

ORACLE® + DATASCIENCE.COM

# Open Source Tools for Enterprise Data Science

An analysis of the open source trends driving change in the data science space using an interactive data exploration tool from Oracle's DataScience.com.



# Introduction

The field of data science has been steadily gaining a foothold in the corporate sector over the past decade, and is now an integral part of business strategy for some of the world's most successful companies. But as the scope of enterprise data science changes, so too have the tools data scientists are using to solve complex problems, from building models to identify and retain high-value customers to creating highly effective product recommendation engines.

Proprietary data science solutions, once the mainstay of enterprise data science, are being eclipsed by open source projects like R, Spark, and TensorFlow; in fact, 62% of analytics professionals now prefer R or Python to a legacy proprietary solution SAS.<sup>1</sup> While there are many reasons for this shift, a major one is that open source tools are available to anyone – and with that comes endless opportunities for collaboration and contribution. No longer are open source tools considered unreliable or limited; instead, they have been embraced by the data science community at large and built out to the point that they provide measurable value, even in an enterprise capacity.

Subsequently, database and data science software providers are jumping on the open source bandwagon instead of fighting its explosive growth. Case and point: The DataScience Cloud, our enterprise data science platform, allows data science teams of any size to work in Python-based notebooks, Apache Spark, R, and more, using their languages and machine learning libraries of choice.

To this end, DataScience has built an interactive tool using our DataScience Cloud platform for users of all technical backgrounds to explore the open source landscape quickly and easily. DataScience Trends leverages data from the GitHub Archive, made public last year through Google's BigQuery, to allow users to instantly visualize data from 2.8 million open source repositories without writing code. In this report, we use the tool to take stock of a few major open source players in the data science space: Google's TensorFlow and deep learning library Keras, visualization libraries matplotlib and ggplot, and permissive open source licenses.

---

<sup>1</sup> Burtch Works, "SAS, R, or Python Survey 2016: Which Tool Do Analytics Pros Prefer?," July 13, 2016  
<http://www.burtchworks.com/2016/07/13/sas-r-python-survey-2016-tool-analytics-pros-prefer/>

# Why Open Source Tools Are the Key to Beating Your Competition

For companies of any size, open source software adoption brings its own set of challenges. From licensing your own modified versions of open source tools, to creating an appropriately sized open source technology stack, there is no one way to integrate open source solutions into your existing data science workflow. But it's highly advantageous if you do so effectively.

It's easy to miss the mark: In many cases, tool sprawl – working across too many disjointed tools, the No. 1 business problem for data-driven companies<sup>2</sup> – can cripple your team's ability to deliver value. Too few open source tools might mean that you're leaning heavily on expensive, proprietary options to fill the gaps. But just the right amount can spell value for your organization.

## Commits:

A commit is a change to a file or set of files. Each commit in GitHub creates a unique ID, providing a record of how many different contributors are iterating on a project in a given time period.

## Stars:

GitHub users who star a repository are essentially bookmarking it and, in effect, showing appreciation to the creator of the repository for their work. These users aren't necessarily contributing to project.

## Pull requests:

A pull request is a method of submitting a contribution to open source project.

---

<sup>2</sup> Forrester Consulting, "Data Science Platforms Help Companies Turn Data Into Business Value," December 2016

# Deep Learning

## Google's TensorFlow Changes the Hierarchy of Deep Learning Libraries

In November 2015, Google open sourced its software library for machine learning, TensorFlow, kicking off a chain reaction across the deep learning space. While it's no surprise that TensorFlow has been wildly popular – the repository was starred 10,893 times within five days of its initial release – what was less expected was the ripple effect it had on other compatible tools.

One of those tools is Keras, a deep learning framework built by Google software engineer and artificial intelligence researcher François Chollet in March 2015 to provide “a set of ‘Lego blocks’ for building Deep Learning models in a fast and straightforward way.”<sup>5</sup> What Keras doesn't do is handle low-level tensor operations, so it has to sit on top of another solution. Initially, that solution was Theano, a Python library developed by a machine learning group at the Université de Montréal – but the release of TensorFlow changed all that.

theano



“When we started Keras in March 2015, Theano was the natural choice,” wrote Chollet in a December 2015 blog post announcing the creation of a TensorFlow backend for Keras. “...Since then, there has been a lot of innovation in the symbolic tensor computation space – a lot of it in the footsteps of Theano.

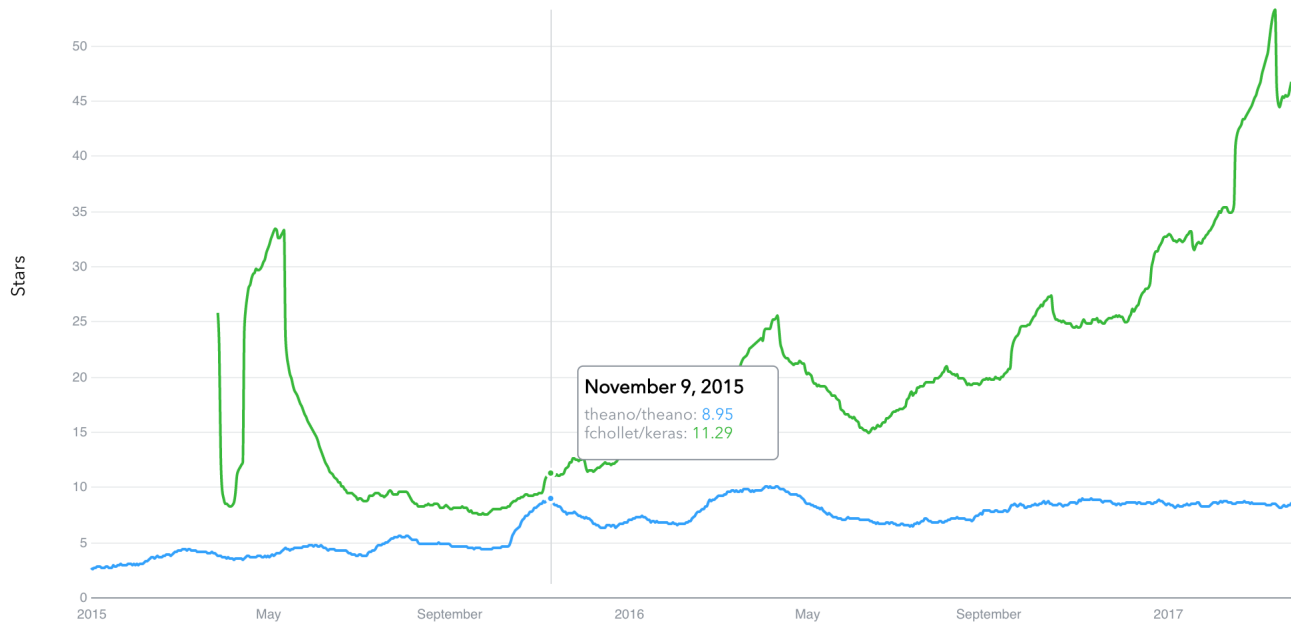
Most notably, we've seen two new frameworks appear, Neon from Nervana Systems and TensorFlow from Google. While Neon is the faster framework right now, TensorFlow has the engineering weight of Google behind it and there is no doubt that it will improve considerably over the next few months.”

---

<sup>5</sup> François Chollet, “Keras, now running on TensorFlow,” December 1, 2015 <https://blog.keras.io/keras-now-running-on-tensorflow.html>

Figure 1.

Popularity of Keras Eclipses Theano With the Release of TensorFlow



As seen in the chart above, Keras saw a major spike in interest just after its release. But with the introduction of TensorFlow approximately 10 months later, the number of stars has trended steadily upwards, significantly overtaking Theano in mid-2016. In fact, since the release of TensorFlow up until the end of March 2017, Keras has been starred 57,201 times; Theano, just 10,305.

Furthermore, TensorFlow is now being used in more than 8,000 open source repositories (an increase of nearly 2,000 repos since February 2017) <sup>6</sup>, considerably outpacing usage of Theano, which is currently being leveraged in approximately 1,500 repositories. That growth is mirrored in adoption of Keras, which now has more than 100,000 users. <sup>7</sup>

The rise of TensorFlow has irrevocably changed deep learning on an enterprise level, with companies like Snapchat, eBay, Airbnb, and Dropbox all building projects using Google’s robust machine learning library. TensorFlow will likely continue to grow its capabilities moving forward, making adoption a good move for companies working on machine learning projects. Its latest release improves the speed and flexibility of model building, in part through the introduction of a new module that is fully compatible with Keras.

### Why should companies consider pairing Keras with TensorFlow?

Keras ultimately makes adoption of TensorFlow easier by abstracting away many of the computations needed for creating neural networks, models used in applications like facial recognition and spam filtering. And its capabilities are making it easier for enterprise companies to work on complex machine learning projects.

“Keras has enabled new startups, made researchers more productive, simplified the workflows of engineers at large companies, and opened up deep learning to thousands of people with no prior machine learning experience,” Chollet wrote in a March 2017 blog post.

<sup>6</sup> Francois Chollet, “Keras, now running on TensorFlow,” December 1, 2015 <https://blog.keras.io/keras-now-running-on-tensorflow.html>

<sup>7</sup> Francois Chollet, “Introducing Keras 2,” March 14, 2017 <https://blog.keras.io/introducing-keras-2.html>

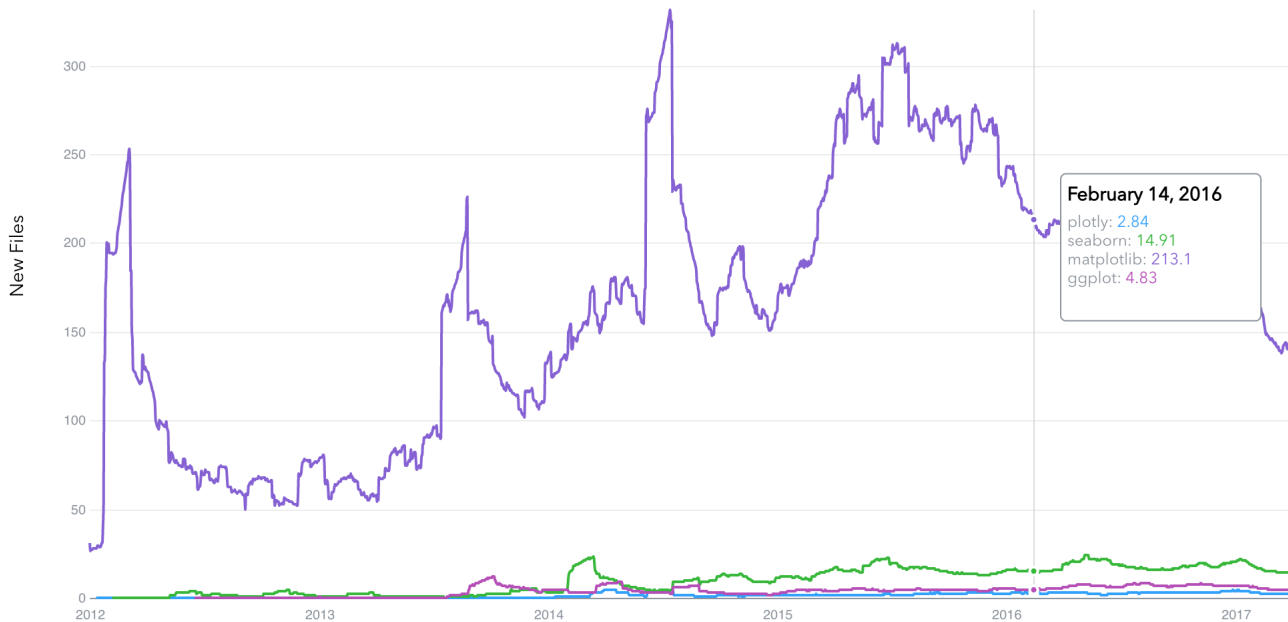
# Data Visualization

## Ggplot Gains Ground Against Data Visualization Giant Matplotlib

Data visualizations are meant to bring clarity to an analysis, making it easier for decision makers to identify trends or patterns in complex datasets. But visualizing data can quickly become costly – either because a company has opted to use an expensive subscription service like Tableau, or in terms of resources if different teams are needlessly using different visualization tools that require different skillsets.

Standardizing data visualization across your company with an open source tool is both efficient and cost effective. But choosing the right one is both a question of current functionality and the possibility of future development. For some time, the clear winner in this category has been matplotlib, a plotting library written in Python that was first released in 2003.

Figure 2. Matplotlib Consistently Dominates Data Visualization Space



In the chart above, you can clearly see that matplotlib dominates the open source data visualization space. Matplotlib is consistently more popular than Seaborn, Plotly, and ggplot based on the aggregated number of new files committed per day that mention each library, dating back to 2012. For the purposes of this analysis, file commits were attributed to a certain library if there was a mention of the library in the raw text contents of the file.

Matplotlib is both versatile and clunky; the library itself establishes very few design decisions by default, ultimately leaving it up to the user to spend extra time establishing how he or she wants certain plots to look. Other tools have tried to improve upon its features, including a few in this analysis: Seaborn was built on top of matplotlib and includes built-in themes that seek to enhance standard matplotlib plots, but Seaborn users often find themselves reverting to matplotlib commands while fine-tuning.

Similarly, ggplot was built on top of matplotlib when it was ported into Python. Originally written in R as ggplot2, ggplot enables the user to programmatically define a graph by concatenating high-level visualization components together, rather than requiring the user to repetitively specify low-level features such as axis ticks and marker sizes. But while ggplot's syntax is powerful, but it can be daunting to the inexperienced user. As a result, ggplot tends to have a steeper learning curve than other data visualization libraries.

Outside of the matplotlib realm is Plotly, an active innovator in the data visualization world in recent years. In addition to a plotting library built on top of d3.js and stack.gl (devoid of any matplotlib dependencies), Plotly's enterprise version boasts rich and interactive graphs, automated cloud storage, dashboard capabilities, and APIs spanning Python, R, Javascript, and beyond. Users of the open source version also have access to APIs that connect it to common data science languages like Python and R.

Despite Plotly's robust offerings, and the supposed improvements to matplotlib offered by Seaborn and ggplot, it would seem that matplotlib will continue to dominate the data visualization space. But the absolute number of files created per day that can be attributed to matplotlib only tells part of the story.

---

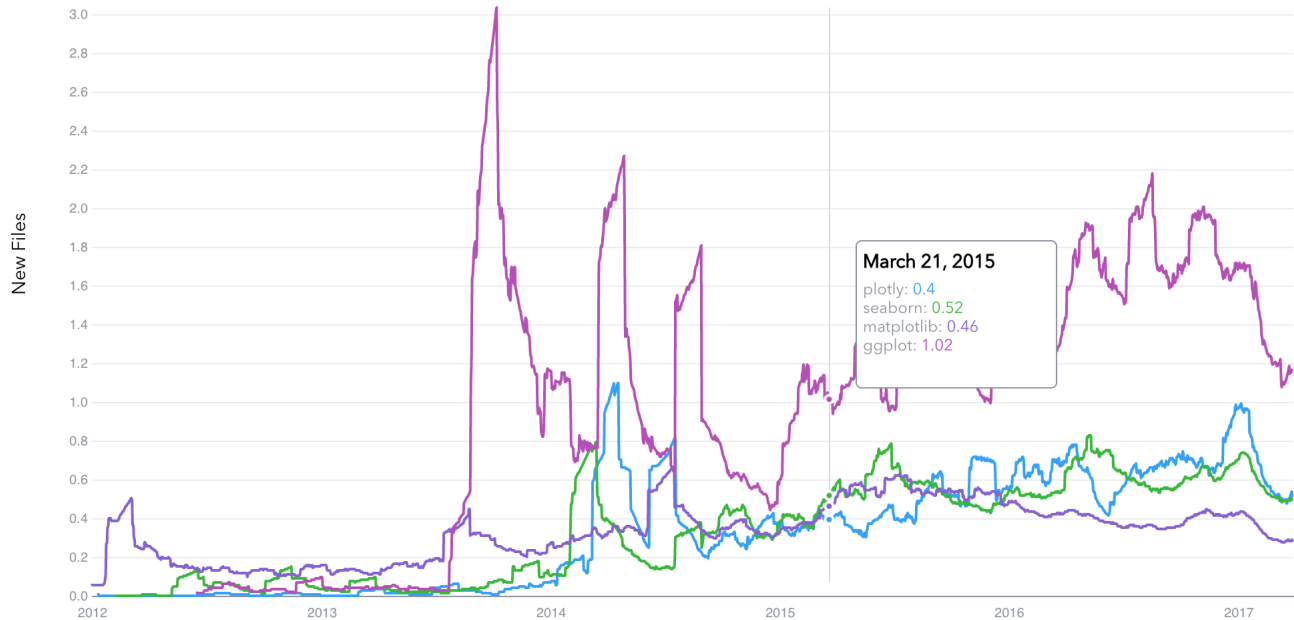
<sup>8</sup> "Ggplot from Yhat," <http://ggplot.yhathq.com/how-it-works.html>

<sup>9</sup> "What is plotly.js?," <https://plot.ly/javascript/>

<sup>10</sup> "Plotly API Libraries," <https://plot.ly/api/>

Figure 3.

## Matplotlib's Momentum Slows While Ggplot Contributions Pick Up



The normalized figure above, in which the rate that new files are being added to each library is shown relative to the library's average rate, clearly demonstrates that matplotlib's momentum is actually slowing down compared to its competitors.

Most strikingly, ggplot's adoption has been surging since mid-2013. At that time, Python users could only access ggplot via matplotlib's style sheet workaround.<sup>11</sup> Then in September 2014, Yhat released its first attempt at porting the popular R plotting library over to the Python community.<sup>12</sup> While the initial release was met with limited success, Yhat corrected several issues when it revamped the port in 2016.<sup>13</sup>

By plotting the frequency of new Python files on GitHub containing "ggplot," we can bring this timeline to life: the introduction of the matplotlib style sheet in 2013, the initial spike in popularity and subsequent disappointment with Yhat's first port in 2014, and the final sustained success of Yhat's revised port in 2016. This emerging trend indicates that ggplot's high-level visualization grammar is gaining acceptance in the open-source community and changing the way practitioners approach data visualization.

<sup>11</sup> "Customizing plots with style sheets," [http://matplotlib.org/users/style\\_sheets.html](http://matplotlib.org/users/style_sheets.html)

<sup>12</sup> "Ggplot for Python," <https://pypi.python.org/pypi/ggplot>

<sup>13</sup> "A new ggplot is here," <http://blog.yhat.com/posts/new-ggplot.html>



# Software Licensing

## MIT Remains License of Choice For Open Source Projects

From Airbnb's Airflow, an open source workflow management platform built in Python, to Stitch Fix's public collection of projects written in Ruby, Python, and Javascript, the open source space is seeing an influx of contributions from both major and up-and-coming companies. In fact, 67% of companies actively encourage developers to contribute to open source projects.<sup>14</sup>

Companies choose to open source for a number of reasons: recruitment, media exposure, and improvement through crowd sourcing. But whatever the reason, it's imperative that they license open source projects appropriately to prevent legal issues or bar would-be competitors from selling modified versions. Google's more than 2,000 open source projects only use code under certain licenses and the company requires contributor license agreements for all the patches it receives.<sup>15</sup>

There is a wide array of open source licensing options, but some of the most popular are Apache, MIT, and the GNU General Public License (GPL). Apache and MIT are on one side of the spectrum – both Apache and MIT allow anyone to use your code, while offering some level of protection for you as the creator; Apache offers a patent license and a retaliation clause, while MIT simply waives your liability. GPL is a "copyleft" license, meaning any improvements made to your code by outside parties will need to be open sourced as well.

The increased protections in the Apache license has made it attractive to companies like Airbnb, which opted to open source Airflow under the license.<sup>16</sup> The Apache license has also successfully garnered recognition for software developers like the creators of etherpad, an open source online editor that became the basis of Hackpad. Hackpad was acquired by Dropbox in 2014, and its code was open sourced under the Apache license – essentially, code from etherpad is now integral to the workflow of every user on Dropbox Paper.

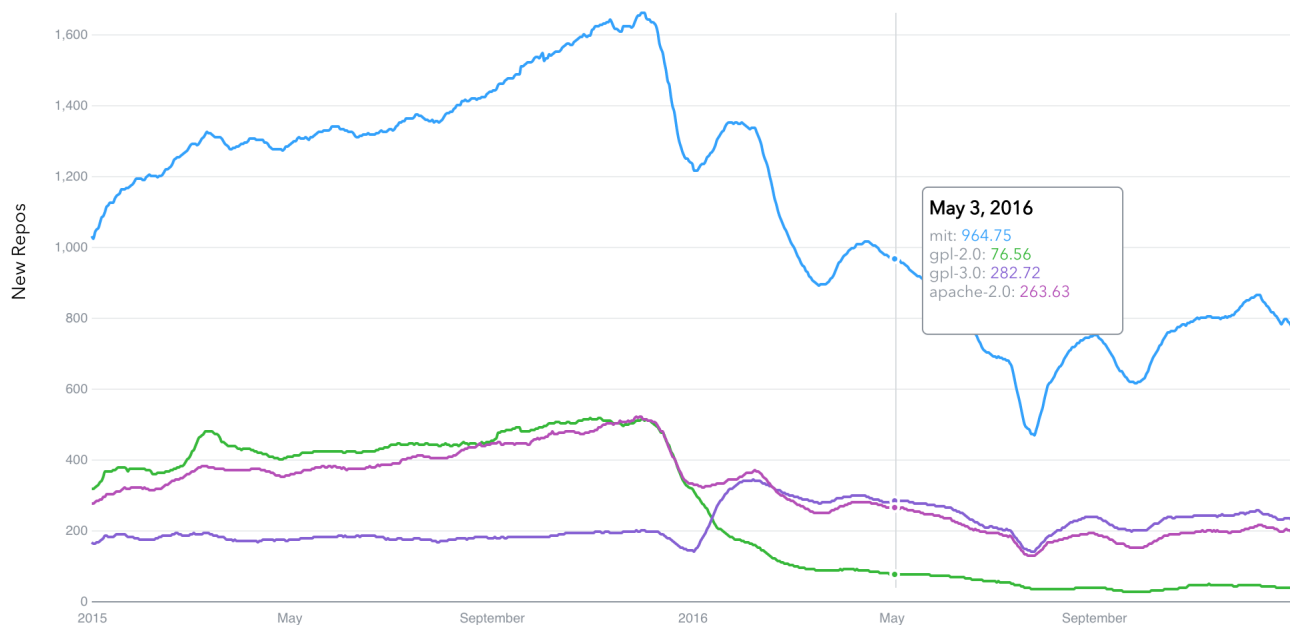
---

<sup>14</sup> "The Tenth Annual Future of Open Source Survey," <https://www.blackducksoftware.com/2016-future-of-open-source>

<sup>15</sup> "Noto Serif CJK is here!," <https://opensource.googleblog.com/>

<sup>16</sup> Maxime Beauchemin, "Airflow: a workflow management platform," June 1, 2015 <http://nerds.airbnb.com/airflow/>

Figure 4. MIT is License of Choice for Protecting Open Source Projects



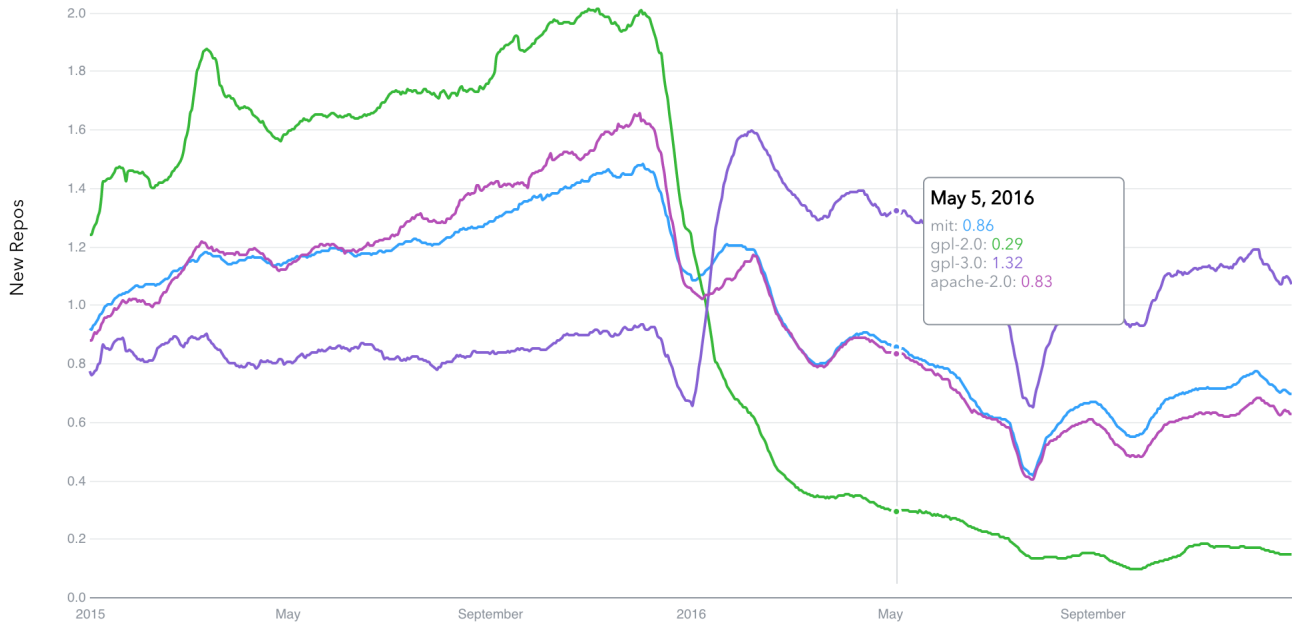
Even so, as seen in the chart above, the MIT license is far and away the dominate license – no doubt owing to its permissiveness and simplicity. GitHub also came to this conclusion in 2015 <sup>17</sup> when it found that 44.69% of licensed projects in its archive used MIT. To update that finding, we queried the number of new public GitHub repositories created per day for each license type since then.

However, it appears that the popularity of the MIT and Apache licenses is beginning to wane. Instead, GPLv3, the third version of the copyleft license published in 2007 <sup>18</sup>, is the only license on the chart trending upward.

<sup>17</sup> Ben Balter, "Open source license usage on GitHub.com," March 9, 2015 <https://github.com/blog/1964-license-usage-on-github-com>

<sup>18</sup> "GNU General Public License," June 29, 2007 <https://www.gnu.org/licenses/gpl-3.0.html>

Figure 5. 'Copyleft' GPL License Sees Growth in 2016



To investigate this growing trend, we normalized each time series by dividing by its mean value. The resulting graph above reveals an interesting reversal: The first quarter of 2016 saw a 60% spike in new repos using the GPLv3 license (relative to its mean), eventually setting around a 10% bump above its mean by the end of 2016. On the other hand, MIT and Apache popularity depreciated about 20-40% by the end of 2016.

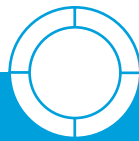
GPL's strong copyleft policy can be a turnoff for corporations that seek to profit from code based on GPL-licensed projects, but GPL can also foster community amongst open-source enthusiasts. Although corporate wariness of GPL licenses has discouraged GPL-licensed authorship in the past, the trend highlighted above may represent the first inklings of GPL's future mainstream adoption across the open-source community and at an enterprise level.

# The Future is Open Source

Open source software is steadily gaining support across every industry, with established companies like Facebook, Microsoft, and General Electric pouring money and resources into public-facing projects. And as corporations increasingly rely on data science to get value from their big data, so too will they embrace the open source tools that primarily make up the artificial intelligence, Internet of Things, and data infrastructure space.

The challenge is identifying which of those tools is relevant and valuable to your business. GitHub is adding millions of projects every year; in fact, while the first million repositories were created in just under four years, the million added by the end of 2013 took just 48 days.<sup>19</sup> Assessing the maturity of these projects, grappling with any licensing issues, and making sure your team has the correct skillset to use them are challenges that many companies are now facing.

Understanding the trends in open source software contribution and usage will go a long way in creating a tech stack that makes sense for the data scientists at your organization. And that's why we've made it easy to view data related to GitHub's most popular and well-loved repositories. You can try it out for yourself at [www.datascience.com/trends](http://www.datascience.com/trends).



## About The DataScience Trends Tool

DataScience Trends is an interactive tool that allows users of every technical level to explore and visualize trends in open source software. DataScience Trends sits on top of more than three terabytes of GitHub data, and features a sleek UI that makes it easy to create and share visualizations of activity across 2.8 million repositories – without ever writing a line of code.

To learn more about DataScience Trends, or to try it for yourself, visit [www.datascience.com](http://www.datascience.com).

---

<sup>19</sup> Brian Doll, "10 Million Repositories," December 23, 2013 <https://github.com/blog/1724-10-million-repositories>

ORACLE® + DATASCIENCE.COM

Connect with us on social media:

 @DataScienceInc

 <https://www.linkedin.com/company/datascienceinc>

 @DataScienceInc

 <https://www.facebook.com/datascience>

Oracle Corporation, World Headquarters 500 Oracle Parkway Redwood Shores, CA 94065 USA

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.