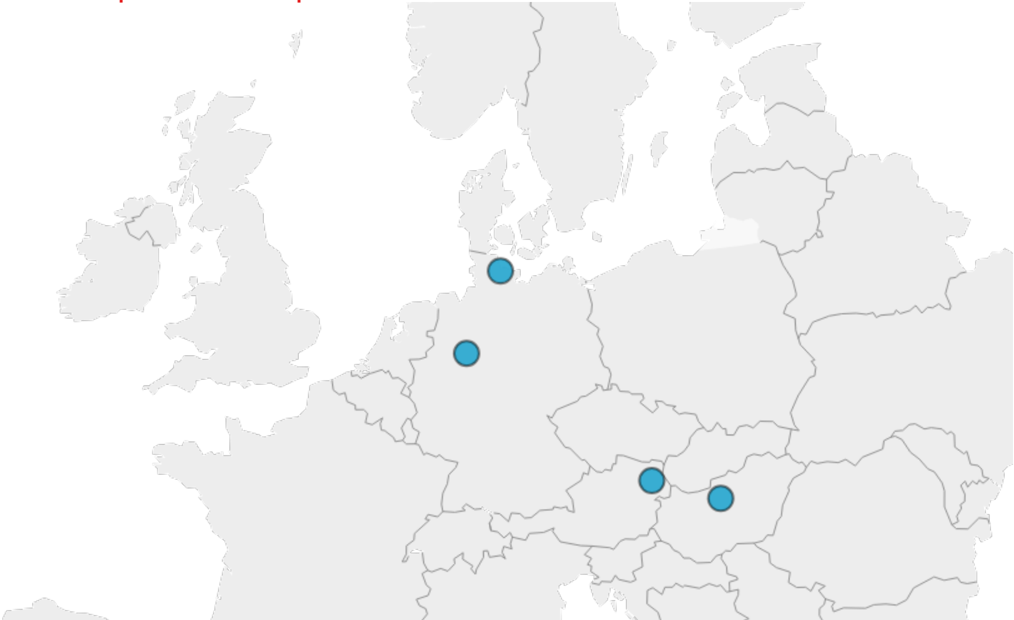
When dispersed teams are more successful: Theory and evidence from software

Gábor Békés (CEU, HUN-REN KRTK, and CEPR) Julian Hinz (Bielefeld University and Kiel Institute for the World Economy) Miklós Koren (CEU, HUN-REN KRTK, CEPR and CESifo) Aaron Lohmann (Bielefeld University and Kiel Institute for the World Economy)

September 26, 2024[1]

[1]This work was funded by the European Union under the Horizon Europe grant 101061123. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Authors’ dispersion in space



Introduction

Research questions

- How and where is open source software developed?
- Can spatially dispersed developers produce quality software?
- How do frictions affect collaboration and software quality?

Two economic puzzles in open source

Why do people work for free?

Altruism, reputation concerns, alternative business models. Sizeable economic literature.

How can spatially dispersed developers produce quality software?

GitHub poll

```
if (poll == "no") {
```

Why Open Source Software (OSS)?

OSS is huge

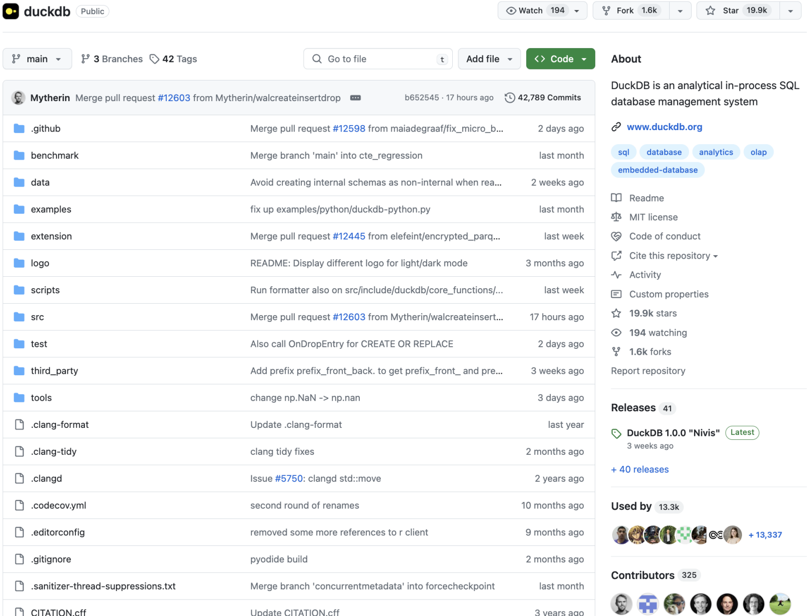
- Software industry – 1% of global GDP
- 90+% of software has open source components

OSS is everywhere

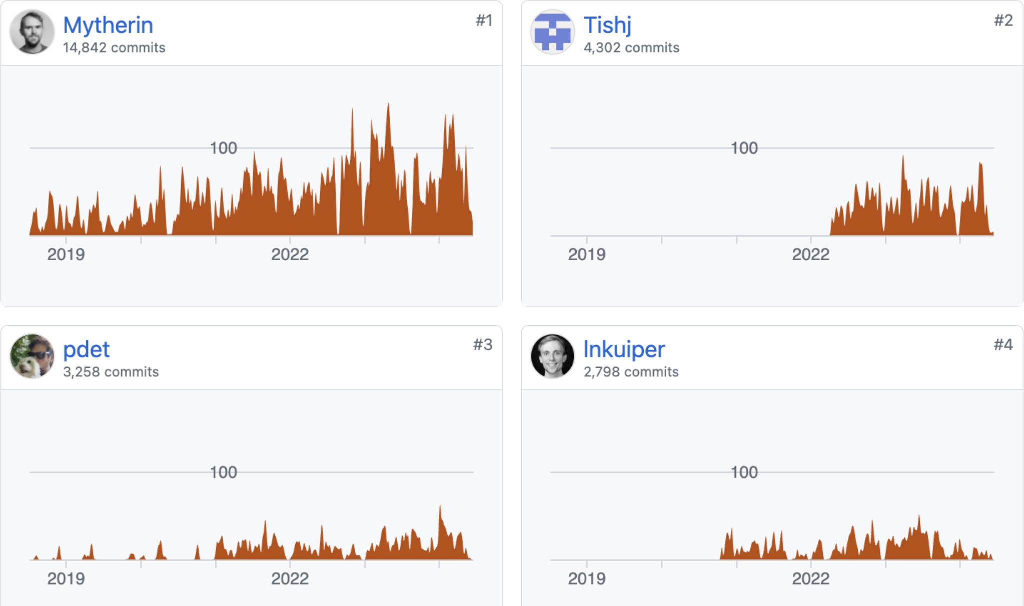
OSS plays an important roles in - Websites (PHP, JavaScript) - Operating systems (Linux, Android) - Data (R Tidyverse, Python Pandas, Julia) - Machine Learning and AI (PyTorch, LLaMA)

OSS is observable

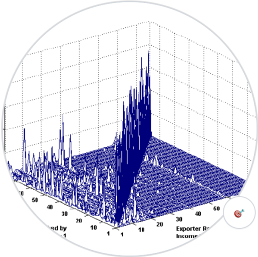
A platform for sharing and discussing code



Not all developers contribute equally



(Many) developers report their location



Mike Waugh
mwaugh0328

Unfollow

Economist and Monetary Advisor at the Minneapolis Fed. Creator/developer of **@tradewartracker**

247 followers · **67** following

Followed by **johnjosephhorton** and 3 more

Federal Reserve Bank of Minneapolis

Federal Reserve Bank of Minneapolis

http://www.waugheconomics.com/

mwaugh0328 / README.md

About me

I'm an Economist and Monetary Advisor at the Federal Reserve Bank of Minneapolis. My research interests lie within the intersection of international trade, macroeconomics, and development. And I'm interested in the use of computational tools to answer quantitative questions in these domains.

About my repositories

I use GitHub to post code I'm working on (and learning about), replication materials from selected papers, and past teaching. I also use GitHub and Heroku to operate the website www.tradewartracker.com providing live, visual display of international trade data. Below are selected repositories.

- Tradewar Tracker Repository for code behind www.tradewartracker.com website.
- Repository to replicate aspects of **The Elasticity of Trade: Estimates and Evidence**, with Ina Simonovska. Journal of International Economics, 92(1): 34-50. January 2014.
- My Garvity-Estimation repository suplements the JIE repo with basic gravity estimation via STATA and then the computation of the Eaton and Kortum model via simmulation. A Julia version is coming soon.
- Repository for **Equilibrium Technology Diffusion, Trade, and Growth** with Jesse Perla and Chris Tonetti. American Economic Review 111 (1), January 2021.
- Repository for **The Welfare Effects of Encouraging Rural-Urban Migration**, with David Lagakos and Mushfiq Mobarak. Econometrica, Vol. 91 (3). May 2023.
- Data Bootcamp Repo (2019 edition) from the course I taught at NYU Stern economics.
- Economics of Global Business (2019 edition) from the course I taught at [NYU Stern economics].

🔭 I'm currently learning / working on...

Open Source vocabulary

`Project`: A software project offering solution to a use case, a.k.a. library, package.

`Repository`: A storage for one project (what we observe)

`Commit`: The smallest unit of contribution

`Git`: Distributed version control system for software projects

`GitHub`: A platform to collaboratively work on software projects

`Dependency`: An imported project that provides a functionality

}

Related literature

- **Geographical Distance / Network formation / Agglomeration**: [@chaney2014network] [@bernard2019production] [@davis2019spatial] [@BaileyGuptaHillenbrandEtAl2021], [@Atkin_2022_F2F]
- **Gravity: Digital:** [@blum2006does] [@anderson2018dark]
- **Frictions in services:** [@stein2007longitude] [@bahar2020hardships]
- **Patents and science**: [@BircanJavorcikPauly2021], [@head_li_minondo_math_2019], [@jaffe1993geographic], Singh (2008) [@AlShebli_nature_2018], [@Li2014-patents-eer]
- **OSS**: [@lerner2002some] , [@Laurentsyeva:2019] [@Wachs_etal_2022] [@fackler_hofmann_laurentsyeva_2023]

Outline

1. Data and stylized facts about OSS production
2. A model of global team formation and collaboration
3. Test(able) implications

Stylized facts

Data

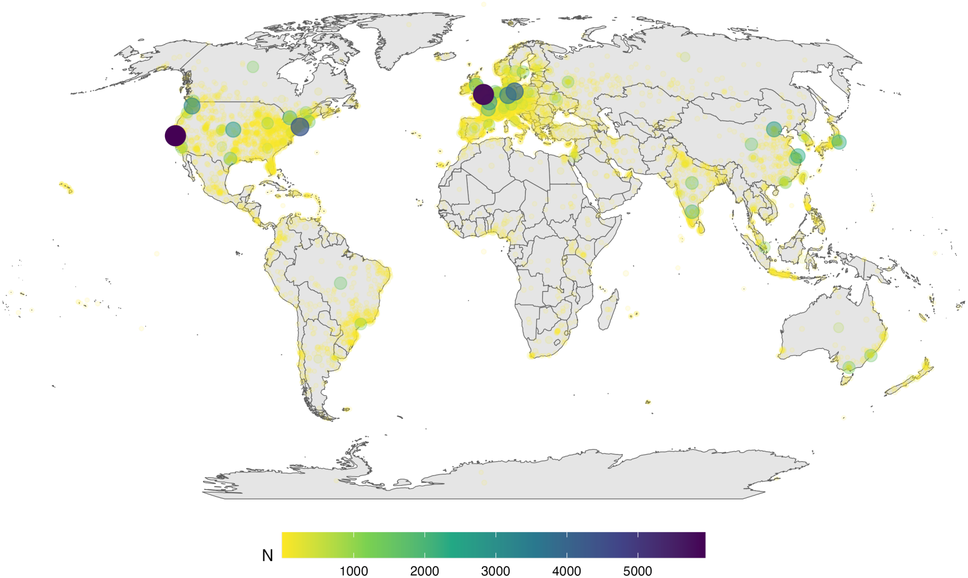
GitHub

Snapshot of all public repositories on GitHub on 2019-06-01. Six largest languages: JavaScript, Python, Java, Ruby, PHP, and C++. Drop smallest and largest projects. 4.4m projects, 2.7m users. Self-reported location for about 1/3 os users.

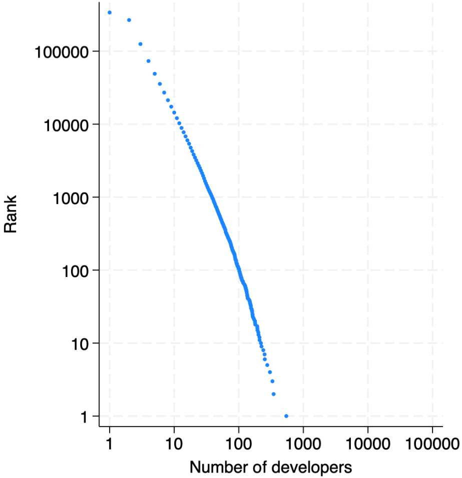
libraries.io

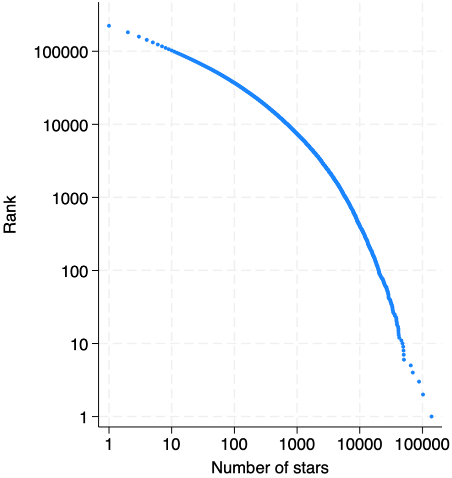
Dependency data for projects on major package managers (npm, PyPI, Maven, RubyGems, etc).

JavaScript developer density around the globe

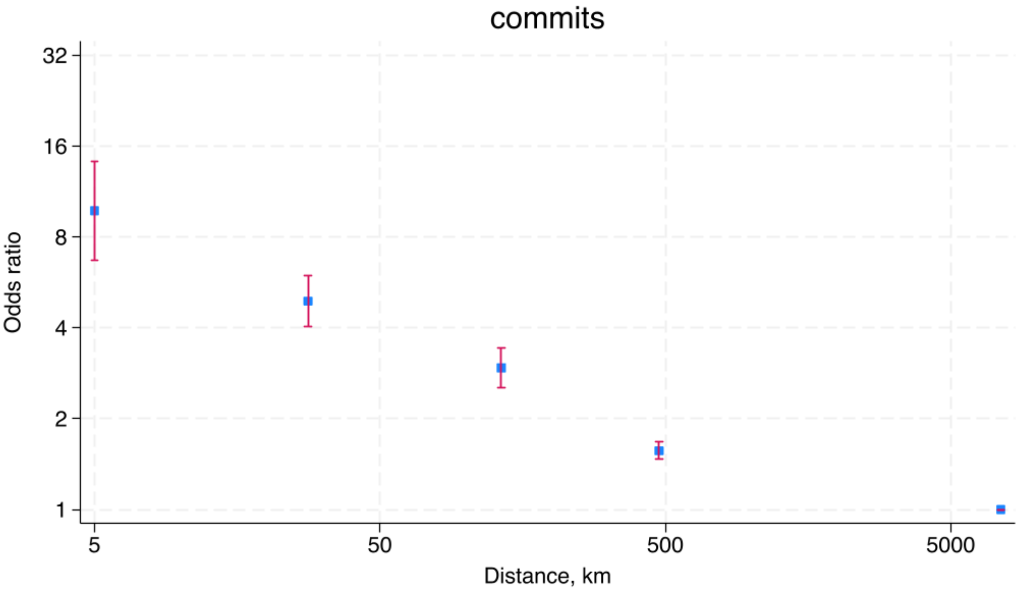


Project size and popularity

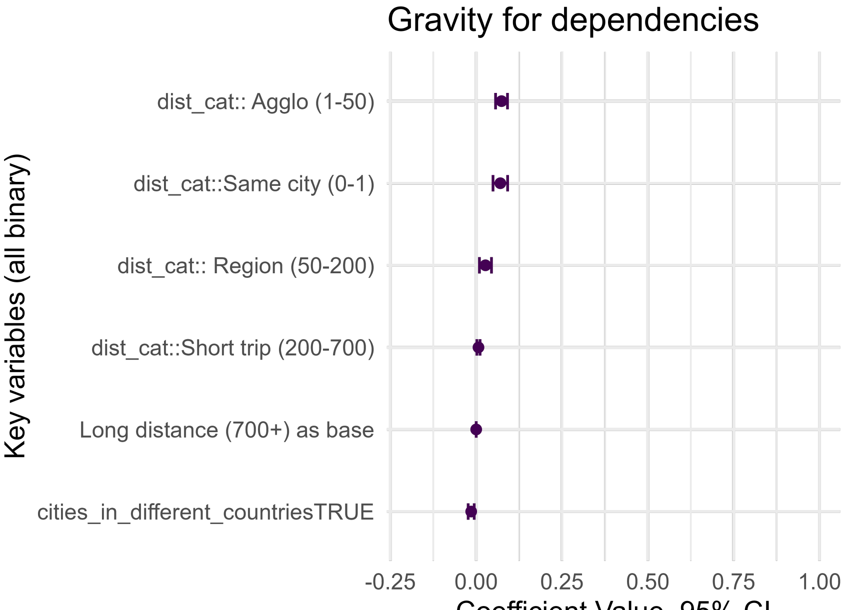




Closer developers are more likely to contribute to the same project



There is no distance penalty for *using* other's software



Better developers build more popular software, but developers' skills are substitutes

Dependent Variable:	n_stars			
Model:	(1)	(2)	(3)	(4)
Variables				
log(stars_lead)	0.4707***		0.4446***	0.5086***
	(0.0260)		(0.0535)	(0.0618)
log(stars_follow)		0.2746***	0.1600***	0.2533***
		(0.0447)	(0.0207)	(0.0410)
log(stars_lead) × log(stars_follow)				-0.0192***
				(0.0073)
Fit statistics				
Observations	17,906	17,339	3,348	3,348
Pseudo R^2	0.22550	0.08336	0.29024	0.29179

Model

Model questions

We take OSS payoffs as given.

higher software quality $\rightarrow$ more payoff ("kudos")

1. How do teams form?
2. How do they collaborate?
3. How do they distribute kudos?

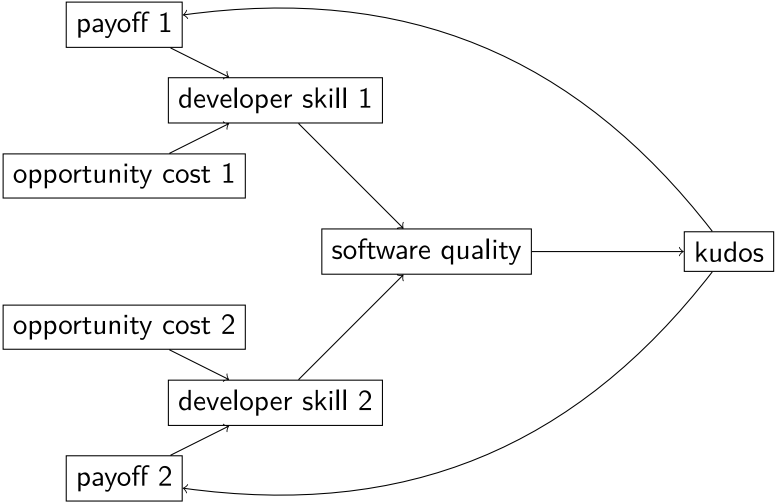
Primitives

- Software developers vary in location, skill Z_i and preference for fame ξ_i.
- Fixed supply of developers at each location.
- Team formation as well as collaboration across locations are costly.

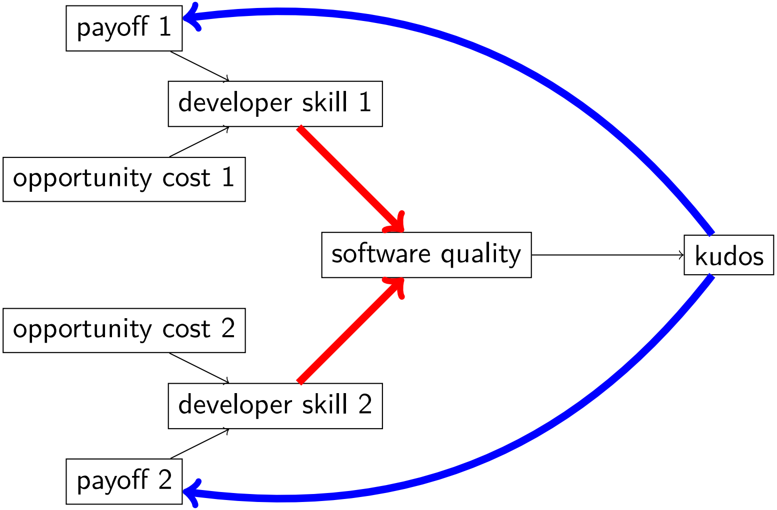
Timing

1. Two developers meet at random
   - partially observe each other's skills
2. Decide whether to do a project together
   - If not, enjoy outside option.
3. Software is developed to a certain quality.
4. Users download it, distributing kudos to developers.

Model outline



Spatial frictions



Team composition

Project p is developed by two developers with skills Z_1 and Z_2.

Developer skill drawn from Fréchet distribution:

$$\Pr(Z_i \leq x) = e^{-T_{ip}x^{-\theta}}$$

Developer skill

T_{ip} observable (programming language, years of experience, etc.)
$1/\theta$ captures importance of unobservable skill

Software production function

Software quality depends on the best idea:

$$X_p := \max\{Z_{1p}, Z_{2p}/\tau_{2p}\}$$

Knowledge sharing cost

$\tau_{ip} \geq 1$. Not all good ideas are heard (language, time zone, culture, clarity).
Normalize $\tau_{1p} = 1$ for presentation.

Gravity

$$\tau_{ip} = \text{distance}_{ip}^{\gamma_k}$$

Distribution of software quality

Software quality is also Fréchet.

$$\Pr(X_p \leq x) = e^{-\Phi_p x^{-\theta}}$$

with

$$\Phi_p := T_{1p} + \tau_{2p}^{-\theta} T_{2p}$$

Testable implications

1. Larger teams produce better software.
2. Better developers produce better software.
3. Knowledge sharing frictions reduce software quality.

Sharing kudos

Overall customer happiness increases in software quality:

$$V_p := e^{X_p}$$

Attribution of kudos

The better-skilled developer gets all the kudos for V_p. ($\approx$ "First author bias")

Expected developer payoff from project p

$$\mathcal{U}_{ip} = \begin{cases} e^{\xi_i Z_i/\tau_{ip}} & \text{if } Z_i/\tau_{ip} > Z_j/\tau_{jp} \\ 0 & \text{otherwise} \end{cases}$$

where ξ_i is a taste parameter for enjoying kudos. In expectation,

$$U_{ip} = \mathsf{E}\,\mathcal{U}_{ip} = e^{-T_{jp}\tau_{ip}^{\theta}Z_i^{-\theta}} e^{\xi Z_i/\tau_{ip}}$$

Increases in Z_i, decreases in T_{jp}, τ_{ip}.

Team formation

Does developer i join project p?

$$U_{ip}(Z_i, T_{jp}, \xi_i) > \text{cost}_i(Z_i, d_{ip}) := e^{d_{ip}\xi_i Z_i}$$

Distribution cost

$d_{ip} \geq 1$. Not all benefits of distant projects can be captured (private cost of participation, time zones, misappropriate of credit).

Gravity

$$d_{ip} = \text{distance}_{ip}^{\gamma_s}$$

where γ_s may be different from γ_k

Join team p if

$$Z_i > \frac{\tau_{ip} T_{jp}^{1/(\theta+1)}}{(\tau_{ip} d_{ip} - 1)^{1/(\theta+1)}} \xi_i^{-1/(\theta+1)}$$

Selection

1. Better skilled developers are more likely to join.
2. Spatial frictions reduce team formation.
3. Projects with high-skilled developers are more selective.

Fréchet magic

Assume Z_i is Fréchet with parameters T_i and θ,

ξ_i is Weibull with κ and $\theta/(\theta+1)$. Then

$$\Pr(Z_i \leq x | i \text{ joins project } p) = e^{-T_{ip}x^{-\theta}}$$

with

$$T_{ip} = T_i + \frac{1}{\kappa}\frac{\tau_{ip}^{\theta}T_{jp}^{\theta/(\theta+1)}}{(\tau_{ip}d_{ip}-1)^{\theta/(\theta+1)}}$$

Closing the model

Both developers want to join, knowing what to expect from the other.

Mutual coincidence of wants

$$T_{1p} = T_1 + \frac{1}{\kappa} \frac{T_{2p}^{\theta/(\theta+1)}}{(d_{1p} - 1)^{\theta/(\theta+1)}}$$
$$T_{2p} = T_2 + \frac{1}{\kappa} \frac{\tau_{2p}^{\theta} T_{1p}^{\theta/(\theta+1)}}{(\tau_{2p} d_{2p} - 1)^{\theta/(\theta+1)}}$$

Team forms with probability

$$\frac{T_1}{T_{1p}} \frac{T_2}{T_{2p}}$$

Testable predictions

Testable predictions

Gravity of team formation

1. Distant developers are less likely to join a team.

Knowledge production

2. Two-person projects are better than one-person projects.
3. Projects with better developers are more successful.
4. Project success depends disproportionately on "lead developer."

Assortative matching

5. Skilled developers team up with skilled developers.

Selection

6. Projects with distant developers are more successful.
7. But not if we condition on developer skill.

Results

Measuring skill and quality

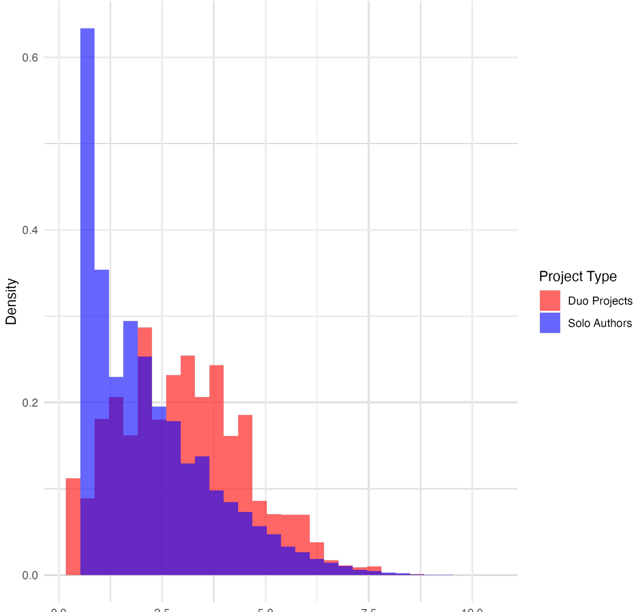
Developer skill

1. Commits in other projects
2. Days worked on other projects
3. Total stars on other projects

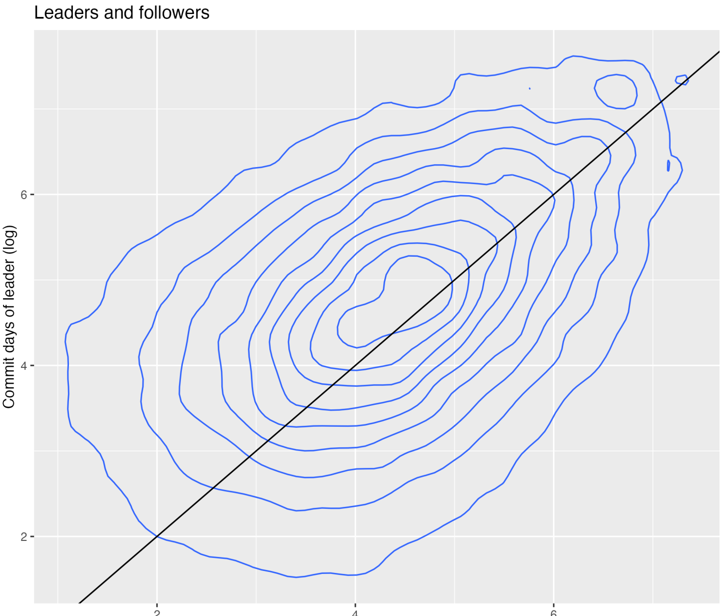
Software quality

1. Number of stars
2. Number of downstream libraries

Two-person projects have better developers



Leaders are better than followers



Collaboration in space

Gravity model of collaboration

Developer i and j collaborate with probability

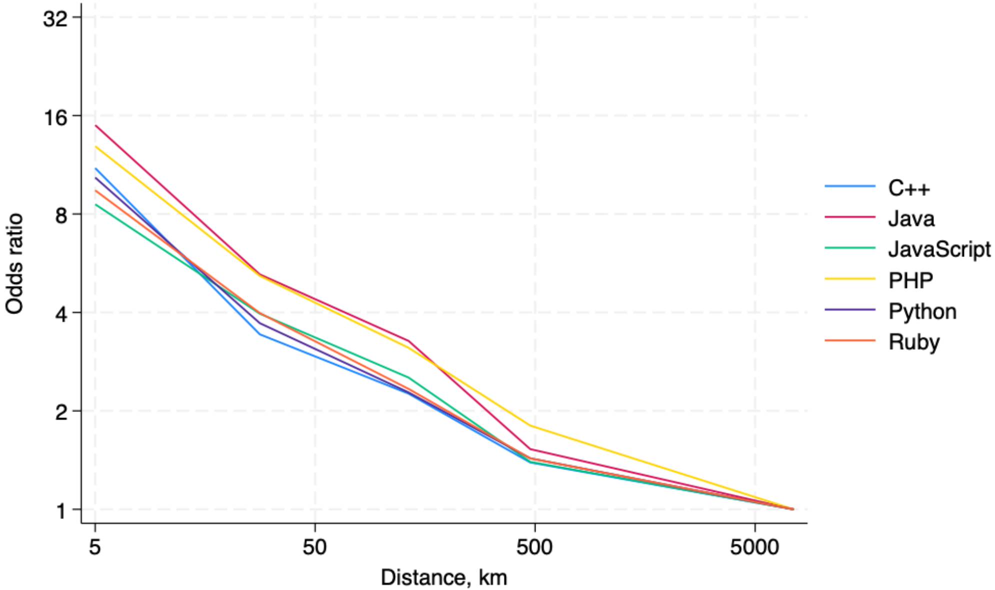
$$\Pr(\text{Collaboration}_{ij}) = \exp(\alpha_i + \beta_j - \gamma \times \text{distance}_{ij})$$

Aggregate across city pairs d and o:

$$E(N_{do,\text{collab}}) = N_o \times N_d \times \exp(\tilde{\alpha}_d + \tilde{\beta}_o - \gamma \times \text{distance}_{do})$$

Estimate this with Poisson maximum likelihood.

Gravity of team formation across languages



Frictions reduce work but increase quality

Dependent Variables:	n_commits	n_days	n_stars	n_downstream
Model:	(1)	(2)	(3)	(4)
Variables				
Average distance (km, log)	0.0399***	0.0217***	0.2252***	0.3326***
	(0.0042)	(0.0011)	(0.0113)	(0.0522)
No. cities (log)	-0.1557***	-0.1528***	0.3228***	1.830***
	(0.0303)	(0.0072)	(0.0805)	(0.3505)
No. countries (log)	-0.2569***	-0.2199***	0.4845***	0.9514***
	(0.0233)	(0.0047)	(0.0473)	(0.2028)
No. developers (log)	1.235***	1.100***	0.5690***	-1.506***
	(0.0295)	(0.0067)	(0.0843)	(0.3246)
Fit statistics				
Observations	267,086	267,086	267,086	267,086
Pseudo R^2	0.24991	0.40519	0.42086	0.23597

Conclusion

Conclusion

1. Tractable modle of global team formation and collaboration.
2. Team formation in OSS is highly localized.
3. Spatial diversity is associated with higher quality of work.

Get in touch

@GaborBekes, @JulianHinz, @korenmiklos, @lohmann_econ