

Statistics for jgehrcke/gipc

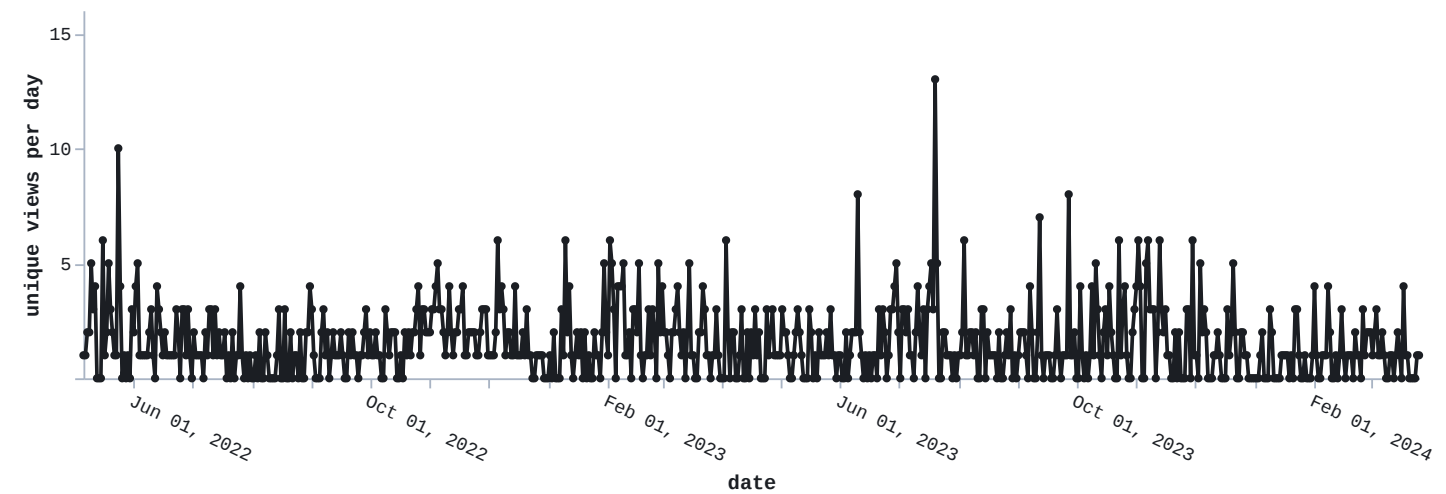
Generated for [jgehrcke/gipc](#) with [jgehrcke/github-repo-stats](#) at 2024-03-26 23:09 UTC.

Table of contents:

- [Views](#)
- [Clones](#)
- [Stargazers](#)
- [Forks](#)
- [Top referrers and paths](#)

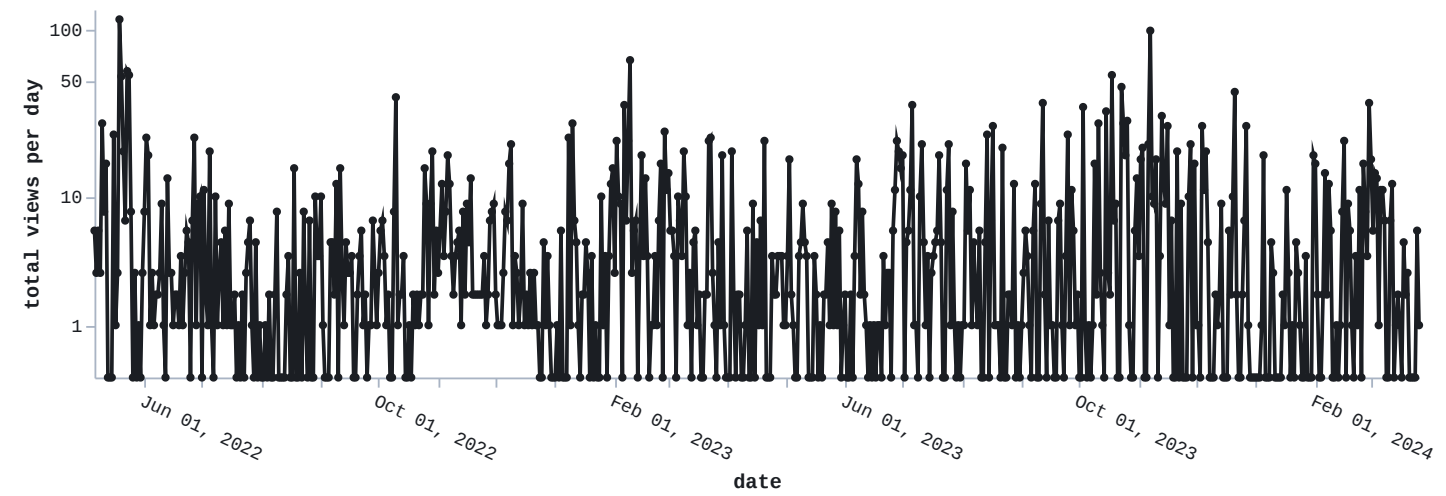
Views

Unique visitors



Cumulative: 1073

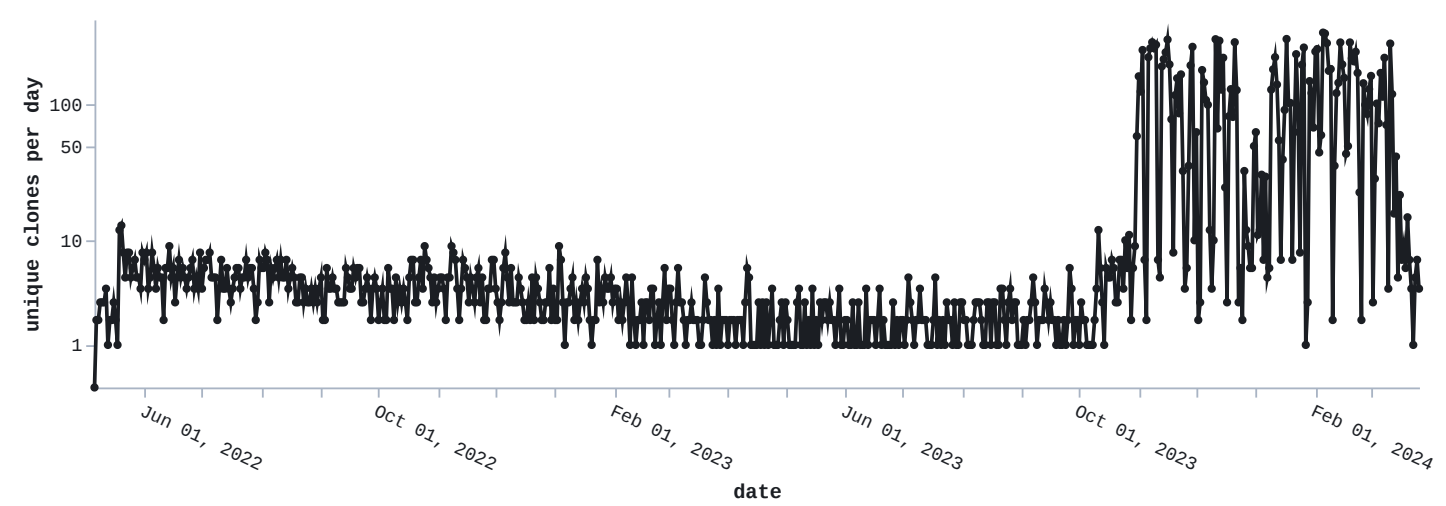
Total views



Cumulative: 3981

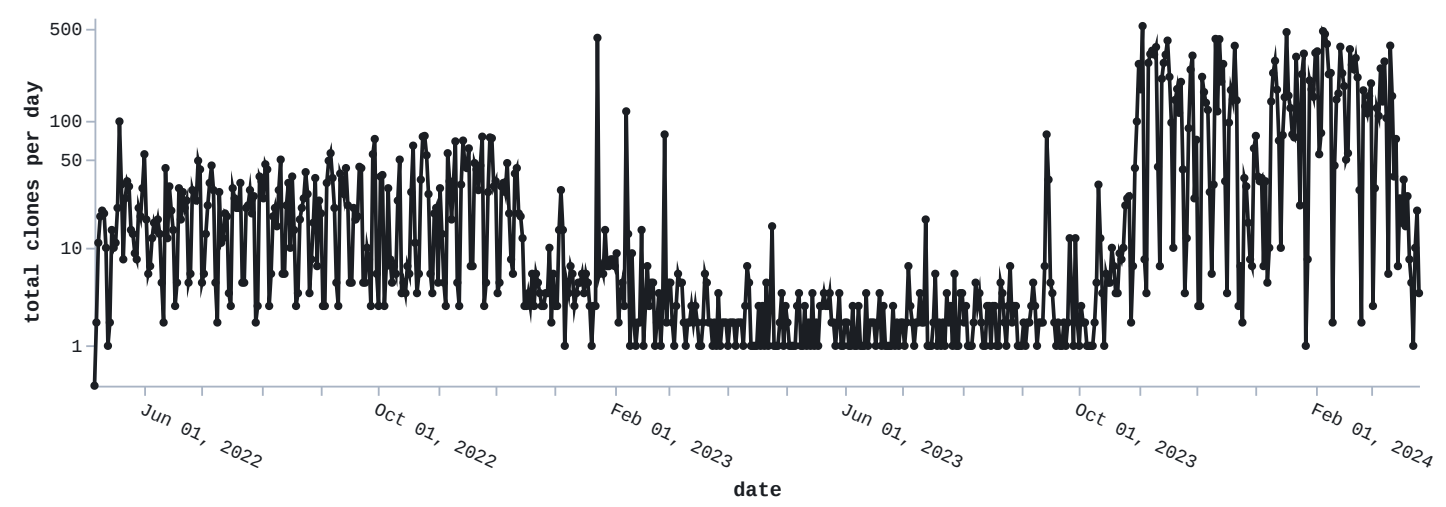
Clones

Unique cloners



Cumulative: 17025

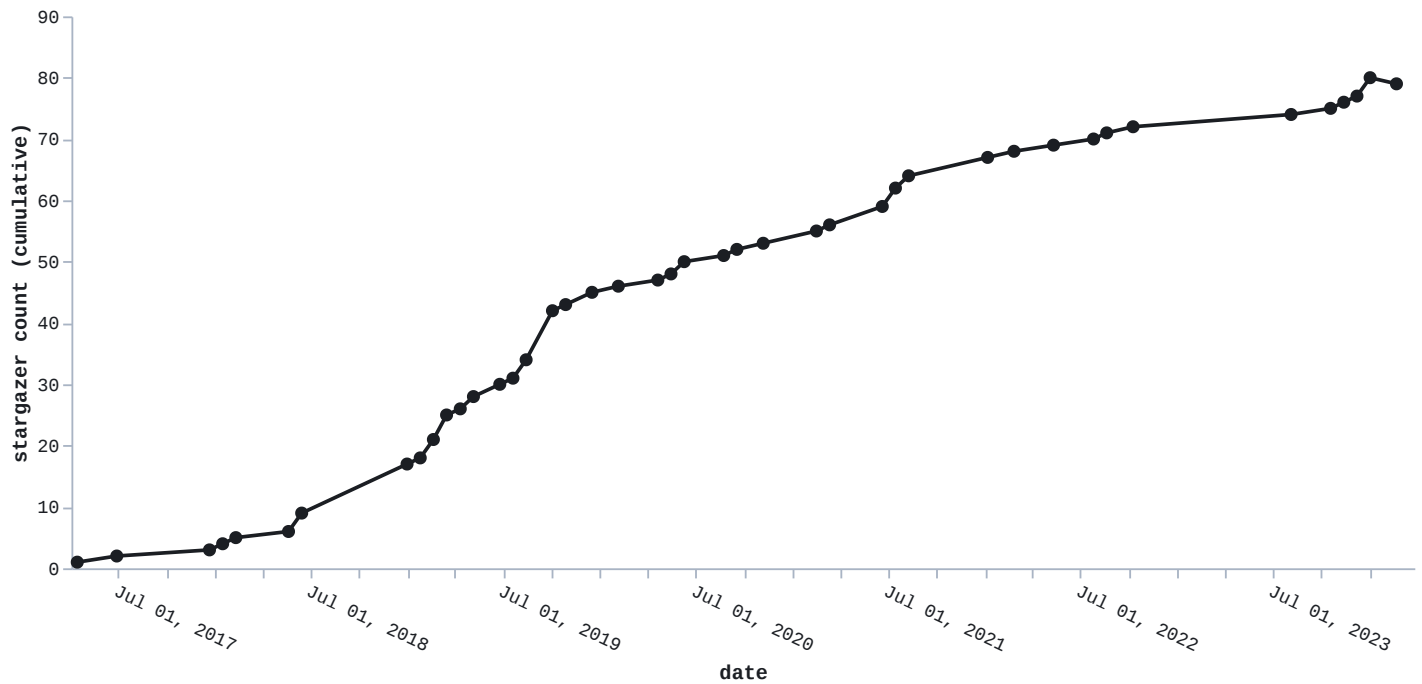
Total clones



Cumulative: 27363

Stargazers

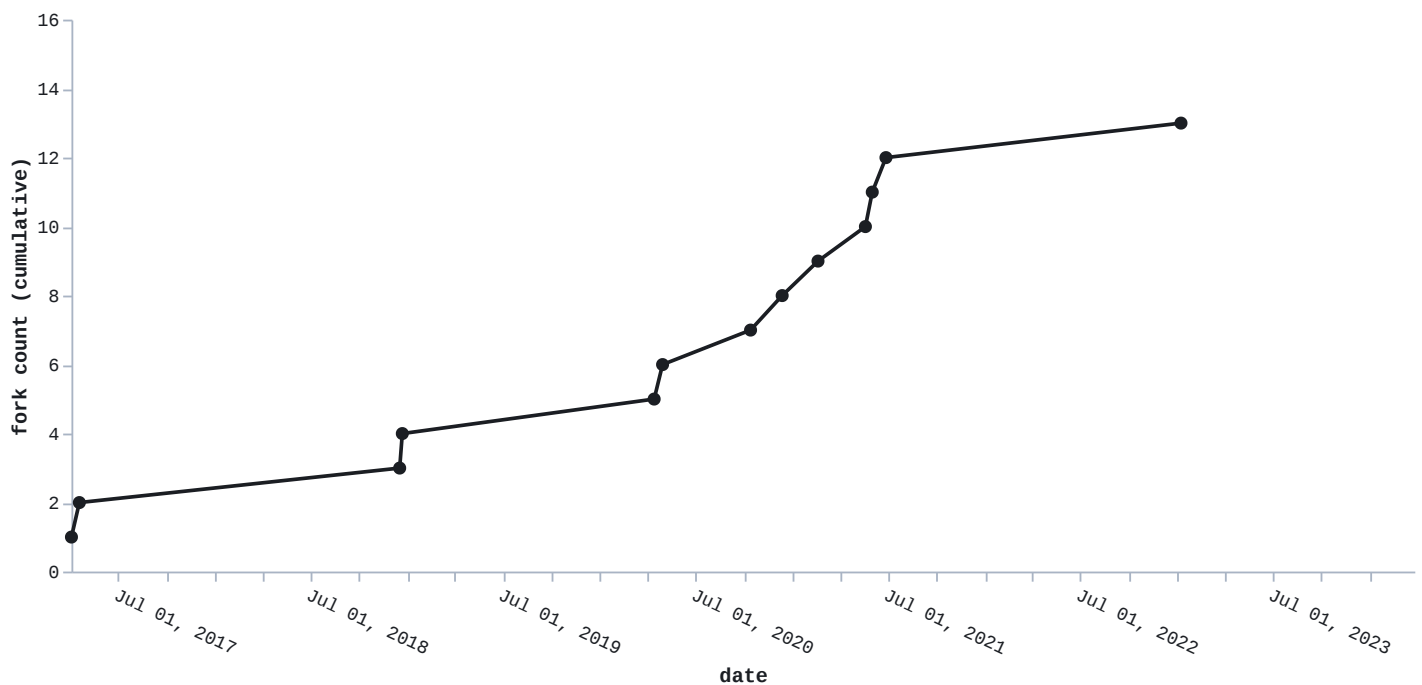
Each data point corresponds to at least one stargazer event. The time resolution is one day.



Note: this plot shows a larger time frame than the view/clone plots above because the star/fork data contains earlier samples.

Forks

Each data point corresponds to at least one fork event. The time resolution is one day.

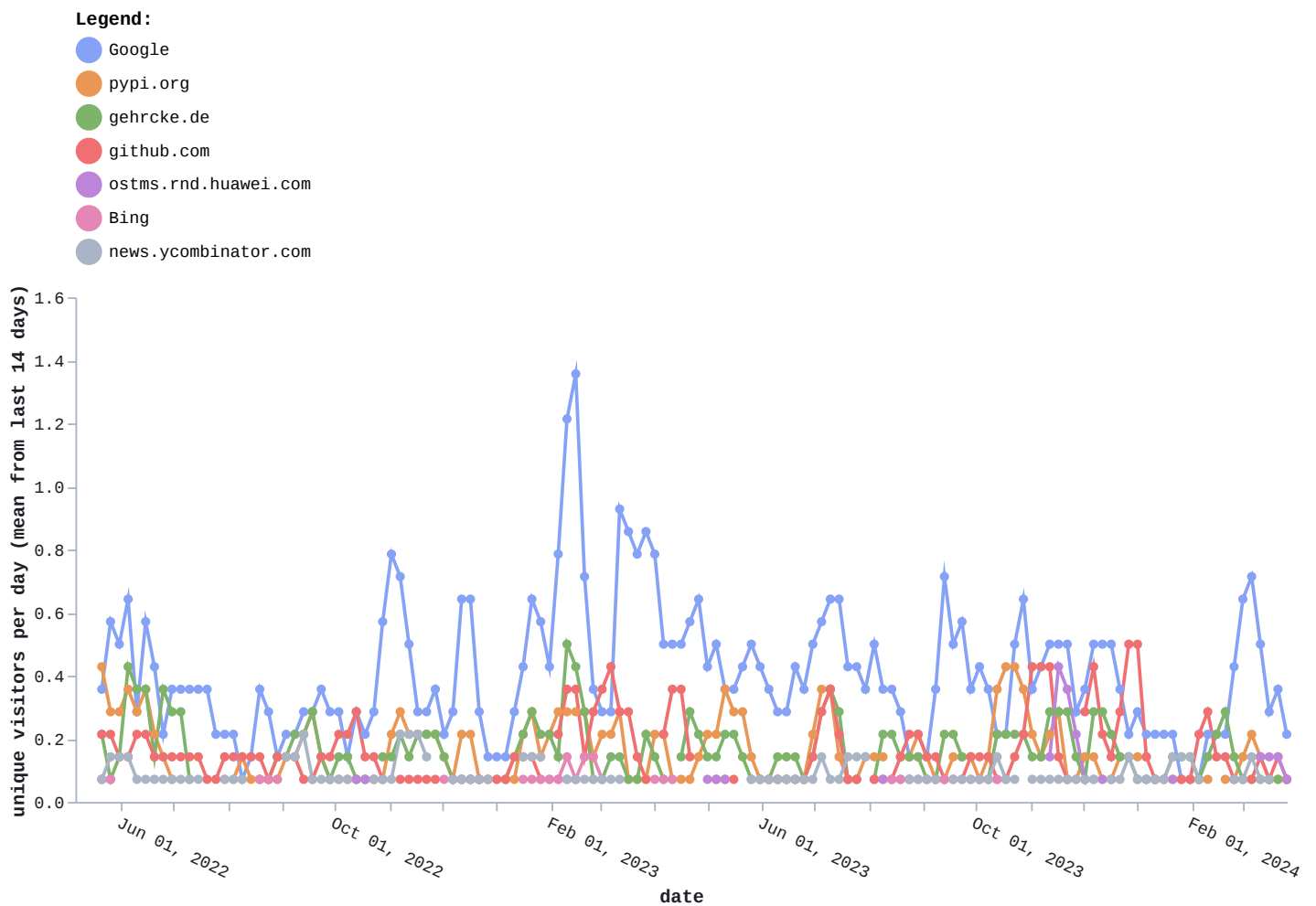


Note: this plot shows a larger time frame than the view/clone plots above because the star/fork data contains earlier samples.

Top referrers and paths

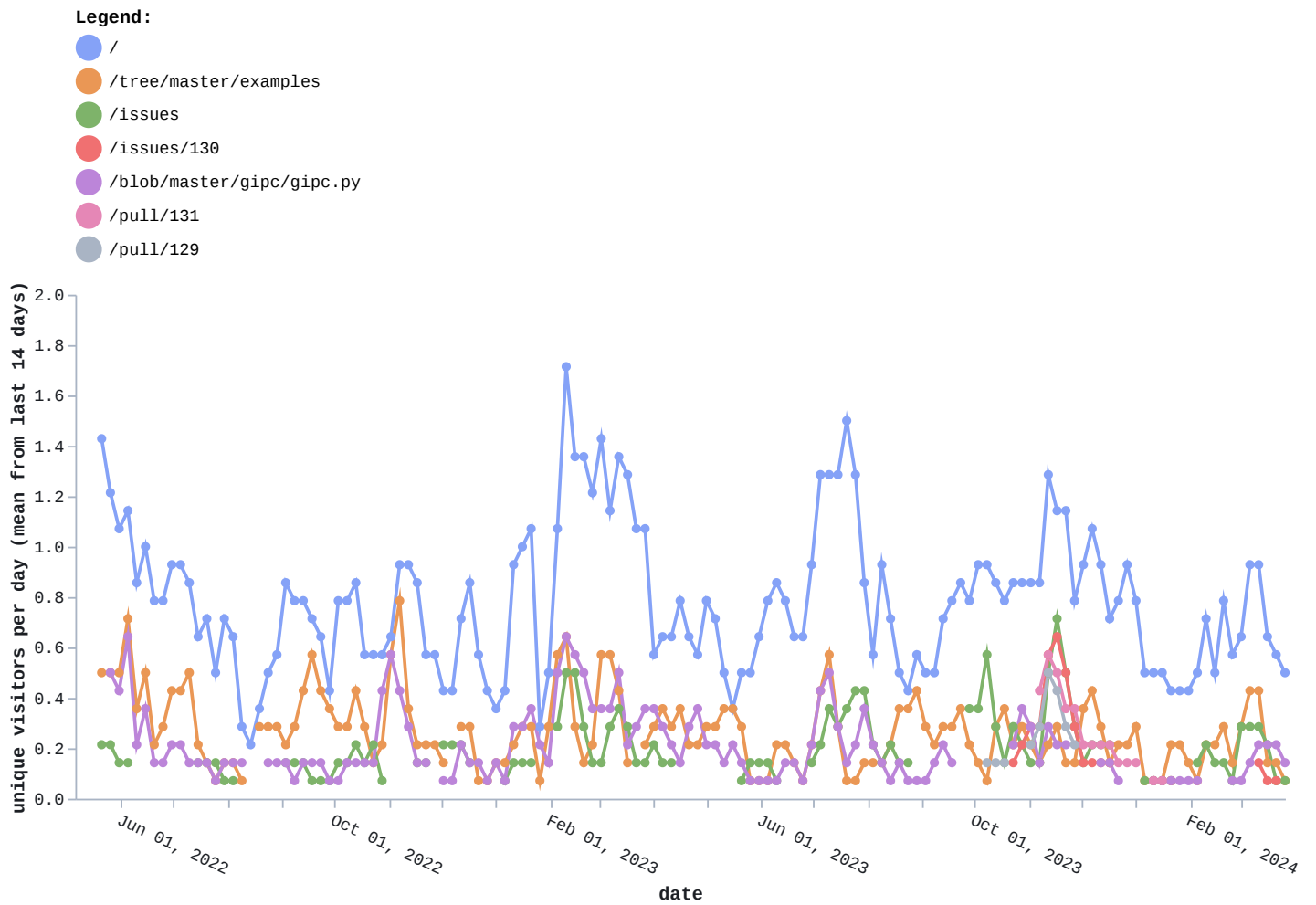
Note: Each data point in the plots shown below is influenced by the 14 days leading up to it. Each data point is the arithmetic mean of the “unique visitors per day” metric, built from a time window of 14 days width, and plotted at the right edge of that very time window. That is, these plots respond slowly to change (narrow peaks are smoothed out).

Top referrers



Top 15 referrers: 01: Google , 02: pypi.org , 03: gehrcke.de , 04: github.com , 05: ostms.rnd.huawei.com , 06: Bing , 07: news.ycombinator.com , 08: DuckDuckGo , 09: codingdict.com , 10: cn.bing.com , 11: groups.google.com , 12: qwant.com , 13: t.co , 14: cnpython.com , 15: Baidu

Top paths



Top 15 paths: 01: / , 02: /tree/master/examples , 03: /issues , 04: /issues/130 , 05: /blob/master/gipc/gipc.py , 06: /pull/131 , 07: /pull/129 , 08: /issues/125 , 09: /blob/master/examples/serverclient.py , 10: /tree/master/gipc , 11: /pull/129/files , 12: /pulls , 13: /issues/8 , 14: /blob/master/examples/wsgimultiprocessing.py , 15: /blob/master/examples/gipc_benchmark.py