Measuring Open Source Software Supply Chains

Audris Mockus
University of Tennessee
audris@utk.edu

[2019-08-30 Fri]

Outline

Open Source Challenges

What are Supply Chains

Big Data: Size, Quality, and Curation

What Research would be Enabled by the FLOSS Supply Chains?

How do I participate?

References

Open Source Software

- 1980-90s: a mere curiosity
  - Free Software GNU (35 years);
  - Linux (28 years);
  - Slackware (26 years);
  - Apache (24 years);
  - Mozilla (25 years)
- Attacked by critics then as
  - Expensive to maintain
  - Unsafe
  - Unsustainable

Open Source Software Miracle

- 2019:
  - Over 100M projects
  - 30M contributors
  - RedHat 3.4B revenue / 15 percent annual growth
- Why:
  - Reuse that works
  - Distributed decision making
  - Code can be trusted to be there
  - Sophisticated collaboration tools

Future is clouded

- What worked over past 20+ years may break
  - Supply chain attacks
    - Easy to exploit OSS-wide dependencies
  - Maintainer exhaustion
    - Aging out in critical projects
    - Increasing complexity
    - Increasing user base
    - Commercial participants not contributing back
  - Move to the cloud
    - No need for software when using a service
    - Amazon appropriating open source to provide competing services

Two outcomes

- Believer:
  - I am scared, tell me how to save OSS?
- Skeptic:
  - I hear emotional BS, are these problems real?

To skeptics

- E.g., NPM has over 1M packages
  - Median number of direct dependents is 2, recursive 200 (emberjs)
  - supply chain (introducing malicious dependencies) attacks are common
- Skeptic: well...
  - NPM is a house of cards built on sandcastles
  - gone with a gust of wind
  - and good riddance...
- Response:
  - Much of web infrastructure is built on NPM
  - Many developers rely/work on it
  - Would be very difficult to replace

Press Releases: arstechnica.com

A rash of supply chain attacks hitting open source software over the past year shows few signs of abating, following the discovery this week of two separate backdoors slipped into a dozen libraries downloaded by hundreds of thousands of server administrators.

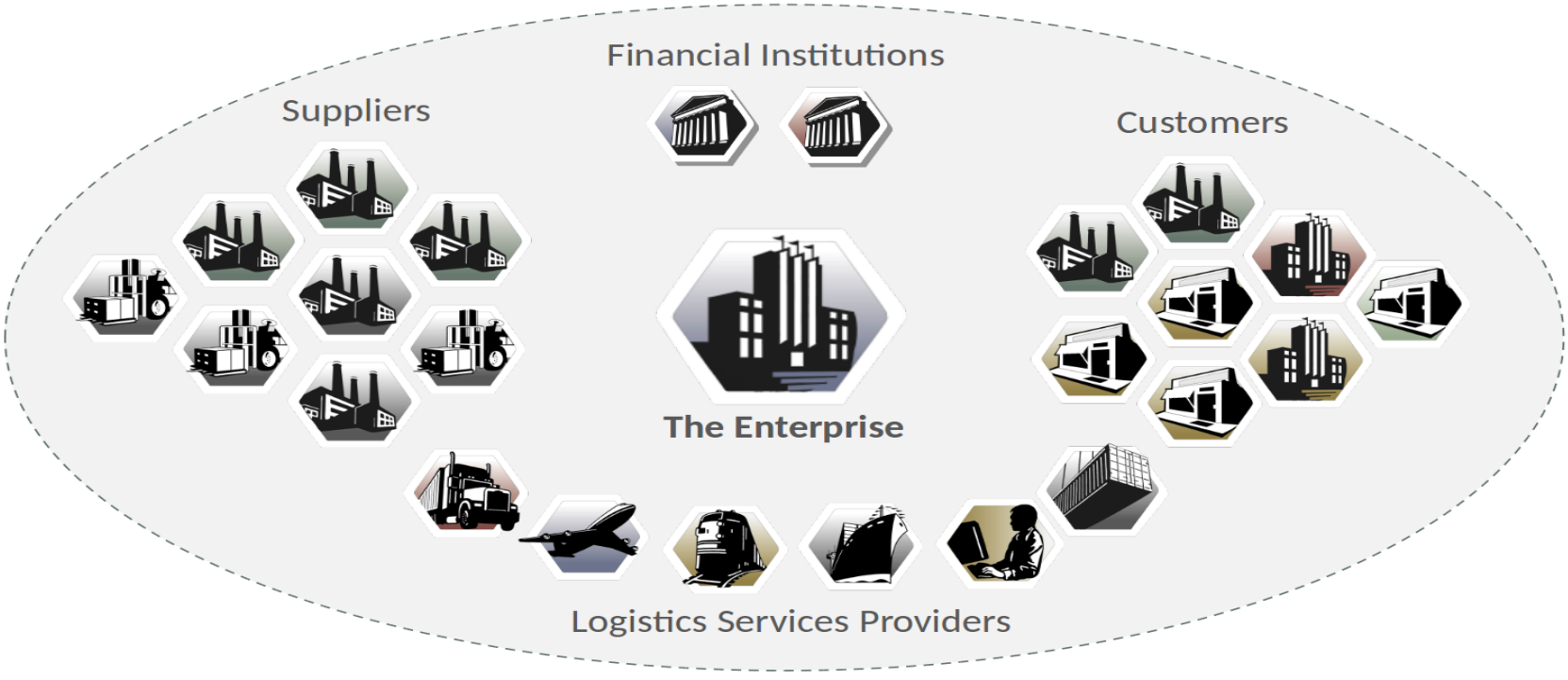
. . . someone compromised the server used to develop new versions of the program. The attacker then used the access to distribute a backdoor that was downloaded more than 900,000 times . . .

The problem may be real, but what to do?

- How to Manage Risks?
- Measure Entire FLOSS Supply Chain
  - (World of Code or WoC)
  - Code to code, change to change, code to projects, code and people
- Develop theoretical frameworks
- Provide data for research and business
- Prior incarnations
  - Within-company context
  - Effective [4, 3]

Definition (What is Supply Chain)

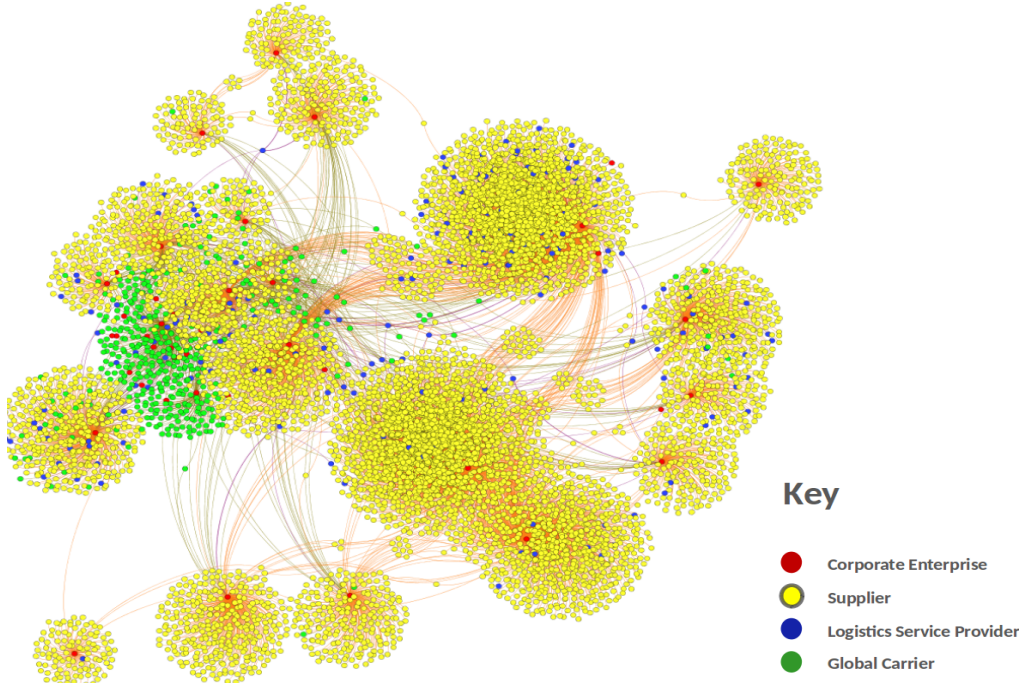
is a set of three or more companies directly linked by one or more of the upstream or downstream flows of products, services, finance, and information from a source to a customer



Graphics courtesy of Diane Palmquist and Greg Kefer, GT Nexus, UT Supply Chain Forum, November 12, 2014

Definition (What is Supply Chain)

is a set of three or more companies directly linked by one or more of the upstream or downstream flows of products, services, finance, and information from a source to a customer



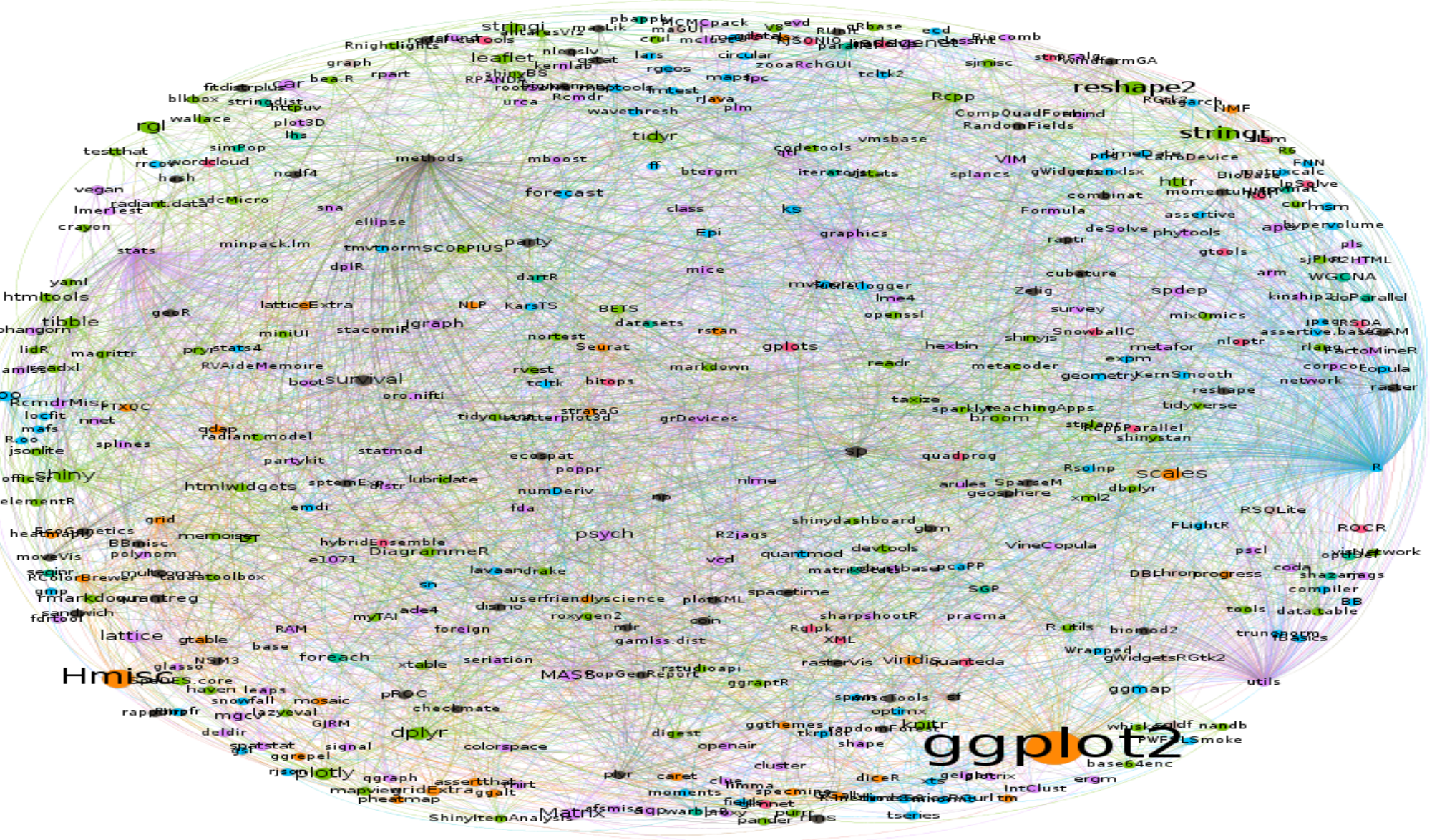
Graphics courtesy of Diane Palmquist and Greg Kefer, GT Nexus, UT Supply Chain Forum, November 12, 2014

Definition (What is Software Supply Chain)

Software Supply Chain (SSC) has

- developers and groups ("companies")
- corporate backers supporting these developers or groups("financing")
- relationships among software projects or packages representing the "chain" of the flow
- changes to the source code (e.g., to files, modules, frameworks, or entire distributions) representing products or information (that flow according to the dependency graph)

Example SSC: CRAN


ggplot2
reshape2
stringr
Hmisc
lattice
dplyr
plotly
Matrix
leaflet
shiny
tidyr
rgl
psych
survival
scales
ggmap
foreach
forecast
methods
mboost
party
Rcpp
RcppArmadillo
RcppEigen
RcppParallel
RcppGSL
RcppProgress
RcppTOML
RcppRoll
RcppSMC
RcppTN
RcppZiggurat
RcppClassic
RcppCNPy
RcppDE
RcppEnsmallen
RcppExamples
RcppFaddeeva
RcppGetconf
RcppGSL
RcppHNSW
RcppMeCab
RcppMLPACK
RcppNumerical
RcppOctave
RcppParallel
RcppProgress
RcppRedis
RcppRoll
RcppSMC
RcppTN
RcppTOML

Visibility

Visibility

Information that developers have about the inputs, processes, sources and practices used to bring the product to consumers/market. This includes complete supply chain visibility including traceability for the entire supply chain. Visibility is, generally, inwardly/developer focused.



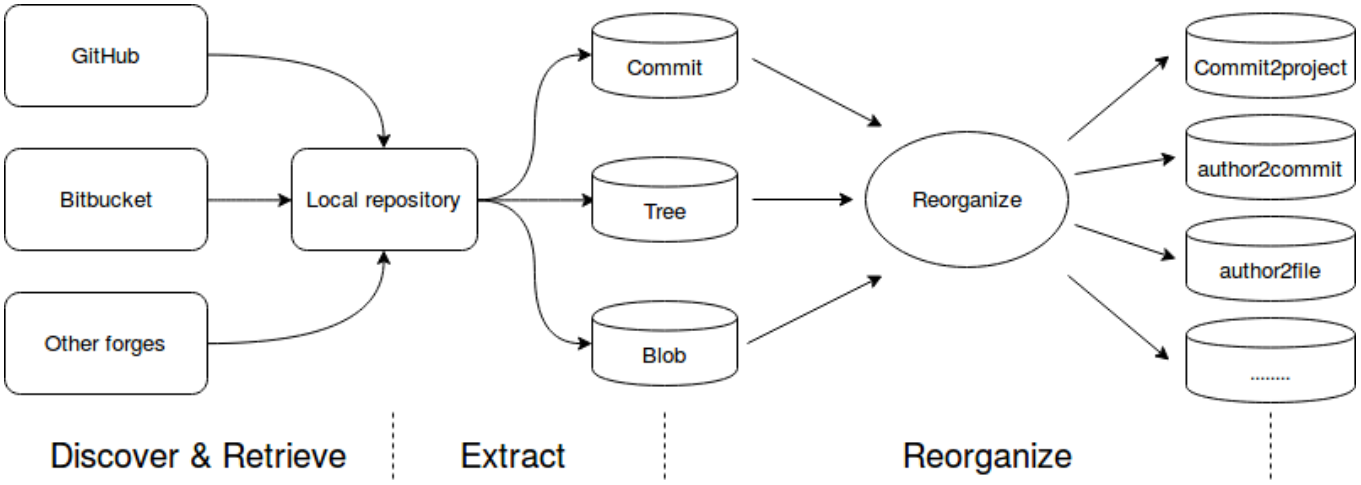
Transparency

Information that developers share with their consumers about the inputs, processes, sources and practices used to bring the product to the consumer. It is more outwardly focused/from the consumer perspective than visibility.



Skeptic: How would you get it all?

- Discover, Retrieve, Store, Analyze, Update [7, 6]



- 73+M projects; 34M Authors; 1.7B Commits; 7B Blobs and Trees
- Simply cloning: $\geq$\$1.8PB
- Specialized database without redundancies: 120TB

Skeptic: measurement may be good, but OSS data are rather bad

Agreed:, e.g. most "productive" authors

Number of commits	Author
10960000	one-million-repo < *mikigal.acc@gmail.com* >
4400778	datakit < *datakit@docker.com* >
2463758	greenkeeper[bot] < *greenkeeper[bot]@users.noreply.github.com* >
2063212	Auto Pilot < *noreply@localhost* >
1864730	Your Name < *you@example.com* >

Silly competitions

- Fake commits with everyone's name to have most contributors

Authors	Repo
389993	cirosantilli/imagine-all-the-people

- The longest chain of commits

Length of commit chain	Repo
9959999	github.com/one-million-repo/biggest-repo-ever

See more examples at bitbucket.org/swsc/overview/fun/

Data That Needs to be Corrected/Augmented

- Author IDs
- Repository forks
- Code dependencies (dozens of languages)
- Project types
- Link to external data sources
- Many other challenges

Example: Identifying Authors

- We have 34M strings
- Are these two the same person?
  - Aaron Lee < aaron.lee@rackspace.com>
  - Aaron Lee < wwkeyboard@gmail.com>
- Text similarity (adjusted for common names)
- Behavioral fingerprints:
  - Similarity of written text (Doc2Vec embedding)
  - Change the same source code files
  - Work in the similar time-zones
- Trained machine learning: 99.99 accuracy

Lesson: Need a Community of/to Support

- Expert users and curators
- Ability to share augmented/corrected data
- Automated curation tools (supervised by experts)

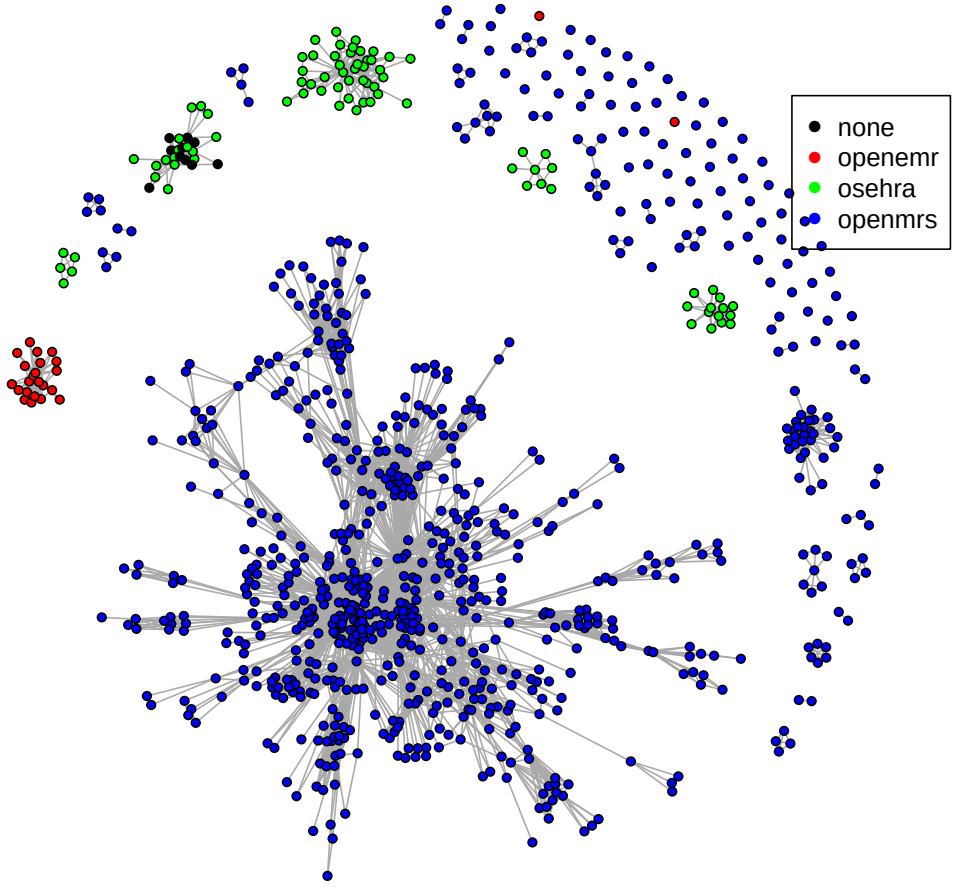
Skeptic: it may work, but why would I use it?

- Its insane to think that economist could do without
  - CENSUS
  - Labor data
  - Business surveys
- . . . epidemiologist could do without
  - Disease reporting infrastructure
- Perhaps the relative scarcity of fundamental theories in SE is caused by
  - the lack of global measurement/reporting infrastructure?

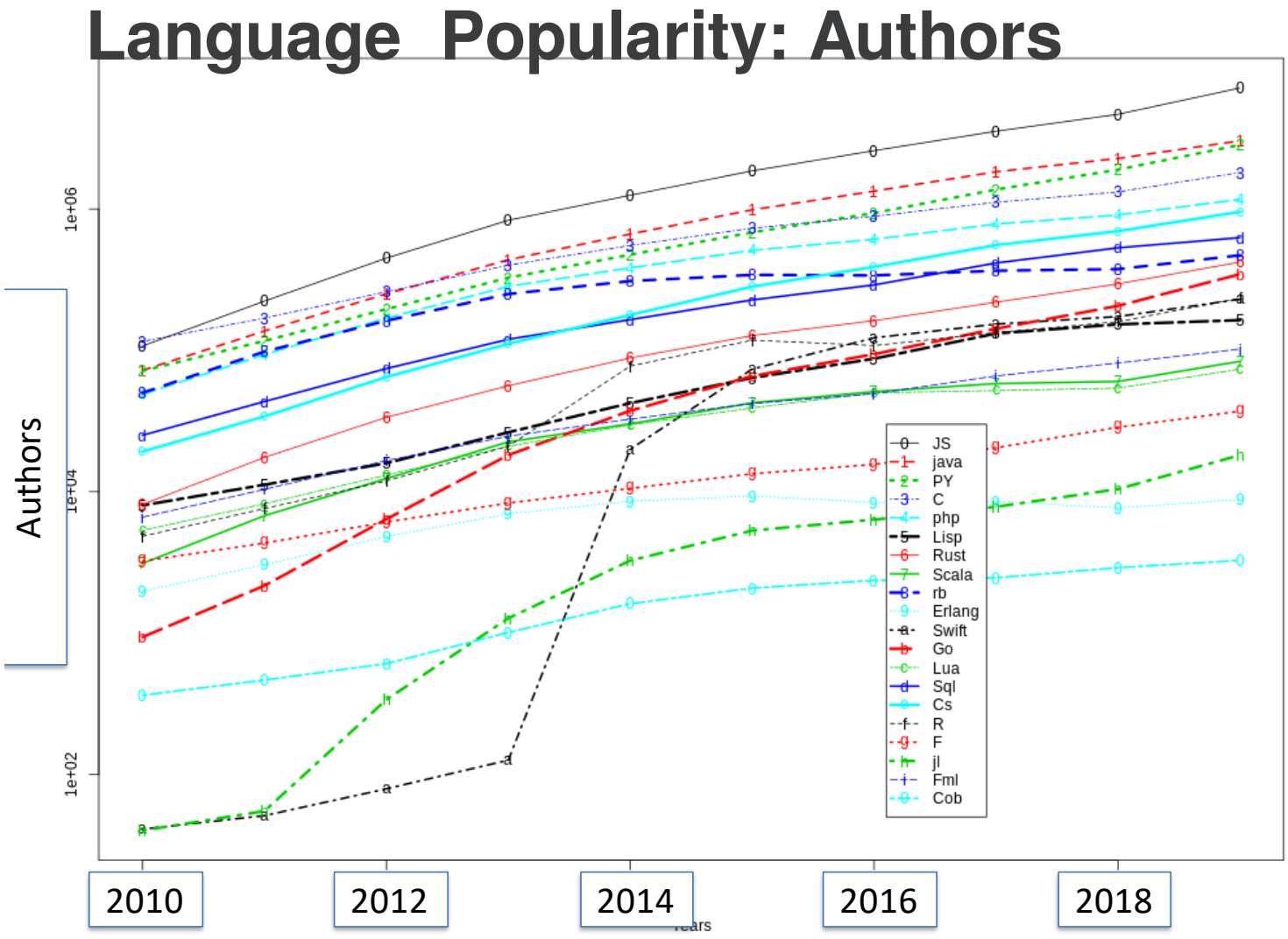
Skeptic: again, why would I use it?

- Eight implemented use cases
  - Relationships: code flow, technical and tool dependencies, knowledge flow [1, 2]
  - Impact measurement [5]
  - Models of Spread: Technologies/Practices/Malware
  - Models of growth/decay
  - Community identification
  - Educational outcomes
  - Natural experiments
- Unlimited data to train machine learning models
  - Code
  - Documentation
  - Change activity

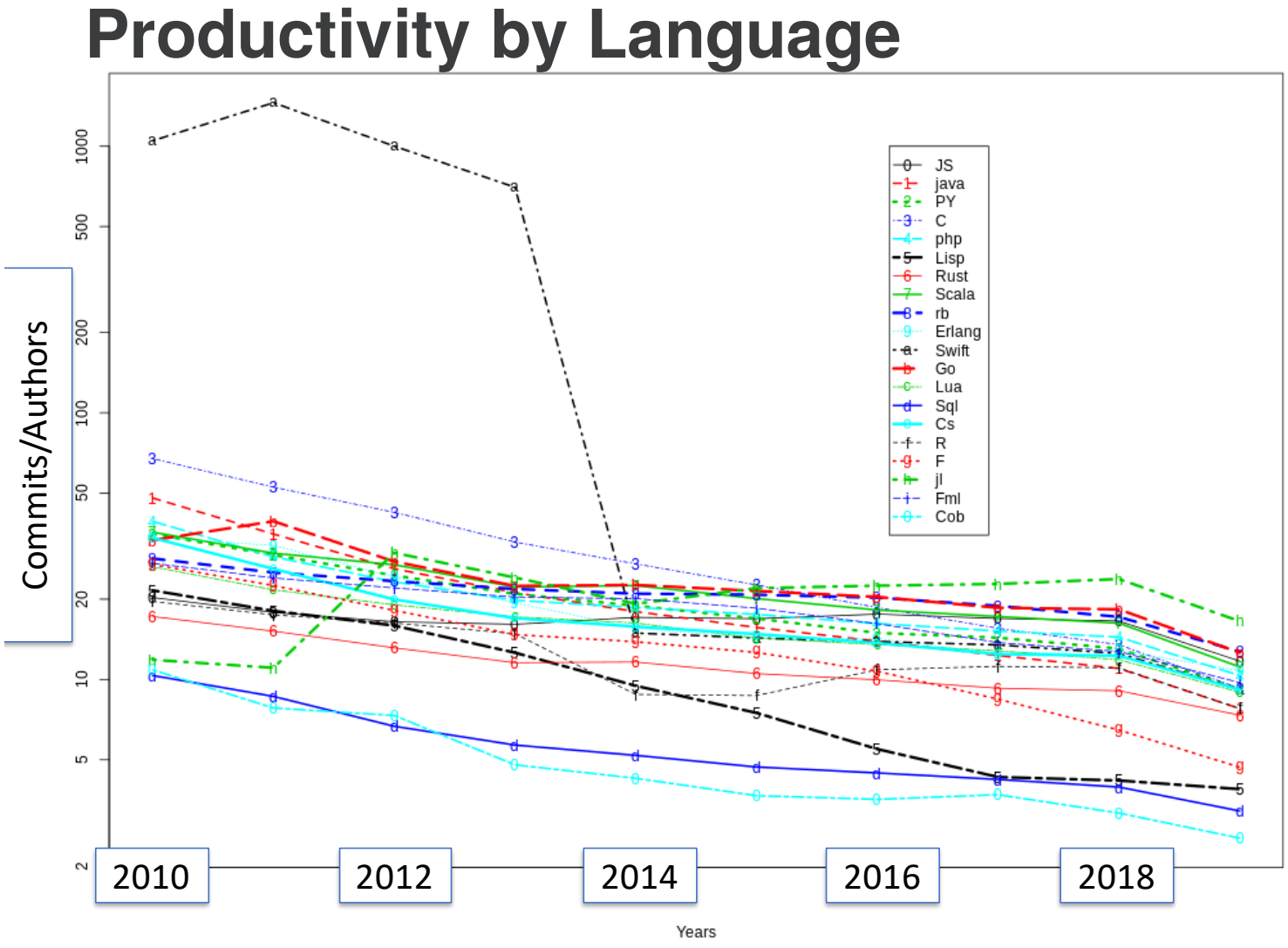
Communities for Health Records Software



What Languages are Popular?



How Difficult Each Language is?



Can we predict which technology will prosper/fail?

- Two data-science technologies: abstraction of data.frame in R
  - tidy vs data.table
- Sample: all R-language files in public repositories
  - 1.5M files, 5M versions
- When was the first time tidy or data.table included?
  - 17,536 projects use data.table
  - 7,032 projects use tidy

Results (Choice and Decision Maker)

- Choice
  - Exposure: Recent and Cumulative deployments, Mentions on StackOverflow
  - Qualities: Activity, developers,responsiveness, open issues
- Decision maker
  - Activity, developers
  - Performance needs
  - Social network
  - Technical network

Note: Red- negative, Green- positive

Skeptic (turned into believer): ok ok, how do I sign up for this?

- bitbucket.org/swsc/overview:
  - related publications / data / analysis
- Hackathon Nov 1-3
  - woc-hack.slack.com : sign up at `http://bit.ly/WoC-Hack`
  - github.com/woc-hack/tutorial
  - Signup form `http://bit.ly/WoC-Signup`

References



Tapajit Dey, Yuxing Ma, and Audris Mockus.

Patterns of effort contribution and demand and user classification based on participation patterns in npm ecosystem.
In *Proceedings of the 15th International Conference on Predictive Models and Data Analytics in Software Engineering*. ACM, 2019.



Tapajit Dey and Audris Mockus.

Are software dependency supply chain metrics useful in predicting change of popularity of npm packages?
In *Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering*, pages 66–69. ACM, 2018.



Randy Hackbarth, Audris Mockus, John Palframan, and David Weiss.

Assessing the state of software in a large enterprise.
Journal of Empirical Software Engineering, 10(3):219–249, 2010.



Randy Hackbarth, Audris Mockus, John Palframan, and David Weiss.

Assessing the state of software in a large enterprise: A 12-year retrospective.
In *The Art and Science of Analyzing Software Data*, pages 411–451. Elsevier, 2016.



Andrey Karnauch, Sadika Amreen, and Audris Mockus.

Developer reputation estimator (dre).
In *ASE'19*, 2019.



Yuxing Ma, Chris Bogart, Sadika Amreen, Russell Zaretzki, and Audris Mockus.

World of code: An infrastructure for mining the universe of open source vcs data.
In *IEEE Working Conference on Mining Software Repositories*, May 26 2019.



Audris Mockus.

Amassing and indexing a large sample of version control systems: towards the census of public source code history.
In *6th IEEE Working Conference on Mining Software Repositories*, May 16–17 2009.