

The Impact of Large Language Models on Open-Source Innovation: Evidence from GitHub Copilot

Raveesh K. Mayya

Stern School of Business
New York University

Joint Work: Doron Yeverechyahu and Gal Oestreicher-Singer
Tel Aviv University, Israel

Georgetown-Worldbank Conference on AI, **Washington DC**
4th September 2024

Impact of LLM on Productivity

Potential to Enhance Individual Productivity

Whether this will translate to Group Productivity?



(Brynjolfsson et al. 2023; Noe and Zhang 2023; Peng et al. 2023)

Collaboration = $f(\langle iteration \rangle, \langle origination \rangle)$

Iterative Tasks

Interpolative
Thinking

- Well-Defined Solution Space
- Inside-the-box Solutions

**Origination
Tasks**

Extrapolative
Thinking

- Vague Solution Space
- Out-of-the-Box Solutions

Do LLMs have an asymmetric impact on the 2 types ?

Open-Source Innovation

Collaborative Innovation on
Public Repositories

Need both “out-of-the-box” and
“inside-the-box” thinking

There is an LLM for that :)



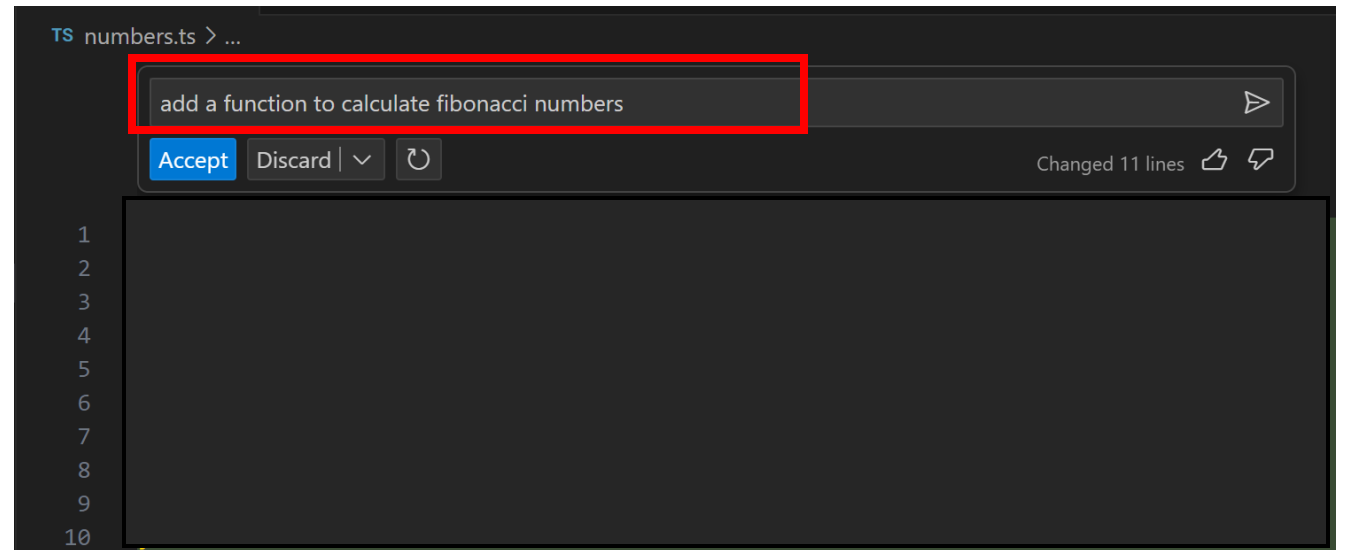
GitHub Copilot



GitHub
Copilot

Code Generation

Code Completion



TS numbers.ts > ...

add a function to calculate fibonacci numbers

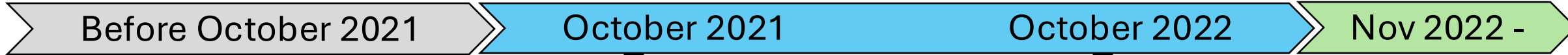
Accept Discard | v ↻

Changed 11 lines

1
2
3
4
5
6
7
8
9
10

```
def connect_to_endpoint(url):  
    response = requests.get(url, auth=bearer_oauth, stream=True)  
    print(response.status_code)  
    for response_line in response.iter_lines():  
        if response_line:  
            print(response_line)
```

Exploiting the LLM Roll-Out



LLM Support

```
def connect_to_endpoint(url):  
    response = requests.get(url, auth=bearer_oauth, stream=True)  
    print(response.status_code)  
    for response_line in response.iter_lines():  
        if response_line:  
            print(response_line)
```



No LLM Support



Research Questions

RQ1: How do LLMs affect the **volume** of open-source innovation?

RQ2: How do LLMs affect the **type** of innovation in open-source?

Extrapolative Reasoning

Interpolative Reasoning

Extension: What value do our results have if *LLMs* gets better?
(assumption: Better LLMs → enhanced utilization of context)

Data Construction

- List of 2k python packages, 11.5k R packages
- Obtain package update data (from PyPI and CRAN)
- Obtain over a **million** commits (from GitHub)

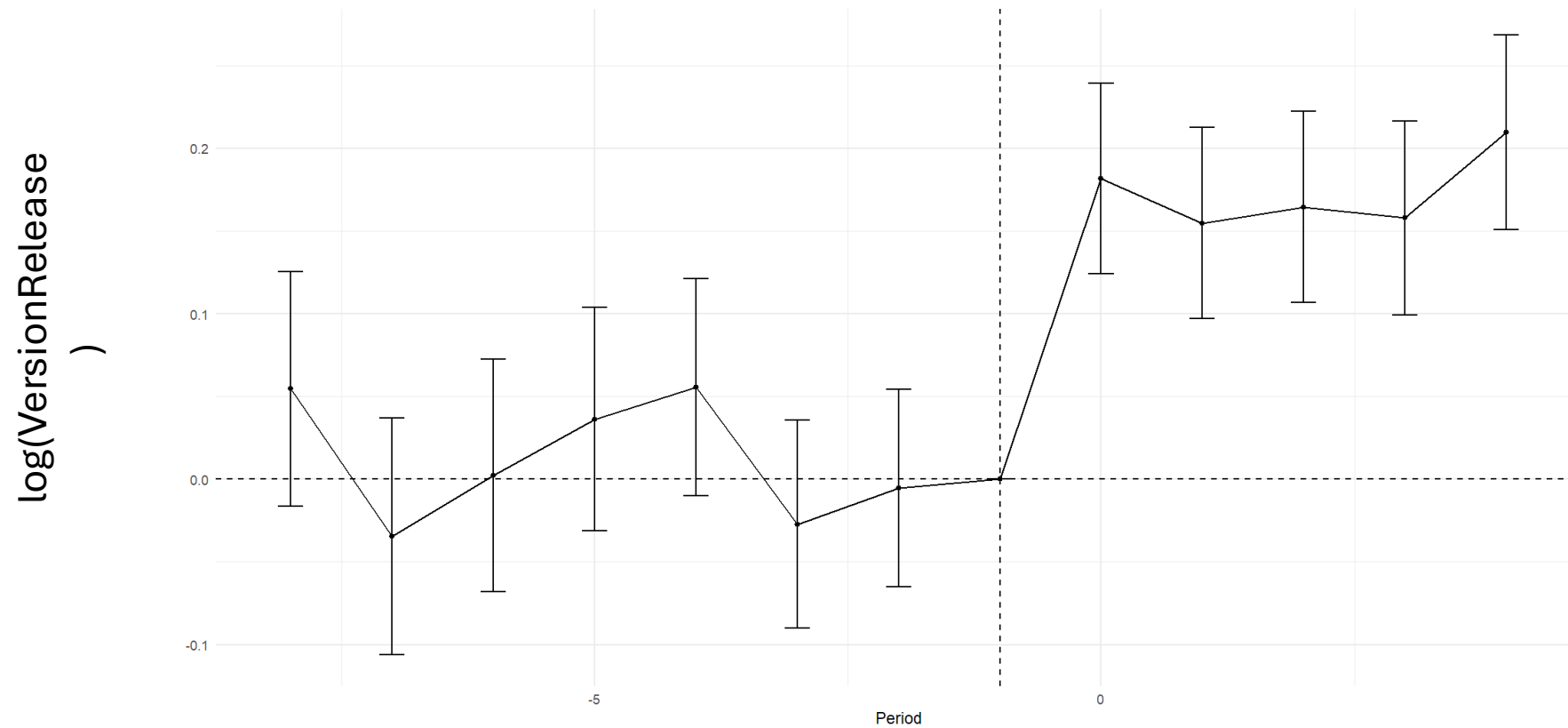
Propensity Score Matching (nn-match, no replacement)

Quarterly snapshots of 1610 Python, 1610 R packages

Commits Data Available: 1089 Python, 1077 R packages

Parallel Trends

Relative Time Model (Autor 2003)



Event Study Plot for treated and control packages.
Y-axis is the logged value of the version released.

Type of Contribution

Challenge: No Standardized Categorization of Commits

Solution: Commits come with text comments

- *Added ability to extract span attributes from falcon request objects. (#1158)*
- *Run one test on travis with --develop to avoid regressions*
- *Fix wrong file driver version error*

Type of Contribution

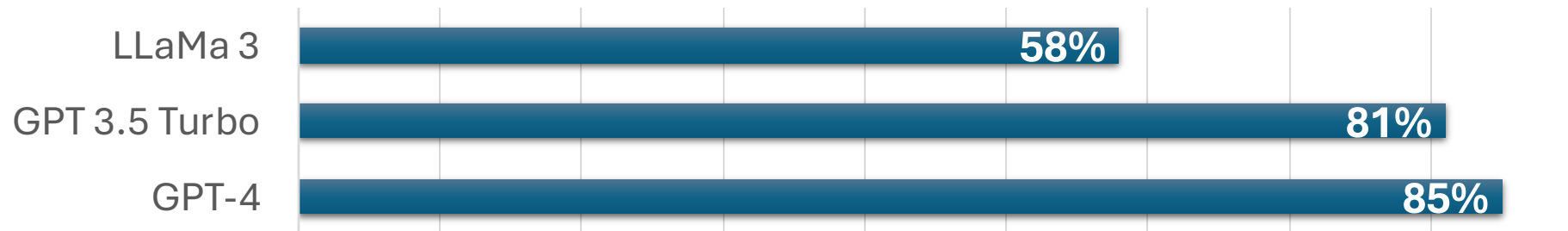
Step 1: Determine the General Categories

Open classification of 500 Python and 500 R comments using GPT 4

RA combined them into similar categories. Total 5

Verify if GPT 4 correctly reclassifies 1000 comments into the 5 categories.

Step 2: Benchmark the best LLM



Step 3: Use GPT-3.5 Turbo to categorize million commits

Code Development

Maintenance

Testing and QA

Documentation

Others

Details about the categorization is in our paper



RQ1: Volume of Open-Source Innovation

Identification: Propensity Score Matching with Difference-in-Differences

Dataset: PSM matched Dataset

$$\log(y_{it} + 1) = \alpha_i + \gamma_t + \beta X \text{ PythonPackage}_i X \text{ AfterAdoption}_t + \epsilon_{it}$$

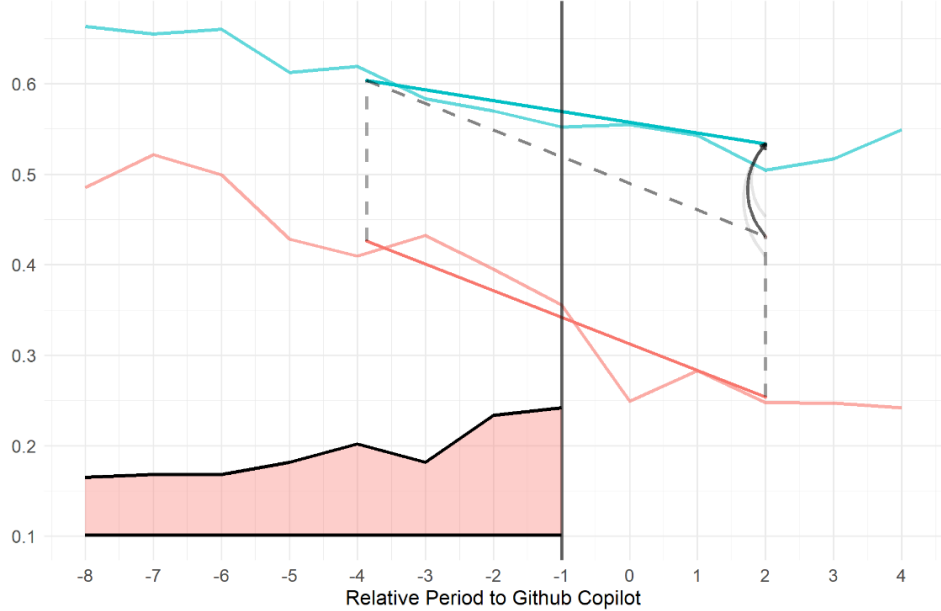
Dependent Variables:

Count of the new version of the package in a quarter

Count of commits committed in a package in a quarter

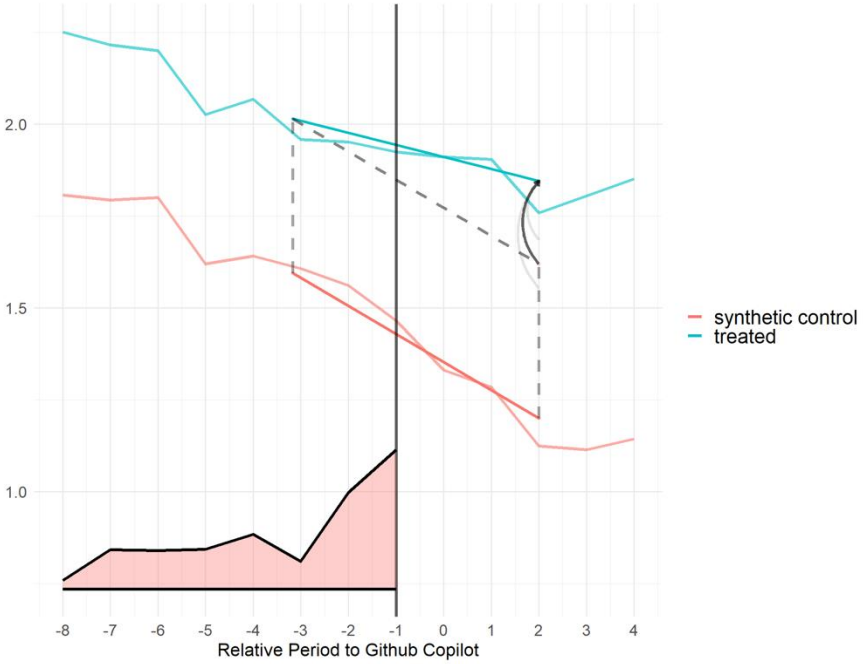
Alternative Identification: Synthetic Difference in Differences

Results: Volume of Open-Source Innovation



Package Update: 9% increase

synthetic control
treated



Commits: 33.1% increase

Findings So Far

RQ1: How do LLMs affect the **volume** of open-source innovation?

- Significantly enhances commits and package releases

RQ2: How do LLMs affect the **type** of innovation in open-source?

Origination Tasks

Iterative Tasks

RQ2: “Nature” of the Open-source Innovation

Code Development

- Feature Development
- Code Optimization
- Implementing Algorithms

Vague Scope and
Solution Space

Out-of-the-Box
thinking Preferred

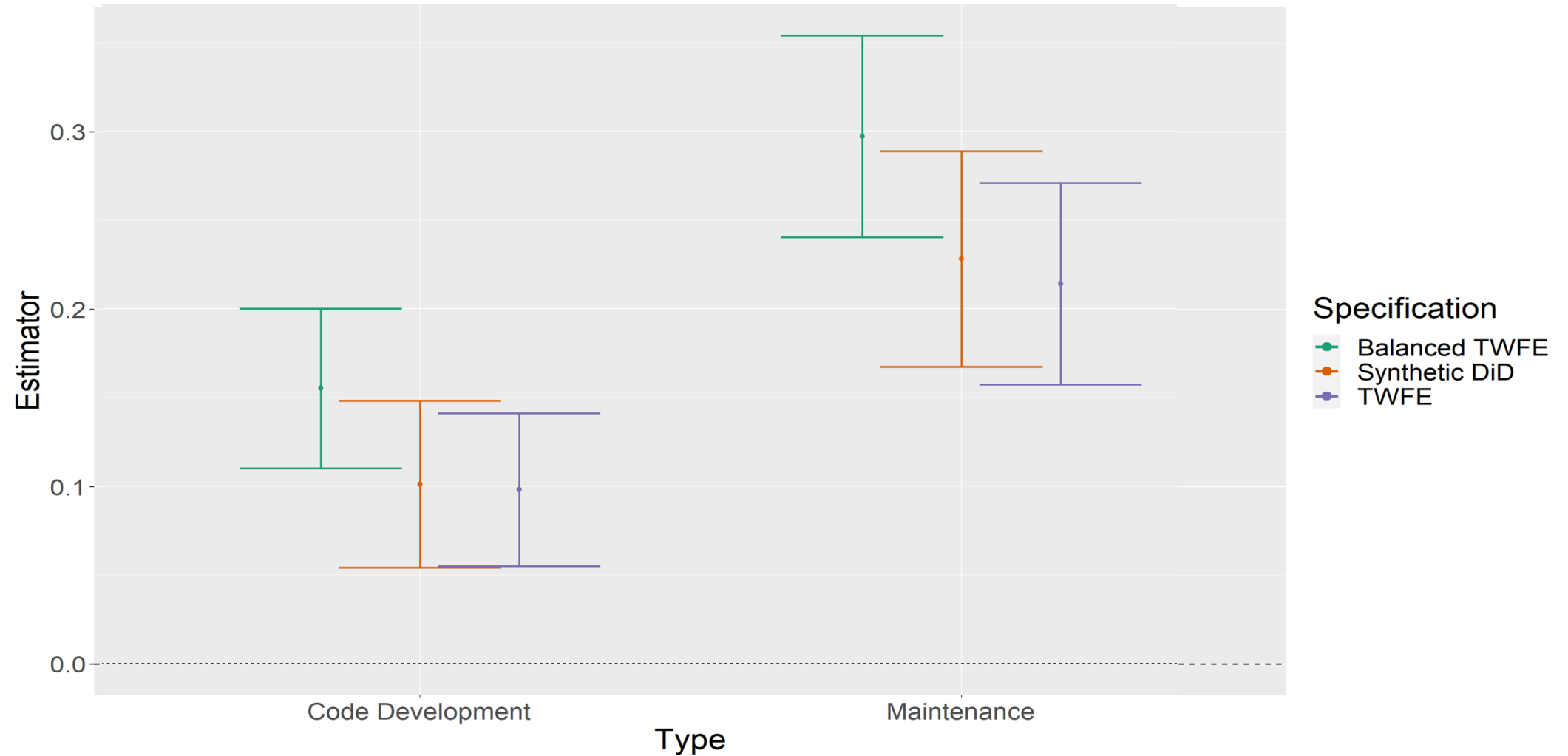
Maintenance

- Bug Fixes
- Code Cleanup
- Dependency Update

Well-defined Scope
and Solution Space

Within-the-Context
thinking Preferred

Results: Type of Open-source Innovation



Code development increases by 13%

Maintenance increases by 24%

Findings So Far

RQ1: How do LLMs affect the **volume** of open-source innovation?

- Significantly enhances commits and package releases

RQ2: How do LLMs affect the **type** of innovation in open-source?

- Significantly higher maintenance commits
- LLMs understand others' work → Benefit *Iterative Tasks*

Extension: What value do our results have if *LLMs* gets better?
(assumption: Better LLMs → enhanced utilization of context)

Study differential Impact on Projects based on their “activities”

RQ3: Distribution of Innovation across Packages

Model: Modify TWFE to include a *DDD* specification

Dataset: PSM matched Dataset

$$\log(y_{it} + 1) = \alpha_i + \gamma_t + \beta X \text{ PythonPackage}_i X \text{ AfterAdoption}_t X \text{ Above_Median}_i + \epsilon_{it}$$

Dependent Variables:

Count of the new version of the package in a quarter

Count of commits committed in a package in a quarter

Results: Distribution of Innovation

	Above Median Packages		Below Median Packages	
Dependent variable – $\log(\text{NewCommits}_{it}+1)$	(1) TWFE	(2) Balanced TWFE	(3) TWFE	(4) Balanced TWFE
$\text{PythonPackage}_i X \text{ afterAdoption}_t$	0.307*** (0.039)	0.248*** (0.042)	0.28 (0.017)	0.022 (0.019)
$\text{PythonPackage}_i X \text{ afterAdoption}_i$	0.199*** (0.031)	0.164*** (0.035)	0.08*** (0.020)	0.095*** (0.022)
Time Fixed Effect	YES	YES	YES	YES
Package Fixed Effect	YES	YES	YES	YES
# of Observations	26072	21,424	26,784	23,478

Code development commits are **higher**, but
Maintenance commits are **even higher** for Popular packages

Summary

The key take-away

LLMs boost outcomes for tasks requiring *inside-the-box* (*interpolative*) solutions, rather than *out-of-the-box* (*extrapolative*) solutions. This difference is likely to stay even if LLMs become better over time.

Thank You

raveesh@stern.nyu.edu

www.raveeshmayya.com

