Empowering healthy communities

Open source is growing, with people from all types of backgrounds contributing to open source as a way to learn and connect. Collaboration and development patterns show us how we can grow and build resilient communities.

The 2020 State of the Vol. V
OCTOVERSE

GitHub
octoverse.github.com

Empowering healthy communities

In this report, we investigate our open source communities: how people use open source to make, build, communicate, and collaborate

Finding balance between work and play

In 2020 we all had to rethink our working spaces and schedules, testing the boundaries between work and home—and we saw that line can be hard to draw.

Finding balance

Productivity report ➜

Securing the world's software

By incorporating automation into workflows, communities are making it easier and faster to identify and remediate vulnerabilities.

Securing software

Security report ➜

//table of contents

Executive summary

56 M+
total developers
on GitHub

As the largest developer platform with more than 56 million developers worldwide, GitHub is in a unique position to provide analysis and speak to trends about communities and collaboration in open source. Understanding how people interact gives us all insights to help create successful models for the future. Our analysis spans several years, depending on the community we are investigating, so we can see how communities grow and change over time.

//executive summary

We begin by analyzing GitHub repositories and see large growth in social interest and data education topics over the past two years. This drives our deeper exploration into three communities that are part of these social and data movements—Python, TensorFlow, and COVID-19—where we analyze user categories and actions. In our findings, we share insights for maintainers and contributors to help them join and build strong communities.

We also share patterns we find in open source collaboration: that developers are sharing and reviewing code faster compared to last year, open source is providing new opportunities to create projects, and Discussions forums allow people to engage with others and share information. We close with a time-series presentation of where open source contributions come from geographically, where they come from now, and a projection of where they are likely to come from in the next ten years.

1.9B+

contributions added in the last year

Key findings

01

GitHub is for more than just software developers.

The number of people on GitHub overall continues to grow, but the proportion of those who identify as developers has decreased, signaling an expanding diversity of those joining the open source community. Growth in education, data, and science categories suggest that future creation and collaboration on the platform will also expand as images, data, and other file types increase the ways that people build and create projects.

02

Resilient communities are about balance.

Some of the best ways to engage newcomers are through issues and new code contributions, but that can burn out maintainers. The strongest communities support growth and participation by engaging newcomers and contributors while also helping provide sustainable work for maintainers. Repositories can outline community norms to set expectations, and use Discussions to foster conversations without burdening maintainers.

03

People are creating and collaborating even more on open source projects.

Open source project creation jumped by up to 40% year over year as people turn to open source as a way to create, learn, and connect with the community. People are also merging pull requests faster than last year, a sign of increased collaboration. This shows us that the community is spending more time on open source projects together.

04

GitHub supports the exploding landscape of distance learning.

More than 900k students used GitHub to learn industry-standard software and build their portfolios, and over 50k teachers automated their course workflows with automated assignments and autograding. When students lost their internships, GitHub partnered with a coalition of companies to connect students with open source projects, mentors, and a stipend to provide a virtual internship.

Take action

Take these actions to build stronger communities.

01

Structure communities to share expertise and foster growth.

The best communities invite contributions without burning out maintainers and senior leaders. Our new Discussions feature is a great way for people to have conversations, and for newcomers to learn community norms. You can use Discussions whether you work in open source or a company: organizational communities of practice can use this model to spread expertise in scalable, sustainable ways.

02

Spread the love and support your global community.

GitHub Sponsors supports thousands of open source contributors, with program availability in 34 regions. One hundred percent of sponsorships go straight to developers. If you use or rely on open source, sponsor an open source contributor to thank them for their work. If you are a contributor, see if Sponsors can help support your open source work and contributions.

03

Reimagine yourself in open source.

Open source is about building software—and so much more. Open source is where anyone like you can find and expand your community, learn new skills, and support a global economy. On GitHub's growing global platform, you can contribute to social good projects, collaborate with world experts to solve pressing problems, teach to a global audience in sustainable ways, and recruit talent from a robust talent pool around the world.

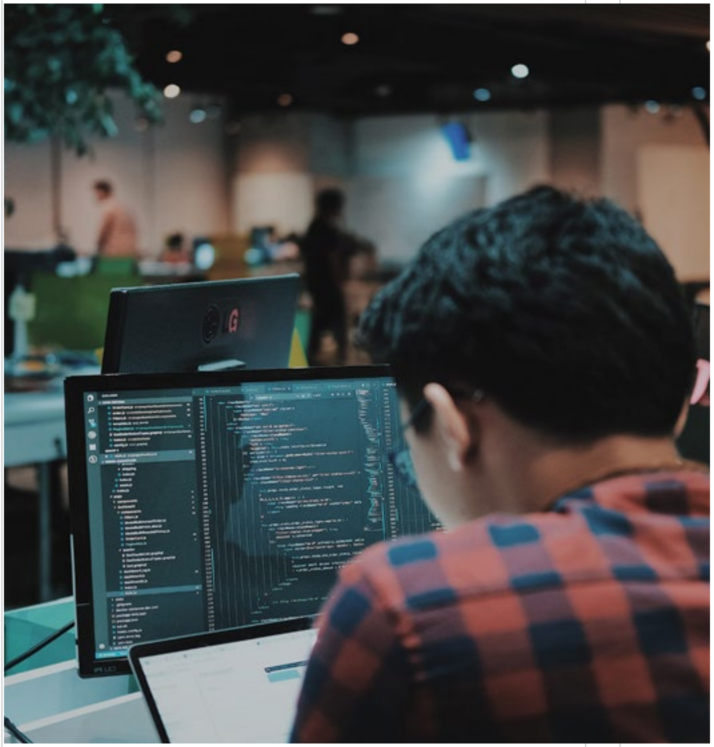
04

Expand your world with GitHub Education.

Whether you're a student or a teacher, using GitHub in the classroom supports learning outcomes such as project management, belonging, and preparation for future work. As a student, you benefit from gaining experience on industry-standard technologies. And as a teacher, you get tools that automate assignment creation, auto-grading, and student progress tracking—giving you more time for mentoring students and work-life balance.

↑ TOC

Data for this report



The data for this report comes from analyzing all GitHub platform activity—public (including open source) and private—year over year. The period of comparison is identified in each section. The periods are often longer than the single year typically used in Octoverse reports because community evolution happens over time while other trends can be seen in yearly and seasonal patterns.

The analysis in the first section of the report is based on the contributions of GitHub platform users, identified as either newcomers or veterans, depending on their tenure, as well as their roles (based on keywords in their bio, such as someone involved in education, software development, or product management). We selected data based on the following:

- Users who had at least one contribution within the community during the measurement window
- Repositories with at least one contribution in the analysis window
- Not classified as spammy, or owned by GitHub staff

GitHub is growing in size and diveristy

People around the world are turning to open source—and to GitHub. It is becoming not only the home for developers, but a platform where people in many roles, doing diverse work, come to learn, engage with the community, and contribute to projects that advance the greater good. As the largest global developer platform, GitHub has a unique opportunity to understand how these individuals identify and interact with each other, and how our community is growing and changing. By reading this analysis, members of the community can gain better insight into the new ways they can build, grow, and create on GitHub.

The changing GitHub community

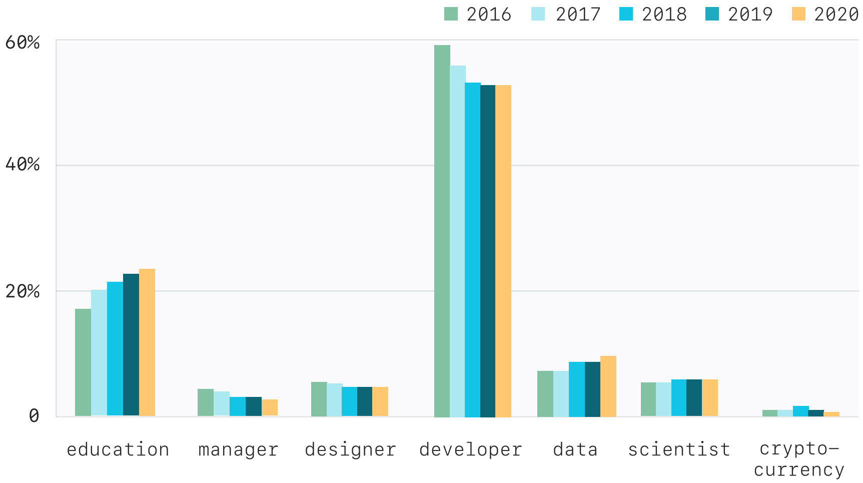
To understand who uses GitHub, we analyzed bios provided in user profiles[1] and see that our community has grown to include more than just developers. Based on keyword analysis, profiles were categorized as education, manager, designer, developer, data, scientist, and cryptocurrency, with more than one category possible per person.

The way people describe themselves and their work has shifted over the past five years. Those who describe themselves as developers remain the largest category of users on GitHub, though it is decreasing from a high of 60% in 2016 to 54% in 2020. Profiles that include education are growing, up from 17% in 2016 to 23% in 2020. The third largest category is data, with 7% of users in 2016 and 10% of users in 2020. This shift shows that the world of GitHub is growing—not just in numbers, but also in diversity, with existing users reimagining the projects they can build and how they collaborate as they invite others to join them in building projects for their world. For those who haven't yet tried GitHub, the shift illustrates opportunities for all types of people to work on a variety of projects in online communities.

The rise in education indicates that this growth will continue into the future, as students and teachers expand on the foundation they've built, both in their professional work and in their collaborations on open source projects. Education creates a bridge to long-term careers in software development, and creates opportunities to reach developers globally in ways that weren't possible just a few years ago.

[1] See methodology for details.

Distribution of roles over time



There are many ways to contribute to open source, and our growing community is exploring and expanding these possibilities.

TOC

GitHub in education: serving the next generation

Over the course of eight years, GitHub Education has served millions of students at thousands of schools. Version control offers a number of affordances for students and teachers who use it to take snapshots of their work, roll back to a known good state, and collaborate on group projects. Students are motivated to build their portfolios, learn industry best practices, and connect with hiring managers. Teachers use Git and GitHub to automate their course workflows and integrate real-world workflows. In fact, using GitHub in the classroom predicts learning outcomes such as project management, belonging in the field, and preparation for future work.

Teaching for the first job in industry

Teachers everywhere are pressed for time and resources. GitHub Classroom, a tool to run courses on GitHub, lets teachers automate assignment creation, access control, and track student progress. Teachers often use their choice of continuous integration and continuous delivery tools to automate feedback when students push their code. This spring, GitHub Classroom added "autograding," which is an implementation of GitHub Actions. As of September 30, 14% of assignments on Classroom use autograding and 360,579 tests have been run across 67,398 repositories, preparing students for test-driven development in the workplace.

Contributed by Vanessa Gennarelli

Hands-on learning with distributed teams

In the spring of 2020, students who hoped to gain valuable experience in summer internships began to receive disheartening news. Many technical organizations couldn't host students on-site and the recession reduced funds for recruitment.

Together with a coalition of companies, GitHub and Major League Hacking launched the MLH Fellowship, which connected students with a paid open source summer project and paid a mentor to review their contributions. Of the maintainers from the 36 open source projects, 76.2% reported that their project was in a better state after the MLH Fellowship. The program has evolved to three different tracks, with Facebook and AWS participating again in the fall of 2020.

The common area in the cloud

Students are eager to find ways to connect and continue their learning journeys. When it became clear that most seniors would not have a typical graduation, GitHub Education offered a unique experience: git remote graduation. By submitting a pull request, seniors could add themselves to the "Class of 2020," celebrate with other graduating students, and receive dedicated swag items. Of the 4,000 pull requests students opened, 25% had not been active in the previous year. Over the course of the event, 2,600 students "walked" across the stage.

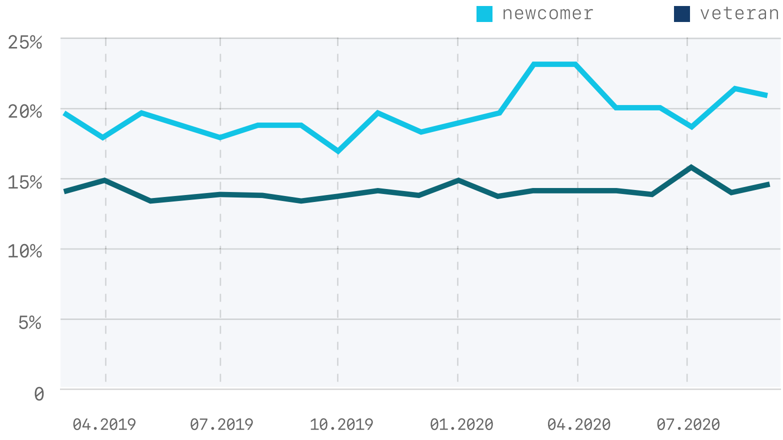
For more about GitHub education,
check out the 2020 Classroom Report. →

Activities of newcomers and veterans

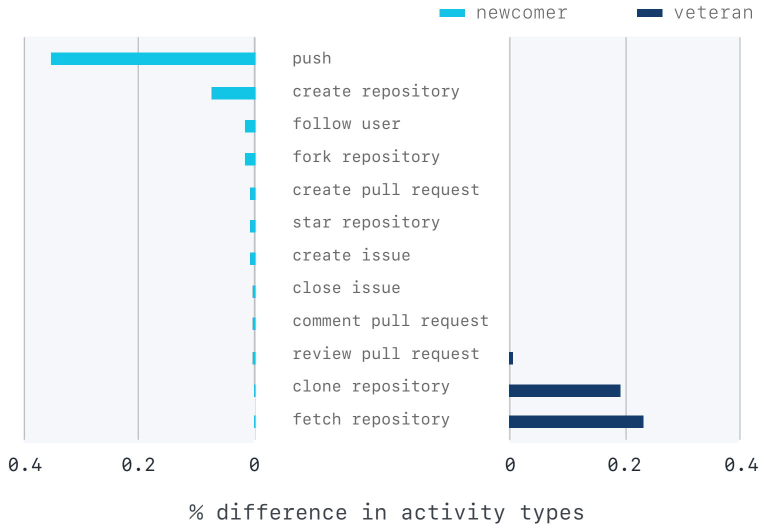
New growth brings new people to the community. Newcomers are engaging on the platform and participating in open source with more than just code: they push data files, images, and other content more often than veterans. Here, we define newcomers as those who created a GitHub account in the previous 28 days, and veterans as those who have had an account for two years or more.

When we compare the actions taken by newcomers and veterans across all GitHub repositories, newcomers pushed code and created repositories much more than veterans, while also interacting a bit more than veterans: creating and commenting on issues and creating pull requests, for example. In contrast, veterans cloned and fetched repositories much more than newcomers—they also reviewed pull requests more than newcomers, though the effect was not as large.

Ratio of pushed files that are non-code



Differences in activity types for newcomers vs. veterans



This shows that newcomers largely use GitHub in straightforward ways: pushing code, creating issues, and starring repositories. Veterans participate with more advanced actions, such as fetching and cloning repositories, and especially those that help shape the direction of a code base, seen in reviewing pull requests.

When we see a platform being used for new types of creation and collaboration, it's often a signal that the platform is becoming recognized as a way to work effectively, efficiently, and across boundaries. This shift to newcomers building more diverse projects is the mark of a maturing and evolving platform, and shows that the GitHub community is already embracing the opportunities.

We will use this frame—the categories we've identified and newcomers/veterans—to see how different communities are growing and engaging on GitHub.

The changing landscape of GitHub project communities

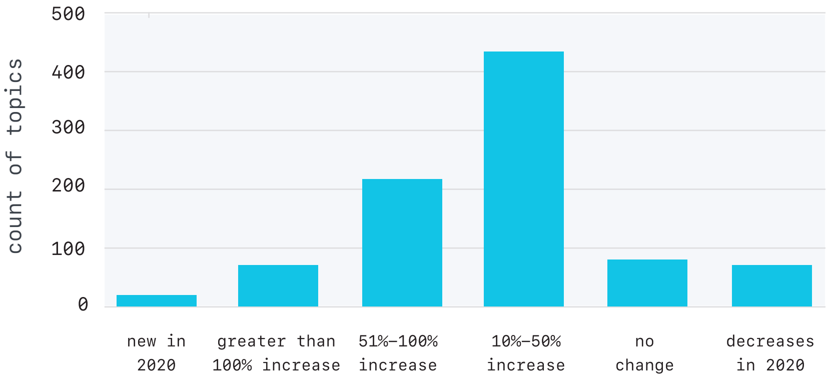
What is a community?

While there is no single definition of an open source community, we often group repositories related to a similar topic to help identify the members and primary concern of a community. For the purposes of the following analysis, we define a community as a group of repositories that work on a similar topic and the people that collaborate on those repositories.

There are over 750,000 unique topics applied to repositories on GitHub. We analyzed growth in topic application to repositories between 2019 and 2020 to frame our deep-dive analysis that follows in this section. We've limited this analysis to only those topics that were classified by our user base to more than 500 public repositories in either period, resulting in over 800 topics of interest.[2]

Of the over 800 topics we analyzed, only a very small percentage, approximately 1.5%, were wholly new in 2020. For those topics that existed in 2019 and 2020, over half of them saw increased usage year over year (between 10% and 50%), while a small amount of topics (6.9%) saw declines in use.

Percent change in topic application to repositories



[2] Not all repositories have topics applied, but when they do, users can either type in their own topic or select from existing topics. If they select existing topics, those are easier for community members and other contributors to find and identify; this consistent identification also aided in our analysis for this report. There are likely more repositories that do not have a topic applied or have an inconsistent topic that are dedicated to the topics we investigate, but they were not included in our analysis.

Open source project communities

We chose three communities to explore: COVID, TensorFlow, and Python. These three allow us to compare and contrast different types of communities, and how users may interact or cluster differently, based on the community characteristics. We can think about the communities we analyzed this way:

Broad community

GitHub open source is the largest community, composed of all open source repositories. There is no unifying theme or topic here, however patterns we observe can be helpful for communities managing large software projects, such as fostering growth and collaboration.

Language-based community

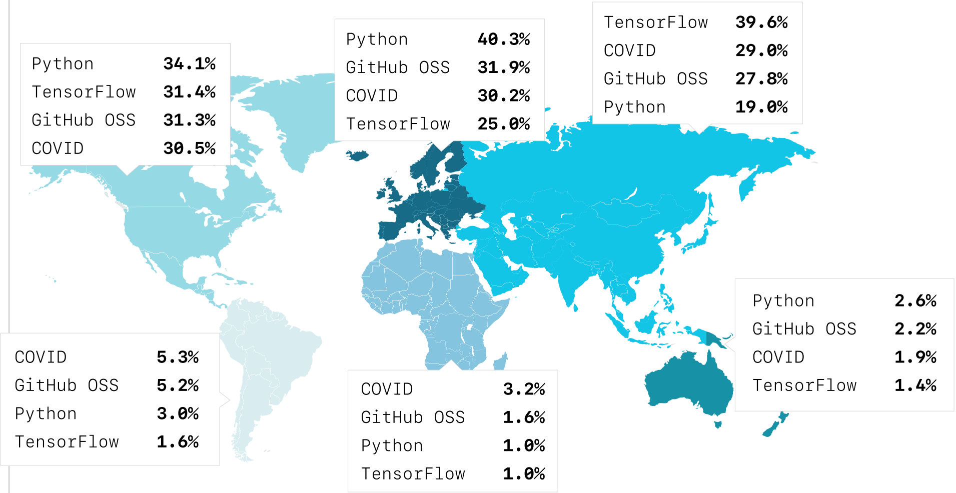
Python is a community made of all repositories in the dependency graph that use the PyPI package manager and those who contribute to them. Python is a "goldilocks" community—not broad, not niche—and use of a single language ties repositories together while still allowing broad work to be done. We selected Python because it is seeing some of the fastest growth on GitHub and is used in many contexts.

Niche communities

TensorFlow is a community made of the top ten repositories related to the TensorFlow project, which is a comprehensive platform for machine learning. The TensorFlow community is a mature, established community, and contributions benefit a technical project.

COVID is a community made of the top 100 COVID-19-related repositories, based on the number of contributions. COVID was selected because it is an open source for good project and is a new, emerging community, providing a comparison to the TensorFlow community. You can read more about open source for good projects later in the report.

Distribution of distinct contributor count to each community by region



Percents per region | All activity since 2013 | OSS = open source

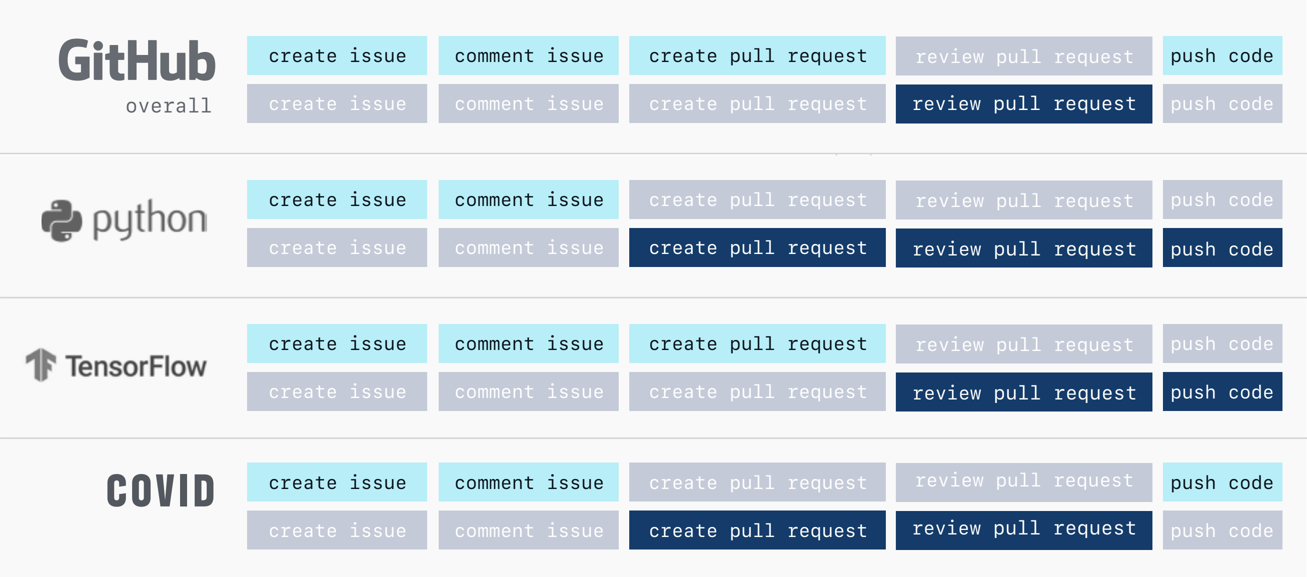
GitHub activity at-a-glance

Comparing and contrasting the analysis of user categories and actions across GitHub reveals insights into how the platform is growing.

Across the communities we investigated, newcomers (those on GitHub 28 days or less) and veterans (those on GitHub 2 years or more) engage differently in communities. Contributions from new members are often core to learning and interacting in a community: creating and commenting on issues. On GitHub and in the COVID community, newcomers also pushed code. Veterans spend time helping shape code: reviewing and creating pull requests. Understanding how people typically work in communities can give us insights to create successful models for expanded interaction.

Our analysis details engaging newcomers, who most often return when they review pull requests or push code, but this isn't common. Other strong predictors are creating and commenting on issues, but this can be difficult for community maintainers. One promising pattern is from the Python community, where Discussions engaged new members. We highlight Discussions as a way for communities to have conversations and for people to learn the norms of a new community without overwhelming maintainers.

We discuss these findings next, beginning with the Python community. The following sections provide detail for the language-based and niche communities chosen for analysis.



GitHub overall	░ create issue	░ comment issue	░ create pull request	▒ review pull request	░ push code
	▒ create issue	▒ comment issue	▒ create pull request	█ review pull request	▒ push code
python	░ create issue	░ comment issue	▒ create pull request	▒ review pull request	▒ push code
	▒ create issue	▒ comment issue	█ create pull request	█ review pull request	█ push code
TensorFlow	░ create issue	░ comment issue	░ create pull request	▒ review pull request	▒ push code
	▒ create issue	▒ comment issue	▒ create pull request	█ review pull request	█ push code
COVID	░ create issue	░ comment issue	▒ create pull request	▒ review pull request	░ push code
	▒ create issue	▒ comment issue	█ create pull request	█ review pull request	▒ push code

Stacked lines show top line activity for newcomers vs. bottom line activity for veterans

▒ activity
░ newcomer more active
█ veteran more active

↑ TOC

Python community

The data for this section of the analysis is from December 2013 to September 2020—the time period for which we have Python data.

The data includes:

- All repositories in the dependency graph using the PyPI package manager, which is approximately 2M repositories
- All users who had at least one contribution within the community during the measurement window
- Activity not classified as spammy, and no repositories owned by GitHub staff

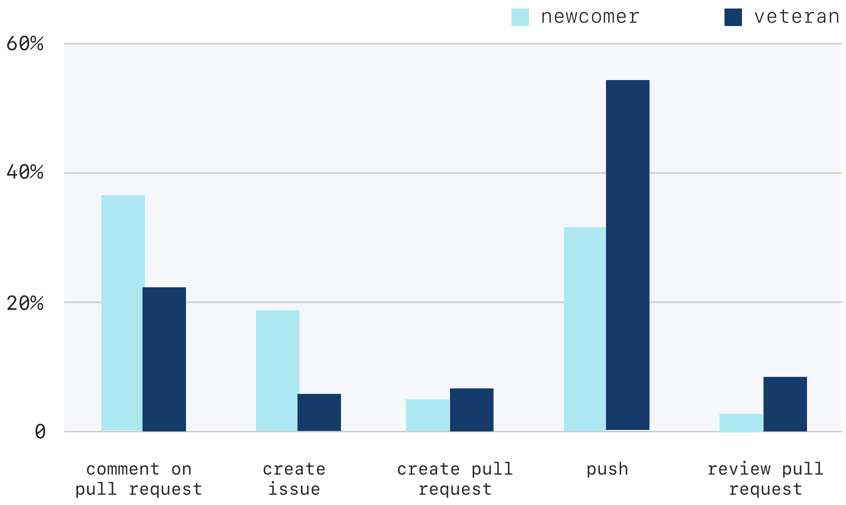
Evolving Python roles

The evolving distribution of user categories in the Python community shows us that developers are still an important and strong contributing group over the past five years. Over the same time period, other groups, including data and scientists, are a growing part of the community, with education showing some of the strongest growth. This signals exciting potential for Python in the future, as educators invest in teaching new students the language, and as those learning the language carry it into their workplaces and long-term open source projects, along with data. These are echoed in growth patterns.

Python community behavior

Newcomers to the Python community create and comment on issues more often than veterans. This is similar to the behavior seen in the broader GitHub community, though here the differences between newcomers and veterans are larger. Newcomers also create and review pull requests less than veterans. One notable difference is in pushing code: veterans push much more code than newcomers, which is different from the behavior observed in the GitHub community overall. This may be because participating in a language community requires more advanced skills, or that many of the repositories in these communities have guidelines that encourage newcomers to participate in other ways prior to contributing code.

Distribution of actions for newcomers vs. veterans in the Python community for all time

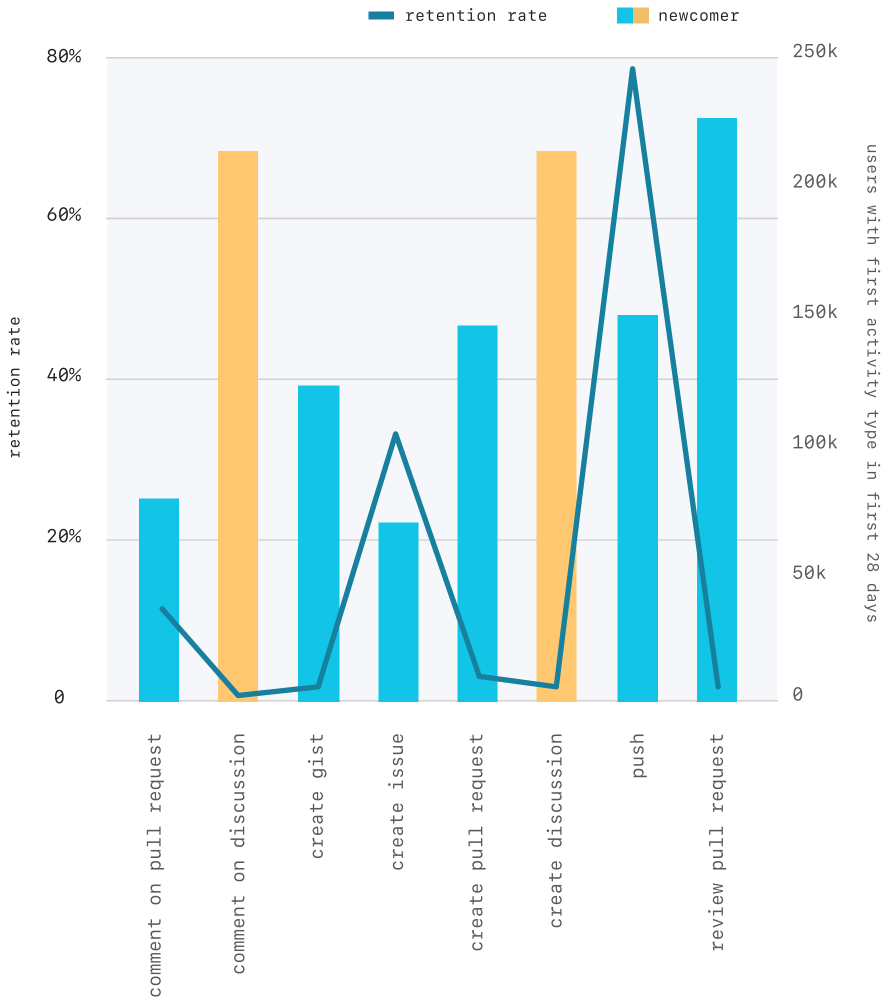


How can open source communities improve retention in their communities? Individual communities have special characteristics, but common patterns provide clues. For this analysis, we look for a first contribution, and then a follow-up contribution in the ensuing 28-day window.

To read the chart, the blue and yellow bars indicates the percentage of people who contributed in the next 28 days, and the dark line indicates how many did that action. For example, in the Python community, reviewing a pull request is the best predictor of someone contributing in the next 28 days, with 73% returning, but only a few in the community do this as their very first action. In contrast, the action done by the most is pushing code (almost 250,000 people), which sees a 49% rate of returning and interacting with the community in the following 28-day window. We also see continued engagement when they create issues.

This suggests that one way to engage in this community is to push code and create issues. However, we also know that creating issues can be intimidating for new contributors who don't know a community, and it's not always sustainable for maintainers. There also may be opportunities to create and comment on team Discussions, which also engage newcomers (yellow bars); we note these are new to the community and not many have interacted with Discussions yet. We encourage communities like this to think about ways to engage with and encourage new users to participate and continue contributing.

Retention rate based on first contribution in Python community



Discussions engage newcomers without burdening maintainers (highlighted in chart)

Discussions: a new way to talk about work

Many conversations that happen on GitHub occur in issues where people can discuss code, plan and track work, and manage tasks.

Earlier this year, GitHub announced Discussions, a way to collaborate and communicate in ways that are more open. Teams can post updates or have a conversation that spans projects or repositories in a forum. This creates a new way to see how we communicate and share information, or learn team norms and values. An exploratory analysis looking at the use of Discussions in the next.js repository shows how people communicate in an open source project in ways that aren't tied to project management or issue tracking.

The next.js repository started using Discussions in January 2020, and by October 2020, was using over 3,000 discussions to communicate and collaborate. Almost half of those who have contributed to the repository in the past year have used Discussions, and 25% of all-time contributors use Discussions, making it a great way for all types of contributors to engage, have conversations, and get to know their community. Of the roughly 10,000 all-time contributors to the next.js repository, almost 2,445 participated in discussions, or 24%. Of those 2,445 who participate in discussions, 1,140 (47%) made contributions to the repository in the past year. Half of those who push code/contribute to the repository also participate in Discussions.

Prior research shows that people value the ability to separate conversations from work, while still sharing a common platform. Our own qualitative research echoes that sentiment, with one user saying, "Discussions is a huge step forward for us...the ability to better spec out features and get [conversations] going in a way that isn't Issues."

Early data from teams using Discussions shows this is a middle ground where anyone can come, observe, learn, and interact.

This provides an opportunity for us to think about different ways for people to engage with open source communities. Participating in and watching Discussions can be a good way for newcomers to learn community norms and patterns in a safe way that doesn't overwhelm maintainers. These patterns have applications in enterprise settings, too. Participating in and watching Discussions lets newcomers safely learn community norms and patterns in sustainable ways.

Python and its role in open source

Python's open source community is more than just its members and what they do. It's also the impact the community makes through an interrelated network of software packages and contributors that rely on Python. By contributing to the Python community, contributors and maintainers help support 266,966 packages, and the work of 361,832 fellow developers and contributors over the previous year from 202 countries and regions.[3]

This impact is due to the work of the entire Python community, but we would like to recognize the top repositories for their work and support for open source.

Top 10 Python packages with the most unique contributors over the last 12 months

No.	Repository	Contributors
1	tensorflow/tensorflow	11,138
2	home-assistant/core	8,162
3	pytorch/pytorch	5,934
4	ansible/ansible	5,150
5	ytdl-org/youtube-dl	4,810
6	huggingface/transformers	3,557
7	Azure/azure-cli	3,501
8	pandas-dev/pandas	3,340
9	FortAwesome/Font-Awesome	2,990
10	tensorflow/models	2,580

[3] Number of contributors and countries reported are limited to those hosted on GitHub.

Behind the scenes

Gina Häußge is a software architect, full stack developer, and GitHub star. Gina is the creator and maintainer of OctoPrint, which is an open source web interface for 3D printers built on Python. Gina has worked full time in open source since 2014 and is fully crowd-funded; one way that she supports her work is through GitHub Sponsors.

If you use open source software, consider sponsoring a contributor to say thanks for their work.
Learn more about GitHub Sponsors in our docs. →

GitHub Sponsors

GitHub Sponsors makes it possible for any open source contributor, like Gina, to work full time in open source. Many open source contributors in the Python community are supported by Sponsors, too, with more than 1,100 Python repositories and over 680 contributors supported by Sponsors, representing more than 40 countries and regions.

With Sponsors, any GitHub user can sponsor any open source developer or organization to recognize contributors working behind the scenes. Sponsors is available in 34 regions and 100% of money sponsored goes straight to developers. Right now, thousands of open source contributors around the world are being supported by tens of thousands of sponsors.

Caleb Porzio's story is an exciting example of how GitHub Sponsors can support open source contributors. After a sabbatical from work, Caleb noticed GitHub Sponsors and signed up for the program. Within seven months, he was able to shift into full-time open source development and has made over $100,000 per year. Check out Caleb's own story for more details about his journey, along with tips and tricks for others to maximize GitHub Sponsors revenue. You can sponsor Caleb, too.

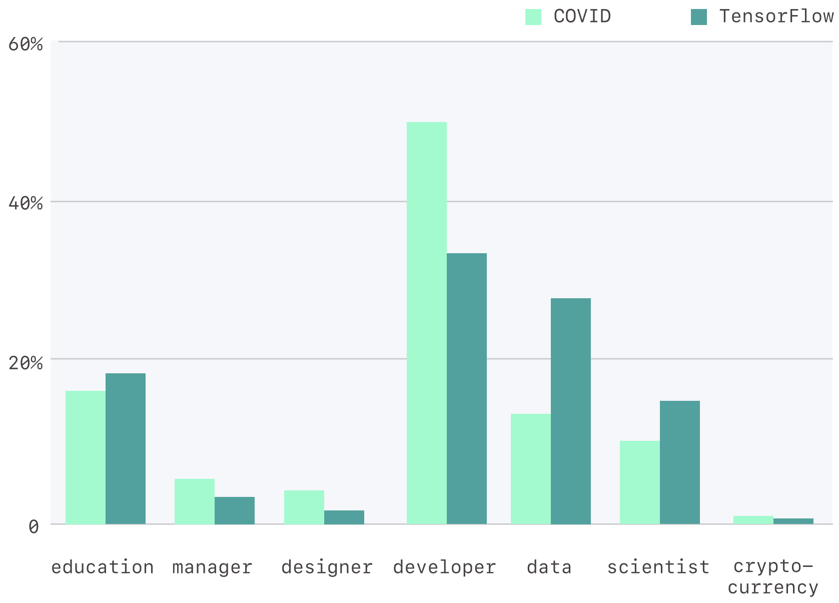
TensorFlow and COVID communities

We present these two communities together because they are both smaller communities than our previous investigations of GitHub and the Python community. They also present an interesting contrast both in terms of age (TensorFlow is a mature community, while COVID is relatively new) and purpose (TensorFlow is a technology good community and COVID is a social good community). The TensorFlow community is defined as the top ten Tensorflow repositories, based on the identified "topics" tag in /explore. The COVID community is defined as the top 100 COVID repositories, based on the total number of contributions.

The data for this section of the analysis comes from:

- Users who had at least one contribution to their respective community during the measurement window. This includes contributions since 2015 for TensorFlow because we have data for TensorFlow going back to 2015, and any contribution in the COVID community beginning in 2020 because it is so new.
- Activity not classified as spammy, and no repositories owned by GitHub staff

Distribution of roles across TensorFlow and COVID communities



Most-represented roles

Similar to the trends on GitHub overall, developers are the most common users in the TensorFlow and COVID communities, with education also showing strong representation. Unsurprisingly, as a machine learning platform, TensorFlow also has strong representation for those focusing on data usage.

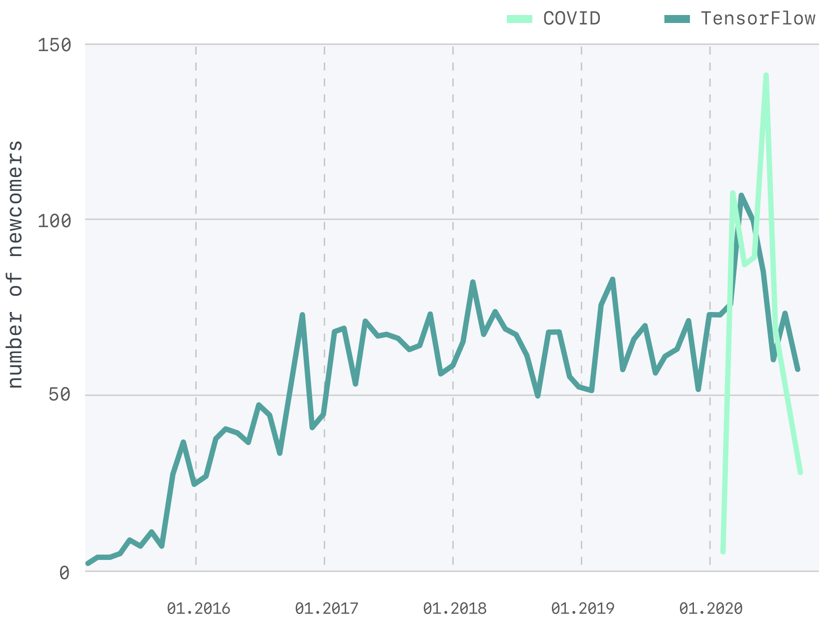
When we look at growth in the TensorFlow community overall, we see how closely data users follow developers, truly co-driving this community. TensorFlow also has strong participation from those in the education space and scientists, signaling the community offers expanding collaboration and contribution opportunities. Growth in the community, indicated by newcomers over the past two years, is strongest in the data, developer, and education categories.

When looking at the COVID community, we see the strongest growth from developers, followed by those in education and data. The strongest spike in activity, expectedly, is as the community forms and core functionality is introduced.

If we explore contributions from newcomers, we see a large spike in early 2020, when both communities saw statistically significant growth. This is correlated to the COVID-19 outbreak and is likely a response to people being drawn to the opportunity to learn and contribute to open source.

The large surge in the COVID community suggests that timely social good projects may present an exciting opportunity for people to get engaged in open source communities. Our analysis warrants additional investigation into these communities.

Newcomers who contribute to communities in their first 28 days



Growth in 2020 is statistically significant

Community actions, patterns, and insights

We see noticeable differences in how different groups use GitHub, primarily in the TensorFlow community. In other communities, new members push code more often than tenured ones, but in TensorFlow, it's almost nonexistent, and even veterans don't push often. Here, the primary mode of interaction is issues. In the COVID community, both newcomers and veterans push code frequently, with new members pushing a bit more often. Newcomers also create and comment on issues whereas veterans perform more advanced actions, such as creating and reviewing pull requests (although they still contribute and engage in many activities in the community). This highlights more than just a comfort in using more advanced functions on GitHub. It underlines the role and influence veterans have in managing and shaping the community. By taking a leading role in how code is changed—suggesting changes, discussing potential modifications, and approving changes—these more-experienced contributors not only shape the codebase, they influence how interactions and conversations happen around these changes.

In both communities, commenting on issues is the most common activity when describing activity across all users. In the TensorFlow community, we notice an uptick in pushes starting mid-2018, likely in anticipation of the January 2019 announcement of TensorFlow 2.0, which then went live in September 2019.

You can see contribution patterns we describe for TensorFlow and COVID communities compared across all developers and newcomers in the appendix.

The increase in diversity in contributor roles demonstrates a rising trend of involvement by those other than developers, and is exciting for the growth and future of open source.

Collaborating for social good

Open source for good (OS for Good) projects advance or positively promote social good causes, particularly around the UN Sustainable Development Goals. Such efforts include supporting domestic violence victims, helping identify safe restroom access for transgender and gender-nonconforming individuals, and strengthening public health responses to COVID-19.

Developers and community members contribute to joint projects in an open source model, but their projects differ from what typically comes to mind when many think of open source. OS for Good projects often focus on standalone tools that have a graphical user interface and address use cases not typically seen in other open source projects, though they can also include tools and projects that contribute to infrastructure technology. In contrast, the projects seen in the broader open source ecosystem are more often developer tools, programming language projects, or infrastructure technology—projects that often lack tools or applications that allow an end user to interact with them.

Contributed by Denae Ford, Yu Huang, Mala Kumar, Tom Zimmermann

Recent research shows that those participating in social good projects have different motivations and reasons for contributing compared to other open source projects:

- Contributors are significantly more motivated by solving societal issues than benefiting themselves by learning skills or building a career portfolio.
- Contributors investigate and care deeply about the owners of projects they contribute to, and tend to prioritize projects that meet global needs, have long-term benefits, and benefit their personal connections.
- Contributors work for social sector organizations that create open source tools.

GitHub's Tech for Social Good program has done foundational work in this area. To get involved and learn more, check out GitHub's Open Source Software in the Social Sector report on the Social Impact site. You can also learn more about Tech for Social Good, sign up for the OS for Good listserv, and more. If you prefer, you can watch or listen to a conference talk on the Social Sector and Open Source from Mala Kumar, an expert on OS for Good.

You can learn more about research studies to support OS for Good, which are built on the Open Source Software in the Social Sector Report, by checking out the project site from Microsoft Research's Yu Huang, Denae Ford, and Tom Zimmermann. →

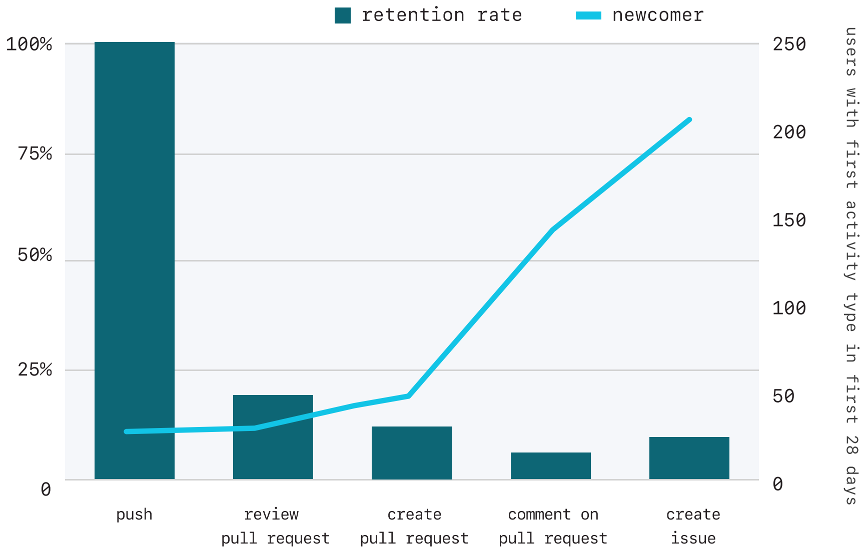
Return to p 13 →

In contrast, newcomer activity is quite different between the two. There is a clear pattern of typical behavior in the TensorFlow community: Most new members comment on issues (50% to 75%), some create issues (25% to 30%), and a few review pull requests (~5%). Only two individuals pushed code. Seeing such a stark contrast in behavior for new members led us to investigate the community guidelines in the tensorflow/tensorflow repository. We found that when joining, members were asked to read community guidelines, and most notably, submit a Contributor License Agreement prior to submitting their own code. The method chosen by this community to communicate and enforce standards for behavior has shaped contribution patterns almost universally. In the case of a mature community, this may be a smart strategy to introduce new members to norms of the group.

In the COVID community, we see a spike in pushes from new members, which fall again after three months. This may be explained by two reasons: first, because this was a new community and once the core functionality was running, no bursts of new code were necessary. Second, as new members join, they "age out" of our definition—and therefore analysis—of a "newcomer." While this is true of all our analyses, this difference is more likely because COVID was a new community creating new projects. As new pushes level off, commenting on issues takes over, which is similar to activity seen in the TensorFlow community. In this emerging community, fast building and collaboration was important, so less-formal rules enabled fast growth. If the COVID repositories had systematically required a formal agreement, they may have seen slower growth.

TensorFlow community guidelines make it uncommon for new members to commit code, but when they do, they contribute again. When they create or comment on issues, they are likely to make another contribution the following month. We don't have enough data to conduct a similar retention analysis in the COVID community.

Retention rate based on newcomers' first contribution in TensorFlow community



Building a community: You can't do it alone

We refer to the TensorFlow and COVID communities as "niche" communities, but they are part of the interconnected world of open source. As such, they have grown stronger and faster through the power of the community.

The TensorFlow community depends on over 11,200 repositories, the work of almost 380,000 contributors, and draws from over 200 countries and regions. The COVID community depends on over 11,700 repositories, the work of almost 383,000 contributors, and draws from over 200 countries.[4] These are truly global efforts that succeed because of the broader open source community that supports them.

Top 10 repositories that COVID and TensorFlow communities depend on

No.	COVID	TensorFlow
1	DefinitelyTyped/DefinitelyTyped	numpy/numpy
2	feross/safe-buffer	tensorflow/tensorflow
3	visionmedia/debug	matplotlib/matplotlib
4	lodash/lodash	scipy/scipy
5	vercel/ms	scikit-learn/scikit-learn
6	isaacs/inherits	python-pillow/Pillow
7	jshttp/mime-db	benjaminp/six
8	ChALkeR/safer-buffer	pandas-dev/pandas
9	npm/node-semver	h5py/h5py
10	jshttp/mime-types	protocolbuffers/protobuf

[4] Number of contributors and countries only captures packages hosted on GitHub.

Open source collaboration and creation

What have we learned about open source collaboration and creation, when much of the world is locked at home? We analyzed two data points to gain insight into this:

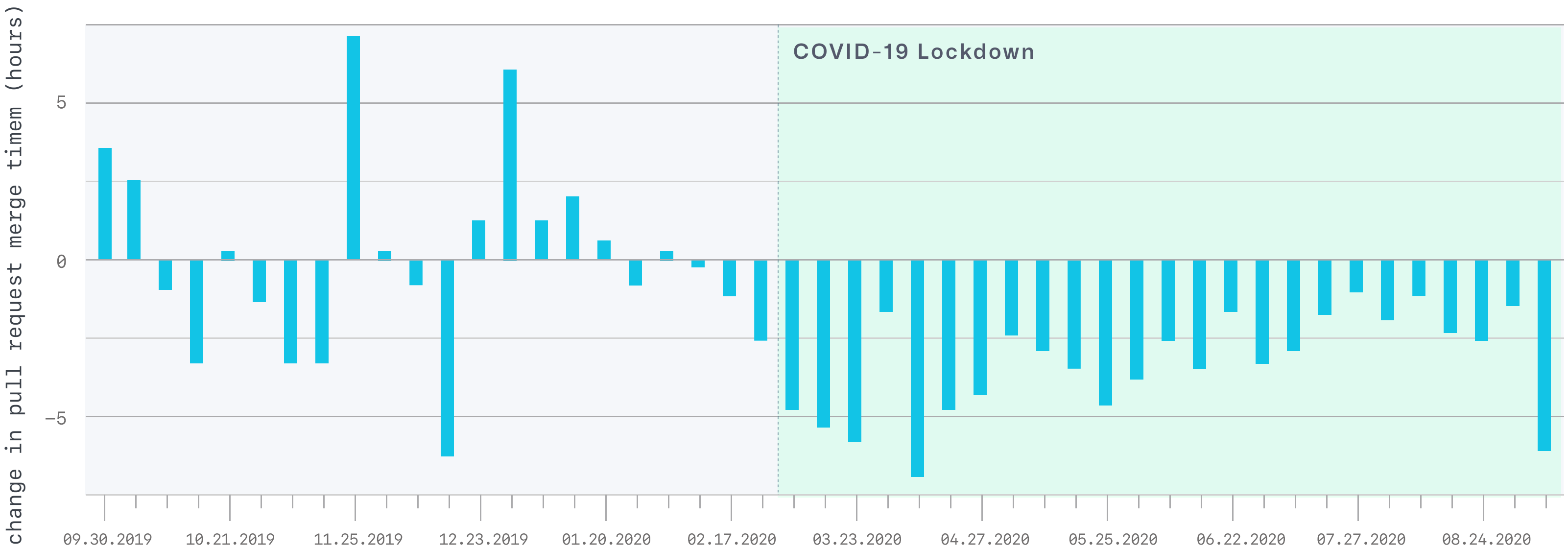
Sharing and reviewing code
We can't merge pull requests without others, so this let us peek into how open source collaboration has changed compared to last year.

Open source creation
The role that open source plays in our lives has changed as people stayed home in 2020.

Sharing and reviewing code

Developers collaborate on code with pull requests—and the speed of pull request merge times indicates the level of collaboration. Early in the year, time to merge pull requests took a few hours longer compared to last year. In March, time to merge begins to be much faster compared to last year, ranging from **45 minutes to almost seven and a half hours faster in comparison**. This could suggest that people are more engaged in their open source projects and more responsive, particularly following shelter-in-place orders, as they are finding more projects they can do from home.

Average change in time to merge pull requests for open source projects vs previous year, weekly



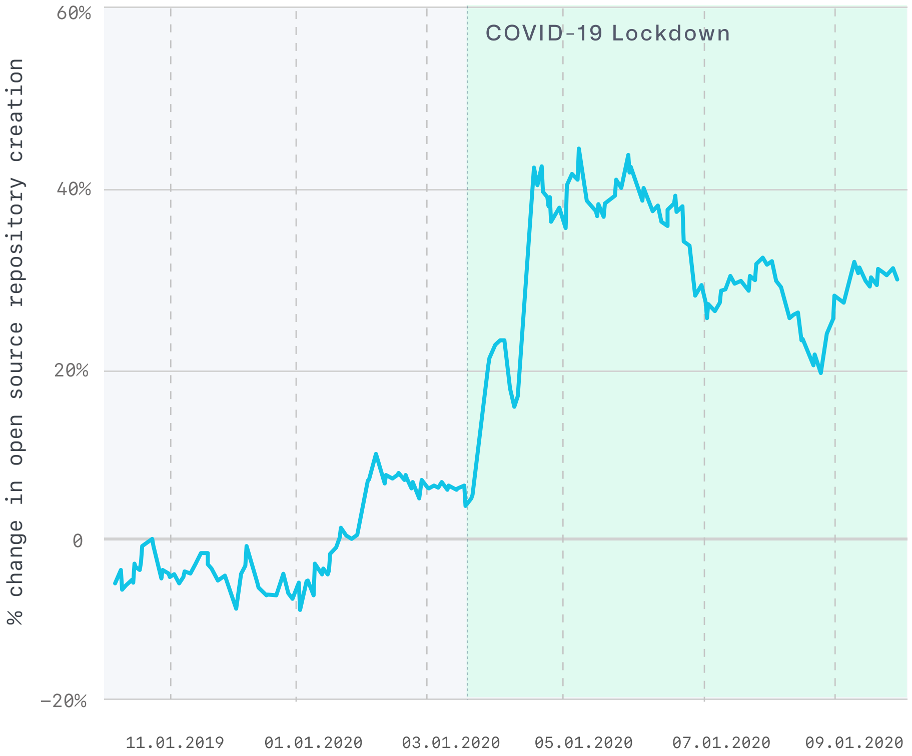
The high and low spike in November is due to the US Thanksgiving holiday

Creating and sharing projects

The first step in sharing a project is starting one. Whether it's their first project or tenth, a hobby project or a technical deep dive, many GitHub developers create a repository and then invite others to join them. In the last year we've seen significant growth in open source project creation. In May, over 40% more repositories were created compared to last year, and since then, roughly 25% more open source repositories have been created compared to the same time period last year.

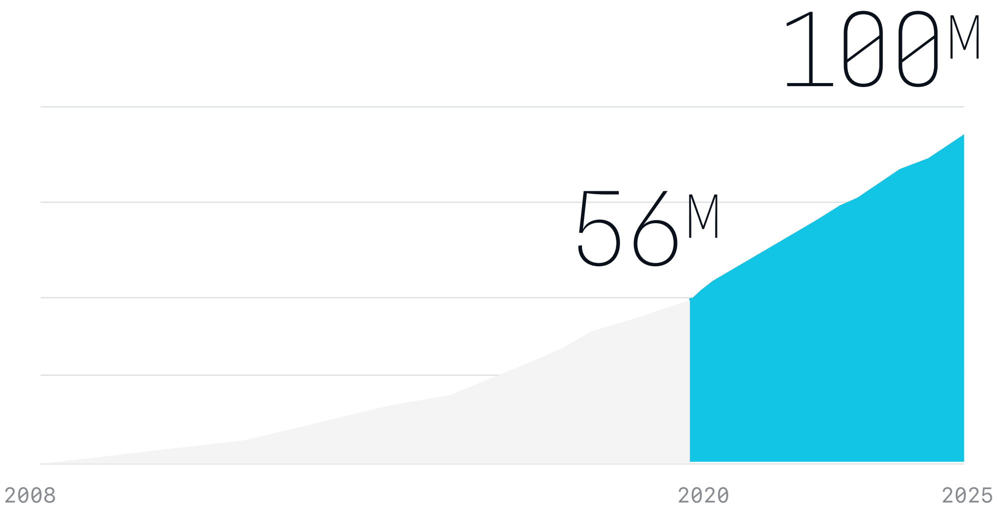
What does this tell us? It signals that people are turning to open source to create, learn, and share—particularly as much of the world has been forced to stay home. And while many caution that technology can be tiring—and this is true—these patterns show that open source may provide a creative outlet and a respite that is noticeably different from the workplace.

Year-over-year change in rate of open source project creation, seven-day rolling average



The future of open source and what it means

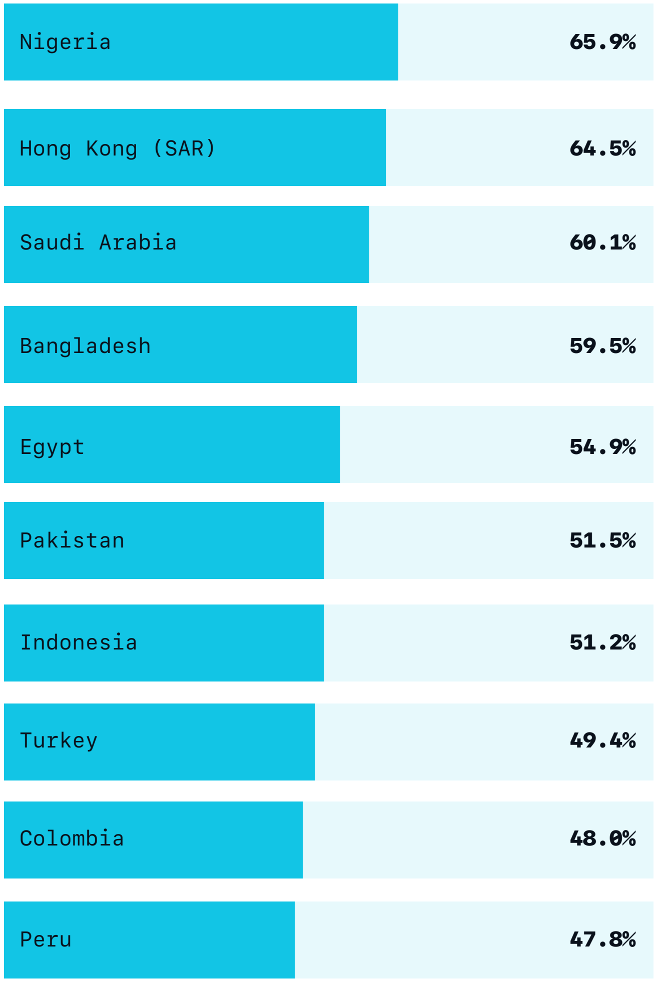
Today, GitHub has more than 56M developers, and we expect that to double to 100M by 2025. Each year, hundreds of thousands of people contribute to open source projects that power our software systems and global economy, touching industries from banking and healthcare to media and transportation.



As our earlier analysis highlighted, growth on GitHub is being driven by more than just developers. People from the education space are joining in increasing numbers, demonstrating the potential of GitHub as a platform to teach programming and collaborative development best practices. It also shows that teachers and professors have an extended sphere of influence, as they use open source to build curricula that reach students in remote locations, and participate in research with collaborative partners around the world. Those working in data as well as scientists and designers are also joining GitHub, suggesting that collaboration is increasingly more than just around code. We see this with people new to GitHub as well. As projects integrate data, images, and other file formats, there are more opportunities for new kinds of projects and ways to learn.

In 2020, open source contributors from the United States have dropped to 22.7%, with many coming from China (9.76%) and India (5.2%). And contributions are coming from a broader range of countries and regions as well, with the strongest growth in the past year from the countries and regions shown.

Countries and regions with strongest growth in contributors

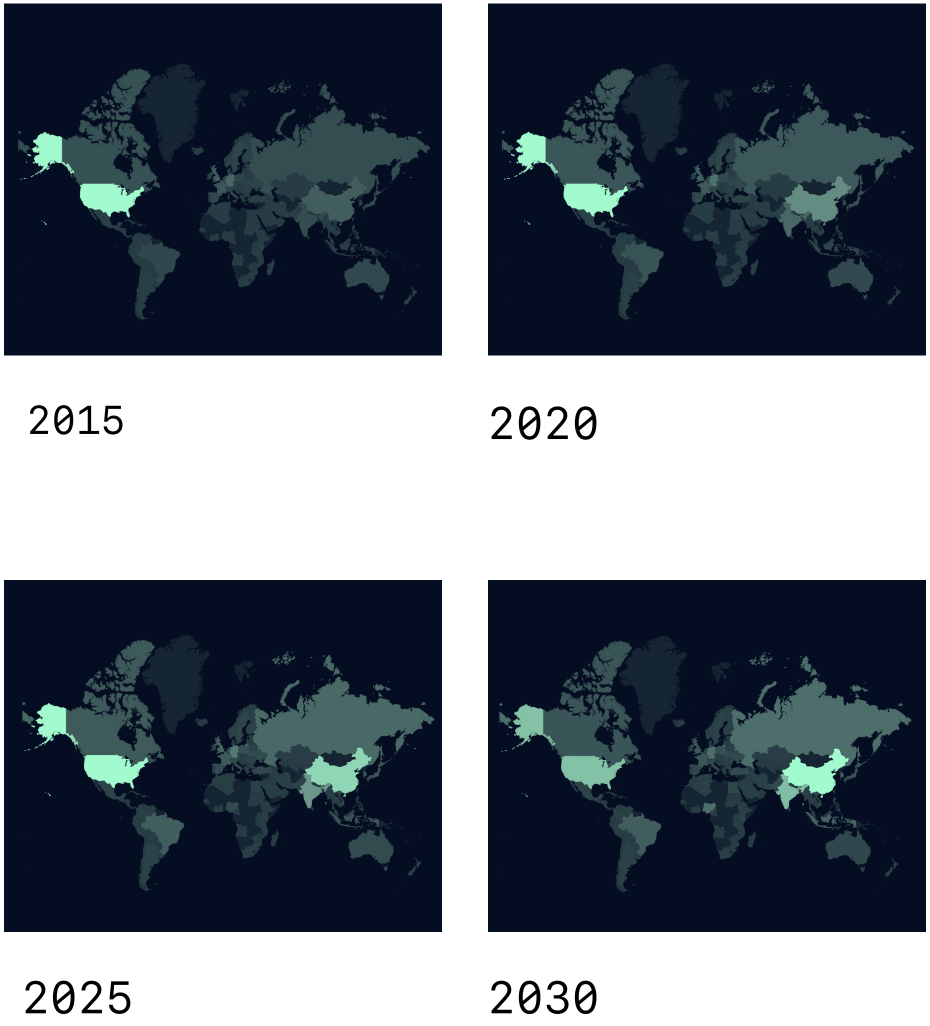


But where will these new open source contributors, who are so important to our technology and global economy, come from? Let's revisit our past and look at what the future holds.

In 2015, most open source contributors came from the United States (30.4%), with other strong contributions from Germany (7.3%) and the United Kingdom (5.8%).

As we look to the future of open source and reaching 100 million developers in 2025, we project open source contributions from the United States dropping to and stabilizing at 16.4%, with strong contributions from China (13.3%) and India (7.9%), and growth in South America and Africa, namely Brazil (3%) and Nigeria (1.5%).

Distribution of open source contributors by geographic location over time



2015

2020

2025

2030

These trends are particularly exciting because of the opportunity for open source development to truly impact lives around the world.

For more insights about how we work

Finding balance between work and play

In 2020 we all had to rethink our working spaces and schedules, testing the boundaries between work and home—and we saw that line can be hard to draw.

Finding balance

Productivity report →

Securing the world's software

By incorporating automation into workflows, communities are making it easier and faster to identify and remediate vulnerabilities.

Securing software

Security report →

There's a place for everyone in open source

Glossary

Throughout our report, you'll see a few key terms and phrases come up.

Dependency graph
A feature that lists all dependencies for a repository and helps identify known vulnerabilities.

Developers
Developers are individual user accounts on GitHub, regardless of their activity.

Issues
GitHub Issues are used to track ideas, enhancements, tasks, or bugs for work on GitHub.

Location
Country information for users is based on their last location, where known. For organizations, we take the best-known location information from the organization profile, or the most-common country organization members are active in. We only use location information in aggregate form to look at things like trends in growth in a particular country or region. We don't look at location information granularity finer than country level.

Newcomer
A user account created in the previous 28 days.

Open source projects
Open source projects are public repositories with an open source license.

Organizations
Organization accounts represent collections of people on GitHub. These can be paid or free, big or small, businesses or nonprofits.

Projects and repositories
We use projects and repositories interchangeably, although we understand that sometimes a larger project can span many repositories.

Veteran
An active user account created two or more years ago.

Methodology

People

We explored the different ways people use and interact with GitHub, based on their tenure on the site and broad categories into which they could be classified.

Tenure

We consider three categories for user tenure. A “newcomer” has been on the site 28 days or less. A "veteran” is someone who has been active for at least two years. The last category is “all users,” which is simply every person regardless of whether they are new or not. These categories enabled us to explore differences in use patterns based on experience.

Role

We also conducted analysis based on “role categories.” These categories identify the role as people describe themselves based on categorization of keywords in user-provided bios. As such, those with the role categories reported here are a subset of all people for a few different reasons. First, not everyone provides text in the bios of their profiles. Second, categorization is based on the presence of certain keywords within those bios. Third, our preliminary analysis was limited to English, meaning that bios in other languages are not considered. If people had several keywords present in their bios, they were attributed to multiple categories.

Role categories

This should not be considered an exhaustive list, but represents a starting point for future work. Nearly 3M unique people with bios were available for analysis, and 1.8M of those could be classified into at least one category. Based on this number, we assume the sample is representative, but there is potential for skew because our analysis was limited to English.

Education

People who indicate they are a student, professor, teacher, or associated with a college or university. Note that we do not explicitly identify those who might be in high school or below, however if one of the keywords appears in their bio, they would be included in this category.

Manager

People who identify as managers, whether they are program managers, project managers, entrepreneurs, etc.

Designer

People who indicate they work in areas such as UI/UX.

Developer

The majority of people on the site, this category ranges from those who identify as engineers to programmers to being affiliated with mobile, and everything in between.

Data

This category captures the growth area of data science, machine learning, and artificial intelligence.

Scientist

While not connected to “data science,” this category includes people who identify as working on scientific areas such as physics and astrophysics, biology, etc. as well as researchers.

Cryptocurrency

While a small group, this captures those working on a variety of currencies as well as blockchain technology.

Future projections

Our total developer forecast is based on an ensemble of autoregressive models that account for cyclical and seasonal effects.

For the projection of open source contribution, we use our total developer forecast to establish a baseline of global developer maximums. For each country-level projection we computed its past five-year compound annual growth rate and used this, controlling for our understanding of each market’s dynamic (whether it’s growing or stable), to project relative open source contribution over the next five years.

acknowledgements

Many thanks to our data scientists, contributors, and reviewers. Each is listed alphabetically by type of contribution.

- -

Authors: Nicole Forsgren

with contributions from
Greg Ceccarelli, Denae Ford, Vanessa Gennarell, Yu Huang, Tom Zimmerman

Data Scientists: Greg Ceccarelli, Derek Jedamski, Scot Kelly, Clair Sullivan

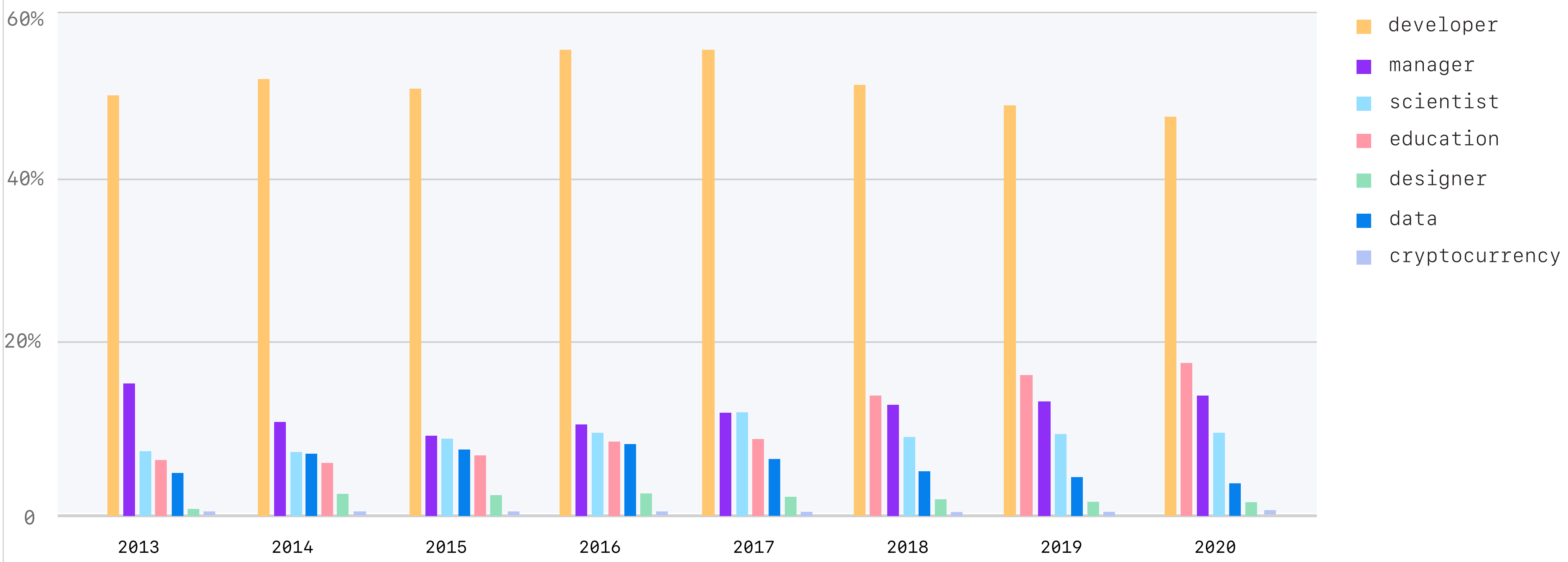
Reviewers: Bobby Dresser, Denae Ford, Nathen Harvey

Copyeditors: Leah Clark, Cheryl Coupé

Designers: Siobhán Doyle, Aja Shamblee

↑
TOC

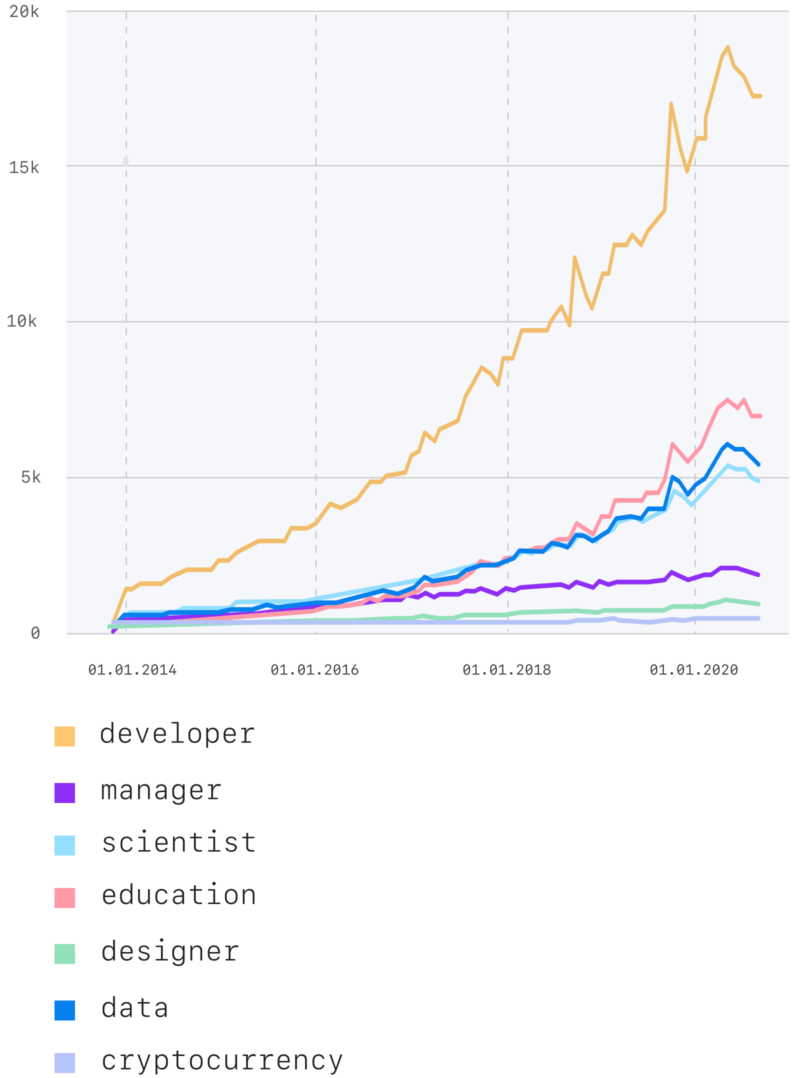
Distribution of roles over time in the Python community



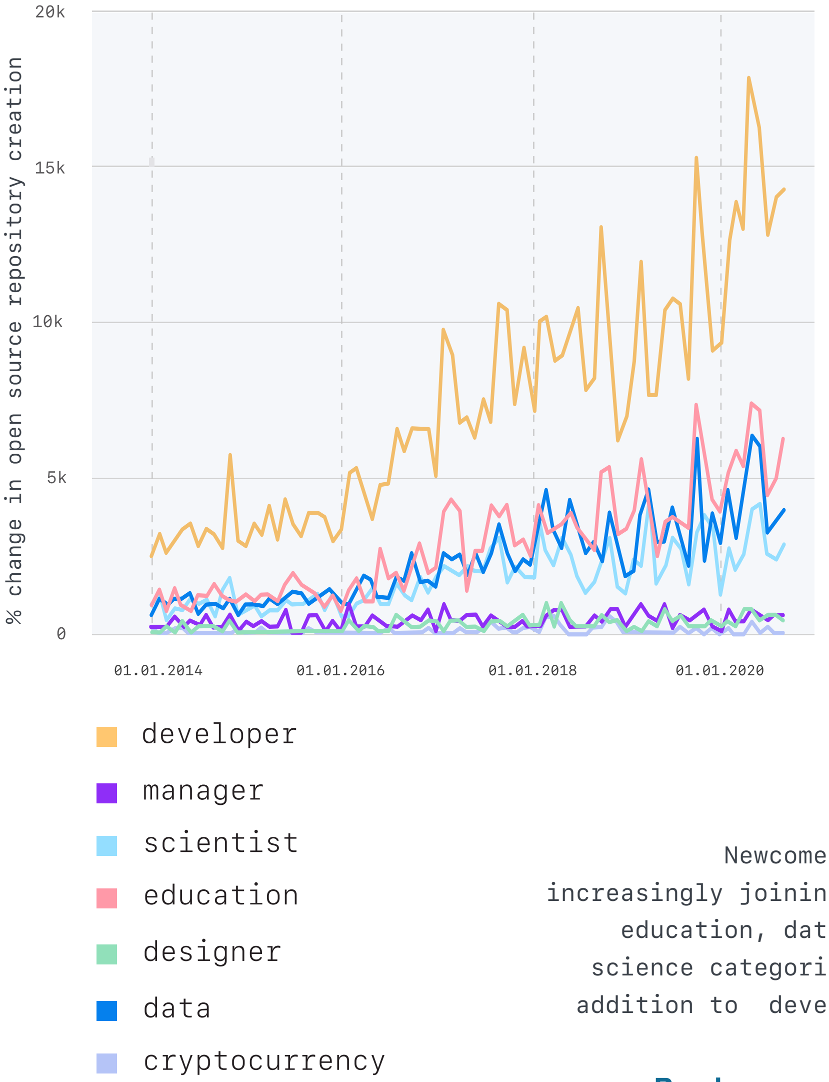
The increased diversity of users joining the Python community echoes trends in broader open source

Read more on p 15 →

Distribution of user categories for all users in the Python community



Distribution of user categories for newcomers in the Python community



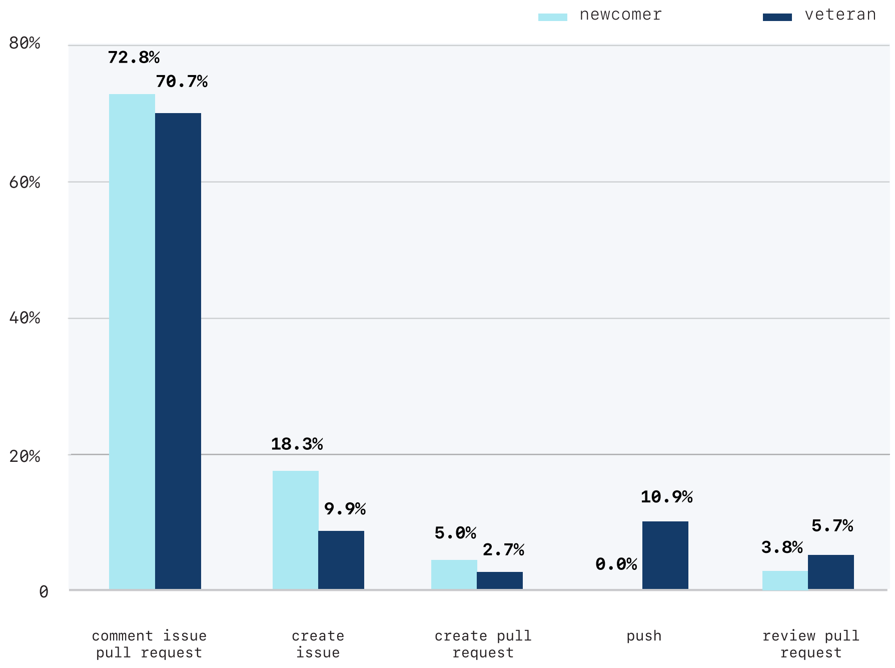
Newcomers are increasingly joining from education, data, and science categories, in addition to developers

Read more on p 15 →

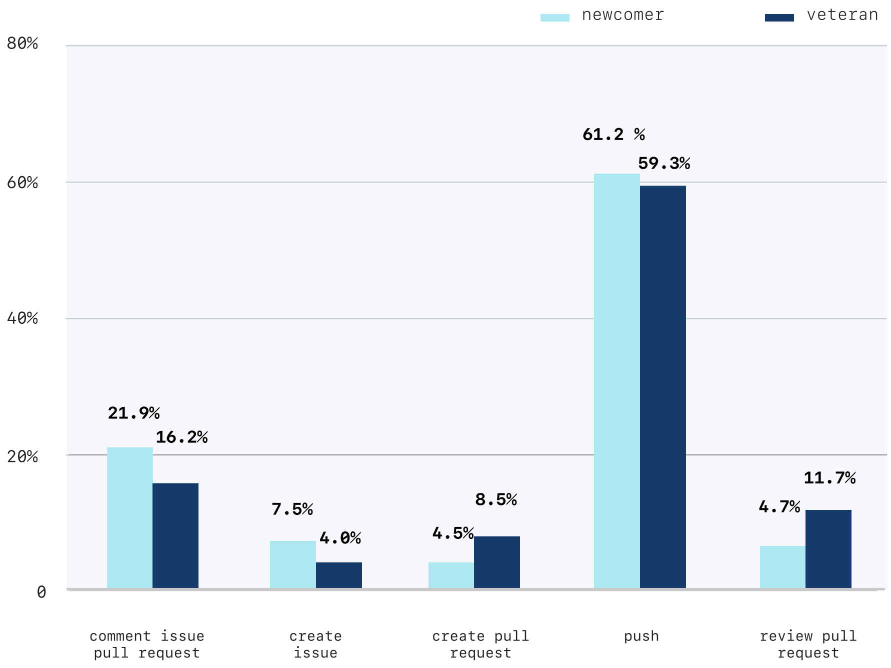
↑ TOC

//appendix

Distribution of action types for newcomers vs. veterans on top 10 TensorFlow repositories

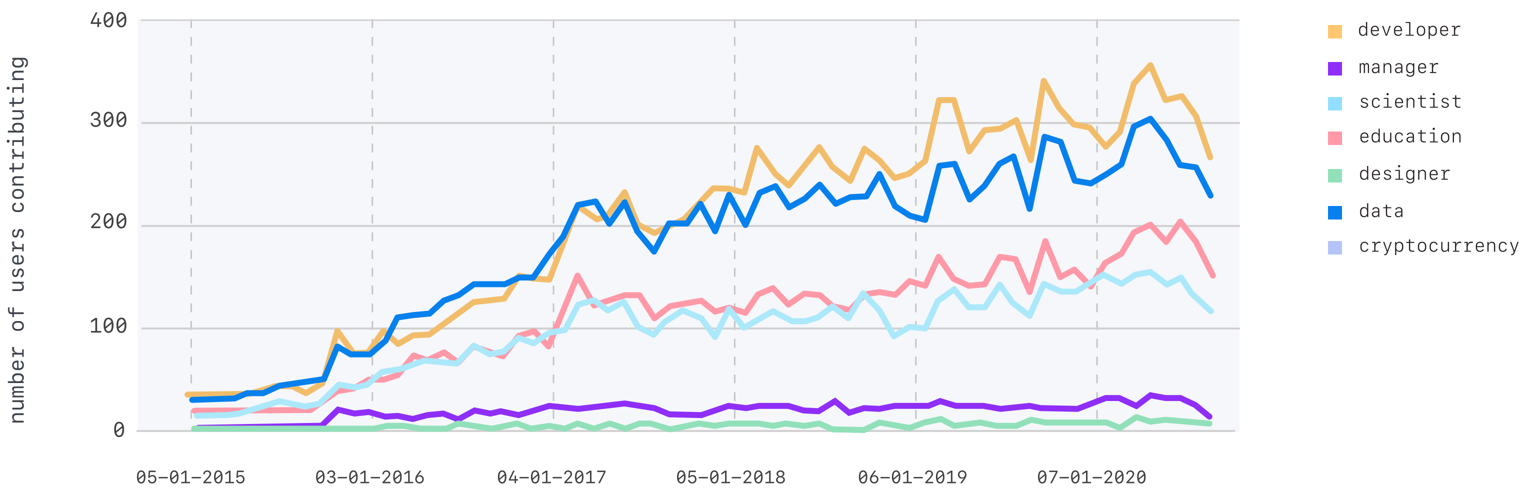


Distribution of action types for newcomers vs. veterans on the top 100 COVID repositories

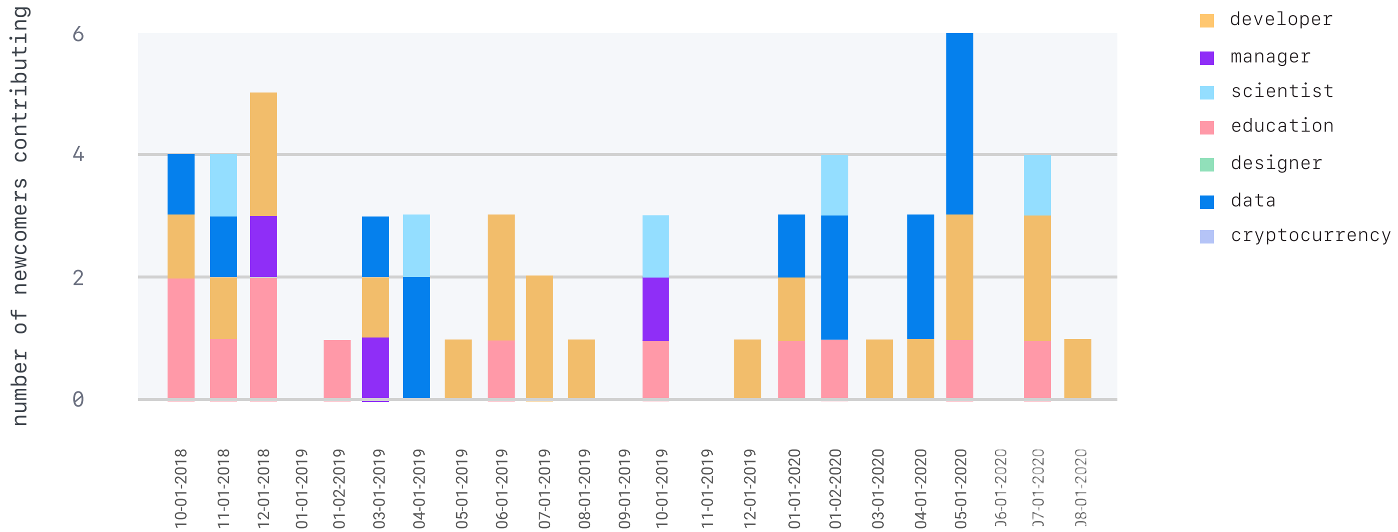


Read more on p 21 →

Distribution of users contributing by month in the TensorFlow community

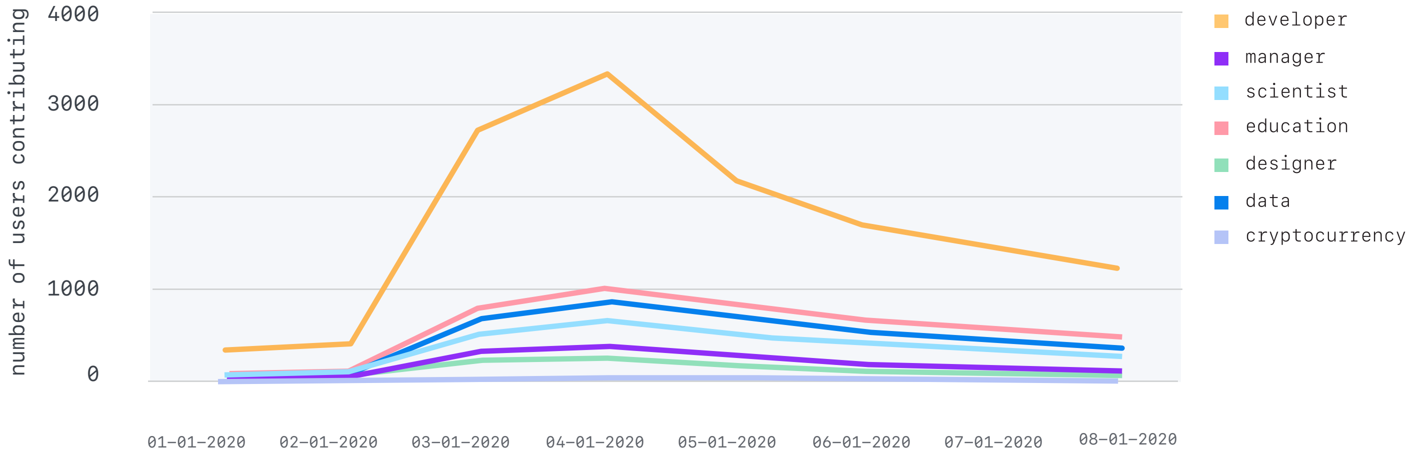


Distribution of newcomers contributing by month in the TensorFlow Community

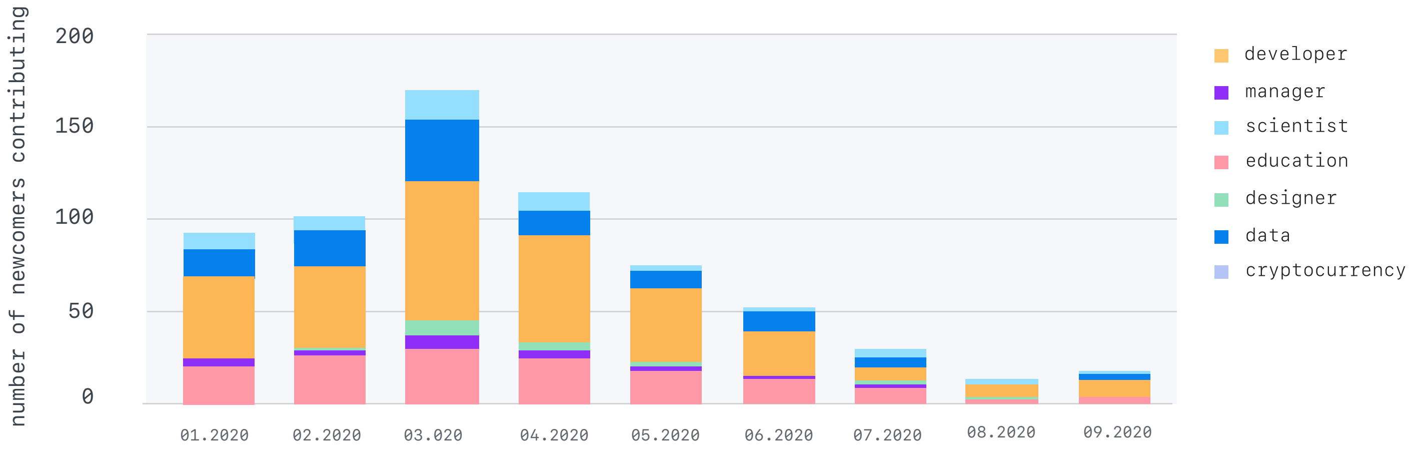


Read more on p 21 →

Distribution of users contributing by month in the COVID community



Distribution of new users contributing by month in the COVID community

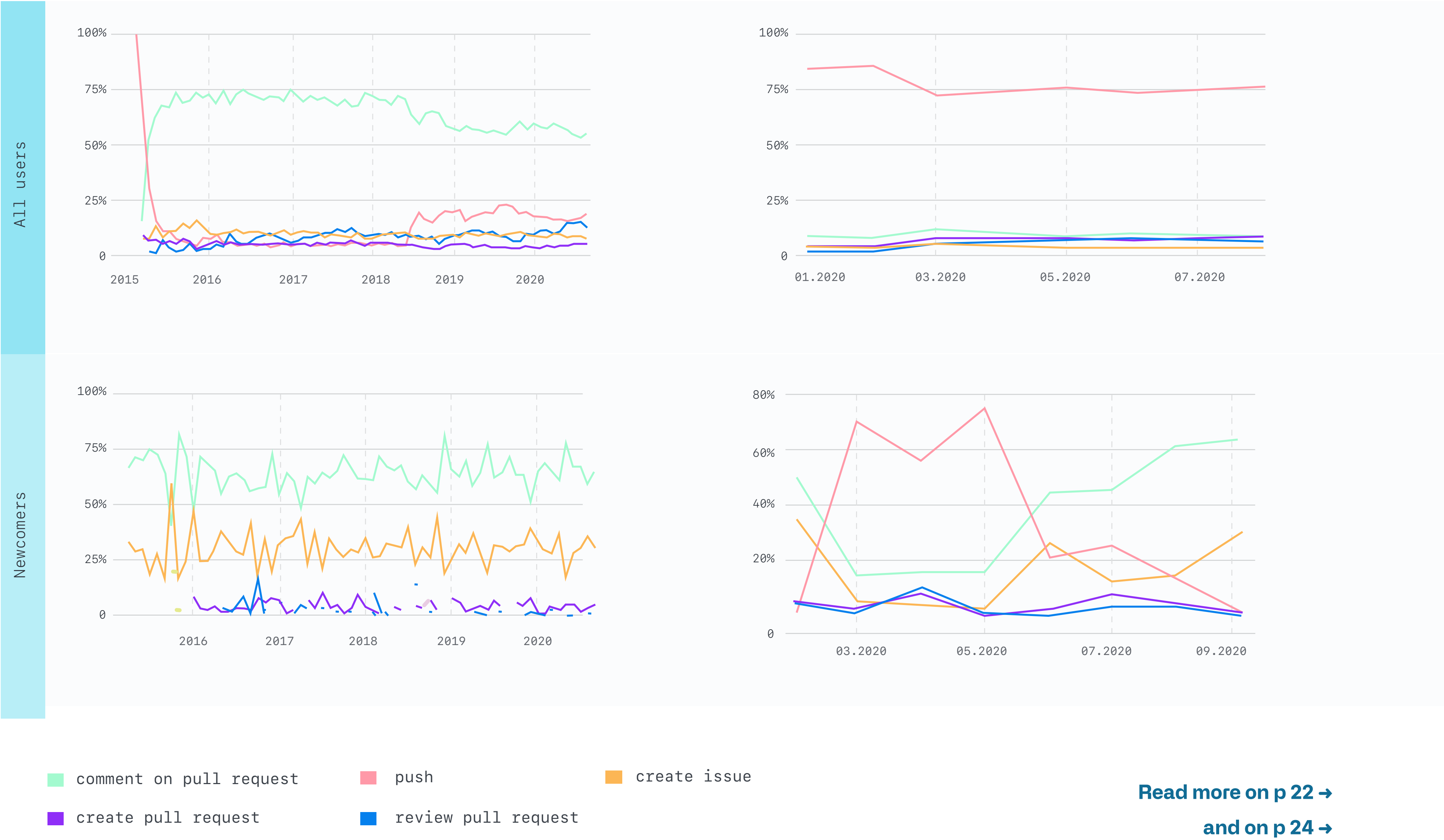


Read more on p 21 →

//appendix

Distribution of TensorFlow community actions per month

Distribution of COVID community actions per month



Read more on p 22 ➜

and on p 24 ➜