Tooling and guidance for translations of Markdown-based R content (Quarto, R Markdown)

Maëlle Salmon

2022-09-29

Signatories

Project team

- Maëlle Salmon, research software engineer at rOpenSci, previously in charge of three ISC funded projects (Catalyzing R-hub adoption through R package developer advocacy, HTTP testing in R, R-Ladies organizational guidance) funded by the ISC, R blogger, volunteer editor for rOpenSci's system of package peer-review, main manager of R-Ladies Global Twitter account.

Contributors

- Paola Corrales, Ph.D. student at the University of Buenos Aires. She is a Trainer and Instructor for The Carpentries and an RStudio certified instructor. She co-founded MetaDocencia, a Spanish-speaking community to help teachers and educator to apply science-based techniques to their clase. She participated in the translation to Spanish of several Carpentries lessons, the book R for Data Science, several RStudio Cheat Sheets and the book Teaching Tech Together. She is also working on the translation of rOpenSci materials.
- Elio Campitelli, Ph.D. student at the University of Buenos Aires in atmospheric sciences and an R package developer. He is a founding member of the R User Group in Buenos Aires and MetaDocencia. He maintains several open-source R packages (e.g., ggnewscale; metR) and contributes to other packages, such as data.table and ggplot2. He contributes to the translation of the book R for Data Science to Spanish, and is working on the translation of rOpenSci materials.
- Yanina Bellini Saibene, Community Manager at rOpenSci, R-Ladies Project Lead, Member of The Carpentries Executive Council and RForwards Core Team. Co-founder of MetaDocencia, LatinR, and R-Ladies Santa Rosa. She lead the collaborative translation of Teaching Teach Together to Spanish and MetaDocencia educational materials to Portuguese. She was involved in the translation to Spanish of R for Data Science, R-Ladies's Rules and Guidelines, some lessons by The Carpentries and severals RStudio Cheat Sheets. She is leading the Multilingual Publishing project at rOpenSci.

Consulted

Problem

Open Source and Open Science are global movements, but most of their material and resources are published in English. Non-English speakers face a significant barrier to being part of these movements. The R community is no stranger to this reality. Publishing multilingual resources can lower the barrier to access to knowledge, help democratize access to quality resources and increase the possibilities of contributing to software and open science projects. There is interest in translating Markdown-based R content (see for instance this thread about multilingual Quarto websites: https://github.com/quarto-dev/quarto-cli/issues/275 ; and the effort of the Latin American R Community to translate books, like R for Data Science (https://es.r4ds.hadley.nz/) and Teaching Tech Together (http://teachtogether.tech/es/index.html) to Spanish, the package datos (https://github.com/cienciadedatos/datos) to Spanish and Portuguese, and several translations to Spanish of RStudio CheatSheets, The Carpentries' and The Programming Historian lessons), but little specific R tooling and guidance as of now.

Tooling and guidance for translations of Markdown-based R content (Quarto, R Markdown)

Overview

rOpenSci is working on the translation of their technical material to other languages than English, starting with Spanish.

rOpenSci materials are technical documents (rendered using software); their translations present unique challenges and opportunities. Translations of technical documents have two aspects (a) the internationalization (i18n) that provides the framework to support translation and requires technical knowledge (need to deal with source code vs. rendered version of the material), and (b) the localization (l10n), which is the task of translating the text and requires linguistic knowledge.

At the same time, technical translations of living documents have two well-defined stages involving different resources: (a) achieving a first version of the translated material and (b) keeping the material updated and synchronized between the different languages.

This proposal focuses on the achievement of a first version of translated material, both technically (tooling to create an automatically translated document) and linguistically (glossary).

rOpenSci now has started experimenting with automatic translation of Markdown-based content as a way to support the work of human translators. We aim to share our workflow with others via the creation of an R package including extensive documentation.

Detail

Our content lives in Quarto books and in a Hugo (blogdown) website. We have developed scripts such that for a target piece of content, we can

- extract the text, transform Markdown to XML while protecting special tags (code) and divs (curly braces for R Markdown or Quarto options),
- send XML to DeepL translation API,
- transform the output to Markdown,
- tweak YAML metadata,
- create a new file,
- push the new content into a pull request.

Human translators can then use the pull request as a starting point for their translation, saving much precious time and focus on linguistic tasks such as the use of inclusive language, look for references in the language of the translation, the location of examples, and the correct translation of phrases, metaphors, or analogies. Our workflow is supported by tinkr for transforming Markdown to and from XML, xml2 for handling the XML, httr2 for API requests, gert and usethis to manage pull requests. (XML translation is not supported by the deeplr package at this point.)

While our current work in progress is well aided by our very specific scripts, we aim to create an R package. Its functionality would include: the full translation & file creation pipeline from "path to a file in a language" to "path to a file in another language" for Hugo or Quarto content; the sharing of a technical glossary for use with DeepL API. At first the pipeline would be specific to our content organization (Quarto "multilingual book" where the Spanish filenames are english-filename.es.qmd; Hugo website where the Spanish blog posts are in a different post bundle than the English blog post) but we would welcome use cases so that we can adapt it to other content organization. Its documentation would include a vignette with a suggestion of a workflow including automatic pull request creation. The package would be created following rOpenSci general guidelines as well as rOpenSci specific HTTP testing guidelines. We expect the work would also help strengthen these guidelines. We would publish the package on rOpenSci R-universe.

Project plan

- Package skeleton 1 day
- Function for translation of Markdown text and Markdown file where the user indicates the output file name 1 day
- Function for translation of Markdown file in a Hugo website 1 day

- Function for translation of Markdown file in a Quarto book 1 day
- Testing 1 day
- Vignette about workflow 1 day
- Further documentation 2 days
- Promotion via a blog post 2 days

Requirements

People

The project would be tackled by Maëlle Salmon.

Processes

The package is covered by rOpenSci Code of Conduct.

Tools & Tech

We shall continue using GitHub infrastructure, including GitHub Actions.

Funding

The main cost is for the writing and development work. This work is broken down into milestones, reflecting the project stages outlined in the Technical Delivery section. The hourly rate of $100, used in previous R Consortium sponsored projects such as R-hub, has been used to arrive at costs.

Item	Cost ($)
Milestone 1: Package with functions and tests	4,000
Milestone 2: Package documentation and promotion via a blog post	4,000
Total	**8,000**

Success

Definition of done

Success of the project will be to publish the package at r-universe and announcement of the package at author's and rOpenSci Twitter accounts. Announcement of the package official release on rOpenSci blog.

Measuring success

- Use of the package for translation of Markdown based R materials (mention of the package, contribution with updates to the technical glossary).

Future work

Include other languages to the technical glossary. Analyze how to incorporated into the materials update pipeline to keep the texts in different languages synchronized.

Key risks

The main risk is the package being rendered useless by newer tooling but our hope is that it will have served its role in the meantime.