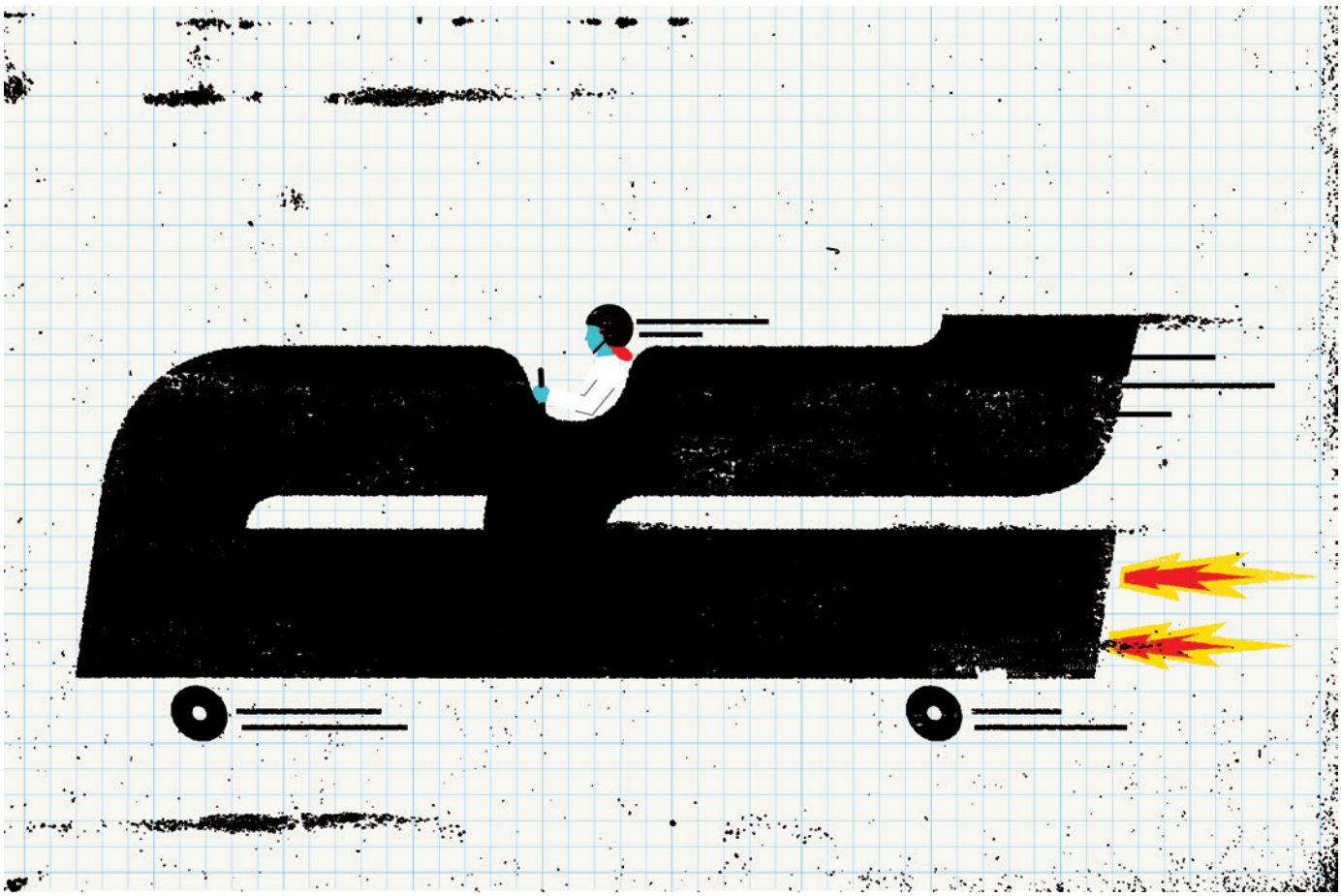
TOOLBOX

PROGRAMMING TOOLS: ADVENTURES WITH R

A guide to the popular, free statistics and visualization software that gives scientists control of their own data analysis.

ILLUSTRATION BY THE PROJECT TWINS



BY SYLVIA TIPPMANN

For years, geneticist Helene Royo used commercial software to analyse her work. She would extract DNA from the developing sperm cells of mice, send it for analysis and then fire up a package called GeneSpring to study the results. "As a scientist, I wanted to understand everything I was doing," she says. "But this kind of analysis didn't allow that: I just pressed buttons and got answers." And as Royo's studies comparing genetic activity on different chromosomes became more involved, she realized that the commercial tool could not keep up with her data-processing demands.

With the results of her first genomic sequencing experiments in hand at the start of a new postdoc, Royo had a choice: pass the sequences over to the experts or learn to analyse the data herself. She took the plunge, and began learning how to parse data in the free, open-source software package R. It helped that the centre she had joined — the Friedrich Miescher Institute for Biomedical Research in Basel, Switzerland — ran regular courses on the software. But she was also following a wider trend: for many academics seeking to wean themselves off commercial software, R is the data-analysis tool of choice.

Besides being free, R is popular partly because it presents different faces to different users. It is, first and foremost, a programming language — requiring input through a command line, which may seem forbidding to non-coders. But beginners can surf over the complexities and call up preset software packages, which come ready-made with commands for statistical analysis and data visualization. These packages create a welcoming middle ground between the comfort of commercial 'black-box' solutions and the expert world of code. "R made it very easy," says Rojo. "It did everything for me."

That, indeed, is what R's developers ▶

intended when they designed it in the 1990s. Ross Ihaka and Robert Gentleman, statisticians at the University of Auckland in New Zealand, had an interest in computing but lacked practical software for their needs. So they developed a programming language with which they could perform data analysis themselves. R got its name in part from its developers' initials, although it was also a reference to the most widely used coding language at the time, S.

In the early days of the World Wide Web, R quickly attracted interest from scientists around the globe who needed statistical software and were willing to contribute ideas. Gentleman and Ihaka decided to make their source code accessible to everybody, and coding-literate scientists quickly developed packages of pre-programmed routines and commands for particular fields. "I can write software that would be good for somebody doing astronomy," says Gentleman, "but it's a lot better if someone doing astronomy writes software for other people doing astronomy."

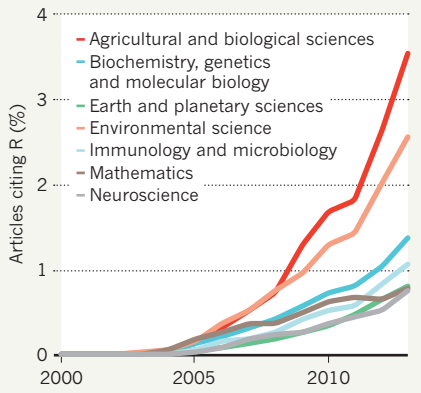
MATHEMATICAL SOLUTIONS

Karline Soetaert, an oceanographer at the Royal Netherlands Institute for Sea Research in Yerseke, took up that idea when, in 2008, she wanted to check the health of zooplankton in the estuary of the river Scheldt. Soetaert wanted to calculate how fast zooplankton were dying, using measurements along the river, but R was not equipped for that. To tackle the problem, she worked with two ecologists to develop deSolve — the first package written in R to solve differential equations. "Other software can do that, but it is expensive and closed source," she notes. Now deSolve is used by epidemiologists modelling infectious diseases, geneticists working on gene-regulatory networks and drug developers working on pharmacokinetics (how compounds behave in living organisms).

By 2003, 10 years after R's first release, scientists had developed more than 200 packages, and the first citations of the 'R Project' appeared. Today, nearly 6,000 packages exist for all kinds of specialized purposes. They allow scientists to compare a human and a Neanderthal genome (using Bioconductor: go.nature.com/s7mq39); to model population growth (IPMpack: go.nature.com/cyhons); predict equity prices (quantmod: go.nature.com/jxqasm); and visualize the results in polished graphics (ggplot2: ggplot2.org) in a few lines of code. Experts can use R to write up manuscripts, embedding raw code in them to be run by the reader (knitr: http://yihui.name/knitr). Nearly 1 in 100 scholarly articles indexed in Elsevier's Scopus database last year cites R or one of its packages — and in agricultural and environmental sciences, the share is even higher (see 'A rising tide of R').

A RISING TIDE OF R

An increasing proportion of research articles explicitly reference R or an R package.



SYLVIA TIPPMANN/SOURCE: ELSEVIER SCOPUS DATABASE

STATISTICAL SUCCESS

For many users, R's quality as statistics software stands out. The tool is on a par with commercial packages such as SPSS and SAS, says Robert Muenchen, a statistician at the University of Tennessee in Knoxville who analyses the popularity of software used in statistical computing. In the past decade, R has caught up with and overtaken the market leaders. "Most likely, R became the top statistics package used during the summer of this year," he says.

In genomics and molecular biology, a software project called Bioconductor was developed on the back of R. It helps scientists to process and compare huge numbers of genetic sequences, to query results against databases such as Gene Expression Omnibus and to upload data to the databases . It includes almost 1,000 packages, some of which help to link the millions of DNA snippets from next-generation sequencing experiments to annotated genes.

For her dive into R, Royo had intensive training: under the supervision of Michael Stadler, head of the Friedrich Miescher Institute's bioinformatics group, she took about half a year to work on R and Bioconductor. But there are plentiful chances to learn, says Karthik Ram, an ecologist at the Berkeley Institute for Data Science in California who founded rOpenSci, an initiative that helps scientists to adopt and develop R (see 'An R starter kit'). He and his colleagues teach free courses that do not require existing programming skills and are targeted towards scientists' specific problems.

One researcher who took that training is Megan Jennings, an ecologist at San Diego State University in California. She tracks bobcats, mountain lions and other wild animals, to understand their movements. Armed with more than 400,000 time-stamped photos to which she had appended species names — taken from 36 cameras running for almost a year — Jennings wanted to follow particular species at particular times of year. At first, she manually selected the photos she wanted and fed them into a black-box program called PRESENCE. But with Ram's help, she is creating an R package that reads in the tagged photos, cleans them up and then sends customized subsets of the data to a pre-existing modelling package in R. "What took me one hour to do manually, I will now be able to do in five minutes," Jennings says.

One of the greatest perks of R is its online support. Discussion forums about R-related topics outstrip online questions about any commercial statistics software says Muenchen.

"It's common to see someone post a question and the person who developed the package answer within half an hour," he says. This rapid response is key for scientists in basic research. "I can find an answer to almost any question online," says Royo. She can confidently do most of her day-to-day data analysis herself, and she helps out less proficient colleagues. Still, "I google things every day", she adds. Learning R, says Royo, has not only taught her coding skills, but has also made her more critical about other scientists' analyses.

Not every scientist is enthusiastic about learning the necessary programming — even though, says Ram, R is less intimidating than languages such as Python (let alone Perl or C). "There are going to be far more scientists that will be comfortable with click-and-drop interfaces than will ever learn to program at any time," Muenchen says. Geneticist Rabih Murr, for example, took the same R course as Royo when he was a postdoc, but preparing a paper for publication gave him little time to practise. To get started and develop research-specific skills in R definitely requires a commitment: "It's a matter of priorities," he says. But after becoming a lab head at the University of Geneva in Switzerland this year, he is planning to hire someone with R experience.

Like any other skill, learning R cannot be done overnight. But Jennings says that it is worth it. "Make that time. Set it aside as an investment: for saving time later, and for building skills that can be used across multiple problems we face as scientists." ■

TUTORIALS

An R starter kit

- Install R at the Comprehensive R Archive Network: http://cran.r-project.org. This also provides an introduction to the system: go.nature.com/jh9jb8.
- Many researchers recommend using a (free) powerful interface called RStudio: www.rstudio.com
- Among many online tutorials are those provided by DataCamp (go.nature.com/qndp6w), rOpenSci (ropensci.org), Software Carpentry (go.nature.com/wg3s9u) and R-bloggers (www.r-bloggers.com).
- For a sample list of R packages in different sciences, see the online version of this article at go.nature.com/zrhdkj.

NATURE.COM
For more on scientific software, apps and online tools, visit:
nature.com/toolbox