

Enhancing creativity in programming with R

Isabella Gollini



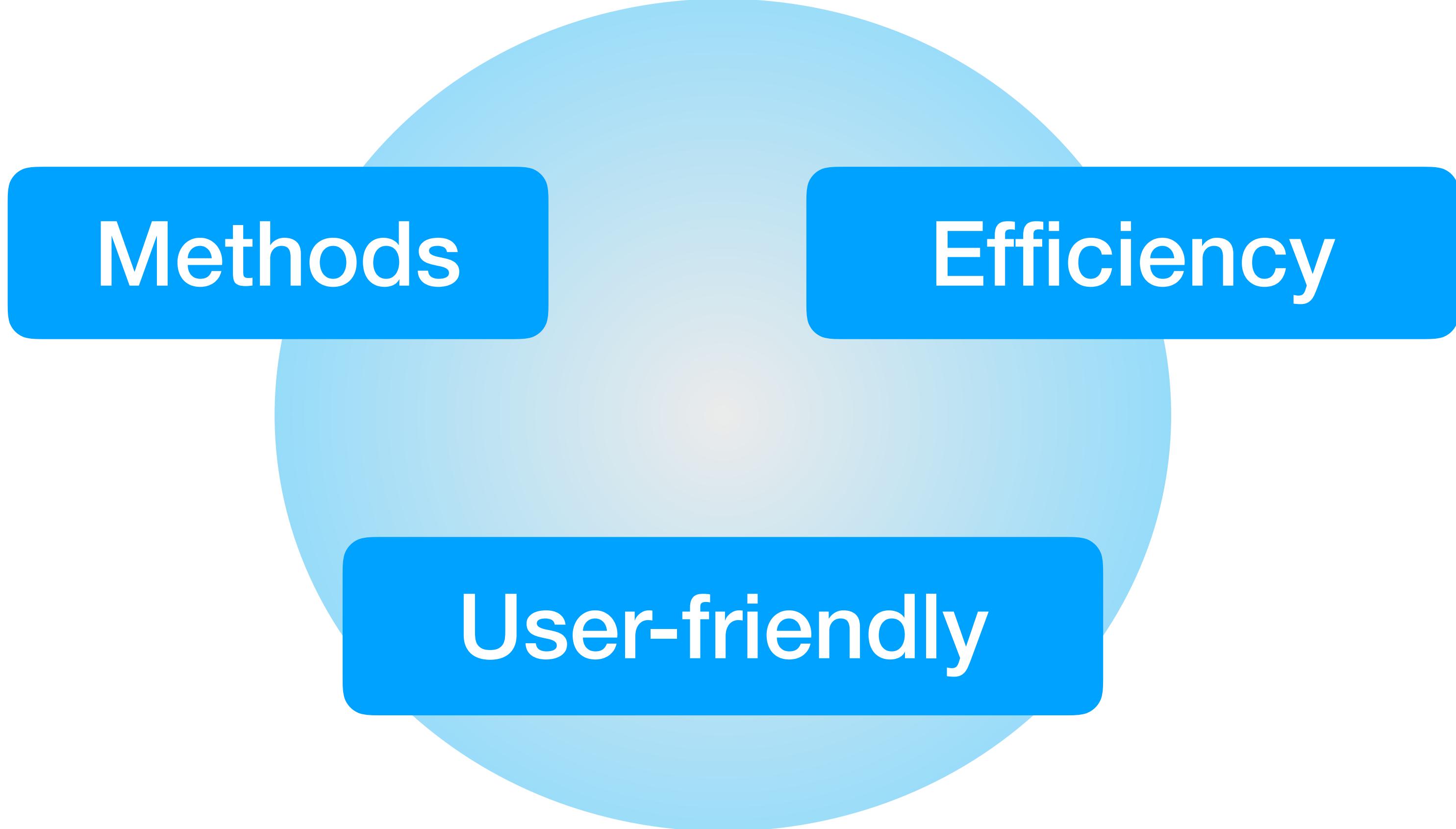
[@IsabellaGollini](https://twitter.com/IsabellaGollini)



isabella.gollini@ucd.ie



R programming



Methods

Efficiency

User-friendly

How can you improve your creativity?



Use different perspectives!

Different perspectives

Developer

Teacher

Learner

Collaborator

1. Developer

- Your programming style should be accessible to allow users to understand it and potential contribute by extending/improving your code

Syntax

Spacing

Place spaces around all infix operators (=, +, -, <-, etc.). The same rule applies when using = in function calls. Always put a space after a comma, and never before (just like in regular English).

```
# Good  
average <- mean(feet / 12 + inches, na.rm = TRUE)
```

```
# Bad  
average<-mean(feet/12+inches,na.rm=TRUE)
```

style.tidyverse.org

1. Developer

- Efficient and reliable code, well maintained and available on software development platforms e.g.,

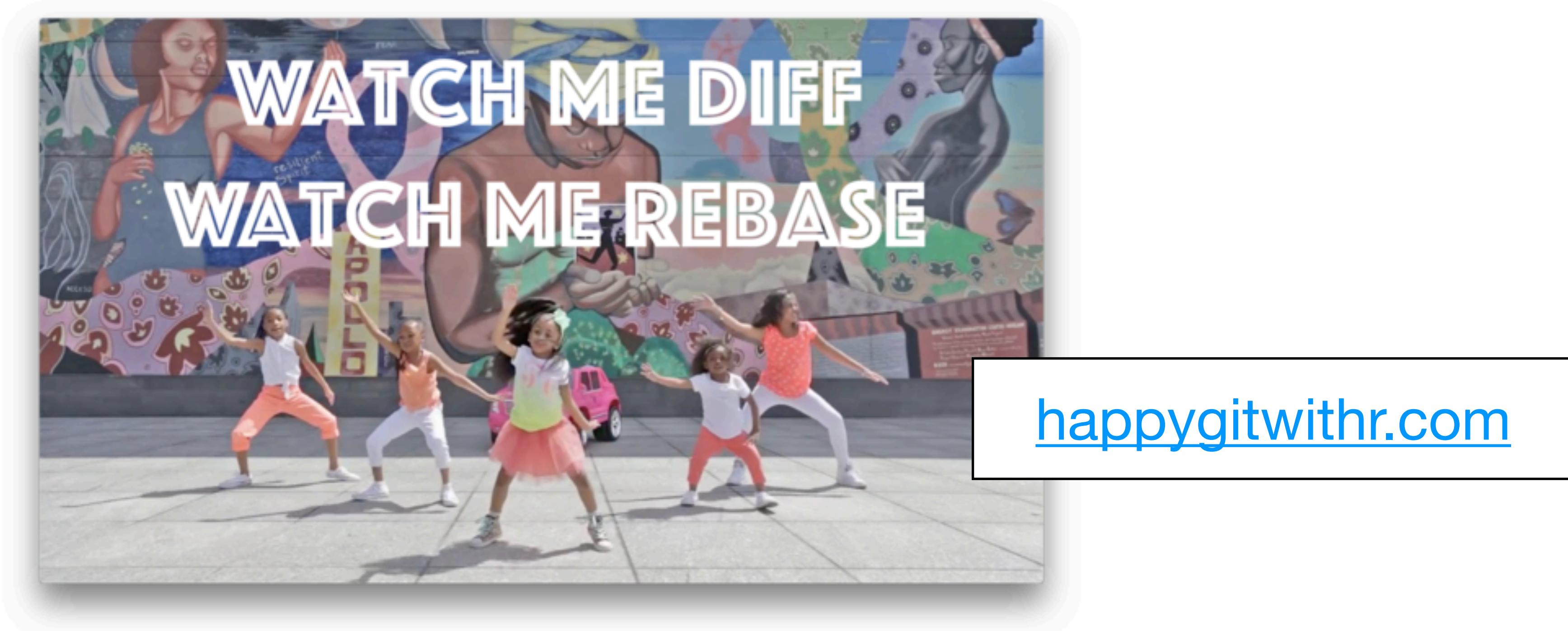
Let's Git started



GitLab

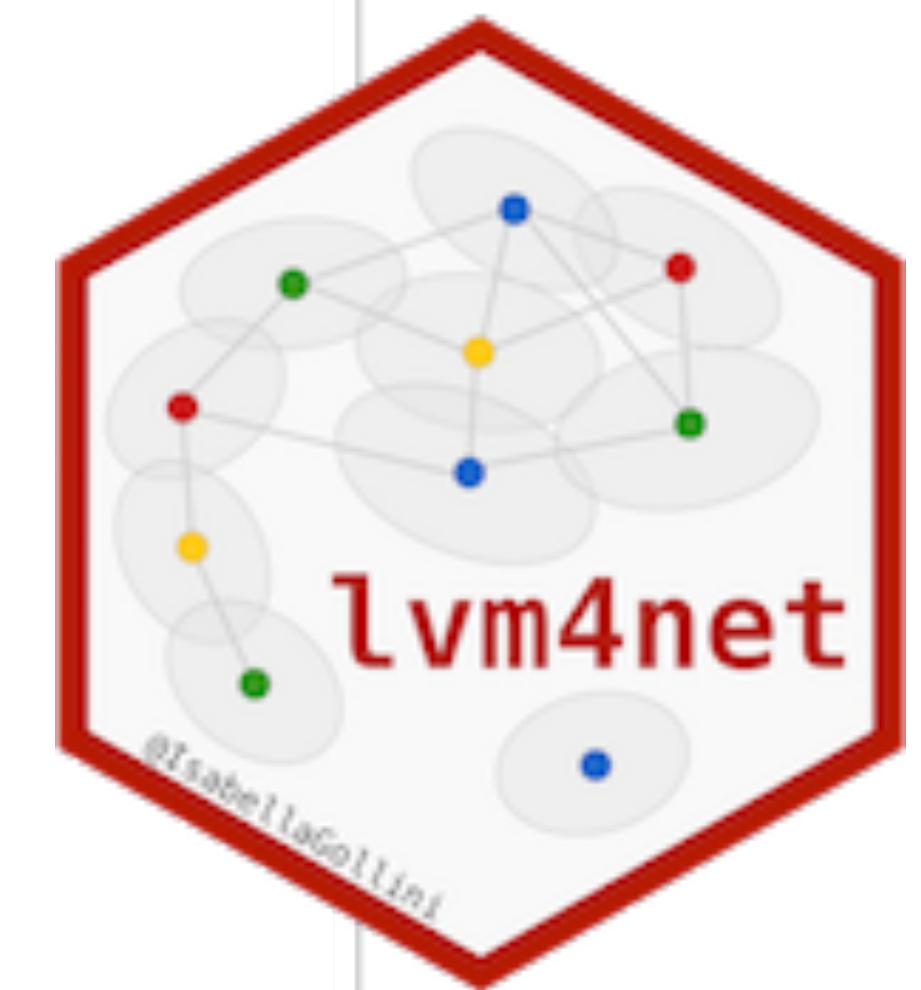
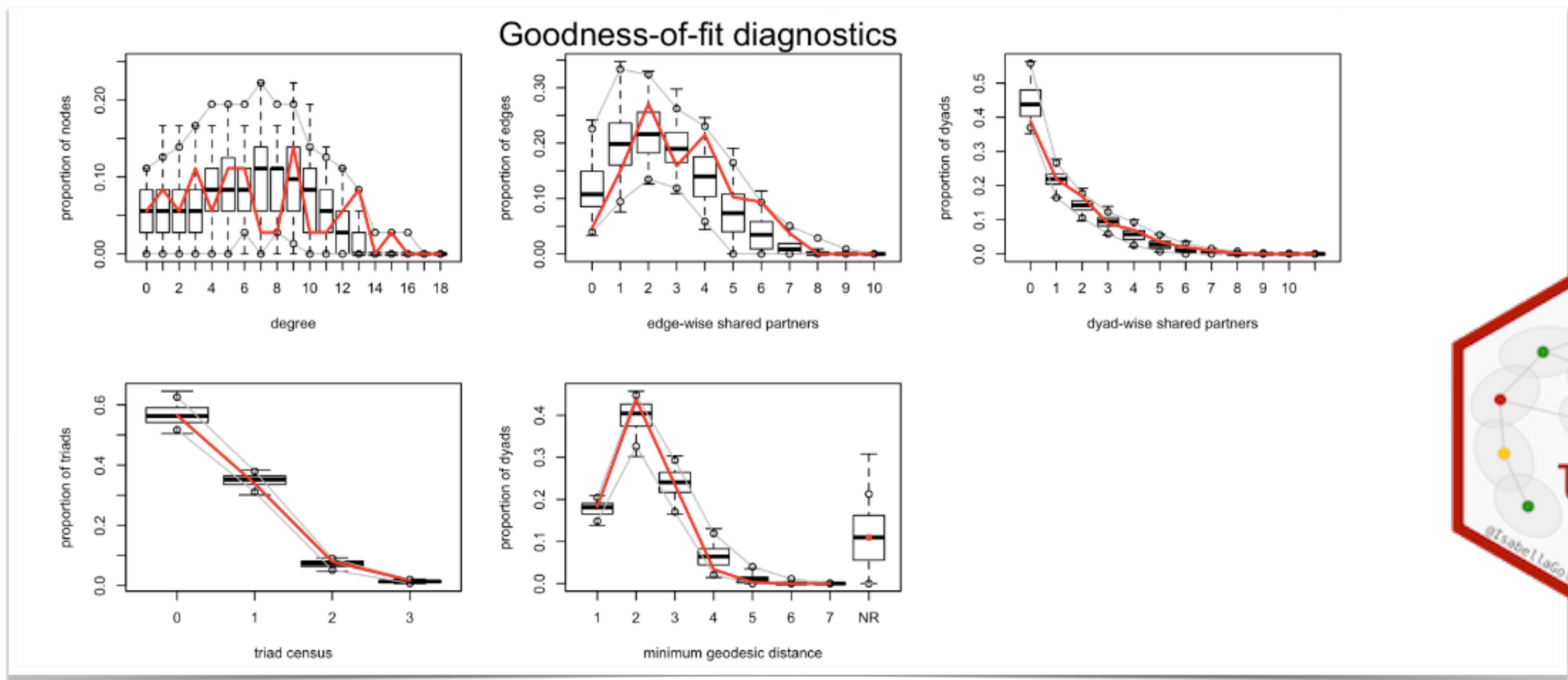


GitHub

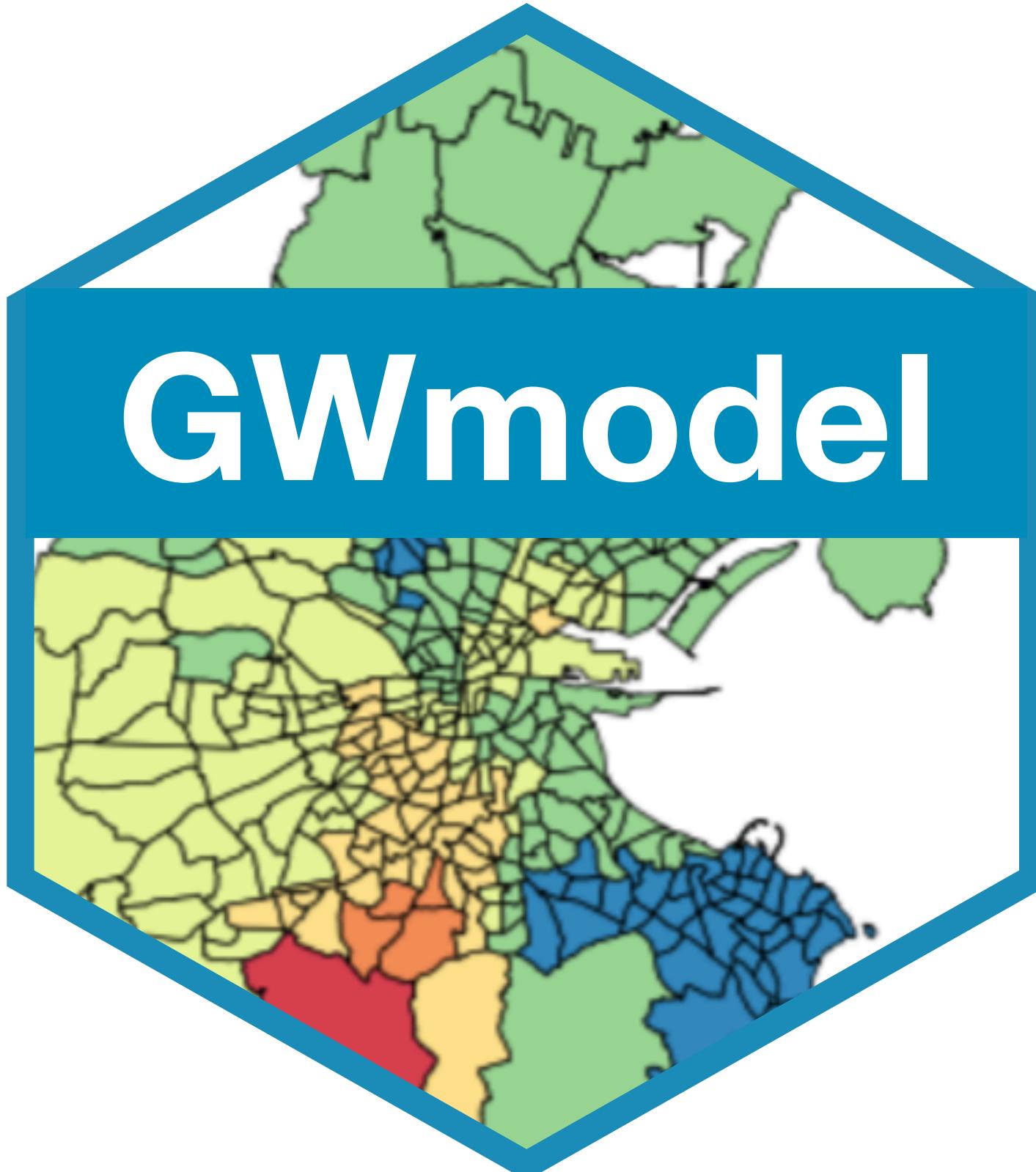


1. Developer

- Your code is under continual development and can be a good starting point for the implementation of more advanced methodologies

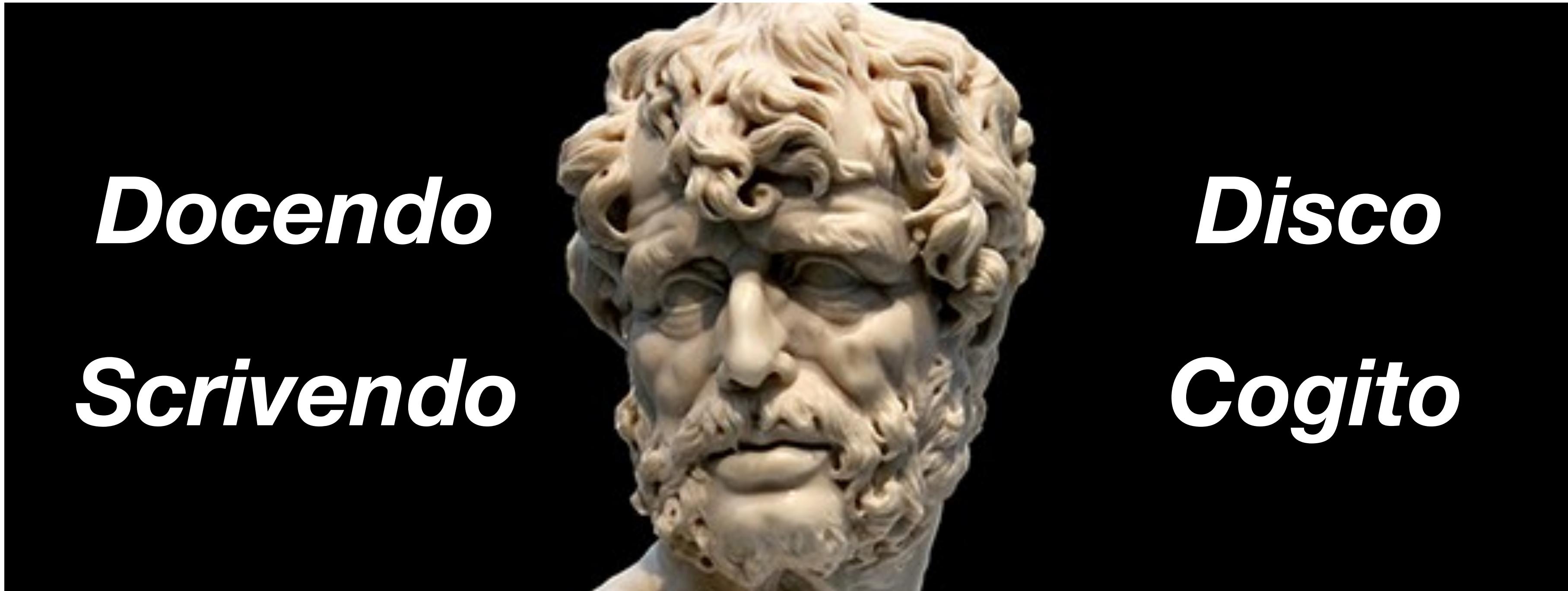


1. Developer



- How is your package positioned in the galaxy of methods and packages available?

2. Teacher



2. Teacher



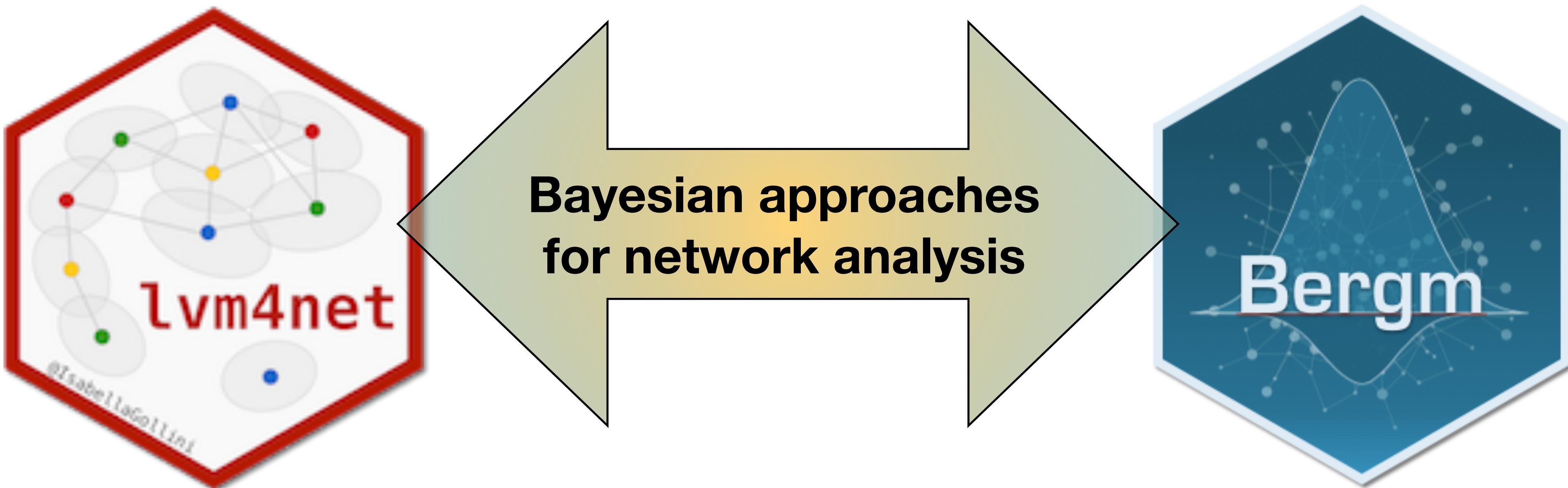
3. Learner

- Engagement & interaction with other practitioners is crucial



4. Collaborator

- Collaborate with other package developers in a synergic way as your package may complement other packages



4. Collaborator

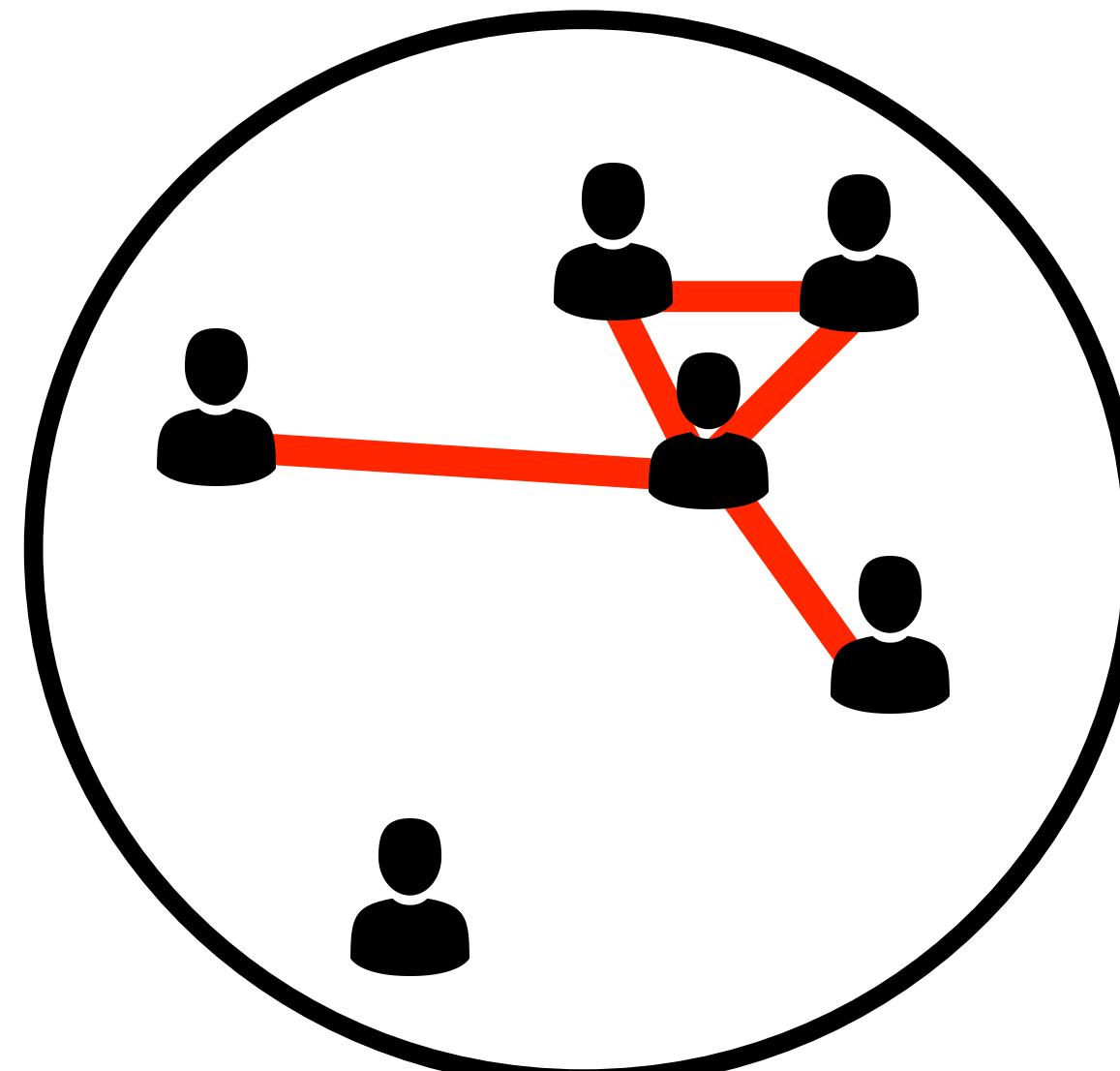
- Collaborate with people with different skills to improve the usage of the package

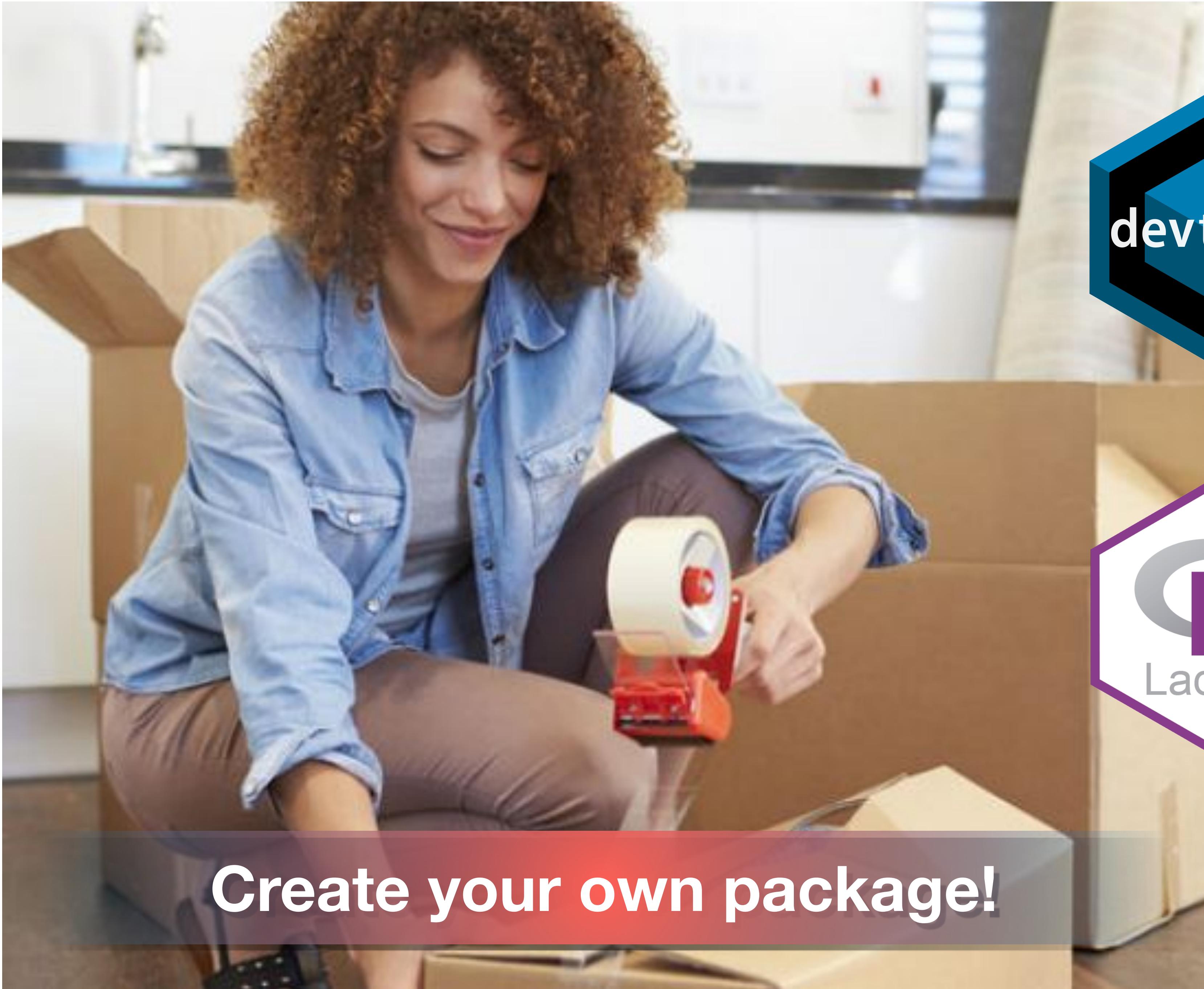


Paolo Campana



UNIVERSITY OF
CAMBRIDGE





Create your own package!



Think outside the box!