Conferência de inventários florestais em campo

Jeffrey M. Perkel, Technology Editor, Nature

06 mar 2022 19:56:49 -03

A mock computed manuscript created in RStudio using {Rmarkdown}. The {Bookdown} and {Rticles} packages were used to output the text in Springer Nature’s desired manuscript format.

# 1 Introdução

“Literate programming” is a style of programming that uses computational notebooks to weave together code, explanatory text, data and results into a single document, enhancing scientific communication and computational reproducibility.1–3 (These references were added into the document using RStudio’s integration with the open-source Zotero reference manager4 plus the [Better BibTeX](https://retorque.re/zotero-better-bibtex/) Zotero plugin.)

Several platforms for creating such documents exist.5 Typically, these documents interleave code and text ‘blocks’ to build a computational narrative. But some, including [R Markdown](https://rmarkdown.rstudio.com/), [Observable](https://www.observablehq.com), and the [Jupyter Book](https://jupyterbook.org/intro.html) extension to the Jupyter ecosystem, also allow authors to include and execute code “inline” – that is, within the text itself.

This makes it possible to create fully executable manuscripts in which the document itself computes and inserts values and figures into the text rather than requiring authors to input them manually. This is in many ways the ‘killer feature’ of computed manuscripts: it circumvents the possibility that the author will enter an incorrect number, or forget to update a figure or value should new data arise. Among other uses, that allows authors to automatically time-stamp their documents, or insert the current version number of the software they use into their methods. For instance, this document was built at **06 mar 2022 19:56:51 -03** and calls the following R packages: {tidyverse} ver. **1.3.1**, {ggbeeswarm} ver. **0.6.0** and {bookdown} ver. **0.24**.

In this manuscript, created in RStudio using the R Markdown language, we will demonstrate a more practical example. (An Observable version is [also available](https://observablehq.com/@jperkel/example-executable-observable-notebook).)

# 2 Material e métodos

# 3 Resultados

## 3.1 Inline computation

Imagine we are analyzing data from a clinical trial. We have grouped subjects in three bins and measured the concentration of some metabolite. (These data are simulated.)

Rather than analyzing those data and then copying the results into our manuscript, we can use the programming language R to do that in the manuscript itself. Simply enclose the code inside backticks, with the letter r. For instance, we could calculate the circumference and area of a circle 10:

# 4 Resultados

You could write “A = `r pi \* r^2` and C = `r 2 \* pi \* r`”. Plugging in the radius *r* = **10**, that evaluates to “A = **314.16** and C = **62.83**”.

# References

1. Shen, H. Interactive notebooks: Sharing the code. *Nature* **515**, 151–152 (2014).

2. Perkel, J. M. A toolkit for data transparency takes shape. *Nature* **560**, 513–515 (2018).

3. Perkel, J. M. Why Jupyter is data scientists’ computational notebook of choice. *Nature* **563**, 145–146 (2018).

4. Perkel, J. M. Streamline your writing and collaborations with these reference managers. *Nature* **585**, 149–150 (2020).

5. Perkel, J. M. Reactive, reproducible, collaborative: computational notebooks evolve. *Nature* **593**, 156–157 (2021).