El-Komboz, Lena Abou; Fackler, Thomas A.; Goldbeck, Moritz

Working Paper

Productivity Spillovers among Knowledge Workers in Agglomerations: Evidence from GitHub

CESifo Working Paper, No. 11277

Provided in Cooperation with:
Ifo Institute – Leibniz Institute for Economic Research at the University of Munich

Suggested Citation: El-Komboz, Lena Abou; Fackler, Thomas A.; Goldbeck, Moritz (2024) : Productivity Spillovers among Knowledge Workers in Agglomerations: Evidence from GitHub, CESifo Working Paper, No. 11277, CESifo GmbH, Munich

This Version is available at:
https://hdl.handle.net/10419/305519

CESifo WORKING PAPERS

11277
2024

August 2024

Productivity Spillovers among Knowledge Workers in Agglomerations: Evidence from GitHub

Lena Abou El-Komboz, Thomas A. Fackler, Moritz Goldbeck

CESifo

Impressum:

CESifo Working Papers
ISSN 2364-1428 (electronic version)
Publisher and distributor: Munich Society for the Promotion of Economic Research - CESifo GmbH
The international platform of Ludwigs-Maximilians University's Center for Economic Studies and the ifo Institute
Poschingerstr. 5, 81679 Munich, Germany
Telephone +49 (0)89 2180-2740, Telefax +49 (0)89 2180-17845, email office@cesifo.de
Editor: Clemens Fuest
https://www.cesifo.org/en/wp
An electronic version of the paper may be downloaded
· from the SSRN website: www.SSRN.com
· from the RePEc website: www.RePEc.org
· from the CESifo website: https://www.cesifo.org/en/wp

CESifo *Working Paper No. 11277*

Productivity Spillovers among Knowledge Workers in Agglomerations: Evidence from GitHub

Abstract

Software engineering is prototypical of knowledge work in the digital economy and exhibits strong geographic concentration, with Silicon Valley as the epitome of a tech cluster. We investigate productivity effects of knowledge worker agglomeration. To overcome existing measurement challenges, we track individual contributions in software engineering projects between 2015 and 2021 on GitHub, the by far largest online code repository platform. Our findings demonstrate individual productivity increases by 2.8 percent with a ten percent increase in cluster size, the share of the software engineering community in a technology field located in the same city. Instrumental variable and dynamic estimation results suggest these productivity effects are causal. Productivity gains from cluster size growth are strongest for clusters hosting between 0.67 and 13.5% of a community. We observe a disproportionate activity increase in high-quality, large, and leisure projects and for co-located teams. Overall, software engineers benefit from productivity spillovers due to physical proximity to a large number of peers in their field.

JEL-Codes: D620, J240, O330, O360, R320.

Keywords: high-skilled labor, geography, innovation, peer effects, collaboration.

*Lena Abou El-Komboz**
ifo Institute – Leibniz Institute for Economic Research at the University of Munich / Germany
abou-el-komboz@ifo.de

Thomas A. Fackler
Surrey Business School, University of Surrey / United Kingdom
t.fackler@surrey.ac.uk

Moritz Goldbeck
ifo Institute – Leibniz Institute for Economic Research at the University of Munich / Germany
goldbeck@ifo.de

*corresponding author

August 7, 2024

We thank Florian Englmaier, Oliver Falck, Anna Kerkhof, and Chris Stanton for valuable comments and suggestions. We also thank conference participants at EEA 2021, EARIE 2021, VfS 2021, 16th North American Meeting of the Urban Economics Association; as well as participants at CRC Retreat Schwanenwerder 2021, CESifo/ifo Junior Workshop on Big Data 2021, and seminars at ifo Institute. Support by the Deutsche Forschungsgemeinschaft through CRC TRR 190 (project number 280092119) and by the bidt Think Tank project "Changing workplaces: Patterns and determinants of technology and skill adoption by firms and individuals" is gratefully acknowledged. Thomas Fackler thanks the Laboratory for Innovation Science at Harvard for hospitality while writing parts of this paper.

1 Introduction

Urban density is associated with higher wages and productivity. One of the main reasons for this relationship is improved diffusion of knowledge through physical proximity (Jaffe et al., 1993; Glaeser, 1999; Atkin et al., 2022). Knowledge spillovers among workers occur when individuals benefit from the skills of their local peers and learn from each other, which increases productivity (Lucas, 1988; Cornelissen et al., 2017; De la Roca and Puga, 2017). Knowledge spillovers are especially important in innovative sectors (Audretsch and Feldman, 1996), where collaboration and learning are crucial (Carlino et al., 2007; Jones, 2009; Azoulay et al., 2010; Combes et al., 2010; Andersson et al., 2014; Catalini, 2018). To exploit localized advantages related to collaboration and knowledge exchange, workers and firms tend to locate near each other, especially within a research field or industry (Alcácer and Chung, 2007; Carlino and Kerr, 2015; Moretti, 2021). This leads to geographical agglomeration of tech industries in few cities (Carlino et al., 2012; Atkinson et al., 2019). Surprisingly, software engineering, a key component of almost any high-tech endeavor today (Chattergoon and Kerr, 2022), is characterized by a particularly high spatial concentration of workers in a couple of large clusters (Kerr and Robert-Nicoud, 2020; Forman and Goldfarb, 2022; Wachs et al., 2022), even though it is highly digitized and codified.

In this paper, we investigate agglomeration effects in software engineering. Specifically, we examine the effect on software engineers' productivity of being located in cities with a larger share of other software engineers in their technology field. To this end, we exploit exogenous variation in cluster size resulting from software engineers moving across cities and joining or leaving a specific technology, an approach pioneered by Moretti (2021). This allows us to estimate the impact of changes in technology-specific cluster size on software engineers' productivity in the respective technology. We deploy a model that features a restrictive number of high-dimensional fixed effects to elicit productivity effects, considering both output quantity and quality as well as effect heterogeneity. Still, estimating agglomeration effects on productivity poses further challenges such as simultaneity and correlated unobserved productivity shocks (Combes et al., 2010). To address these challenges, we investigate effect dynamics and employ an instrumental variable approach by predicting variation in local cluster size from changes originating elsewhere. This shift-share approach ensures that the variation in cluster size is independent of technology-specific local productivity shocks, mitigating potential bias in estimates of the elasticity of productivity with respect to cluster size.

Data from *GitHub*, the by far largest online code repository platform, allows us to track software engineers' productivity at unprecedented resolution. Our data has several crucial advantages over patent data, which the existing literature almost exclusively relies upon as a measure of productivity in the knowledge economy (see, e.g., Jaffe et al., 1993; Carlino et al., 2007; Carlino and Kerr, 2015; Guzman and Stern, 2020). While only a small share of knowledge workers files patents and there are large differences across fields and idea types (Cohen and Lemley, 2001; Carlino and Kerr, 2015), coding is a much more widespread activity and part of almost any high-tech project today (Andreessen et al., 2011; Tambe et al., 2020). *GitHub*

data captures even smallest individual contributions to collaborative projects instantaneously with an exact timestamp. In contrast, for inventor teams, it is unclear who contributed what and when. Only one team outcome, the final patent application, is observable with a significant reporting lag. In addition, patents differ widely in market value and often are never used in production (Boldrin and Levine, 2013; Kogan et al., 2017). Code uploaded to *GitHub* is, by definition, more applied and always used in a software product or component. We, therefore, propose code changes by users on *GitHub*, called commits, as a novel measure of knowledge worker productivity and exploit the granularity and richness of the information from public projects in the *GHTorrent* database (Gousios, 2013), such as the integrated social features on the platform, to track the quantity and quality of software engineers' individual output over time.

Our findings indicate that cluster size, the share of other users in a field located in the same city, positively impacts software engineers' productivity. Specifically, a ten percent increase in technology-specific cluster size is associated with a 2.8 percent increase in user output in that technology. Non-parametric estimation shows the elasticity of productivity with respect to cluster size is largest for clusters hosting between 0.67 and 13.5% of a technology-specific community. Agglomeration effects are smaller for clusters with a community share below or above this range, indicating clusters need a critical mass of users to reap significant productivity benefits from agglomeration. An extensive set of fixed effects precludes that the productivity effect is driven by unobserved heterogeneity or trends. Additionally, contemporaneous effects and IV estimation mitigate potential remaining concerns regarding endogeneity due to sorting and simultaneity.

Heterogeneity analyses suggest that the effects are significantly larger for high-quality projects with increased use-value for the community as measured by stars and forks on the platform. Relative to the baseline estimates, activity increases disproportionately with cluster size in longer-running, larger, and co-located projects with more team members, indicating that especially collaborative projects are able to tap productivity spillovers from the wider local community. Additionally, we observe a higher activity increase in leisure projects with a high share of commits out of business hours, which are typically not integrated in a formal structure of an organization. Additional analyses demonstrate robustness of our results with respect to measurement and modeling choices as well as sample construction.

This study contributes to three strands of literature. First, we add to the extensive literature exploring agglomeration effects. There is growing descriptive evidence documenting increasing geographic concentration of innovative activity (Verspagen and Schoenmakers, 2004; Bettencourt et al., 2007; Balland et al., 2020) where collaboration and teamwork are essential (Wuchty et al., 2007; Jones, 2009). Agglomeration is much less pronounced in manufacturing (e.g., Ellison and Glaeser, 1997), and recent evidence by Chattergoon and Kerr (2022) links growing concentration to the rise in software intensity. Rising concentration in knowledge-intensive sectors is remarkable as adoption of information and communication technology is high and tends to reduce geographic frictions (Agrawal and Goldfarb, 2008; Steinwender, 2018; Goldbeck, 2023). Presence of strong localized knowledge spillovers (e.g., Audretsch and Feldman, 1996; Ganguli et al., 2020; Catalini, 2018; Rosenthal and Strange, 2020; Bikard and Marx, 2020) might explain rising geographic

concentration. Notably, Moretti (2021) estimates aggregate effects on inventor productivity of geographic clustering. We are first to focus explicitly on software engineering and demonstrate that individual-level productivity effects of agglomeration in this field are significantly higher.

Second, we advance the measurement of innovative activity by introducing a novel proxy for knowledge worker productivity, the number of single code contributions to software engineering projects. This metric helps us overcome several shortcomings of existing measures based on patent data, which the literature almost exclusively relies upon (Acs et al., 2002; Lerner and Seru, 2022). With the rise of the service economy (Buera and Kaboski, 2012) software becomes ubiquitous in innovation (Andreessen et al., 2011; Chattergoon and Kerr, 2022). At the same time, software and information technology constitute an increasingly important blind spot of patent data (Acikalin et al., 2022; Lin and Rai, 2024). Our measure addresses this gap by proposing a more appropriate and reliable metric for innovative activity in software engineering. Furthermore, our measurement approach is more broad-based, capturing a less exclusive set of individuals compared to inventors, and granular both in terms of time resolution and assessment of individual output.

Third, our paper contributes to the understanding of peer effects. A large literature tries to quantify the extent to which individuals benefit from their peers (Angrist, 2014; Herbst and Mas, 2015; Sacerdote, 2014). With a historically strong focus on learning in educational institutions (Manski, 1993; Sacerdote, 2001; Jackson and Bruegmann, 2009) and science (Azoulay et al., 2010; Waldinger, 2012), this body of research extends to studies of the workplace and professional domain (Moretti, 2004; Mas and Moretti, 2009; Cornelissen et al., 2017). We add to this literature by using plausibly exogenous variation in the density of local peers to study their effect on individual-level productivity on a broad sample of knowledge workers in software engineering. Our technology field-specific definition of relevant communities of peers shows that even within software engineering, a fairly narrow domain according to traditional industry classifications, peer effects are confined to specific sub-fields.

The remainder of this paper is organized as follows. We discuss the setting and data in Section 2. Section 3 introduces our empirical strategy. In Section 4, we report the results and Section 5 concludes with a brief discussion.

2 Background and data

Today, software engineering is a crucial part of almost any scientific and innovative endeavor or high-tech product (Andreessen et al., 2011; Webb et al., 2018; Tambe et al., 2020; Chattergoon and Kerr, 2022; Aum and Shin, 2024), be it in artificial intelligence, engineering, app development, or the bio-pharmaceutical industry. For example, software engineers at the biotech company *Moderna* designed an artificial intelligence that greatly improved the speed of mRNA drug discovery and development, leading to one of the first vaccines against Covid-19 on the market (Bean, 2024). In practice, the vast majority of software engineering projects is hosted on the online code repository platform *GitHub*, which is based on the `git` version control

system. The platform launched in 2008 and since then rapidly evolved as the main online platform for hosting code and collaborative software development (Fackler et al., 2020). A free basic version and its ease of use due to seamless integration into software engineering tech stacks make *GitHub* attractive for over 100 million users (Dohmke, 2023). In addition, the platform exhibits features of a social network in line with its motto "social coding" (Lima et al., 2014).

On the platform, users can create and collaborate in projects (*repositories*) to which code can be *pushed*, i.e., uploaded. The smallest unit of user activity in projects is a *commit*, which captures the sum of code changes a user sends to the project during a session. We introduce commits as a novel measure of software developer productivity. Using commits has several advantages over patent data, the most commonly used measure in the literature. Coding is essential in software development and therefore widespread, in contrast to patenting, which also differs widely across different fields and idea types (Cohen and Lemley, 2001; Carlino and Kerr, 2015). In addition, commits capture even small contributions by each individual with an exact timestamp. In patent data, only one team outcome is observed with a significant reporting lag and neither the nature nor the timing of individual members' contributions are observed. Patents also differ widely in use and value (Boldrin and Levine, 2013; Kogan et al., 2017); commits capture more applied activity and are used in software by definition. The *GitHub* platform contains further information. For example, users may *star* a project so that it is bookmarked for future reference. The number of stars per project measures popularity among other users and is a proxy for project quality (Lima et al., 2014). User profiles allow users to showcase their work and display public projects and activity as well as biographical information such as a name, location and organizational affiliation.

We tap *GHTorrent*, a relational database that mirrors the *GitHub* REST API and creates approximately biannual snapshots of public user profiles and activity on the platform. To obtain time-varying user information, we query ten snapshots dated between September 2015 and March 2021 for profiles of users with location in the US or Canada.[1] For these users, we extract the activity stream with timestamped information on commits and project activity from the latest available snapshot (March 2021). We then combine the activity stream and user profiles into a panel with ten time intervals arising from the snapshot dates.[2] Based on their self-reported location, we assign users to one of the 179 US economic areas defined by the *Bureau of Economic Analysis* or the Canadian equivalent, i.e., one of the 76 economic regions by *Statistics Canada* to city coordinates via exact name matching.[3] Economic areas delineate the "relevant regional markets surrounding metropolitan or micropolitan statistical areas" (Johnson and Kort, 2004). Generally, economic areas are

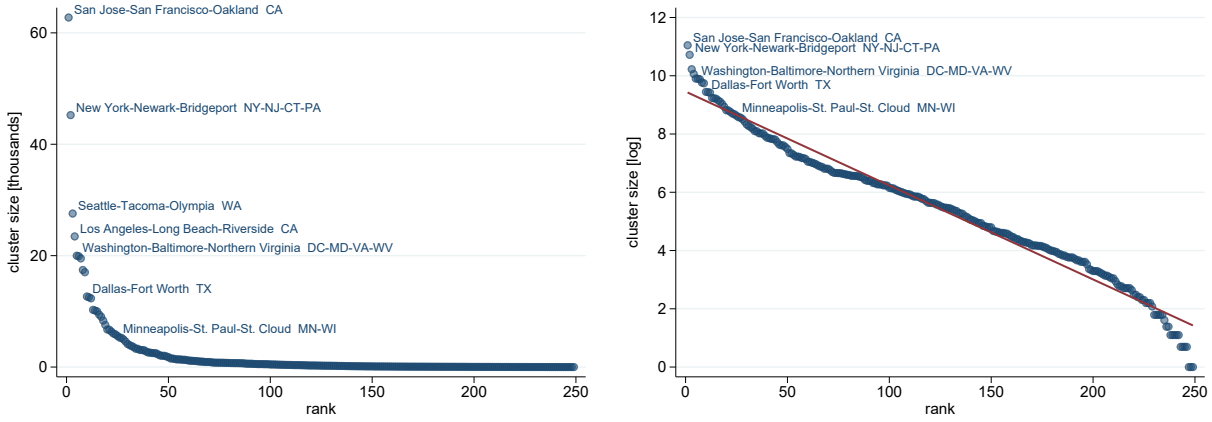
[1] Specifically, snapshots dates in our data are 2015/09/25 (201509), 2016/01/08 (201601), 2016/06/01 (201606), 2017/01/19 (201701), 2017/06/01 (201706), 2018/01/01 (201801), 2018/11/01 (201811), 2019/06/01 (201906), 2020/07/01 (202007) and 2021/03/06 (202103). Goldbeck (2023) validates user locations in *GHTorrent*. For users with a reporting gap in the location information, we impute their location from the previous or next snapshot if possible.

[2] In *GHTorrent*, users are assigned a unique identifier. In principle, commits can be linked to users via `author_id` or `committer_id`. Since users may commit code authored by someone else, we link by `author_id`. This method ensures close connection to individual productivity, but is conservative as many users possess multiple accounts (Casalnuovo et al., 2015).

[3] US and Canadian city coordinates are sourced form `maps` (Becker and Wilks, 2018) and *SimpleMaps* (Simplemaps, 2021).

similar to Metropolitan Statistical Areas (MSAs), but tend to be larger than corresponding MSAs for big cities to capture entire economic regions. Henceforth, we refer to this geographic definition as 'cities'.

Figure 1: Agglomeration in software engineering



(a) rank-size distribution **(b)** logarithmic size

Sources: GHTorrent, own calculations.

Figure 1 displays the strong spatial concentration of software engineers (see, e.g., Kerr and Robert-Nicoud, 2020; Forman and Goldfarb, 2022; Wachs et al., 2022; Goldbeck, 2023) by plotting the number of users in each city as rank-size distribution. Silicon Valley (i.e., the economic area "San Jose–San Francisco–Oakland, CA") clearly stands out as the epitome of a tech cluster with more than 60 thousand users in our data. Cluster size rapidly decays with city rank, with the next largest cities being New York, Seattle, Los Angeles, and Washington, DC. About 50% of users are located in the ten largest cities. In contrast, the vast majority of cities host only few users. The right panel displays the rank distribution using logarithmic city size. Even here, geographic concentration in few large cities is prominently visible as the largest cities lie well above the linear power-law approximation of the distribution.

Technology clusters Since agglomeration benefits from localized knowledge spillovers are concentrated within related fields (see, e.g., Alcácer and Chung, 2007; Bloom et al., 2013; Carlino and Kerr, 2015; Moretti, 2021), we define cluster size on the city $\times$ technology level. For this purpose, we exploit that a programming language is recorded for each project and assign this programming language to every commit in that project.[4] We use the 18 most frequently occurring programming languages that cover about 90% of all commits.[5] Since different programming languages can be closely related, we group programming

[4]Programming languages are broadly defined and include databases and frameworks. Note that a project may contain files in several programming languages. *GHTorrent* assigns the programming language that makes up the largest number of bytes in the project.

[5]Limiting the total number of 404 programming languages to 18 avoids having a large number of cities with only one user in a particular programming language.

languages into five 'technologies' based on being frequently used together according to a developer survey (StackOverflow, 2020).[6] We determine the technology of a user in each time interval via her commit activity. For example, a user who commits to projects in technologies 1 and 3 in the second time interval and lives in Los Angeles is part of the clusters Los Angeles × Technology 1 and Los Angeles × Technology 3 in that time interval. Figure A.2 plots the rank-size distribution by technology, which shows a similar pattern within technologies as for all technologies together (Figure 1). The top ten clusters by technology and their respective user share are listed in Table A.2.

We hypothesize users benefit from being located in a city that hosts a larger share of the community in a specific technology. To robustly compute cluster size, we require a minimum user activity of committing in at least two time intervals.[7] There are 478,957 such users with a location in the US or Canada. Cluster size S for user i in time t in technology f in city c is computed as

$$S_{-ifct} = \frac{\sum_{j \neq i} N_{jfct}}{\sum N_{jft}}, \tag{1}$$

where the summation of users N across all users j in city c in technology f in time t, excluding user i, is divided by the total number of users N in technology f in time t. The accuracy of our measure of cluster size relies on users providing correct location information and maintaining up-to-date profiles. To maximize benefits of the social network functionality and increase visibility for local peers, users generally have an incentive to maintain correct profile information. Reassuringly, we exactly match 98.6% of locations. In addition, Goldbeck (2023) finds no bias in the location information compared to patent data and Abou El-Komboz and Goldbeck (2023) verify the timing of users' location changes on the platform.

Sample For our regression analyses, we select North American users active throughout the observation period, i.e., non-zero commits in all time intervals. This results in a sample of 21,116 users and 2,527,496 user-project-time observations. Summary statistics are reported in Table A.1. The median user makes 56 public code contributions per time interval, i.e., within about six months, and is active in two technologies. Like on most online platforms, activity is heavily right-skewed. Only few projects receive stars and forks. The median city hosts users active in 17 programming languages and all five technologies. Overall, our sample captures a broad base of software engineers with constant activity on the platform that allows us to measure meaningful changes in output.

[6] A visualization of the technology clusters can be found at https://insights.stackoverflow.com/survey/2020#correlated-technologies; last accessed on 03/17/2023. Technology 1 contains JavaScript, CSS, HTML, PHP, C# and TypeScript; Technology 2 Python, Shell, Go, Jupyter Notebook, and R; Technology 3 Ruby; Technology 4 Java, Objective-C, and Swift; and Technology 5 C++, C and Rust.

[7] Note that actual user activity likely is much higher as only public activity is observed. We include users whose account was created in the last time interval and who commit in that time interval.

3 Estimation strategy

We study the effect of cluster size on productivity by estimating the following fixed-effects panel data model via ordinary least squares:

$$\ln(y_{ijflct}) = \alpha + \beta \ln(S_{-ifct}) + d_i + d_j + d_{cf} + d_{cl} + d_{lt} + d_{ct} + \mu_{ijflct}, \tag{2}$$

where y_{ijflct} is the number of commits of user i in time interval t to project j located in city c in the technology f and programming language l and S_{-ifct} is the cluster size in city c of the technology f in time interval t, excluding user i. μ_{ijflct} is an error term. We cluster standard errors at the city $\times$ technology level to account for serial correlation. Importantly, this specification allows us to include a large amount of (high-dimensional) fixed effects d that address many potential concerns regarding identification and ensures that the identifying variation in cluster size originates from users moving between cities and starting or stopping to be active in a technology field.

In particular, user fixed effects d_i capture time-invariant differences in user activity, and project fixed effects d_j account for project-specific activity differences. In addition, we include city $\times$ technology d_{cf} and city $\times$ programming language d_{cl} fixed effects to control for city-specific productivity differences within technologies and programming languages. For example, if programmers in Toronto focused on artificial intelligence within projects, these fixed effects would account for the fact that such a specialization could systematically affect observed activity. Similarly, programming language $\times$ time fixed effects d_{lt} account for programming language-specific time trends and city $\times$ time fixed effects d_{ct} consider changes in average productivity over time for each city as well as changes in city size over time. These fixed effects would capture activity patterns over time, e.g., caused by new cohorts of students learning to program in a language in project-based courses at the start of the academic year.

Our coefficient of interest β captures the relationship between cluster size and user productivity conditional on fixed effects. The identifying variation net of fixed effects comes from users relocating to another city and starting or stopping to commit in a specific technology, similar to Moretti (2021). Thus, this relationship can be causally interpreted if the included fixed effects eliminate endogeneity and the error term μ_{ijflct} is orthogonal to cluster size S_{-ifct}. Productivity spillovers from agglomeration are present if β is greater than zero and absent if β is zero. In particular, a positive β implies a user's productivity in a technology increases with cluster size, i.e., the share of other users in that technology being located in the same city.

An endogeneity concern when estimating agglomeration effects are unobserved determinants in the error term μ_{ijflct} simultaneously affecting productivity and cluster size (Combes and Gobillon, 2015). In particular, potential concerns are sorting and simultaneity. Equation 2 accounts for most forms of sorting into cities and technologies, e.g., due to (changes in) local amenities and infrastructure, by ability, or differences in technology-specific productivity differences across cities. Still, reverse causality might arise when users

whose productivity would have increased anywhere sort into larger clusters. Note that sorting into large clusters on ability is not a concern, nor is sorting to the extent that it leads to an increase in cluster size affecting productivity. Only when users with expected future productivity increases select into growing clusters. A more salient potential concern is simultaneity due to unobserved time-varying productivity shocks that are technology-specific, such as policies at the city level that target a specific technology coinciding with cluster size growth.

We address potential bias due to unobserved time-varying productivity shocks at the city $\times$ technology level using an instrumental variable (IV) approach similar to Autor et al. (2013). The idea is to use only the part of variation in local cluster size that is arguably exogenous because it originates elsewhere. By that, unobserved local productivity shocks at the city $\times$ technology level that affect both productivity and cluster size simultaneously do not affect our estimate. To construct a valid instrument, we leverage a key feature of online code platforms, namely the possibility to commit to projects from anywhere. We instrument local cluster size by commits to local projects that originate elsewhere. Users on *GitHub* frequently contribute to non-local projects, which provides sufficient variation in the number of committers from different cities. At the same time, increases in activity originating elsewhere are unlikely to be an outcome of local productivity gains and are arguably unrelated to unobserved local productivity shocks at the city $\times$ technology level.

In particular, we predict cluster size by changes in the number of non-local users in all projects of a particular technology to which other local users commit, excluding the focal user's projects[8], relative to the change in the overall number of users in that technology. We denote the sum of users committing to project j in time interval t and technology f, excluding city c, as $N_{jf(-c)t}$ and its change between $t-1$ and t as $\Delta N_{jf(-c)t} = N_{jf(-c)t} - N_{jf(-c)(t-1)}$. We compute our instrument as

$$IV_{ifct} = \sum_{s \neq j_i} D_{sfc(t-1)} \frac{\Delta N_{sf(-c)t}}{\Delta N_{ft}}, \tag{3}$$

where $D_{sfc(t-1)}$ indicates if project s in technology f was present in city c at time $t-1$. $N_{sf(-c)t}$ is the logarithm of the sum of users committing to project s in technology f at time t in all cities but city c, and to which user i does not commit. Consequently, $\Delta N_{sf(-c)t}$ is the change in the logarithm of the number of users committing to project s in technology f at time t for all cities but city c and ΔN_{ft} is the change in the logarithms of the total number of users in technology f between time $t-1$ and t.

[8] We consider a user to be connected to a project if she ever committed to that project, not only in the current time interval.

4 Results

4.1 Main results

Table 1 reports the results from our baseline model in Equation 2. The first column conditions on user, project, programming language, technology, city, and time fixed effects. The estimated elasticity of user productivity with respect to cluster size in this specification is 0.1144, suggesting a positive relationship of productivity and cluster size. Adding programming language $\times$ time fixed effects in the second column accounts for trends in programming languages and technologies as well as language-specific productivity shocks common to all users. The decrease in effect size hints that larger clusters experience higher productivity gains from increased popularity of programming languages most frequently used there. After including city $\times$ technology fixed effects in the third column, the elasticity of cluster size increases to 0.1966 and becomes statistically significant at the five percent level. This specification takes into account time-invariant technology-specific factors at the city level that affect user productivity. Higher task complexity in large clusters (Balland et al., 2020) causing users to take longer for each commit compared to equally productive workers elsewhere is a possible explanation for this increase in effect size. Accounting for city $\times$ language fixed effects in column four leaves estimates virtually unchanged, suggesting that our definition of technologies and clusters appropriately captures relevant software engineering communities.

Our preferred specification in column five adds city $\times$ time fixed effects to account for unobserved productivity shocks at the city level common to all technologies like policies improving local digital infrastructure or the establishment of a presence by a large tech firm. This results in an estimated elasticity of productivity with respect to cluster size of 0.2777, which is statistically significant at the five percent level. The increase compared to column four suggests that city-specific productivity shocks or sorting on local amenities are especially pronounced in smaller clusters. Overall, these results consistently point to significant agglomeration effects in software development. Users are more productive when located in a city with a higher share of other users in their technology. Our preferred estimate implies users on average make 2.8% more commits in a given technology when the share of other users in that technology is ten percent higher. This finding suggests that, for example, a user's number of commits in Technology 1 is expected to increase by 19% if she moves from Chicago to Seattle due to the larger community of users in Technology 1 there.

Compared to the agglomeration effect for top inventors estimated by Moretti (2021), we thus find a four times larger elasticity for software engineers. Several factors might explain these stronger agglomeration effects in software engineering. First, software engineers tend to be younger than patenting inventors and, therefore, learning skills is more important to them in the human capital accumulation phase of their life cycle (Ben-Porath, 1967).[9] Second, the high degree of specialization in software development implies a higher probability that the activity of local peers is relevant to the focal user, leading to a larger potential

[9]Survey results suggest that most software engineers in the US are aged 25-35 years (Patel, 2024; Stackoverflow, 2024), whereas inventors are significantly older (Jones, 2010) with an average age of 45 years (Kaltenberg et al., 2023).

for knowledge spillovers. Third, software is a particularly fast-moving field with a high rate of skill obsolescence (Deming and Noray, 2020) even within STEM fields, requiring continuous learning to maintain and possibly increase productivity. Larger knowledge spillovers in software engineering compared to other fields are a strong incentive for agglomeration, which might, at least partly, explain the particularly high geographic clustering of programmers.

Table 1: Productivity and cluster size

Dep. var.: Commits [log]	(1)	(2)	(3)	(4)	(5)	(6)
Cluster size [log]	0.1144	0.1070	0.0929	0.1966**	0.1935**	0.2777**
	(0.1099)	(0.0785)	(0.0744)	(0.0949)	(0.0962)	(0.1253)
Fixed effects						
User	Yes	Yes	Yes	Yes	Yes	Yes
Project	Yes	Yes	Yes	Yes	Yes	Yes
Technology	Yes	Yes	Yes	Yes	Yes	Yes
Language	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Time	Yes	Yes	Yes	Yes	Yes	Yes
Technology × time		Yes	Yes	Yes	Yes	Yes
Language × time			Yes	Yes	Yes	Yes
City × technology				Yes	Yes	Yes
City × language					Yes	Yes
City × time						Yes
Users	21,116	21,116	21,116	21,116	21,116	21,116
Observations	2,527,496	2,527,496	2,527,496	2,527,496	2,527,496	2,527,496
Adjusted R^2	0.287	0.289	0.290	0.291	0.291	0.292

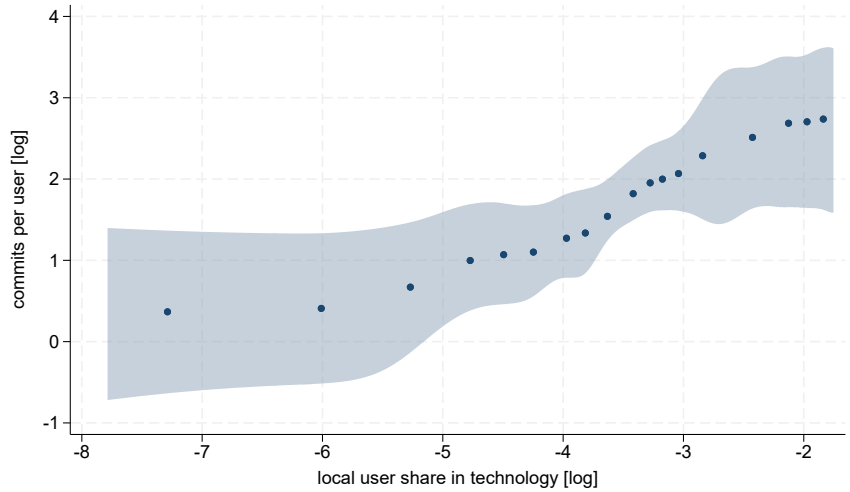
Notes: Language refers to programming language. Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

The elasticity of productivity with respect to cluster size might change depending on the position of cities in the size distribution. For example, productivity spillovers potentially require a certain minimum cluster size to occur as the benefits to individual productivity of only few other co-located users might be smaller. In contrast, similar increases in cluster size might result in smaller relative productivity gains in the largest clusters where already many users are co-located. The presence of both channels could give rise to an S-shaped relationship of the elasticity with respect to cluster size. Au and Henderson (2006), for example, estimate a bell-shaped relation between productivity and city size for Chinese cities and Cattaneo et al. (2023) demonstrate an S-shape pattern for the elasticity of US inventors in Moretti (2021).

Figure 2 depicts a binscatter plot to investigate monotonicity and potential non-linearity in the effect. Following the principled approach of Cattaneo et al. (2024), we obtain a suitable data-driven visualization of the conditional mean function. The relationship between productivity and cluster size is positively monotonous and follows a slight S-shape. The function increases only slightly for very small cluster sizes, while productivity increases are larger for cluster sizes between approximately 0.67 and 13.5 percent. Above this range, the increase is, again, less pronounced for the largest clusters. This suggests that significant agglomeration

effects require a minimum cluster size of around 0.67 percent of the community in a technology being located in the same city. At the same time, when cluster size reaches a level of approximately 13.5 percent, there are little additional productivity gains from further growth in cluster size. This also suggests that our effect is not driven by few large clusters such as the Bay Area. Rather, the effect is present across the entire size distribution and features a slight S-shape with especially medium-sized clusters profiting from increases in cluster size.

Figure 2: Non-parametric estimation



Notes: Graph plots a binscatter representation of the relationship between software engineer productivity and cluster size using `binsreg` (Cattaneo et al., 2023). Specification includes fixed effects for time, technology, language, project, city, and user as well as for time × city, time × technology, and city × technology. *Sources:* GHTorrent, own calculations.

4.2 Heterogeneity

We explore potential heterogeneity of the effect with respect to user and project characteristics. To explore the relation between cluster size and quality of users' activity, we focus on commits to the top ten projects measured by the number of stars received from the community. Table 2 reports the main results for this subsample. Generally, the point estimates are significantly larger across all specifications compared to our baseline estimates. Effects are more precisely estimated, as well, even though the sample size is much smaller, pointing to a tighter relationship between cluster size and productivity in high-quality projects. For the preferred specification with the full set of fixed effects, the elasticity between cluster size and productivity of 0.3239 implies that a user commits about 3.2 percent more in a technology to projects with at least five stars with a ten percent increase in cluster size. This result indicates that the effect on high-quality activity is about 4.6 percentage points (or 16.6%) higher relative to the full sample in Table 1. Note that compared to specifications without city × time fixed effects, this difference is significantly smaller. This

stresses accounting for time-varying unobservables at the city level like the opening of new large tech firm establishments is especially important for high-quality activity.

Table 2: Quality

Dep. var.: Commits [log]	(1)	(2)	(3)	(4)	(5)	(6)
Cluster size [log]	0.1451	0.1359	0.1229	0.2649***	0.2637***	0.3239**
	(0.1043)	(0.0866)	(0.0828)	(0.0860)	(0.0867)	(0.1462)
Fixed effects						
User	Yes	Yes	Yes	Yes	Yes	Yes
Project	Yes	Yes	Yes	Yes	Yes	Yes
Technology	Yes	Yes	Yes	Yes	Yes	Yes
Language	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Time	Yes	Yes	Yes	Yes	Yes	Yes
Technology × time		Yes	Yes	Yes	Yes	Yes
Language × time			Yes	Yes	Yes	Yes
City × technology				Yes	Yes	Yes
City × language					Yes	Yes
City × time						Yes
Users	6,711	6,711	6,711	6,711	6,711	6,711
Observations	392,984	392,984	392,984	392,984	392,984	392,984
Adjusted R^2	0.407	0.408	0.409	0.410	0.412	0.413
$\Delta(\beta_{top10} - \beta_{all})$	0.0307	0.0289	0.0300	0.0683	0.0702	0.0462
$\Delta(\beta_{top10} - \beta_{all})/\beta_{all}$	0.2684	0.2701	0.3229	0.3474	0.3628	0.1664

Notes: Regressions based on the top decile of projects by stars. β_{top10} denotes the estimated coefficient on cluster size. β_{all} refers to the estimated coefficient of cluster size from the corresponding specification in Table 1. Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

Table 3 explores heterogeneity with respect to further characteristics by estimating the relation of cluster size and productivity by quartiles. The first specification reports the effects for each cluster size quartile. Similar to Figure 2, the results point to a slight S-shape of the elasticity of productivity with respect to cluster size. The differences are not pronounced as indicated by the Wald test, which yields a *p*-value of 0.170. The second specification investigates differences with respect to project age, measured in months since project creation. Theoretically, especially established projects might profit from cluster size as the initial set-up is typically trivial while in later phases external impulses are more beneficial to further improve the project (e.g., Ayoubi et al., 2017). Indeed, the elasticity increases with project age from 0.2639 (youngest quartile) to 0.2899 (oldest quartile). This variation is confirmed significant as a Wald test is rejected with a *p*-value of 0.046, suggesting that knowledge spillovers are larger for older projects. Next, we study differences in the elasticity between business and leisure projects, which we elicit by the share of commits made during business hours. We find a significant variation in the elasticity (Wald test *p*-value of 0.006), with leisure projects benefiting more from increases in cluster size. Leisure projects typically exhibit less structure and are not embedded in a professional environment with a higher degree of knowledge organization and thus can

profit more from spillovers from the wider local community. The fourth specification tests for differences in the elasticity with respect to user activity. Active users are often integrated more in local communities and therefore might experience larger productivity gains. We find sizable differences in point estimates, but the variation in the elasticity is not statistically significant (Wald test p-value 0.213). Thus, agglomeration effects are not significantly lower for less active users.

Table 3: Heterogeneity (by quartiles)

Dep. var.: Commits [log]	(1) cluster size	(2) project age	(3) business	(4) activity
1st Quartile (Smallest/Youngest/Leisure/Low)	0.2748**	0.2639**	0.2801**	0.2700**
	(0.1250)	(0.1228)	(0.1238)	(0.1234)
2nd Quartile	0.2688**	0.2661**	0.2806**	0.2716**
	(0.1258)	(0.1248)	(0.1261)	(0.1244)
3rd Quartile	0.2609**	0.2725**	0.2836**	0.2774**
	(0.1272)	(0.1268	(0.1254)	(0.1273)
4th Quartile (Largest/Oldest/Business/High)	0.2651**	0.2899**	0.2456**	0.3013**
	(0.1268)	(0.1263)	(0.1254)	(0.1287)
Full set of FE	Yes	Yes	Yes	Yes
Users	21,116	21,116	21,116	21,116
Observations	2,527,496	2,527,496	2,527,496	2,527,496
Adjusted R^2	0.292	0.292	0.292	0.292
Wald (joint nullity) [p-value]	0.170	0.046	0.006	0.213

Notes: Robust standard errors clustered at the city $\times$ technology level in parenthesis. *** p < 0.01, ** p < 0.05, and * p < 0.1. *Sources:* GHTorrent, own calculations.

In Table 4, we assess heterogeneity regarding binary characteristics by estimating the elasticity separately for subgroups. Specifications one and two distinguish between small and large teams, where projects with at least 5 team members are considered large. The estimated elasticity is almost twice as high for large team projects, indicating that projects with more team members tend to benefit more from the wider local community. This is in line with evidence suggesting that sourcing knowledge from community networks is facilitated by larger team size (Lima et al., 2014). Specifications three and four confirm this notion by contrasting commits to distributed and fully co-located teams. Results show that the productivity increase within fully co-located teams is significantly larger, also pointing towards knowledge spillovers compounding in local teams. Furthermore, we use information on project ownership to separate full-fledged collaborative coding projects from single-person projects that might not require following guidelines with clear expectations on how a contribution should look like (Elazhary et al., 2019). We do so by separately considering commits to projects where the project owner is a different or the focal user in columns five and six. Results show that agglomeration benefits occur almost exclusively in projects owned by other users. This strongly points towards productivity gains in meaningful coding projects with a certain contribution standard.

Table 4: Heterogeneity (binary)

Dep. var.: Commits [log]	team size		geography		ownership	
	(1) small	(2) large	(3) distributed	(4) co-located	(5) others	(6) own
Cluster size [log]	0.1706	0.3243**	0.2736**	0.2932	0.3061**	0.0669
	(0.1217)	(0.1599)	(0.1303)	(0.2504)	(0.1494)	(0.1417)
Full set of FE	Yes	Yes	Yes	Yes	Yes	Yes
Users	21,116	16,061	19,295	21,098	20,644	20,917
Observations	2,118,134	409,362	830,118	168,362	1,423,404	1,104,092
Adjusted R^2	0.317	0.401	0.359	0.299	0.324	0.314

Notes: Robust standard errors clustered at the city × technology level in parenthesis. *** p < 0.01, ** p < 0.05, and * p < 0.1. *Sources:* GHTorrent, own calculations.

4.3 Endogeneity

Although our baseline fixed effects specification already precludes numerous endogeneity concerns, sorting of users with an expected future productivity increase or simultaneous unobserved time-varying productivity shocks on the city × technology level are remaining threats to identification. We address these concerns by estimating the instrumental variable model in Equation 3.

The instrumental variable approach in Equation 3 addresses potential simultaneity of cluster size changes and unobserved productivity shocks at the city × technology level. The first-stage results in Table 5 show that cluster size changes elsewhere are a strong instrument for local cluster size changes as indicated by an F-test of 1,480 in our preferred specification. The negative sign indicates cluster size growth outside the local cluster is associated with a decrease in the share of users in that cluster locally. Using only this plausibly exogenous variation in cluster size triggered by changes in cluster size elsewhere, the second-stage results present an estimate of the elasticity of productivity with respect to cluster size that is unaffected by potential technology-specific local simultaneity. Given the sample differences, the preferred specification in column 4 yields a significant and comparable effect size to our baseline results and suggests simultaneity does not drive our results.

We further assess the plausibility of endogeneity arising from sorting on expected future productivity growth by investigating the dynamics of productivity changes. Sorting of users with future productivity growth independent of their location into larger clusters is unlikely to be tightly connected to the exact timing of changes in cluster size due to movers and entry or exit into technologies of local users. In contrast, observing a strong contemporaneous reaction of productivity to cluster size growth would support agglomeration effects as driver of productivity growth. We estimate the contemporaneous effect in a three-period model with a lead, the contemporaneous period, and a lag in Table A.9. Results suggest a contemporaneous effect with a magnitude comparable to our main effect. In the preferred specification, we find a significant contem-

Table 5: 2SLS estimates

Dep. var.: Δln(commit)	(1)	(2)	(3)	(4)
First Stage	-0.00001***	-0.00001***	-0.00001***	-0.00001***
	(0.00000)	(0.00000)	(0.00000)	(0.00000)
Δln(cluster size)	0.20336	0.29913***	0.29436***	0.19829**
	(0.19268)	(0.08786)	(0.08690)	(0.09711)
Fixed effects				
Time	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes
User		Yes	Yes	Yes
Language			Yes	Yes
Language × time				Yes
Users	18,302	18,302	18,302	18,302
Observations	500,665	500,665	500,665	500,665
F-test (1st stage)	466.53	1,317.96	1,336.73	1,479.94

Notes: Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

poraneous productivity increase of 0.2676 and insignificant reaction of productivity with a point estimate close to zero before. This mitigates potential concerns regarding sorting on unobserved future productivity shocks.

4.4 Robustness

We assess the robustness of our results via additional checks. Our main specification uses user density to measure cluster size, i.e., the share of users in a technology located in a given city. We generally prefer user density as it supports the notion of communities clustering geographically and users benefiting from being in such a hub. As some models rely on absolute cluster size, we test for differences in such a specification in Table A.4. Results vary only marginally, with an elasticity of 0.2777 in the specification with the full set of fixed effects. The distribution of cluster sizes using both measures depicted in Figure A.1 is similar, as well.

Our preferred non-parametric estimation for functional form assessment in Figure 2 uses 18 bins to elicit effect non-linearity with respect to cluster size. In Figure A.3, we extend the number of fixed effects used and use a smaller number of bins that are IMSE-optimally selected according to Cattaneo et al. (2023) for better representation of confidence bands. We generally observe a similar pattern across all binscatter representations with a slight S-shape of productivity with respect to cluster size. As the higher number of bins teases out the S-shape more clearly due to more narrowly spaced point estimates, we opt for this representation as our preferred specification.

On the GitHub platform, most registered users are inactive in most snapshots. Thus, we impose an activity requirement on our sample to study our population of interest, i.e., software engineers. In our main model,

we require public activity in each of the ten time intervals in our sample to extract users with meaningful involvement in software engineering on the platform. Note that this improves upon the existing literature that is mainly focused on the top contributors (Vidoni, 2022). Nevertheless, in Table A.3 we relax this activity requirement to demonstrate our results generalize to broader samples. As productivity changes become less tractable for only occasional contributors, the estimated effect gets slightly smaller when reducing the minimum number of time intervals with activity. Still, the effect size is generally similar in magnitude. Note that we capture activity in public projects. As a consequence, overall activity of users is likely higher since it also includes contributions in private repositories. Reassuringly, however, Goldbeck (2023) validates public contributions to be representative of overall regional software developer activity.

A potential concern with respect to our productivity measure would be automatic activity, e.g., by bots. Non-human activity due to bots is typically observed at high-frequency. We, therefore, exclude the top percentile of users by number of commits in Table A.6. Note that this approach risks losing the most active software engineers on the platform and therefore potentially underestimates our effect. Additionally, we estimate our baseline model excluding projects with more than 40 users or 100 commits in Table A.5 in order to ensure our results are not driven by large projects only. Similarly, in Table A.7 we present a specification without the ten largest cities. We find results comparable to our baseline specification across all these specifications, which indicates a broad-based effect that is not driven by automated contributions. An alternative measure for quality on the platform are forks, i.e., copies of repositories into other repositories. Like stars, this indicates use-value and community interest. Table A.8 reports the results, which show an even larger effect than for stars.

5 Conclusion

Software is ubiquitous, and understanding the economics of its production by knowledge workers is crucial. Yet, widely-used patent data has significant blind spots in software innovation that prevent comprehensive studies of this important sector. We introduce a novel measure of individual software engineer productivity based on granular data from the largest online code repository platform to overcome this challenge. We use our measure to show that higher agglomeration effects compared to other industries can explain the strong geographic concentration in the industry despite its high degree of digitization and, therefore, remote-work capability. Specifically, we estimate individual productivity increases by 2.8 percent for a ten percent increase in cluster size.

Our results have important policy and managerial implications. Most importantly, policymakers, firms, and workers should incorporate the significant effects of localized knowledge spillovers in software engineering into their decision making. The sizable heterogeneity in agglomeration effects on knowledge worker productivity has strong implications for regional policy. Results show effects are largest for cities hosting above 0.67% but below 13.5% of a technology-specific community. Subsidizing new establishments such

as Amazon's HQ2 could thus be a more beneficial strategy for regions within that range. For smaller cities, specialization in niche sub-fields where it is easier to attract a critical share of the community could be a more viable path. On the other end of the spectrum, the largest cities with cluster sizes above 13.5% reap smaller benefits from further community growth and might be better off with regional policies that deepen knowledge exchange between existing knowledge workers.

Firms that are too small to significantly affect cluster size themselves can benefit from knowledge spillovers from larger local communities, which is relevant for location decisions such as opening new or expanding existing establishments. At the same time, our findings suggest that firms may be able to avoid the very largest tech hubs – along with their high labor and real estate costs – while sacrificing little in terms of knowledge spillovers. Our results imply significant spillovers to individual productivity from the wider community that are higher for open innovation. A limitation of our data is that we do not observe activity in private projects and therefore are unable to assess spillovers within organizations if contributions are not made public. Further, the agglomeration effect is confined to specific sub-fields of software engineering suggesting that defining relevant peer groups is crucial for assessments of potential productivity benefits from agglomeration. For workers, our results highlight the importance of the location decision for individual productivity in this fast-moving field and the benefits arising from the local community.

References

Abou El-Komboz, Lena and Moritz Goldbeck, "Career Concerns As Public Good: The Role of Signaling for Open Source Software Development," *CRC Discussion Paper*, 2023.

Acikalin, Utku U., Tolga Caskurlu, Gerard Hoberg, and Gordon M. Phillips, "Intellectual property protection lost and competition: An examination using machine learning," *NBER Working Paper*, 2022.

Acs, Zoltan J, Luc Anselin, and Attila Varga, "Patents and innovation counts as measures of regional production of new knowledge," *Research policy*, 2002, *31* (7), 1069–1085.

Agrawal, Ajay and Avi Goldfarb, "Restructuring research: Communication costs and the democratization of university innovation," *American Economic Review*, 2008, *98* (4), 1578–1590.

Alcácer, Juan and Wilbur Chung, "Location strategies and knowledge spillovers," *Management science*, 2007, *53* (5), 760–776.

Andersson, Martin, Johan Klaesson, and Johan P Larsson, "The sources of the urban wage premium by worker skills: Spatial sorting or agglomeration economies?," *Papers in Regional Science*, 2014, *93* (4), 727–747.

Andreessen, Marc et al., "Why software is eating the world," *Wall Street Journal*, 2011, *20* (2011), C2.

Angrist, Joshua D, "The perils of peer effects," *Labour Economics*, 2014, *30*, 98–108.

Atkin, David, M Keith Chen, and Anton Popov, "The returns to face-to-face interactions: Knowledge spillovers in Silicon Valley," Technical Report, National Bureau of Economic Research 2022.

Atkinson, Robert, Mark Muro, and Jacob Whiton, "The case for growth centers," *Brookings Institution: Washington, DC, USA*, 2019.

Au, Chun-Chung and J Vernon Henderson, "Are Chinese cities too small?," *The Review of Economic Studies*, 2006, *73* (3), 549–576.

Audretsch, David B. and Maryann P. Feldman, "R&D spillovers and the geography of innovation and production," *American Economic Review*, 1996, *86* (3), 630–640.

Aum, Sangmin and Yongseok Shin, "Is Software Eating the World?," *NBER Working Paper*, 2024.

Autor, David H, David Dorn, and Gordon H Hanson, "The China syndrome: Local labor market effects of import competition in the United States," *American economic review*, 2013, *103* (6), 2121–2168.

Ayoubi, Charles, Michele Pezzoni, and Fabiana Visentin, "At the origins of learning: Absorbing knowledge flows from within the team," *Journal of Economic Behavior & Organization*, 2017, *134*, 374–387.

Azoulay, Pierre, Joshua S Graff Zivin, and Jialan Wang, "Superstar extinction," *The Quarterly Journal of Economics*, 2010, *125* (2), 549–589.

Balland, Pierre-Alexandre, Cristian Jara-Figueroa, Sergio G Petralia, Mathieu PA Steijn, David L Rigby, and César A Hidalgo, "Complex economic activities concentrate in large cities," *Nature human behaviour*, 2020, *4* (3), 248–254.

Bean, Randy, "How Moderna Is Embracing Data And AI To Transform Drug Discovery," *Forbes*, 2024.

Becker, Richard A. and Allan R. Wilks, "Package 'Maps'," https://cran.r-project.org/web/packages/maps/maps.pdf 2018. Accessed: 2021-05-11.

Ben-Porath, Yoram, "The production of human capital and the life cycle of earnings," *Journal of Political Economy*, 1967, *75* (4), 352–365.

Bettencourt, Luís MA, José Lobo, Dirk Helbing, Christian Kühnert, and Geoffrey B West, "Growth, innovation, scaling, and the pace of life in cities," *Proceedings of the national academy of sciences*, 2007, *104* (17), 7301–7306.

Bikard, Michaël and Matt Marx, "Bridging academia and industry: How geographic hubs connect university science and corporate technology," *Management Science*, 2020, *66* (8), 3425–3443.

Bloom, Nicholas, Mark Schankerman, and John Van Reenen, "Identifying technology spillovers and product market rivalry," *Econometrica*, 2013, *81* (4), 1347–1393.

Boldrin, Michele and David K Levine, "The case against patents," *Journal of Economic Perspectives*, 2013, *27* (1), 3–22.

Buera, Francisco J and Joseph P Kaboski, "The rise of the service economy," *American Economic Review*, 2012, *102* (6), 2540–2569.

Carlino, Gerald A., Jake Carr, Robert M. Hunt, and Tony E. Smith, "The Agglomeration of R&D Labs," *Federal Reserve Bank of Philadelphia*, 2012.

Carlino, Gerald A, Satyajit Chatterjee, and Robert M Hunt, "Urban density and the rate of invention," *Journal of Urban Economics*, 2007, *61* (3), 389–419.

Carlino, Gerald and William R. Kerr, "Agglomeration and Innovation," *Handbook of Regional and Urban Economics*, 2015, *5*, 349–404.

Casalnuovo, Casey, Bogdan Vasilescu, Premkumar Devanbu, and Vladimir Filkov, "Developer onboarding in GitHub: the role of prior social links and language experience," in "Proceedings of the 2015 10th joint meeting on foundations of software engineering" 2015, pp. 817–828.

Catalini, Christian, "Microgeography and the direction of inventive activity," *Management Science*, 2018, *64* (9), 4348–4364.

Cattaneo, Matias D., Richard K. Crump, Max H. Farrell, and Yingjie Feng, "Binscatter Regressions," *Unpublished*, 2023.

_ , _ , _ , and _ , "On Binscatter," *American Economic Review*, 2024.

Chattergoon, Brad and William R. Kerr, "Winner takes all? Tech clusters, population centers, and the spatial transformation of US invention," *Research Policy*, 2022, *51* (2), 104418.

Cohen, Julie E and Mark A Lemley, "Patent scope and innovation in the software industry," *Calif. L. Rev.*, 2001, *89*, 1.

Combes, Pierre-Philippe and Laurent Gobillon, "The empirics of agglomeration economies," in "Handbook of regional and urban economics," Vol. 5, Elsevier, 2015, pp. 247–348.

_ , Gilles Duranton, Laurent Gobillon, and Sébastien Roux, "Estimating agglomeration economies with history, geology, and worker effects," in "Agglomeration economics," University of Chicago Press, 2010, pp. 15–66.

Cornelissen, Thomas, Christian Dustmann, and Uta Schönberg, "Peer effects in the workplace," *American Economic Review*, 2017, *107* (2), 425–56.

Deming, David J. and Kadeem Noray, "Earnings dynamics, changing job skills, and STEM careers," *Quarterly Journal of Economics*, 2020, *135* (4), 1965–2005.

Dohmke, Thomas, "100 million developers and counting," *GitHub Blog*, 2023.

Elazhary, Omar, Margaret-Anne Storey, Neil Ernst, and Andy Zaidman, "Do as i do, not as i say: Do contribution guidelines match the github contribution process?," in "2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)" IEEE 2019, pp. 286–290.

Ellison, Glenn and Edward L Glaeser, "Geographic concentration in US manufacturing industries: a dartboard approach," *Journal of Political Economy*, 1997, *105* (5), 889–927.

Fackler, Thomas A, Yvonne Giesing, and Nadzeya Laurentsyeva, "Knowledge remittances: Does emigration foster innovation?," *Research policy*, 2020, *49* (9), 103863.

Forman, Chris and Avi Goldfarb, "Concentration and Agglomeration of IT Innovation and Entrepreneurship: Evidence from Patenting," in "The Role of Innovation and Entrepreneurship in Economic Growth," University of Chicago Press, 2022, pp. 95–122.

Ganguli, Ina, Jeffrey Lin, and Nicholas Reynolds, "The paper trail of knowledge spillovers: Evidence from patent interferences," *American Economic Journal: Applied Economics*, 2020, *12* (2), 278–302.

Glaeser, Edward L, "Learning in cities," *Journal of Urban Economics*, 1999, *46* (2), 254–277.

Goldbeck, Moritz, "Bit by bit: colocation and the death of distance in software developer networks," *CRC Discussion Paper*, 2023.

Gousios, G., "The GHTorrent Dataset and Tool Suite," in "Proceedings of the 10th Working Conference on Mining Software Repositories" IEEE Press San Francisco 2013, p. 233–236.

Guzman, Jorge and Scott Stern, "The state of American entrepreneurship: New estimates of the quantity and quality of entrepreneurship for 32 US States, 1988–2014," *American Economic Journal: Economic Policy*, 2020, *12* (4), 212–243.

Herbst, Daniel and Alexandre Mas, "Peer effects on worker output in the laboratory generalize to the field," *Science*, 2015, *350* (6260), 545–549.

Jackson, C Kirabo and Elias Bruegmann, "Teaching students and teaching each other: The importance of peer learning for teachers," *American Economic Journal: Applied Economics*, 2009, *1* (4), 85–108.

Jaffe, Adam B, Manuel Trajtenberg, and Rebecca Henderson, "Geographic localization of knowledge spillovers as evidenced by patent citations," *The Quarterly Journal of Economics*, 1993, *108* (3), 577–598.

Johnson, Kenneth P. and John R. Kort, "2004 Redefinition of the BEA Economic Areas," *Survey of Current Business*, 2004, *75* (2), 75–81.

Jones, Benjamin F, "The burden of knowledge and the "death of the renaissance man": Is innovation getting harder?," *The Review of Economic Studies*, 2009, *76* (1), 283–317.

_ , "Age and great invention," *The Review of Economics and Statistics*, 2010, *92* (1), 1–14.

Kaltenberg, Mary, Adam B. Jaffe, and Margie E. Lachman, "Invention and the life course: Age differences in patenting," *Research Policy*, 2023, *52* (1), 104629.

Kerr, William R. and Frederic Robert-Nicoud, "Tech Clusters," *Journal of Economic Perspectives*, 2020, *34* (3), 50–76.

Kogan, Leonid, Dimitris Papanikolaou, Amit Seru, and Noah Stoffman, "Technological innovation, resource allocation, and growth," *The Quarterly Journal of Economics*, 2017, *132* (2), 665–712.

la Roca, Jorge De and Diego Puga, "Learning by working in big cities," *The Review of Economic Studies*, 2017, *84* (1), 106–142.

Lerner, Josh and Amit Seru, "The use and misuse of patent data: Issues for finance and beyond," *The Review of Financial Studies*, 2022, *35* (6), 2667–2704.

Lima, Antonio, Luca Rossi, and Mirco Musolesi, "Coding together at scale: GitHub as a collaborative social network," *Proceedings of the international AAAI conference on web and social media*, 2014, *8* (1), 295–304.

Lin, Yu-Kai and Arun Rai, "The scope of software patent protection in the digital age: evidence from alice," *Information Systems Research*, 2024, *35* (2), 657–672.

Lucas, Robert E., "On the mechanics of economic development," *Journal of monetary economics*, 1988, *22* (1), 3–42.

Manski, Charles F, "Dynamic choice in social settings: Learning from the experiences of others," *Journal of Econometrics*, 1993, *58* (1-2), 121–136.

Mas, Alexandre and Enrico Moretti, "Peers at work," *American Economic Review*, 2009, *99* (1), 112–45.

Moretti, Enrico, "Workers' education, spillovers, and productivity: evidence from plant-level production functions," *American Economic Review*, 2004, *94* (3), 656–690.

_ , "The effect of high-tech clusters on the productivity of top inventors," *American Economic Review*, 2021, *111* (10), 3328–75.

Patel, Ravikumar, "Software Development Statistics: Market Trends and Insights," *Radix*, 2024.

Rosenthal, Stuart S and William C Strange, "How close is close? The spatial reach of agglomeration economies," *Journal of Economic Perspectives*, 2020, *34* (3), 27–49.

Sacerdote, Bruce, "Peer effects with random assignment: Results for Dartmouth roommates," *The Quarterly journal of economics*, 2001, *116* (2), 681–704.

_ , "Experimental and quasi-experimental analysis of peer effects: two steps forward?," *Annu. Rev. Econ.*, 2014, *6* (1), 253–272.

Simplemaps, "United States Cities Database," https://simplemaps.com/data/us-cities 2021. Accessed: 2021-05-10.

StackOverflow, "Stack Overflow Developer Survey 2020," https://insights.stackoverflow.com/survey/2020#overview 2020. Accessed: 2022-05-30.

Stackoverflow, "Developer Survey," *Report*, 2024.

Steinwender, Claudia, "Real effects of information frictions: When the states and the kingdom became united," *American Economic Review*, 2018, *108* (3), 657–696.

Tambe, Prasanna, Lorin Hitt, Daniel Rock, and Erik Brynjolfsson, "Digital Capital and Superstar Firms," *NBER Working Paper*, 2020.

Verspagen, Bart and Wilfred Schoenmakers, "The spatial dimension of patenting by multinational firms in Europe," *Journal of Economic Geography*, 2004, *4* (1), 23–42.

Vidoni, Melina, "A systematic process for Mining Software Repositories: Results from a systematic literature review," *Information and Software Technology*, 2022, *144*, 106791.

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.

Waldinger, Fabian, "Peer effects in science: Evidence from the dismissal of scientists in Nazi Germany," *The review of economic studies*, 2012, *79* (2), 838–861.

Webb, Michael, Nick Short, Nicholas Bloom, and Josh Lerner, "Some facts of high-tech patenting," *NBER Working Paper*, 2018.

Wuchty, Stefan, Benjamin F Jones, and Brian Uzzi, "The increasing dominance of teams in production of knowledge," *Science*, 2007, *316* (5827), 1036–1039.

A Appendix

A.1 Tables

Table A.1: Summary statistics

Variable	var	mean	median	min.	max.
Activity					
Commits per user	691,964,175.97	2,298	1,059	25	3,767,493.00
Commit per user per Snapshot	16,341,488.33	230	56	1	1,299,828.00
Technology per user per snapshot	1.23	3	3	1	5.00
Technology per user	1.27	2	2	1	5.00
Programming language per user per snapshot	4.19	7	7	1	18.00
Programming language per user	7.58	3	3	1	17.00
Projects					
Users per project	27.62	2	1	1	2,381.00
Commits per project per snapshot	1.23	19	3	1	1,298,112.00
Stars per project	1,301,008.47	85	0	0	259,118.00
Forks per project	72,701.27	11	0	0	145,997.00
Project age [years]	6.10	5	5	0	13.36
Own project	0.24	0	0	0	1.00
Business share	0.14	1	1	0	1.00
Weekend share	0.09	0	0	0	1.00
Out of hour share	0.11	0	0	0	1.00
Local share	0.05	1	1	0	1.00
Clusters					
Technology per city	0.88	5	5	1	5.00
Technology per city per snapshot	1.84	4	4	1	5.00
Programming language per city	22.05	14	17	1	18.00
Programming language per city per snapshot	27.15	10	11	1	18.00

Table A.2: Top 10 clusters by technology

City	cluster size
Technology 1	
San Jose-San Francisco-Oakland, CA	0.10638
New York-Newark-Bridgeport, NY-NJ-CT-PA	0.08784
Seattle-Tacoma-Olympia, WA	0.05366
Los Angeles-Long Beach-Riverside, CA	0.04403
Indianapolis-Anderson-Columbus, IN	0.04387
Toronto	0.03732
Washington-Baltimore-Northern Virginia, DC-MD-VA-WV	0.03484
Boston-Worcester-Manchester, MA-NH	0.03233
Chicago-Naperville-Michigan City, IL-IN-WI	0.03200
Dallas-Fort Worth, TX	0.02390
Technology 2	
San Jose-San Francisco-Oakland, CA	0.13441
New York-Newark-Bridgeport, NY-NJ-CT-PA	0.09031
Seattle-Tacoma-Olympia, WA	0.05627
Boston-Worcester-Manchester, MA-NH	0.04299
Los Angeles-Long Beach-Riverside, CA	0.04095
Washington-Baltimore-Northern Virginia, DC-MD-VA-WV	0.04039
Toronto	0.03375
Indianapolis-Anderson-Columbus, IN	0.03250
Chicago-Naperville-Michigan City, IL-IN-WI	0.03073
Denver-Aurora-Boulder, CO	0.02385
Technology 3	
San Jose-San Francisco-Oakland, CA	0.14154
New York-Newark-Bridgeport, NY-NJ-CT-PA	0.11862
Seattle-Tacoma-Olympia, WA	0.04749
Chicago-Naperville-Michigan City, IL-IN-WI	0.04181
Los Angeles-Long Beach-Riverside, CA	0.04000
Washington-Baltimore-Northern Virginia, DC-MD-VA-WV	0.03808
Boston-Worcester-Manchester, MA-NH	0.03748
Denver-Aurora-Boulder, CO	0.03744
Toronto	0.03028
Indianapolis-Anderson-Columbus, IN	0.02649
Technology 4	
San Jose-San Francisco-Oakland, CA	0.13405
New York-Newark-Bridgeport, NY-NJ-CT-PA	0.08113
Indianapolis-Anderson-Columbus, IN	0.05250
Seattle-Tacoma-Olympia, WA	0.05092
Los Angeles-Long Beach-Riverside, CA	0.04059
Toronto	0.03623
Washington-Baltimore-Northern Virginia, DC-MD-VA-WV	0.03382
Boston-Worcester-Manchester, MA-NH	0.03259
Chicago-Naperville-Michigan City, IL-IN-WI	0.03208
Dallas-Fort Worth, TX	0.02942
Technology 5	
San Jose-San Francisco-Oakland, CA	0.13050
New York-Newark-Bridgeport, NY-NJ-CT-PA	0.06623
Seattle-Tacoma-Olympia, WA	0.05999
Los Angeles-Long Beach-Riverside, CA	0.04048
Indianapolis-Anderson-Columbus, IN	0.03647
Boston-Worcester-Manchester, MA-NH	0.03572
Washington-Baltimore-Northern Virginia, DC-MD-VA-WV	0.03342
Toronto	0.02954
Dallas-Fort Worth, TX	0.02884
Dayton-Springfield-Greenville, OH	0.02735

Table A.3: User sample

Dep. var.: Commits [log]	(1)	(2)	(3)	(4)	(5)
Min. time intervals with consecutive activity:	1	2	3	4	5
Cluster size [log]	0.1989*	0.1986*	0.1894*	0.1911*	0.2104**
	(0.1082)	(0.1076)	(0.1029)	(0.1018)	(0.1005)
Users	243,443	229,140	124,808	81,459	55,458
Observations	6,363,687	6,320,343	5,295,433	4,636,989	4,053,452
Adj. R^2	0.173	0.180	0.227	0.249	0.261
Dep. var.: Commits [log]	(6)	(7)	(8)	(9)	(10)
Min. time intervals with consecutive activity:	6	7	8	9	10
Cluster size [log]	0.2209**	0.2175**	0.2432**	0.2524**	0.2775**
	(0.1027)	(0.1043)	(0.1081)	(0.1127)	(0.1255)
Users	45,007	38,157	31,669	27,011	21,116
Observations	3,722,638	3,470,489	3,197,671	2,961,254	2,527,496
Adj. R^2	0.268	0.273	0.278	0.284	0.292

Notes: Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

Table A.4: Cluster size (number of users)

Dep. var.: Commits [log]	(1)	(2)	(3)	(4)	(5)	(6)
Cluster size [absolute, log]	-0.2367***	0.1070	0.0929	0.1966**	0.1935**	0.2777**
	(0.0607)	(0.0785)	(0.0744)	(0.0949)	(0.0962)	(0.1253)
Fixed-effects						
User	Yes	Yes	Yes	Yes	Yes	Yes
Project	Yes	Yes	Yes	Yes	Yes	Yes
Technology	Yes	Yes	Yes	Yes	Yes	Yes
Language	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Time	Yes	Yes	Yes	Yes	Yes	Yes
Technology × time		Yes	Yes	Yes	Yes	Yes
Language × time			Yes	Yes	Yes	Yes
City × technology				Yes	Yes	Yes
City × language					Yes	Yes
City × time						Yes
Users	21,116	21,116	21,116	21,116	21,116	21,116
Observations	2,527,496	2,527,496	2,527,496	2,527,496	2,527,496	2,527,496
Adj. R^2	0.288	0.289	0.290	0.291	0.291	0.292

Notes: Language refers to programming language. Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

Table A.5: Robustness (excluding largest projects)

Dep. var.: Commits [log]	(1)	(2)	(3)	(4)	(5)	(6)
Cluster size [log]	0.1006	0.0927	0.0787	0.1625*	0.1586*	0.2773**
	(0.1002)	(0.0674)	(0.0636)	(0.0941)	(0.0954)	(0.1168)
Fixed effects						
User	Yes	Yes	Yes	Yes	Yes	Yes
Project	Yes	Yes	Yes	Yes	Yes	Yes
Technology	Yes	Yes	Yes	Yes	Yes	Yes
Language	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Time	Yes	Yes	Yes	Yes	Yes	Yes
Technology × time		Yes	Yes	Yes	Yes	Yes
Language × time			Yes	Yes	Yes	Yes
City × technology				Yes	Yes	Yes
City × language					Yes	Yes
City × time						Yes
Users	20,905	20,905	20,905	20,905	20,905	20,905
Observations	2,382,259	2,382,259	2,382,259	2,382,259	2,382,259	2,382,259
Adjusted R^2	0.256	0.258	0.260	0.260	0.260	0.261

Notes: Language refers to programming language. Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

Table A.6: Robustness (excluding most active users)

Dep. var.: Commits [log]	(1)	(2)	(3)	(4)	(5)	(6)
Cluster size [log]	0.0841***	0.0797**	0.0672**	0.1353**	0.1341**	0.1724**
	(0.0321)	(0.0325)	(0.0321)	(0.0595)	(0.0596)	(0.0798)
Fixed effects						
User	Yes	Yes	Yes	Yes	Yes	Yes
Project	Yes	Yes	Yes	Yes	Yes	Yes
Technology	Yes	Yes	Yes	Yes	Yes	Yes
Language	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Time	Yes	Yes	Yes	Yes	Yes	Yes
Technology × time		Yes	Yes	Yes	Yes	Yes
Language × time			Yes	Yes	Yes	Yes
City × technology				Yes	Yes	Yes
City × language					Yes	Yes
City × time						Yes
Users	20,905	20,905	20,905	20,905	20,905	20,905
Observations	2,277,873	2,277,873	2,277,873	2,277,873	2,277,873	2,277,873
Adjusted R^2	0.283	0.284	0.285	0.285	0.285	0.286

Notes: The 1% most active users (476 users) are excluded. Language refers to programming language. Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

Table A.7: Robustness (excluding largest clusters)

Dep. var.: Commits [log]	(1)	(2)	(3)	(4)	(5)	(6)
Cluster size [log]	0.1239	0.1133	0.0978	0.2006**	0.1973**	0.3003**
	(0.1143)	(0.0824)	(0.0776)	(0.0960)	(0.0975)	(0.1305)
Fixed effects						
User	Yes	Yes	Yes	Yes	Yes	Yes
Project	Yes	Yes	Yes	Yes	Yes	Yes
Technology	Yes	Yes	Yes	Yes	Yes	Yes
Language	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Time	Yes	Yes	Yes	Yes	Yes	Yes
Technology × time		Yes	Yes	Yes	Yes	Yes
Language × time			Yes	Yes	Yes	Yes
City × technology				Yes	Yes	Yes
City × language					Yes	Yes
City × time						Yes
Users	20,640	20,640	20,640	20,640	20,640	20,640
Observations	2,451,163	2,451,163	2,451,163	2,451,163	2,451,163	2,451,163
Adjusted R^2	0.289	0.291	0.293	0.293	0.293	0.294

Notes: The 5% largest cities (10 cities) are excluded. Language refers to programming language. Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

Table A.8: Quality (forks)

Dep. var.: Commits [log]	(1)	(2)	(3)	(4)	(5)	(6)
Cluster size [log]	0.1438	0.1354	0.1234	0.2779***	0.2742***	0.3789***
	(0.1030)	(0.0851)	(0.0815)	(0.0881)	(0.0891)	(0.1448)
Fixed effects						
User	Yes	Yes	Yes	Yes	Yes	Yes
Project	Yes	Yes	Yes	Yes	Yes	Yes
Technology	Yes	Yes	Yes	Yes	Yes	Yes
Language	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Time	Yes	Yes	Yes	Yes	Yes	Yes
Technology × time		Yes	Yes	Yes	Yes	Yes
Language × time			Yes	Yes	Yes	Yes
City × technology				Yes	Yes	Yes
City × language					Yes	Yes
City × time						Yes
Users	7,135	7,135	7,135	7,135	7,135	7,135
Observations	427,991	427,991	427,991	427,991	427,991	427,991
Adjusted R^2	0.405	0.405	0.406	0.408	0.409	0.411
$\Delta(\beta_{top10} - \beta_{all})$	0.0294	0.0284	0.0305	0.0813	0.0807	0.1012
$\Delta(\beta_{top10} - \beta_{all})/\beta_{all}$	0.2045	0.2097	0.2472	0.2926	0.2943	0.2671

Notes: Regressions based on the top decile of projects by forks. These are 7,135 projects with at least four forks.β_{top10} denotes the estimated coefficient on cluster size. β_{all} refers to the estimated coefficient of cluster size from the corresponding specification in Table 1. Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

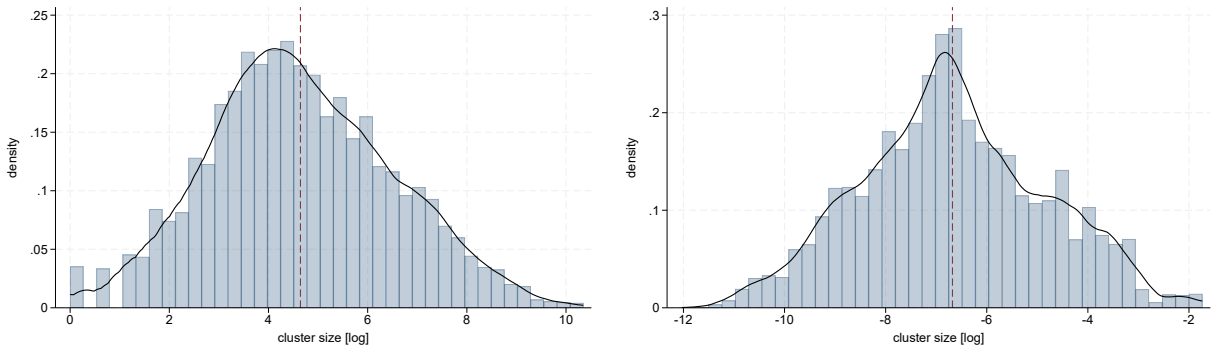
Table A.9: Dynamic estimates

Dep. var.: Commits [log]	(1)	(2)	(3)	(4)	(5)	(6)
$\beta(t=-1)$	0.0001	-0.0006	-0.0004	-0.0001	0.0011	0.0015
	(0.0101)	(0.0089)	(0.0087)	(0.0089)	(0.0091)	(0.0094)
$\beta(t=0)$	0.1204	0.1120	0.0973	0.1301	0.1302	0.2676**
	(0.1097)	(0.0793)	(0.0753)	(0.1232)	(0.1239)	(0.1267)
$\beta(t=1)$	-0.0021	-0.0023	-0.0029	-0.0027	-0.0023	-0.0016
	(0.0109)	(0.0111)	(0.0112)	(0.0111)	(0.0112)	(0.0110)
Fixed effects						
User	Yes	Yes	Yes	Yes	Yes	Yes
Project	Yes	Yes	Yes	Yes	Yes	Yes
Technology	Yes	Yes	Yes	Yes	Yes	Yes
Language	Yes	Yes	Yes	Yes	Yes	Yes
City	Yes	Yes	Yes	Yes	Yes	Yes
Time	Yes	Yes	Yes	Yes	Yes	Yes
Technology × time		Yes	Yes	Yes	Yes	Yes
Language × time			Yes	Yes	Yes	Yes
City × technology				Yes	Yes	Yes
City × language					Yes	Yes
City × time						Yes
Users	21,116	21,116	21,116	21,116	21,116	21,116
Observations	1,532,335	1,532,335	1,532,335	1,532,335	1,532,335	1,532,335
Adjusted R^2	0.331	0.332	0.333	0.334	0.334	0.335

Notes: Robust standard errors clustered at the city × technology level in parenthesis. *** $p < 0.01$, ** $p < 0.05$, and * $p < 0.1$. *Sources:* GHTorrent, own calculations.

A.2 Figures

Figure A.1: Technology cluster size distribution

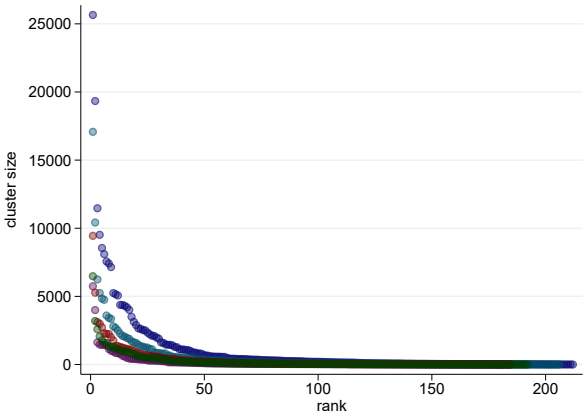


(a) number of users in cluster

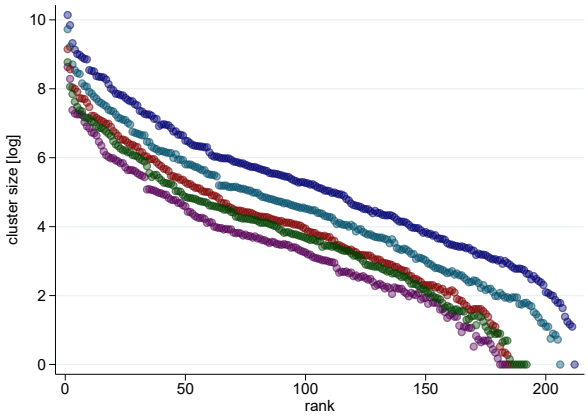
(b) cluster size

Sources: GHTorrent, own calculations.

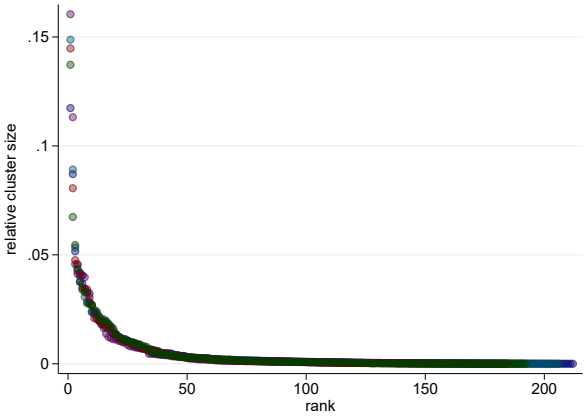
Figure A.2: Agglomeration by technology



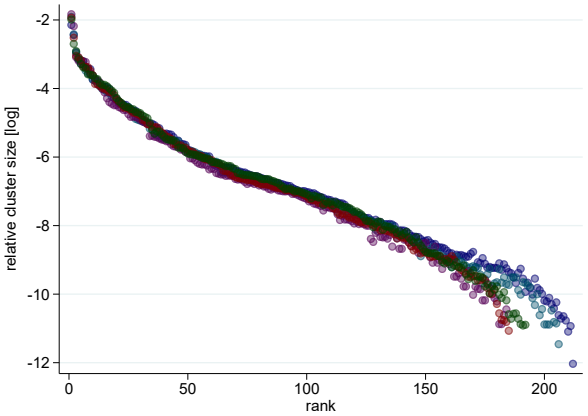
(a) rank-size (levels)



(b) power law (levels)



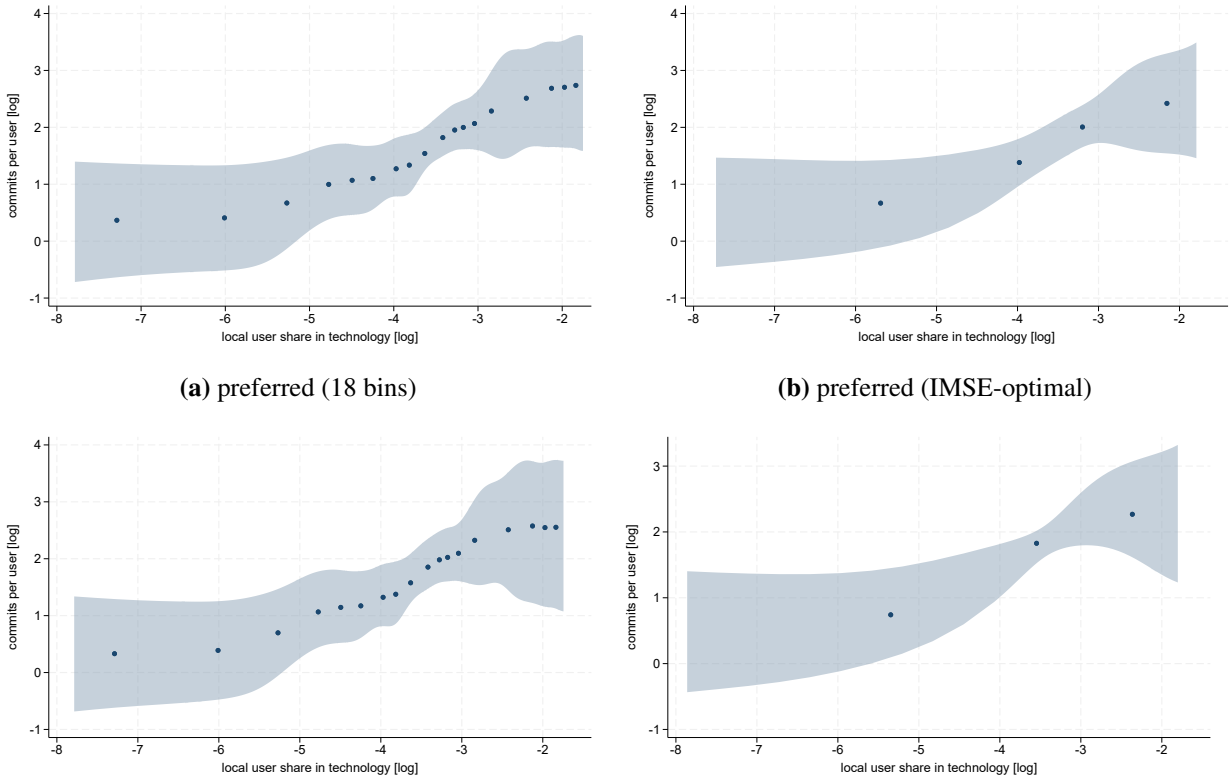
(c) rank-size (share)



(d) power law (share)

Sources: GHTorrent, own calculations.

Figure A.3: Binscatter specification



(a) preferred (18 bins)

(b) preferred (IMSE-optimal)

(c) extended (18 bins)

(d) extended (IMSE-optimal)

Notes: Graph plots a binscatter representation of the relationship between software engineer productivity and cluster size using `binsreg` (Cattaneo et al., 2023). Our preferred specification includes fixed effects for time, technology, language, project, city, and user as well as for time × city, time × technology, and city × technology. The extended specification additionally features time × language and language × city fixed effects. Shaded areas represent 90% confidence intervals. *Sources:* GHTorrent, own calculations.