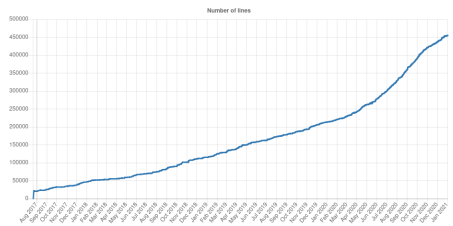
The Lean community in 2020

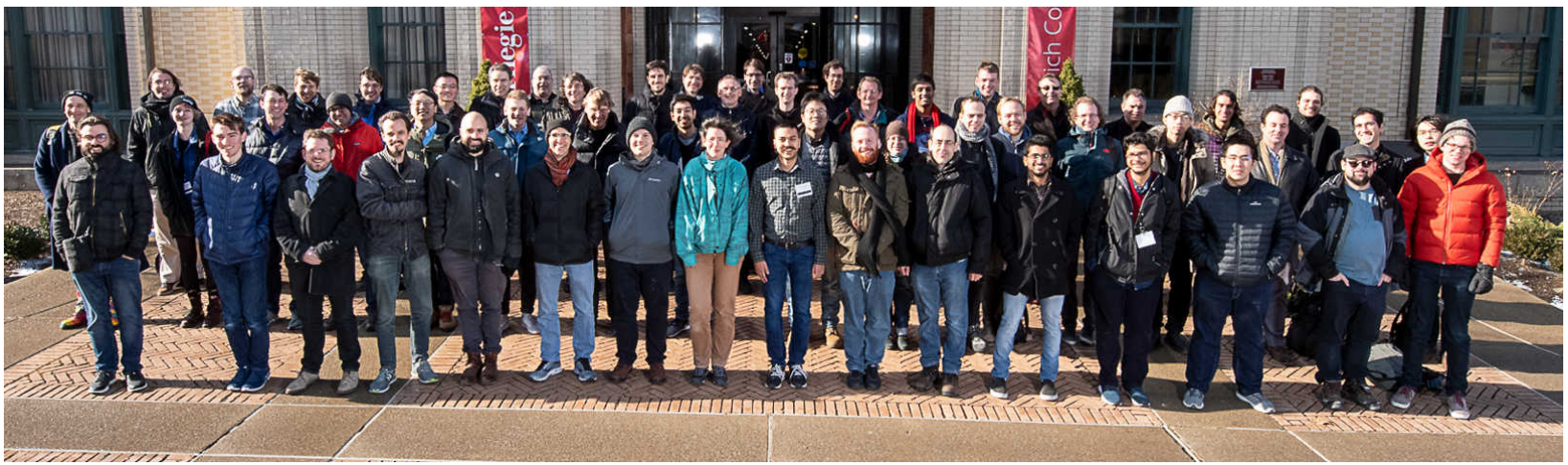
Robert Y. Lewis and Patrick Massot

Vrije Universiteit Amsterdam and Université Paris-Saclay at Orsay



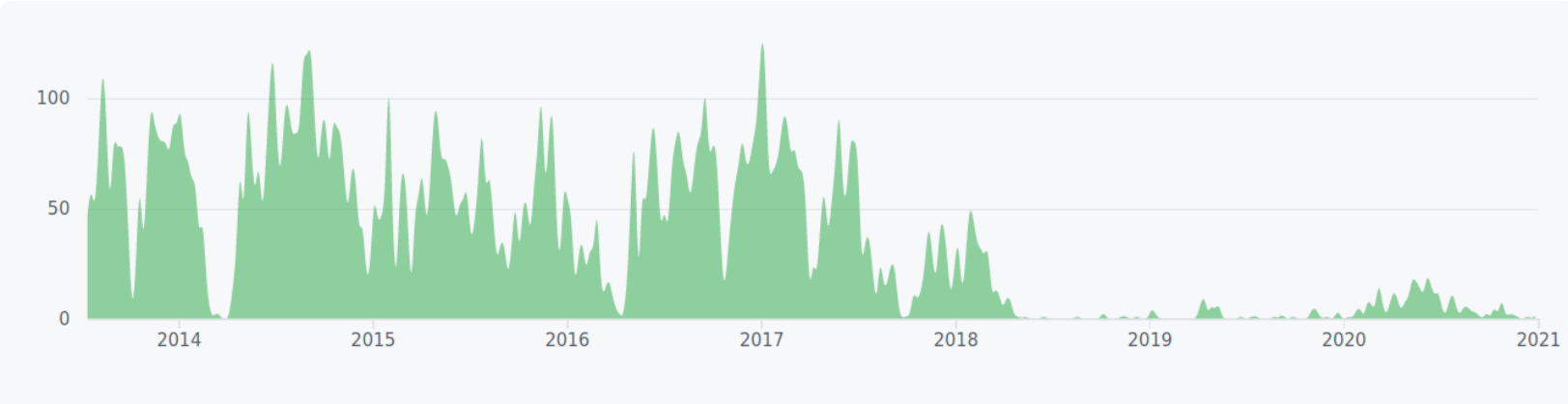
January 5, 2021

Lean together 2020



Carnegie Mellon University, in Pittsburgh, Pennsylvania.
90 participants.

Lean 3 community edition

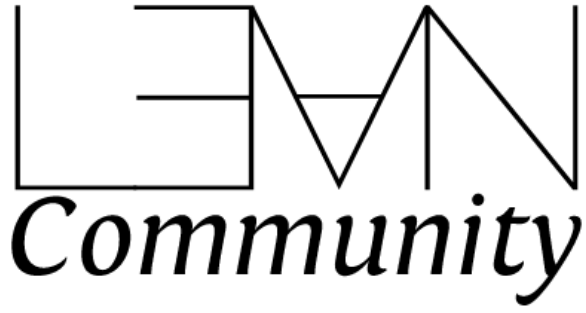


leanproject

```
pmassot@massot-orsay:~$ leanproject get mathlib
Cloning from git@github.com:leanprover-community/mathlib.git
configuring mathlib 0.1
Looking for local mathlib oleans
Looking for remote mathlib oleans
Trying to download https://oleanstorage.azureedge.net/mathlib/0837fc30f425b2f777
addbf5a80a96bf97220a67.tar.xz to /home/pmassot/.mathlib/0837fc30f425b2f777addbf5
a80a96bf97220a67.tar.xz
100%|                                                | 33.6M/33.6M [00:15<00:00, 2.20MiB/s]
Found mathlib oleans at https://oleanstorage.azureedge.net/mathlib/
pmassot@massot-orsay:~$
```

olean files are hosted on Azure through a Microsoft Research grant

Community website



Lean and its Mathematical Library

The Lean theorem prover is a proof assistant developed principally by Leonardo de Moura at Microsoft Research.

The Lean mathematical library, *mathlib*, is a community-driven effort to build a unified library of mathematics formalized in the Lean proof assistant. The library also contains definitions useful for programming. This project is very active, with many regular contributors and daily activity.

The contents, design, and community organization of mathlib are described in the paper The Lean mathematical library, which appeared at CPP 2020. You can get a bird's eye view of what is in the library by reading the library overview. You can also have a look at our repository statistics to see how it grows and who contributes to it.

Try it!

You can try Lean in your web browser, install it in an isolated folder, or go for the full install. Lean is free, open source software. It works on Linux, Windows, and MacOS.

- Try the online version of Lean
- Installation instructions
- Working on Lean projects

Learn to Lean!

You can learn by playing a game, following tutorials, or reading books.

- Learning resources
- Theorem Proving in Lean (an introduction)
- API documentation of mathlib

Meet the community!

Lean has very diverse and active community. It gathers mostly on a Zulip chat and on GitHub. You can get involved and join the fun!

- Meet us
- How to contribute
- Papers involving Lean

Learning resources

Learning Lean

There are many ways to start learning Lean, depending on your background and taste. They are all fun and rewarding, but also difficult and occasionally frustrating. Proof assistants are still difficult to use, and you cannot expect to become proficient after one afternoon of learning.

Hands-on approaches

- Whatever your background, if you want to dive right away, you can play the Natural Number Game by Kevin Buzzard and Mohammad Pedramfar. This is a online interactive tutorial to Lean focused on proving properties of the elementary operations on natural numbers.
- For a faster paced and broader dive, you can get the tutorials project. (You already have it if you installed an autonomous bundle or followed the instructions on this page.) This tutorial is specifically geared towards mathematics rather than computer science. The last files of this project are easier if you have already encountered the definition of limits of sequences of real numbers.
- The lfctm2020 exercises, developed for the July 2020 virtual meeting Lean for the Curious Mathematician, are another good resource. There are corresponding tutorial videos from the meeting.
- A brand new resource that is still under construction is *Mathematics in Lean*. It can be read online, or downloaded as a pdf, but it is really meant to be used in VSCode, doing exercises on the fly (see the instructions). It currently covers roughly the same ground as the tutorials project.
- Once you know the basics, you can also learn by solving Lean puzzles on Codewars.

Whatever resource you choose to use from the above list, it could be useful to have a copy of our tactic cheat sheet at hand, for reference.

Textbooks

- If you prefer reading a book (with exercises), the standard reference is Theorem Proving in Lean. You almost certainly want to read it at some point anyway, since it explains foundational things much better than any hands-on tutorial could do.
- If you are very new to the concept of logic and proofs, you can read Logic and Proof, a textbook that is a first rigorous proving course that teaches Lean at the same time.
- If you have a computer science background, and are primarily interested in software verification, then you can read The Hitchhiker's Guide to Logical Verification (pdf) (tablet edition optimized for on-screen viewing), course notes for an MSc-level course at VU Amsterdam.
- If you want a systematic exposition of syntax and commands, then you can read the reference manual.

Mathlib overview

Analysis

Normed vector spaces normed vector space over a normed field, topology on a normed vector space, equivalence of norms in finite dimension, finite dimensional normed spaces over complete normed fields are complete, Heine-Borel theorem (finite dimensional normed spaces are proper), continuous linear maps, norm of a continuous linear map, Banach open mapping theorem, absolutely convergent series in Banach spaces, Hahn-Banach theorem, dual of a normed space, Fréchet-Riesz representation of the dual of a Hilbert space, isometric inclusion in double dual, completeness of spaces of bounded continuous functions.

Differentiability differentiable functions between normed vector spaces, derivative of a composition of functions, derivative of the inverse of a function, Rolle's theorem, mean value theorem, C^k functions, Leibniz formula, local extrema, inverse function theorem, implicit function theorem, analytic function.

Convexity convex functions, characterization of convexity, Jensen's inequality (finite sum version), Jensen's inequality (integral version), convexity inequalities, Carathéodory's theorem.

Special functions logarithms, exponential, trigonometric functions, inverse trigonometric functions, hyperbolic trigonometric functions, inverse hyperbolic trigonometric functions.

Measures and integral calculus sigma-algebras, measurable functions, the category of measurable space, Borel sigma-algebras, positive measure, Lebesgue measure, Giry monad, integral of positive measurable functions, monotone convergence theorem, Fatou's lemma, vector-valued integrable functions (Bochner integral), L^1 space, Bochner integral, dominated convergence theorem, fundamental theorem of calculus, part 1, Fubini's theorem.

Geometry

Affine and Euclidean geometry affine spaces, affine functions, affine subspaces, barycenters, affine spans, Euclidean affine space, angles of vectors.

Differentiable manifolds smooth manifold (with boundary and corners), smooth map between manifolds, tangent bundle, tangent map, Lie group.

Algebraic geometry prime spectrum, Zariski topology, locally ringed space, scheme.

Mathlib documentation

```
def implicit_function_data.implicit_function {𝕜 : Type u_1} [nondiscrete_normed_field 𝕜]
      {E : Type u_2} [normed_group E] [normed_space 𝕜 E] [complete_space E] {F : Type u_3}
      [normed_group F] [normed_space 𝕜 F] [complete_space F] {G : Type u_4} [normed_group G]
      [normed_space 𝕜 G] [complete_space G] (φ : implicit_function_data 𝕜 E F G) :
  F → G → E
```

source

Implicit function theorem. If `f : E → F` and `g : E → G` are two maps strictly differentiable at `a`, their derivatives `f'`, `g'` are surjective, and the kernels of these derivatives are complementary subspaces of `E`, then `implicit_function_of_is_compl_ker` is the unique (germ of a) map `φ : F → G → E` such that `f (φ y z) = y` and `g (φ y z) = z`.

▸ Equations

Project tracking

Lean projects

While much Lean development takes place in the mathlib repository, there are many other projects using Lean that are developed and maintained by members of the community. We list here a selection. Many of these projects are designed to be imported as dependencies in other developments. At the bottom of this page, you can see a summary of which projects are compatible with which Lean versions. If two projects both support the same Lean version, you can likely use them together.

To add a project to this list, please see the directions at the leanprover-contrib repository.

This list and the repository that manages it are both works in progress. Please add your own project and report any problems in that repository.

lean-perfectoid-spaces

maintained by @kbuzzard @jcommelin @PatrickMassot

Perfectoid spaces are sophisticated objects in arithmetic geometry introduced by Peter Scholze in 2012. We formalised enough definitions and theorems in topology, algebra and geometry to define perfectoid spaces in Lean. See also our project webpage and paper.

View on GitHub (★ 80)

lftcm2020

maintained by @jcommelin @PatrickMassot

This repository contains tutorials about Lean and mathlib that were developed for the workshop Lean for the Curious Mathematician, held in July 2020. The tutorials range from introductory lessons on numbers, logic, and sets to advanced lessons on category theory and manifolds. In addition to the materials found in this repository, we recommend watching the videos of the tutorials and lectures from the workshop.

View on GitHub (★ 33)

topos

maintained by @b-mehta

sudoku

maintained by @TwoFx

Automatic upgrade reports

Automatic upgrade has failed #89

Closed **github-actions** bot opened this issue on Oct 6, 2020 · 1 comment

github-actions bot commented on Oct 6, 2020

Oh no! We have failed to automatically upgrade your project to the latest versions of Lean and its dependencies.

If your project currently builds, this is probably because of changes made in its dependencies:

- mathlib: changes

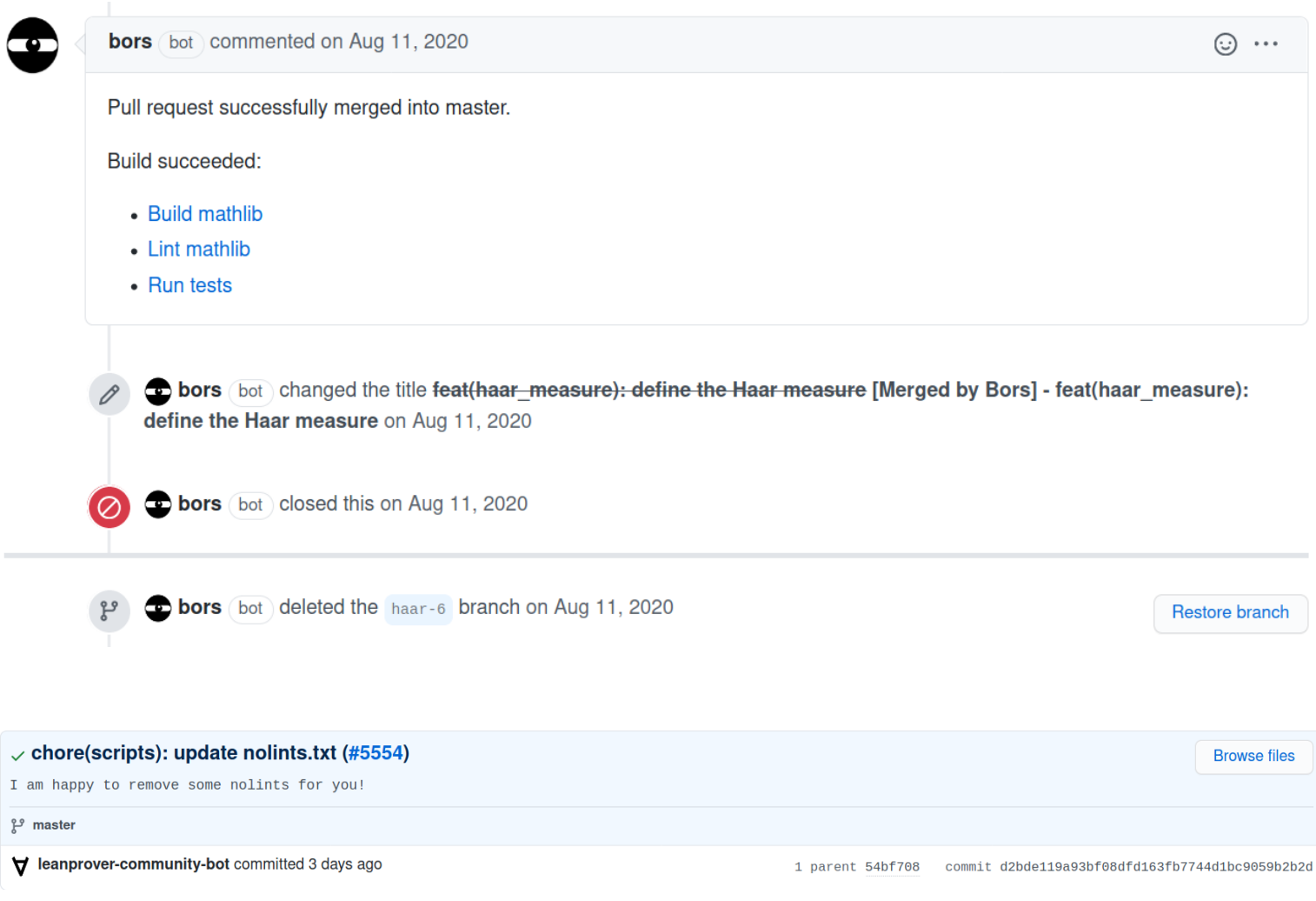
You can see the errors by running:

```
leanproject up
leanproject build
```

github-actions bot commented on Oct 7, 2020 Author

This issue has been resolved!

Continuous integration improvements



bors bot commented on Aug 11, 2020

Pull request successfully merged into master.

Build succeeded:

- Build mathlib
- Lint mathlib
- Run tests

bors bot changed the title ~~**feat(haar_measure): define the Haar measure**~~ **[Merged by Bors] - feat(haar_measure): define the Haar measure** on Aug 11, 2020

bors bot closed this on Aug 11, 2020

bors bot deleted the `haar-6` branch on Aug 11, 2020 Restore branch

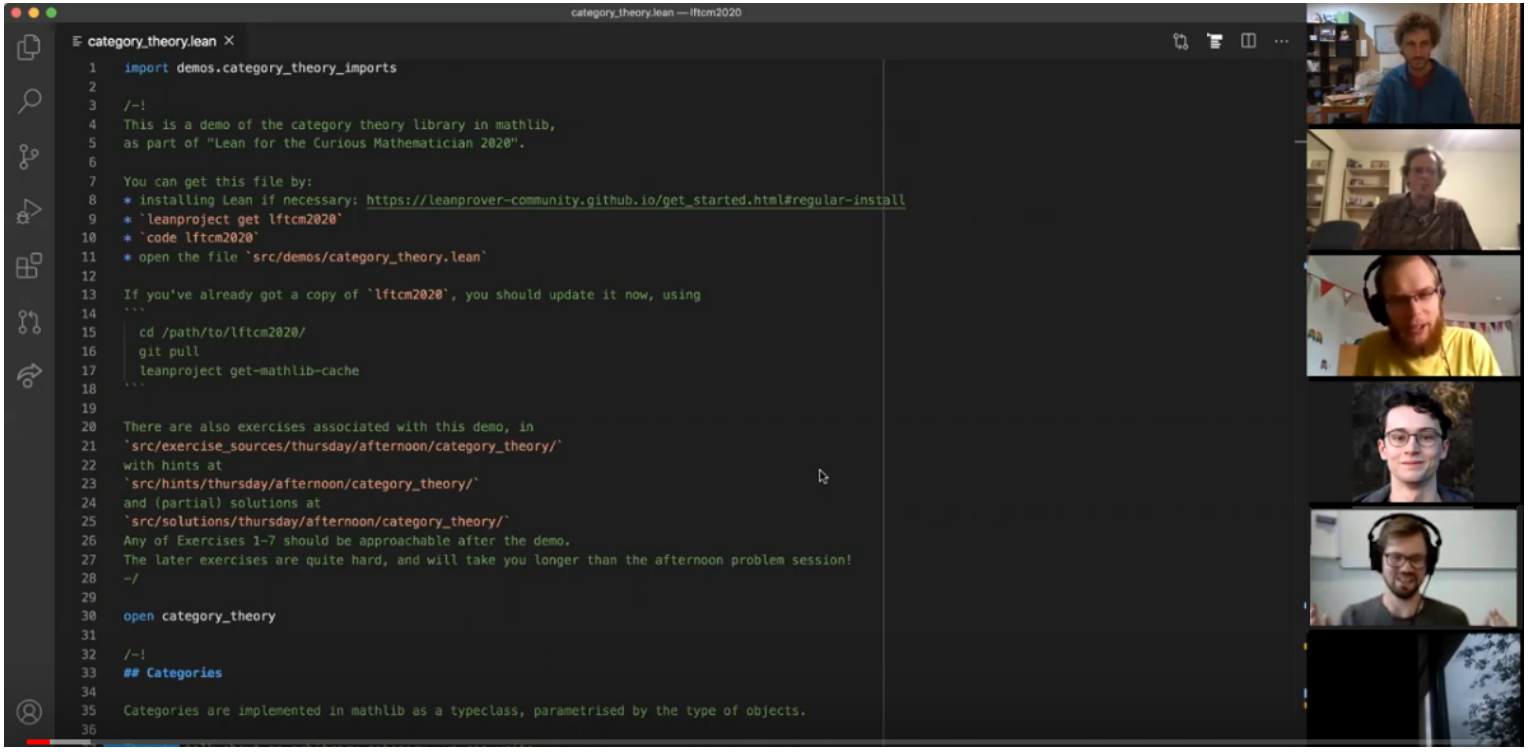
chore(scripts): update nolints.txt (#5554) Browse files

```
I am happy to remove some nolints for you!
```

master

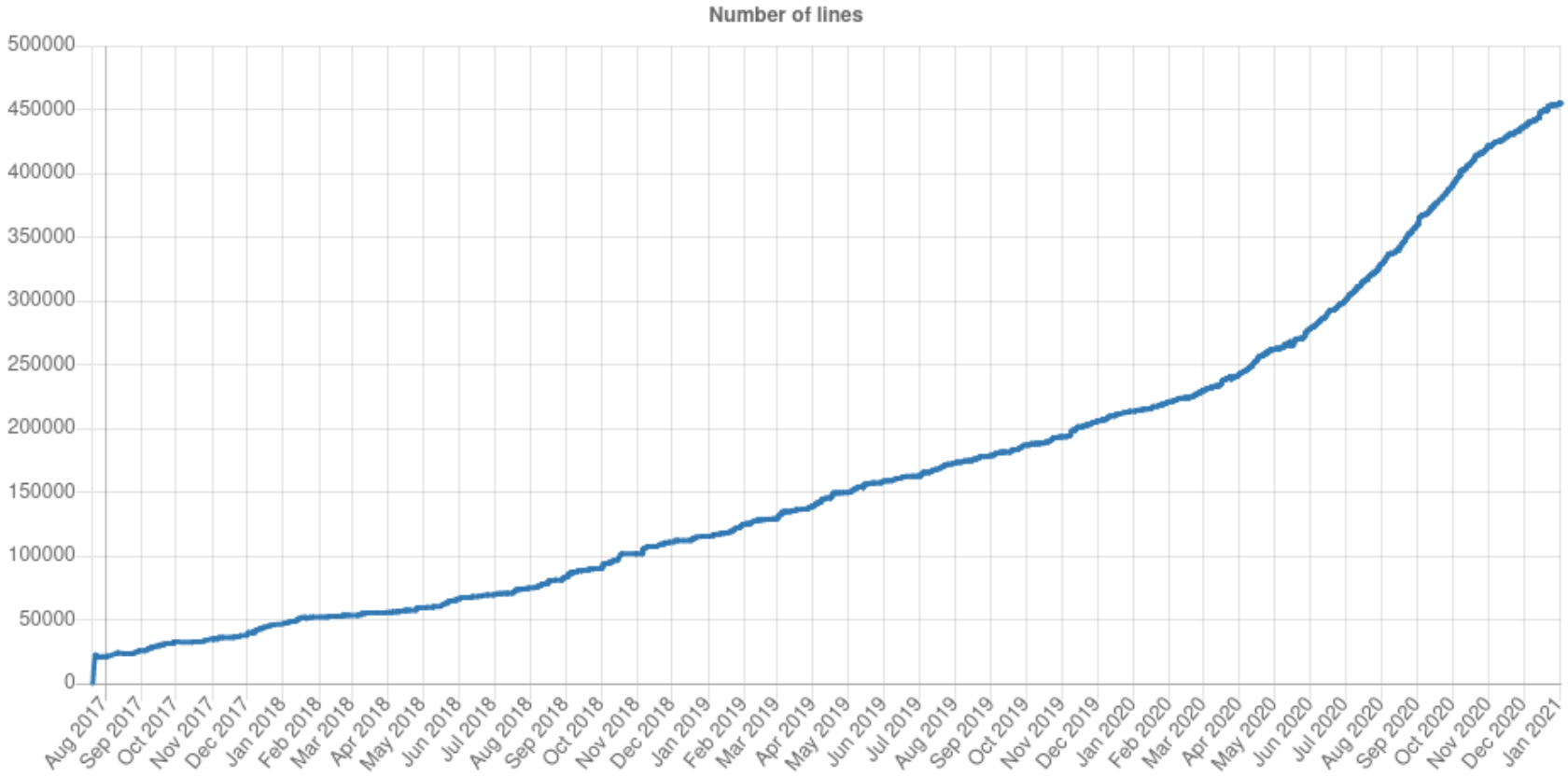
leanprover-community-bot committed 3 days ago 1 parent 54bf708 commit d2bde119a93bf08dfd163fb7744d1bc9059b2b2d

Lean for the curious mathematician 2020



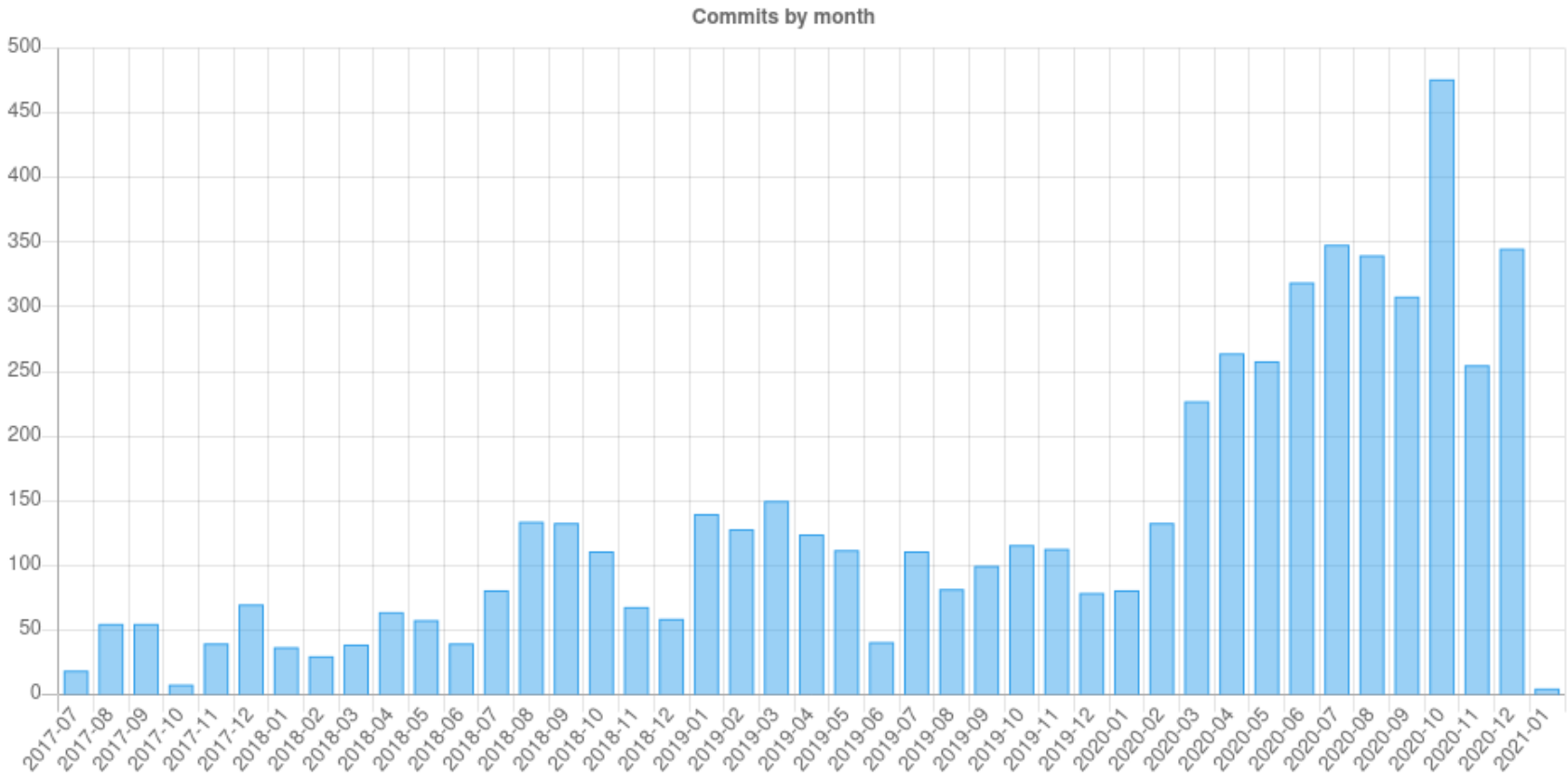
A virtual summer school with 75 participants.

Mathlib growth



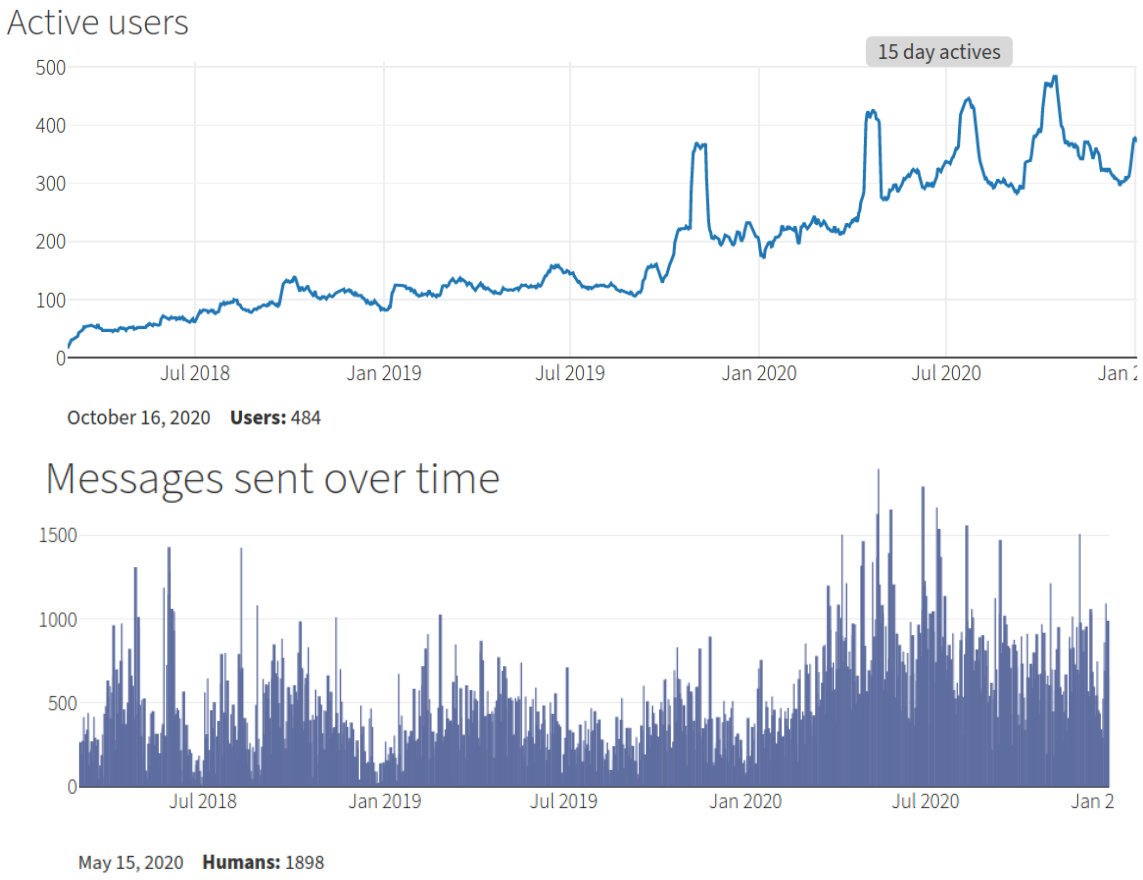
Numbers of contributors, lines of code and declarations all roughly doubled in 2020.

Mathlib growth



Numbers of contributors, lines of code and declarations all roughly doubled in 2020.

Community growth



Adopted the Contributor Covenant Code of Conduct in August.

62 new contributors

Aaron Anderson, Adam Topaz, Adrián Doña Mateo, Alena Gusakov, Alexandru Bosinta, Alex Peattie, Anatole Dedecker, Angela Li, Ashvni Narayanan, Benjamin Davidson, Bolton Bailey, Carl Friedrich Bolz-Tereick, Chris M, Chris Wong, Daan van Gent, Damiano Testa, Daniel Fabian, Daniel Selsam, Dan Stanescu, David Wärn, Devon Tuma, Ed Ayers, Eric Wieser, Filippo Nuccio, Fox Thomson, Frédéric Dupuis, Frédéric Le Roux, Gihan Marasingha, Heather Macbeth, Jalex Stark, James Arthur, Jannis Limperg, Jason Yi, Joseph Myers, Jujian Zhang, Julian Bernan, Junyan Xu, Kenji Nakagawa, Ken Lee, Kevin Lacker, Kyle Miller, Malo Jaffré, Markus Himmel, Martin Zinkevich, Matj Grabovsky, Nicolò Cavalleri, Patrick Lutz, Patrick Stevens, Paul van Wamelen, Qian Hong, Rémy Degenne, Riccardo Brasca, Ruben van de Velde, Shing Tak Lam, Sophie Morel, Stanislas Polu, Thomas Browning, Thomas Read, Utensil Song, Vaibhav Karve, Xi Wang, Yakov Pechersky

Some new things in mathlib

Abelian category, sheaf with values in a category, abstract homology
Eigenvalue, eigenvector, Cayley-Hamilton
Tensor algebra, exterior algebra, classical Lie algebras
Cyclotomic polynomial, algebraic closure of a field, Galois correspondence
valuation, perfection of a ring, Witt vector, Dedekind domain
discrete valuation ring, fractional ideals, adic completion
Affine space, Euclidean geometry, Carathéodory, Mazur-Ulam
Prime spectrum, Zariski topology, scheme
Bolzano-Weirstrass, Heine-Cantor, uniformization of compact spaces
Haar measure, Fubini, antiderivative, Jensen's inequality
Hölder and Minkowski inequalities, Hahn-Banach, Fréchet-Riesz
Analytic function, Inverse function and implicit function theorems, smooth functions between manifolds, Lie groups, rotation number

`slim_check`, `list_unused_decls`, `induction'`, `zify`, `nlinarith`, `group`

LT2021 practical information

- Goal of this workshop: what are people working on these days?
- Mix of WIP and more complete projects

- Talks on Zoom
- Extended discussions on Zulip
- Social time on Wonder