Peer Effects among Software Engineers: Evidence from Github

- **Research question:** How do software engineers affect each other's productivity?
- **Relevance:** Most evidence on peer effects, i.e., how workers affect each other, stem from manual workers. Yet, high-skilled professions such as software engineers predominantly work in teams. This project is the first to quantify productivity spillovers among software engineers. Quantifying these spillovers allows to infer optimal team composition (i.e., if we pair engineers with very high and very low productivity, does this increase overall productivity at the firm level?)
- **Idea:** Exploit variation in peers' commit-frequency working on the same projects (=repositories) across time to identify productivity responses of individual software engineers
- **Data:** Based on the universe of activity-logs of Github (January 2005-June 2019; ~1TB of metadata on users, repositories, commits, pull requests etc.), I identify so-called commercial open source software companies (COSS; e.g., Docker, Elastic, nextcloud, jitsi), which are companies, which host the software code publicly implying that I can observe all contributions of developers to these companies.
- **Data cleaning and feature engineering:** Define the number of weekly/monthly commits as my measure of productivity (in a first step, ignore opening/closing of issues and discussions in PRs, which do not result in a commit). Need to exclude accounts which are bots (e.g., some companies use CI/CD tools, which commit themselves) and isolate authors of commits rather than commiters. Moreover, I focus on potential full-time employees of these firms (identified based on their contribution-patterns; for now, I restrict attention to users that have at least 50 commits and are active for more than 6 months).
  - *To do:* Need to incorporate information from opening issues, and discussions as part of pull requests as other productive dimensions. Potentially, can also move to the actual code rather than the metadata and analyse the code quality at a later point.
- Approach: Set up an economic model in which own productivity is based on own ability and a spillover effect δ of peer ability:

$$\underbrace{y_{ij(i)t}}_{\substack{\#\text{ contributions of}\\ i\text{ in project } j(i)\\ \text{at time } t}} = \underbrace{\alpha_i}_{\substack{\text{Own}\\ \text{ability}}} + \delta \underbrace{f_{j(i)t}(\alpha_k; k \neq i)}_{\text{Peer ability}} + \underbrace{\rho_{j(i)}}_{\substack{\text{Project}\\ \text{fixed effect}}} + \underbrace{\tau_t}_{\substack{\text{Time}\\ \text{fixed effect}}} + \underbrace{\epsilon_{ij(i)t}}_{\substack{\text{Unobservable}\\ \text{factors}}}$$

- Define "peer ability" as follows: average of peer k's ability, weighted by k's share of contributions to project $j(i)$ at time t and i's exposure to peers from different projects (based on i's own contributions):

$$f_{j(i)t}(\alpha_k; k \neq i) = \sum_{j(i)} \underbrace{\frac{c_{ij(i)t}}{\sum_{j(i)} c_{ij(i)t}}}_{\substack{i\text{'s relative share} \\ \text{of project} j(i) \text{ at } t}} \sum_k \underbrace{\frac{c_{kj(i)t}}{\sum_l c_{lj(i)t}}}_{\substack{k\text{'s relative share} \\ \text{in project} j(i) \text{ at } t}} \underbrace{\alpha_k}_{\text{Ability FE}}$$

- To estimate the model, solve the following optimization problem:

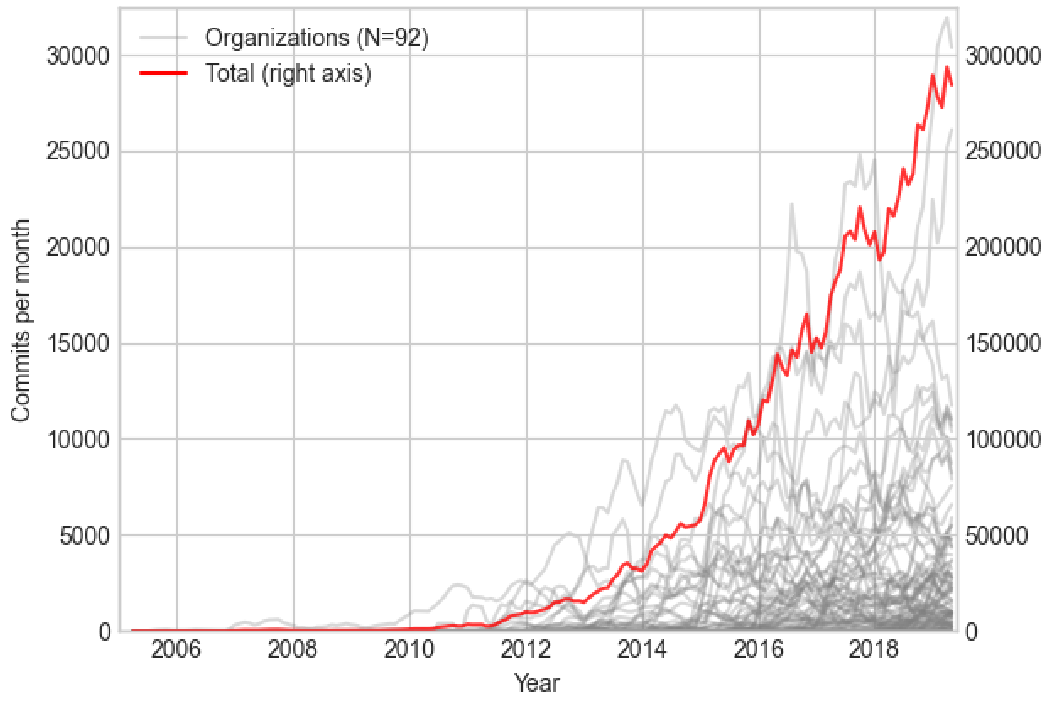
$$\min_{\alpha,\delta,\rho,\tau} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(y_{ij(i)t} - \alpha_i - \delta f_{j(i)t}(\alpha_k; k \neq i)\rho_{j(i)} - \tau_t \right)^2$$

  - Problem: High dimensionality (one fixed effect α_i for each engineer), leading to a curse of dimensionality
  - Solution: Can show that an iterative procedure (estimating α-terms first given the others, estimate δ given the other parameters etc.) converges to the true non-linear least squares solution.

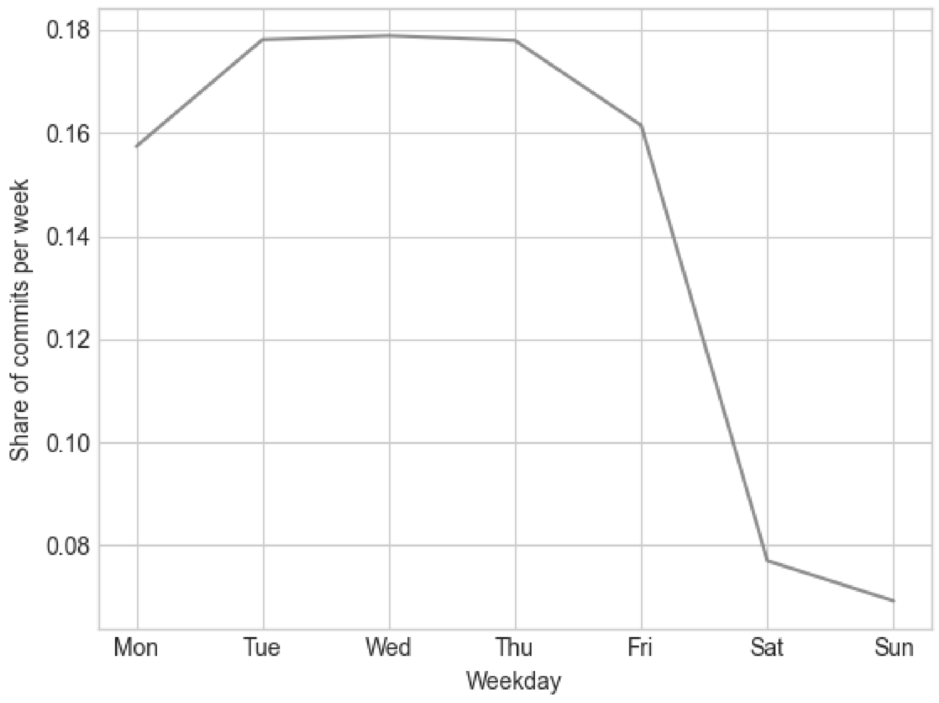
- Estimating this model, requires variation in peer ability as defined above. Some exploratory analyses to support this follow.

(Preliminary) Exploratory analyses

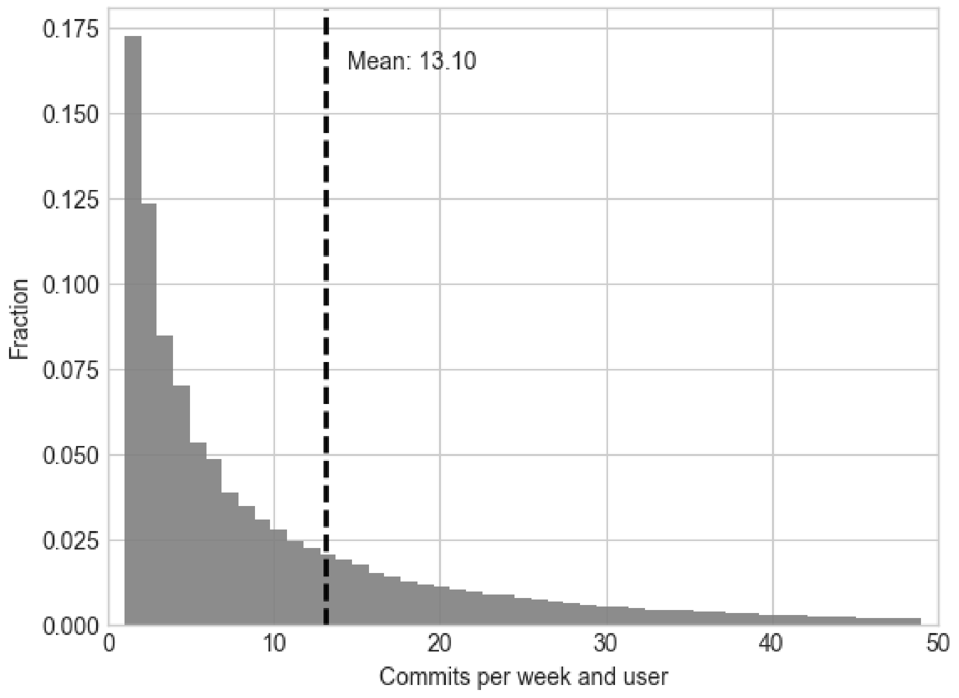
- Number of monthly contributions increases over time to almost 300.000 contributions as of mid-2019



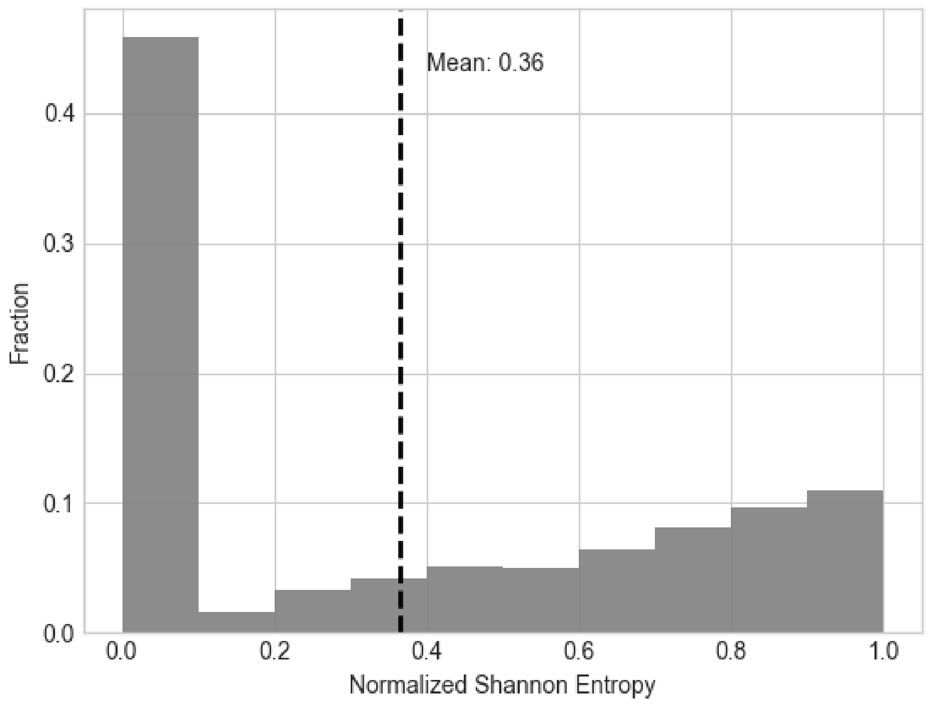
- Consistent with my selection procedure to identify full-time employees (FTEs), I observe the vast majority of contributions are done on weekdays



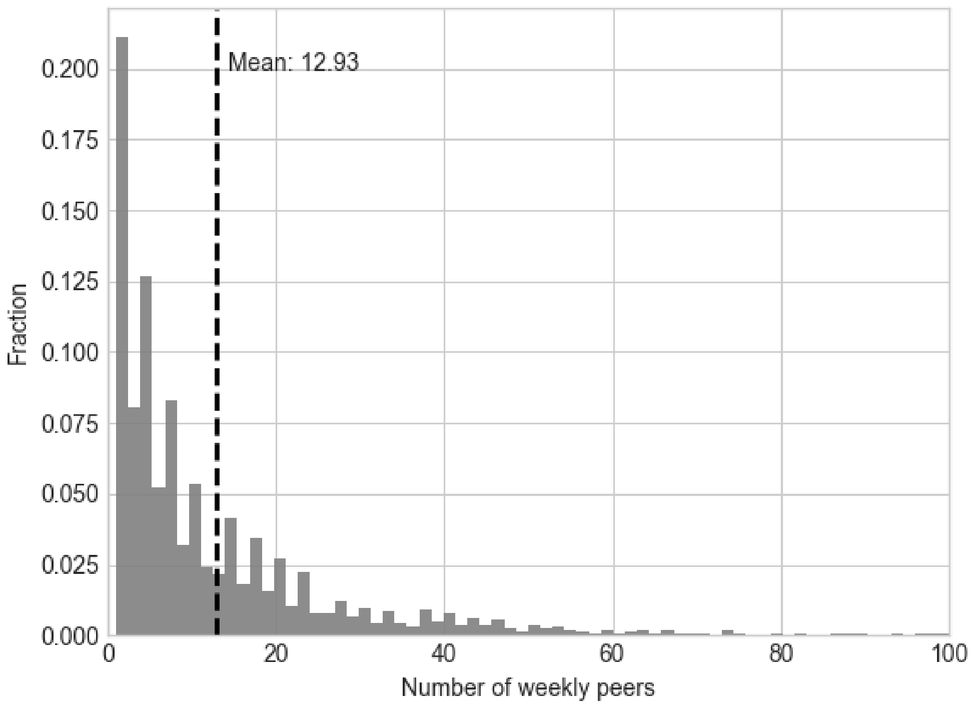
- On average, users make about 13 commits per week; strongly skewed distribution

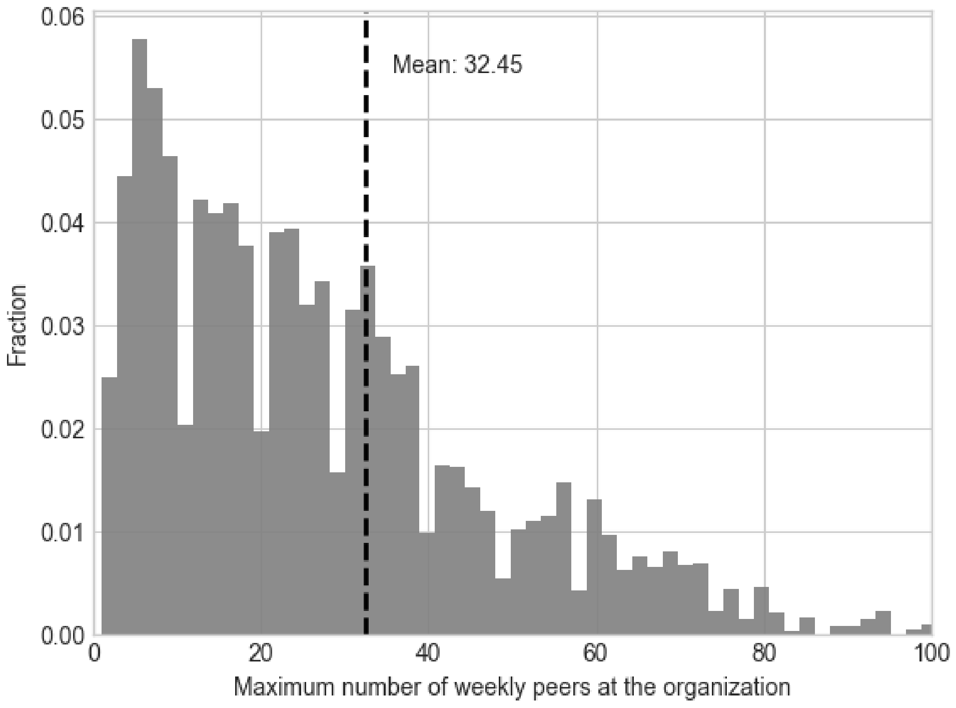


- Plot the (normalized) Shannon entropy associated with a users weekly contributions to check whether they work on one project (Shannon entropy of 0) or distribute their work effort across many projects (entropy larger than 1). If the Shannon entropy equals one, users contribute equally to all projects they are actively working on. While 47% of users contribute only to a single repository per week, 53% contribute to more than one. The mean number of repositories each developer is contributing to each week is approx. 2.5 (not shown), while for each organization, there are on average more than 20 active repositories every week.



- How much overlap is between users in a given company/organization? To study this, compare for each user the number of unique peers this user is interacting with to the number of peers working on any project in the company. On average, each user observed about 13 other developers, while there are 32 developers on average who are active in any given week. Thus, there is sufficient variation in the peers each developer is observing, supporting the empirical strategy outlined above.


Mean: 12.93
Fraction
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
0
20
40
60
80
100
Number of weekly peers


Mean: 32.45
Fraction
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0
20
40
60
80
100
Maximum number of weekly peers at the organization