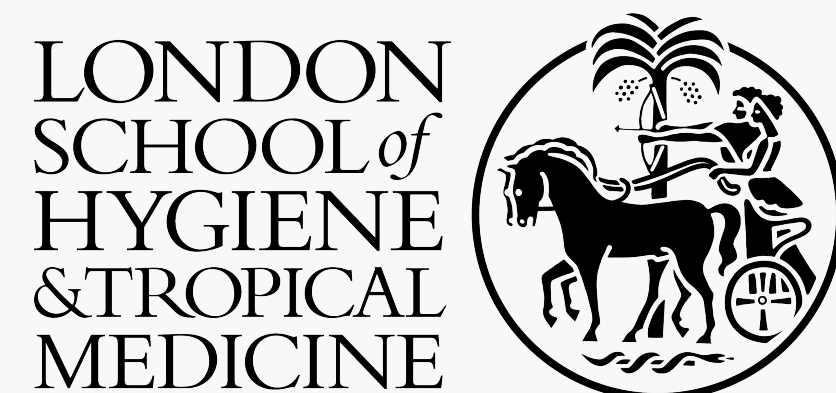


Superseding R packages – lessons learned from transitioning to *{epichains}* from *{bpmmodels}*

James M. Azam

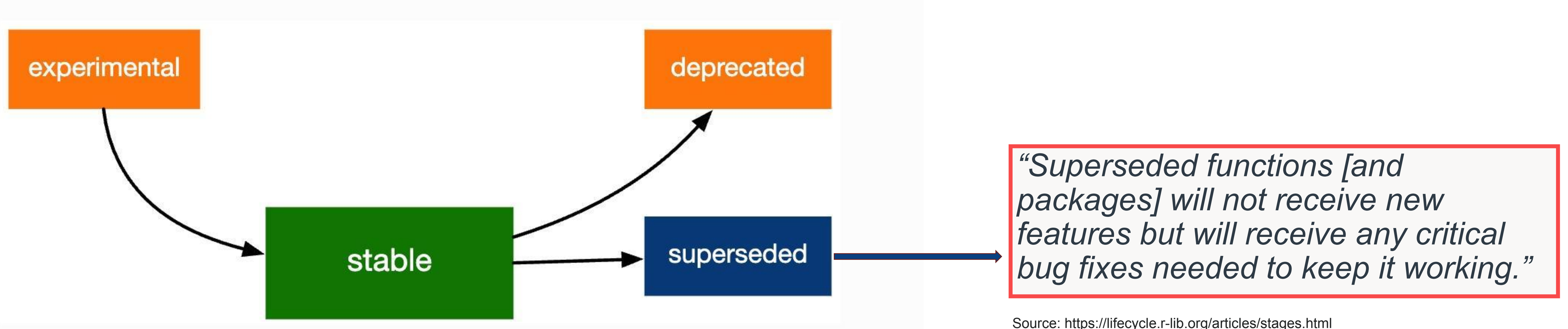
Research Software Engineer, Epiverse Initiative (London School of Hygiene and Tropical Medicine)

Presentation at RSLondonSouthEast 2023



Introduction

R package lifecycles



Source: <https://lifecycle.r-lib.org/articles/stages.html>

Superseding R packages (reasons)

- *Sometimes not exactly clear why but...*
- Re-imagination | reboot:
 - New infrastructure -> Name change:
 - `{ggplot}` -> `{ggplot2}`
- Scope changes:
 - `{plyr}` -> `{dplyr}` for data.frames OR `{purrr}` for lists
 - `{reshape}` -> `{reshape2}` -> `{tidyr}`
- Function & NAMESPACE redesign:
 - Many examples in `{tidyverse}`, etc.
- Interoperability

Lessons learned so far

*bpm*models: Methods for analysing the size and length of transmission chains from branching process models

R v0.3.0 R-CMD-check passing codecov 94% contributors 6 License MIT

*bpm*models is an R package to simulate and analyse the size and length of branching processes with a given offspring distribution. These models are often used in infectious disease epidemiology, where the chains represent chains of transmission, and the offspring distribution represents the distribution of secondary infections caused by an infected individual.

*epi*chains: Methods for analysing the size and length of transmission chains from branching process models

R v0.2.1 R-CMD-check passing codecov unknown contributors 5 License MIT lifecycle experimental

*epi*chains is an R package to simulate, analyse, and visualize the size and length of branching processes with a given offspring distribution. These models are often used in infectious disease epidemiology, where the chains represent chains of transmission, and the offspring distribution represents the distribution of secondary infections caused by an infected individual.



Lesson 1: Be a reproducibility champion

- **Consider**: Keeping old analysis scripts running (reproducibility)
- **Implications**:
 - Maintenance overhead:
 - Maintain old package (exist forever?)
 - Create new package

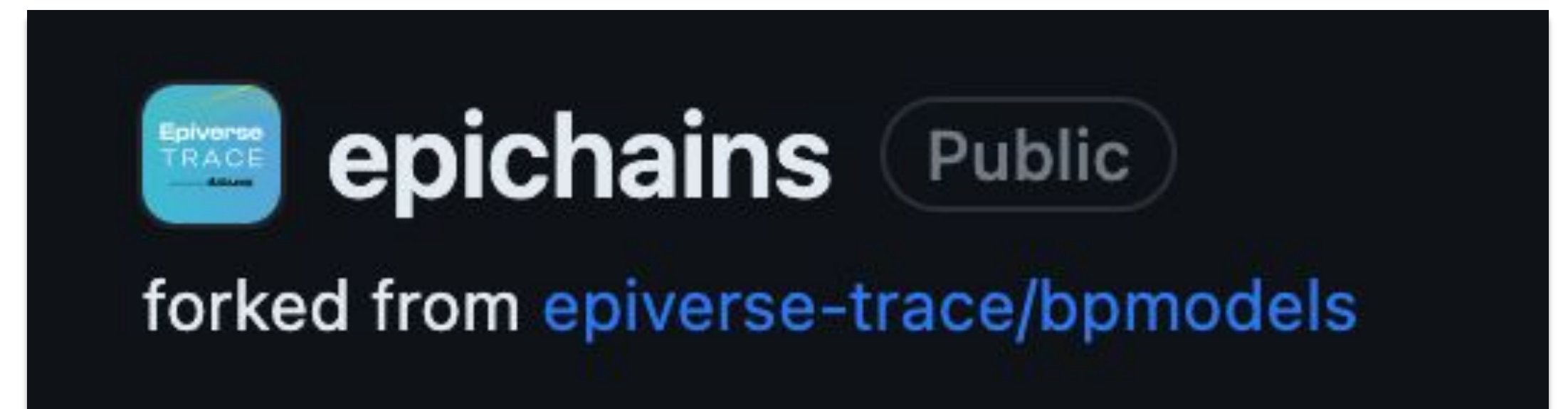
Implementation

All analysis was carried out using [R](#) version 3.6.2 ⁸. The branching process model was implemented using the `bpmodels 0.1.0` package ⁹. The analysis is available as an open-source R package ¹⁰. A dockerfile has been made available with the code to ensure reproducibility ¹¹.

crosses 1,000 and 10,000 reported cases, and then evaluate the 50% and 95% quantiles of these dates to determine the forecast interval. The model was built in the R statistical programming language, using the `bpmodels` package [13], and the `data2019nCoV` package for the SITREP data [14]. All analysis code is available at <https://github.com/SACEMA/COVID10k>.

Lesson 2. Acknowledge contributions

- **Consider**: Preserving contributions (commits, issues, pull requests, etc):
- **Implications**:
 - Choose the right Git workflow:
 - **Fork (and detach)**
 - Original codebase as initial commit
 - Squash-commits no-fork



Lesson 3. Be transparent with the user community

- **Consider**: Informing the users of development updates
- **Implications**:
 - Add Lifecycle section in README
 - Add a retirement explanation vignette
 - Print console messages

epichains re-implements bpmmodels by providing dedicated classes that allow easy manipulation and interoperability with other existing packages for handling transmission chain and contact-tracing data.

Other considerations

- Semantic versioning:
 - “Reset” to development version vs continue from version of superseded
- Duplicated effort:
 - Documentation: manuals/vignettes and user guides

Conclusion

- Superseding R packages is quite common:
 - reasons often unclear
- The decision can be costly, leading to many considerations:
 - Maintaining both the superseded and new package
 - Preserving contributions
 - Communication with users, and
 - Semantic versioning
- Audience: Any experiences and considerations to share?

References

1. R package lifecycles: <https://lifecycle.r-lib.org/>
2. `{bpmmodels}`: <https://github.com/epiverse-trace/bpmmodels>
3. `{epichains}`: <https://github.com/epiverse-trace/epichains>
4. `{reshape}`: <https://github.com/hadley/reshape>
5. `{plyr}`: <https://github.com/hadley/plyr>

Contact

James M. Azam

Twitter: @james_azam

Github: @jamesmbaazam

Epiverse-TRACE

Twitter: @epiverse_TRACE

Github: github.com/epiverse-trace