

CI / CD Pipeline Tools and Their Signals

Bogdan Vasilescu
Microsoft Faculty Summit 2018

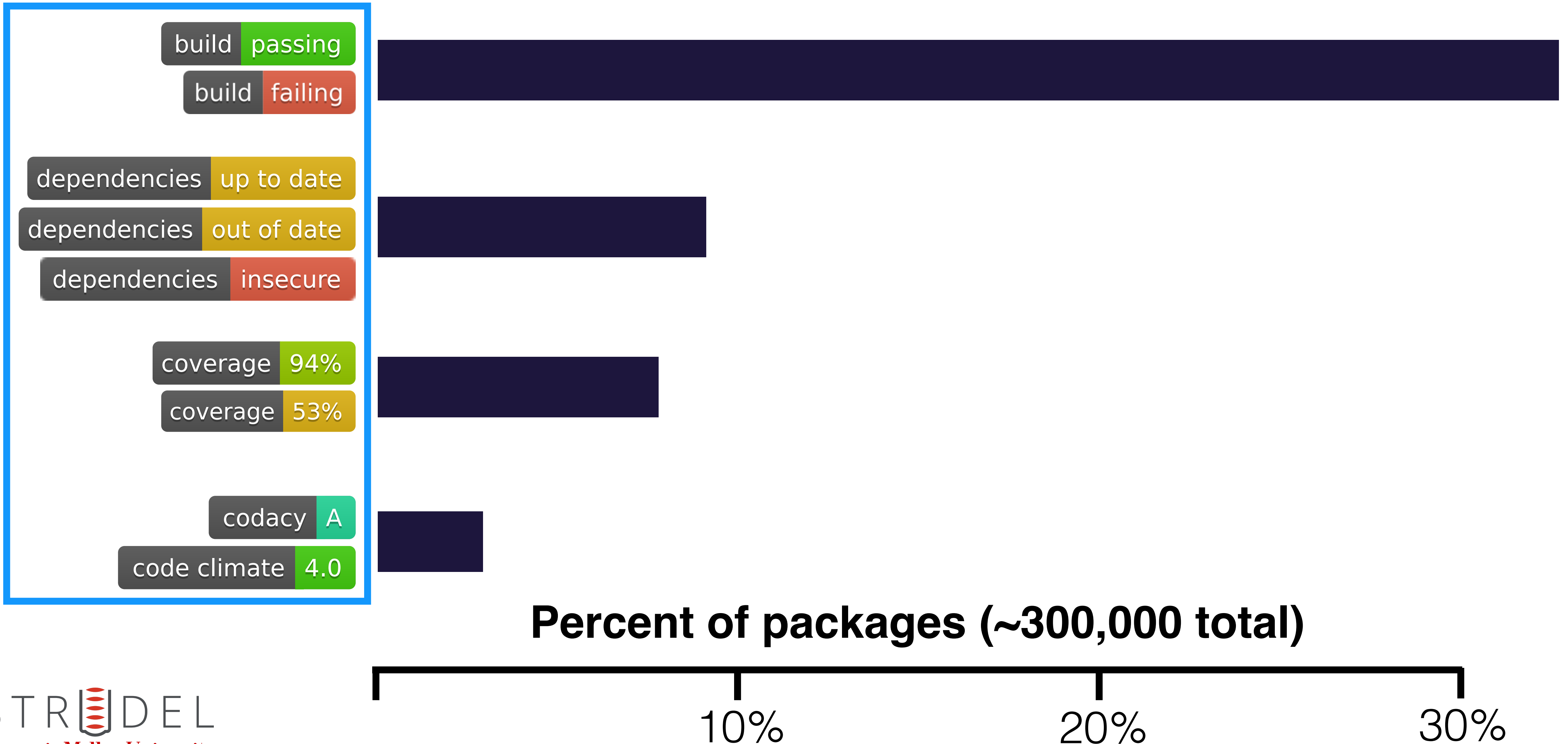
joint work with Trockman, Zhou, Kästner. ICSE 2018

Tools being used in

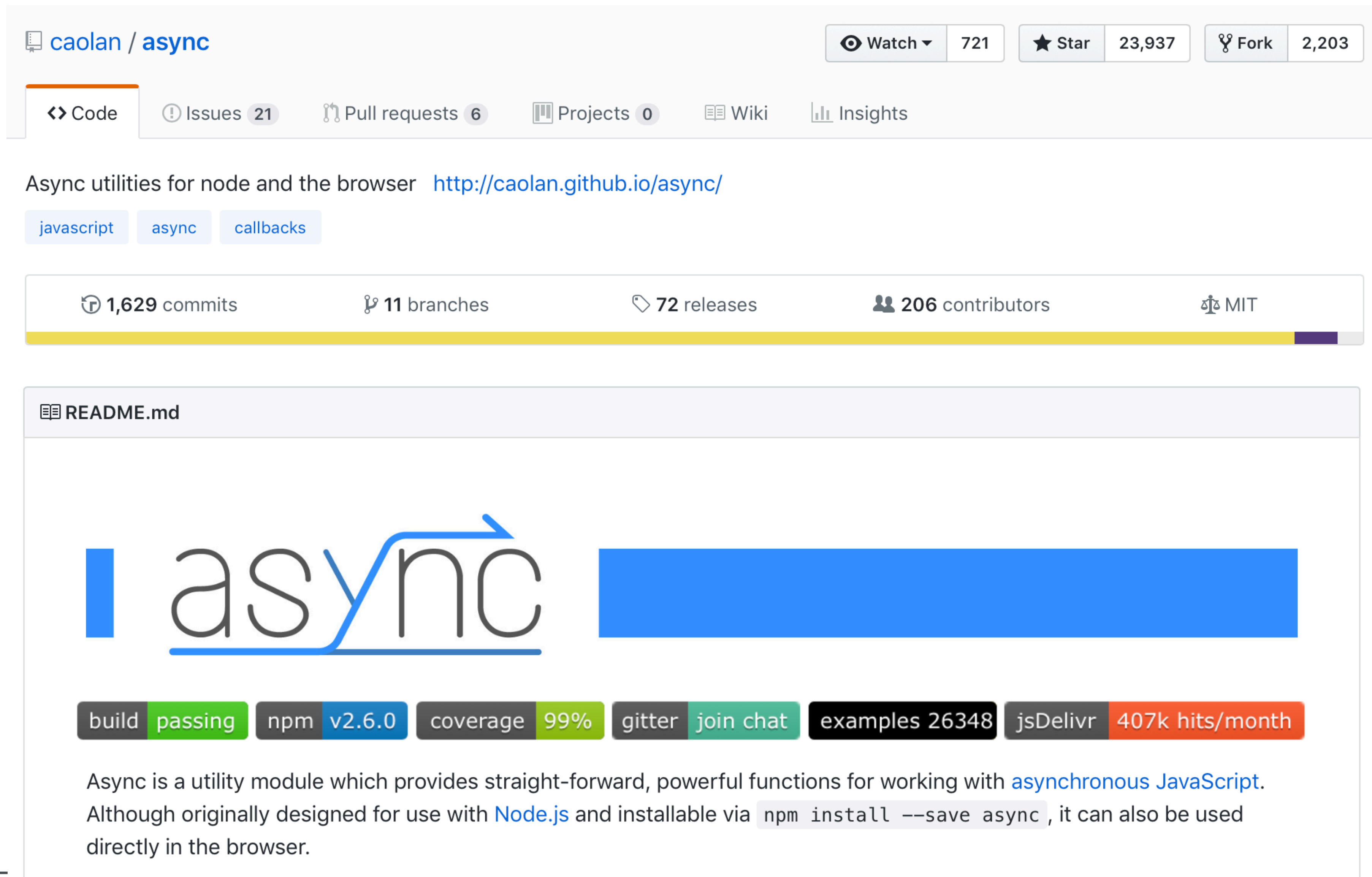


Percent of packages (~300,000 total)

Tools being used in



GitHub Repository Badges



The screenshot displays the GitHub repository page for 'caolan / async'. At the top, the repository name is shown with navigation links for 'Code', 'Issues' (21), 'Pull requests' (6), 'Projects' (0), 'Wiki', and 'Insights'. To the right, there are buttons for 'Watch' (721), 'Star' (23,937), and 'Fork' (2,203). Below this, a description states 'Async utilities for node and the browser' with a link to 'http://caolan.github.io/async/'. There are also tags for 'javascript', 'async', and 'callbacks'. A horizontal bar shows repository statistics: '1,629 commits', '11 branches', '72 releases', '206 contributors', and 'MIT' license. The main content area shows the 'README.md' file with the 'async' logo, a blue bar, and a row of badges: 'build passing', 'npm v2.6.0', 'coverage 99%', 'gitter join chat', 'examples 26348', 'jsDelivr 407k hits/month'. Below the badges, the README text describes the module as a utility for asynchronous JavaScript, mentioning its use with Node.js and the command 'npm install --save async'.

caolan / async

Watch 721 Star 23,937 Fork 2,203

Code Issues 21 Pull requests 6 Projects 0 Wiki Insights

Async utilities for node and the browser <http://caolan.github.io/async/>

javascript async callbacks

1,629 commits 11 branches 72 releases 206 contributors MIT

README.md

async

build passing npm v2.6.0 coverage 99% gitter join chat examples 26348 jsDelivr 407k hits/month

Async is a utility module which provides straight-forward, powerful functions for working with [asynchronous JavaScript](#). Although originally designed for use with [Node.js](#) and installable via `npm install --save async`, it can also be used directly in the browser.

Social coding: Transparency & signaling

request / request

Watch

417

Star

18,384

Fork

2,196

<> Code

Issues 578

Pull requests 52

Projects 0

Wiki

Insights

Simplified HTTP request client.

2,199 commits

17 branches

134 releases

270 contributors

Apache-2.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

mikeal

committed on Sep 27, 2017 2.83.1

Latest commit 253c5e5 on Sep 27, 2017

.github	small change to template wording	a year ago
examples	Adds example for Tor proxy	2 years ago
lib	refactor(lint): replace eslint with standard (#2579)	11 months ago
tests	lint fix, PR from pre-standard was merged with passing tests	10 months ago
.gitignore	Updating deps.	5 months ago
.travis.yml	Add Node.js v8 to Travis CI	7 months ago
CHANGELOG.md	Update changelog	11 months ago
CONTRIBUTING.md	Update contributing guidelines	2 years ago
LICENSE	Adding license information.	7 years ago

Projects

776

Followers

38

Starred

15

Following

Organizations











Users

Contributions

Repositories

Public activity

Follow

Popular repositories

Repositories contributed to

breakfast-repo

208 ★
a collection of videos, recordings, and podcast...

x86-kernel

48 ★
a simple x86 kernel, extended with Rust

ashleygwilliams.github.io

37 ★
hi, i'm ashley. nice to meet you.

jsconf-2015-deck

32 ★
deck for jsconf2015 talk, "if you wish to learn e..."

ratpack

32 ★
sinatra boilerplate using activerecord, sqlite, a...

npm/docs

44 ★
The place where all the npm docs live.

mozilla/publish.webmaker.org

2 ★
The teach.org publishing service for goggles a...

npm/marky-markdown

104 ★
npm's markdown parser

artisan-tattoo/assistant-frontend

5 ★
ember client for assistant-API

npm/npm-camp

1 ★
a community conference for all things npm

Public contributions

Summary of pull requests, issues opened, and commits. [Learn how we count contributions.](#)

Less More

Contributions in the last year

1,886 total
Jan 24, 2015 – Jan 24, 2016

Longest streak

37 days
October 7 – November 12

Current streak

7 days
January 18 – January 24

Signals can also be physical

MONITORING YOUR CONTINUOUS INTEGRATION SERVER WITH TRAFFIC LIGHTS AND AN ARDUINO

Note: If you like this post, you'll probably also like [the update on our Jenkins](#)

So how does one connect a 48" tall traffic light to a continuous integration server course.



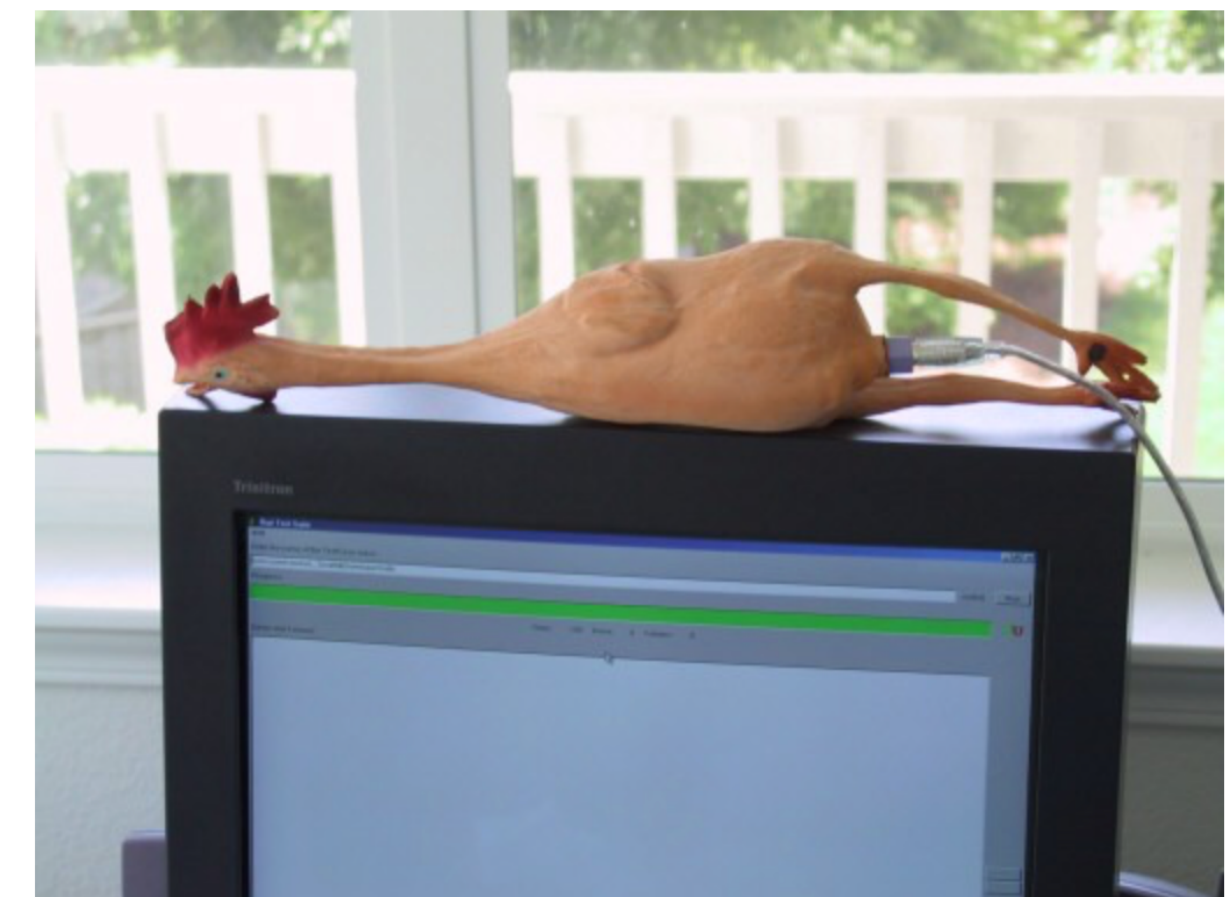
Signalling Your Jenkins Build Status with a Mini USB Traffic Light

by Julian Heise - 8 Jun 2017



Continuous Integration of Shame

A conversation on Twitter recently popped up on the use of physical objects to denote build status. You know what I mean: “You broke the build, so you get the rubber chicken! And have to display it on your desk until *someone else* breaks the build!”



Build status: plucky. Photo credit: James Shore.

“Peer monitoring”

Phillips, Zimmermann, Bird. Understanding and improving software build teams. ICSE 2014

What do developers expect from badges?

“welcoming contributions”

- 32 Maintainers
 - What do you intend to signal?
 - What effects do you expect?

“expectations of contribution quality”

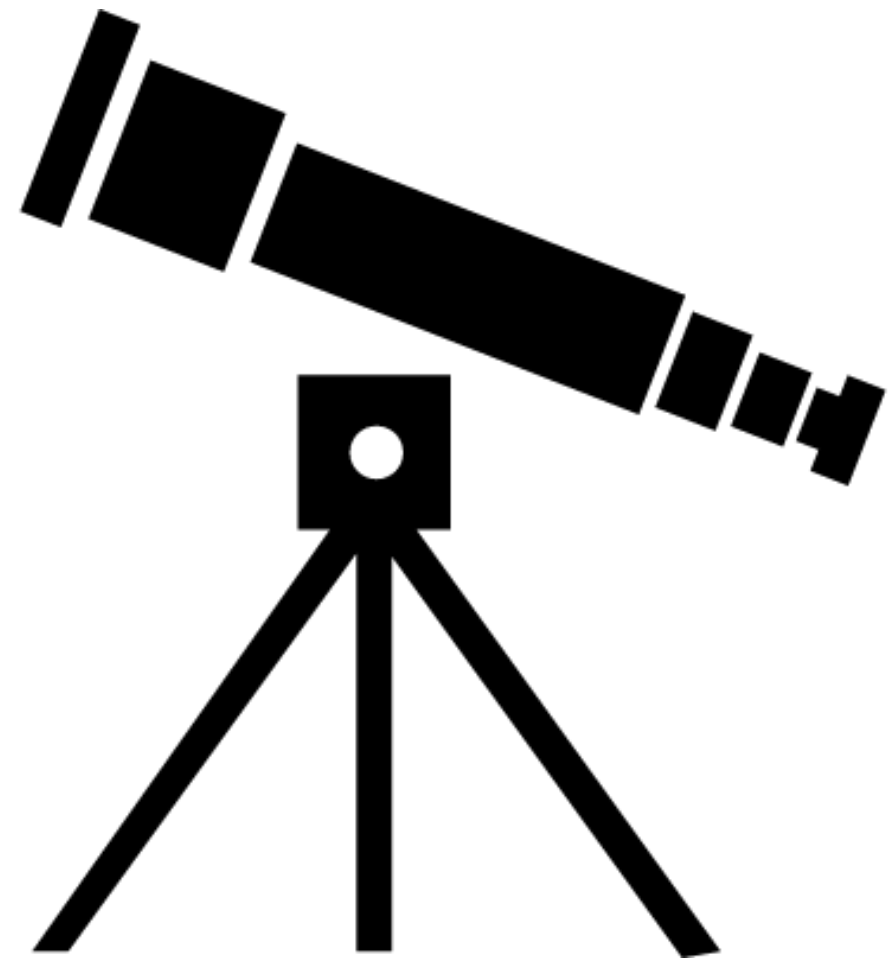
- 57 Contributors
 - What do badges tell you?

“dedicated to offering support”

“reduced chances of conflicting versions of dependencies”

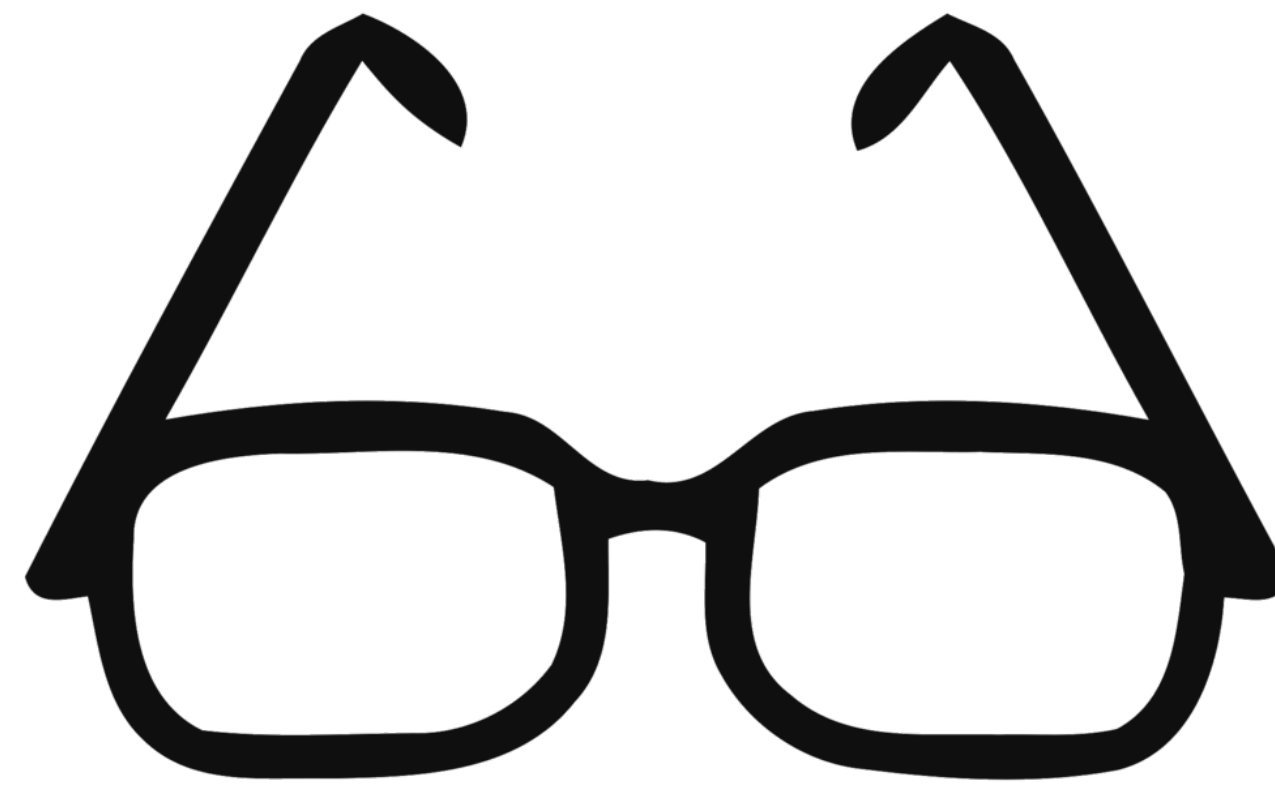
“indicator of product quality”

Analysis



Correlation

If all you saw was the badge, how much would that tell you?



Regression Analysis

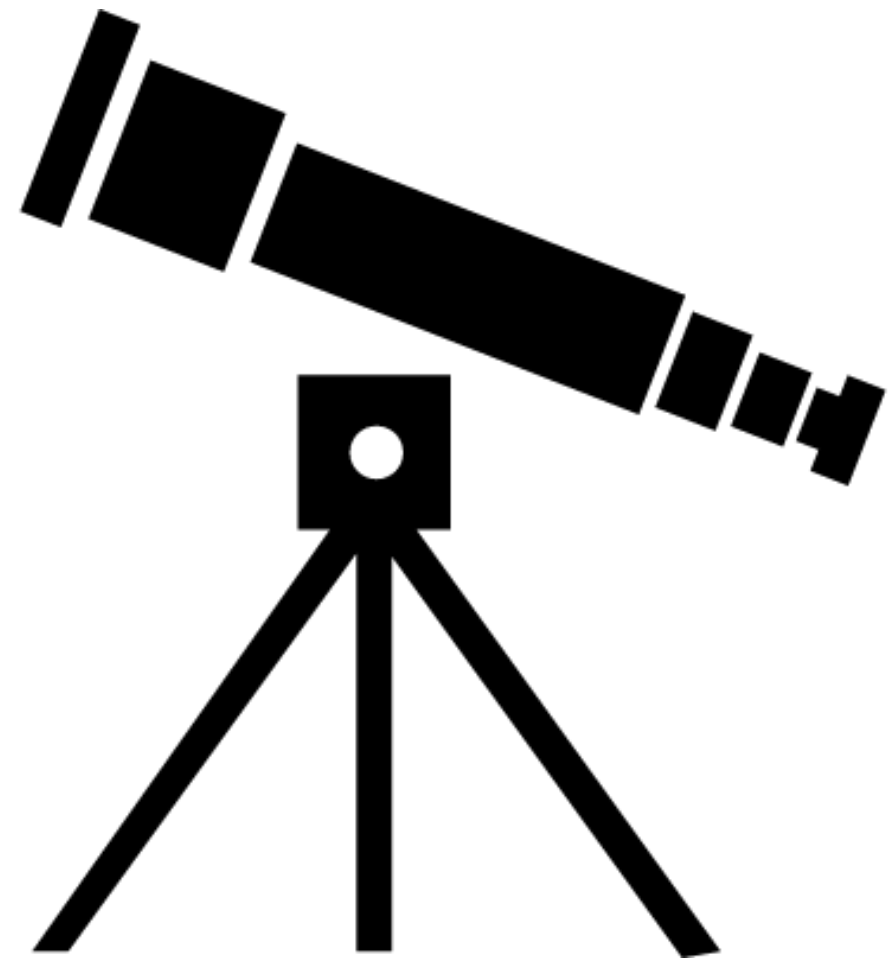
How much more does the badge tell you, relative to existing signals?



Time Series Analysis

How do things change after adding the badge?

Analysis



Correlation

If all you saw was the badge, how much would that tell you?



Regression Analysis

How much more does the badge tell you, relative to existing signals?

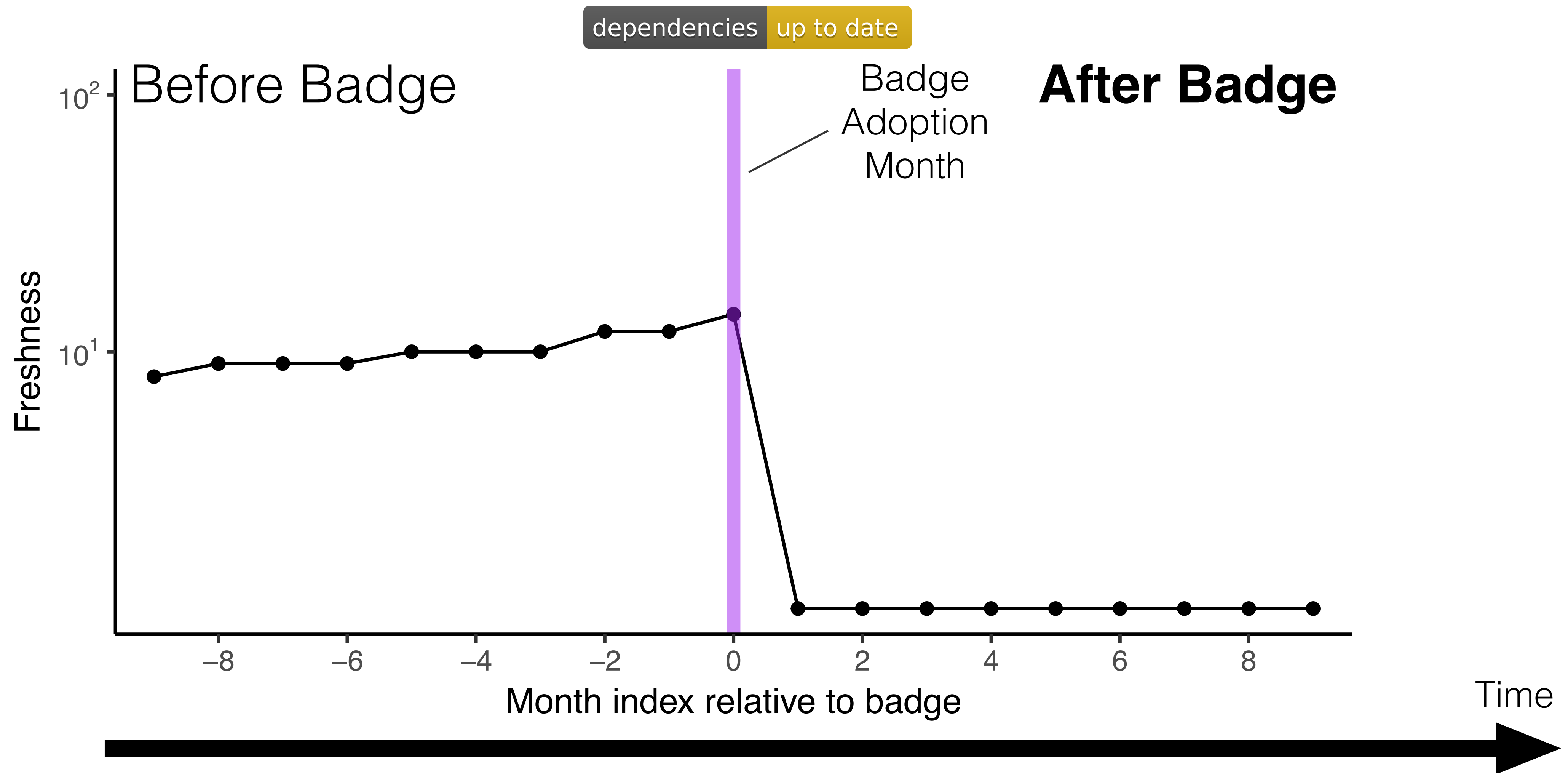


Time Series Analysis

How do things change after adding the badge?

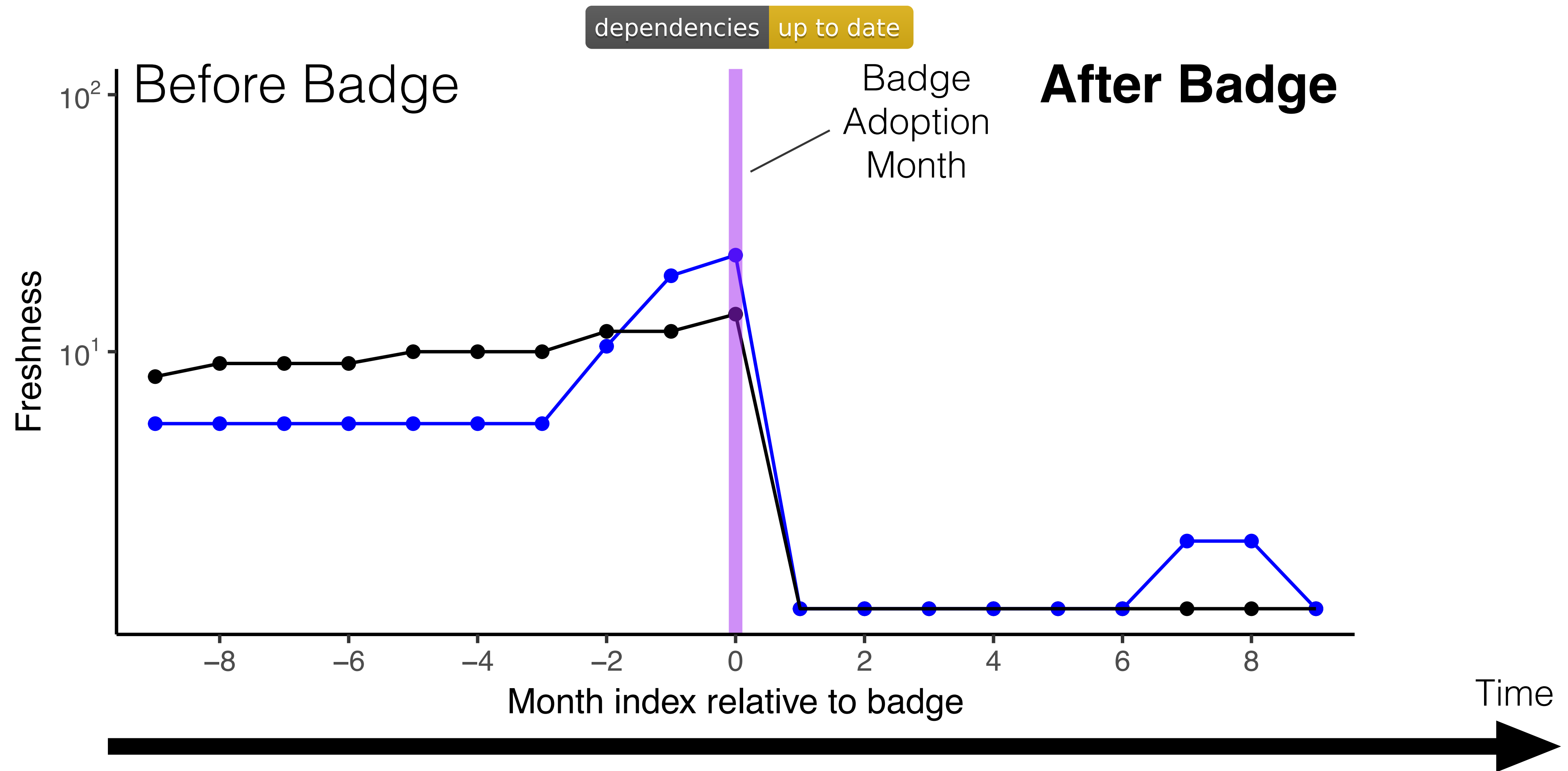
Step 3: Time Series Analysis

Signals of fresh dependencies



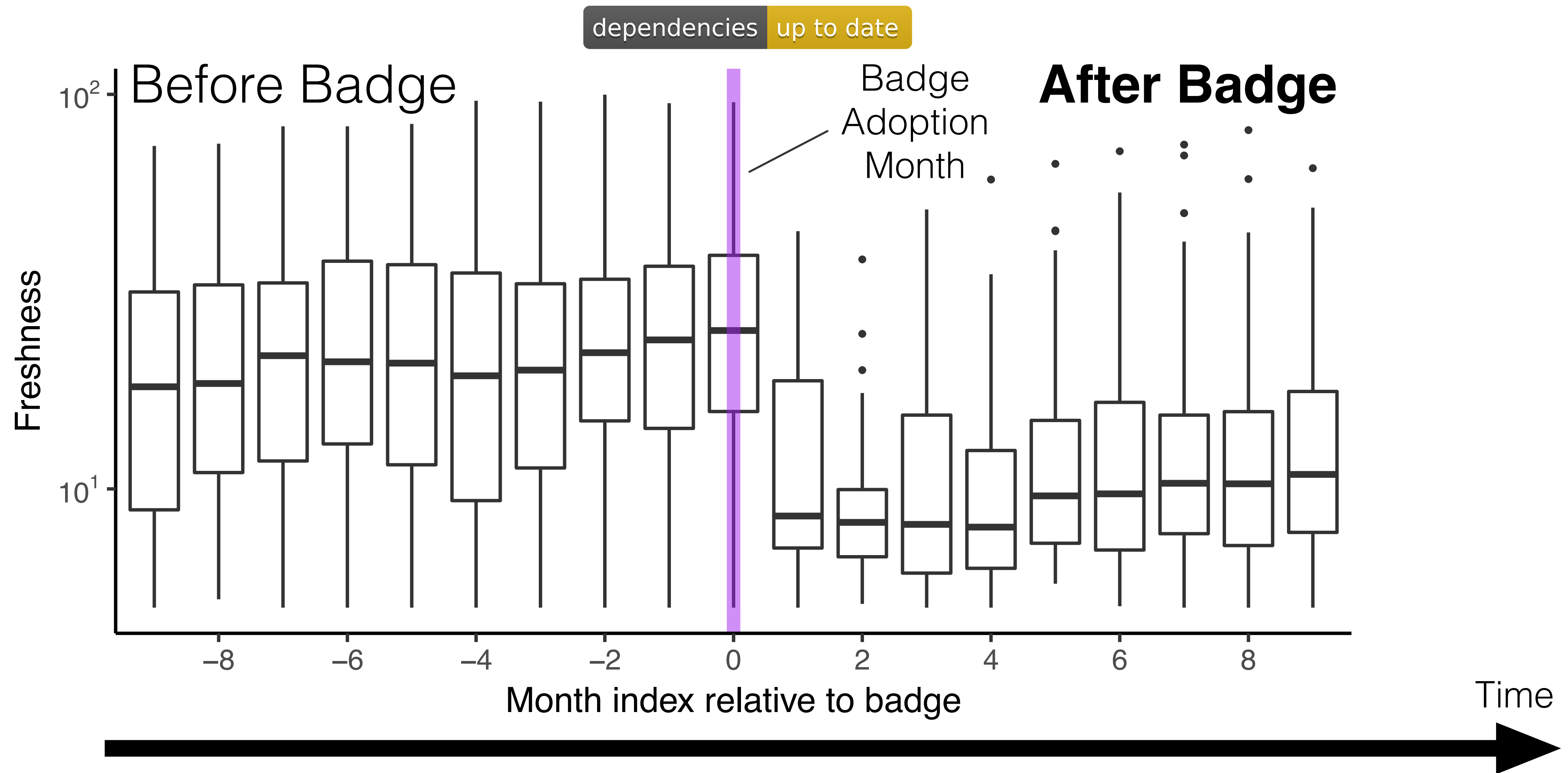
Step 3: Time Series Analysis

Signals of fresh dependencies



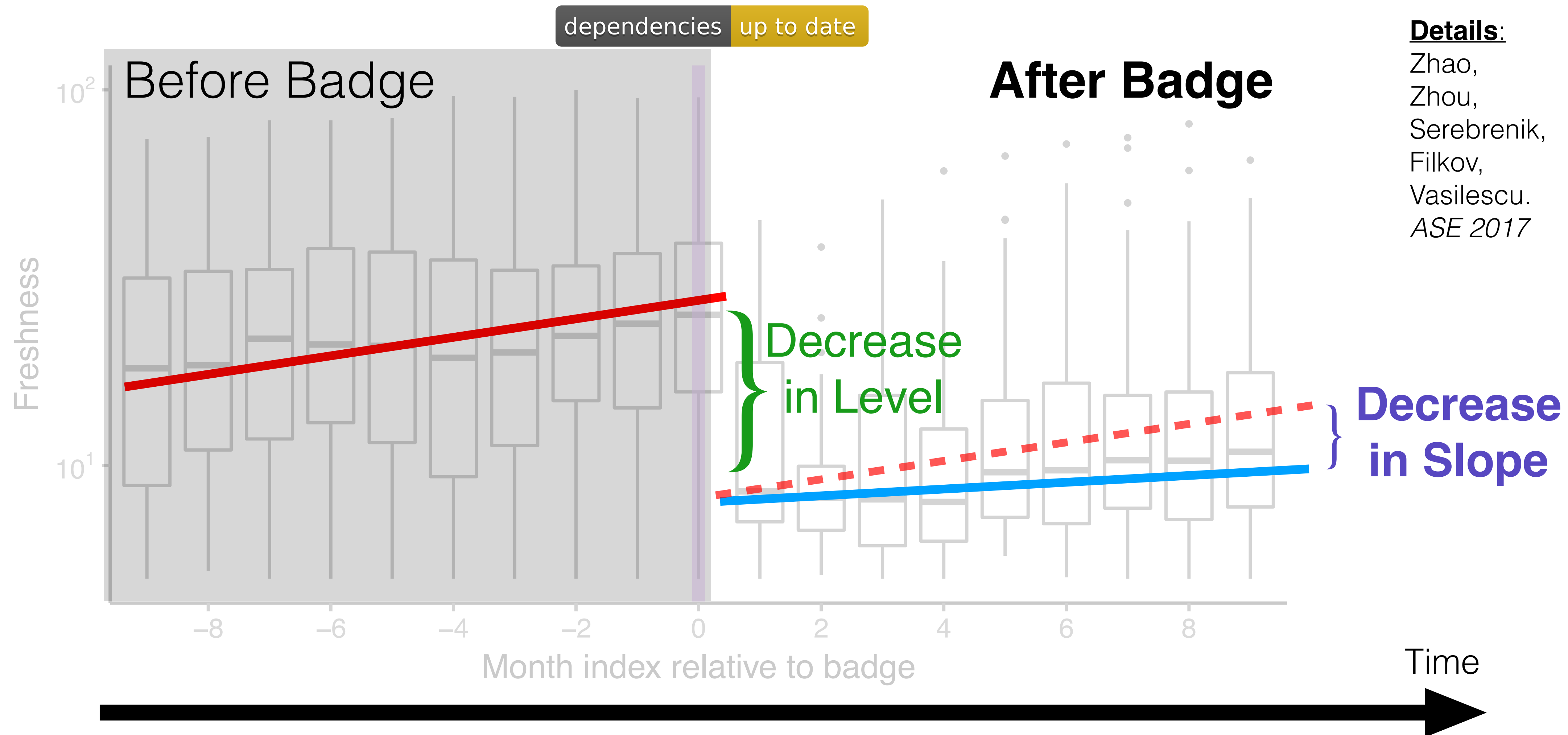
Step 3: Time Series Analysis

Signals of fresh dependencies



Step 3: Time Series Analysis

Signals of fresh dependencies



Details:

Zhao,
Zhou,
Serebrenik,
Filkov,
Vasilescu.
ASE 2017

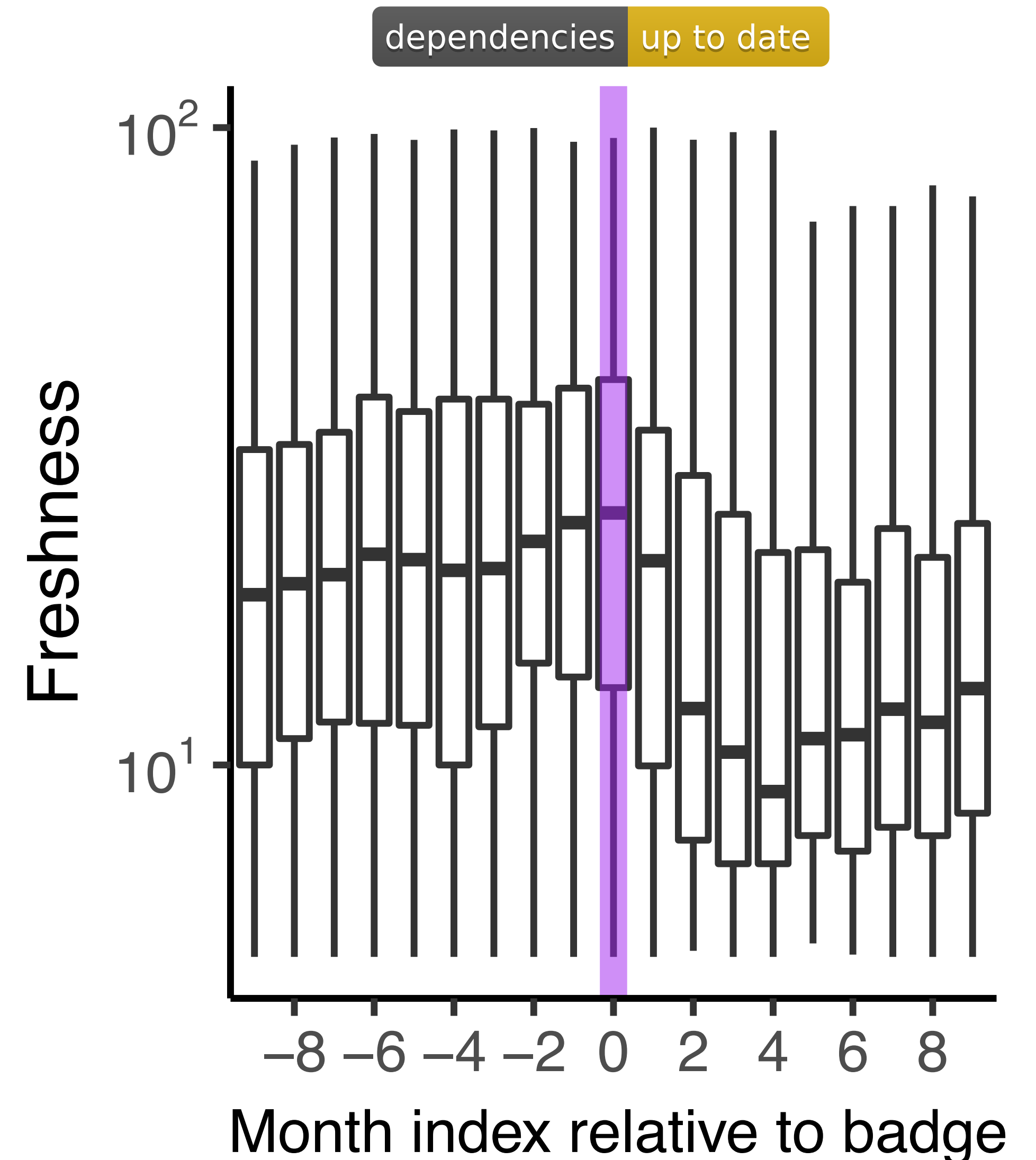
Step 3: Time Series Analysis

Signals of fresh dependencies

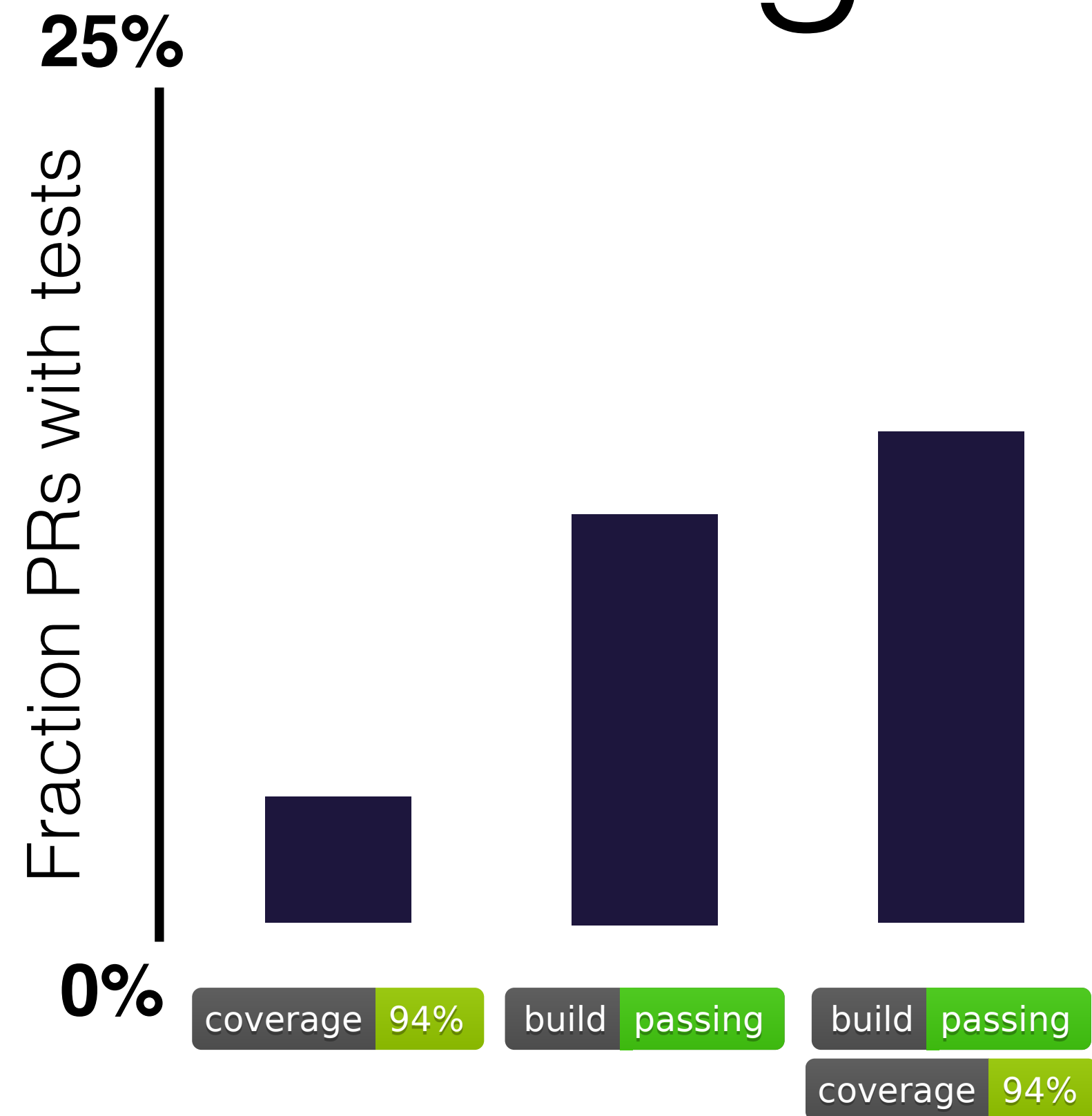


- **Based on survey:** The adoption of dependency management badges correlates with fresher dependencies
- **Freshness metric:** *lower* is better
 - (More up-to-date deps.)

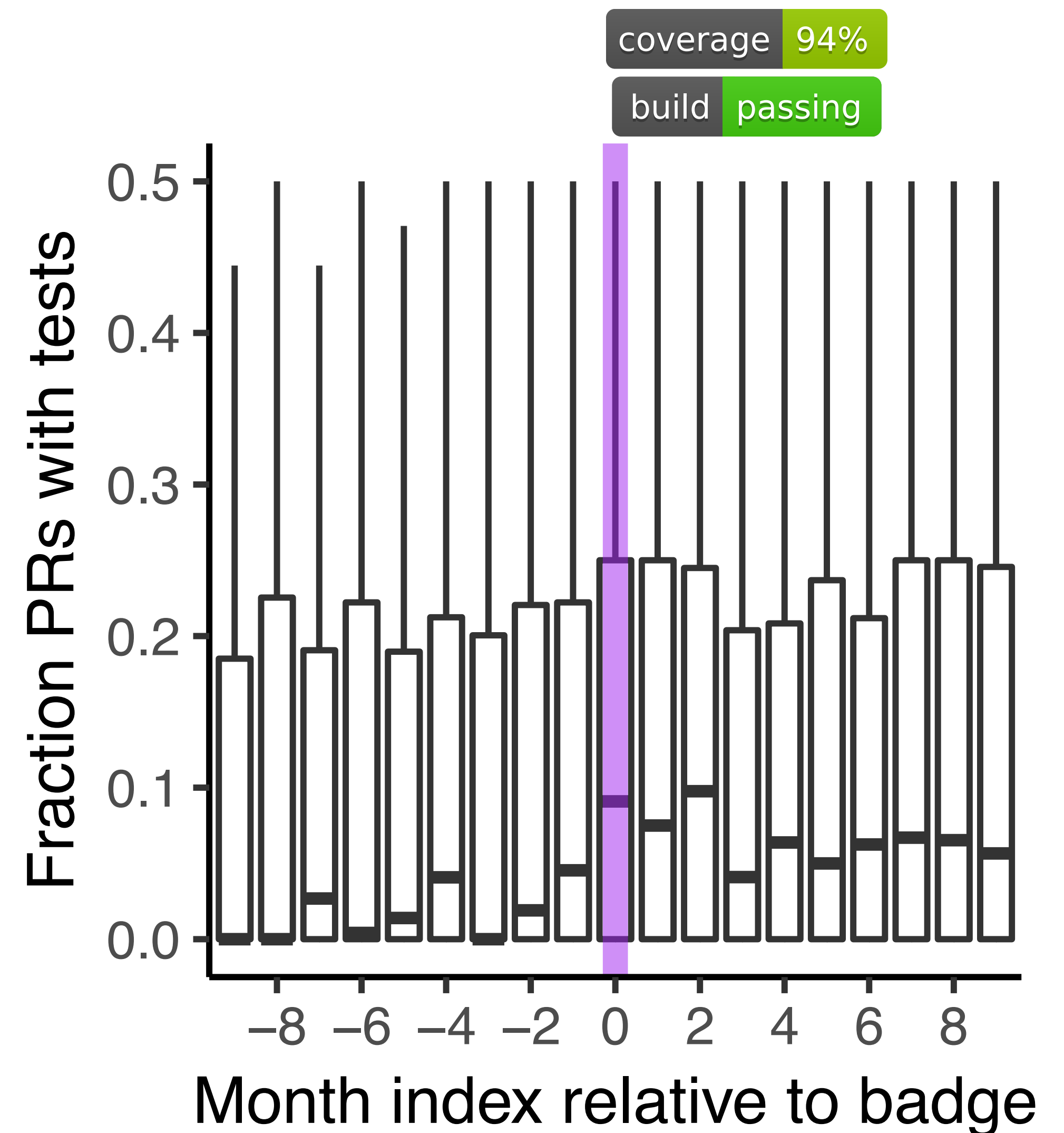
Result: Dep. badges indicate *improved dep. management practices*



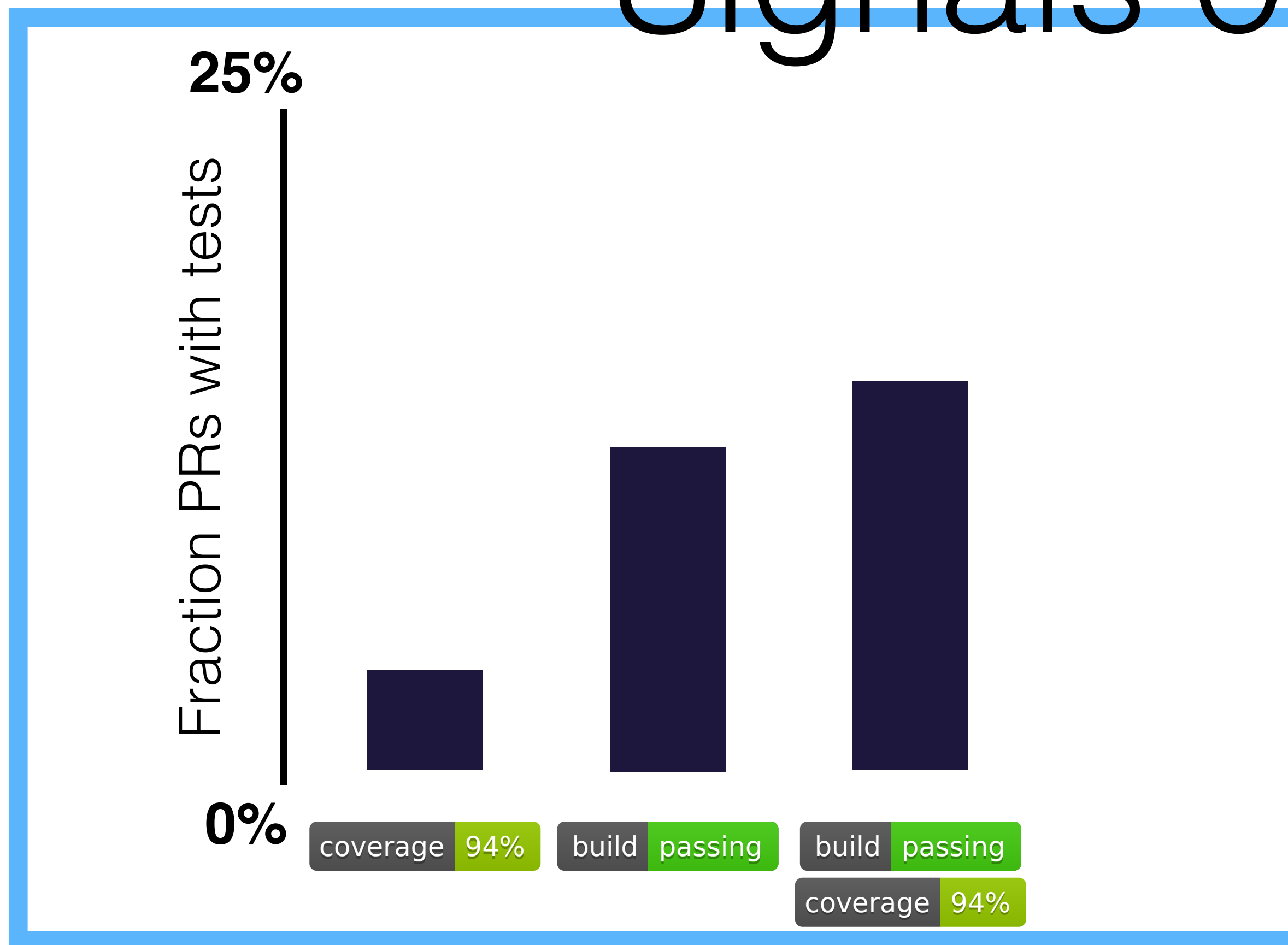
Signals of PR quality



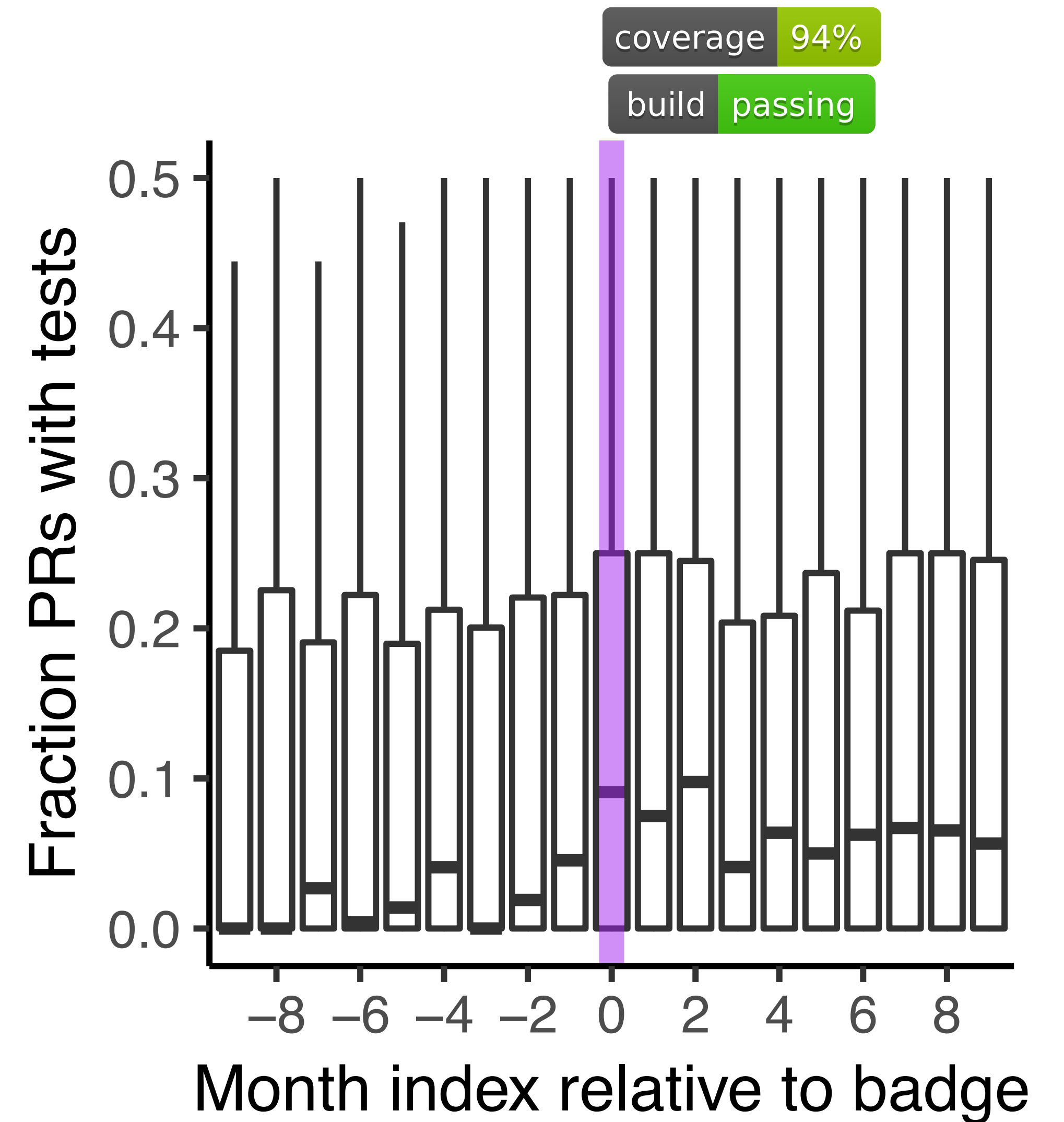
Result: Build status+code coverage badges indicate *more tests in PRs*



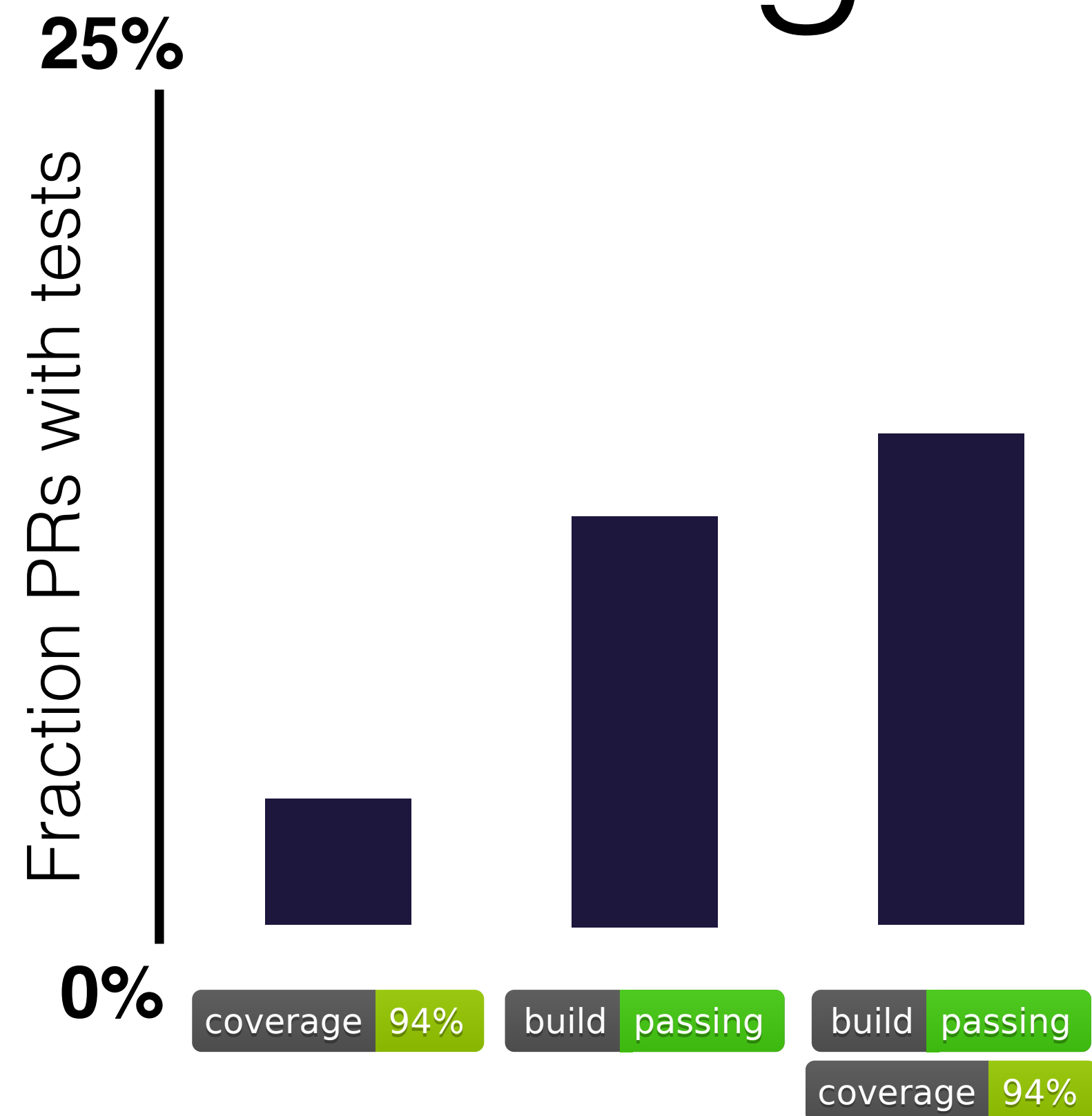
Signals of PR quality



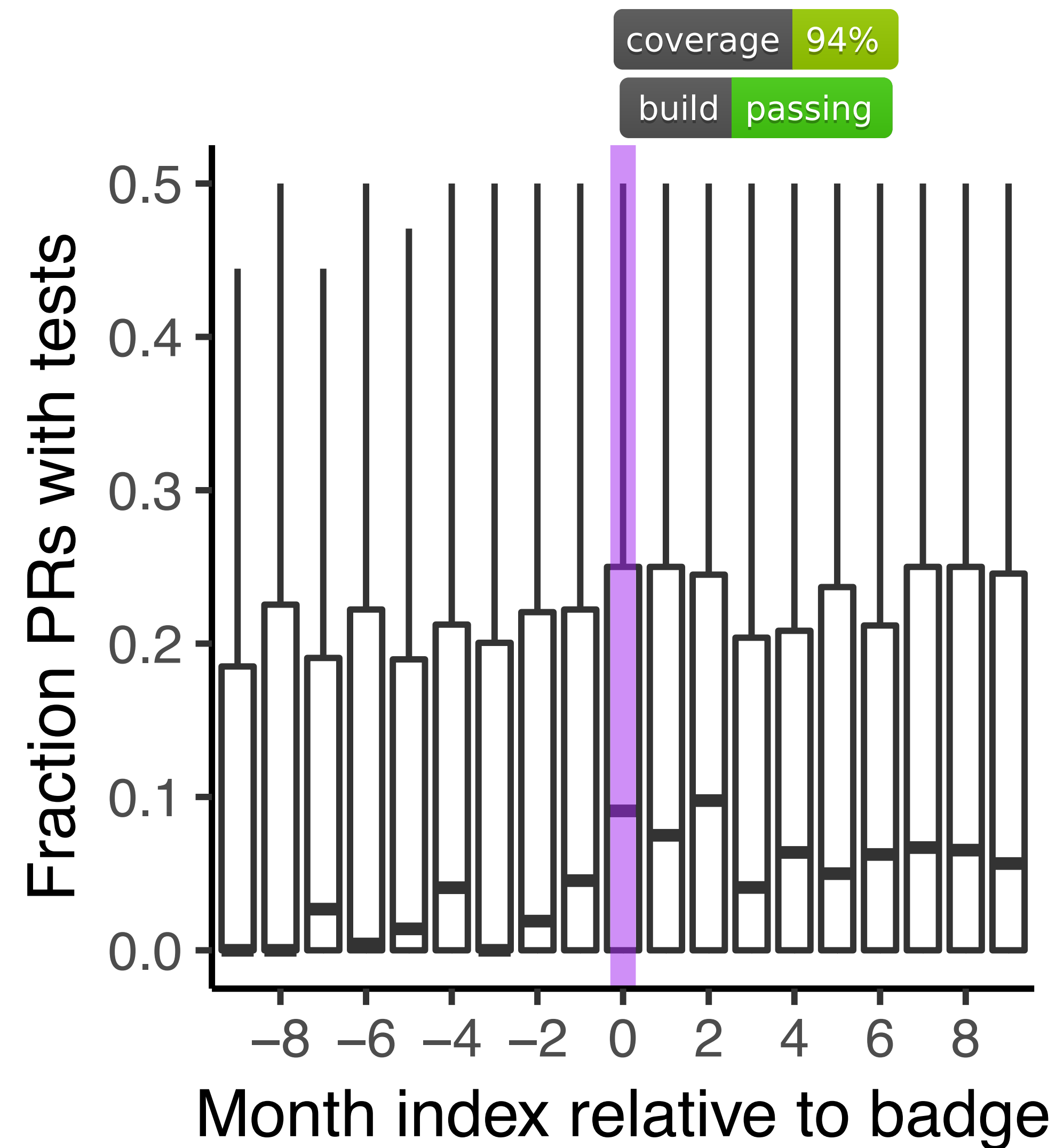
Result: Build status+code coverage badges indicate *more tests in PRs*



Signals of PR quality



Result: Build status+code coverage badges indicate *more tests in PRs*

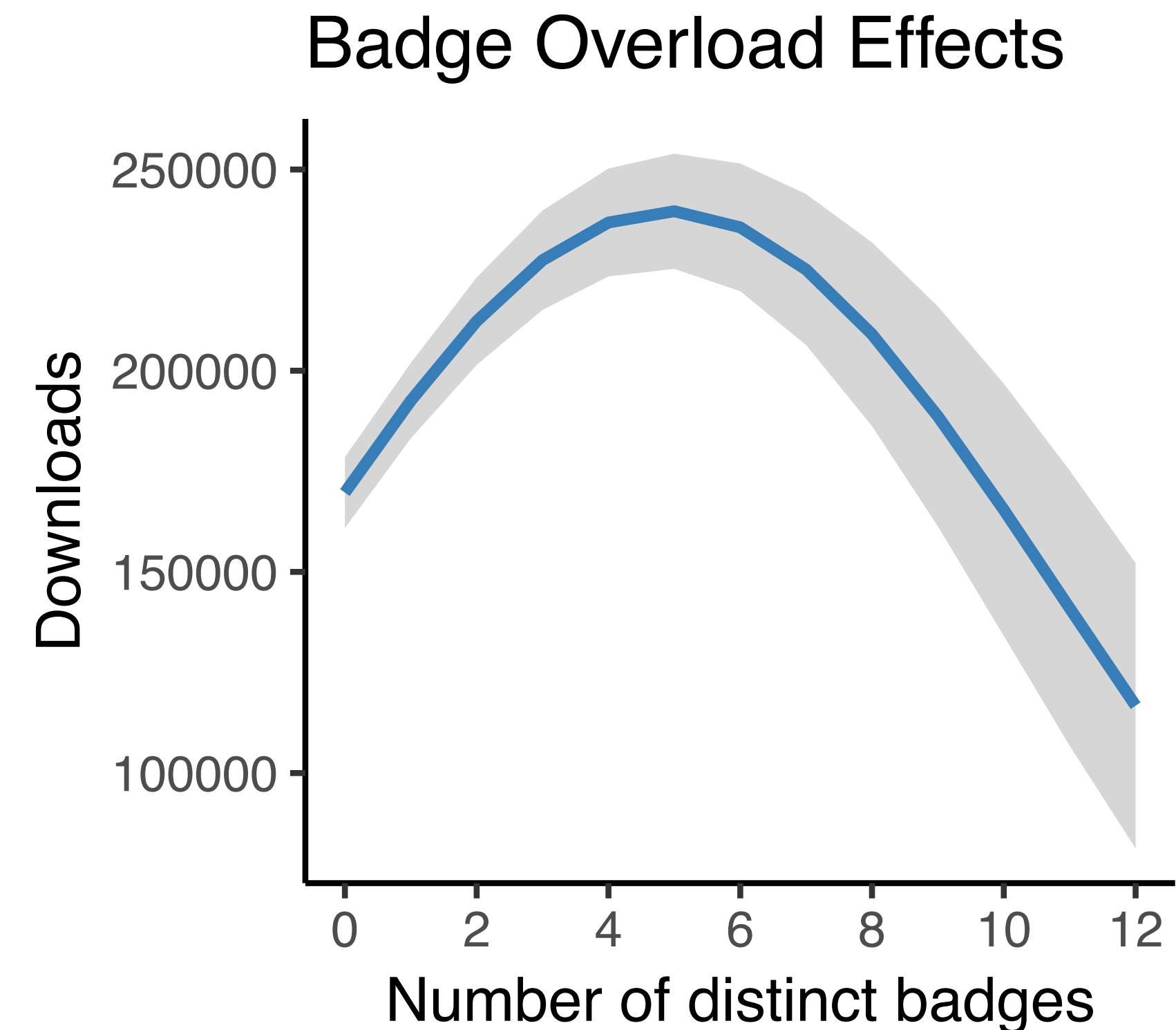


Too many badges?

- **Based on survey:**

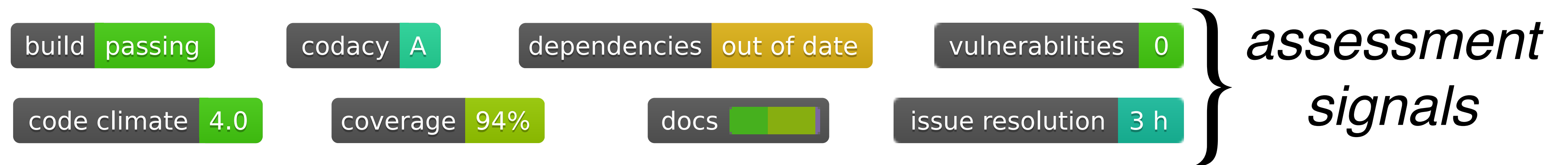
“People tend to overwhelm visitors with too many (useless) badges, thus creating a contra effect and loosing the initial purpose of having useful information.”

Result: Attractiveness wears off beyond five badges

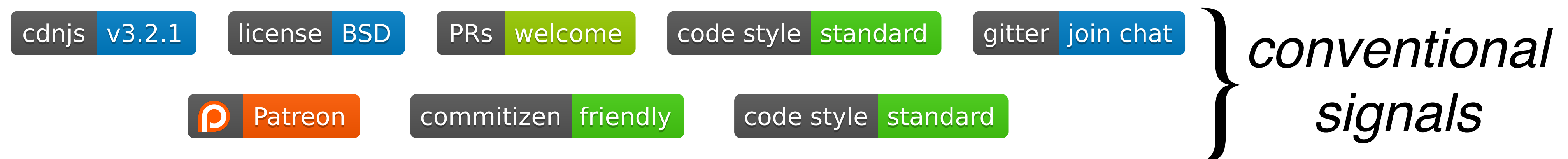


Take-aways

Badges with **underlying analyses**:



are **stronger predictors** than badges that merely state intentions or provide links:

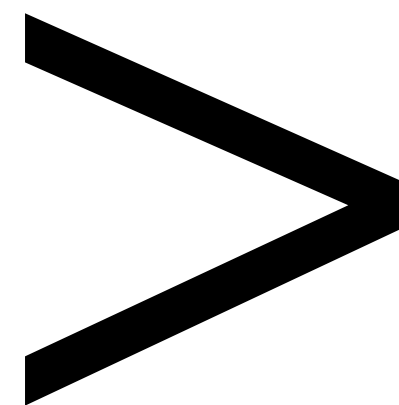


Take-aways

When possible,
design or choose the badge that takes the **most work**:



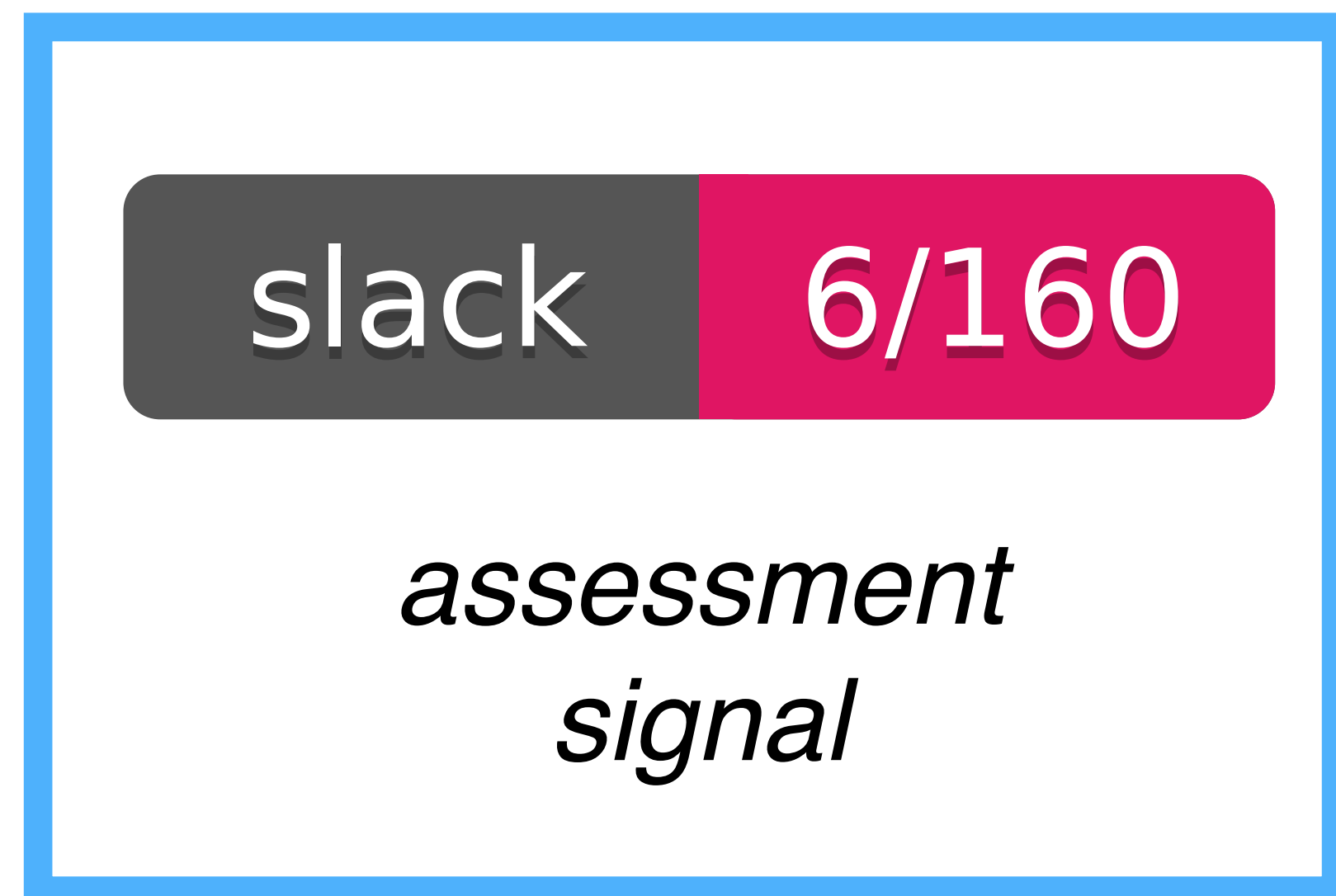
*assessment
signal*



*conventional
signal*

Take-aways

When possible,
design or choose the badge that takes the **most work**:



>



*conventional
signal*

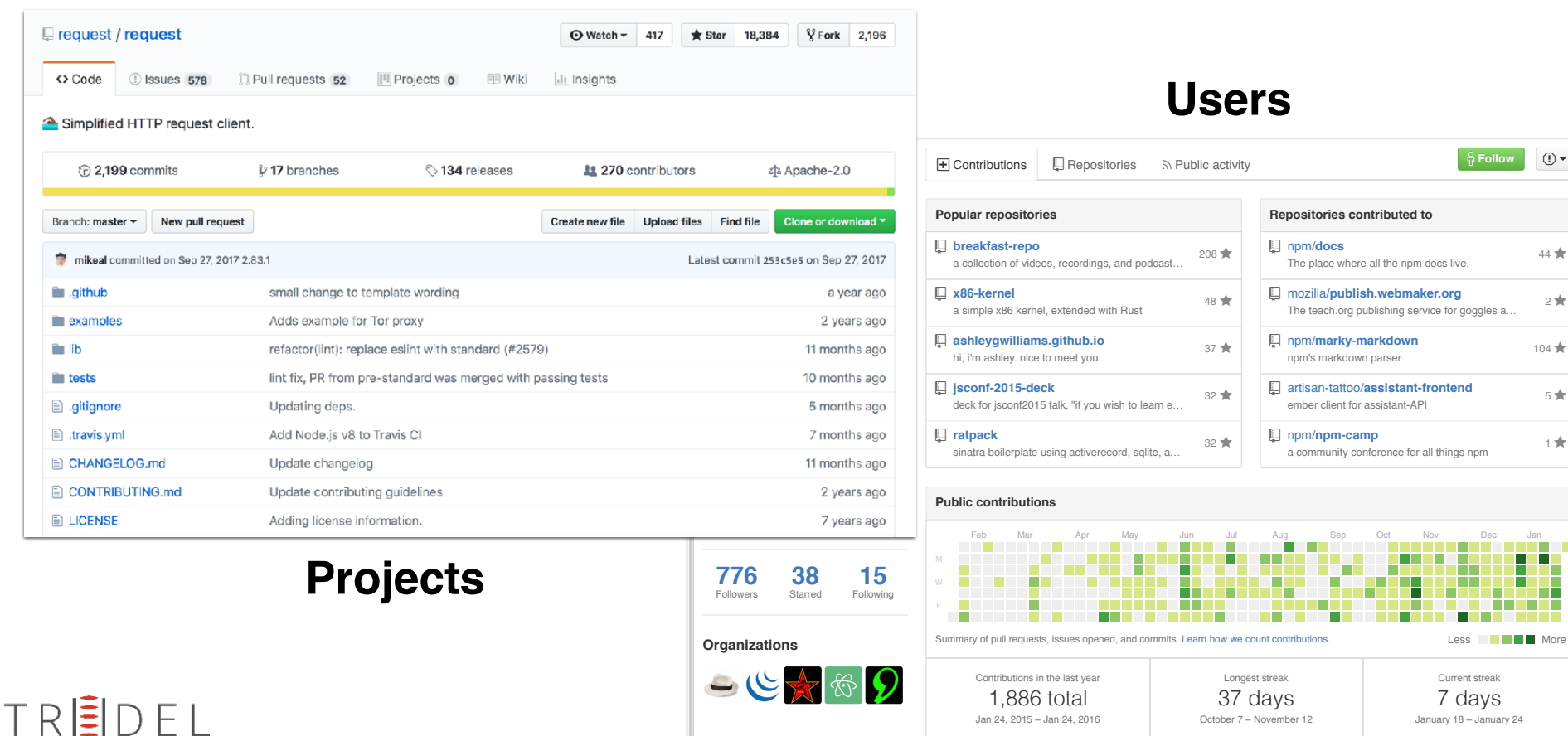
Adding Sparkle to Social Coding

badges

mostly reliable

<https://cmustrudel.github.io/badges>

Key features: Transparency & signaling



STRIDEL
Carnegie Mellon University

Mixed methods study



Survey

- 32 maintainers, 57 contributors
- Maintainers:
 - What do you intend to signal?
 - What effects do you expect?
- Contributors:
 - What do badges tell you?



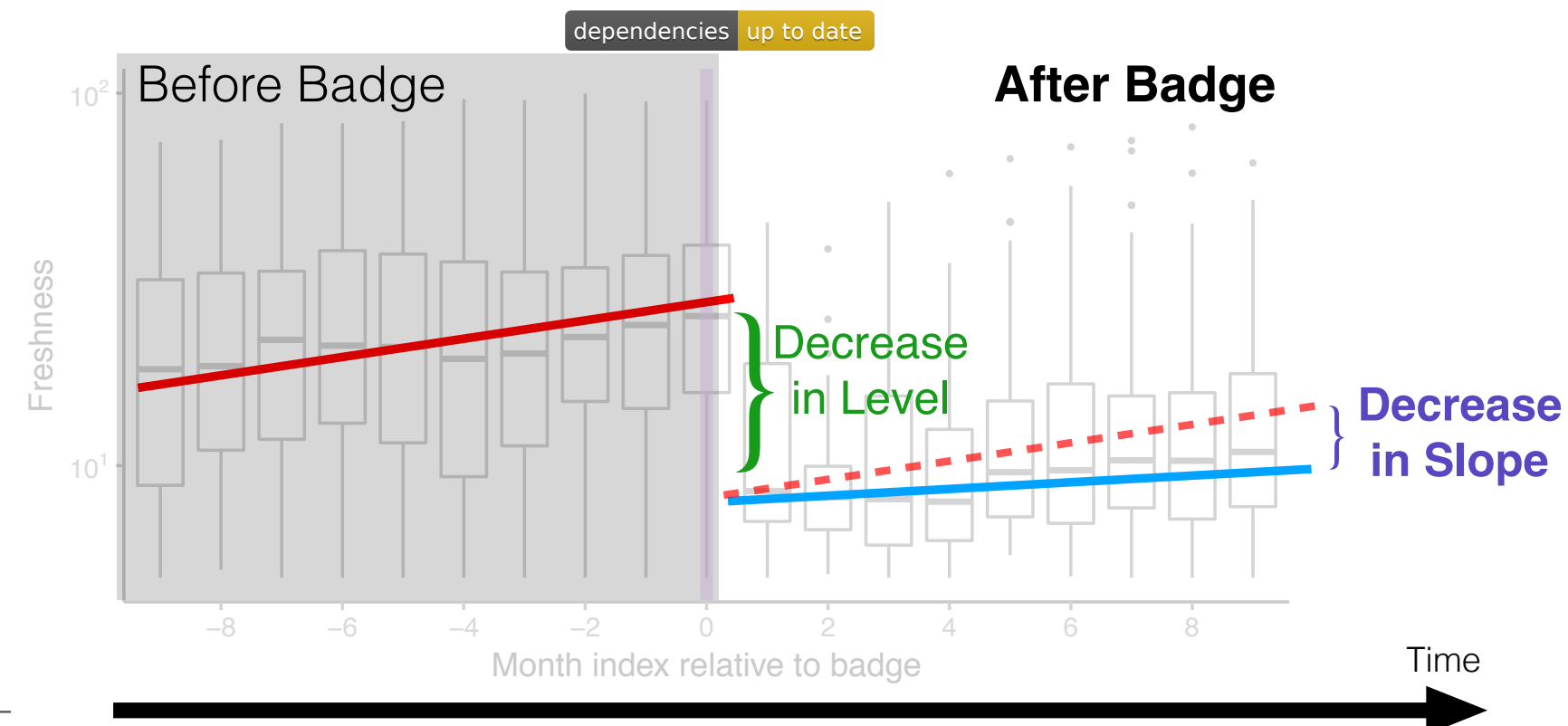
Repository Mining

- 294,941 *npm* packages
- Mined badge adoptions/removals from README files
- Measured proxies for code quality, test suite quality, popularity, dependency freshness, ...

STRIDEL
Carnegie Mellon University

Step 2: Time Series Analysis

Signals of fresh dependencies



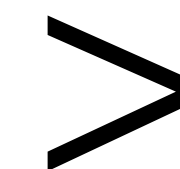
STRIDEL
Carnegie Mellon University

Take-aways

When possible,
design or choose the badge that takes the **most work**:

slack 6/160

assessment
signal



slack join

conventional
signal

STRIDEL
Carnegie Mellon University