Generative AI and Market Structure

in Open-Source Software: Evidence from GitHub Copilot

Michail Batikas
Nova School of Business and Economics

July 23, 2026 | NBER SI 2026, Digital Economics & AI

Does AI concentrate the markets it mediates?

Individual-level productivity gains are established (Peng et al. 2023; Cui et al. 2026), and project-level activity rises (Hoffmann et al. 2024; Yeverechyahu et al. 2024). The open question is the **aggregate effect at the market level** when these gains add up.

Proliferation

AI lowers the cost of trying packages.

Developers reach further into the long tail.

Market HHI ↓

Bakos 1997; Brynjolfsson et al. 2003.

Consolidation

AI completions reflect their training prior.

Popular packages get surfaced over and over.

Market HHI ↑

Fleder & Hosanagar 2009; Calvano et al. 2025.

Both can be right at once, because adoption is not one decision.

Adoption has three layers, each a bigger commitment

Layer	Commitment	Action
L1 Engagement	**Low**	fork, star, PR
L2 Usage	**Medium**	install, run
L3 Integration	**High**	declare in manifest

Klemperer (1987): search cost → implementation cost → lock-in.
Cheap layers: falling search cost **broadens** engagement across many packages.
Costly layer: developers fall back on a prior (and the AI *is* that prior), so the market **concentrates** on incumbents.

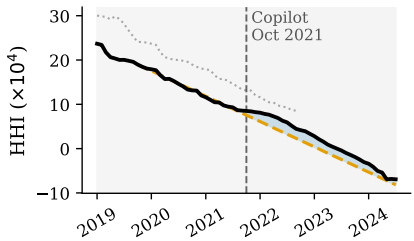
The natural experiment: GitHub Copilot, Python vs R

- Python: full GitHub Copilot support. R: none.
- Why Oct 2021: opens to JetBrains/Neovim, access jumps <5K to ~100K in a month.
- Analysis at **two levels**: the language ecosystem as a market, and the individual package.
- Sample: 1,351 **multi-homing** packages on Python's PyPI and R's CRAN, 36 months.

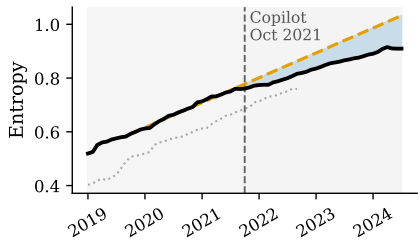
Estimand: intent-to-treat, the effect of Copilot *availability*, not of Copilot *use*. R has less exposure, not zero, so R-side use biases the contrast toward zero.

Estimator: Callaway & Sant'Anna (2021), chosen because it adjusts for pre-treatment differences (doubly robust) and lets the effect vary over time. Same result across various estimators.

The dependency market concentrates



(a) HHI, Python−R; line = Oct 2021, dashed = pre-trend



(b) Entropy, Python−R; line = Oct 2021, dashed = pre-trend

- HHI +1.90; entropy −0.030 ≈ **28 fewer effective packages** of 961.
- Oct-2020 placebo: null on all four metrics (HHI, entropy, top-10 share, Gini).
- Replicates in an independent Rust-vs-Haskell pair (+3.39, −0.055).
- Star, Fork, Contributor HHIs: all **null**.

Formal integration concentrates at the top

Sample	Forks (log)	Downloads (log)	Adoption flow (new dependents/month)
Full sample	+2.6%***	+10.3%***	n.s.
By pre-treatment dependency quartile (Q4 = the packages most built on)			
DepQ1	n.s.	n.s.	n.s.
DepQ2	n.s.	+8.7%***	n.s.
DepQ3	+2.8%**	+29.0%***	n.s.
DepQ4	**+5.4%***	**+8.8%****	**+0.219****

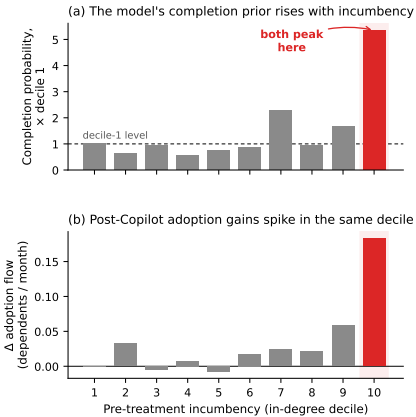
*$p < 0.05$, **$p < 0.01$, ***$p < 0.001$; n.s. = not statistically significant.

What the data show: the Engagement-Adoption Gap

	Package level	Market level
L1 Engagement	**broad effect**	null
L2 Usage	**broad effect**	null
L3 Integration	full-sample null, **top-quartile gain**	**effect**

Read each layer at the level where the decision is made: package level for L1/L2, market level for L3 (a project chooses a *set* of packages).

Mechanism: Copilot suggests what it already knows



- **The AI prefers the packages everyone already builds on**: ask Copilot-class models to complete real `import` statements, and they suggest top-quartile packages 2.33× as often.
- **Preference plus availability makes adoption**: the same preference exists on R, but the gains appear only on Python, where Copilot arrived.
- **Suggestion behavior, not corpus counts**: raw training-data frequency predicts nothing, while the model's completions predict the gains.

What to take home

Volume **up**, variety **down**.

Engagement broadens, integration concentrates. The *Engagement–Adoption Gap*.

≈ 28 fewer effective packages of **961**

a 0.030 entropy decline under Weitzman's translation

Lower bound: autocomplete leaves a human before the manifest. Agentic AI writes it directly.

Questions?

Thank you.

michail.batikas@novasbe.pt

References

Bakos, Y. (1997). Reducing buyer search costs: Implications for electronic marketplaces. *Management Science* 43(12), 1676–1692.

Brynjolfsson, E., Hu, Y. J., and Smith, M. D. (2003). Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers. *Management Science* 49(11), 1580–1596.

Callaway, B., and Sant'Anna, P. H. C. (2021). Difference-in-differences with multiple time periods. *Journal of Econometrics* 225, 200–230.

Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2025). Artificial intelligence, algorithmic recommendations and competition. SSRN Working Paper 4448010, CEPR DP 18176.

Cui, Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., and Salz, T. (2026). The effects of generative AI on high-skilled work: Evidence from three field experiments with software developers. *Management Science*, Articles in Advance.

Fleder, D., and Hosanagar, K. (2009). Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management Science* 55(5), 697–712.

Hoffmann, M., Boysel, S., Nagle, F., Peng, S., and Xu, K. (2024). Generative AI and the nature of work. Harvard Business School Working Paper 25-021.

Klemperer, P. (1987). Markets with consumer switching costs. *The Quarterly Journal of Economics* 102(2), 375–394.

Peng, S., Kalliamvakou, E., Cihon, P., and Demiralp, C. (2023). The impact of AI on programming: A controlled experiment with GitHub Copilot. arXiv:2302.06590.

Weitzman, M. L. (1992). On diversity. *The Quarterly Journal of Economics* 107(2), 363–405.

Yeverechyahu, D., Mayya, R., and Oestreicher-Singer, G. (2024). The impact of large language models on open-source innovation: Evidence from GitHub Copilot. SSRN Working Paper.