

# The geography of production and sourcing in the weightless economy: Evidence from open-source software

---

Gábor Békés<sup>1</sup>   Julian Hinz<sup>2</sup>   Miklós Koren<sup>3</sup>   Aaron Lohmann<sup>4</sup>

December 12, 2023

<sup>1</sup>Central European University, KRTK, CEPR

<sup>2</sup>Bielefeld University, Kiel Institute for the World Economy

<sup>3</sup>Central European University, KRTK, CEPR, Cesifo

<sup>4</sup>Bielefeld University, Kiel Institute for the World Economy

## **Big Picture:**

- How dispersed developers create great products.

## **Big Picture:**

- How dispersed developers create great products.

## **Focus of This Paper:**

- How and where good Open Source Software (OSS) is produced.
- OSS no fixed costs, and no need for face-to-face interaction - pure online.
- Geography may not significantly impact OSS development.

## **Big Picture:**

- How dispersed developers create great products.

## **Focus of This Paper:**

- How and where good Open Source Software (OSS) is produced.
- OSS no fixed costs, and no need for face-to-face interaction - pure online.
- Geography may not significantly impact OSS development.

## **Data:**

- Writing code together – Collaboration (Github)
- Using other people's code – imported dependencies (Dependencies.io).

# Open Source Software

---

# Open Source Software (OSS) is everywhere

Open Source Software (OSS) has a vast landscape, GitHub hosts over 330 million repositories.

OSS plays an important roles in

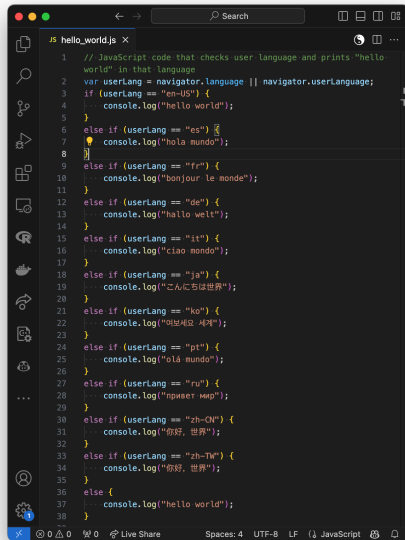
- Websites (JavaScript)
- Operating systems (Linux, Android)
- Data (R Tidyverse, Python Pandas, Julia)
- Machine Learning and AI (PyTorch, LLaMA)

OSS mostly free, but present in fee-based platforms

- Overleaf

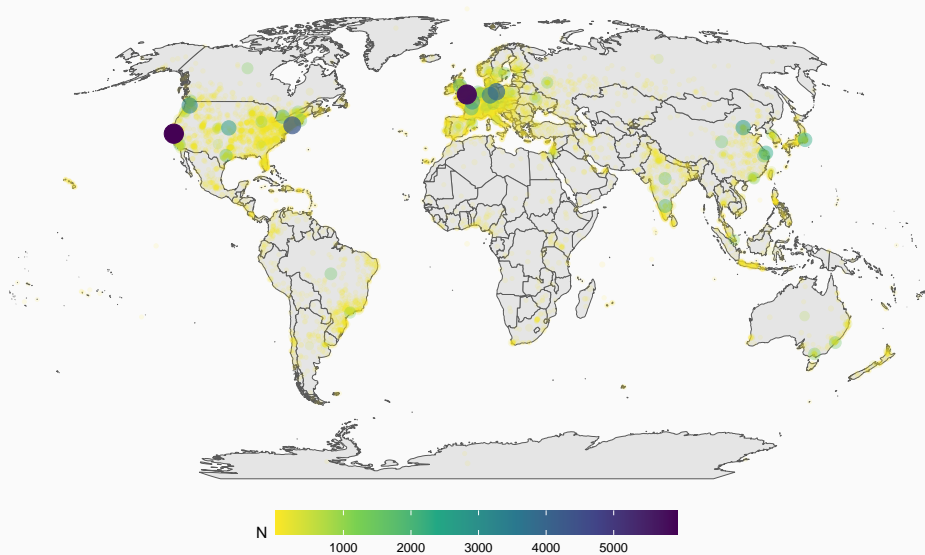
# Focus on JavaScript

- JavaScript is one of the biggest programming languages
- used in web development and app development
- NPM is a package manager
- organizes packages and provides access



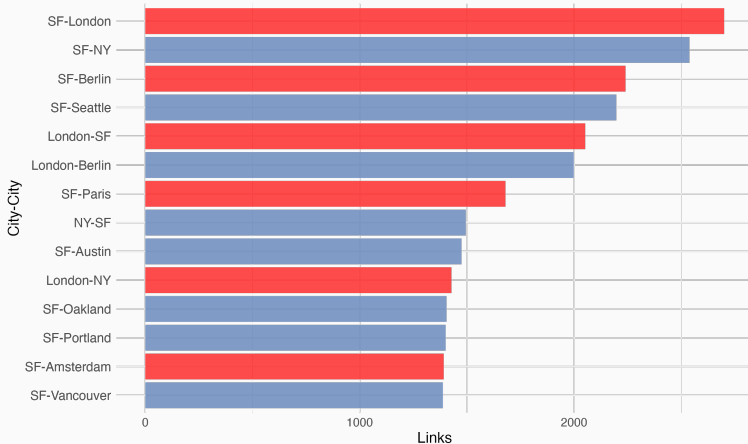
```
1 // JavaScript code that checks user language and prints "hello
  world" in that language
2 var userLang = navigator.language || navigator.userLanguage;
3 if (userLang == "en-US") {
4     console.log("hello world");
5 }
6 else if (userLang == "es") {
7     console.log("hola mundo");
8 }
9 else if (userLang == "fr") {
10    console.log("bonjour le monde");
11 }
12 else if (userLang == "de") {
13    console.log("hallo welt");
14 }
15 else if (userLang == "it") {
16    console.log("ciao mondo");
17 }
18 else if (userLang == "ja") {
19    console.log("こんにちは世界");
20 }
21 else if (userLang == "ko") {
22    console.log("안녕하세요 세계");
23 }
24 else if (userLang == "pt") {
25    console.log("olá mundo");
26 }
27 else if (userLang == "ru") {
28    console.log("привет мир");
29 }
30 else if (userLang == "zh-CN") {
31    console.log("你好，世界");
32 }
33 else if (userLang == "zh-Tw") {
34    console.log("你好，世界");
35 }
36 else {
37    console.log("hello world");
38 }
```

## Global industry: Number of JavaScript developer per city





## Top city pairs per number of developers



Most frequent city-pairs for repos developed from 2 cities

# Search, Agglomeration and Selection

- Costs of setting up a partnership and maintaining it
- Search costs of inputs (code chunks)
  - Written together – finding a collaborator
  - Using already published code – finding a package
- Is successful code written within a city or over many locations?
- Agglomeration vs "selection to export"
  - Agglomeration – F2F meeting key, depends on "distance"
  - Selection – developers of best projects are able to cover this cost

## Preview of Findings

- Distance matters a lot for collaboration:
- Distance matters a little for importing repos
- Code written in more cities more likely to be popular
  - Selection strong – great products made spatially dispersed
- Being in organization reduce spatial friction

- **Geographical Distance / Network formation / Agglomeration:** Chaney (2014) Bernard et al. (2019) Davis and Dingel (2019) Bailey et al. (2021)
- **Gravity: Digital:** Blum and Goldfarb (2006) Anderson et al. (2018)
- **Frictions in services:** Stein and Daude (2007) Bahar (2020)
- **Patents and science:** Bircan et al. (2021), Head et al. (2019), Jaffe et al. (1993), Singh (2008)
- **OSS:** Lerner and Tirole (2002) , Laurentsyeve (2019) Wachs et al. (2022) Fackler et al. (2023)

# Open source software vs patents and academia

- R&D and patenting
  - Need machines, secrecy, often top-down
  - Distance matters in collaboration
  - More cited patents – geographically focused authors
- Science (math, academic papers)
  - Similar, but often longer projects, not open, F2F important to think and discuss
  - Distance matters in collaboration
  - Major role of top Universities / Centers

## **Data, methods and results**

---

- **Git**: Distributed version control system for software projects
- **GitHub**: A platform to collaboratively work on software projects
- **Dependency**: An imported package that provides a functionality
- **Package**: A unit of software, provision of a (bundle of) functionality
- **Repository**: A storage for a package (what we observe)
- **Commit**: The smallest unit of contribution

Collaboration — Working on the same code with others

- GHTorrent: Tracks metadata on GitHub usage
- Commits, locations and user organisations
- Row: One commit from a developer to a repository
- Focus on links: binary if a developer committed at all to a repository

Dependencies — Sourcing of intermediate inputs

- Libraries.io: Tracks data on single software repositories
- dependency linkages
- Row: An imported dependency (package) to repo 1 from repo 2
- Can be mapped to repositories on GitHub



## Scope of data

- Data coverage: 2013 – 2019
- We know location as city for developers
- Contributions by 217K developers,
- 300K repos
- 17% of repos have multiple developers (ie have collaboration)
- 70K organizations, with 120K developers

- formally a (GitHub) permission system
- Collection of users
- Collection of projects
- We observe public memberships
  - Underestimate their presence.

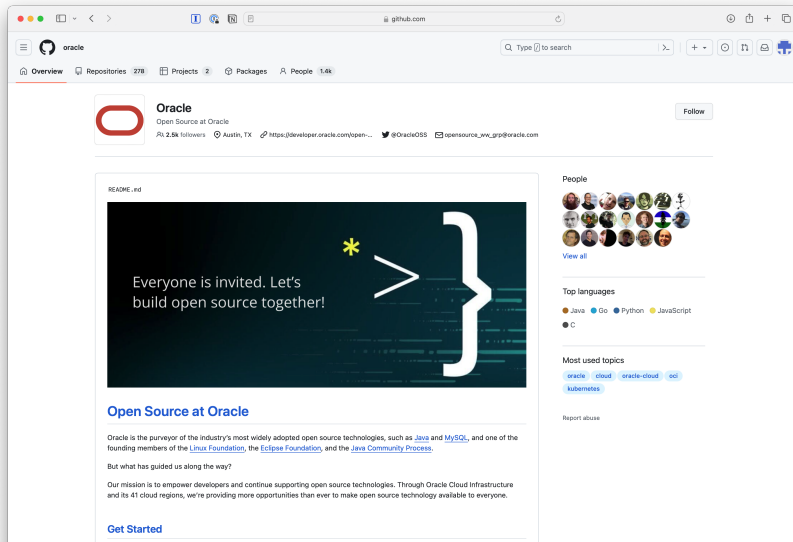
# Organizations — Example 1: Tidier for Julia

- No formal org
- share info, practice
- 3 members

The screenshot shows the GitHub organization page for TidierOrg. At the top, the navigation bar includes the organization name, a search bar, and links to Overview, Repositories (9), Projects, Packages, and People (3). The main header features the Tidier logo and the tagline "Making Julia a Tidier Place for R Users". Below this, the README for the organization is displayed, starting with "Welcome to Tidier". The text describes the organization's goal of creating tools for Julia users familiar with the tidyverse ecosystem, highlighting the flagship package `Tidier.jl` as a 100% Julia implementation of the R tidyverse meta-package. A section for `Tidier.jl` further explains it as a 100% Julia implementation that re-exports several other packages. On the right sidebar, there are sections for "People" (showing 3 members), "Top languages" (showing Julia), and a "Report abuse" link. The bottom of the image shows a Windows taskbar with a temperature of 14°C and various application icons.

# Organizations — Example 2: Oracle

- Company
- share info, practice, showcase
- 1.4k members



# Raw data to regressions

- Collaboration – link developers who contribute to the same repo.
- Dependencies – link developers from one package using another
- One observation is one link
- Aggregated at city (city pair) level

**MORE:** [▶ Aggregation](#)

$$\Pr(Y_{od}|x_o, x_d, d_{od}) \approx \text{Poisson}[N_o \times N_d \times \exp(\beta_1 x_i + \beta_2 x_j + \beta_3 d_{ij})]$$

- Outcome: Number of links between cities  $o, d$
- $d_{ij}$  Distance measured as a set of indicators / log-linear
- Origin and destination city FE
- $N_o \times N_d$  -Exposure: Number of developers in city  $o \times d$

**MORE:** [▶ From logit to Poisson](#)

# Modelling search and maintenance costs

- Personal communication – distance in terms of travel
  - Same city – e.g. universities, office parks
  - Agglomeration (1-50km) – regional events
  - Regional (50-200km) – national conferences
  - Short trip (200-700km) – big conferences
  - Beyond 700km (*as base*) – global events
- Travel difficulty
  - Crossing borders
  - Visa requirement
  - Timezone difference
- Homophily/Communication costs — shared language

## Results: Distance key in contributions, bit for dependencies

- Exclude links within organizations.
- Exclude small cities: 3.4m city pairs
- Origin city FE
- Dest. city FE.

Dependent Variables: Model:	contr._value (1)	dep._value (2)	contr._value (3)	dep._value (4)
dist_cat = Samecity(0-1)	0.7440*** (0.0765)	0.0556*** (0.0106)		
dist_cat = Agglo(1-50)	0.5610*** (0.0793)	0.0592*** (0.0115)		
dist_cat = Region(50-200)	0.2400*** (0.0303)	0.0270*** (0.0094)		
dist_cat = Shorttrip(200-700)	0.0628*** (0.0107)	0.0067** (0.0032)		
cities_in_same_country	0.1635*** (0.0134)	0.0145** (0.0071)		
common_language	0.0815*** (0.0140)	0.0244*** (0.0060)		
visafree_travel	0.1348*** (0.0327)	0.0325** (0.0148)		
tz_gap_nothigh	0.0319*** (0.0088)	0.0102*** (0.0033)		
ln_dist			-0.0953*** (0.0075)	-0.0109*** (0.0020)
Pseudo R <sup>2</sup>	0.84942	0.98867	0.84883	0.98865

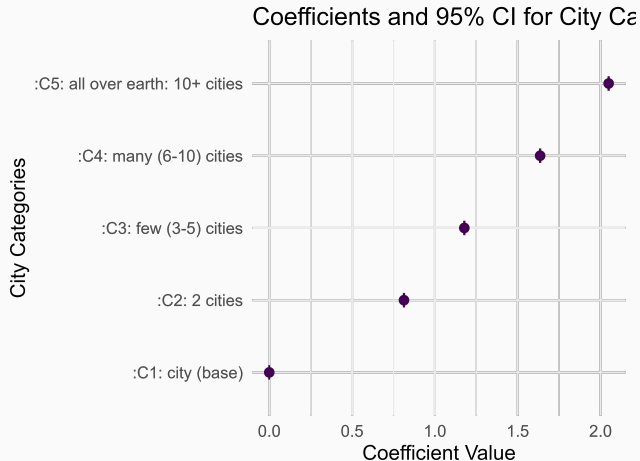


## Success (popularity) and spatial dispersion

$$\Pr(Y_i|.) \approx \text{Poisson}[\exp(\beta_1 \ln\_devs_i + \beta_2 \ln\_country / dev_i + \beta_3 \ln\_city / country_i)]$$

- Outcome: Number of repos importing this repo  $i$
- $\ln\_devs_i$  number of developers (log)
- $\ln\_country / dev_i$  number of countries / developers (log)
- $\ln\_city / country_i$  number of cities / countries (log)

## Results 2: Popularity of repos and Spatial dispersion



- **Popularity:** how many times a repo was used as dependency for other repos
- **Spatial dispersion:** Number of cities (in categories) developers are located
- More widely adopted repos will be produced by developers located in more cities

## Results 2: More popular dependency - higher spatial dispersion

- Only repos used as dependency at least once
- Exclude top few packages
- Robust to size (commits) and N of developers, N dependencies as confounders.
- Col3: only NPM

Dependent Variables: Model:	quality2_count_dependents_all (1)	quality2_count_dependents_all (2)	quality1_count_dependents_npm (3)
<i>Variables</i>			
Constant	8.499*** (0.0002)	8.748*** (0.0002)	3.176*** (0.0017)
ln_count_cities	0.6906*** (0.0001)		
ln_count_developers		0.7332*** (0.0001)	1.722*** (0.0011)
ln_cities_per_country		1.326*** (0.0004)	1.790*** (0.0038)
ln_countries_per_developer		1.664*** (0.0003)	2.526*** (0.0033)
Observations	15,812	15,812	15,812
Pseudo R <sup>2</sup>	0.03537	0.05010	0.19981

Signif. Codes: \*\*\*: 0.01, \*\*: 0.05, \*: 0.1

## Results 3: Organizations are super important

- Confounder: Developers within organization collaborate and import dependencies *much much* more
- Heterogeneity: for collaboration it reduces distance elasticity by 1/3
- For repos in organization, dispersion across countries is 50% less relevant, cities 20% less relevant

**MORE:** [► Role of organizations](#)

## Results 3: Organizations are super important

- Exclude links within organizations.
- Exclude small cities: 3.4m city pairs
- Origin city FE
- Dest. city FE.

Dependent Variables: Model:	contr_value (1)	dep_value (2)	contr_value (3)	dep_value (4)
dist_cat = Samecity(0-1)	0.5475 *** (0.0530)	0.0591 *** (0.0118)		
dist_cat = Agglo(1-50)	0.4096 *** (0.0680)	0.0120 (0.0176)		
dist_cat = Region(50-200)	0.2311 *** (0.0363)	0.0180 ** (0.0091)		
dist_cat = Shorttrip(200-700)	0.0535 *** (0.0117)	0.0055 * (0.0032)		
cities_in_same_country	0.1780 *** (0.0157)	0.0184 ** (0.0074)		
common_language	0.0970 *** (0.0150)	0.0310 *** (0.0074)		
visafree_travel	0.1717 *** (0.0521)	0.0449 * (0.0246)		
tz_gap_nothigh	0.0286 *** (0.0102)	0.0099 ** (0.0039)		
<b>In_dist</b>			<b>-0.0869 ***</b> (0.0073)	<b>-0.0114 ***</b> (0.0028)
<b>Within same organization</b>	<b>5.663 ***</b> (0.0886)	<b>3.374 ***</b> (0.1499)	<b>5.680 ***</b> (0.0888)	<b>3.374 ***</b> (0.1507)
Pseudo R <sup>2</sup>	0.83384	0.98608	0.83312	0.98605

## Results 3: Organizations flatten distance

- Exclude links within organizations.
- Exclude small cities: 3.4m city pairs
- Origin city FE
- Dest. city FE.
- *Language, TZ, visa not shown*

Dependent Variables: Model:	contr_value (1)	dep_value (2)	contr_value (3)	dep_value (4)
dist_cat = Samecity(0-1)	0.6730*** (0.0820)	0.0530*** (0.0100)		
same_org × dist_cat = Samecity(0-1)	-0.4099*** (0.1514)	0.0154 (0.0678)		
...	...	...		
cities_in_same_country	0.1761*** (0.0126)	0.0160** (0.0066)		
same_org × cities_in_same_country	-0.0452 (0.1176)	0.1467 (0.1592)		
...	...	...		
Within same organization	5.650*** (0.1125)	3.282*** (0.1418)	5.437*** (0.1545)	3.425*** (0.3437)
ln_dist			-0.0937*** (0.0079)	-0.0112*** (0.0021)
<b>ln_dist × same org</b>			<b>0.0338**</b> (0.0133)	<b>-0.0075</b> (0.0296)
Pseudo R <sup>2</sup>	0.83402	0.98612	0.83318	0.98605

## Results 4: More popular dependency - higher spatial dispersion

- Only repos used as dependency at least once
- Exclude top few packages

Dependent Variable:	quality2_count_dependents_all	
in_org	Not in org	Within org
Model:	(1)	(2)
<i>Variables</i>		
Constant	8.819*** (0.0002)	8.335*** (0.0004)
ln_count_developers	0.6391*** (0.0002)	1.223*** (0.0004)
ln_countries_per_developer	1.702*** (0.0004)	1.491*** (0.0007)
ln_cities_per_country	1.529*** (0.0004)	0.5648*** (0.0008)
Observations	11,737	4,075
Pseudo R <sup>2</sup>	0.05366	0.05326

Signif. Codes: \*\*\*: 0.01, \*\*: 0.05, \*: 0.1

## Recap of results

- Strong localization of developers. Distance matters to find partners to develop with.
- Only limited role of geography for dependencies.
- Good code created in dispersed cities
- Within organisations distance matters less.
- Very much in progress.
  - Mechanism, role of selection
  - Dynamics
  - Other languages – other types of devs, distribution management



## References

---

**Anderson, James E, Ingo Borchert, Aaditya Mattoo, and Yoto V Yotov**, “Dark costs, missing data: Shedding some light on services trade,” *European Economic Review*, 2018, 105, 193–214.

**Bahar, Dany**, “The hardships of long distance relationships: time zone proximity and the location of MNC’s knowledge-intensive activities,” *Journal of International Economics*, 2020, 125, 103311.

**Bailey, Mike, Abhinav Gupta, Sebastian Hillenbrand, Theresa Kuchler, Robert Richmond, and Johannes Stroebe**, “International Trade and Social Connectedness,” *Journal of International Economics*, Mar 2021, 129, 103418.

- Bernard, Andrew B, Andreas Moxnes, and Yukiko U Saito**, “Production networks, geography, and firm performance,” *Journal of Political Economy*, 2019, 127 (2), 639–688.
- Bircan, Cagatay, Beata Javorcik, and Stefan Pauly**, “Creation and Diffusion of Knowledge in the Multinational Firm,” 2021. Working Paper.
- Blum, Bernardo S and Avi Goldfarb**, “Does the internet defy the law of gravity?,” *Journal of international economics*, 2006, 70 (2), 384–405.
- Chaney, Thomas**, “The network structure of international trade,” *The American Economic Review*, 2014, 104 (11), 3600–3634.
- Davis, Donald R and Jonathan I Dingel**, “A spatial knowledge economy,” *American Economic Review*, 2019, 109 (1), 153–170.
- Fackler, Thomas, Michael Hofmann, and Nadzeya Laurentsyevea**, “Defying Gravity: What Drives Productivity in Remote Teams?,” Technical Report, LMU CRCT Discussuion Paper 427 2023.

- Head, Keith, Yao Amber Li, and Asier Minondo**, “Geography, Ties, and Knowledge Flows: Evidence from Citations in Mathematics,” *The Review of Economics and Statistics*, 10 2019, 101 (4), 713–727.
- Jaffe, Adam B, Manuel Trajtenberg, and Rebecca Henderson**, “Geographic localization of knowledge spillovers as evidenced by patent citations,” *Quarterly journal of Economics*, 1993, 108 (3), 577–598.
- Laurentsyevea, Nadzeya**, “From friends to foes: National identity and collaboration in diverse teams,” Technical Report, CESifo Discussion Paper 2019.
- Lerner, Josh and Jean Tirole**, “Some simple economics of open source,” *The journal of industrial economics*, 2002, 50 (2), 197–234.
- Stein, Ernesto and Christian Daude**, “Longitude matters: Time zones and the location of foreign direct investment,” *Journal of International Economics*, 2007, 71 (1), 96–112.

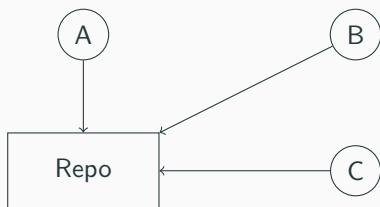
**Wachs, Johannes, Mariusz Nitecki, William Schueller, and Axel Polleres,** “The Geography of Open Source Software: Evidence from GitHub,” *Technological Forecasting and Social Change*, 2022, 176, 121478.

# Aggregation

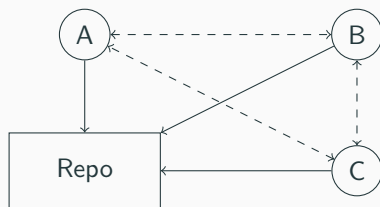
---

- Start with the developer's link to a repository (via commits)
- Directed but (mostly fully) symmetric
- Transform it to developer to developer links
- Aggregate at city level

## Links in the contribution network



**Figure 1:** Developers committing to a repository.



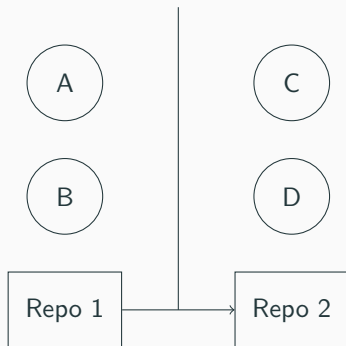
**Figure 2:** Developers committing to a repository including implied contributor to contributor links.

Solid lines are what we **observe**. Dashed lines is what we **infer**.

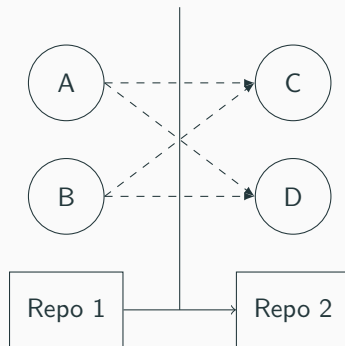
- We observe a repository importing another one as dependency.
- Directed, not symmetric
- Transform it to developer to developer links
  - Use knowledge of producers of the dependency as well
- Aggregate at city level



## Links in the dependency network



**Figure 3:** Dependency of repository 1 on repository 2 with the respective developers.



**Figure 4:** Dependency of repository 1 on repository 2 with the respective developers. Dashed lines indicate implied links between developers.

Again, solid lines are what we **observe**. Dashed lines is what we **infer**.

- Two people with three repository will generate 3 links
- Unweighted by intensity in terms of commits

## Aggregation - Variation at city level

- Key developer observable: city.
- We can aggregate up to city pairs.
- How many collaborations are there between London and San Francisco?
- How many dependencies made in London are imported by packages made in SF?
- Take into account that dependencies may be made in multiple cities. Single import – multiple cities get links

## Example I: City-level aggregation of collaboration

Repository 1 developers:  $A$  in Vienna,  $B$  and  $C$  in Budapest.

- Vienna - Budapest: 1
- Budapest - Vienna: 2
- Budapest - Budapest: 1
- Vienna - Vienna: 0

## Example II: City level aggregation of dependencies

Let repository 2 (  $D$  lives in Kiel) depend on repository 1 ( $A$  live in Vienna and  $B$  and  $C$  in Budapest.).

- Vienna - Budapest and Budapest - Vienna: 0
- Budapest - Kiel: 2
- Vienna - Kiel: 1

Note, that even though links are on level of repository, city links are based on the users.

# Most packages are done in a few, few by many cities

**Table 1:** Cities per repository excluding single-developer repositories

- Most repositories are *located* in 1-3 cities.
- 1.6% done in dozen(s) of cities

Number of cities per repository	Share of Repositories
1	38.31%
2	38.20%
3 - 5	17.98%
6 - 10	3.89%
11-100	1.57%
100+	0.03%

## Estimation metrics

---

Collaboration or dependency link between developer  $i$  and  $j$ ,

$$\Pr(Y_{ij} = 1 | x_i, x_j, d_{ij}) = \Pi(\beta_1 x_i + \beta_2 x_j + \beta_3 d_{ij})$$

with

$$\Pi(z) = e^z / (1 + e^z)$$

the logistic function

**Assumption:** Independence across links, add fixed effects



In practice, distance only varies at the city level. Take origin city  $o$  and destination city  $d$ .

$$Y_{od} := \sum_{i \in o} \sum_{j \in d} Y_{ij}$$

$$\Pr(Y_{od} | x_o, x_d, d_{od}) = \text{Binomial}[N_o \times N_d, \Pi(\beta_1 x_i + \beta_2 x_j + \beta_3 d_{ij})]$$

Here  $N_o \times N_d$  is the total number of *potential* links between cities  $o$  and  $d$ .

When  $\Pi$  is small,

$$\Pr(Y_{od} | x_o, x_d, d_{od}) \approx \text{Poisson}[N_o \times N_d \times \exp(\beta_1 x_i + \beta_2 x_j + \beta_3 d_{ij})]$$

## Aggregate to Poisson (2)

We may also look at a subsample (like users not in the same GitHub organization)

$$Y_{od, \text{not org}} := \sum_{i \in o} \sum_{j \in d, j \notin \text{org}(i)} Y_{ij}$$

This changes the *exposure variable*,

$$\Pr(Y_{od, \text{not org}} | x_o, x_d, d_{od}) \approx \text{Poisson}[N_{od, \text{not org}} \times \exp(\beta_1 x_i + \beta_2 x_j + \beta_3 d_{ij})],$$

with  $N_{od, \text{not org}}$  the number of user pairs in city  $o, d$ , *not sharing* an organization.

Important:  $N_{od, \text{not org}}$  may be zero.

# What is a Poisson regression?

First-order conditions for Maximum Likelihood:

$$\sum_o \sum_d x_i [Y_{od} - N_{od} \exp(\beta_1 x_i + \beta_2 x_j + \beta_3 d_{ij})] = 0$$

$$\sum_o \sum_d x_j [Y_{od} - N_{od} \exp(\beta_1 x_i + \beta_2 x_j + \beta_3 d_{ij})] = 0$$

$$\sum_o \sum_d d_{ij} [Y_{od} - N_{od} \exp(\beta_1 x_i + \beta_2 x_j + \beta_3 d_{ij})] = 0$$

- Level (not log) error terms are orthogonal to RHS variables.
- Exposure variable has fixed exponent of 1 ( $\approx$  weighting).
- Standard errors computed from GMM, not ML. E.g., we allow for two-way city clustering.

# Organizations and causality

---

# Organizations and causality?

- Causal statement is: Organizations reduce spatial frictions ( $\text{org} \rightarrow \text{distance}$ )
- Instead we see that link frequency within organizations tend to be less dependent on distance
- More dispersed developers create organization to co-ordinate ( $\text{distance} \rightarrow \text{org}$ )
- Top developers gather in org and they don't care about distance confounding results.  
( $\text{topdev} \rightarrow \text{org} + \text{topdev} \rightarrow \text{distance}$ )

- Another way to think about organizations is bringing some professionalism and "organization"
- Confirm when we look at top 10% of repos in terms of commits – distance friction also drops