Network Analysis:

The Hidden Structures behind the Webs We Weave

17-338 / 17-668

Network Analysis of Open Source Software

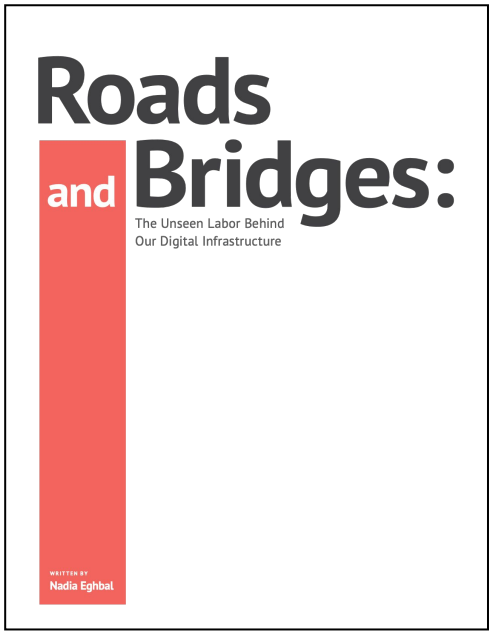
Tuesday, October 22, 2024

Patrick Park & Bogdan Vasilescu

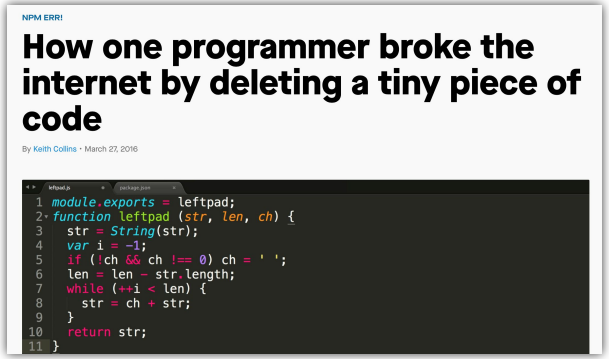
Carnegie Mellon University
School of Computer Science

S3D
Software and Societal Systems Department

Open Source as digital infrastructure:
Needs regular upkeep and maintenance



- Everybody uses open source code:
  - Fortune 500 companies
  - major software companies
  - startups
  - government
  - …
- If undermaintained:
  - Risks for downstream users
  - Slows down innovation
  - …



Creating sustainable open source communities is hard

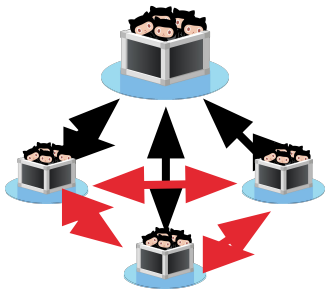
In some ways harder today than ever before … because of how open source has changed

Today: more problems than solutions

How has open source changed?

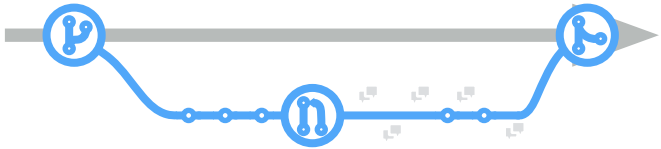
Change #1: GitHub standardized the practices

- Git version control



- GitHub UI

- The Pull Request model



- Lower barrier to entry
- Easier to contribute

→ More production

Change #2: More open source now than ever before

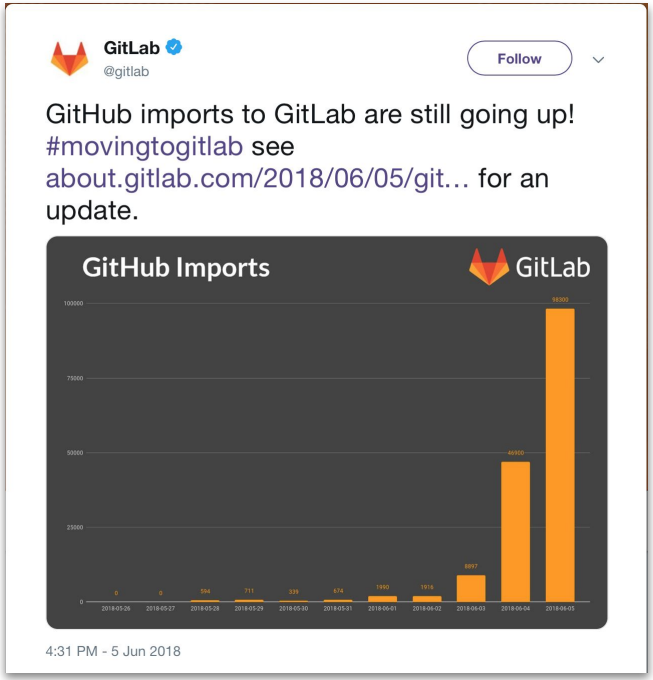
- Explosion of production in the past seven years

github SOCIAL CODING

Bitbucket

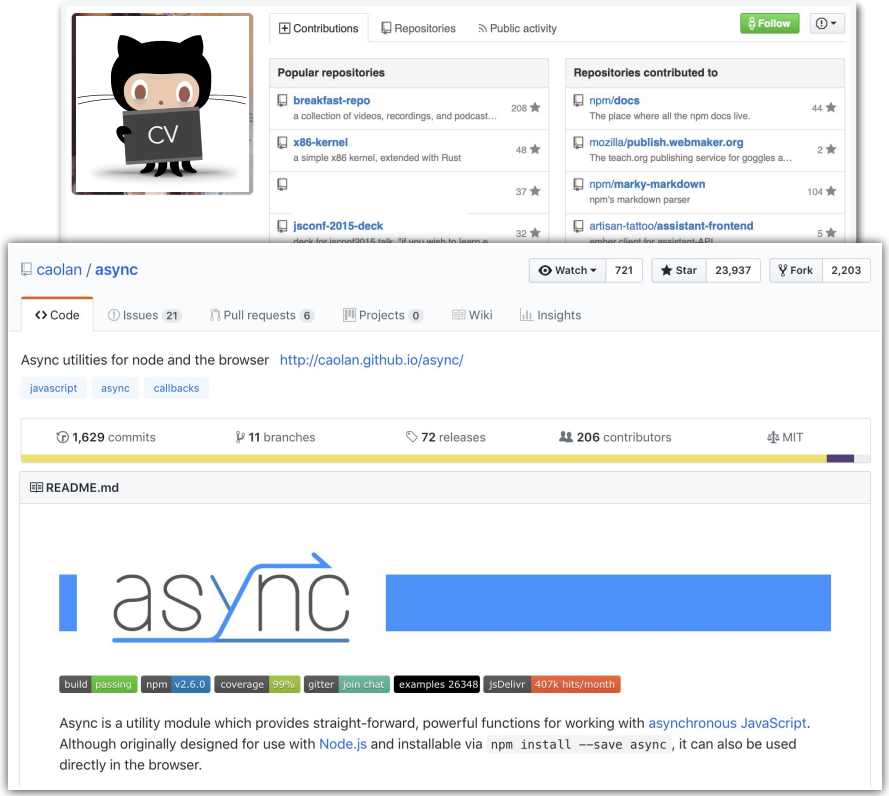
6 million users

(March 2019)

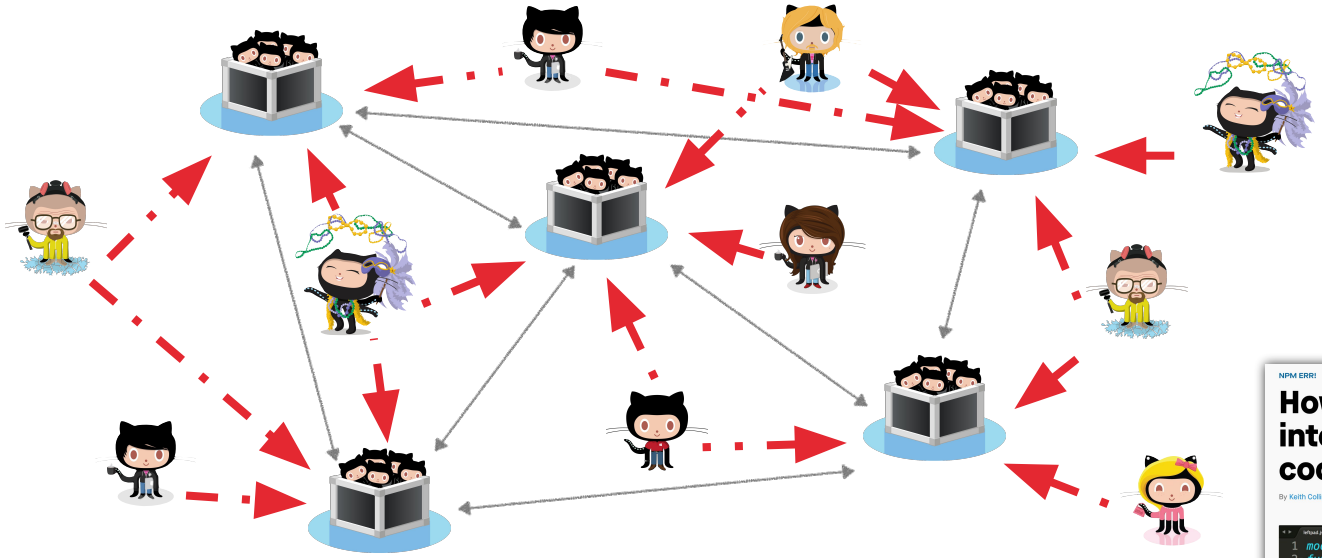


Change #3: High level of transparency

- Profile pages for users and projects
- Rich inferences about people's expertise and level of commitment
- Impacts collaboration, but also recruiting and hiring
  - (Dabbish et al. 2012), (Marlow et al. 2013), (Marlow and Dabbish 2013)

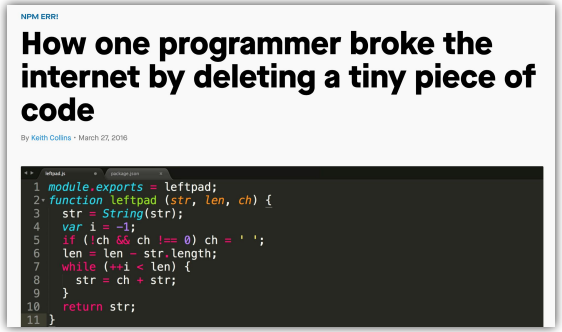


Change #4: Complex socio-technical ecosystems



Interconnections between people and projects

Can be brittle



Change #5: Increasing commercialization and professionalization

- Historically
  - Mostly community-based projects (Python, RubyGems, Twisted)

- Currently
  - Lots of commercial involvement
    - Companies (Go - Google, React - Facebook, Swift - Apple)
    - Startups (Docker, npm, Meteor)

- 23% of respondents to 2017 GitHub survey: job duties include contributing to open source

Change #6: High expectations toward the quality, reliability, and security of open source infrastructure

- Equifax (market cap $14 billion) built products on top of open-source infrastructure,[11] including Apache Struts
- Equifax did not make any contributions to open source projects
- A flaw in Apache Struts contributed to the breach (CVE-2017-5638)
- Equifax publicly blamed (with national news coverage) Apache Struts for the breach

Equifax confirms Apache Struts security flaw it failed to patch is to blame for hack

The company said the March vulnerability was exploited by hackers.

By Zack Whittaker | September 14, 2017 -- 01:27 GMT (18:27 PDT) | Topic: Security

https://www.zdnet.com/article/equifax-confirms-apache-struts-flaw-it-failed-to-patch-was-to-blame-for-data-breach/

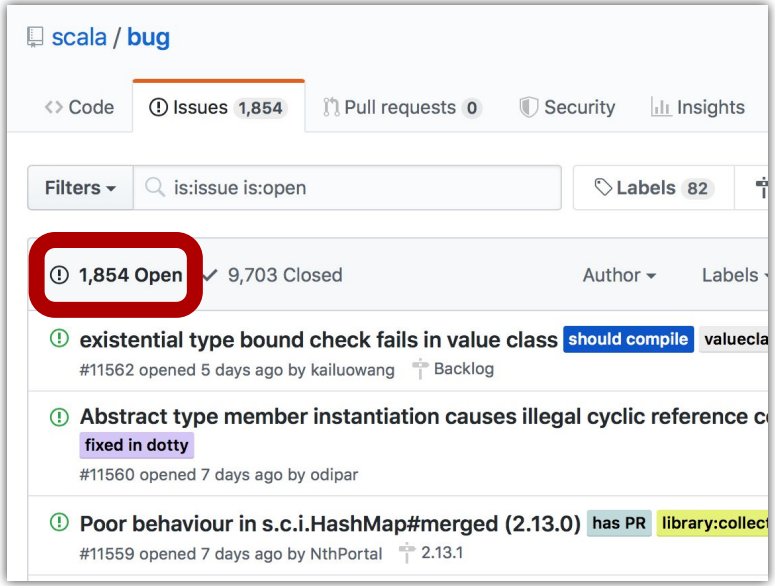
Change #7: High level of demands & stress

- Easy to report issues / submit PRs
  - Growing volume of requests

12

- Social pressure to respond quickly
  - Otherwise, off-putting to newcomers (Steinmacher et al. 2015)

- Entitlement, unreasonable requests from users:
  - *"I have been waiting 2 years for Angular to track the 'progress' event and it still can't get it right?!?!"*
  - *"Thank you for your ever useless explanations."*
  - …



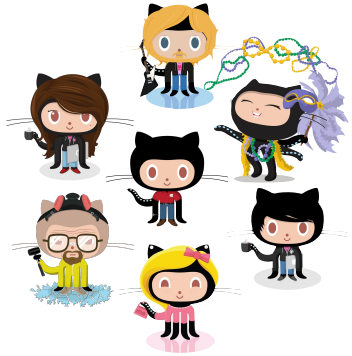
Change #8: Low demographic diversity

- Gender representation reality

stackoverflow github SOCIAL CODING

open source 10.9% Google 18% 16.6%

- Expectation



"*More about the* contributions to the *code* than the 'characteristics' of the person"

"*Any demographic identity is irrelevant*"

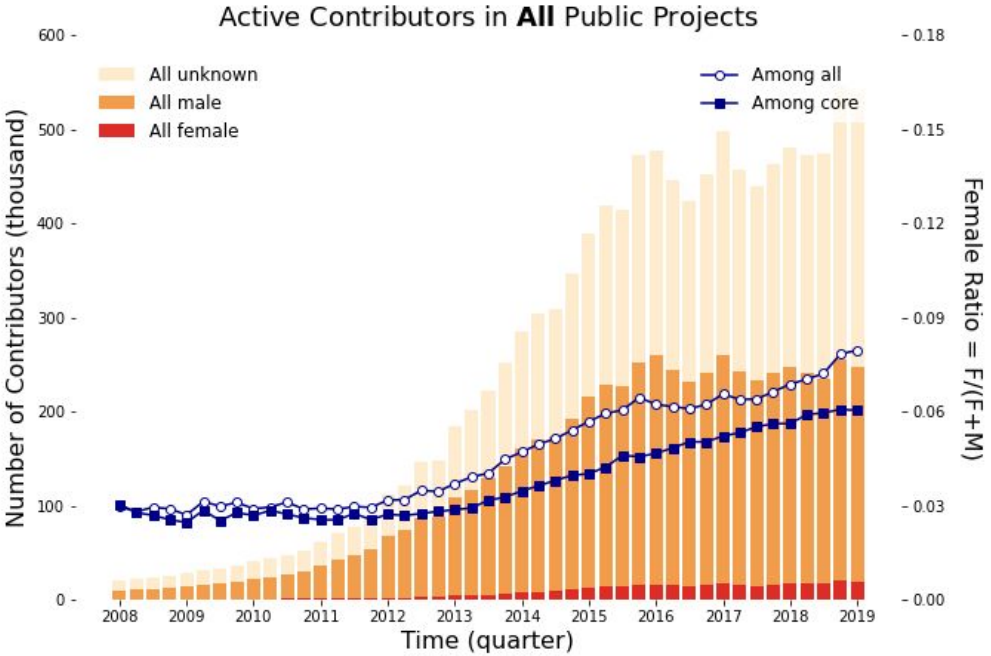
"*Code sees no color or gender*"

- Perceptions of Diversity on GitHub: A User Survey. Vasilescu, B., Filkov, V., and Serebrenik, A. *CHASE 2015*

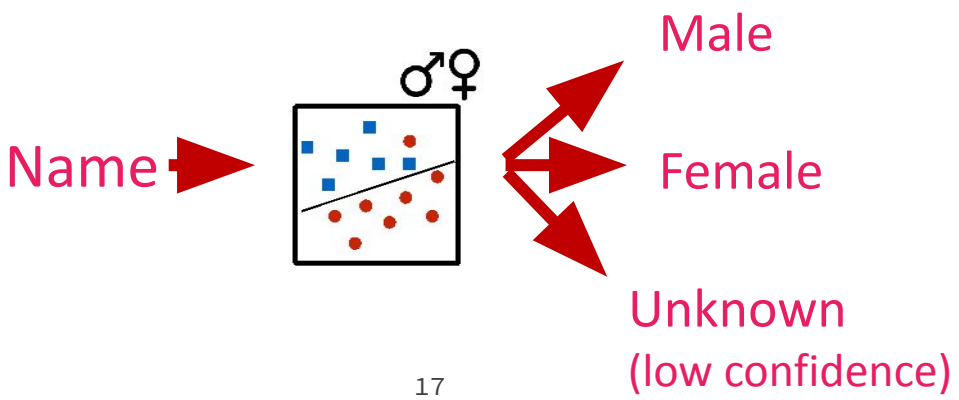
“Going farther together: The impact of social capital on sustained participation in open source”
Qiu* et al, ICSE 2019

* Thanks to Sophie Qiu for slides

Skewed gender ratio: more than 90% of the OSS population is male



Research scope - binary gender, GitHub

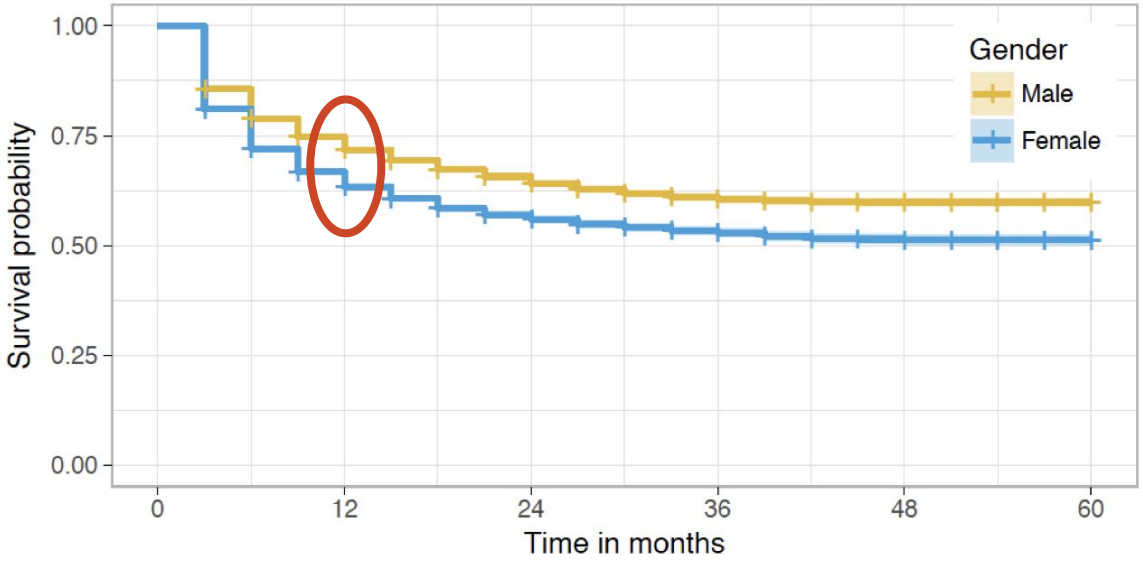


Gender diversity = Women + Men

A simplifying assumption: gender is binary

On GitHub, women disengage earlier than men

After one year ca. 70% of men are still active but only ca. 60% of women



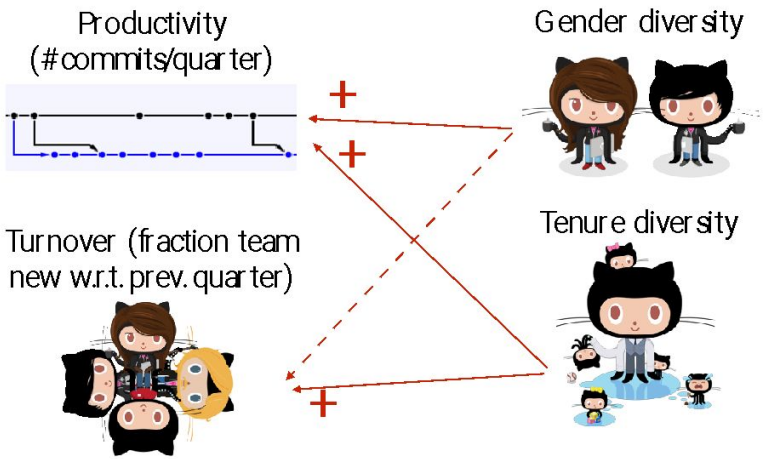
Low gender diversity as a challenge to OSS sustainability: limits contributor pool

https://w3techs.com/technologies/history_overview/web_server

(Greenstein and Nagel, 2016)

Low gender diversity as a challenge to OSS sustainability: harms project success

Results



[Vasilescu et al., 2015]

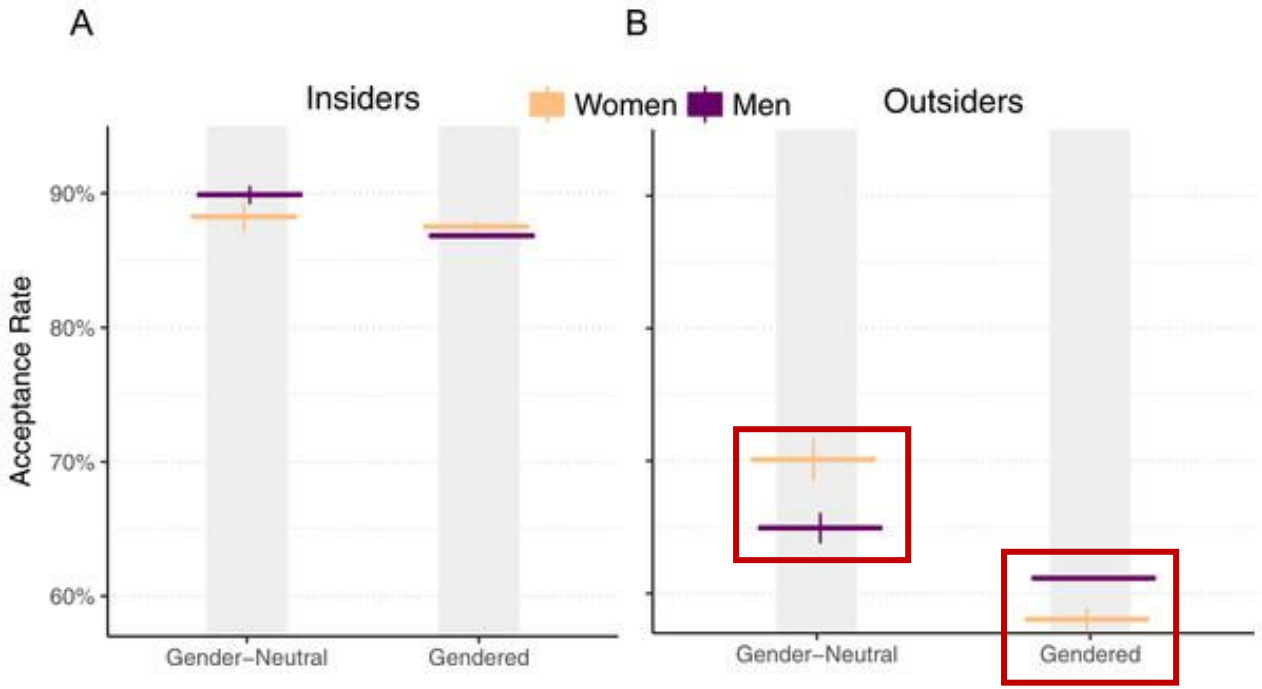
Low gender diversity as a challenge to OSS sustainability: limits opportunities

Employers (and job seekers) use open-source experience to make inferences (or form impressions) about a candidate's technical skills.

(Marlow et al., 2013)

https://codeformore.com/how-to-write-up-open-source-experience-when-you-dont-have-any/

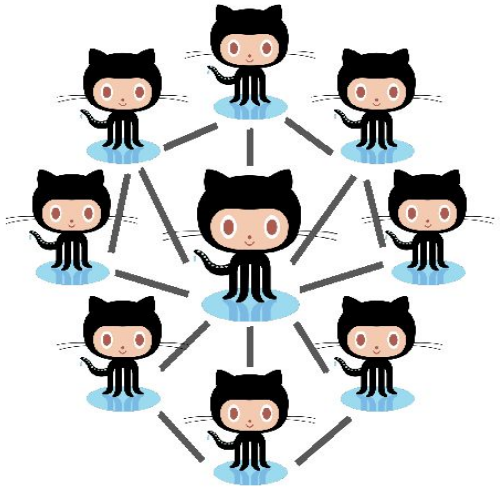
Minorities face bias and discrimination.



[Terrell et al., 2017]

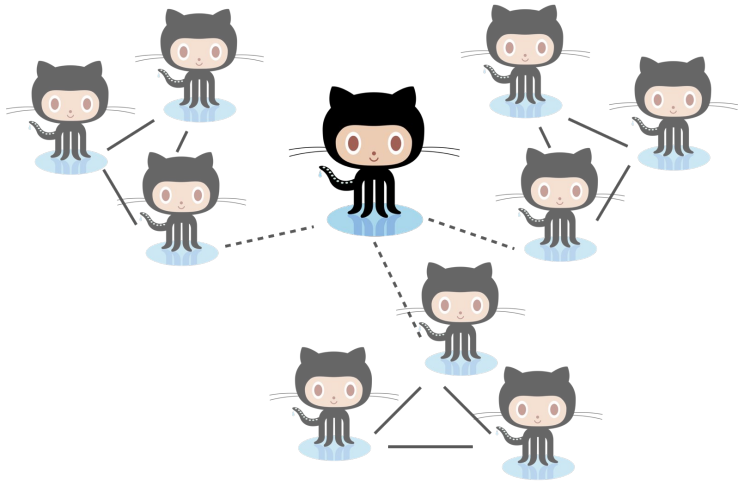
Social capital theory for sustained participation

Bonding social capital:
benefiting from strongly connected network



Willingness to continue
(Coleman, 1990)

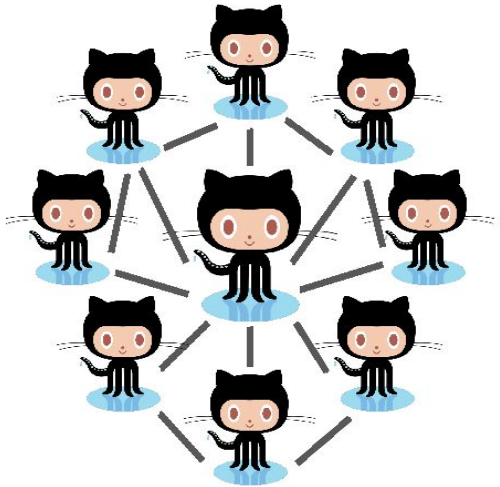
Bridging social capital:
benefiting from network with diverse info



Opportunity to continue
(Burt, 1998, 2001)

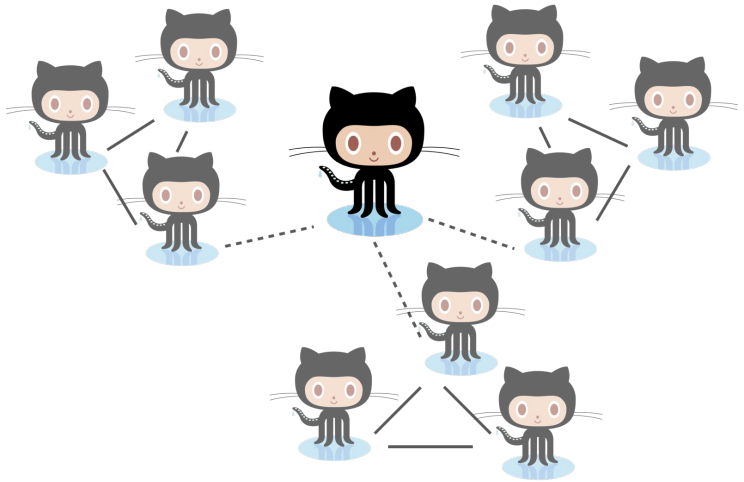
H1: more social capital ~ more prolonged engagement

Bonding social capital:
benefiting from strongly connected network



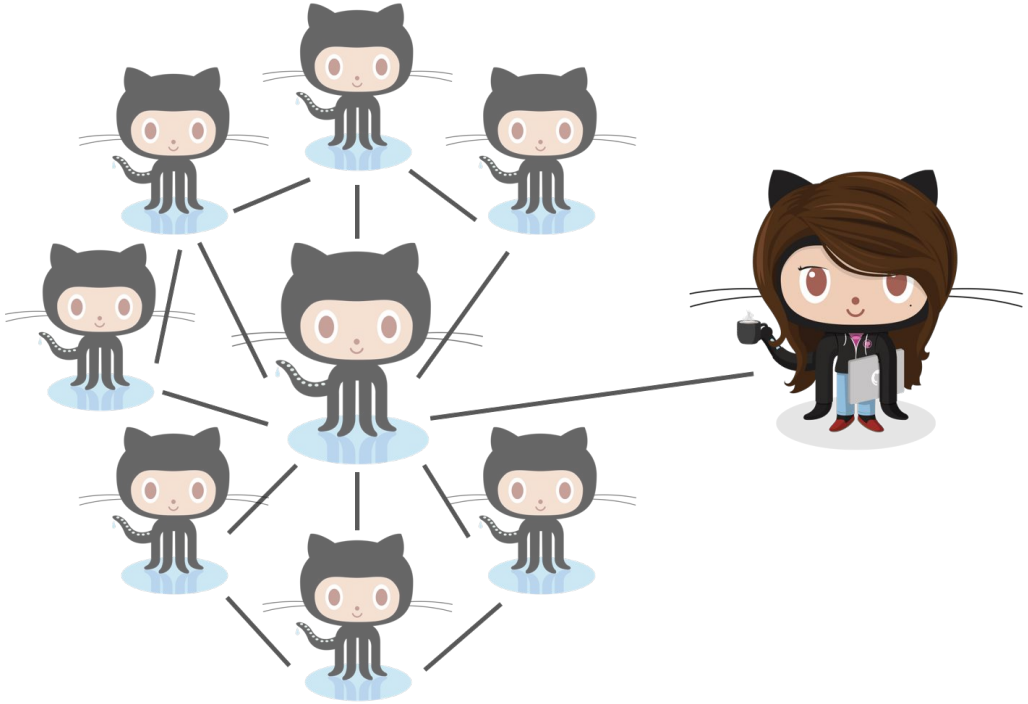
Willingness to continue
(Coleman, 1990)

Bridging social capital:
benefiting from network with diverse info



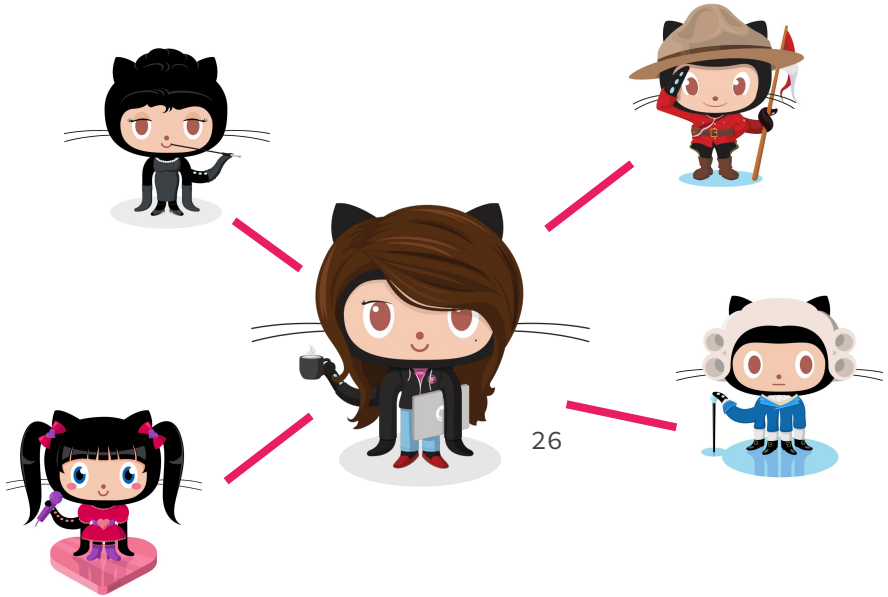
Opportunity to continue
(Burt, 1998, 2001)

Cohesive network might foster discrimination and exclusion

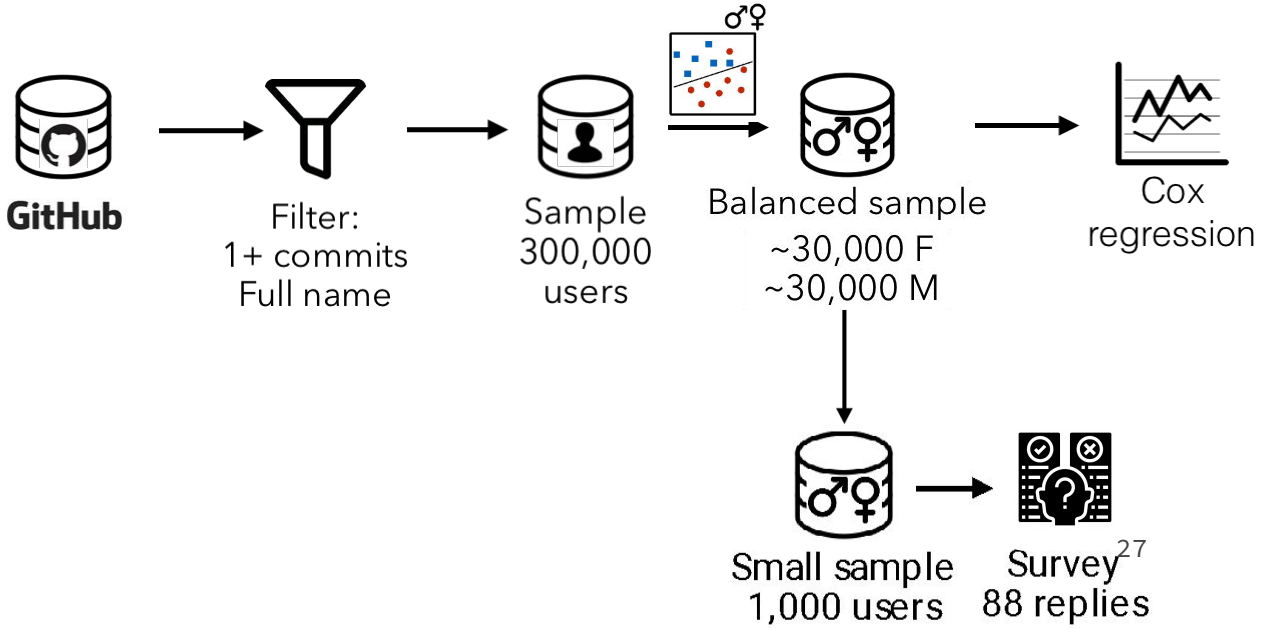


H2: Teams with more diverse information ~ more prolonged engagement, esp. for women

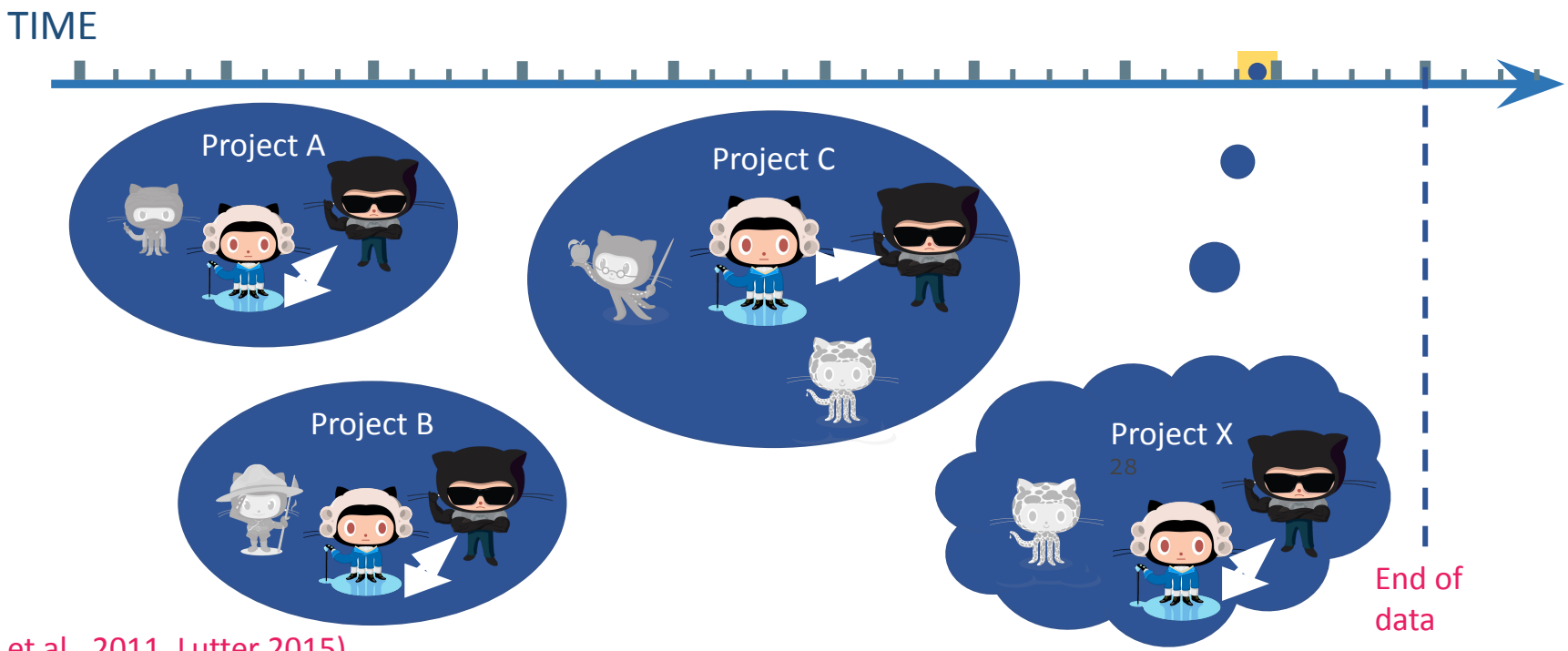
Information diversity should reduce the risk of demographic-based echo chambers.



Large-scale mixed-methods study

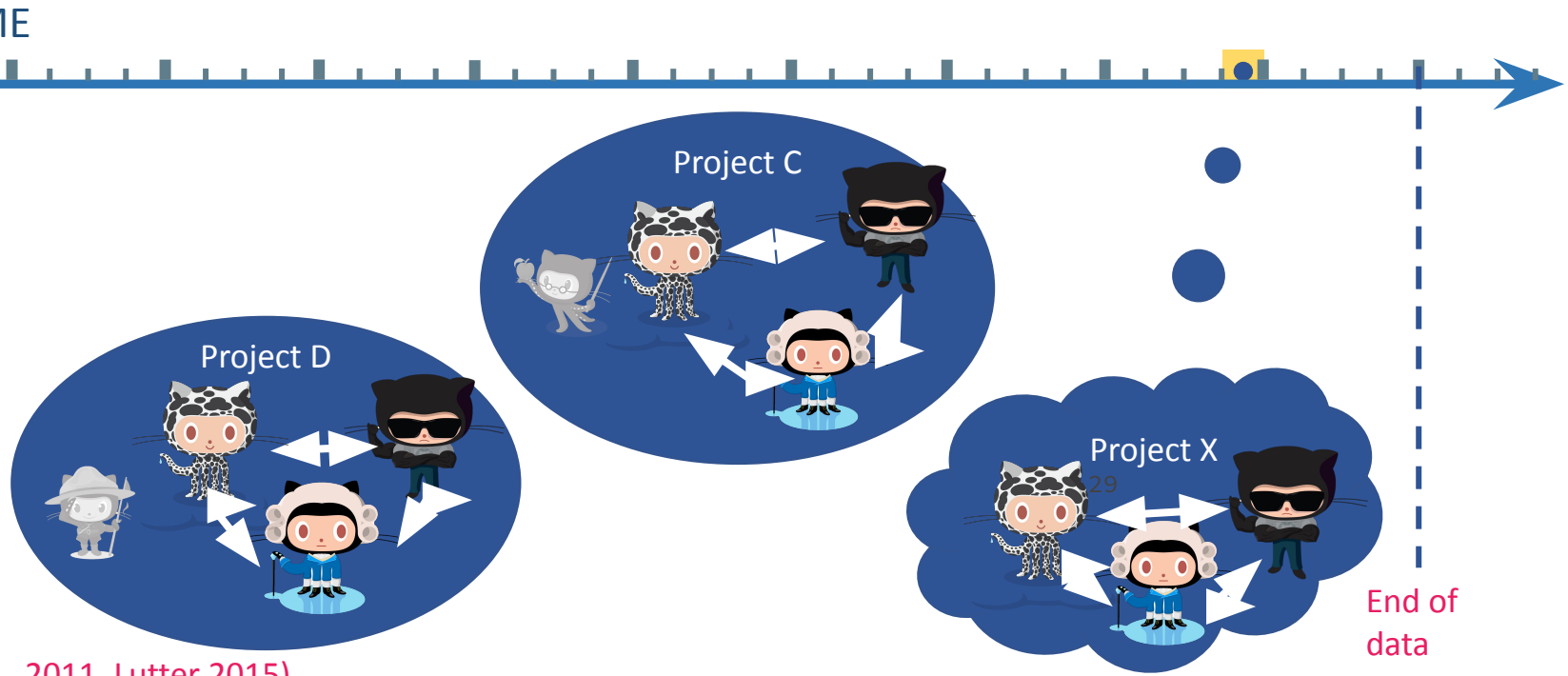


Bonding social capital – Team Familiarity



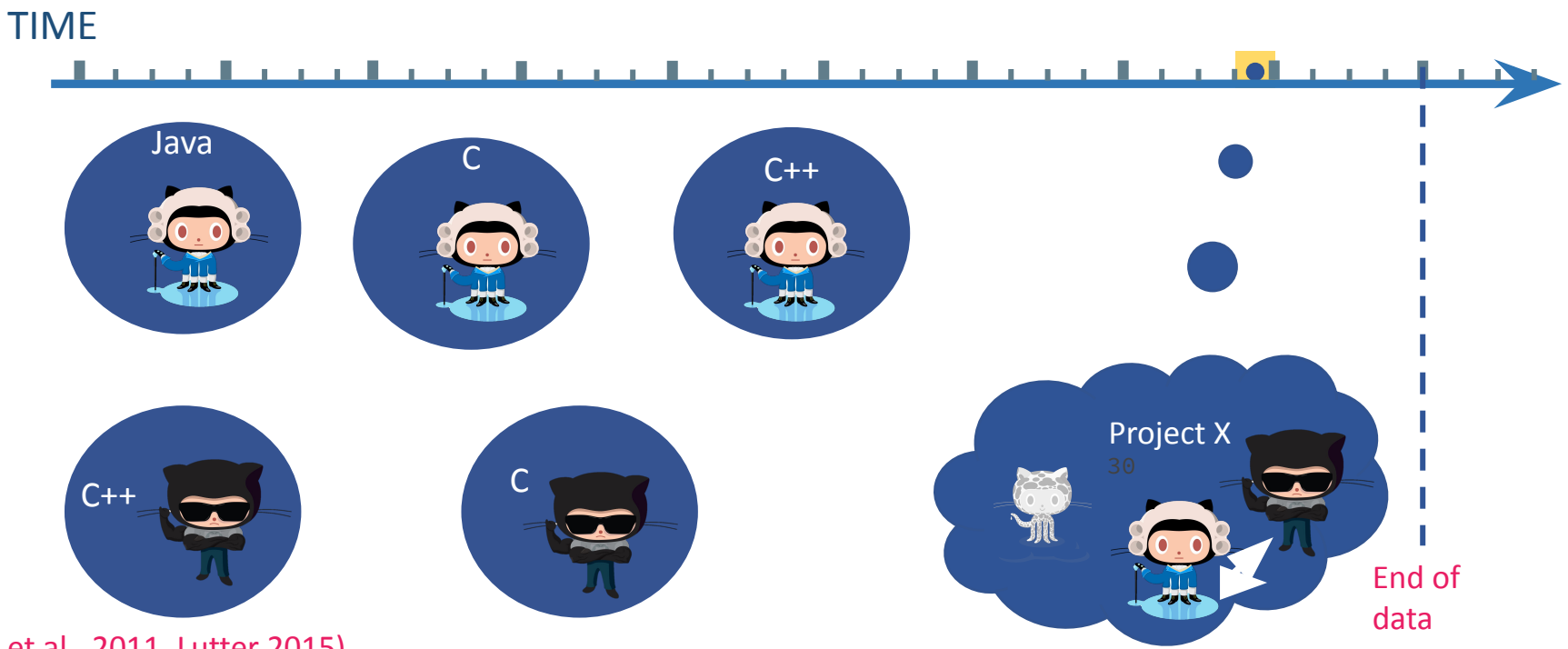
(de Vaan et al., 2011, Lutter 2015)

Bonding social capital – Recurring Cohesion



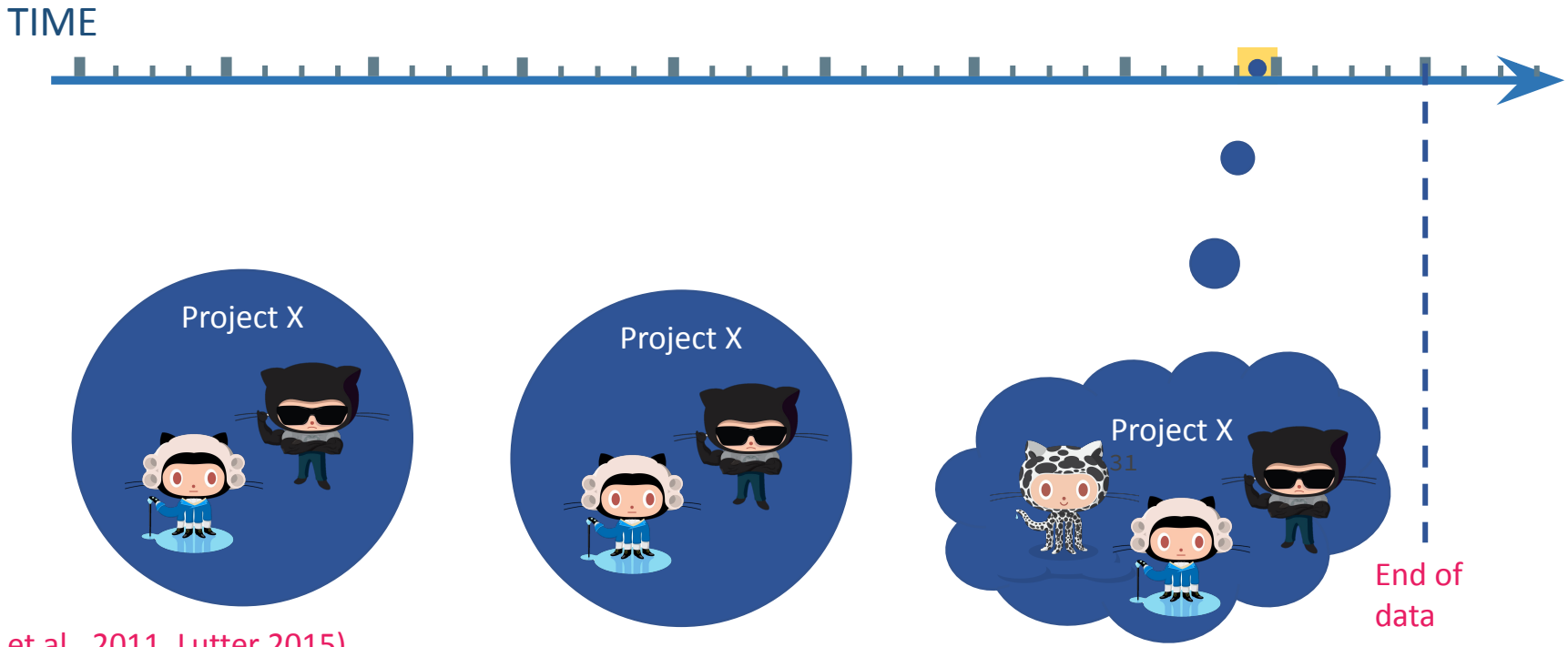
(de Vaan et al., 2011, Lutter 2015)

Bridging social capital – Language Diversity



(de Vaan et al., 2011, Lutter 2015)

Bridging social capital – Share of Newcomers

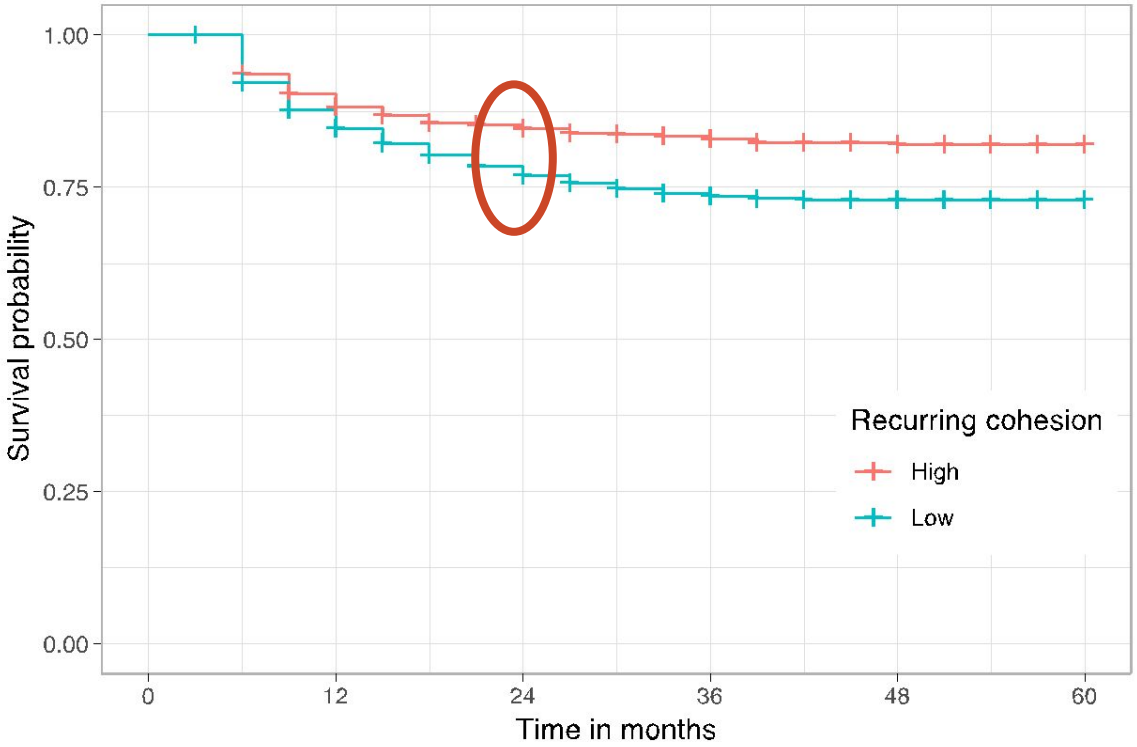


(de Vaan et al., 2011, Lutter 2015)

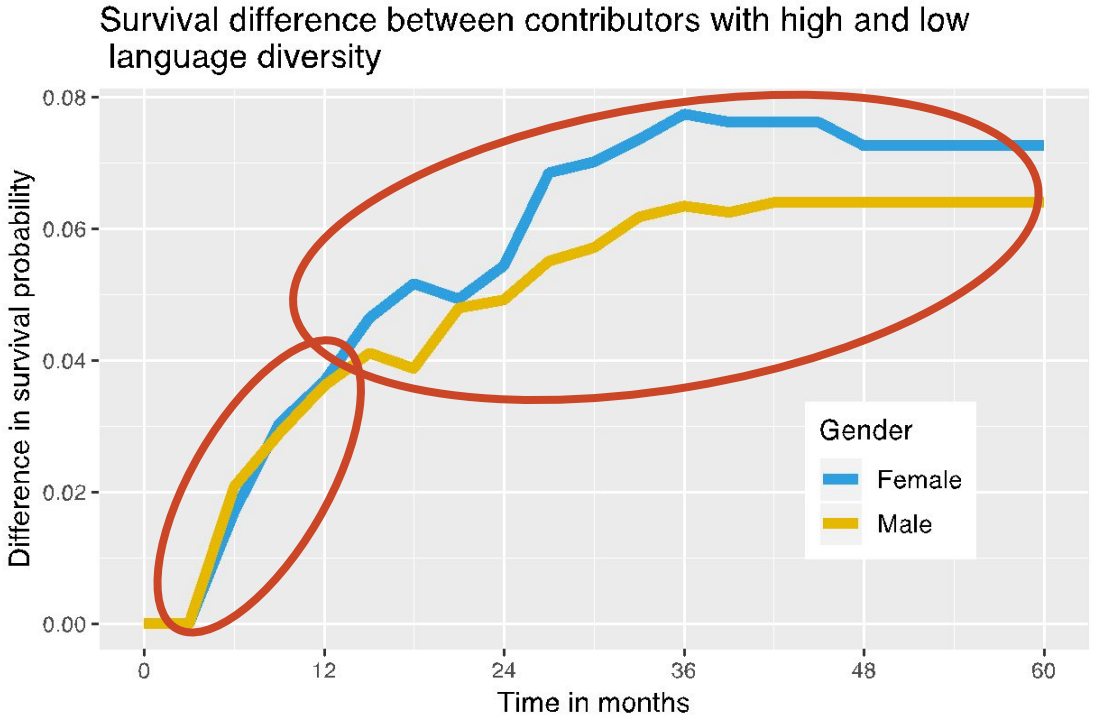
COX regression model

Contributor	Time	Active	Social capital	Control variables
	2008 Jan – Mar	True	Team familiarity Recurring cohesion Language diversity Share of newcomers	Project size Project owner ……
	2008 Jan – Mar	True	Team familiarity Recurring cohesion Language diversity Share of newcomers	Project size Project owner ……
	2009 Apr – Jun	False	Team familiarity Recurring cohesion Language diversity Share of newcomers	Project Size Not project owner ……

H1: more social capital ~ more prolonged engagement



H2: Language diversity interacts with gender



Innovation and the strength of weak ties

Open-source software development is an avenue for innovation and creative expression.

(Lakhani & Wolf, 2005)

"How creative a person feels when working on the project is the strongest and most pervasive driver [of participation in open source]"

"Free software is directly responsible for today's current startup renaissance."

(Eghbal, 2016)

How to define innovation in software?

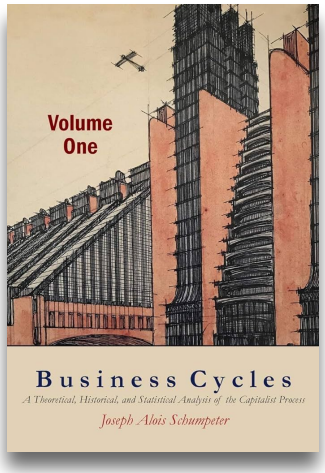
How to measure it?

How does innovation emerge?

What are its consequences?

DALL·E 3 - “An old-looking map with uncharted territory, here be dragons

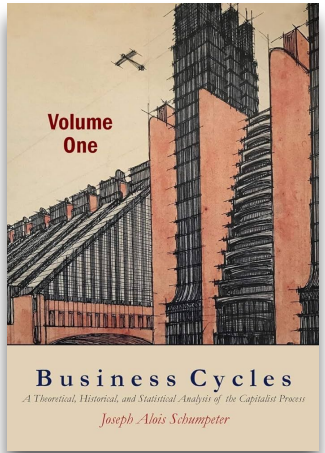
Key idea: Innovation as novel recombination



"[We may say] that innovation combines factors in a new way, or that it consists in carrying out new combinations."

(Schumpeter, 1939)

Key idea: Innovation as novel recombination



(Schumpeter, 1939)

"[We may say] that innovation combines factors in a new way, or that it consists in carrying out new combinations."

"… how scientists search for ideas is premised in part on the idea that teams can span scientific specialties, effectively combining knowledge that prompts scientific breakthroughs."

Atypical Combinations and Scientific Impact

Brian Uzzi,[1,2] Satyam Mukherjee,[1,2] Michael Stringer,[2,3] Ben Jones[1,4]*

Novelty is an essential feature of creative ideas, yet the building blocks of new ideas are often embodied in existing knowledge. From this perspective, balancing atypical knowledge with conventional knowledge may be critical to the link between innovativeness and impact. Our analysis of 17.9 million papers spanning all scientific fields suggests that science follows a nearly universal pattern: The highest-impact science is primarily grounded in exceptionally conventional combinations of prior work yet simultaneously features an intrusion of unusual combinations. Papers of this type were twice as likely to be highly cited works. Novel combinations of prior work are rare, yet teams are 37.7% more likely than solo authors to insert novel combinations into familiar knowledge domains.

Scientific enterprises are increasingly concerned that research within narrow boundaries is unlikely to be the source of the most fruitful ideas (*1*). Models of creativity emphasize that innovation is spurred through original combinations that spark new insights (*2–10*). Current interest in team science and how scientists search for ideas is premised in part on the idea that teams can span scientific specialties, effectively combining knowledge that prompts scientific breakthroughs (*11–15*).

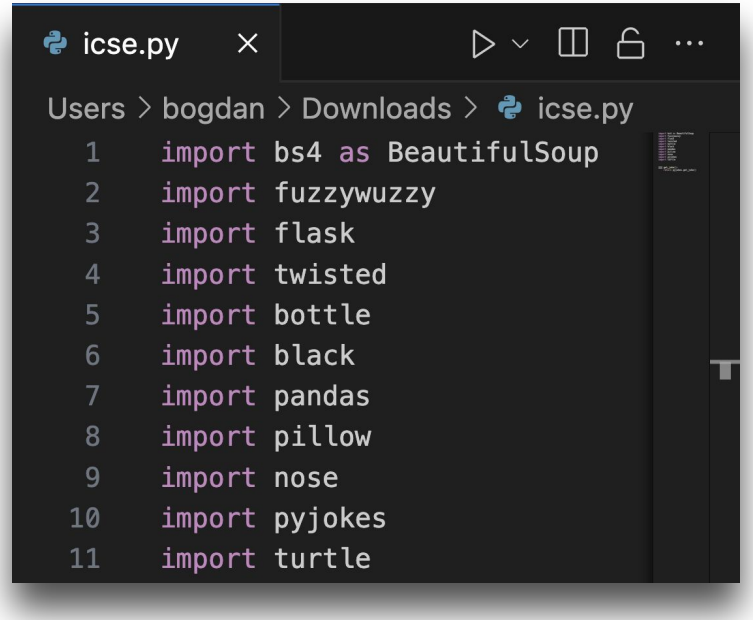
Yet the production and consumption of boundary-spanning ideas can also raise well-known challenges (*16–21*). If, as Einstein believed (*21*), individual scientists inevitably become narrower in their expertise as the body of scientific knowledge expands, then reaching effectively across boundaries may be increasingly challenging (*4*), especially given the difficulty of searching unfamiliar domains (*17*, *18*). Moreover, novel ideas can be difficult to absorb (*19*) and communicate, leading scientists to intentionally display conventionality. In his *Principia*, Newton presented his laws of gravitation using accepted geometry rather than his newly developed calculus, despite the latter's importance in developing his insights (*22*). Similarly, Darwin devoted the first part of the *Origin of Species* to conventional, well-accepted knowledge about the selective breeding of dogs, cattle, and birds. From this viewpoint, the balance

[1]Kellogg School of Management, Northwestern University, 2001 Sheridan Road, Evanston, IL 60208, USA. [2]Northwestern Institute on Complex Systems, Northwestern University, 600 Foster, Evanston, IL 60208, USA. [3]Datascope Analytics, 180 West Adams Street, Chicago, IL 60603, USA. [4]National Bureau of Economic Research, 1050 Massachusetts Avenue, Cambridge, MA 02138, USA.

*Corresponding author. E-mail: bjones@kellogg.northwestern.edu

(Uzzi et al, 2013)

Software innovation as novel recombination of software libraries



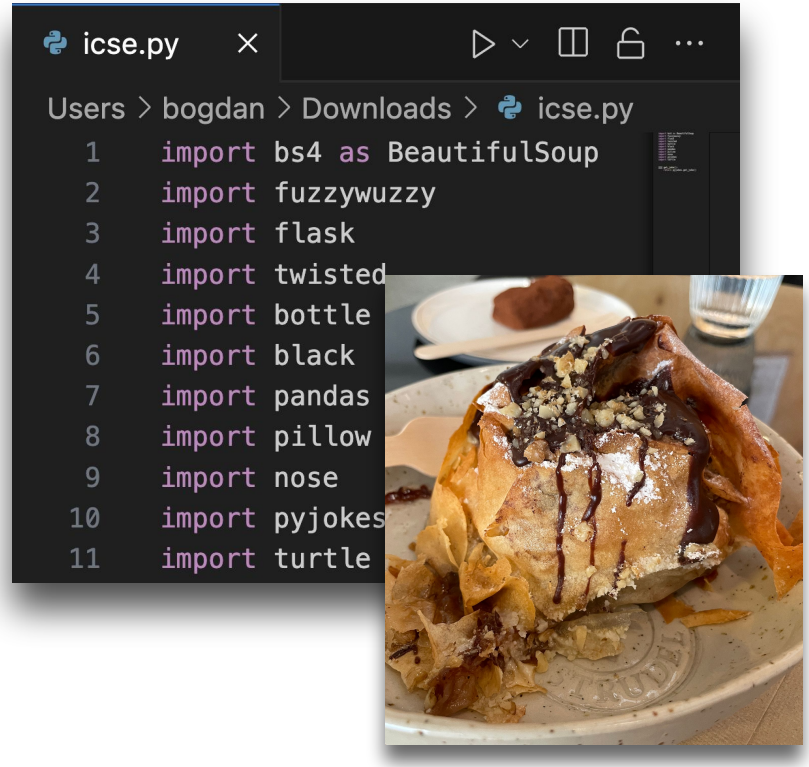
Lots of combinations:

- (twisted, bottle)
- (turtle, nose)
- (black, pandas)
- (fuzzywuzzy, pillow)
- …

C(n,2) unique pairs of packages.

Some of these may be highly innovative because they are atypical.

Software innovation as novel recombination of software libraries



Lots of combinations:

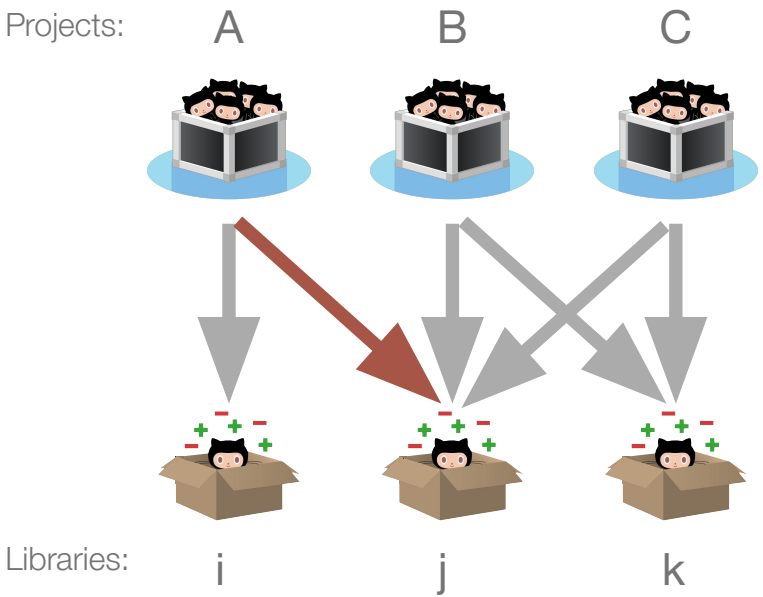
- (twisted, bottle)
- (turtle, nose)
- (black, pandas)
- (fuzzywuzzy, pillow)
- …

C(n,2) unique pairs of packages.

Dark chocolate + apple strudel is arguably innovative because it is atypical.

Key idea from network science: Comparison to null (random) model

Observed reality:

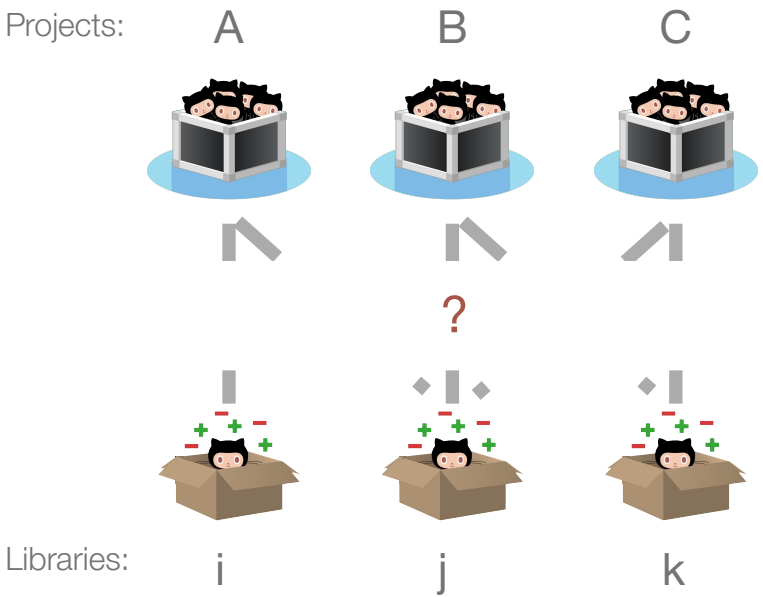


Project A adds a dependency on package j.
New combinations are formed, e.g., (i, j).

How atypical is (i, j)?

Key idea from network science: Comparison to null (random) model

Counterfactual:

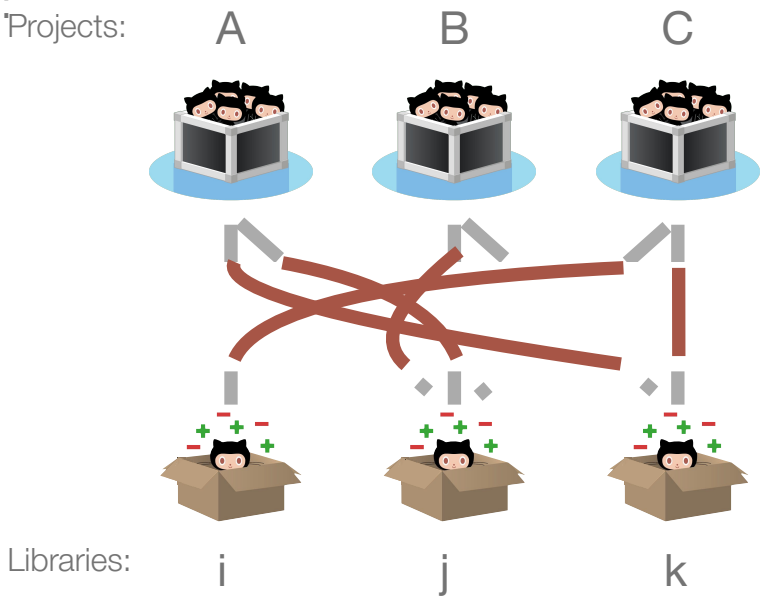


Preserve:

- all the projects
- all the libraries
- the distribution of imports per project
- the distribution of imports per library

Key idea from network science: Comparison to null (random) model

Counterfactual



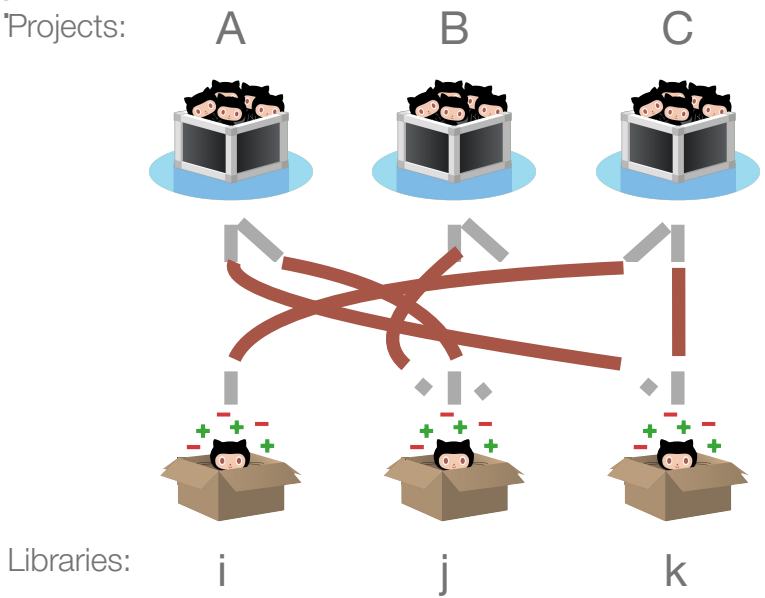
Preserve:

- all the projects
- all the libraries
- the distribution of imports per project
- the distribution of imports per library

But randomly rewire the network.

Key idea from network science: Comparison to null (random) model

Counterfactual



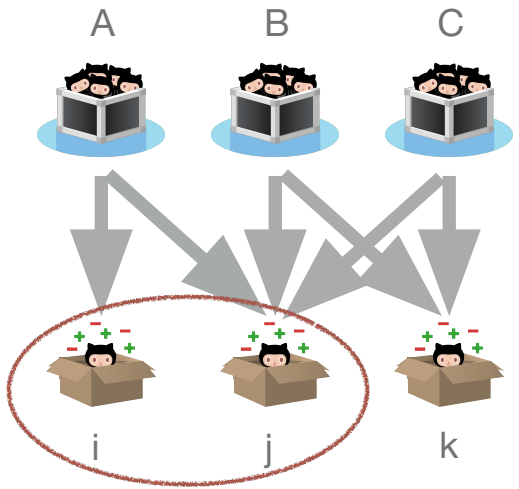
Preserve:

- all the projects
- all the libraries
- the distribution of imports per project
- the distribution of imports per library

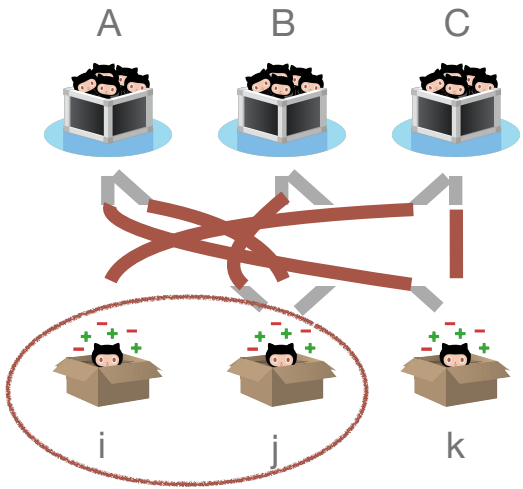
But randomly rewire the network.

And repeat many times.

This z-score estimates if two packages are used together more, less, or about as much as could be expected by chance.



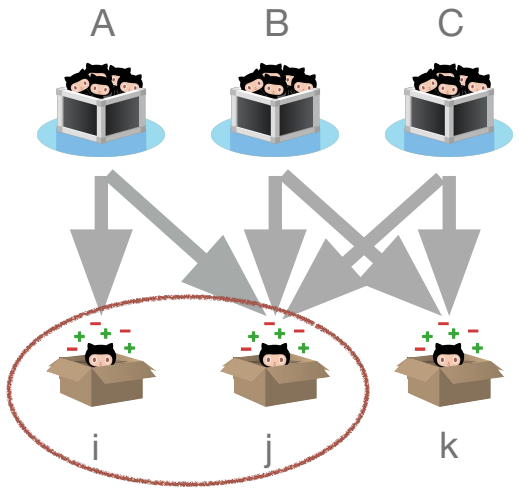
Observed number of times packages i and j appeared together until year t.



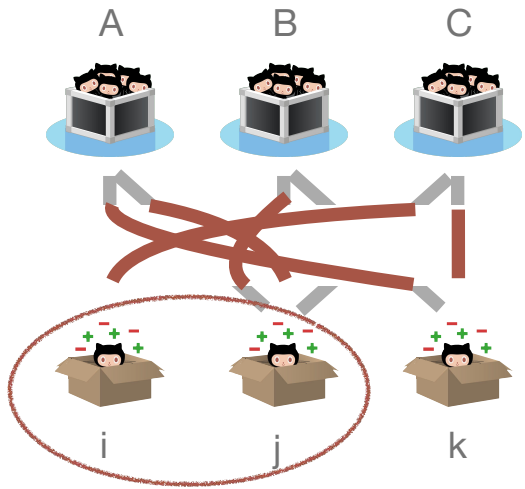
Average (i.e., expected) number of times packages i and j appeared together over N simulations.

$$z_{ijt} = (obs_{ijt} - exp_{ijt})/(\sigma_{ijt})$$

This z-score estimates if two packages are used together more, less, or about as much as could be expected by chance.



Observed number of times packages i and j appeared together until year t.



Average (i.e., expected) number of times packages i and j appeared together over N simulations.

low

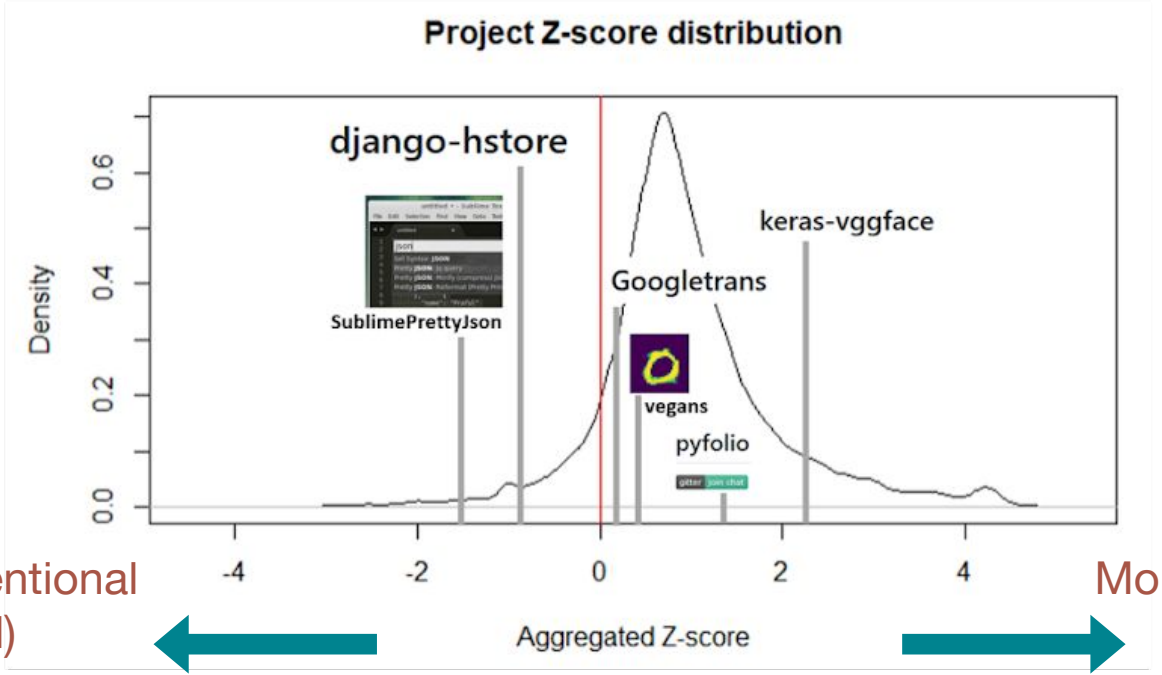
high

⇒ atypical combination

$$z_{ijt} = (obs_{ijt} - exp_{ijt})/(\sigma_{ijt})$$

Project-level aggregation is the average of pairwise atypicality z-scores

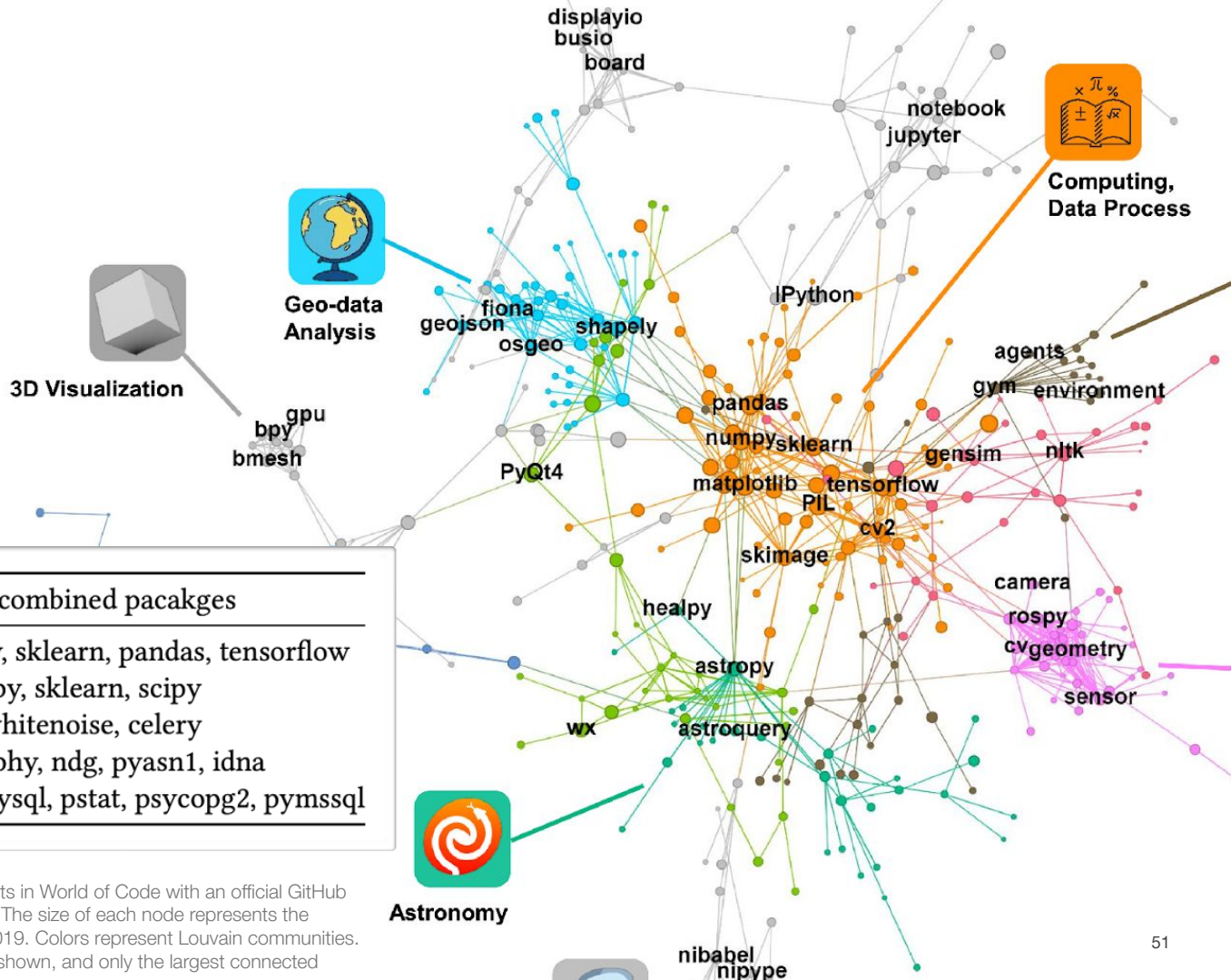
On average, projects are quite conventional.



Sanity checking

No ground truth on atypical package combinations, but at least the typical combinations should be meaningful!

Focal package	Top five mostly combined pacakges
numpy	matplotlib, scipy, sklearn, pandas, tensorflow
tensorflow	keras, cv2, numpy, sklearn, scipy
django	rest, dj, south, whitenoise, celery
OpenSSL	ntlm, cryptography, ndg, pyasn1, idna
pymysql	MySQLdb, aiomysql, pstat, psycopg2, pymssql



Fine print: Starting data consists of all Python projects in World of Code with an official GitHub release (75,388 projects and 7,728 packages total). The size of each node represents the number of projects that imported the package by 2019. Colors represent Louvain communities. Only top 0.006% of edges with the highest z-score shown, and only the largest connected

Software innovation as novel recombination of software libraries

Combining software libraries that are not often used together is like using unusual ingredients in your cooking.

- People may be impressed by your culinary creativity.
- Serving unusual dishes can be risky if the chefs are unable to perfect the recipes and the customers are unwilling to try new things.

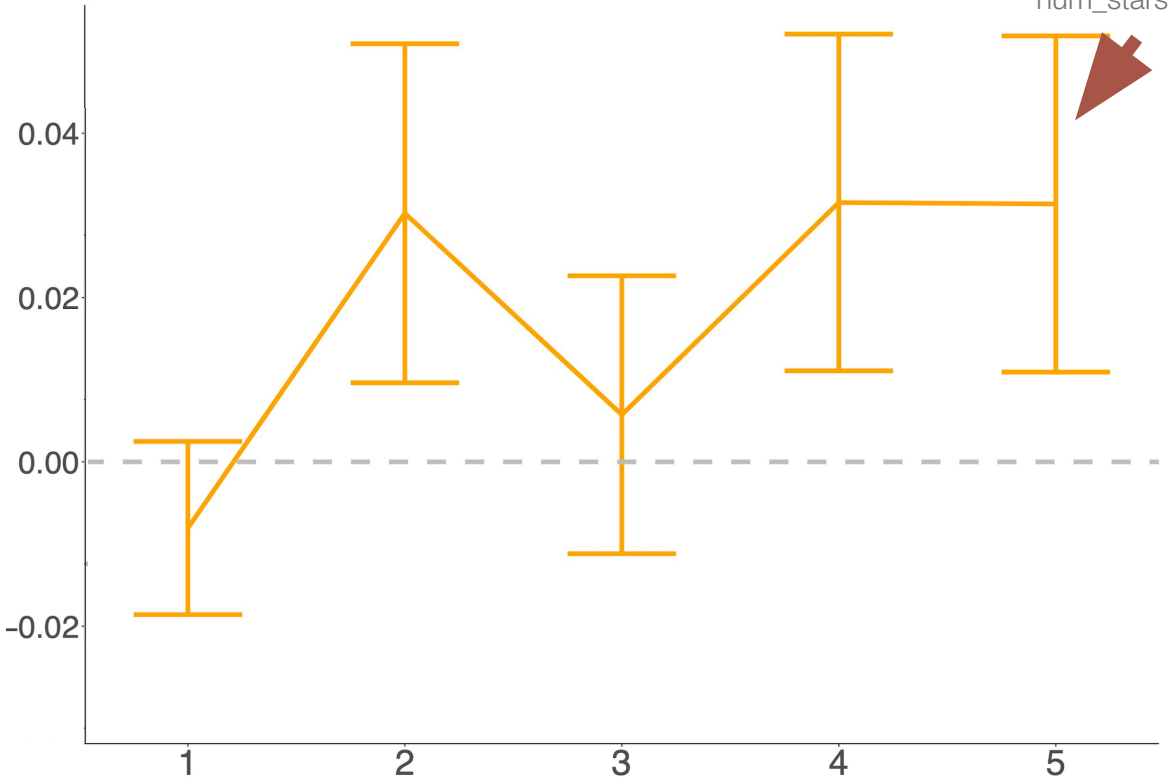
Software innovation as novel recombination of software libraries

Combining software libraries that are not often used together is like using unusual ingredients in your cooking.

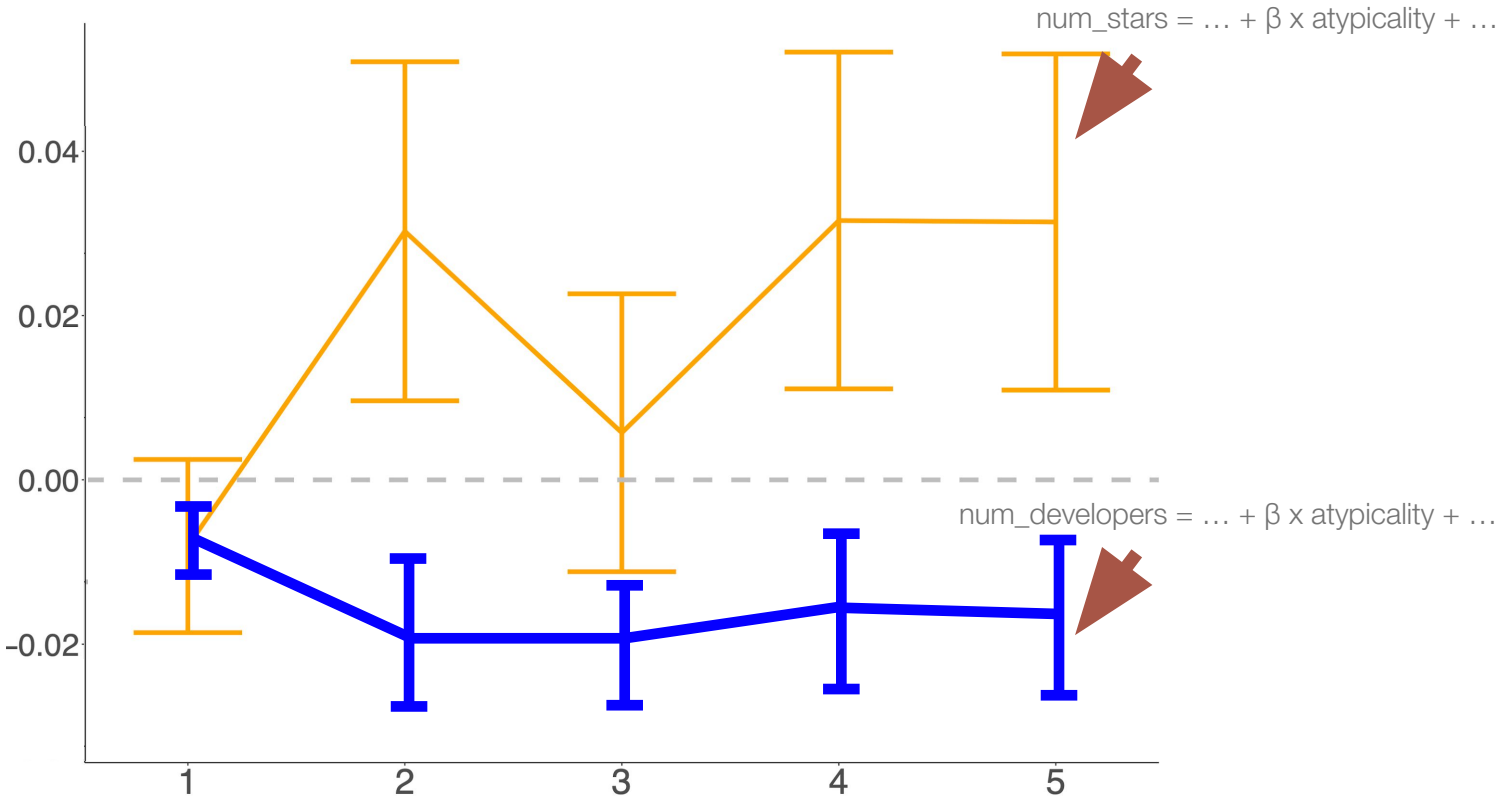
- Hyp: Projects that use more atypical combinations of libraries tend to be more popular.
- Hyp: More innovative projects tend to be less sustainable in the long term.

Atypical (novel) projects tend to have more stars.

num_stars = … + β x atypicality + …

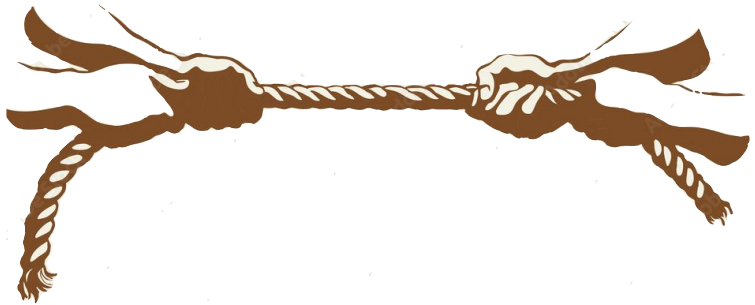


Atypical (novel) projects tend to have smaller teams (and higher probability of becoming abandoned).



Tension between innovation and open source sustainability?

Incentive to create ever-new things



The “grunt work” of maintaining existing systems

- Creative expression is a main driver of contributing to open source
- Innovation seems to be rewarded with increased popularity

Will it become increasingly harder to ensure that sufficient maintenance attention (developers, funding, etc) is being allocated to the projects that need it the most?

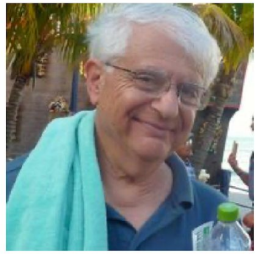
Now, how does innovation emerge?

Once upon a time, a PhD student at Harvard University was writing their dissertation …

Stanford
Sociology
SCHOOL OF HUMANITIES AND SCIENCES

Mark Granovetter

Joan Butler Ford Professor in the School of Humanities and Sciences; Professor of Sociology



A.B. Princeton University 1965
Modern European and American History
Ph.D. Harvard University 1970
Sociology

https://sociology.stanford.edu/people/mark-granovetter

The Strength of Weak Ties[1]

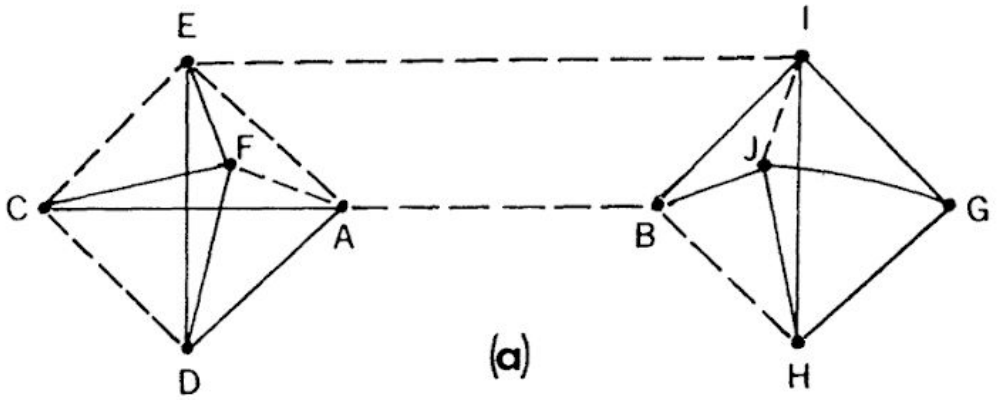
Mark S. Granovetter
Johns Hopkins University

Analysis of social networks is suggested as a tool for linking micro and macro levels of sociological theory. The procedure is illustrated by elaboration of the macro implications of one aspect of small-scale interaction: the strength of dyadic ties. It is argued that the degree of overlap of two individuals' friendship networks varies directly with the strength of their tie to one another. The impact of this principle on diffusion of influence and information, mobility opportunity, and community organization is explored. Stress is laid on the cohesive power of weak ties. Most network models deal, implicitly, with strong ties, thus confining their applicability to small, well-defined groups. Emphasis on weak ties lends itself to discussion of relations *between* groups and to analysis of segments of social structure not easily defined in terms of primary groups.

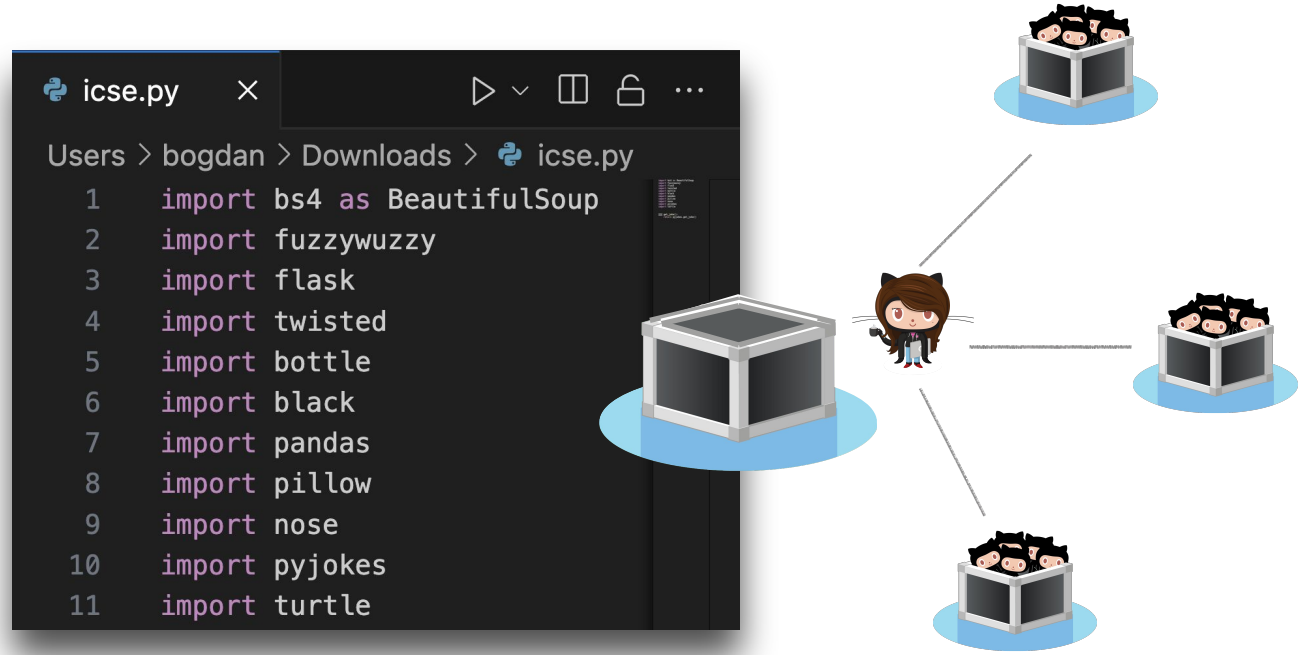
Weak ties are more effective in job searches because they act as bridges.

The majority of people found their jobs through acquaintances (weak ties) rather than close friends or family (strong ties).

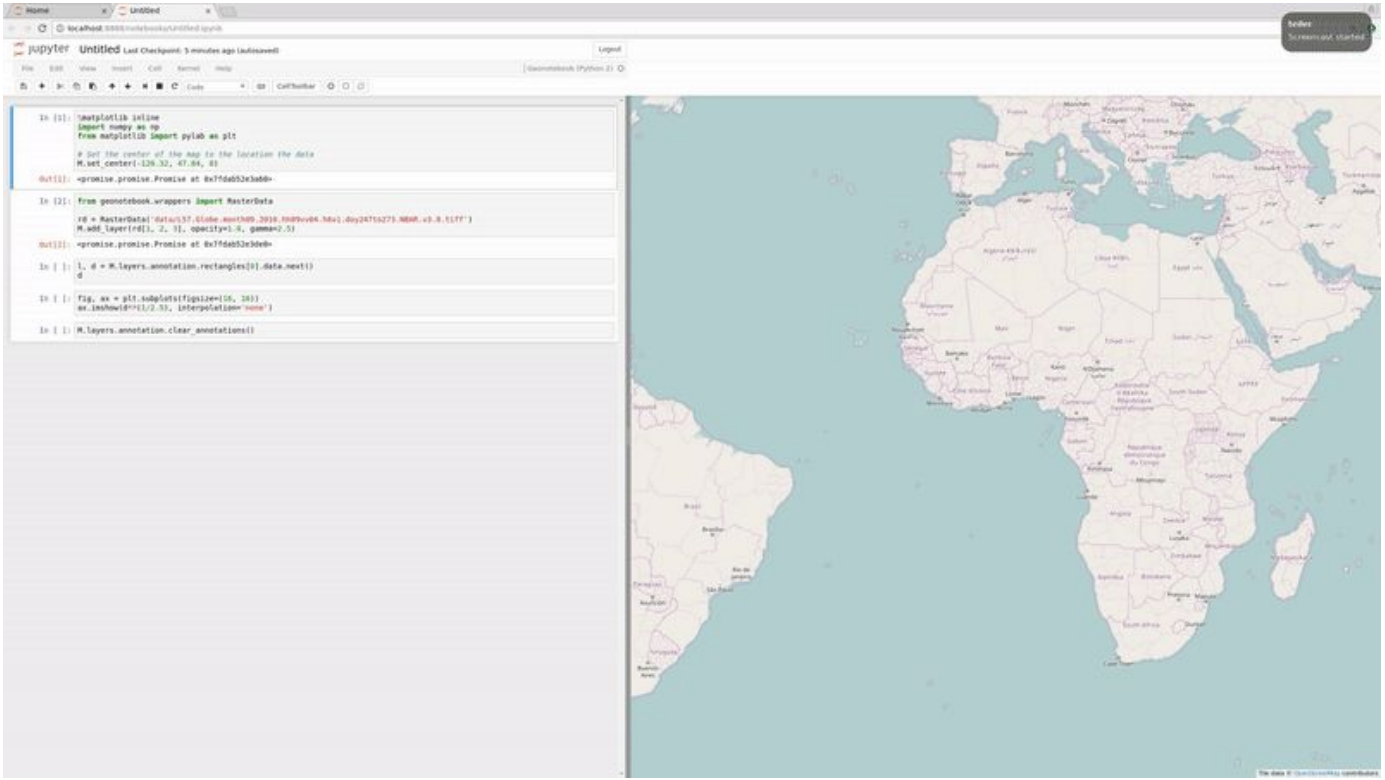
In a random sample of recent professional, technical, and managerial job changers living in a Boston suburb, I asked those who found a new job through contacts how often they *saw* the contact around the time that he passed on job information to them. I will use this as a measure of tie strength.[15] A natural a priori idea is that those with whom one has strong ties are more motivated to help with job information. Opposed to this greater motivation are the structural arguments I have been making: those to whom we are weakly tied are more likely to move in circles different from our own and will thus have access to information different from that which we receive.



Do OSS developers also find their new ideas through weak ties?

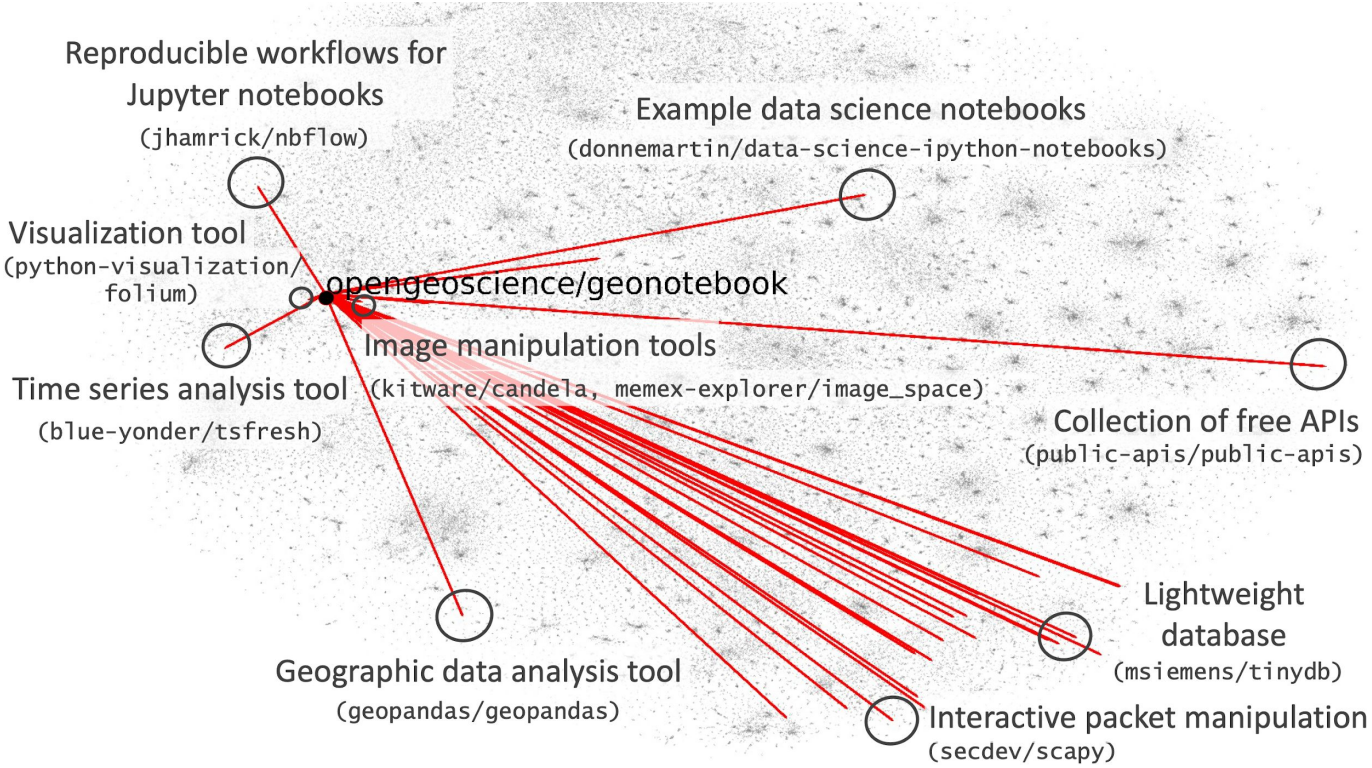


Do OSS developers also find their new ideas through weak ties?



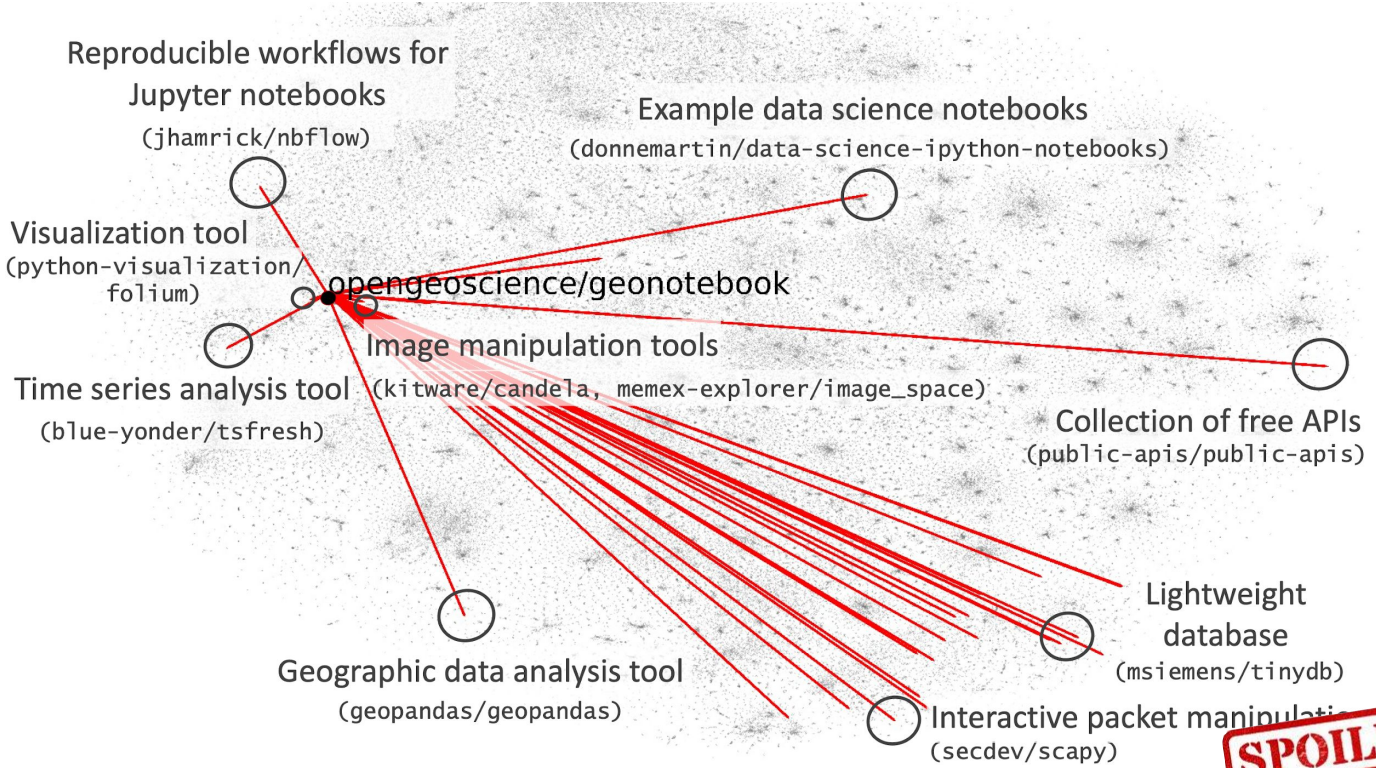
https://github.com/opengeoscience/geonotebook

Do OSS developers also find their new ideas through weak ties?



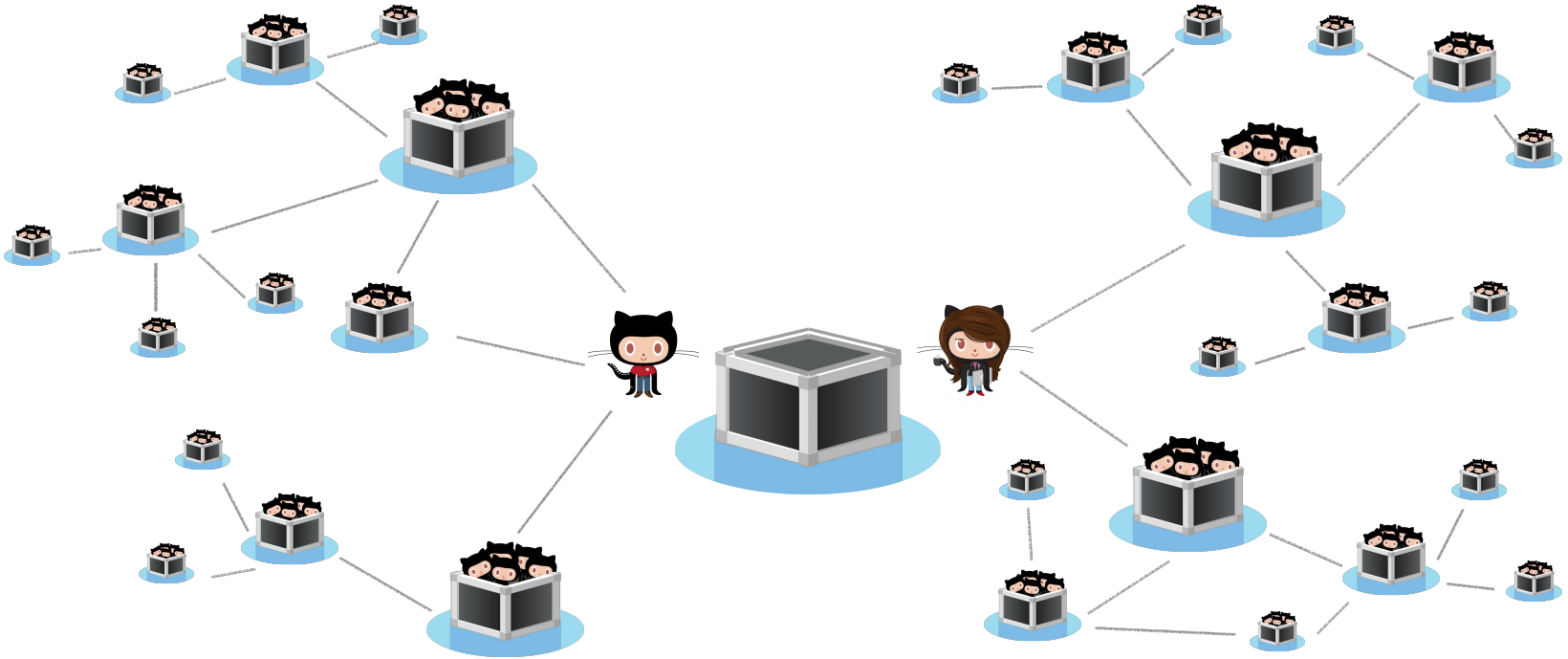
Anecdotally, yes

Do OSS developers also find their new ideas through weak ties?

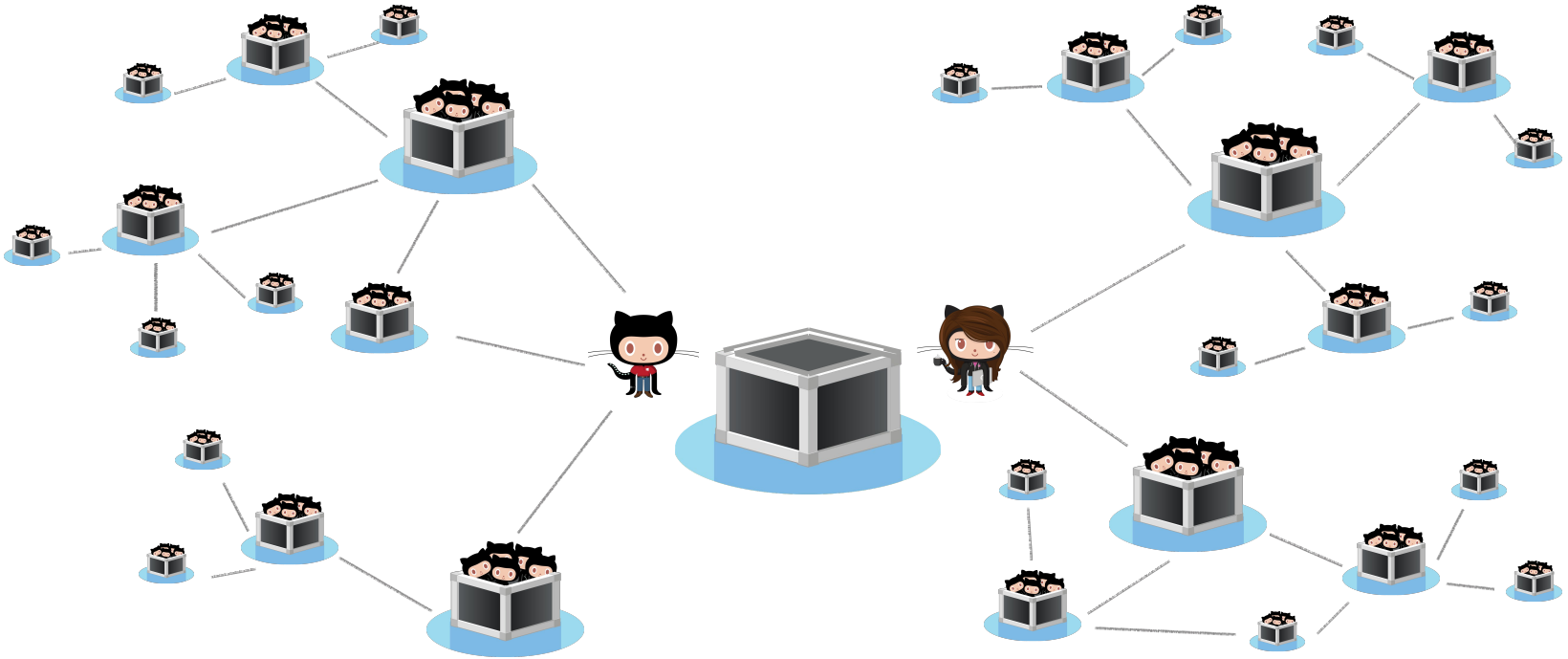


Amazingly, statistically also yes!

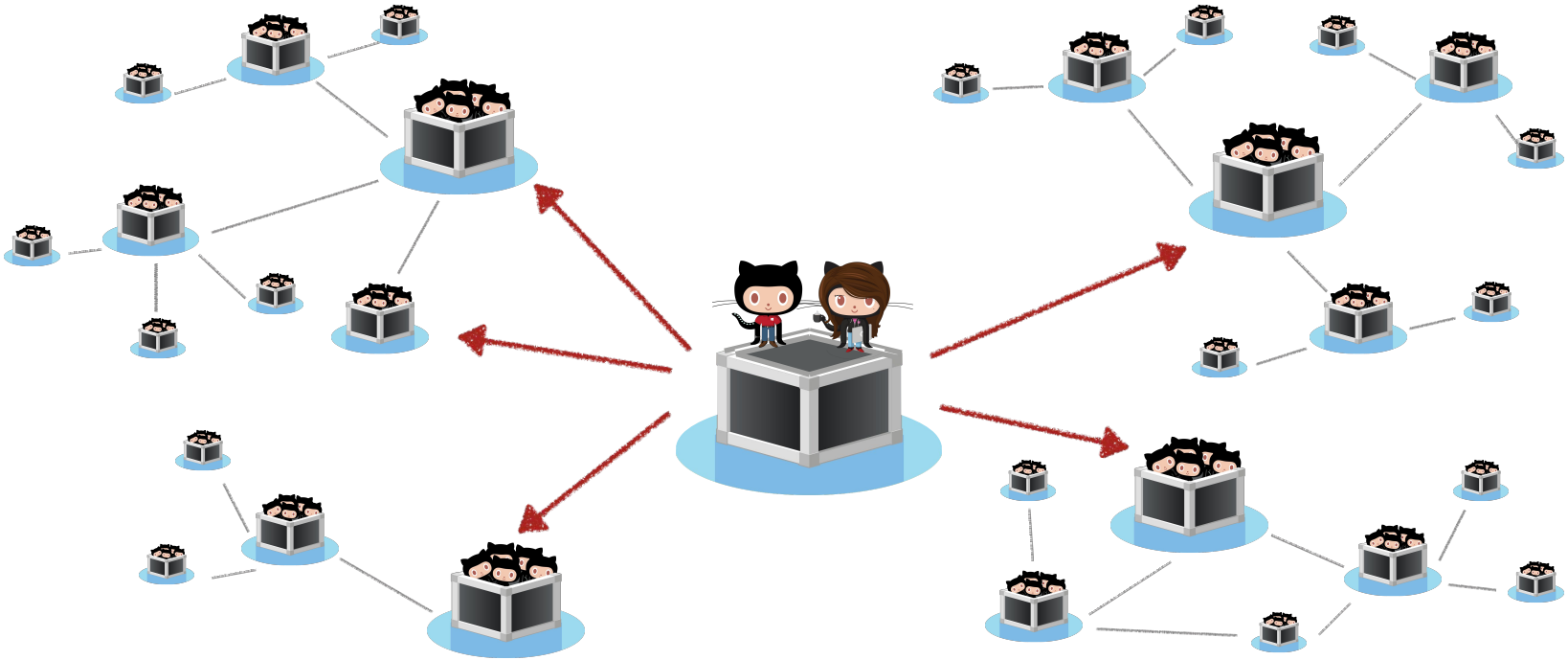
People interact with artifacts and with each other. This creates ties.



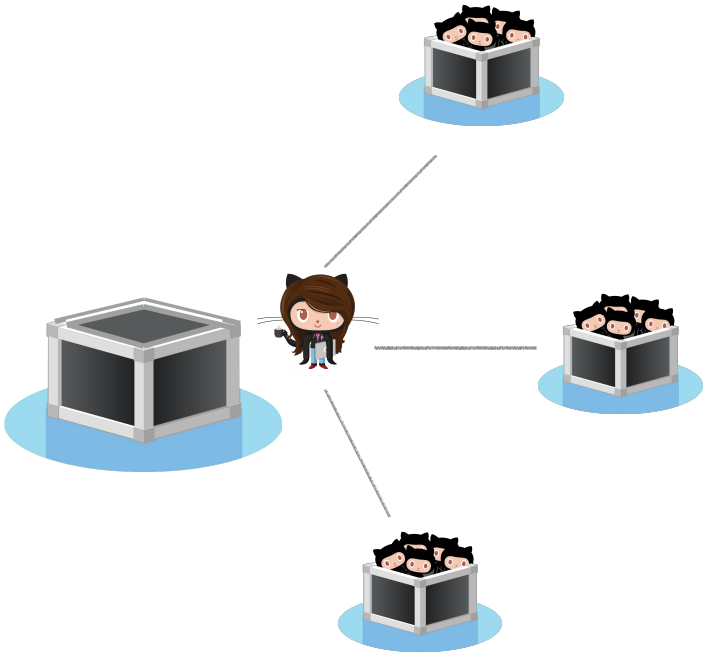
Hypothesis 1: The bigger developers' networks are, the better informed they are, and the more innovative their projects are.



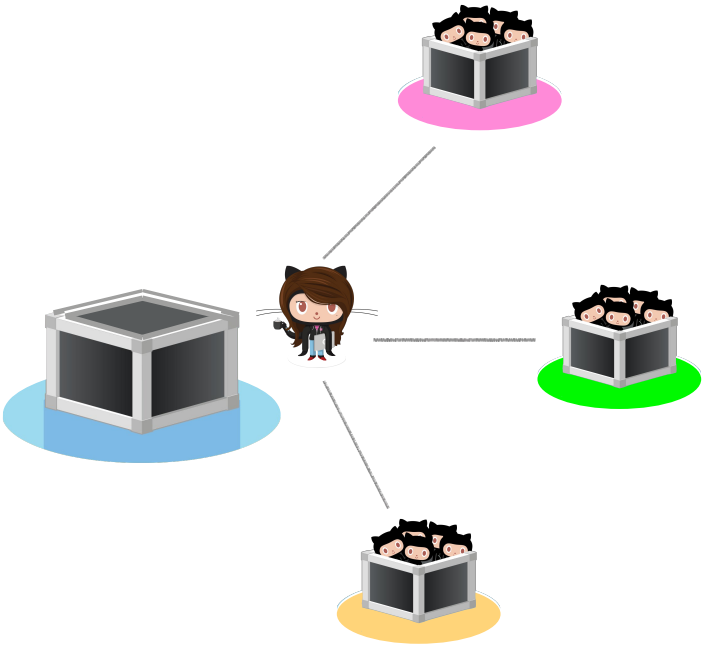
Measure: Out-degree centrality



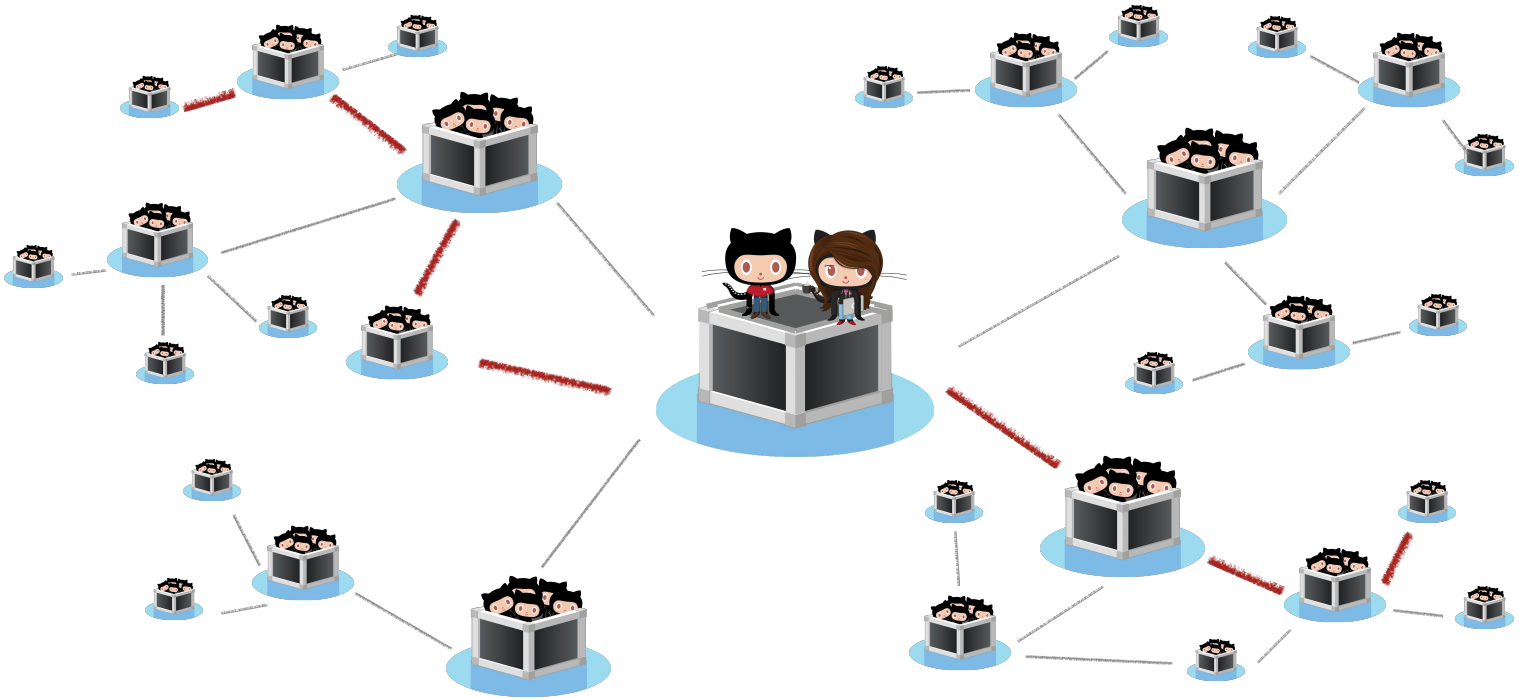
Hypothesis 2: The greater the informational diversity of developers' networks, the more innovative their projects are.



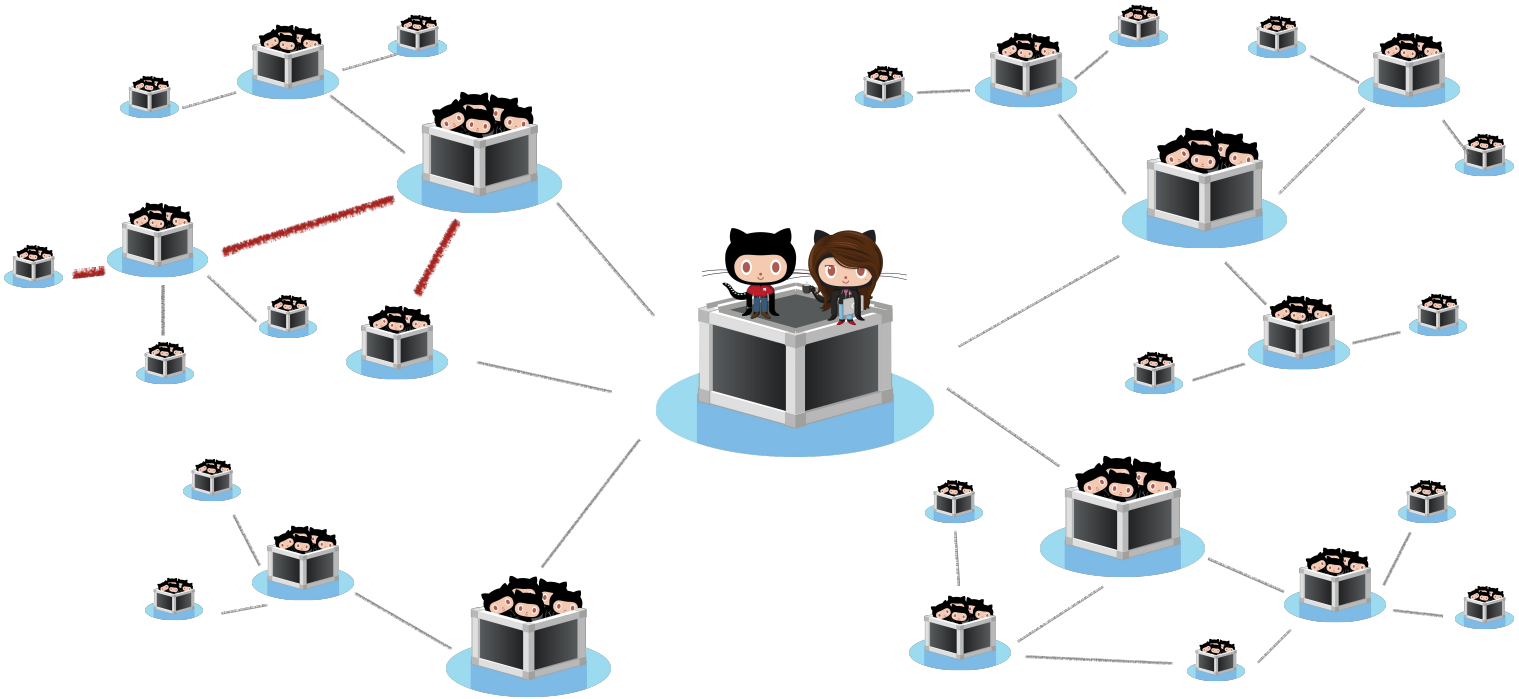
vs.



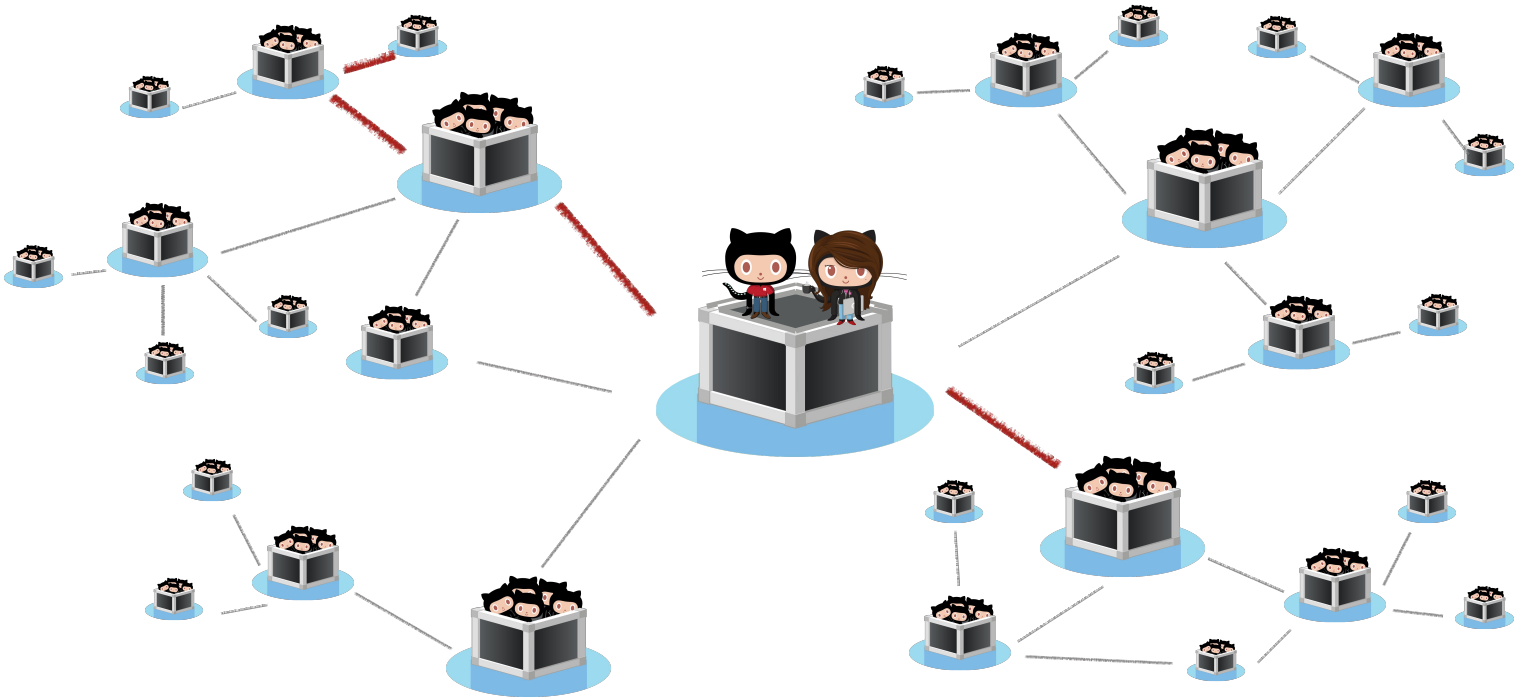
Measure: First, we generate Node2Vec embeddings for each project



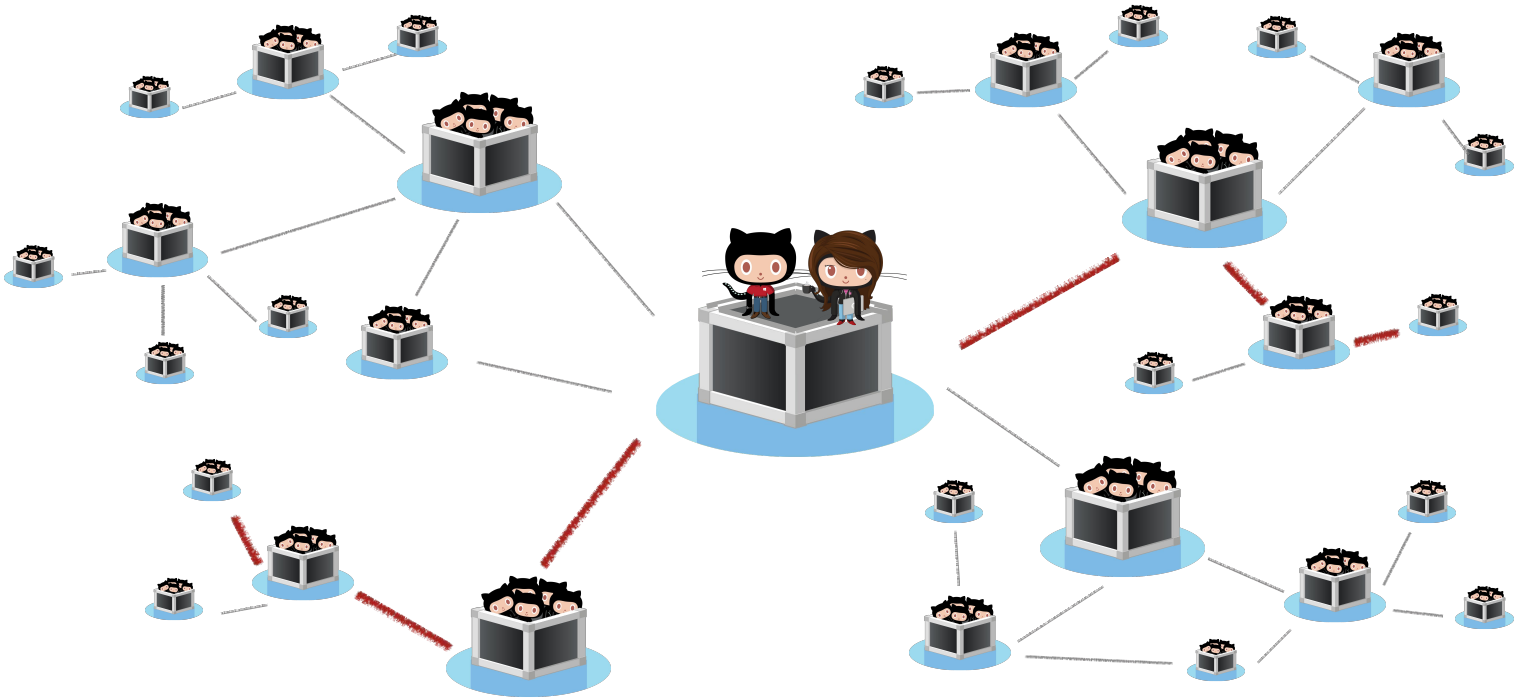
Measure: First, we generate Node2Vec embeddings for each project



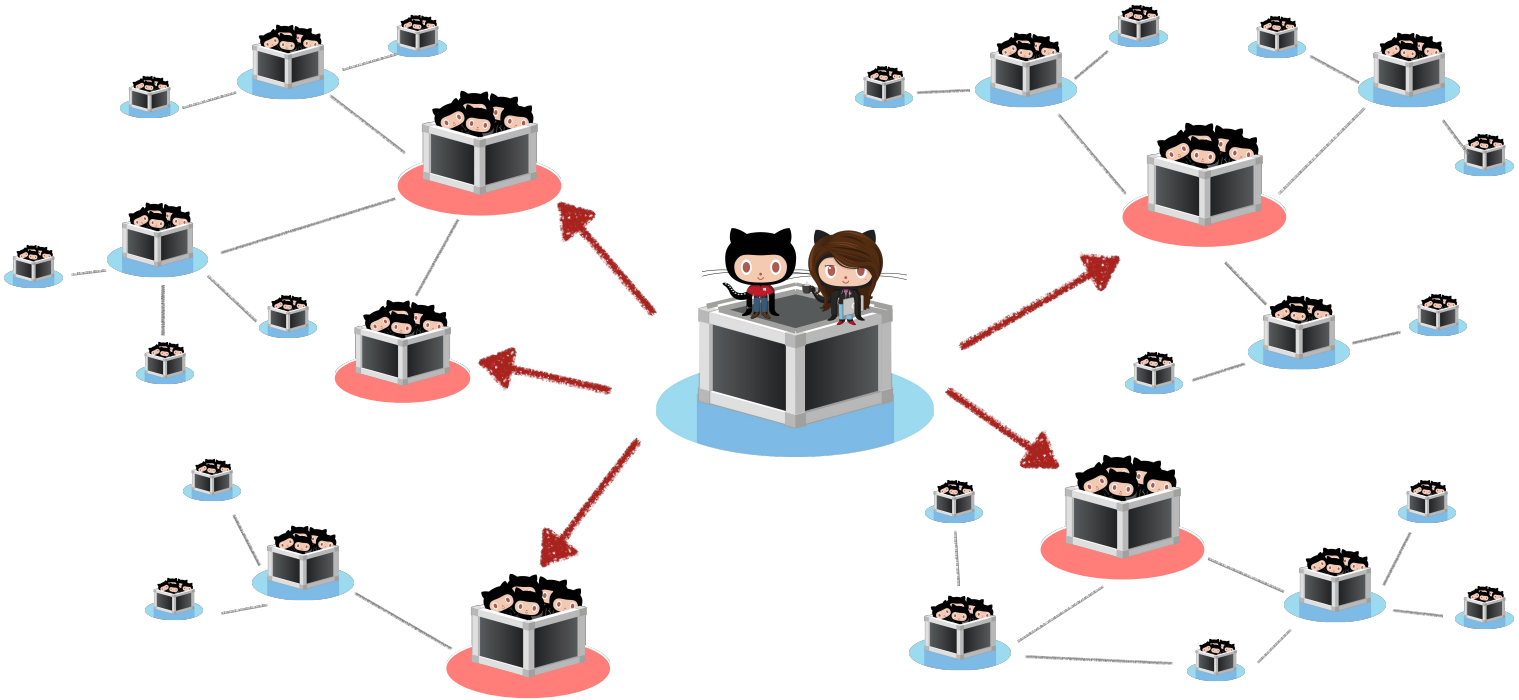
Measure: First, we generate Node2Vec embeddings for each project



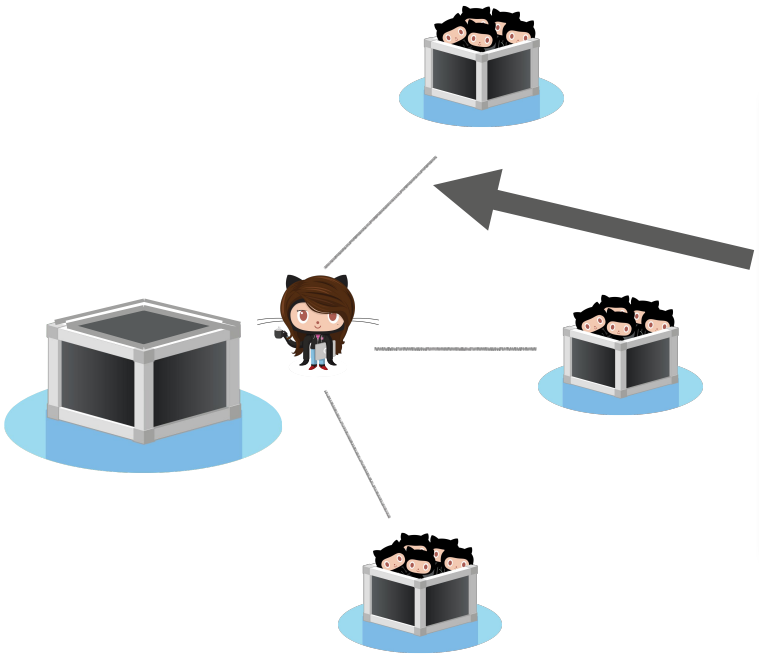
Measure: First, we generate Node2Vec embeddings for each project



Measure: Then, we compute the average pairwise distance (inverse cosine similarity) between a focal project's direct neighbors



From interactions to ties of varying strength



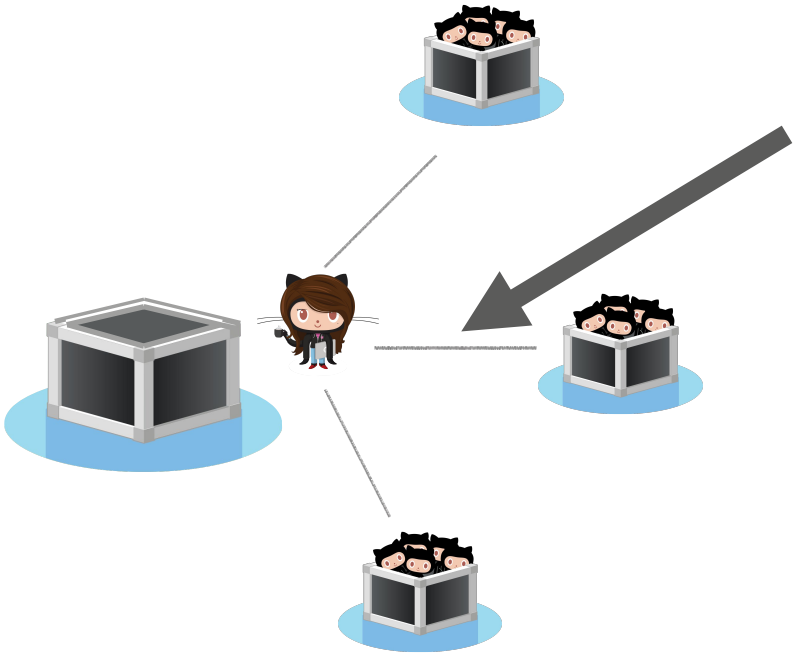
1 file changed +1 -1 lines changed

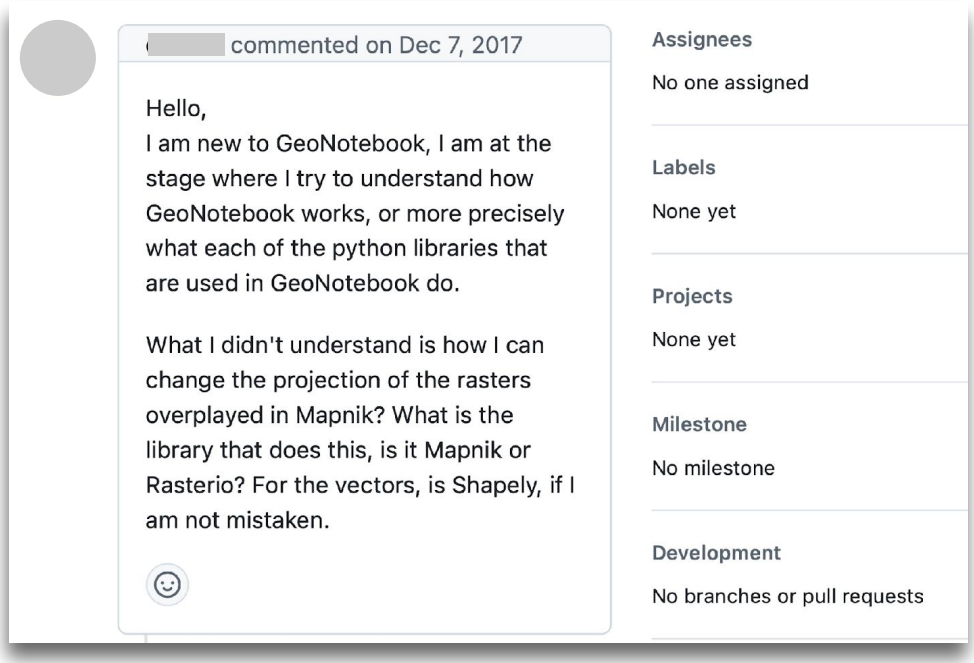
js/config/resolve.js +1 -1

```
@@ -1,6 +1,6 @@
1  var path = require('path');                  1  var path = require('path');
2                                               2
3  - var renderer =                             3  + var renderer =
   process.env.GEONOTEBOOK_MAP_RENDERE             process.env.GEONOTEBOOK_MAP_RENDERE
   R || 'geojs';                                   R || 'ol';
4                                               4
5  module.exports = {                           5  module.exports = {
6    alias: {                                   6    alias: {
```

Commits to the codebase
(relatively deep understanding of the codebase)

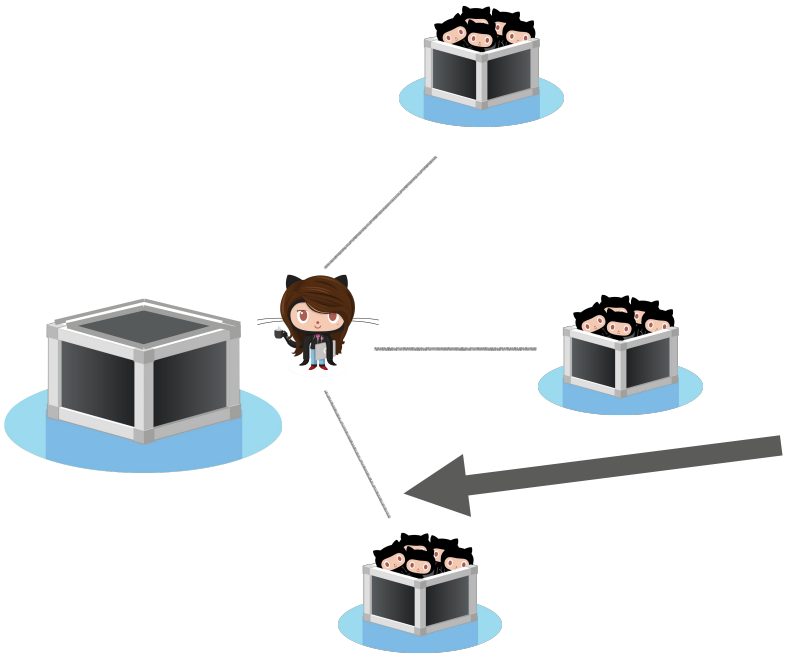
From interactions to ties of varying strength

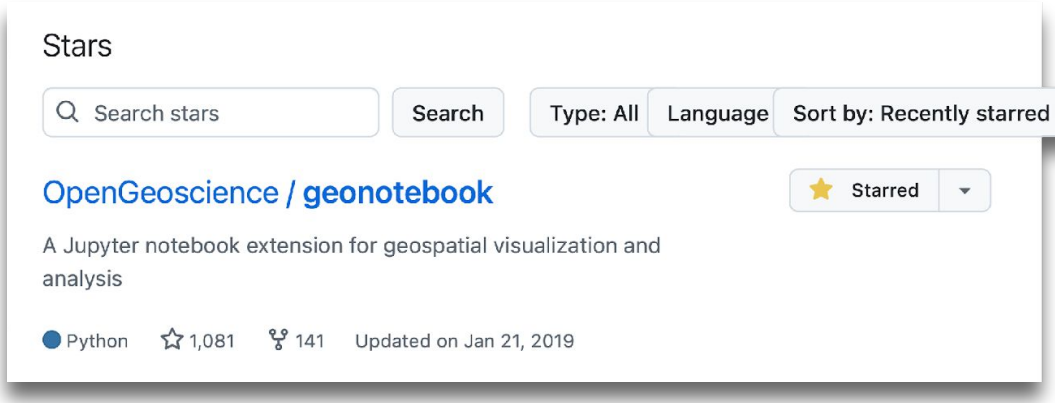




Issue reports
(some understanding of the project)

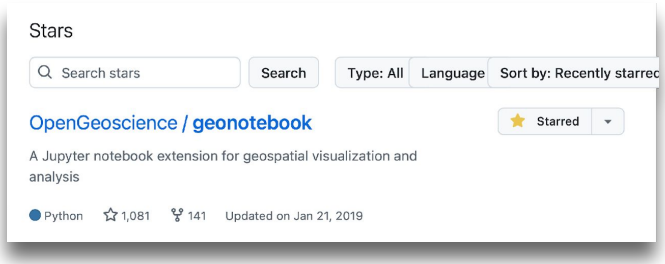
From interactions to ties of varying strength



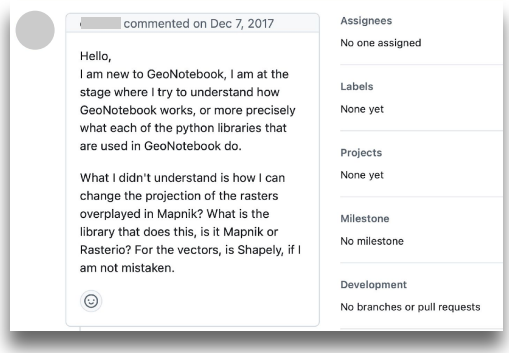


Stars
(awareness of the project)

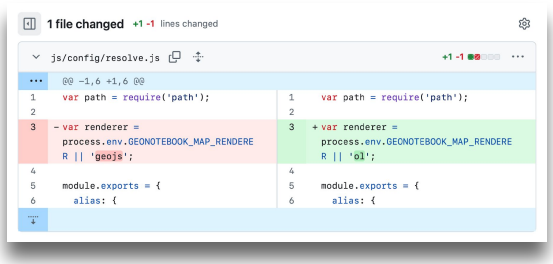
Many interactions are possible, these were just three examples.



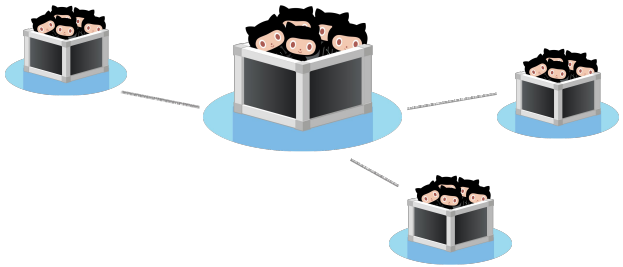
Stars



Issues



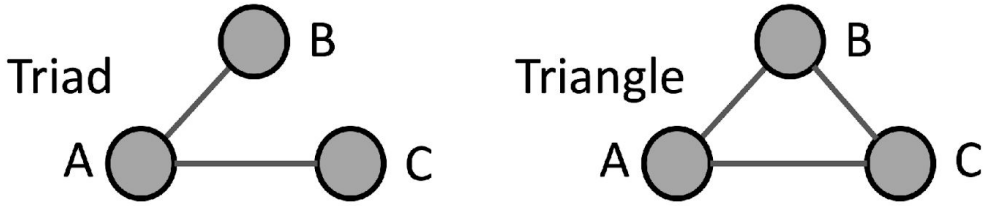
Commits



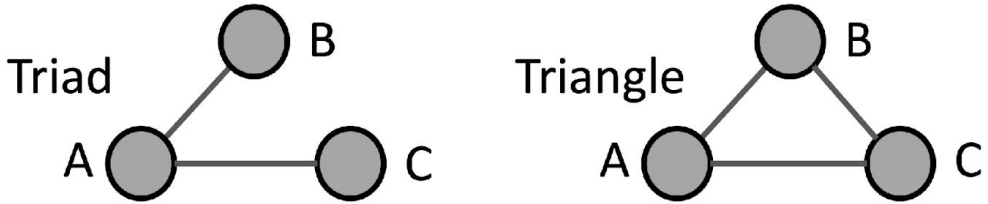
Weaker ties

Stronger ties

In strongly-tied social networks, triads are unlikely.



There is ~an order of magnitude (10×) difference in transitivity values between each pair of networks.

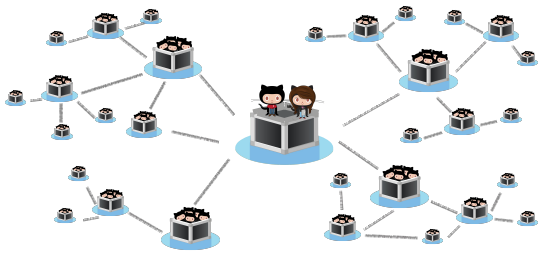


Interaction	#Nodes	#Edges	Transitivity ($\times 10^{-2}$)
Commits	763,062	1,926,978	30.04
Issues	278,945	727,255	3.42
Stars	480,394	3,658,543	0.23

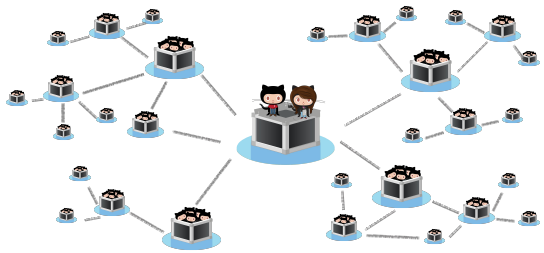
$$\text{Transitivity} = 3 * N_{\text{triangles}} / N_{\text{triads}}$$

Commits >> Issues >> Stars

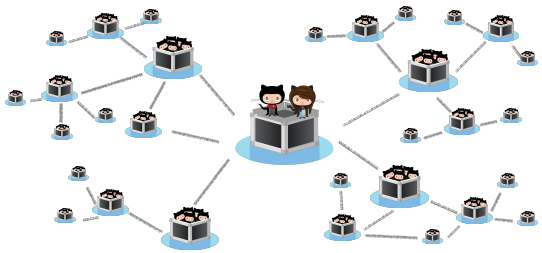
Now what?



Stars

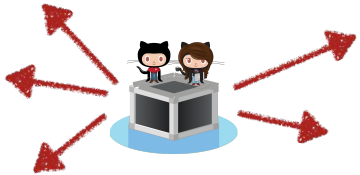


Issues

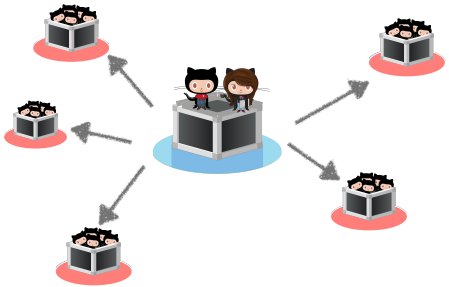


Commits

Out-degree centrality x 3?



Information diversity index x3?



The first two PCs cumulatively explain over 80% of the variance.

	Out-deg. centrality			Diversity index		
	PC1	**PC2**	**PC3**	**PC1**	**PC2**	**PC3**
D_{commit}	0.60	−0.45	0.67	0.63	−0.36	0.69
D_{issue}	0.61	−0.28	−0.74	0.64	−0.24	−0.72
D_{star}	0.52	0.85	0.11	0.43	0.90	0.08

PC1: Average volume of information available / Average diversity of the knowledge space (hyp 2)

The first two PCs cumulatively explain over 80% of the variance.

	Out-deg. centrality			Diversity index		
	PC1	**PC2**	**PC3**	**PC1**	**PC2**	**PC3**
D_{commit}	0.60	−0.45	0.67	0.63	−0.36	0.69
D_{issue}	0.61	−0.28	−0.74	0.64	−0.24	−0.72
D_{star}	0.52	0.85	0.11	0.43	0.90	0.08

PC2: Where the connectivity / diversity comes from

(The strength of weak ties)

Hypothesis 3: The more the informational diversity can be attributed to weak ties, the more innovative the projects are.

	Out-deg. centrality			Diversity index		
	PC1	**PC2**	**PC3**	**PC1**	**PC2**	**PC3**
D_{commit}	0.60	−0.45	0.67	0.63	−0.36	0.69
D_{issue}	0.61	−0.28	−0.74	0.64	−0.24	−0.72
D_{star}	0.52	0.85	0.11	0.43	0.90	0.08

PC2: Where the connectivity / diversity comes from

(The strength of weak ties)

Finally, the novelty regression:

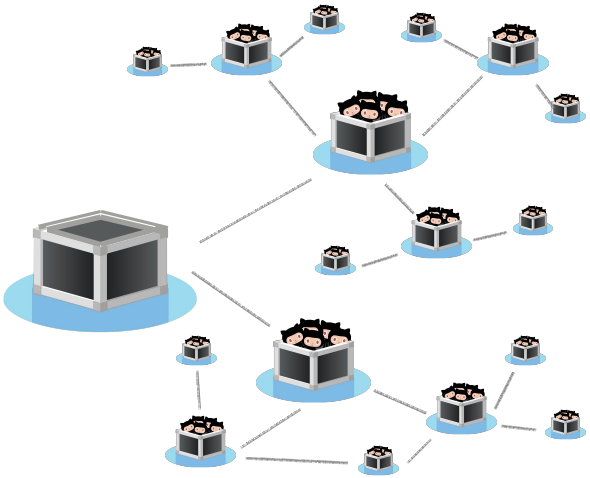
- Hypothesis 1 (greater connectivity): weak/inconsistent effects
- Hypothesis 2 (greater info diversity): small but clear effects (25—75 percentile: 4% change in the distribution)
- Hypothesis 3 (strength of weak ties): clear effects, comparable size

	Model III	Model IV
Variables of interest		
Deg_{ave} $(\mathbf{H_1})$		−0.002***
		(0.001)
$Deg_{weakness}$		−0.005***
		(0.001)
Div_{ave} $(\mathbf{H_2})$	0.007***	0.008***
	(0.001)	(0.001)
$Div_{weakness}$ $(\mathbf{H_3})$	0.005***	0.007***
	(0.001)	(0.001)
Observations	38,164	38,164

*p<0.05; **p<0.01; ***p<0.001

Exposure to diverse ideas through weak ties predicts novel combinations of packages.

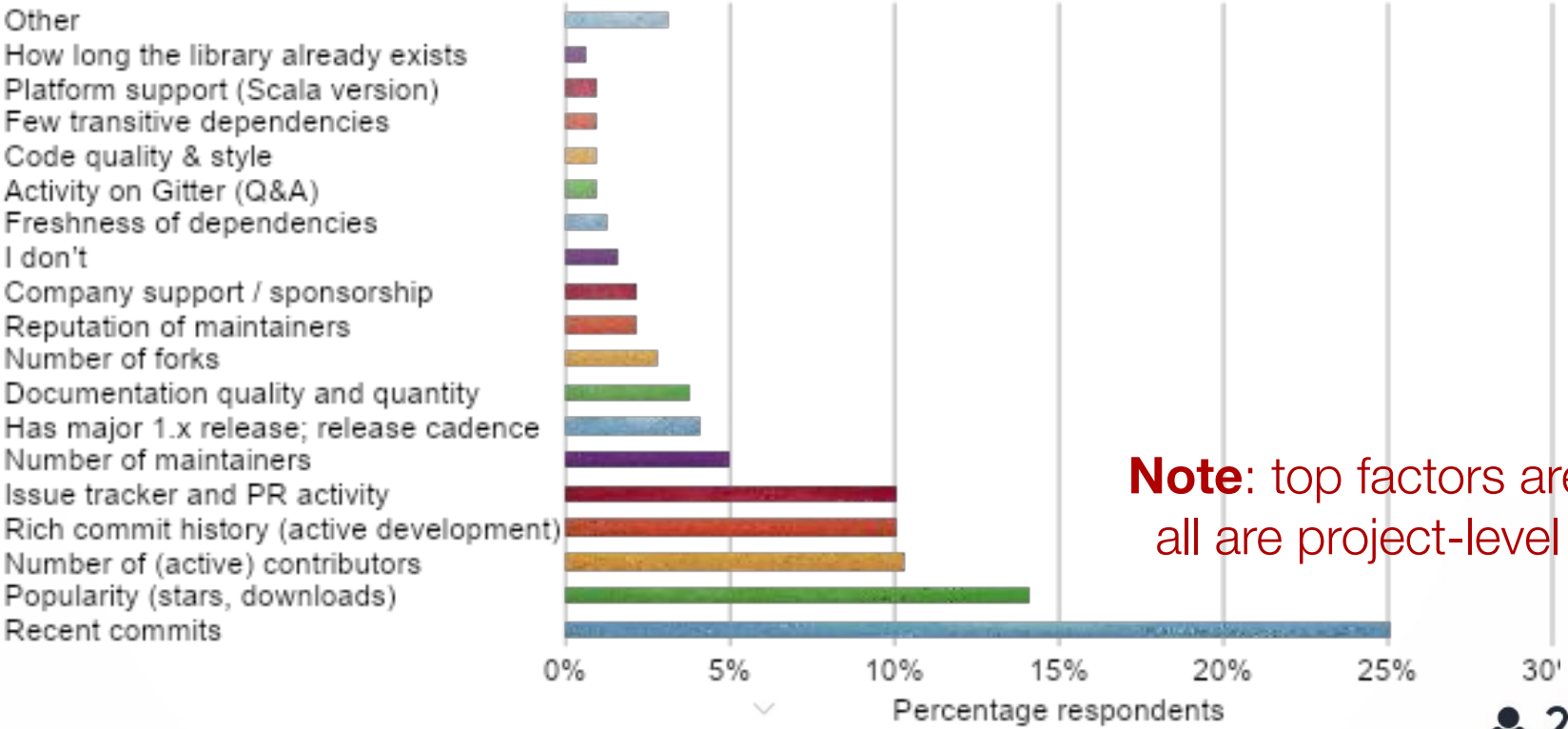
- Lurking on the GitHub platform seems to have quantifiable benefits. Redesign the Trending page?
- Automated project recommendation tools may be counterproductive?
- Well-informed but not necessarily highly active developers may also be experts at their craft?
- How to track and give credit to ideas?
- Surface-level vs deep-level diversity?
- AI-generated code: novel or regression to the mean?



“Ecosystem-level determinants of sustained activity in open-source projects: A case study of the PyPI ecosystem” Valiev et al, FSE 2018

How do you screen open source libraries to make sure they would still be maintained in the future?

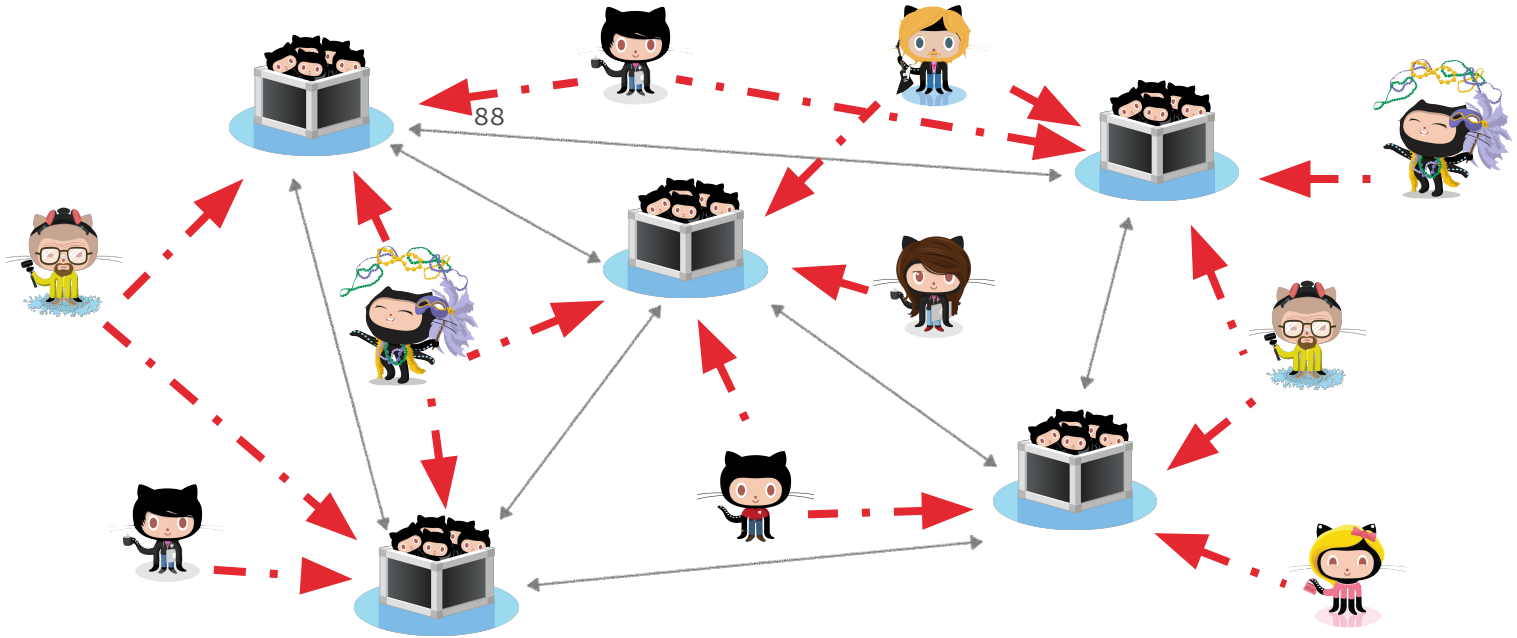
Mentimeter



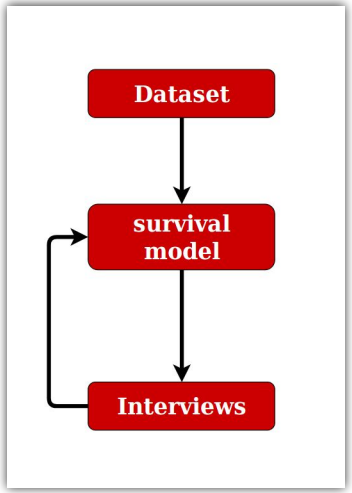
Note: top factors are all are project-level

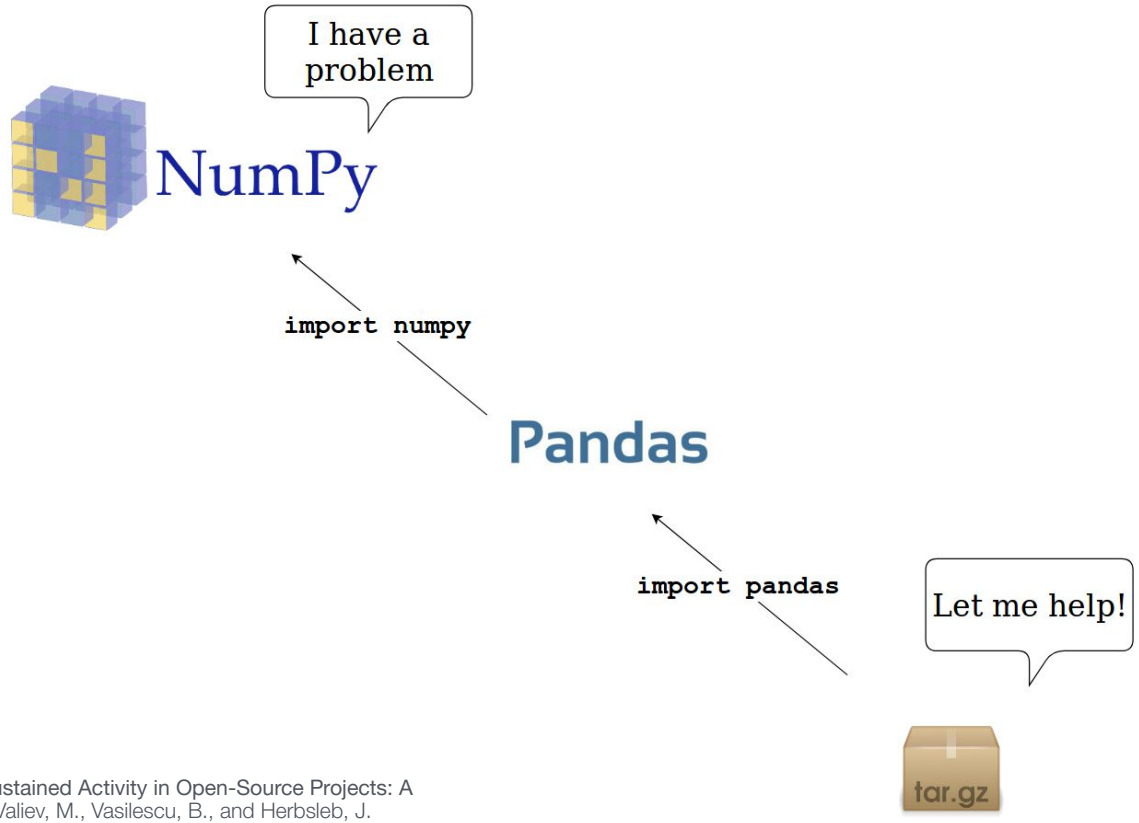
290

But projects are often part of larger ecosystems



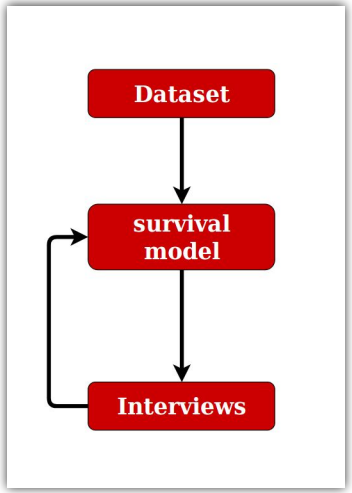
Transitive downstream dependencies are .....

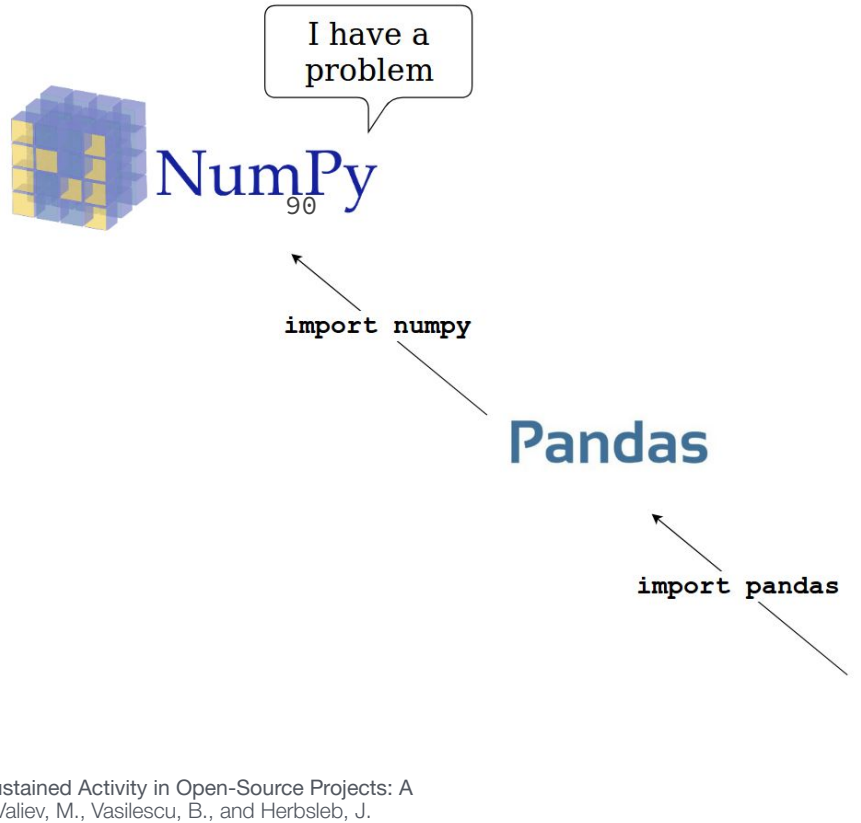




- Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem. Valiev, M., Vasilescu, B., and Herbsleb, J. *ESEC/FSE 2018*

Transitive downstream dependencies are harmful





Survival models

Early stage: **-12%** survival

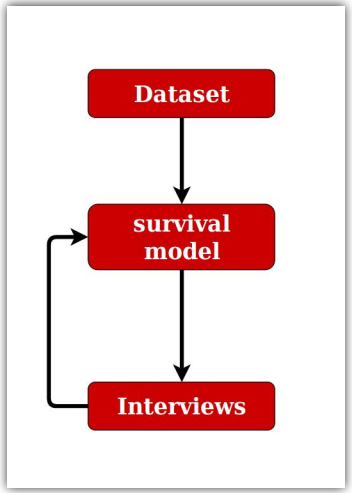
Long term: **-27%** survival

Interviews:

- less likely to fix
- just as likely to complain

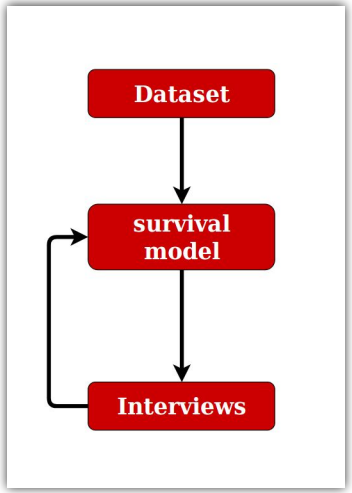
- Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem. Valiev, M., Vasilescu, B., and Herbsleb, J. *ESEC/FSE 2018*

Commercial involvement is .....



- Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem. Valiev, M., Vasilescu, B., and Herbsleb, J. *ESEC/FSE 2018*

Commercial involvement is harmful



Survival models

Early stage: **-51%** survival

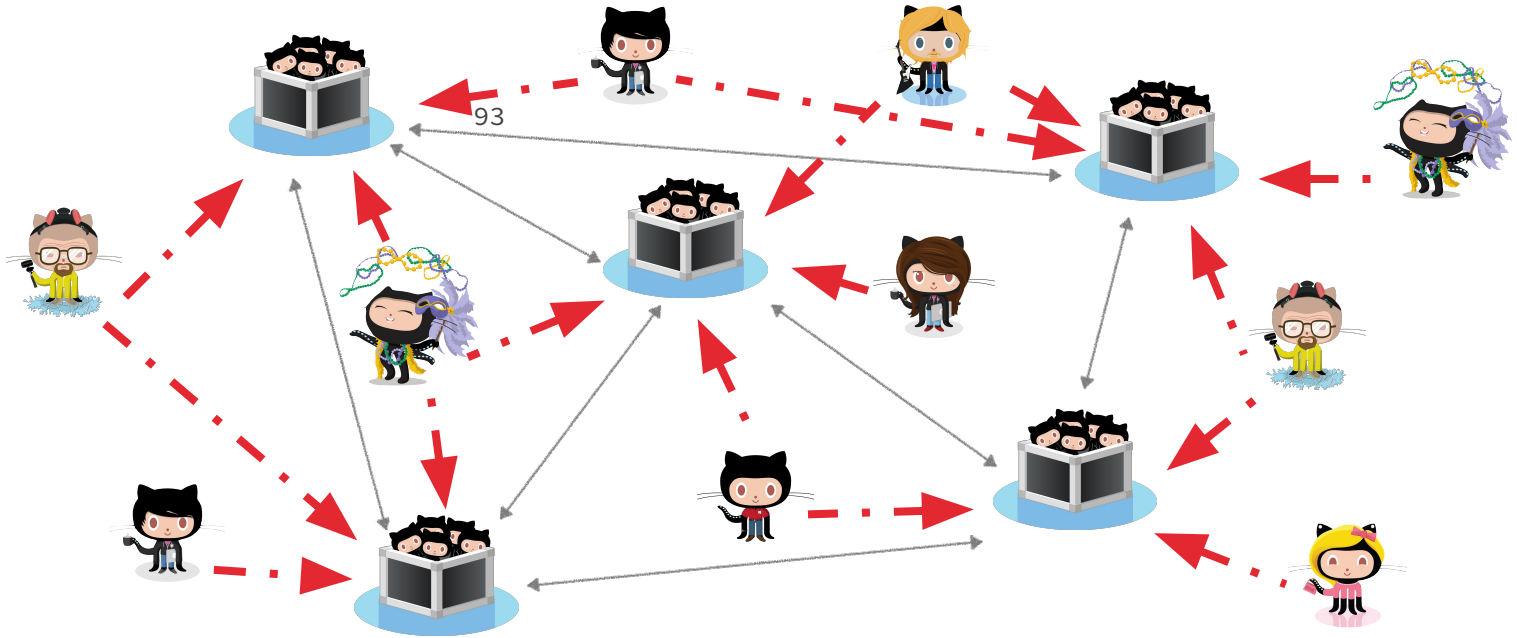
Long term: **-15%** survival

Interviews:

- more resources
- but can withdraw anytime

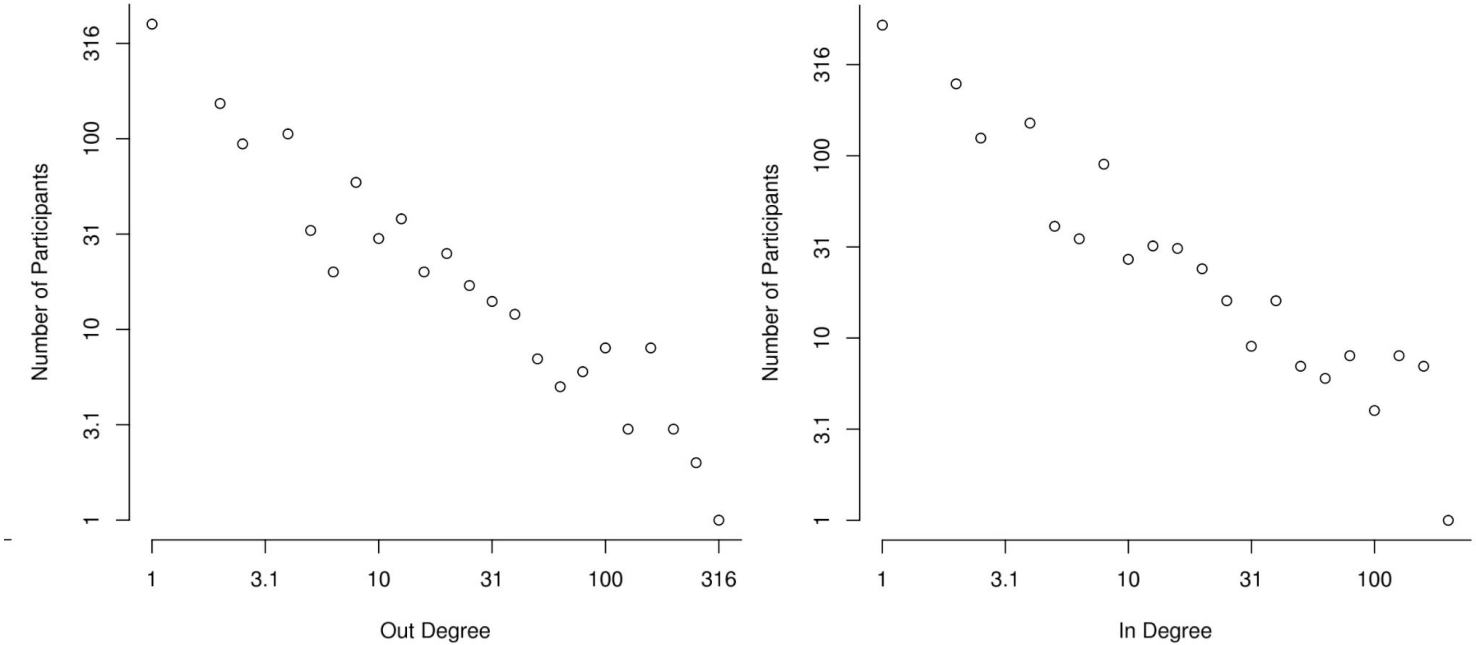
- Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem. Valiev, M., Vasilescu, B., and Herbsleb, J. *ESEC/FSE 2018*

Take away: Network effects!



“Mining Email Social Networks”
Bird et al, MSR 2006

Email social networks are scale free



Out degree is an indication of status, as it indicates the number of different people who replied to the ego's messages.

“Latent Social Structure in Open Source Projects” Bird et al, FSE 2008

Do OSS projects have some latent structure?

Are there dynamic, self-organizing subgroups that spontaneously form and evolve?

Hypothesis 1 – Subcommunities of participants will form in the email social networks of large open source projects and the levels of modularity will be statistically significant.

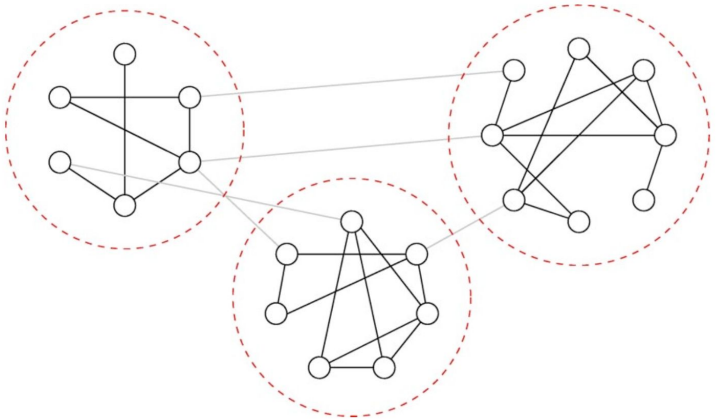


Figure 1: A network with strong community structure. *Modularity*, the measure of strength of community structure, which ranges from 0 to 1, has a value of 0.493 for the given division of nodes in this graph.

Two types of discussions on the development mailing lists

“Product” – development activity, function interfaces, APIs, bug fixes, feature implementation, etc.

“Process” – policy decisions, high-level architectural changes, release plans, licensing issues, and admission of newcomers.

Hypothesis 2 – Social networks constructed from product-related discussions will be more modular than those relating to non-product related discussions or all discussions.

The subcommunities should be related to the software engineering activities in a meaningful way.

Hypothesis 3 – Pairs of developers within the same subcommunity will have more files in common than pairs of developers from different subcommunities.

Hypothesis 4 – The average directory distance between files committed to by developers in the same subcommunity will be less than similar sized groups of developers drawn different subcommunities.

Mining the developer mailing list archives and source code repositories for a set of popular OSS projects.

Name	Apache	Ant	Python	Perl	PostgreSQL
Begin Date	1995-02-27	2000-01-12	1999-04-21	1999-03-01	1998-01-03
End Date	2005-07-13	2006-08-31	2006-07-27	2007-06-20	2007-03-01
Messages	101250	73157	66541	112514	132698
List Participants	2017	1960	1329	3621	3607
Files	1092	7682	4290	13308	6083
Developers	57	40	92	25	29
Commits	28517	58254	48318	92502	111847

Table 1: **Information on the data gathered for the projects studied.**

Finding community structure

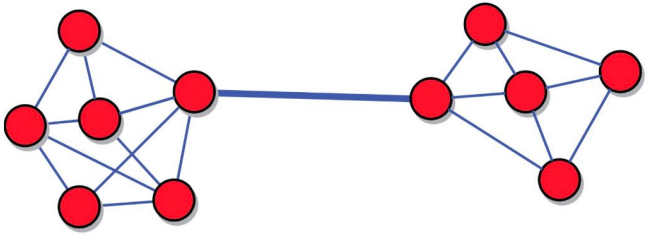
“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

Finding community structure

“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

3.1. Bridge removal

Key idea: Find links with high betweenness and remove them.



Link betweenness defined similarly to node betweenness centrality in previous lecture – fraction of shortest paths that run through that link.

Link betweenness should be higher for bridges than for links inside a cluster.

36

Finding community structure

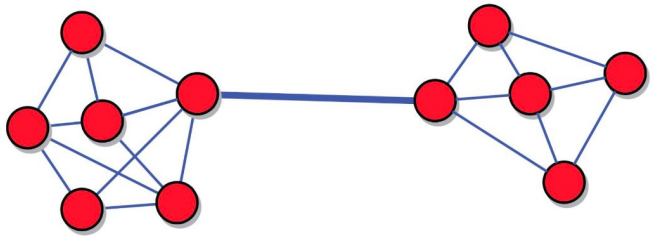
“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

Girvan-Newman algorithm (similar to hierarchical clustering)

We start by calculating the betweenness for all links. Then, each iteration of the algorithm consists of two steps:

1. Remove the link with largest betweenness; in case of ties, one of them is picked at random.
2. Recalculate the betweenness of the remaining links.

The procedure ends when all links are removed and the nodes are isolated.



38

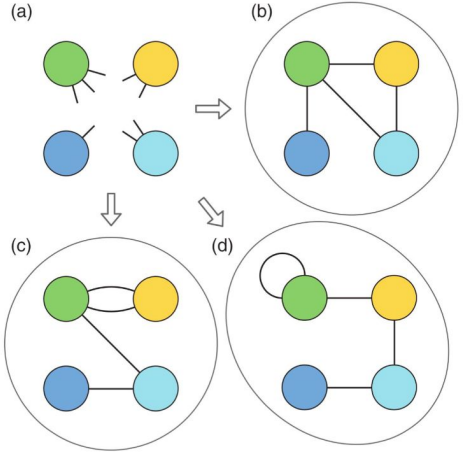
Finding community structure

“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

Modularity

The difference between the number of links internal to all clusters and the expected equivalent number in a randomized network.

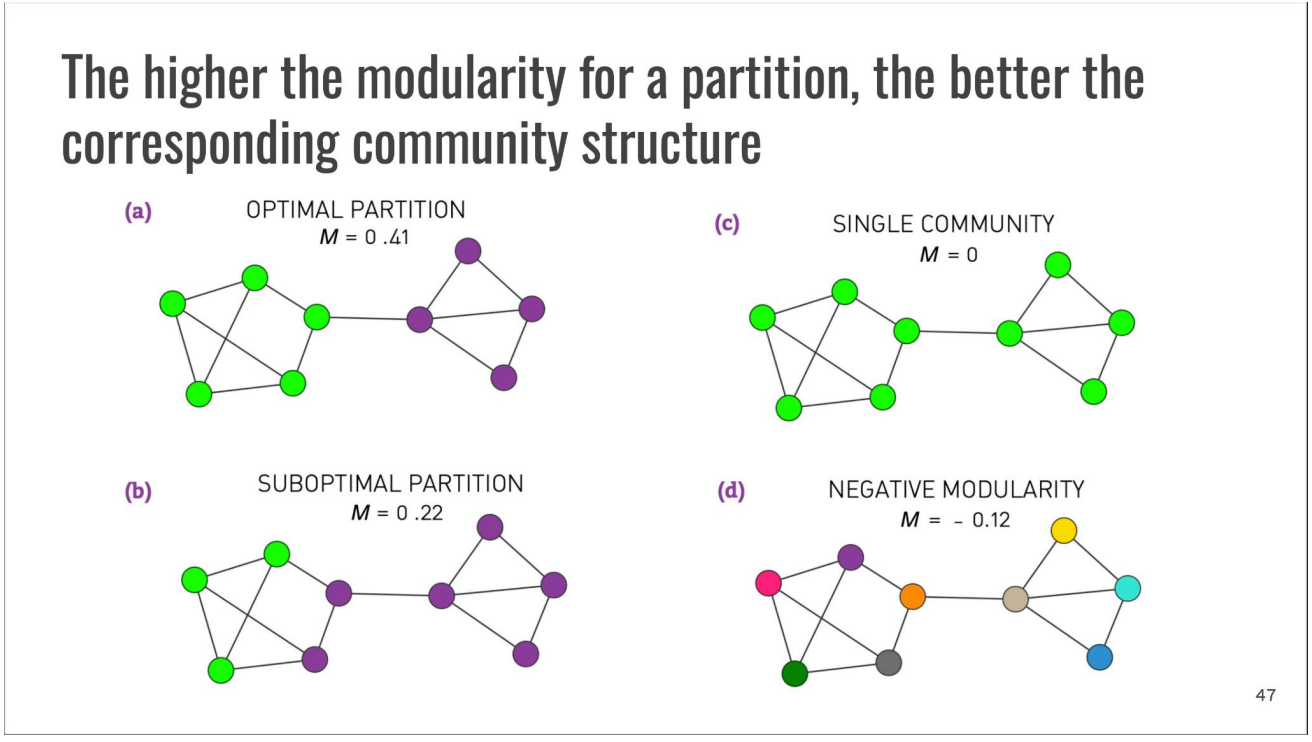
Randomization strategy: maintain number of nodes and degree sequence, shuffle links.



42

Finding community structure

“To find and quantify the latent community structure that exists in the OSS networks, we have created a variant of the Newman algorithm.”

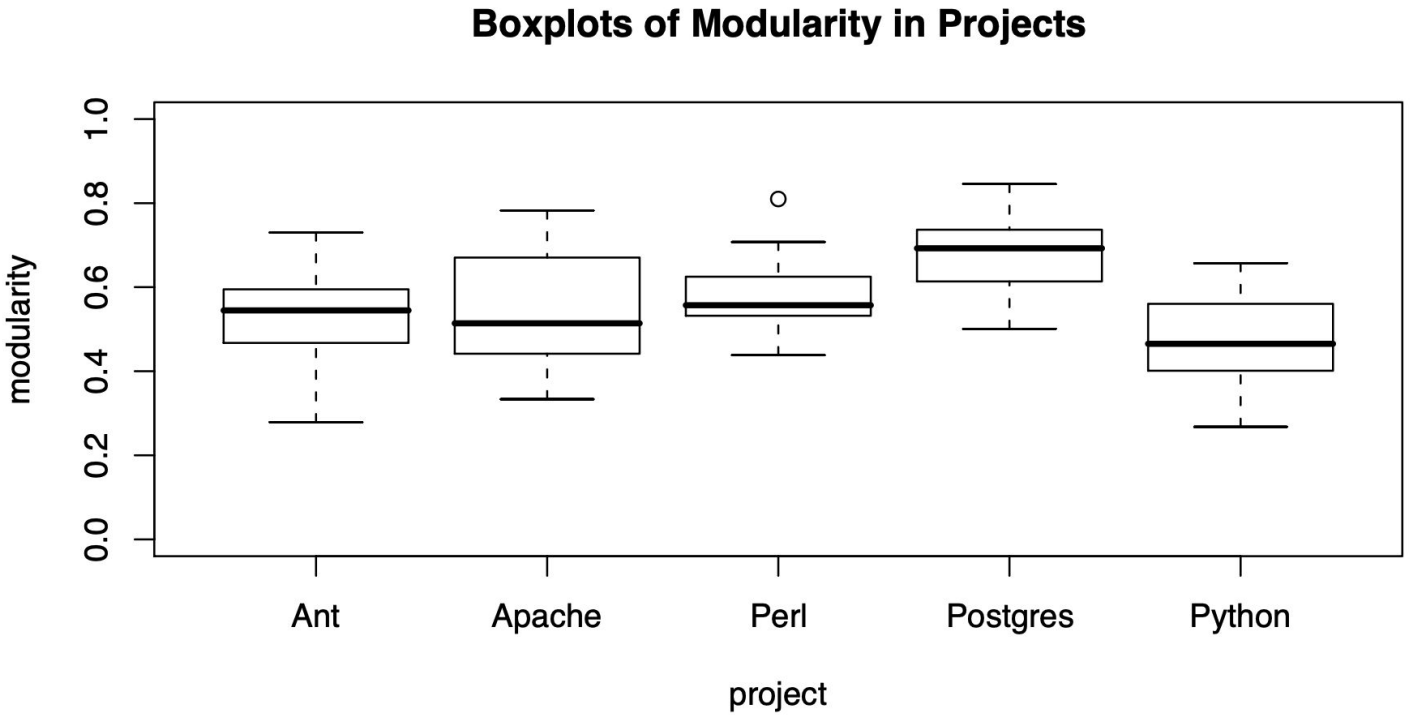


Finding community structure

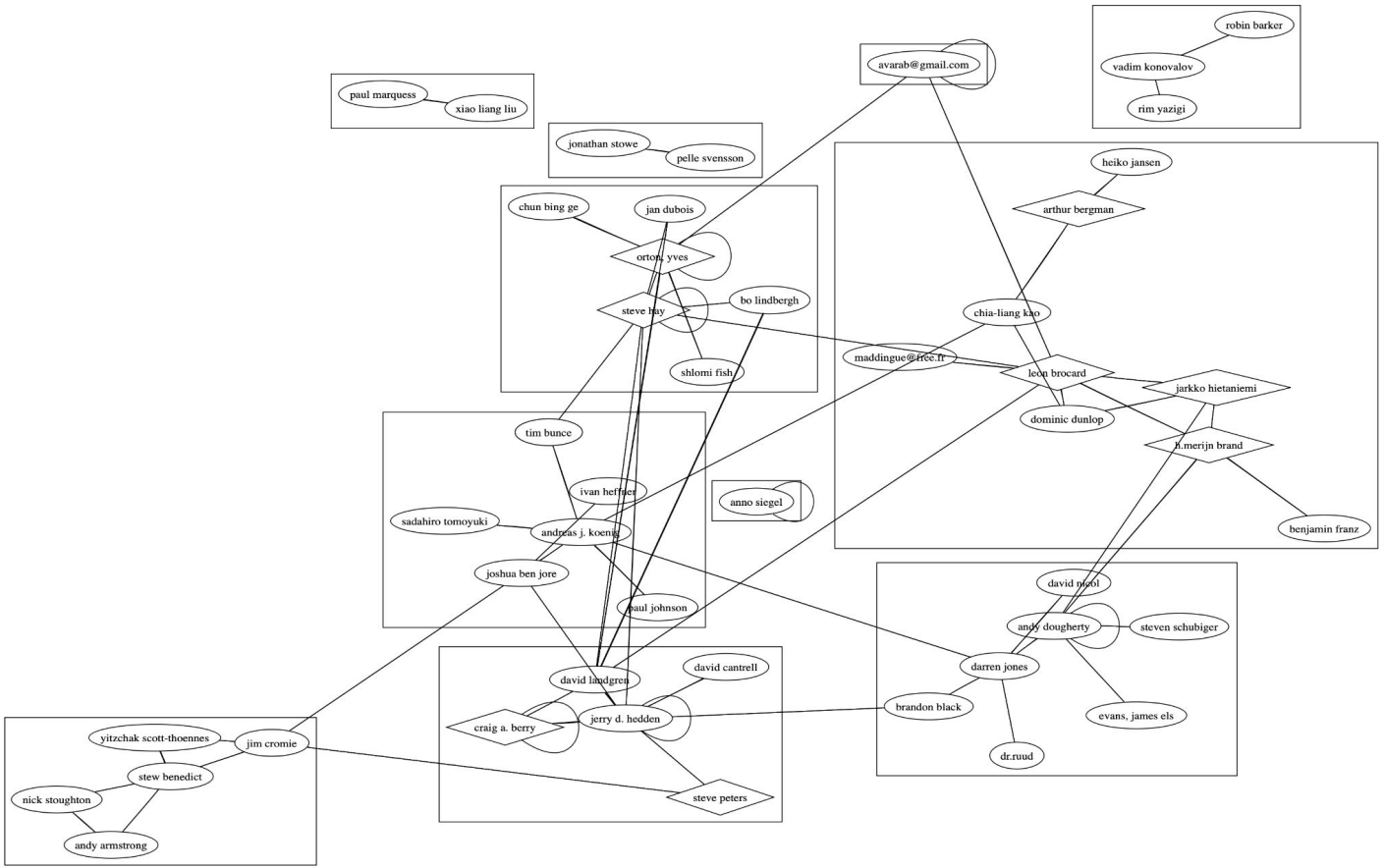
“Girvan and Newman’s original algorithm [...] doesn’t handle networks with weighted edges. Our social networks contain weighted edges, representing the number of emails exchanged between two participants in each time period. A high number of messages between a pair of participants should increase their likelihood of being in the same group.

[...] we modified our social networks by introducing one edge between each pair of nodes per email sent between them (i.e. creating a multi-edge network) and modified Newman’s algorithm above to handle multi-edge networks.”

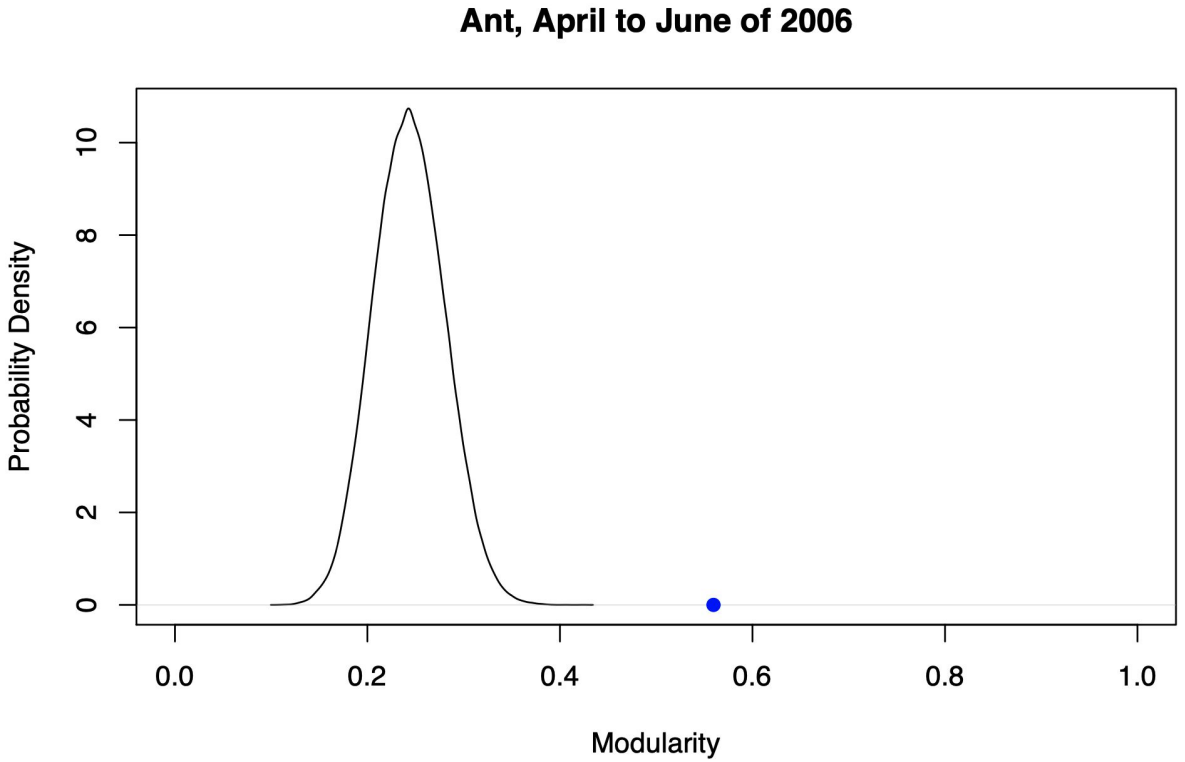
Community structure exists



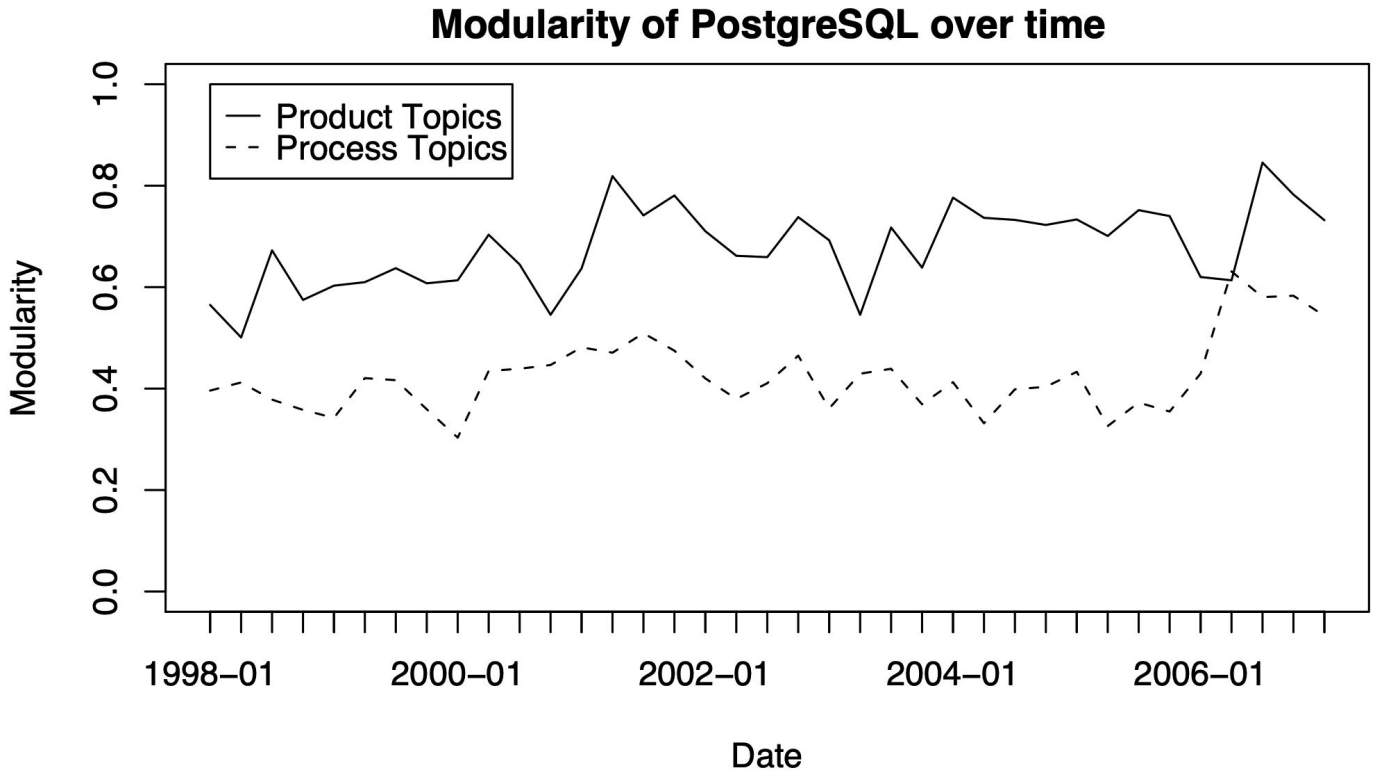
The community structure of Perl from April to June 2007



The distribution of modularity values for 100,000 random graphs with the same degree distribution as the observed network.

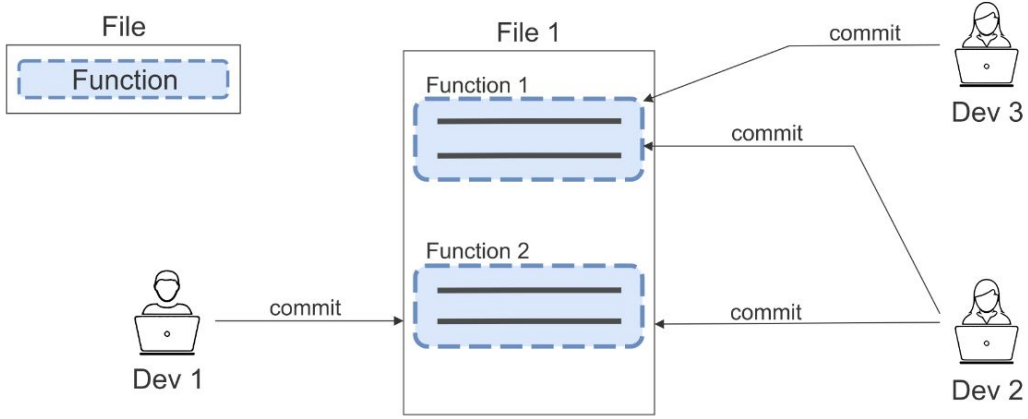


Grouping into subcommunities is much stronger for discussions directly related to the source code.

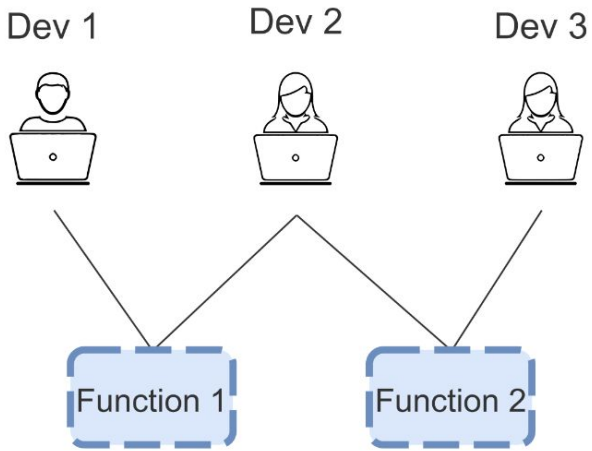


“From developer networks to verified communities: A fine-grained approach” – Joblin et al, ICSE 2015

Developer activity (a) recorded in a version control system at the granularity of functions is abstracted as a two-mode network (b)



(a) Developer Activity



(b) Two-mode Network

(Joblin & Apel, TOSEM 2022)

File-based vs function-based community detection

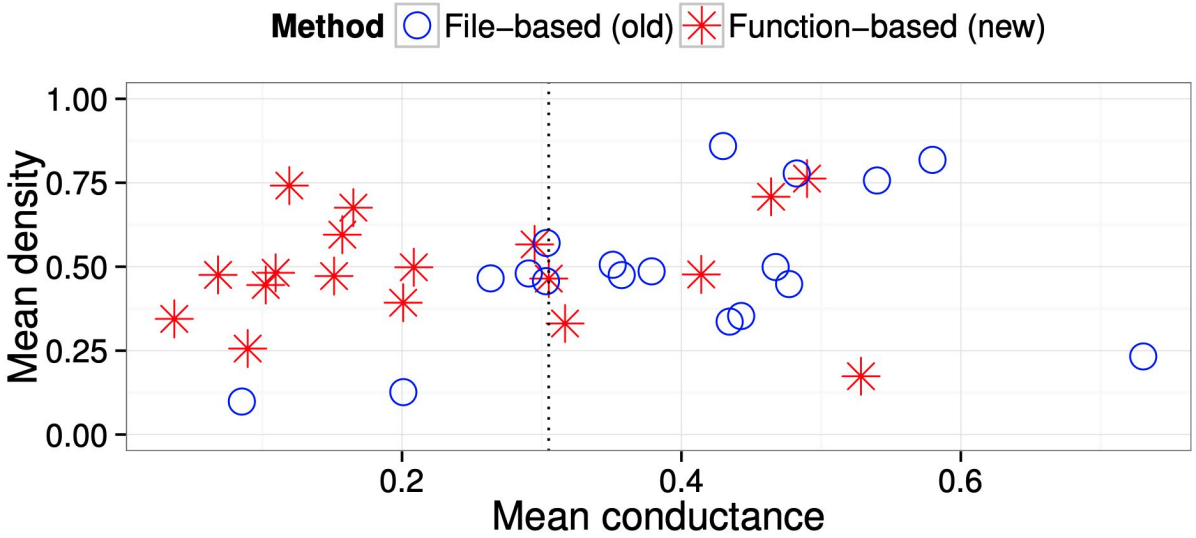


Fig. 2: Scatter plot of projects analyzed using both file-based and function-based methods for two different revisions. A clustering by crosses (left) and circles (right) is visible; the function-based approach is able to resolve more significant communities without compromising density.

“Validity of Network Analyses in Open Source Projects” – Nia et al, MSR 2010

OSS communication and coordination networks

“One can derive social networks from the online mailing list archives.

The nodes are the people sending messages on the list.

If a person A replies to a message from another person B, then there is an edge connecting the node representing A to that representing B.”

Incorrect information flow due to temporal aggregation

How much temporal data aggregation can be tolerated before SNA results become unreliable?

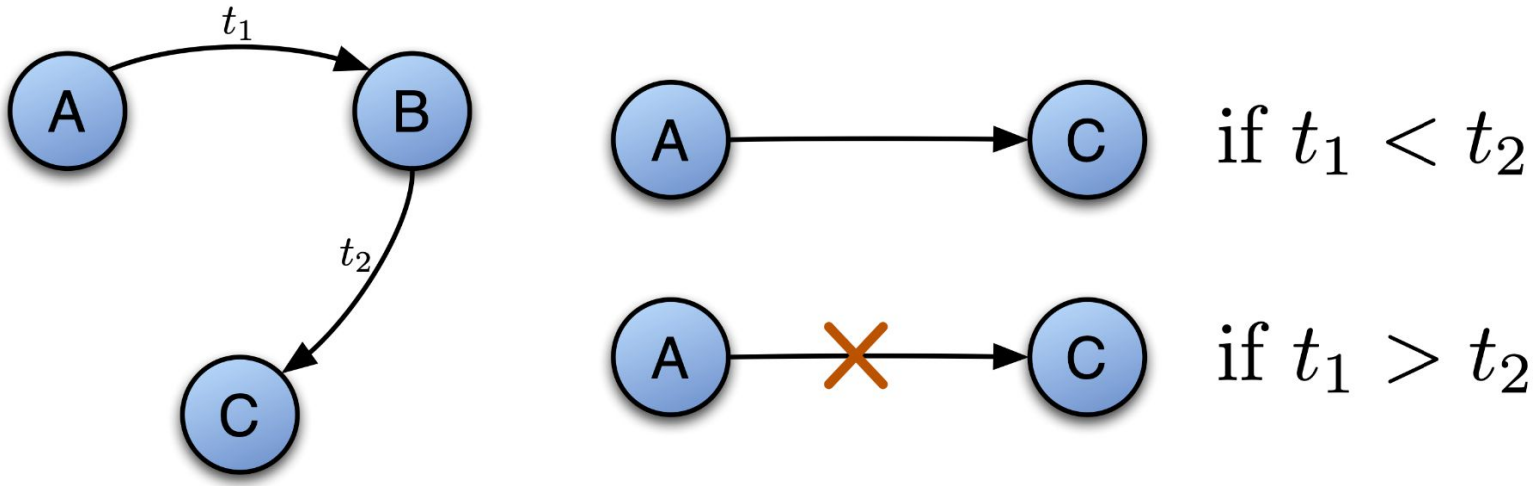


Fig. 1: The same topology, left, may apply to two different cases based on the order in which the messages were posted. If $t_1 < t_2$, then information can flow from A to C. But if $t_1 > t_2$ no information can flow from A to C

Information flow in the presence of inadequate or missing data

“Typically, social networks are derived from mailing list archives, using the ‘reply-to’ field in messages.

[...] If B read’s a message posted by A, but does not reply, then there is information flowing from A to B, but there is no way for us to know that.”

To what extent does missing data influence SNA metrics?

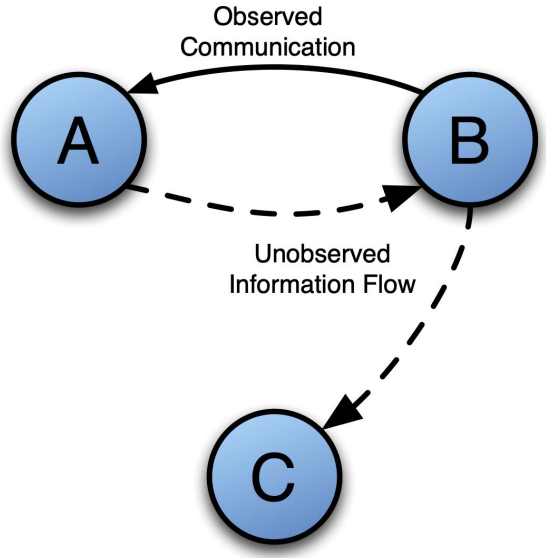


Fig. 2: Observed communication (solid edges) is evidence of information flow from B to A. However, C may read B’s message and B may have read A’s response, which indicates unobserved information flow (dashed edges).

It doesn't matter?

"We find that while transitive faults can be as frequent as 50%, their frequency is highly dependent on the time interval of aggregation, and that even when very frequent, they do not change results from SNA analysis critically."

B. Network Measures

In this paper we use the following SNA measures.

- *Number of 2-paths (2P)* — The number of 2-paths through a node is a measure of local social status as defined previously [27].
- *Betweenness Centrality (BW)* — The betweenness centrality of a node is a function of the how many communication paths a node lies on and is often used a measure of global social status [28].
- *Clustering Coefficient (CC)* – The clustering coefficient measures the local connectivity density, or local structure in the graphs [29].

Summary

… to be continued

Tons of data and research opportunities in OSS, join us!

“Classifying developers into core and peripheral: An empirical study on count and network metrics.” – Joblin et al, ICSE 2017

“Core” vs “peripheral”

Core developers

- driving the system architecture
- forming the general leadership structure
- have substantial, long-term involvement

Peripheral developers

- typically involved in bug fixes or small enhancements
- have irregular or short-term involvement

“Core” vs “peripheral”

Core developers

- driving the system architecture
- forming the general leadership structure
- have substantial, long-term involvement

Peripheral developers

- typically involved in bug fixes or small enhancements
- have irregular or short-term involvement

Distinction based on activity level – typically, the top 20% of contributors are responsible for 80% of the contributions.

Core and peripheral developers in developer networks

Degree centrality – local importance

Eigenvector centrality – global importance by either connecting to many developers or by connecting to developers that are themselves globally central

Hierarchy – core developers should have a high degree and low clustering coefficient, placing them in the upper region of the hierarchy

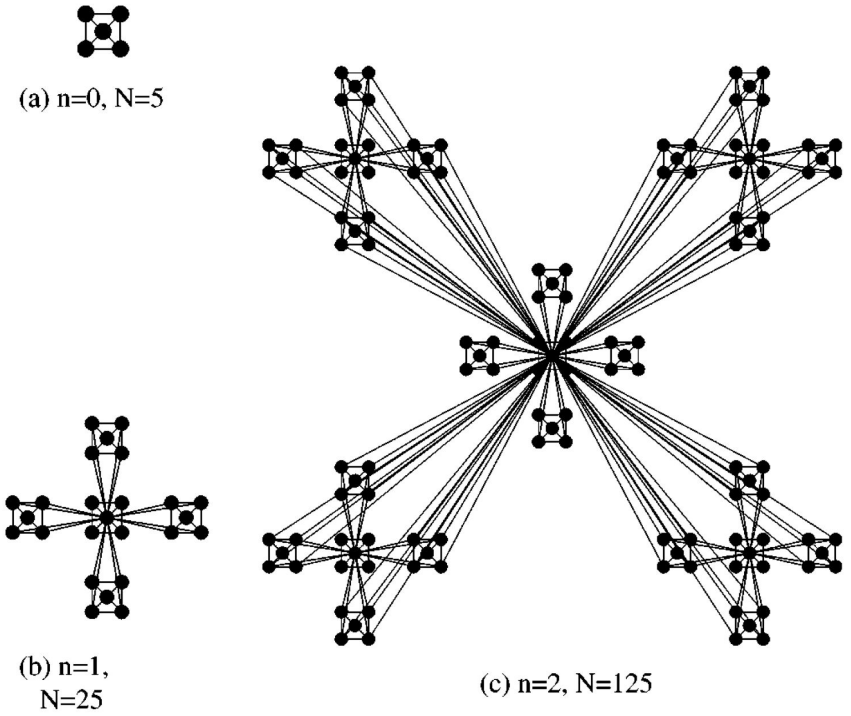
Core–peripheral block model – the core–core region of the matrix is a 1-block (i.e., completely connected), the core–peripheral regions are imperfect 1-blocks, and the peripheral–peripheral region is a 0-block

Aside: The hierarchical network model

Recall the earlier scale-free property vs clustering discussion.

Small world model – short paths, clustering, but no hubs.

Preferential attachment – short paths, hubs, but not enough clustering.



(Ravasz & Barabasi, 2003)

Aside: The hierarchical network model

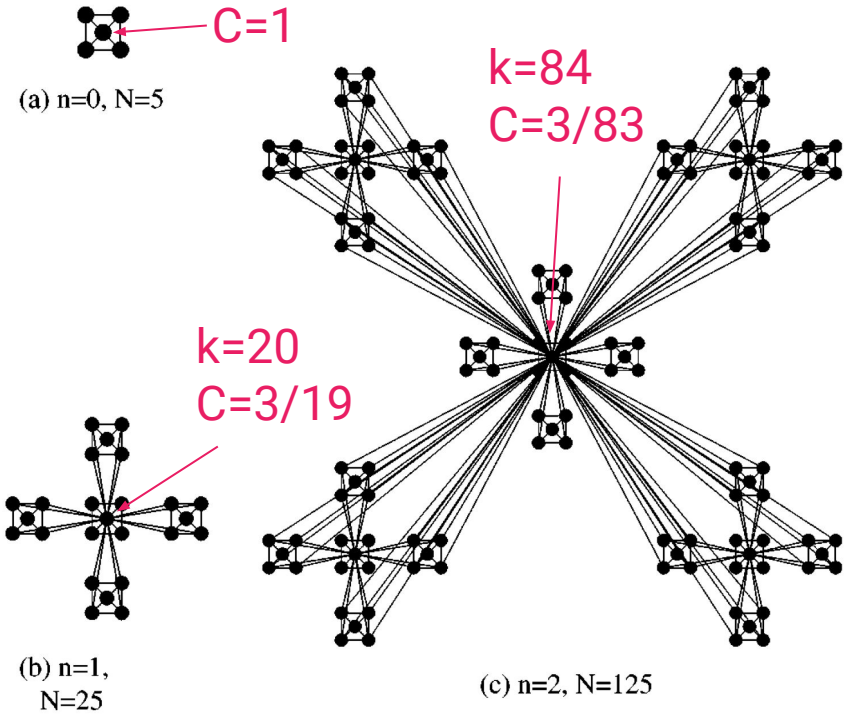
Hierarchical network: scale-free property & high degree of clustering.

Example on the right:

- power-law degree distribution with degree exponent γ = 2.16
- clustering coefficient C = 0.74 is independent of network size
- hierarchical architecture

Scaling law for clustering coefficient:

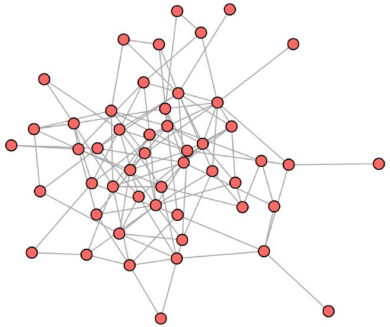
$$C(k) \sim k^{-1}$$

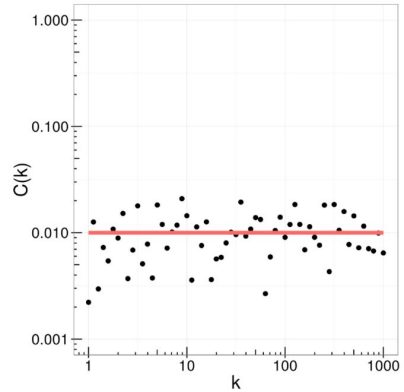


(Ravasz & Barabasi, 2003)

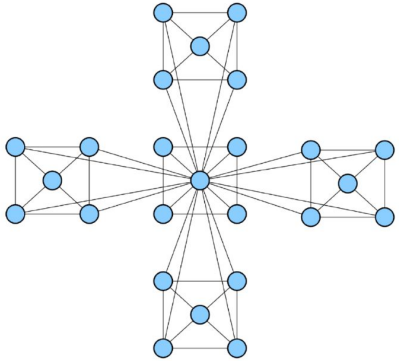
Aside: The hierarchical network model

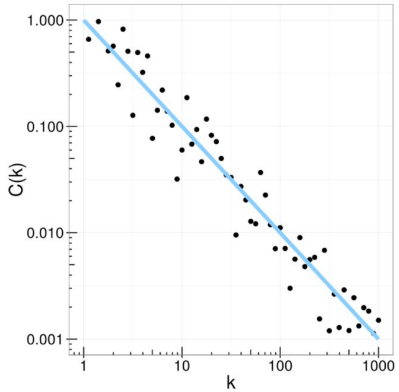
ER Random Network





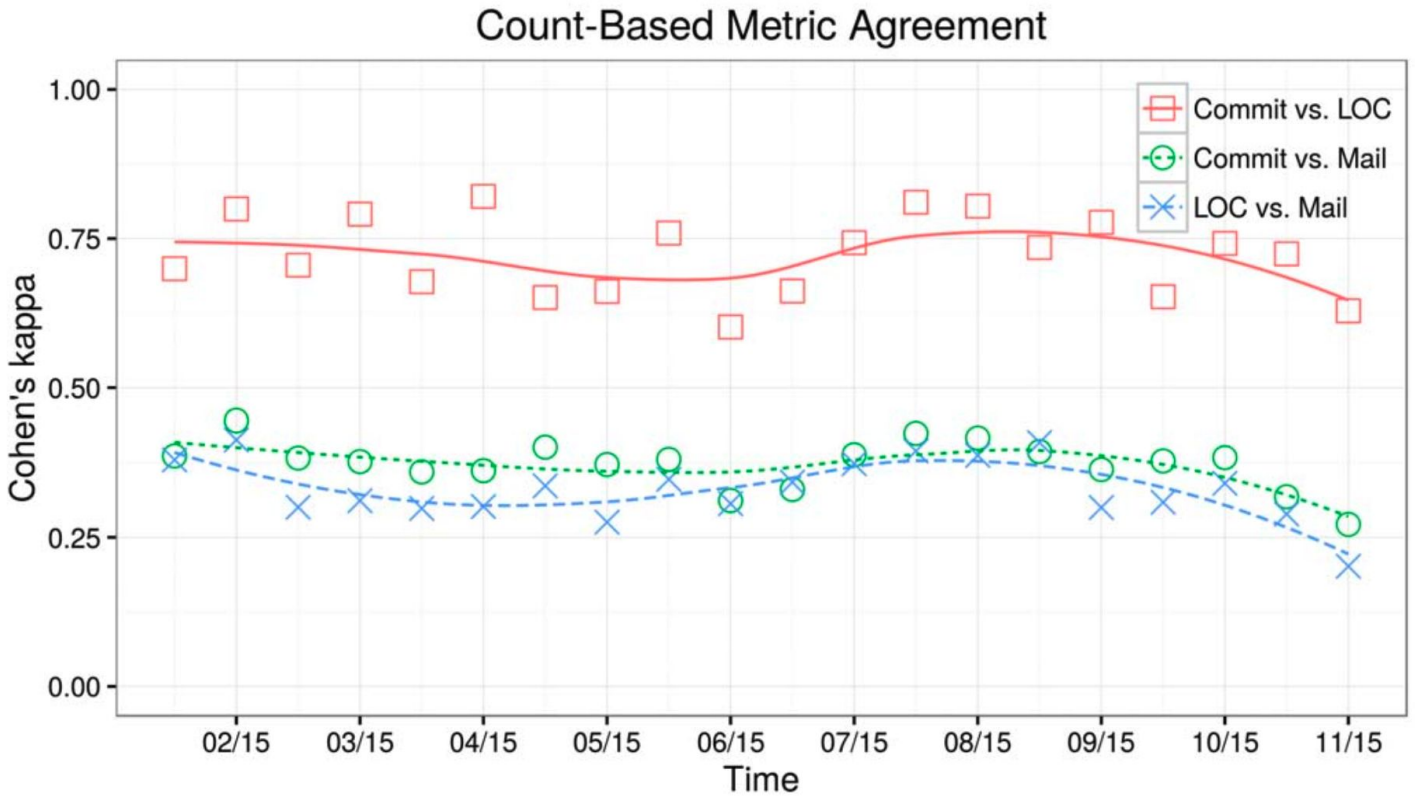
Hierarchical Network





(Joblin et al, TOSEM 2023)

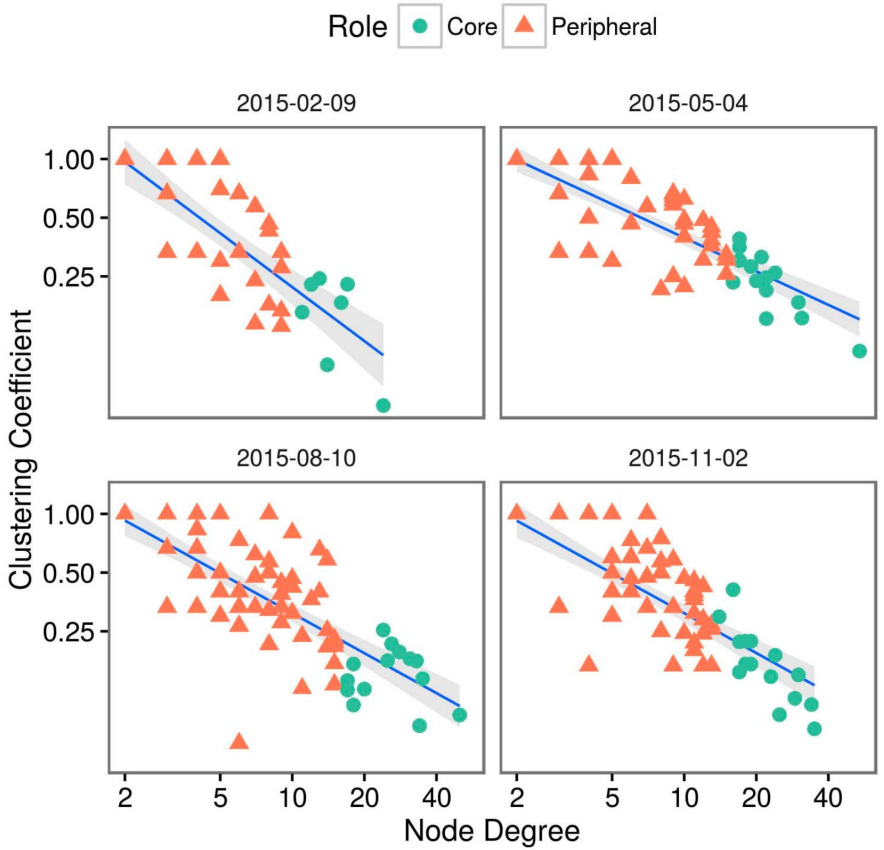
Agreement between count metrics is fair to substantial.



The linear dependence between clustering coefficient and degree expresses the hierarchy.

Also block model:

$p_{core-core} > p_{core-periph} > p_{periph-periph}$



Network-based and count-based operationalizations are mostly consistent.

Also, the network perspective always improves the agreement with developer perception over the simple count-based operationalizations.

TABLE II: Agreement with developer perception

		Cohen's kappa	p value
Counts	Commit Count	0.387	3.12e-06
	LOC Count	0.355	1.91e-05
	Mail Count	0.421	2.08e-05
Networks	VCS Degree	0.465	4.48e-08
	VCS Hierarchy	0.437	2.22e-07
	VCS EigenCent	0.404	1.74e-06
	Mail Degree	0.497	8.23e-07
	Mail EigenCent	0.427	1.26e-05

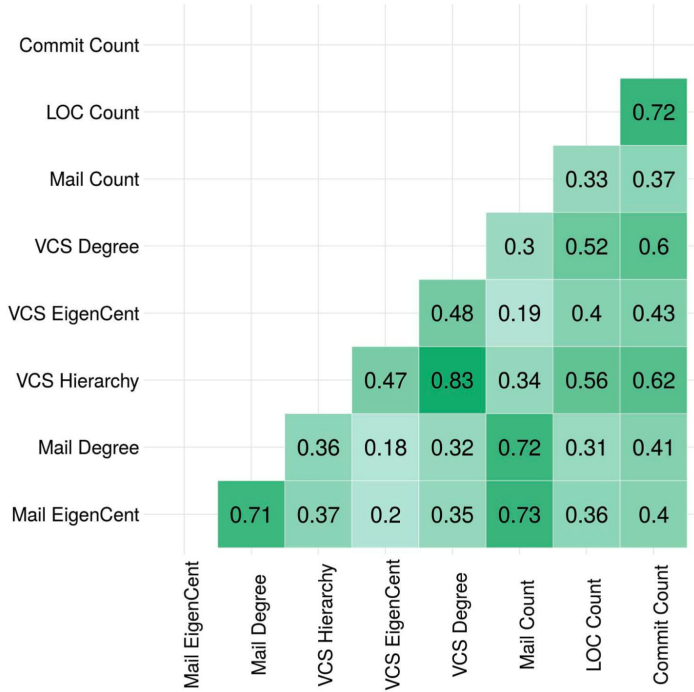
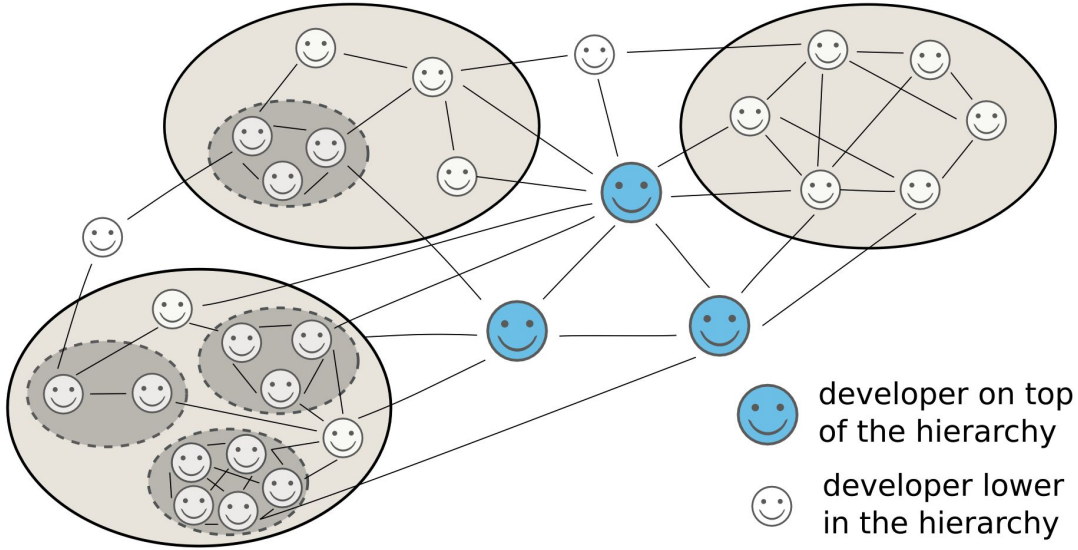


Fig. 4: Time-averaged agreement in terms of Cohen's kappa for QEMU. The pairwise agreement is shown for the count-based and network-based operationalizations

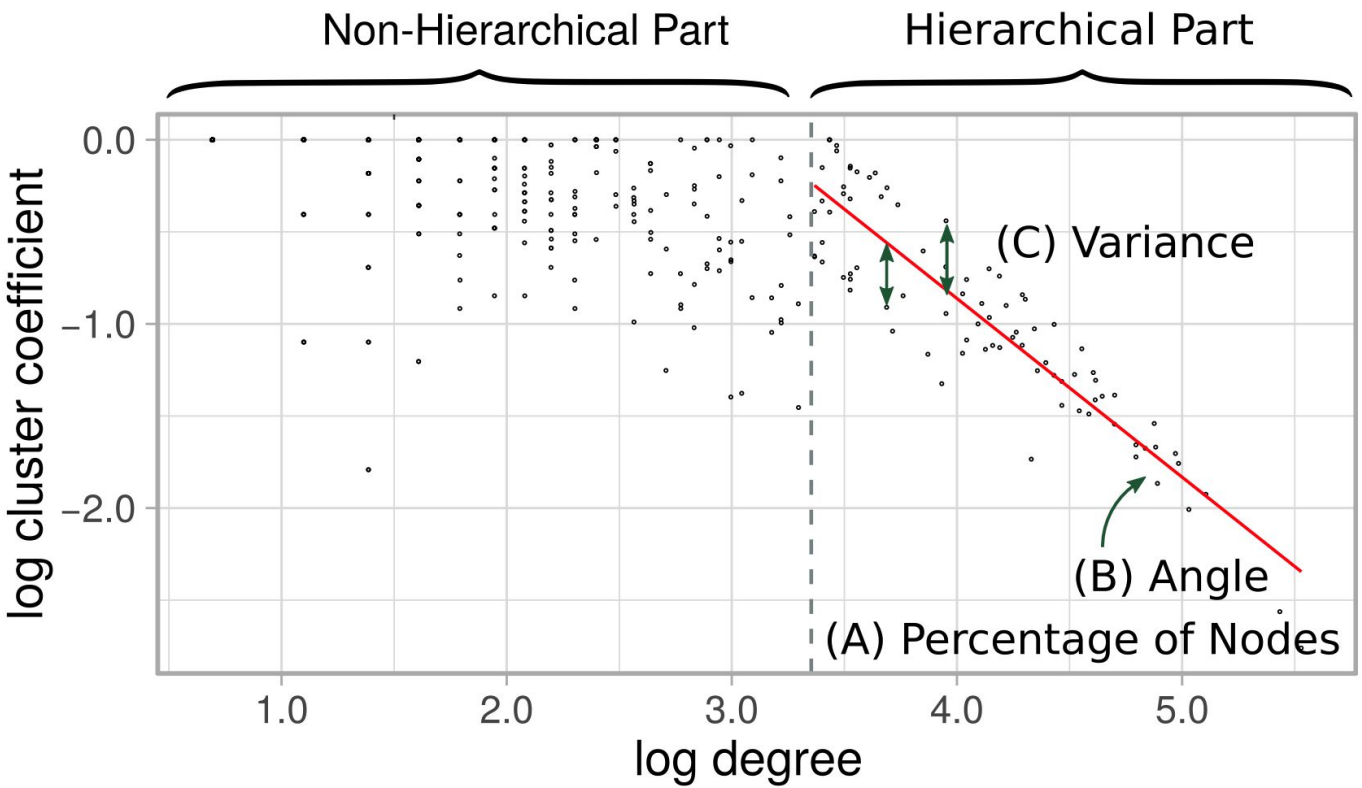
“Hierarchical and Hybrid Organizational Structures in Open-source Software Projects: A Longitudinal Study” – Joblin et al, TOSEM 2023

Hierarchical structure emerges in OSS projects

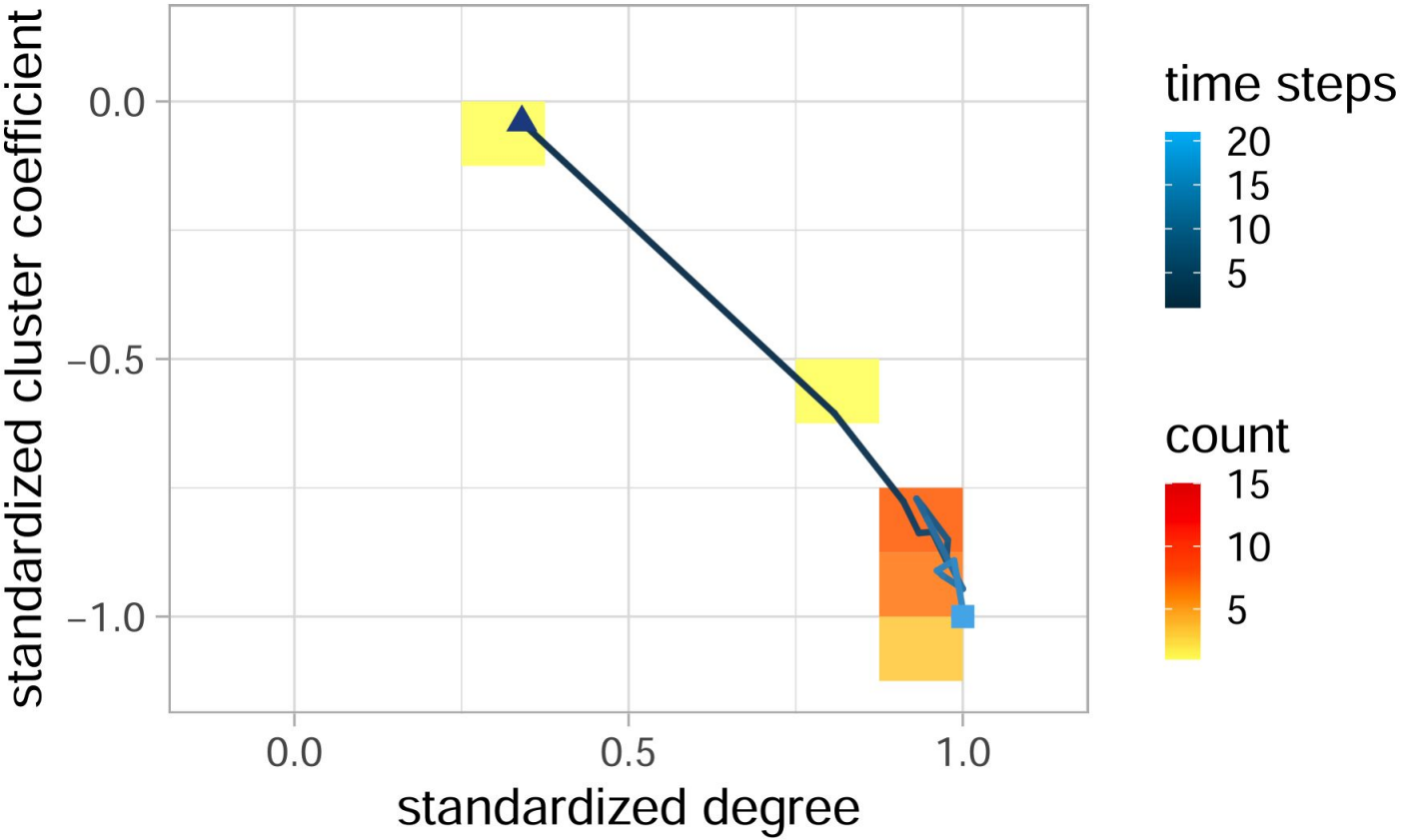
Affords both the scale-free property and the community property.



Or, rather, a hybrid organizational structure



Over time, shift from non-hierarchical part to hierarchical part



Summary

Tons of data and research opportunities in OSS, join us!