Github Analytics

Arun somasundaram

Objective

- Analyze Five public Git repositories
  - Opencv
  - Tensorflow
  - Pytorch
  - SpaCy
  - Ant-design
- Generate insights
  - Throughput
  - Bandwidth
  - Trends
  - Identify areas of weak or strong performance
- Automate the process
- Apply automation on additional repositories

Approach

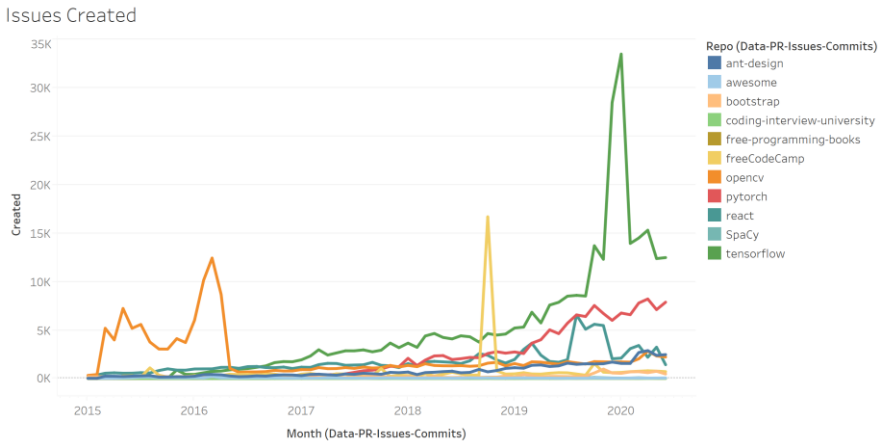
- Browse the target repos in Github.com
- Explore issues, pull_requests, Insights (commits, code frequency, contributors) features
- Explore search criteria on all the above
- Read and understand documentation on API search
- Determine the metrics to be analyzed
- Perform manual search on one repository for one measure and visualize
- Write Python script to generate the metrics for other repositories and other measures
- Use Tableau to visualize the metrics

Measures Analyzed

- Issues
  - Created Count
  - Closed Count
  - Throughput (Derived)
  - Turnaround time
- Pull Request
  - Created Count
  - Closed Count time
  - Turnaround
  - Linked to Issues
  - Comments(interactions) Count
- Commit
  - Count
  - Merge:true or false
- Contributions by individuals
  - Weekly Additions
  - Weekly Deletes
  - Weekly Commits
- Code
  - Additions
  - Deletions

Measure 1 : Issues Created

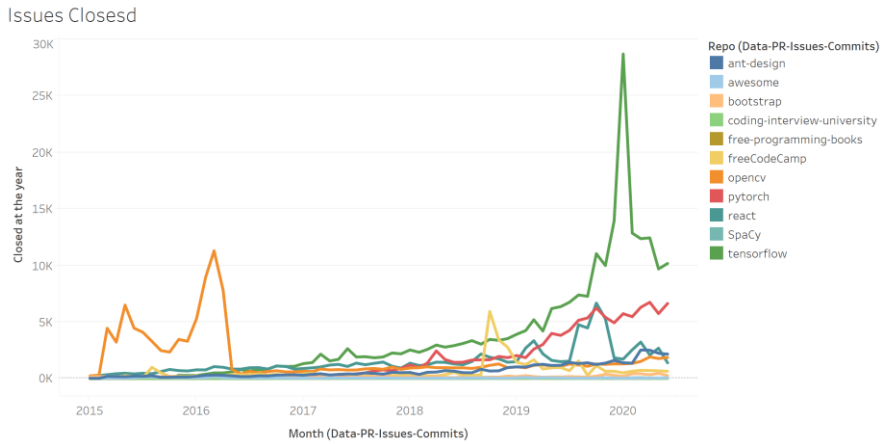
- Collected Monthly count of issues created since 2015
- Used in measuring the throughput
- Issues Growth can be used to calculate bandwidth.



- Repositories have varying degree of issues count however, the general trajectory is upward
- There are spikes in certain month for some repositories which should be inspected

Measure 2 : Issues Closed

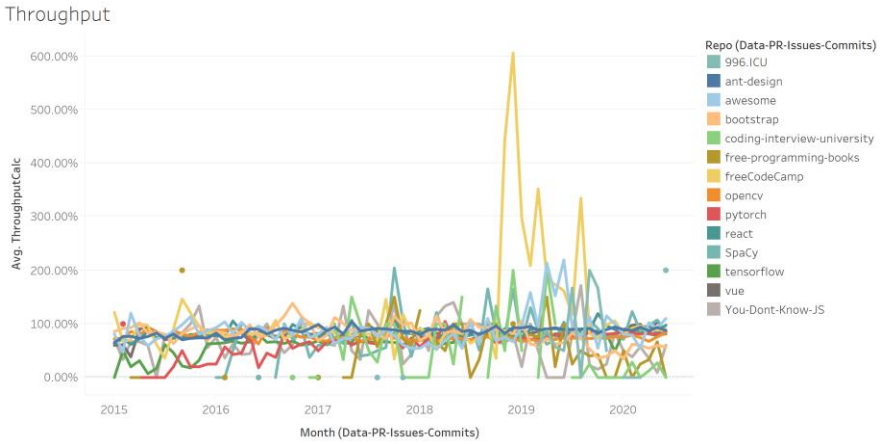
- Collected Monthly count of issues closed since 2015
- Used in measuring the throughput
- This can be used to calculate the bandwidth



- Observed that the different repositories have slightly varying degree of closure however the trajectory is upward.
- There are spikes in certain month for some repositories which is good but needs to be understood

Measure 3 : Throughput

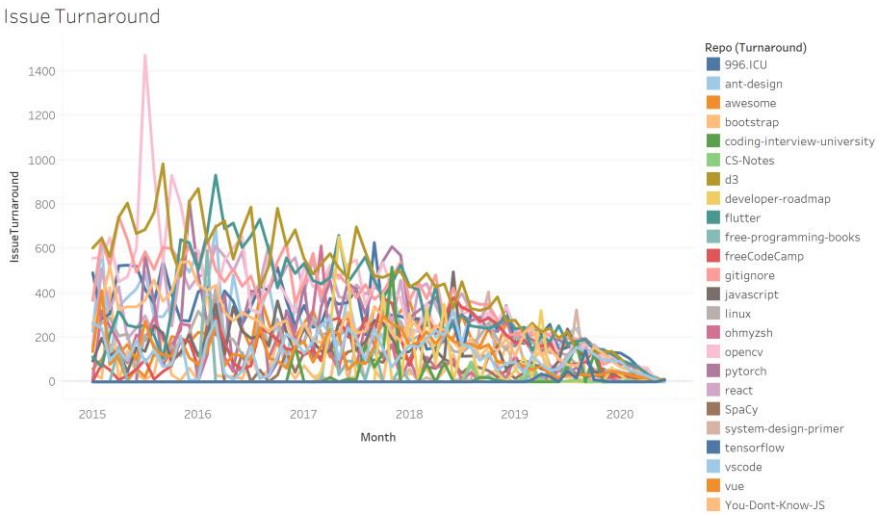
- This is calculated as a percentage of IssuesClosed over IssuesCreated montly
- Factors impacting the throughput are wide
  - Complexity
  - Bottleneck in Pull Request process
  - Team size
  - Responsiveness



Observed that the throughput is generally in 80% range.

Measure 4 : Issues Turnaround

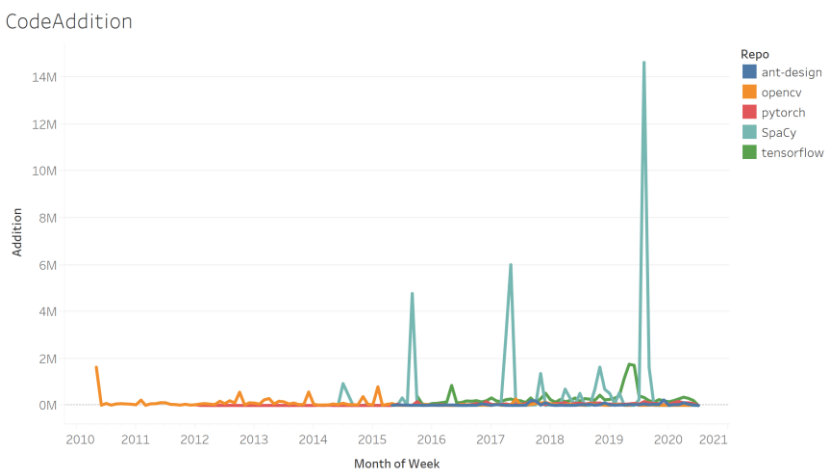
- This is calculated as a difference in days between closed_at and created_at timestamps



Observered that turnaround time in downward trajectory

Measure 5 : Code Addition / Deletion

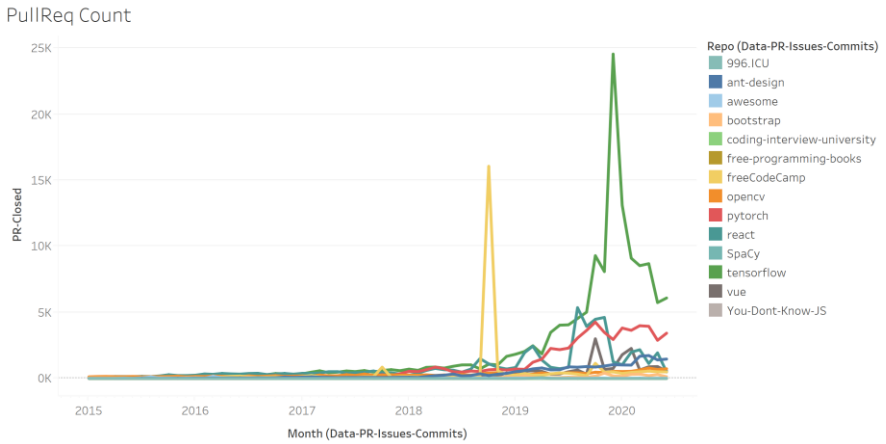
- Data collected weekly addition /deletion at the repo level as well as the contributor level
- High code addition can be tied to issues created later
- Individual participation can be measured
- Also computed
  - Code addition per contributor
  - LOC : Issue ratio



Observation : Some repos have high spikes in code addition in some months

Measure 6: Pull Request Count

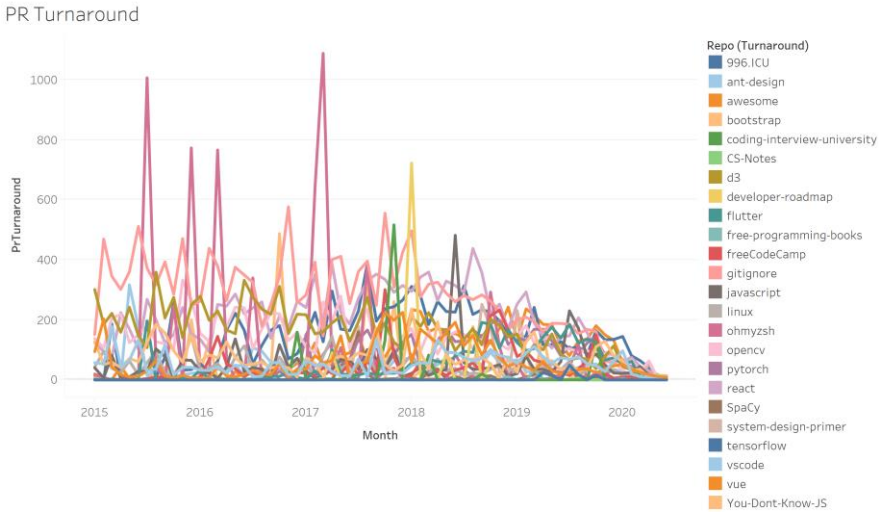
- Pull Requests are key to solving issues
- Data for Monthly count of pull requests created was collected
- Additional slice of data where pull requests were linked to issues was collected
  - This could indicate how disciplined the team was in handling the pull requests
- Additionally data for the pull requests that had comments was also collected
  - This could indicate the health of the change management
- Although not collected it is possible to compute the Turn around time for pull requests



- Observed that the different repositories have slightly varying degree of PR however the trajectory is upward.
- There are spikes in certain month for some repositories which is good but needs to be understood

Measure 7 : Pullrequest Turnaround

- This is calculated as a difference in days between closed_at and created_at timestamps

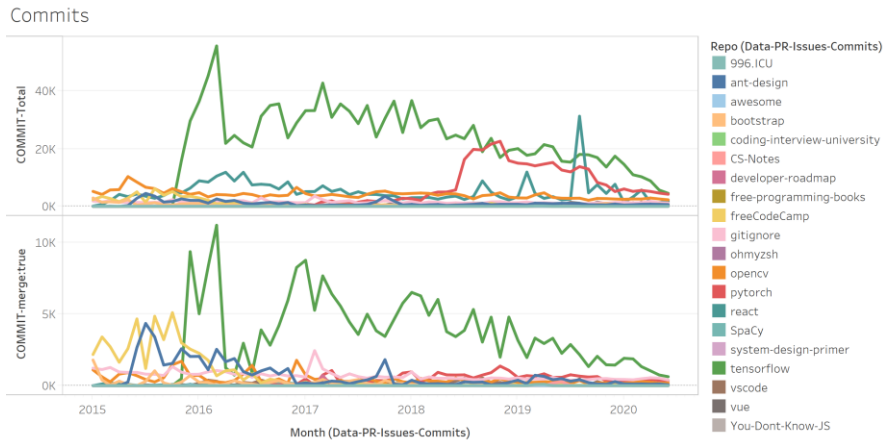


Observed that the turnaround time in downward trajectory

Measure 8: Commit count

- Data was collected for monthly count of commits to the repository
- Slice of data whether the commit was ultimately merged was also collected

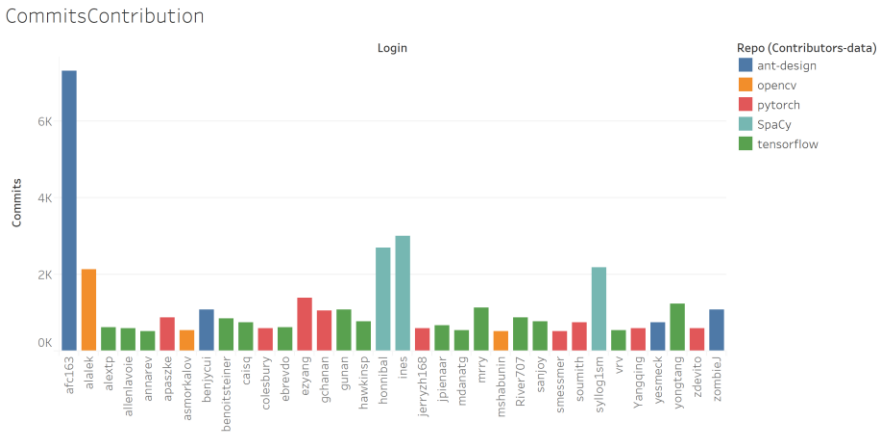
- Although not collected, it is possible to collect the turn around time and the commits linked to issues



- Observed that commits count and percentage of merged commits in all repositories are generally range bound

Measure 9 : Contributions

- Weekly contributions (commit count) by individuals were collected but rolled up.
- This can indicate the engagement of individuals.



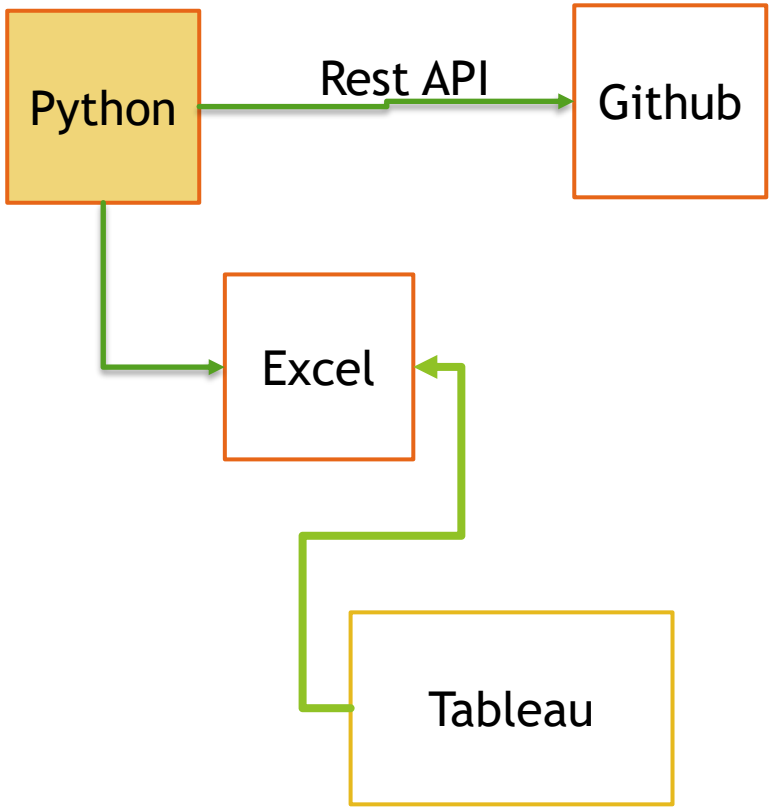
Some Repositories have one anchor contributor supported by strong contributors

Key Performance Measures Ranked

	Spacy	pyTorch	Tensorflow	Opencv	Ant-design
IssuesCreated(low=1)	1	3	5	4	2
Issues Closed(high=1)	5	3	1	2	4
Throughput (high=1)	2	5	4	3	1
Code Additions(low=1)	5	2	4	3	1
Code Deletions(low=1)	5	2	4	3	1
PRcomments >20(high=1)	5	2	1	3	4
PRcomments=0(low=1)	1	4	5	2	3
Merge Commit(high=1)	1	2	5	3	2
LOC to Issue (high=1)	4	1	5	2	3

Automation

https://github.com/asomasundaram/GitInsights



- Uses openpyxl, jsonpath_ng libraries
- The inputs are defined in Excel itself
- Functions are parameterized
- Takes Github developer security token as the argument
- Computes 66 months of 13 measures per repo
- Takes an ~hour to process a repo with the rate limit of 30 per minute

Automation (continued)

- Calls to the Github REST API has rate limit constraints. For unauthenticated users the search APIs are limited to 10 calls per minute, for authenticated users the rate limit is 30 per minute. For corporate accounts the quota is 5000-12000 per hour.
- The program makes a rate_limit call first before every search and waits until the next reset time if the quota is about to be exceeded. In the event the call fails it waits for a minute and retries.
- Ideal architecture will have a database where data is collected at detailed level and refreshed periodically and stored with a proper database schema
- For better user experience a web application is the best.

Additional Repositories automated

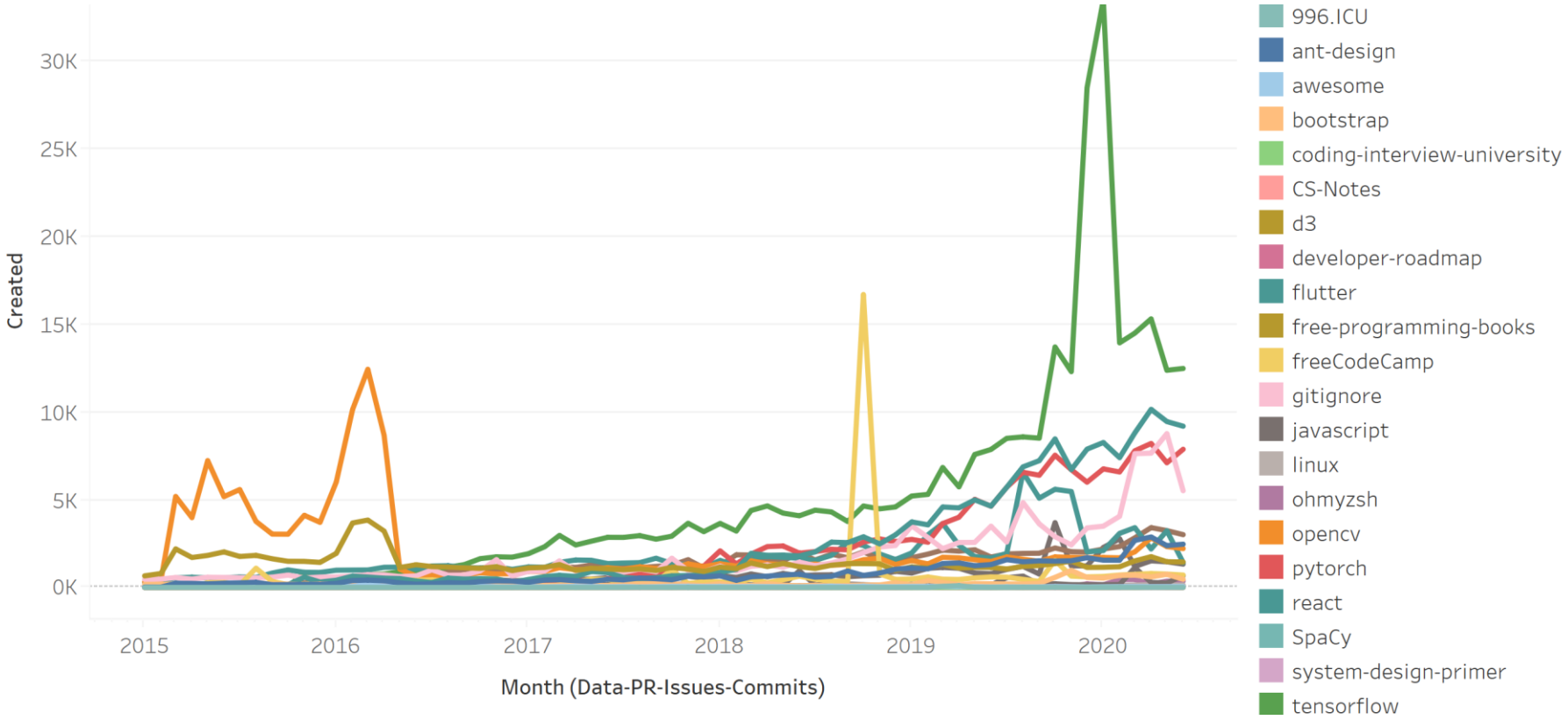
- freeCodeCamp/freeCodeCamp
- 996icu/996.ICU
- vuejs/vue
- EbookFoundation/free-programming-books
- facebook/react
- twbs/bootstrap
- sindresorhus/awesome
- getify/You-Dont-Know-JS
- jwasham/coding-interview-university
- kamranahmedse/developer-roadmap
- ohmyzsh/ohmyzsh
- CyC2018/CS-Notes
- github/gitignore
- donnemartin/system-design-primer
- microsoft/vscode
- airbnb/javascript
- flutter/flutter
- torvalds/linux
- d3/d3

Identified by querying for repos with > 90000 stars

Summary

- Individual measures have limited insights. Measures should be corelated with other measures. This requires involved effort by Data Scientists and Software Engineers.
- Analyzing repositories in automated fashion is a significant development and maintenance effort.
- Build vs Buy : Products are available for buy: Velocity, GitPrime are good examples

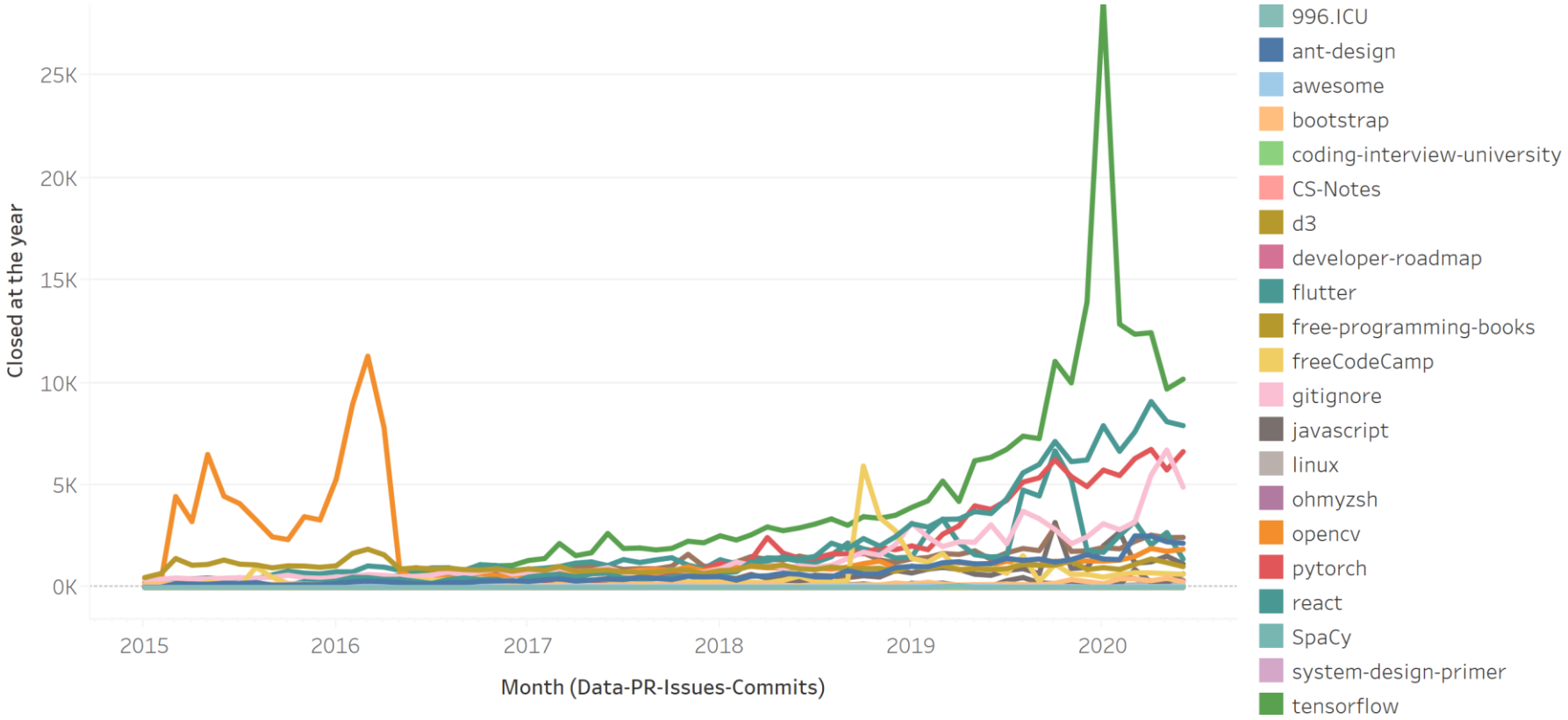
Appendix



Issues Created

Monthly volume of issues by the repositories

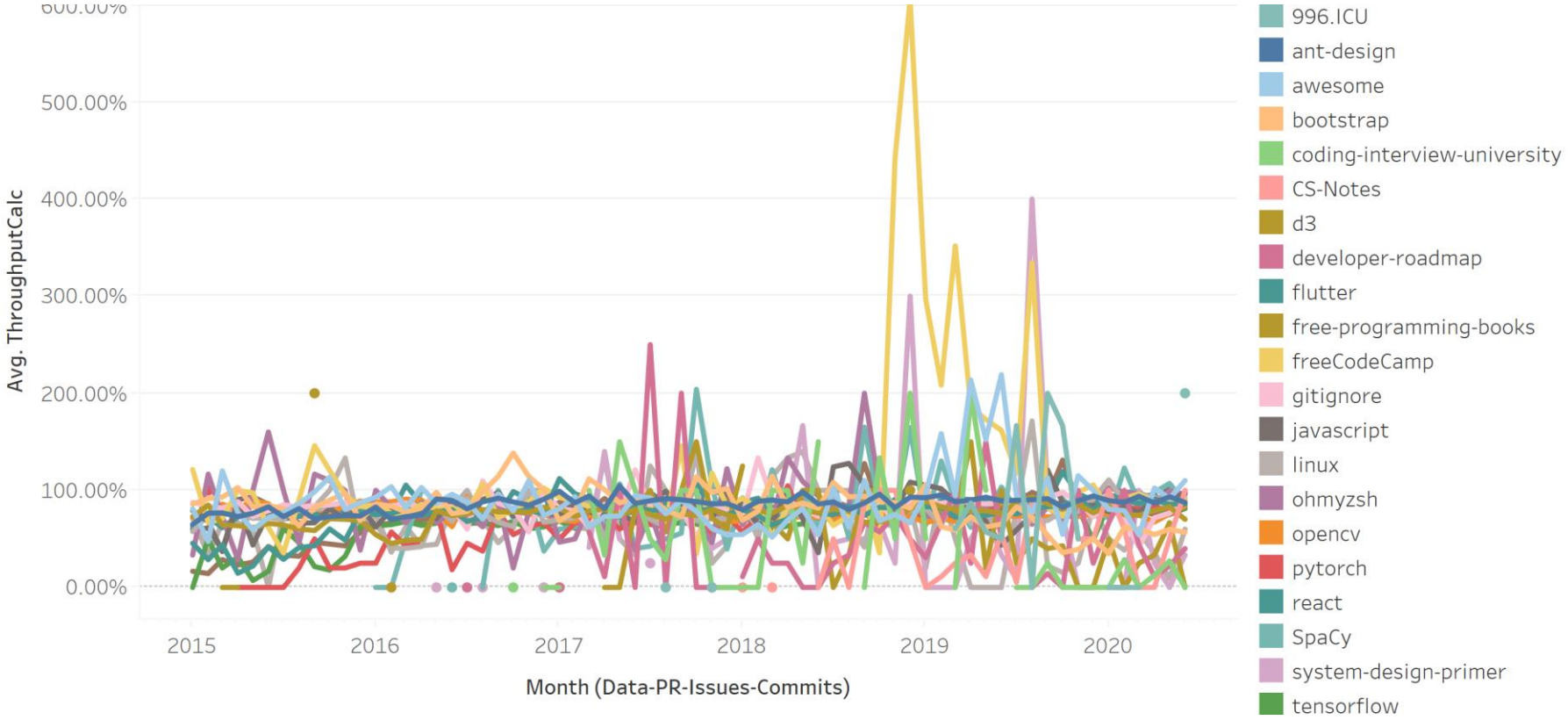
Arun Somasundaram



Issues Closed

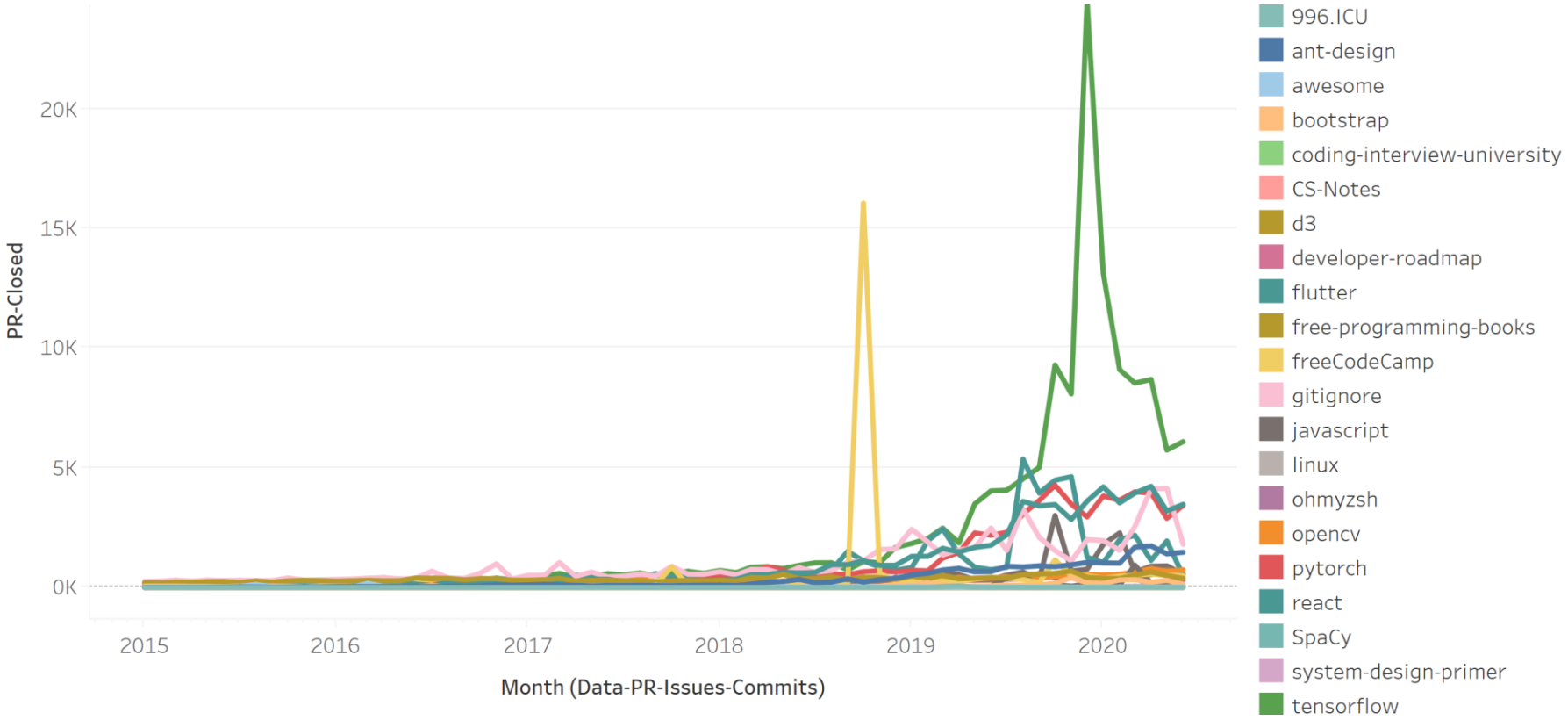
Monthly volume of issues closed by repositories.

Arun Somasundaram



Throughput

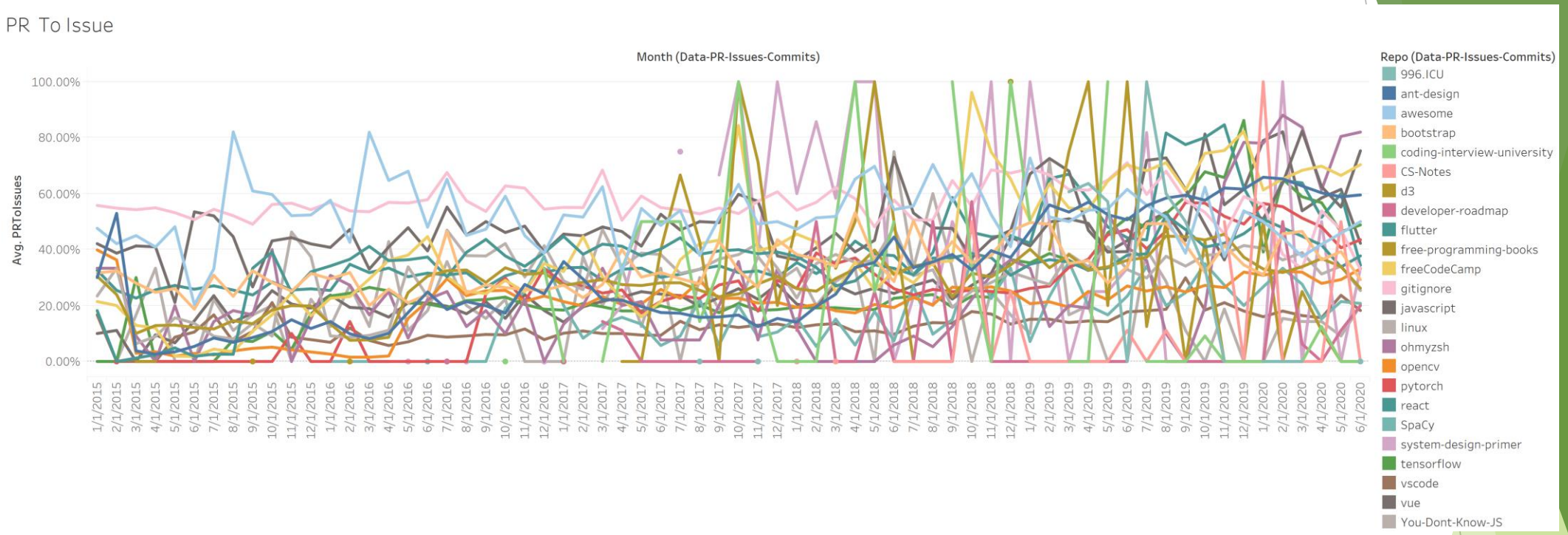
Percent of issues closed over issues created. Calculated monthly.



Pull Request Count

Monthly volume of pull requests by repositories

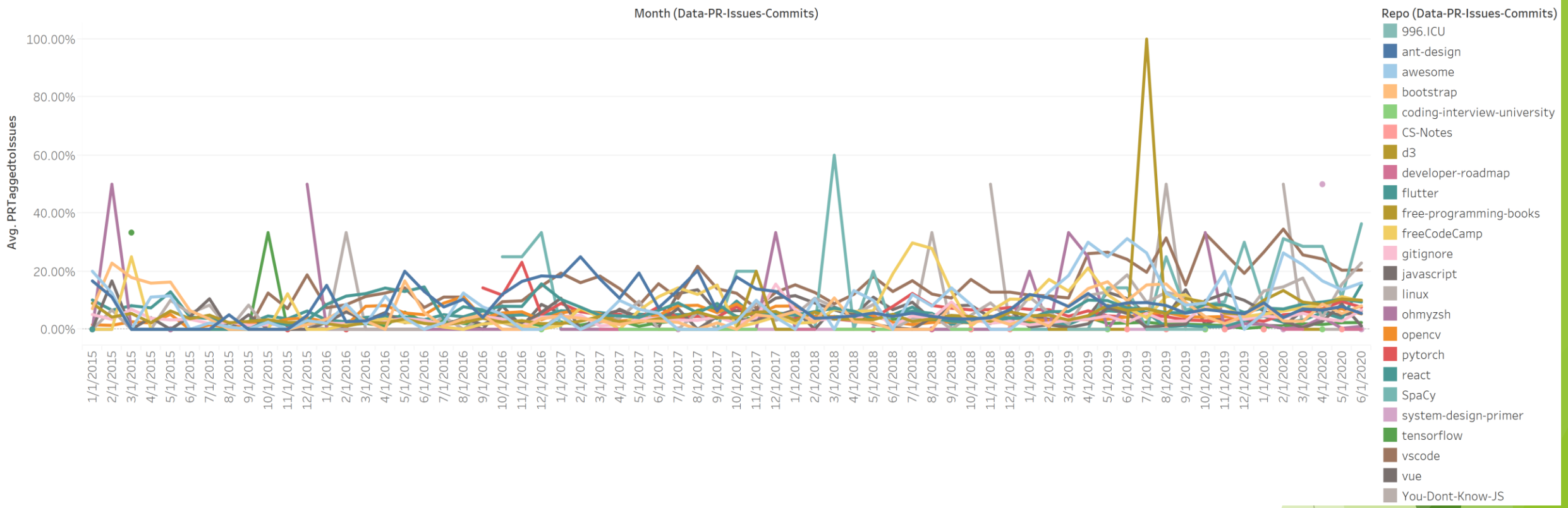
Arun Somasundaram



Percent of Pull Requests over Issues

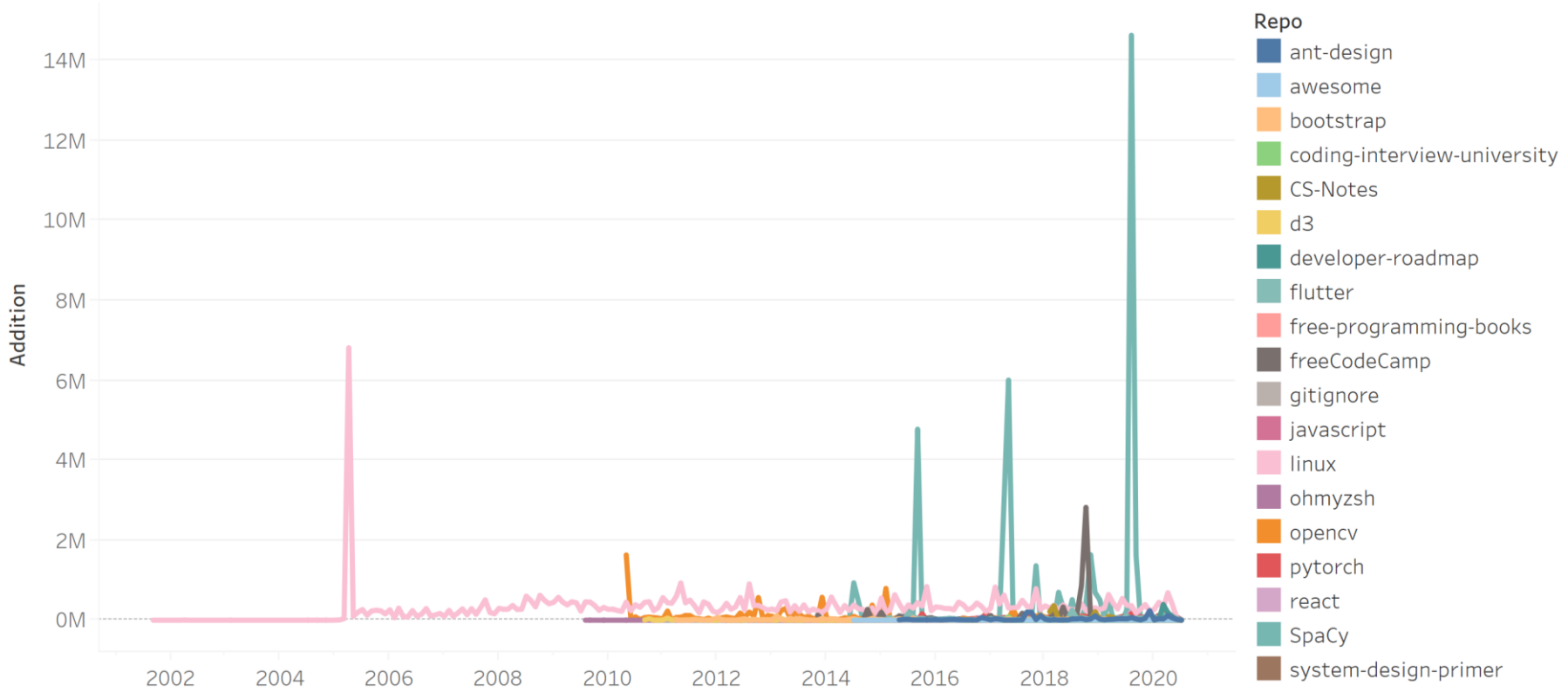
Arun Somasundaram

PR Tagged To Issues



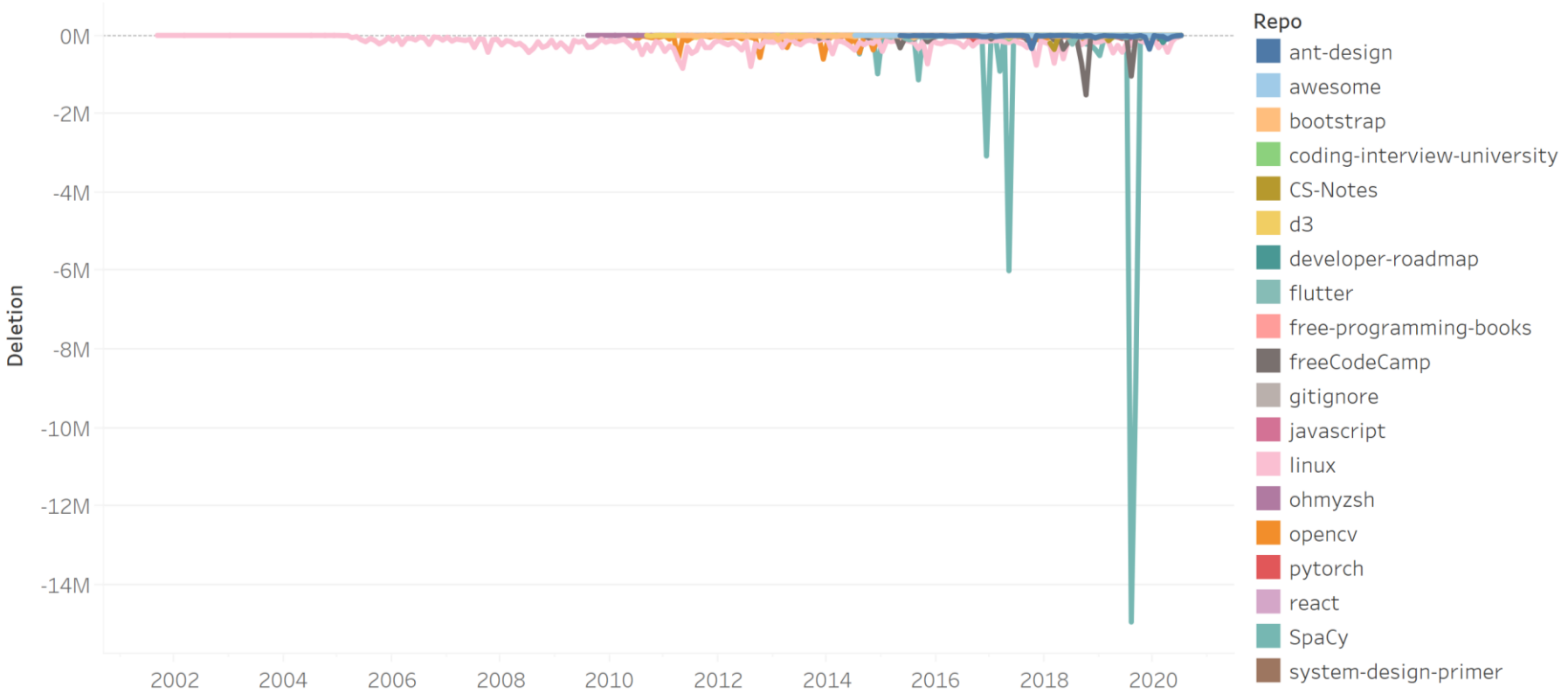
Percent of Pull Requests Tagged to Issues

Arun Somasundaram



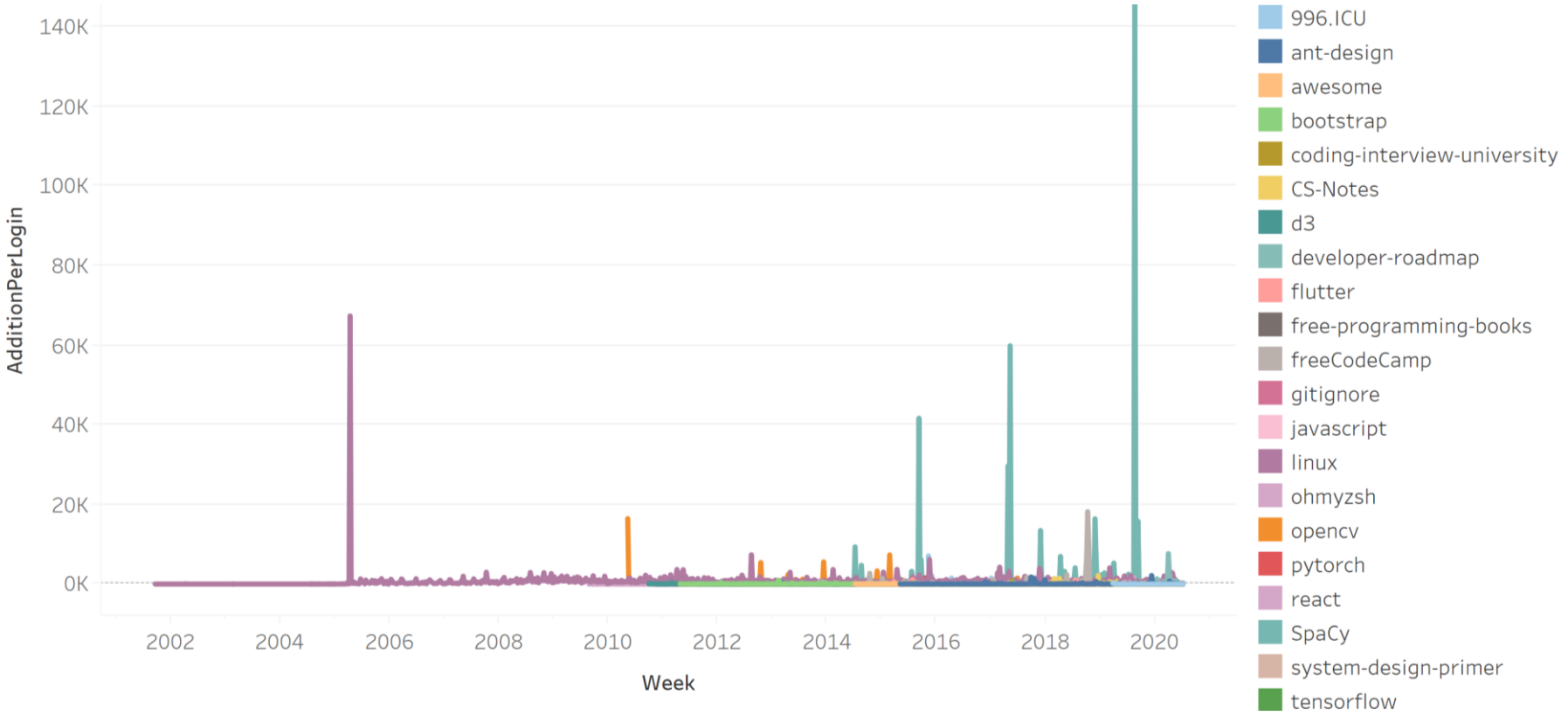
Monthly volume of code addition to repositories

Arun Somasundaram



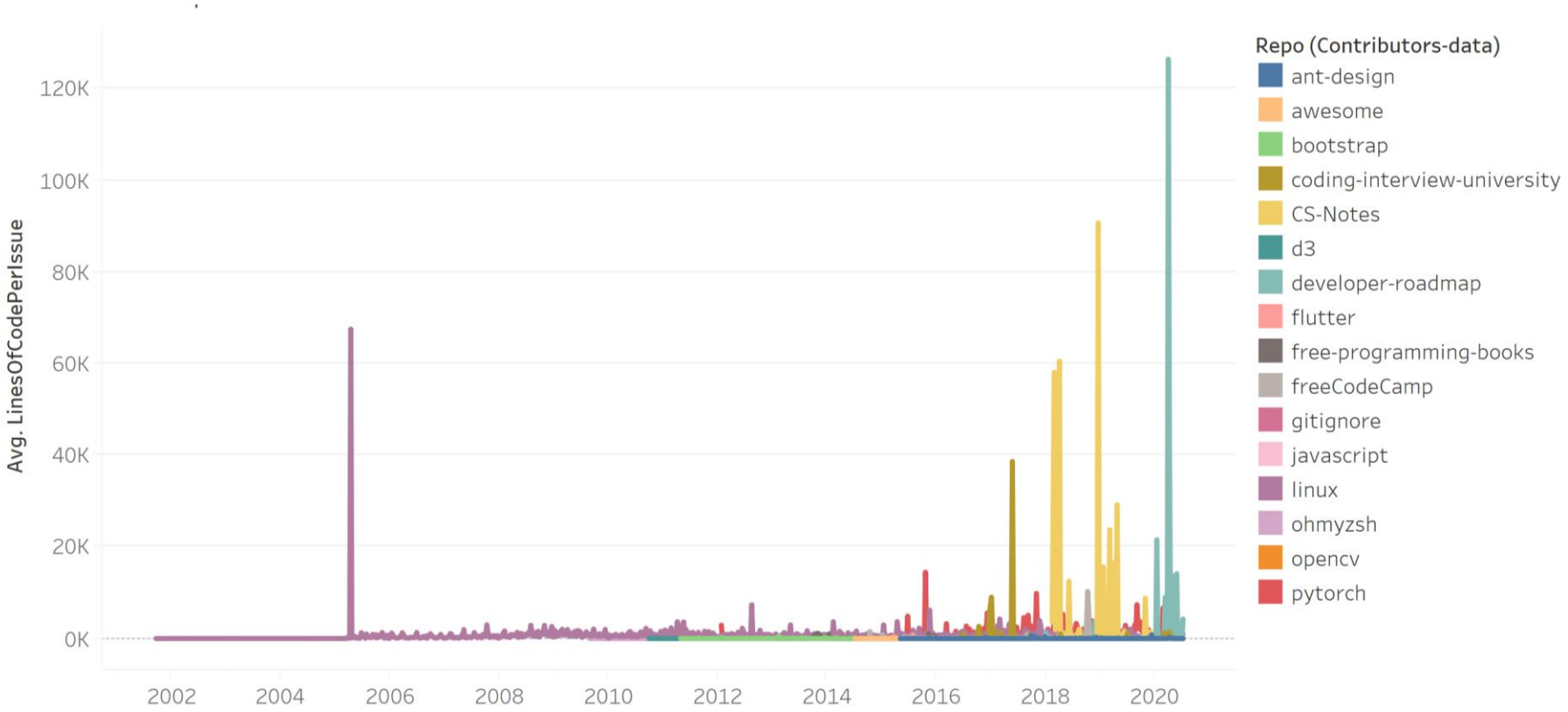
Monthly volume of code deletion to depositories

Arun Somasundaram



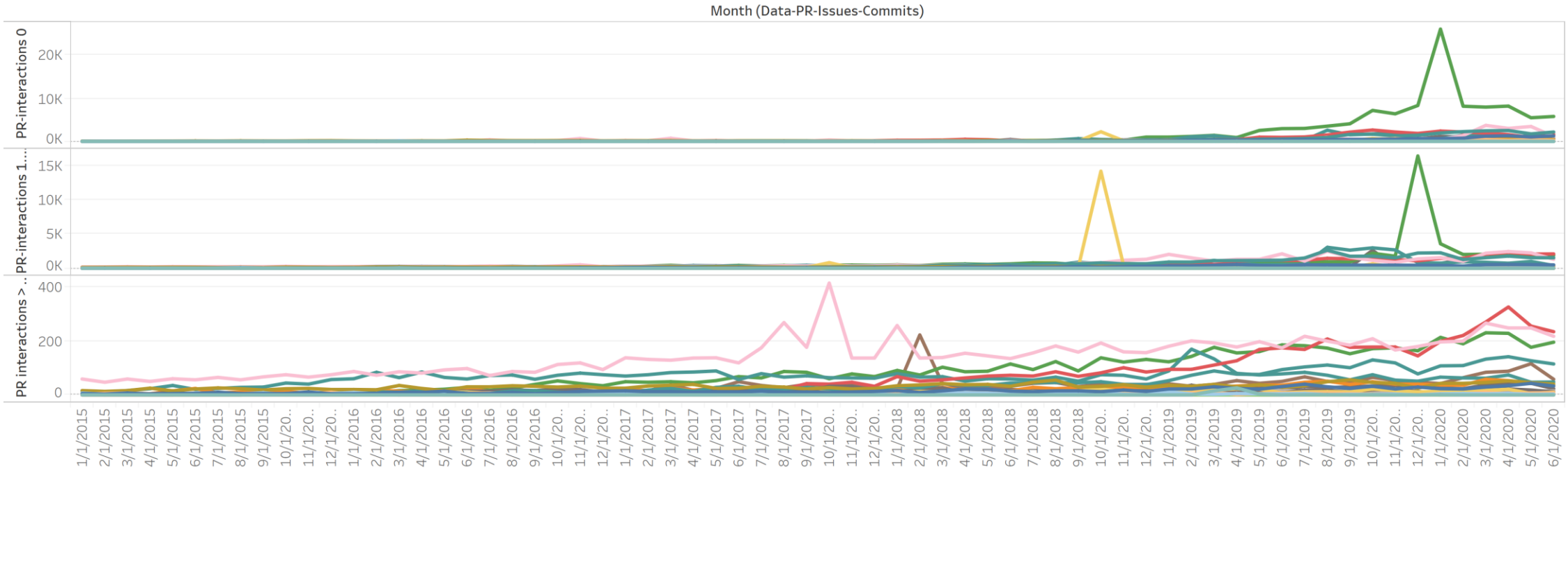
Average Lines of code addition by each contributor

Arun Somasundaram



Number of lines code to an Issue

interactions

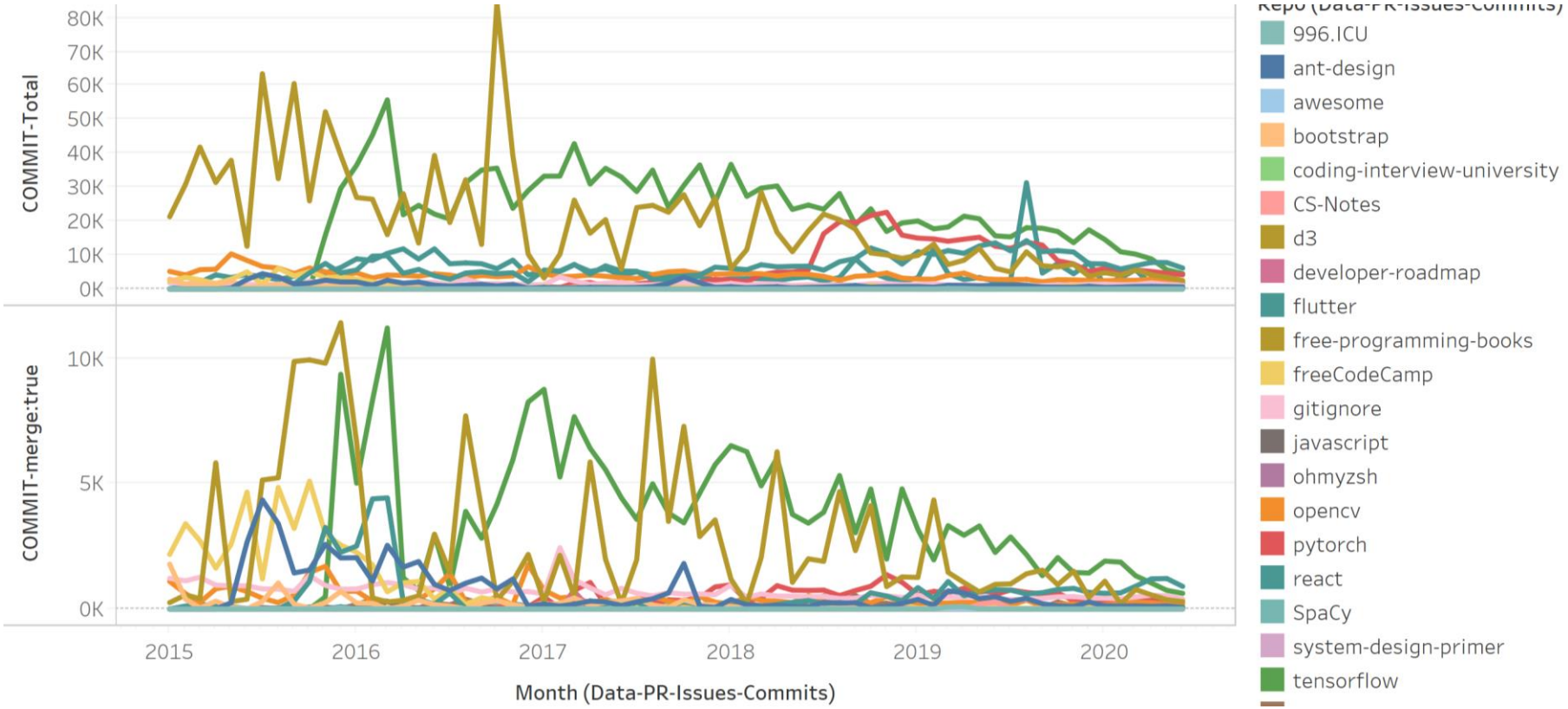


Pull Requests with comments

Top : Pull requests with 0 comments

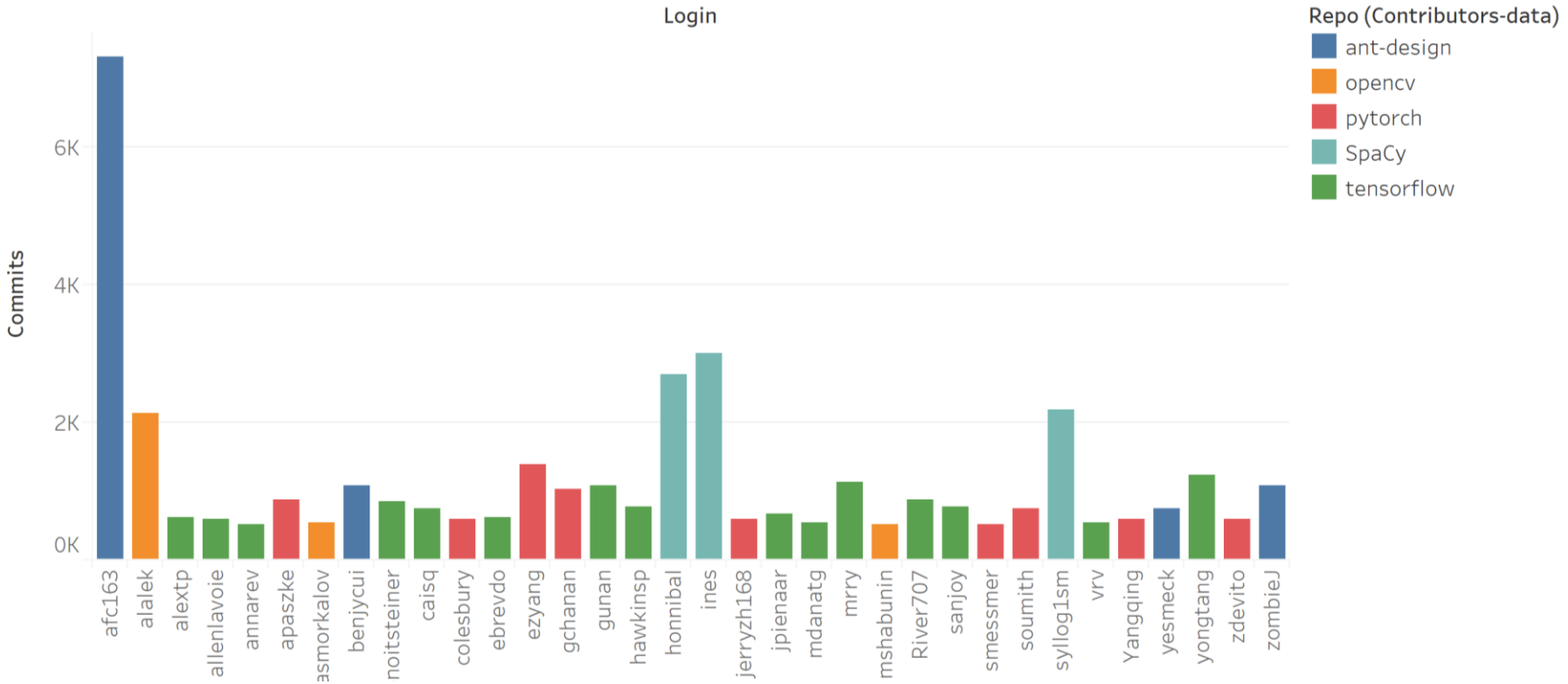
Middle : Pull Requests with 1-20 comments

Bottom : Pull Requests with > 20 comments



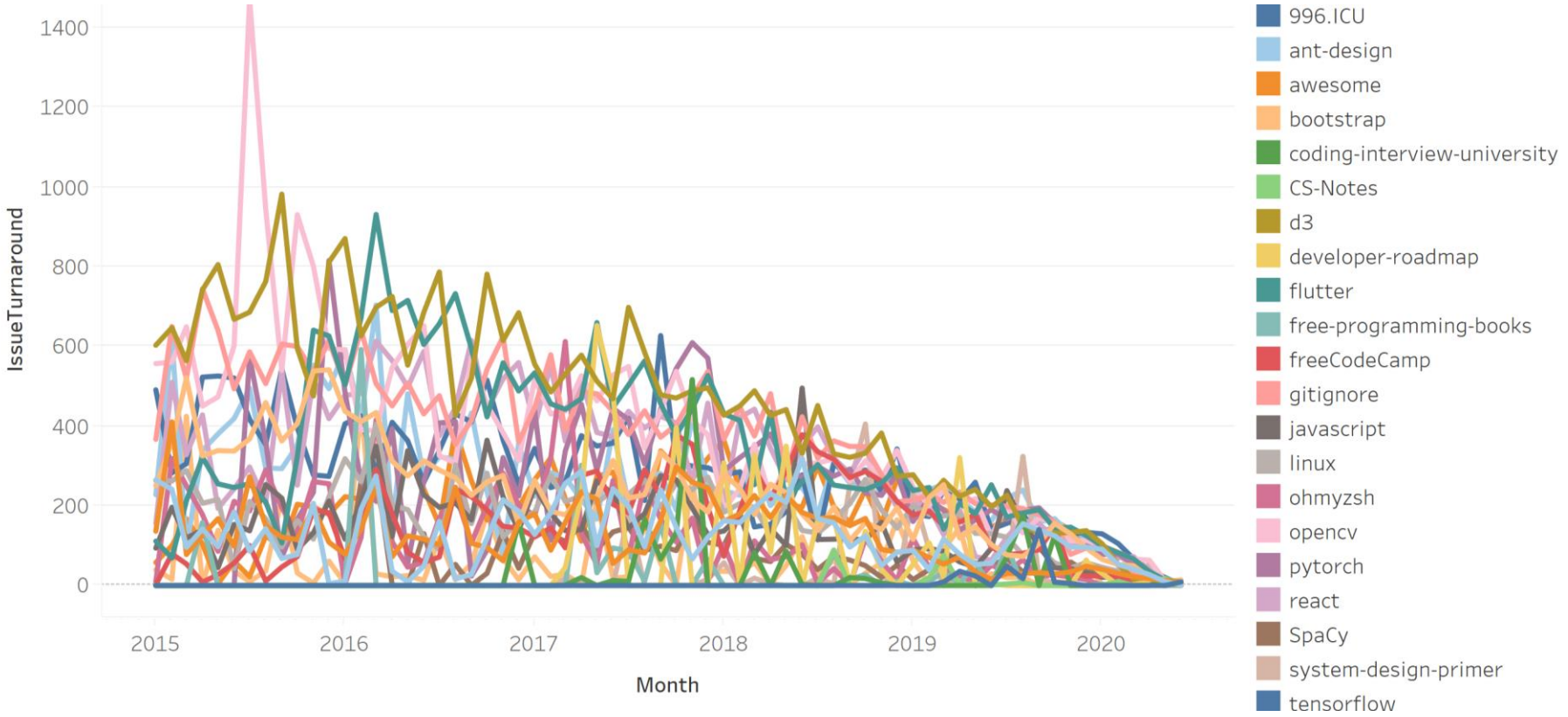
Monthly volume of commits by repository

Linux is excluded – it skews the chart



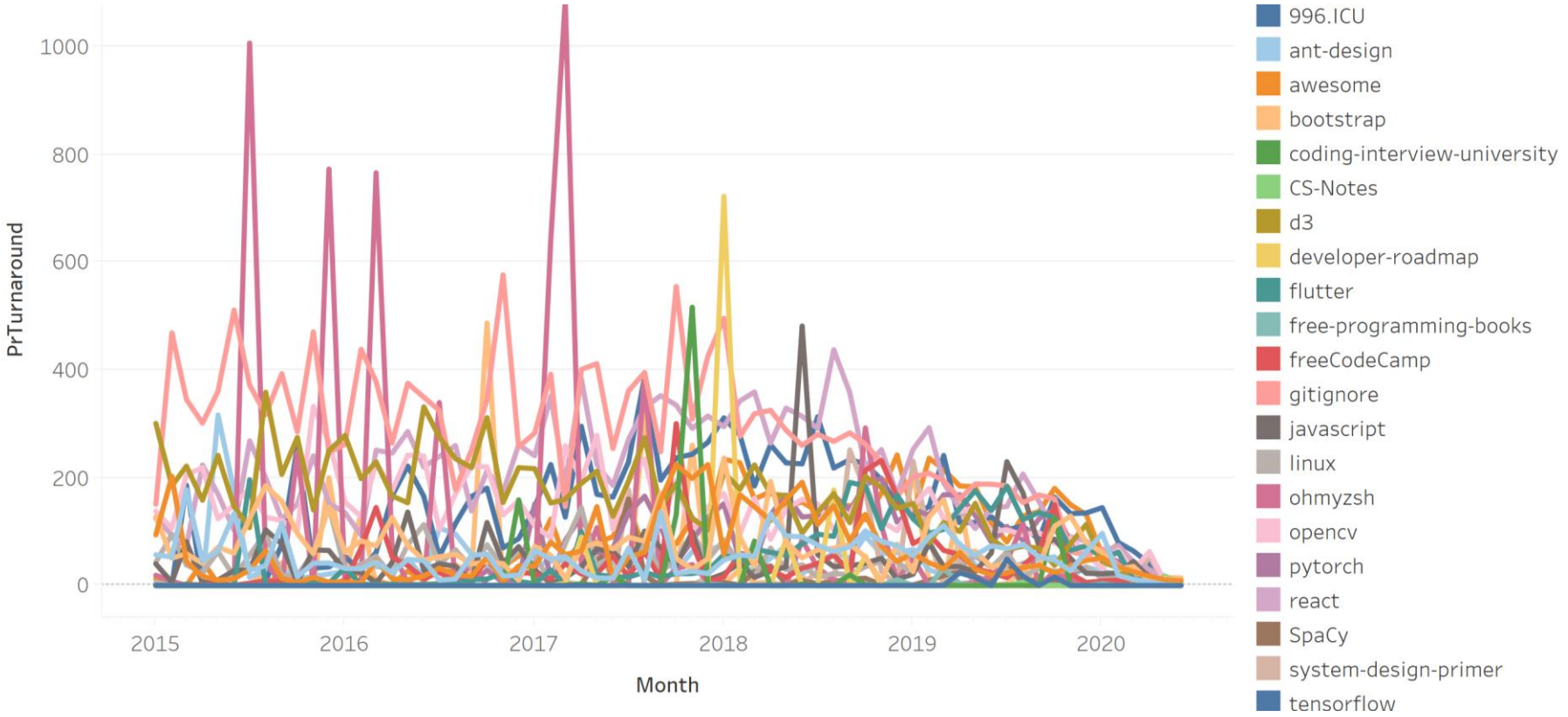
Individual Contributions

Only commits between 500 and 15k included.



Issues Turnaround

Calculated as time in days between closed time and created time



PR Turnaround time

Calculated as time in days between closed time and created time