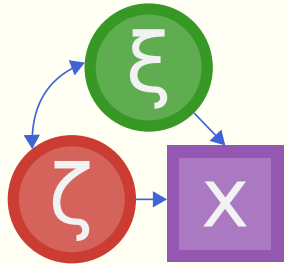
Extensible Software for Research

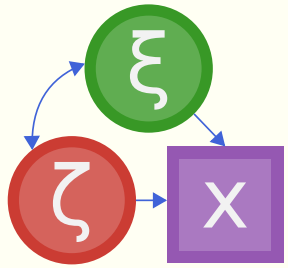


Maximilian Ernst

Aaron Peikert

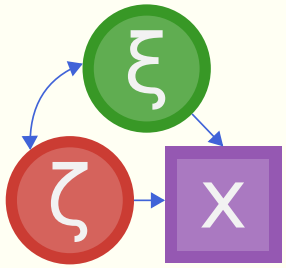
Contents

- Principles of Extensible Research Software
- Application: StructuralEquationModels.jl



A day in the life of …

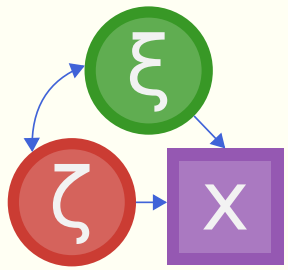
a statistical methods researcher



A day in the life of …

a statistical methods researcher

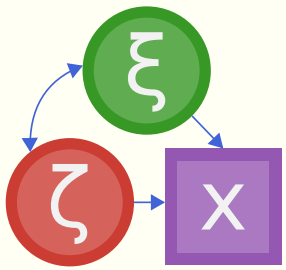
- work with a specific type of model
  - regression, structural equation models, deep learning, …



A day in the life of …

a statistical methods researcher

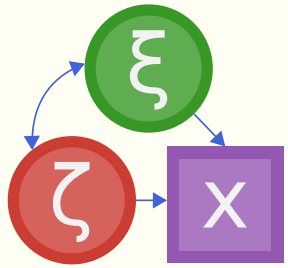
- work with a specific type of model
  - regression, structural equation models, deep learning, …
- have an idea



A day in the life of …

a statistical methods researcher

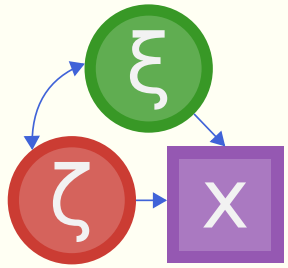
- work with a specific type of model
  - regression, structural equation models, deep learning, …
- have an idea
- test it



A day in the life of …

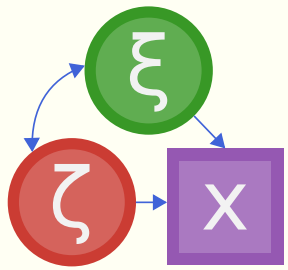
a statistical methods researcher

- work with a specific type of model
  - regression, structural equation models, deep learning, …
- have an idea
- test it
- make it available to applied researchers



We need to write Software

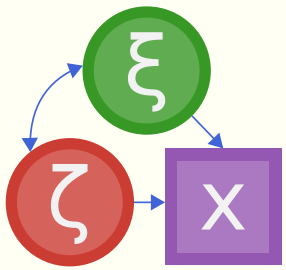
- to test $\rightarrow$ prototype
- to make it available $\rightarrow$ distribute



We need to write Software

- to test $\rightarrow$ prototype
- to make it available $\rightarrow$ distribute

What's the fastest way to get there?

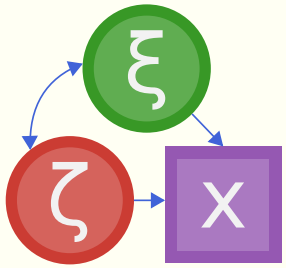


We need to write Software

- to test $\rightarrow$ prototype
- to make it available $\rightarrow$ distribute

What's the fastest way to get there?

We are already working with existing software.



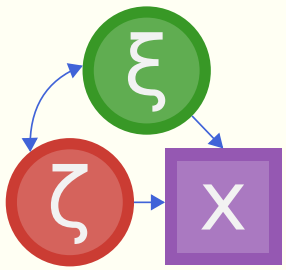
We need to write Software

- to test $\rightarrow$ prototype
- to make it available $\rightarrow$ distribute

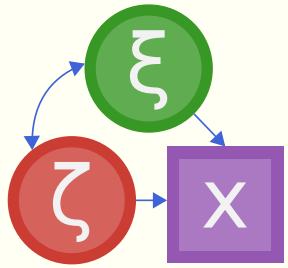
What's the fastest way to get there?

We are already working with existing software.

It would be nice if we could extend existing software!

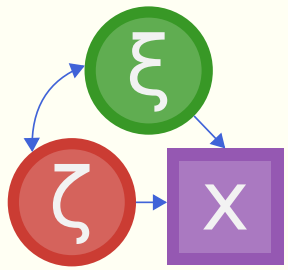


But …



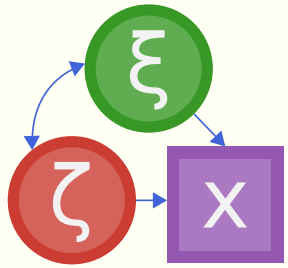
But …

- **understand** 1000s of lines of code



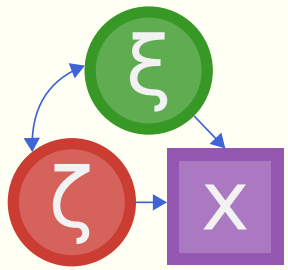
But …

- **understand** 1000s of lines of code
- make **changes**, possibly breaking stuff



But …

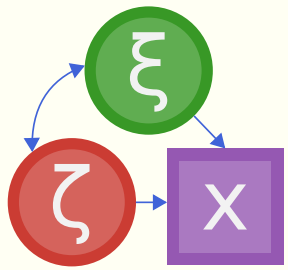
- **understand** 1000s of lines of code
- make **changes**, possibly breaking stuff
- get maintainers to **adopt** our changes



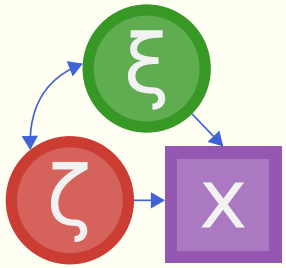
But …

- **understand** 1000s of lines of code
- make **changes**, possibly breaking stuff
- get maintainers to **adopt** our changes

These hurdles are often too high!

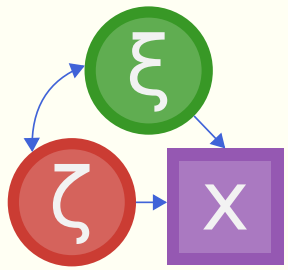


A day in the life of …



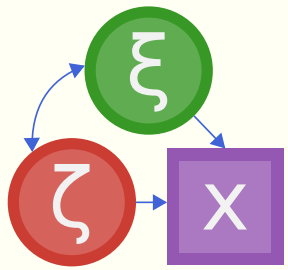
A year in the life of …

- to test: minimal reimplementation
  - waste of time
  - not well tested
  - hard to reproduce
  - slow



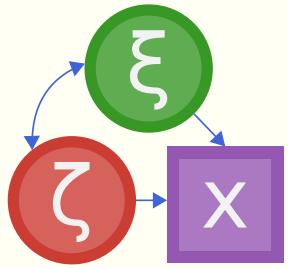
A year in the life of …

- to test: minimal reimplementation
  - waste of time
  - not well tested
  - hard to reproduce
  - slow
- to deploy: put code on github
  - bad user interface, no documentation
  - missing features
  - incompatible to existing software



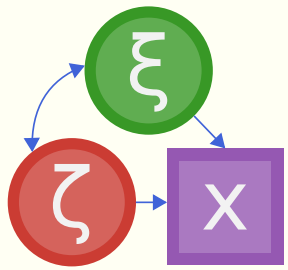
My Experience

- from R → julia



Culture

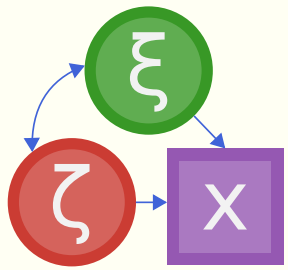
- care about extensibility
- developer documentation
- assume their code is read



An Example

Let’s make a thought experiment...

- encyclopedia
  - add an entry
  - some syntactical requirements
    300-400 words, alphabetical, …
- book
  - a draft already exists
  - add something everywhere it is applicable



An Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec interdum est, et suscipit elit. Aenean imperdiet augue sed arcu iaculis mollis. Aenean felis augue, fringilla ac diam non, dapibus commodo tellus. Donec laoreet a magna id vestibulum. Suspendisse sapien turpis, dictum sed scelerisque ac, malesuada ac augue. Integer id mattis ipsum. Fusce nec dui eu tellus elementum efficitur. Aenean iaculis lorem sem. Aenean eu placerat augue. Cras eget fermentum augue. Nullam in orci ut erat aliquet lacinia. Vivamus rhoncus, mauris vel pulvinar dapibus, tellus ante vestibulum lorem, sed tristique erat orci quis orci. Integer at laoreet neque, id lobortis turpis.

Quisque ultricies ultrices massa ut rhoncus. Sed finibus neque purus, sed ullamcorper lectus ultricies eu. Integer malesuada sem eget feugiat tincidunt. Aenean laoreet vulputate metus non iaculis. Nullam nec viverra purus, a elementum est. Mauris consequat nunc ut urna aliquam, a sollicitudin dui tincidunt.

Mauris et urna non justo faucibus cursus eu sit amet orci. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Proin at purus nisl. Quisque nec mi eget tellus vestibulum imperdiet. Praesent convallis dui urna, eu volutpat mi porta id. Cras euismod metus quam, a sollicitudin magna aliquet a. Ut placerat nunc ut leo viverra, ac bibendum est vehicula. Maecenas non finibus velit. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vivamus sit amet dolor lacinia, dignissim libero in, laoreet neque. Maecenas aliquet velit rhoncus, iaculis sem eget, eleifend velit.

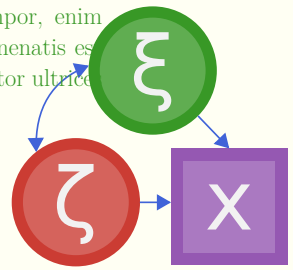
Vestibulum at sem felis. Suspendisse vitae euismod sapien, eget gravida massa. Nullam ac porttitor elit. Vestibulum in pharetra risus. Quisque condimentum porttitor massa, ut ornare turpis convallis sed. Nullam ut vestibulum sem. Aliquam eu risus. Sed commodo posuere ante. Nam vulputate sit amet mauris a aliquet. In tempor, enim ullamcorper auctor accumsan, odio felis sagittis erat, a venenatis est purus sed risus. Donec interdum dui at urna facilisis, porttitor ultrices lorem dapibus. Aenean scelerisque nisl at neque placerat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec interdum est, et suscipit elit. Aenean imperdiet augue sed arcu iaculis mollis. Aenean felis augue, fringilla ac diam non, dapibus commodo tellus. Donec laoreet a magna id vestibulum. Suspendisse sapien turpis, dictum sed scelerisque ac, malesuada ac augue. Integer id mattis ipsum. Fusce nec dui eu tellus elementum efficitur. Aenean iaculis lorem sem. Aenean eu placerat augue. Cras eget fermentum augue. Nullam in orci ut erat aliquet lacinia. Vivamus rhoncus, mauris vel pulvinar dapibus, tellus ante vestibulum lorem, sed tristique erat orci quis orci. Integer at laoreet neque, id lobortis turpis.

Quisque ultricies ultrices massa ut rhoncus. Sed finibus neque purus, sed ullamcorper lectus ultricies eu. Integer malesuada sem eget feugiat tincidunt. Aenean laoreet vulputate metus non iaculis. Nullam nec viverra purus, a elementum est. Mauris consequat nunc ut urna aliquam, a sollicitudin dui tincidunt.

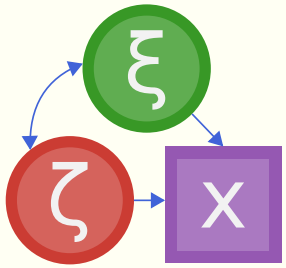
Mauris et urna non justo faucibus cursus eu sit amet orci. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Proin at purus nisl. Quisque nec mi eget tellus vestibulum imperdiet. Praesent convallis dui urna, eu volutpat mi porta id. Cras euismod metus quam, a sollicitudin magna aliquet a. Ut placerat nunc ut leo viverra, ac bibendum est vehicula. Maecenas non finibus velit. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vivamus sit amet dolor lacinia, dignissim libero in, laoreet neque. Maecenas aliquet velit rhoncus, iaculis sem eget, eleifend velit.

Vestibulum at sem felis. Suspendisse vitae euismod sapien, eget gravida massa. Nullam ac porttitor elit. Vestibulum in pharetra risus. Quisque condimentum porttitor massa, ut ornare turpis convallis sed. Nullam ut vestibulum sem. Aliquam eu risus. Sed commodo posuere ante. Nam vulputate sit amet mauris a aliquet. In tempor, enim ullamcorper auctor accumsan, odio felis sagittis erat, a venenatis est purus sed risus. Donec interdum dui at urna facilisis, porttitor ultrices lorem dapibus. Aenean scelerisque nisl at neque placerat.



Software Design

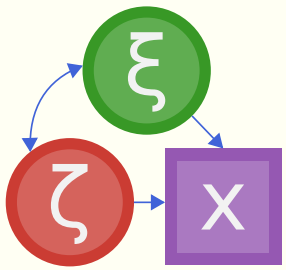
Not all research articles can be encyclopedias, but maybe all research software can be...



Software Design

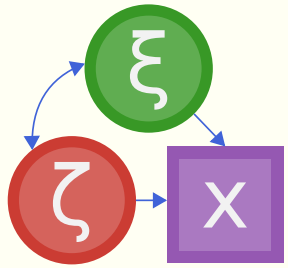
You need to be able to add new features...

- without **understanding** existing code
- without **changing** existing code
- syntactical requirements need to be **clear** and **easy communicable**!

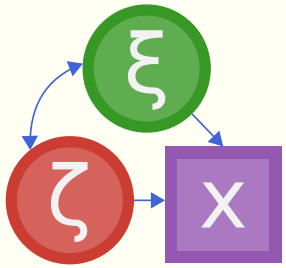


The Benefits

- Applied Researchers
  - user interface
  - better documentation
  - faster availability of new features
  - less bugs
  - higher performance
- Statistical Researchers
  - no re-implementation $\rightarrow$ less time-consuming
  - more users
  - no software engineering skills needed
- Maintainers
  - changes are easier to integrate

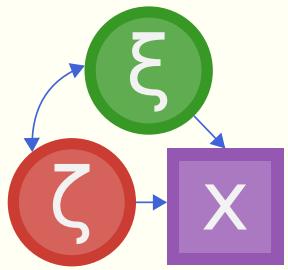


Less Abstract



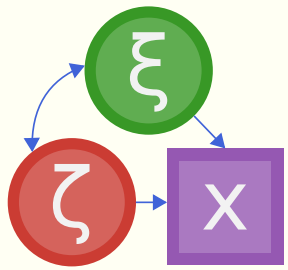
A few days in a methods researchers life

- Do you have any ideas why this does not converge?
- Staring puzzled at the theory (should work?!).
- Staring very puzzled at the implementation in C++.
- Rinse and repeat for a couple of days and researchers.



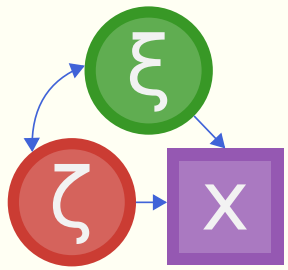
An hour in our life

- Look at the formula: $ridge(x, \lambda) = \lambda \sum_{j=1}^{p} x^2$
- Implement in Julia: `ridge(x, λ) = λ * sum(x.^2)`
- add 30 lines of API (formal requirements)
- Enjoy.



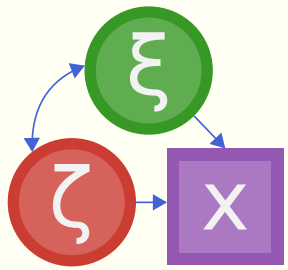
Two hours in our life

- Simulation in Juila works (converges as it should)



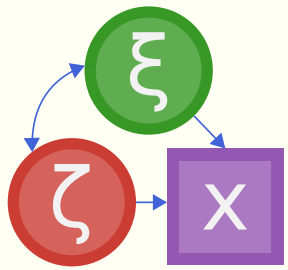
Two hours in our life

- Simulation in Juila works (converges as it should)
- Original simulation takes weeks on a dedicated workstation.
- Original simulation freezes our cluster due to poor parallelization.
- Simulation in Julia takes 2 hours on my laptop.



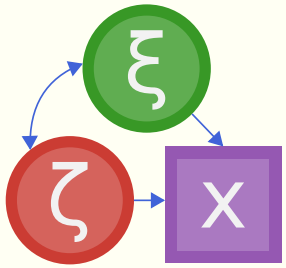
Why?

- Some investments in extensibility
- division of labor:
  - optimizing linear algebra is done by Intel
  - numerical optimization is done by dedicated experts
  - differentiation is automated
- modern infrastructure



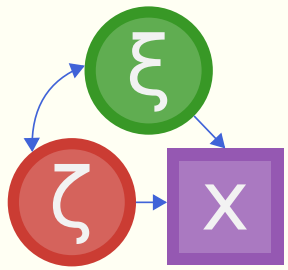
But why?

convenience

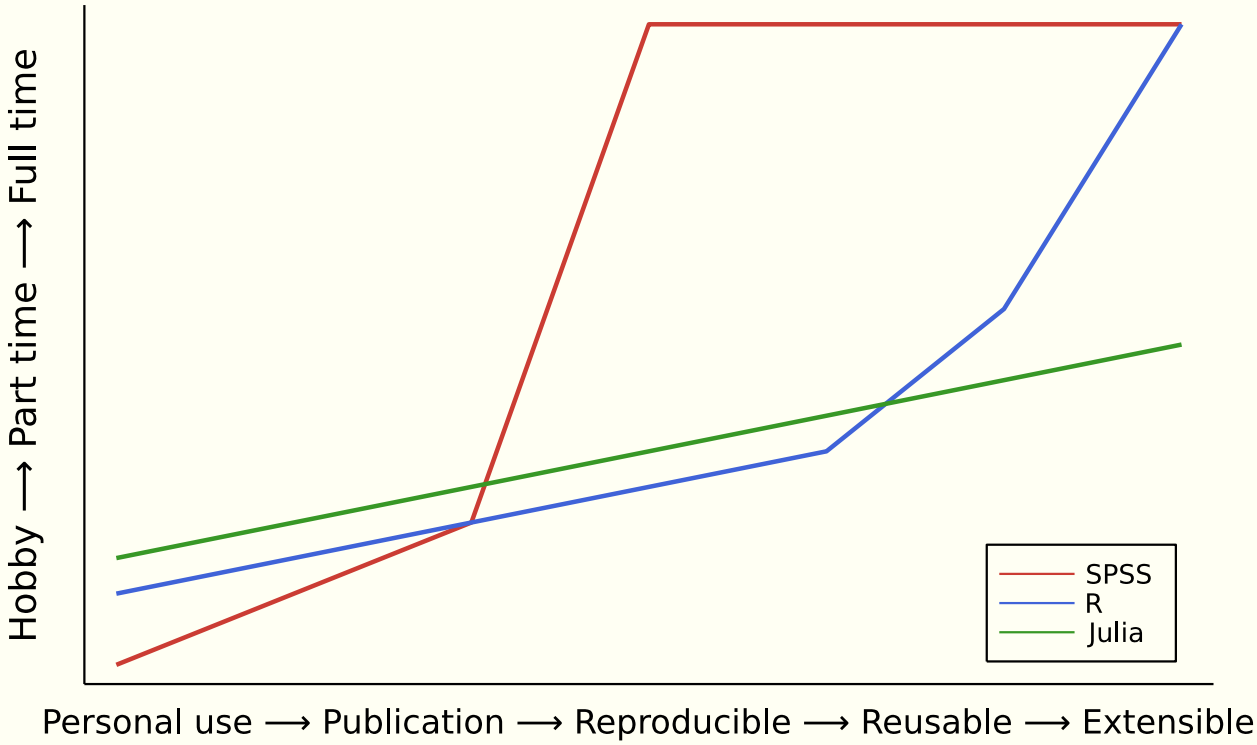


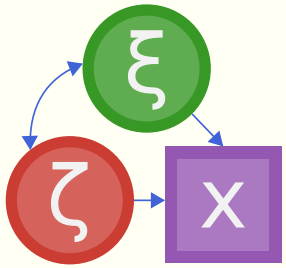
How to improve convenience?

1. Extensible Software
2. Documentation
3. User Interface

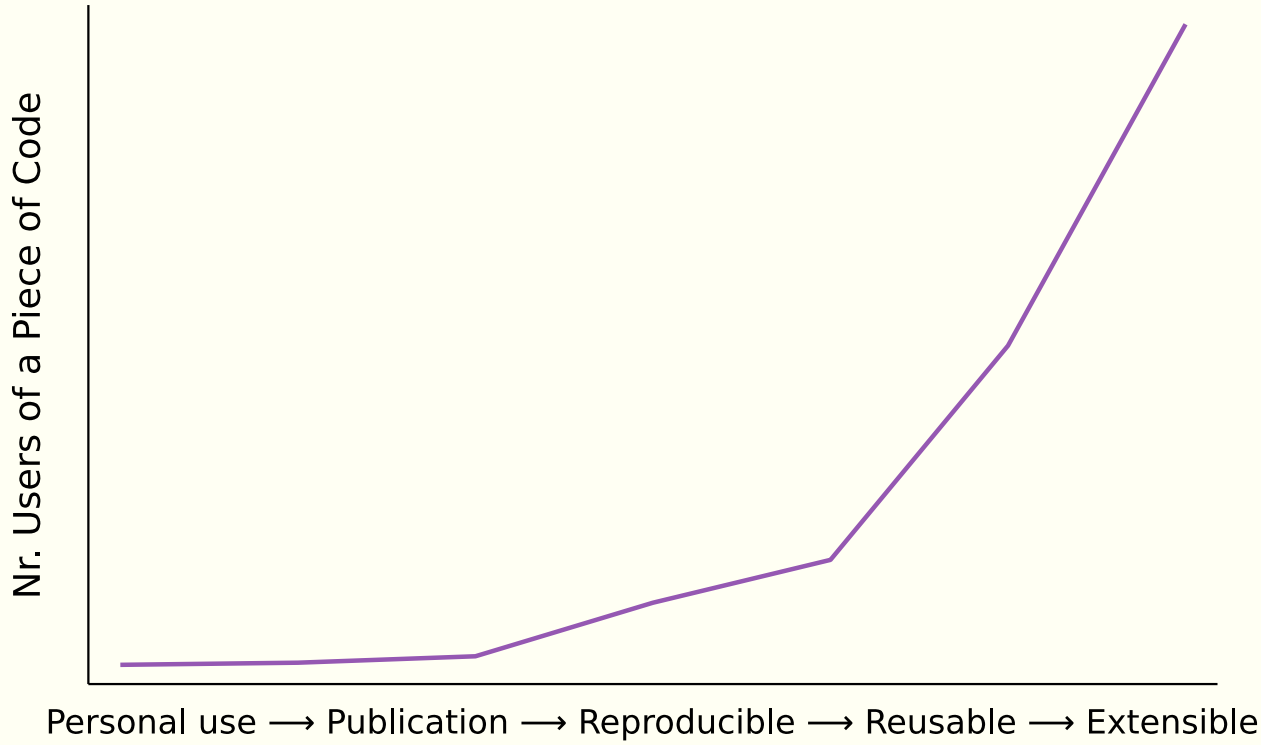


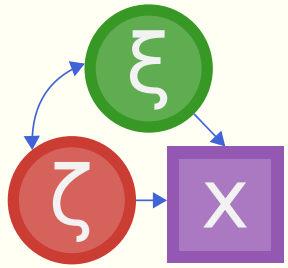
Time is limited





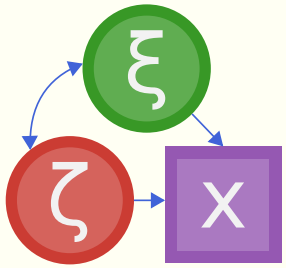
The leverage of extensible software





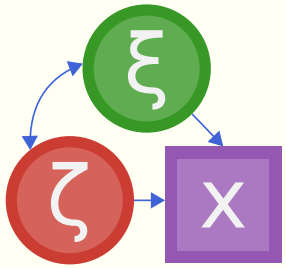
Documentation

- Documentation for users
- Documentation for contributors/developers



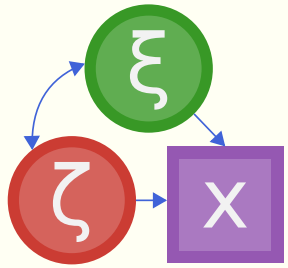
Documentation

- Documentation for users
- Documentation for contributors/developers
- Documentation is not always called documentation (e.g. papers/talks/blog posts)



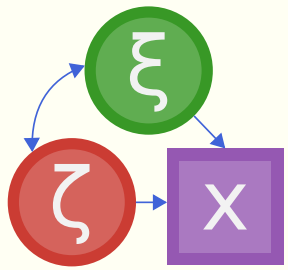
Documentation

- Documentation for users
- Documentation for contributors/developers
- Documentation is not always called documentation (e.g. papers/talks/blog posts)
- Code itsself is the best developer documentation

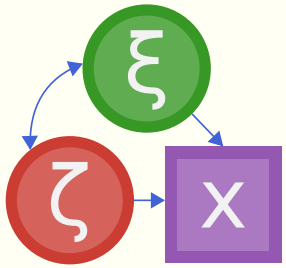


User Interface

- Frictionless
- Connected to prior knowledge



Less Abstract



StructuralEquationModels.jl