2025-09-10

momer: Exposing the Python library momepy in R

Claudiu Forgaci
Department of Urbanism, Delft University of Technology

John Doe
Digital Competence Center, Delft University of Technology

Executive Summary

Signatories

r-spatial community members SDSL community Rbanism community members ?

Project team

Claudiu Forgaci is assistant professor of urban design and analytics engaged in urban morphology research. Claudiu is experienced in R package development (see rcrisp, rcoins, and visor), he is a member of the Spatial Data Science across Languages. He is also one of the initiators of the Rbanism community promoting reproducibility, automation and scalability in urbanism research using R. Claudiu will lead the development of `momer`.

John Doe is a research software engineer at the TU Delft Digital Competence Center … **Jane Doe** is a research software engineer at the TU Delft Digital Competence Center … John and Jane will carry out the software engineering work planned in this project with a shared 0.4FTE for the planned duration of the project.

Contributors

The creator of `momepy` **Martin Fleischmann** and members of the SDSL community **Robin Lovelace**, **???** and **???** provided input at the proposal stage.

Consulted

ISC member **Josiah Parry**, also member of the SDSL community, provided feedback on the proposal.

The Problem

Urban morphometrics is an emerging sub-field of urban morphology that enables scalable quantitative analysis of urban form. The field of urban morphometrics is inherently programmatic and has been heavily driven and promoted by the Python library `momepy` . While `momepy` has gained quite some prominence, its user base does not reach beyond Python. This is a problem because

By expanding the tools of morphometrics to the R community

Current R packages, `foot`, `vectormetrics`

An example in-text citation (Wickham 2016).

The proposal

Overview

The project aims to develop `momer`, an R package that wraps the Python library `momepy` using the `reticulate` package and sets up a synchronised documentation system.

`momer` will enable R users to perform urban morphometric analyses without needing to switch to Python, thereby broadening the reach of `momepy` and increasing the adoption of morphometric methodologies.

The development of `momer` is timely as `momepy` is approaching a major release (v1.0.0) which will stabilise its API, making it easier to maintain the R wrapper in the long term.

The project will be structured around four main milestones over a ???-month period, from initial setup and basic wrappers to final review and release.

Detail

Minimum Viable Product

The minimum viable product is a package that wraps all core functions of `momepy` and mirrors its documentation (without synchronising).

Architecture

`momer` will wrap `momepy` functions with `reticulate` and will be organised into the same nine modules, namely: "Managing morphological elements", "Measuring dimension", "Measuring shape", "Measuring spatial distribution", "Measuring intensity", "Measuring diversity", "Measuring connectivity", "Measuring streetscape", and "Data preprocessing".

`momer` will also be set up with a script on CI to automate the generation of documentation

Assumptions

One of the main assumptions is that, according to creator and maintainer of `momepy` Martin Fleischmann, with the upcoming major release, the `momepy` API will be kept stable, which will allow `momer` to be feasibly maintained.

We also assume that Python documentation can be

External dependencies

The project primarily depends on the wrapped Python library `momepy` and the R package `reticulate` used for wrapping.

Project plan

Start-up phase

To enable collaboration and contributions, we will set up the project on GitHub. The repository will include guidelines for contributors, a code of conduct, and a `README.md` file with an overview of the project.

A project team will be set up to track tasks and milestones, and issues will be used for feature requests and bug reports. Reporting will be done quarterly, with updates shared on the R Consortium blog and social media platforms to keep the community informed of progress.

We aim to release `momer` under the permissive Apache 2.0 license which is compatible with the BSD-3-Clause license of `momepy`. `momer` will only wrap `momepy`, i.e., call it at runtime, which means that no BSD-licensed code is redistributed and so mentioning the license in the documentation should suffice.

Technical delivery

The project will be structured by the following milestones:

1. Initial set-up and basic wrappers (Month 1-2):
   - Set up the GitHub repository with necessary documentation and contribution guidelines.
   - Implement basic wrappers for core `momepy` functions using `reticulate`.
   - Initial testing to ensure functionality.
2. Comprehensive wrappers and testing (Month 3-6):
   - Expand wrappers to cover all major functionalities of `momepy`.
   - Develop a suite of unit tests to ensure reliability and correctness.
   - Begin drafting detailed documentation for each function.
3. Documentation and vignettes (Month 7-10):
   - Set up automatic synchronisation of `momepy` documentation with `momer` on CI
   - Finalize comprehensive documentation, including examples and use cases.
   - Create vignettes demonstrating typical workflows and applications of `momer`.
   - Conduct user testing and gather feedback for improvements.
4. Final review and release (Month 11-12):
   - Address any issues or feedback from user testing.
   - Perform a final review of the code base and documentation.
   - Prepare for the official release of `momer` on CRAN and GitHub.
   - Publicize the release through blog posts, social media, and relevant forums.

Other aspects

The project will be open-source, hosted on GitHub, and licensed under the Apache 2.0 license. This will enable broad use and contributions from the community.

Updates on progress and milestones achieved will be shared on the R Consortium blog and on the Rbanism community's blog on a quarterly basis. Updates will also be shared on the social media platforms Mastodon (Fosstodon), LinkedIn and Bluesky. To promote the wide adoption of

momer, we will announce its release at the UseR! conference and other similar events where the R spatial community is present. We also aim to introduce the package in the International Seminar of Urban Form (ISUF) where most of the research community conducting urban morphology analysis is present.

Budget & funding plan

TODO: I think this is simply the division of the estimated labor cost per milestone. Labor cost to be checked with the DCC.

The requested budget will be used for labor costs divided by period, as follows:

1. Initial set-up and basic wrappers (Month 1-2): €?
2. Comprehensive wrappers and testing (Month 3-6): €?
3. Documentation and vignettes (Month 7-10): €?
4. Final review and release (Month 11-12): €?

Success

Definition of done

A successful project will deliver a functional and well-dcumented momer package. The package will provide R users with access to the functionalities of the Python library momepy, enabling them to perform urban morphometric analyses within R.

Measuring success

Success can be measured through several key indicators:

- Timely completion of all project milestones;
- Positive feedback from user testing and community engagement;
- Adoption of the momer package by the R community, as seen in the number of downloads, citations, and contributions;
- Successful integration and functionality of all wrapped momepy features.
- Synchronised documentation between momepy and momer triggered at every new momepy release;
- Synchronised, comprehensive and clear documentation, including vignettes and examples, aligned with CRAN policies;
- Active post-release maintenance and updates, reflecting ongoing community needs and developments in the field of urban morphometrics.

Future work

momer is expression of the sustained interest of the Spatial Data Science across Languages (SDSL) community in cross-language projects. It is part of the momex initiative aiming to build a cross-language infrastructure for morphometrics. In the long term, we aim to gradually develop low-level infrastructure with bindings to multiple languages, including R. Carried out in an incremental way and targeting the most challenging use cases, such an infrastructure will lead to higher performance

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.