HAL open science

Lessons for Interactive Theorem Proving Researchers from a Survey of Coq Users

Ana de Almeida Borges, Annalí Casanueva Artís, Jean-Rémy Falleri, Emilio Jesús Gallego Arias, Érik Martin-Dorel, Karl Palmskog, Alexander Serebrenik, Théo Zimmermann

To cite this version:

Ana de Almeida Borges, Annalí Casanueva Artís, Jean-Rémy Falleri, Emilio Jesús Gallego Arias, Érik Martin-Dorel, et al.. Lessons for Interactive Theorem Proving Researchers from a Survey of Coq Users. 14th International Conference on Interactive Theorem Proving (ITP 2023), Jul 2023, Bialystok, Poland. pp.1-18, 10.4230/LIPIcs.ITP.2023.12 . hal-04098856v2

HAL Id: hal-04098856

https://telecom-paris.hal.science/hal-04098856v2

Submitted on 26 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Lessons for Interactive Theorem Proving Researchers from a Survey of Coq Users

Ana de Almeida Borges
University of Barcelona, Spain

Annalí Casanueva Artís
Ifo Institut, München, Germany

Jean-Rémy Falleri
Univ. Bordeaux, Bordeaux INP, CNRS, UMR 5800 LaBRI, F-33400 Talence, Institut Universitaire de France, France

Emilio Jesús Gallego Arias
Université Paris Cité, CNRS, Inria, IRIF, F-75013, Paris, France

Érik Martin-Dorel
IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, France

Karl Palmskog
KTH Royal Institute of Technology, Stockholm, Sweden

Alexander Serebrenik
TU Eindhoven, The Netherlands

Théo Zimmermann
LTCI, Télécom Paris, Institut Polytechnique de Paris, France

Abstract

The Coq Community Survey 2022 was an online public survey of users of the Coq proof assistant conducted during February 2022. Broadly, the survey asked about use of Coq features, user interfaces, libraries, plugins, and tools, views on renaming Coq and Coq improvements, and also demographic data such as education and experience with Coq and other proof assistants and programming languages. The survey received 466 submitted responses, making it the largest survey of users of an interactive theorem prover (ITP) so far. We present the design of the survey, a summary of key results, and analysis of answers relevant to ITP technology development and usage. In particular, we analyze user characteristics associated with adoption of tools and libraries and make comparisons to adjacent software communities. Notably, we find that experience has significant impact on Coq user behavior, including on usage of tools, libraries, and integrated development environments.

2012 ACM Subject Classification Software and its engineering → Formal methods; General and reference → Empirical studies

Keywords and phrases Coq, Community, Survey, Statistical Analysis

Digital Object Identifier 10.4230/LIPIcs.ITP.2023.12

Supplementary Material *Software (Source Code and Data)*: `https://doi.org/10.5281/zenodo.7930567` [12]

Acknowledgements The authors would like to thank the survey participants, the Coq Survey Working Group members who are not simultaneously authors of this paper (Yves Bertot, Nathan Cassee, Jim Fehrle, Jerome Hugues, Barry Jay, Matthieu Sozeau and Enrico Tassi), the survey beta testers, and the translator team (Yishuai Li, Oling Cat and Weidu Kuang).

© Ana de Almeida Borges, Annalí Casanueva Artís, Jean-Rémy Falleri, Emilio Jesús Gallego Arias, Érik Martin-Dorel, Karl Palmskog, Alexander Serebrenik, and Théo Zimmermann;
licensed under Creative Commons License CC-BY 4.0
14th International Conference on Interactive Theorem Proving (ITP 2023).
Editors: Adam Naumowicz and René Thiemann; Article No. 12; pp. 12:1–12:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Online surveys are employed by organizations [16, 28] and researchers [30] to obtain aggregate data about software communities, e.g., with the aim of understanding community demographics, programming practices, and desired improvements to a software ecosystem.

Some interactive theorem provers (ITPs) such as Coq, Isabelle/HOL, and Lean have many active users across academia and industry, and have comparable popularity to programming languages on developer platforms such as GitHub and Stack Overflow [22]. Yet, to our knowledge, the only public summary for an ITP community survey is Braibant's for the Coq Community Survey 2014 [5]. However, due to the development of Coq itself and changes in the Coq community since 2014, many of its results and conclusions are currently obsolete.

In this paper, we present the results of a new survey of Coq users, which we call the Coq Community Survey 2022. The survey was conducted publicly online during February 2022, using Inria's LimeSurvey platform. The main goal of the survey was to obtain an updated picture of the Coq user community and inform future decisions taken by the Coq development team (*Coq Team* for short) and other stakeholders in the Coq ecosystem. In particular, the survey aimed to help the Coq Team to better prioritize issues and features for the Coq system, and enable effective decision-making about matters pertaining to the software ecosystem maintained by Coq users in academia and industry [3]. One specific Coq Team impetus for the survey was a community-wide discussion on Coq's name and logo in April 2021 on the Coq mailing list [31], centering on the name's pronunciation in English.[1]

The survey was designed and deployed by a working group (WG) that was formed following a public call by the Coq Team. The survey had 109 questions, some of which were conditionally visible, and was available in English and Chinese [8, 7]. Broadly, the survey asked about 1) use of Coq features, integrated development environments (IDEs), libraries, plugins, and tools, 2) views on renaming and improving Coq, and 3) demographic data, such as age, gender, and experience with Coq. The survey received 466 submitted responses.

The initial analysis and presentation of the survey [9, 10, 11] was limited to descriptive response summaries and basic inferred data, such as defining *graduate students* by combining responses to highest completed academic degree and student status. Our response analysis here goes beyond the initial presentation by performing statistical analyses – multiple variable regression analyses that unveil characteristics of Coq users associated with different usage habits. We also compare aggregate data of Coq survey respondents with data from similar surveys in related software communities (Haskell [15] and Stack Overflow [28]), which we believe can give some perspective on the contention that the Coq community is oriented towards programming languages [4]. We make the following contributions:

1. We present the first Coq community survey since 2014, and to our knowledge the survey of an ITP community with the largest number of respondents thus far.
2. We have communicated descriptive results to the Coq community early, by publishing a series of articles on the Coq Discourse forum and GitHub issues to make specific stakeholders (mostly IDE maintainers) aware of the results of highest interest to them.
3. We provide novel survey data analyses by performing multiple variable regression analyses to answer one main research question: **How do different categories of respondents use Coq?** In addition, we look at satisfaction and needs for improvements for the same categories of respondents. Our code and results are part of our public dataset [12].

[1] While the renaming issue is outside paper scope, aggregate renaming responses are available [12, 2].

The paper is structured as follows. Section 2 describes the survey design and deployment, the data analysis process, and the methodology for our statistical analyses, including the definition of our respondent categories of interest. Section 3 mentions the previous descriptive analysis of the survey results, including some examples, and performs a comparison between the Coq community and the Stack Overflow and Haskell communities, based on their respective 2022 surveys. Section 4 compares Coq use across different categories of Coq users. Section 5 compares differing satisfaction and user needs across the same categories of Coq users. Section 6 discusses threats to validity of our results, and measures taken to mitigate them. Finally, Section 7 discusses our results and concludes.

2 Methodology

The survey was designed, deployed, managed, and analyzed by members of a working group (WG), which was formed after a public call for participation by the Coq core team. Members of this WG met weekly during a year and a half, first to design the survey, then to advertise it to the Coq community, and finally to analyze and share the results.

2.1 Survey Design and Deployment

Survey questions were collaboratively created by WG members. The WG took inspiration from the previous Coq Community Survey in 2014 [5], and similar surveys for programming languages [20]. The WG made efforts to adhere to survey best practices [19], e.g., on ordering demographic questions late to avoid their answers affecting other questions.

The survey was available in English [8] and Chinese [7], having been first created in English and then translated to Chinese by a team of volunteers. These volunteers also translated the answers to the open questions back to English. Ideally, the survey would have been translated to more languages, particularly the ones that have an existing community of Coq users who are not necessarily proficient in English. For example, there are Coq learning resources in Chinese [26] and Japanese [25], as well as dedicated categories for several languages in the Coq Discourse forum. The WG contacted colleagues with the necessary language knowledge to perform Chinese, Japanese, and Russian translations, but only secured a commitment for the Chinese translation.

The survey had 109 questions, some of which were only visible depending on answers to previous questions. Broadly, the survey asked about 1) use of Coq features, IDEs, libraries, plugins, and tools, 2) views on renaming Coq, Coq improvements, and issues such as inclusion, and 3) demographic data, such as age, gender, and length of experience with Coq and other ITPs or functional programming languages. All the questions were optional.

After a preliminary version of the survey was finalized, it was sent to a small group of beta testers, who were asked to fill out that preliminary version, time themselves, and send feedback on their experience. Some changes were made due to this feedback, and the initial prediction of the time it would take to fully complete the closed-question part of the survey (30 minutes) was estimated based on this beta testing. We made this estimation available to potential survey respondents on the first page of the survey and in our announcements.

Before survey deployment, we contacted the Inria Ethics Review Board (ERB), who informed us that full ERB approval of our survey was not required and referred us to Inria's Data Protection Officer (DPO) for data protection compliance matters. Based on our communication with the DPO, we developed a GDPR conformance statement and followed best practices for data protection. In particular, we removed the raw data and only share aggregated data, including open text answers which we manually sanitized to remove any personally identifiable elements.

The survey was deployed on Inria's LimeSurvey platform and was open during the month of February 2022. The deployed survey was advertised to Coq users primarily via the following Coq community forums: Discourse, Zulip chat, Coq-club mailing list, Coq Twitter, Coq Reddit, and the Coq website. The WG also advertised in more general venues and other related communities, such as the Types-announce mailing list, Agda and Lean Zulip chats, and on language-specific forums in Chinese and in French (mailing lists of GDR IM and GPL). These announcements asked only previous or current Coq users to answer. Still, the survey included the question "How long have you used Coq?" with the option to answer "I have never used Coq." Those who chose that option were still asked a number of questions, but most of the survey was hidden, since most questions assumed previous Coq experience.

2.2 Analysis Process

After the survey closed, WG members collaboratively analyzed the response data. For GDPR compliance, the WG decided to keep the raw survey data within the EU. Therefore, the WG used Inria-hosted tools and restricted raw data access to members located in the EU. The Chinese translators translated the Chinese answers back into English, and then two authors wrote the analysis code in Python within a Jupyter notebook, using the `pandas` and `matplotlib` libraries. Plots were created for each closed-answer survey question (often multiple plots for each question) and automatically deployed to a static Inria-hosted website for later inclusion in summary blog posts (see Section 3). In order to perform the statistical analyses described in Section 4, we exported pre-processed data from the Jupyter notebook for a third author to perform regressions using Stata.[2] We provide the Jupyter notebook and Stata code as supplementary material [12] to allow scrutiny of the source code, and also provide all the generated plots for the survey questions. However, due to privacy concerns (risk of re-identifying specific respondents), we do not provide the raw survey data, which is required to reproduce the analyses using the Jupyter notebook.

2.3 Statistical Analyses

We do a multiple variable regression analysis to unveil the characteristics of Coq users associated with different usage habits and expectations. This statistical method allows us to establish the relationship between multiple regressors (or independent variables) and one dependent variable (or outcome) [32]. Including several variables at the same time is needed to disentangle the distinct effect of possibly correlated variables. If we did not include such correlated variables simultaneously, part of the effect of the excluded variables would be captured by the coefficient of the included correlated ones. One must however keep in mind that: 1) these results cannot be interpreted causally and 2) there can still be variables correlated with the dependent or independent variables that were not included in the regression.

For each category of users, we test several hypotheses. This will lead to over-rejecting null hypotheses (i.e., some associations between variables will be considered true when in reality they are not present). To overcome this problem, it is necessary to explicitly consider the multiplicity of the testing framework. Thus, for all our estimates, we report, in addition to classic standard errors (in parentheses), the Romano-Wolf-corrected p-values for multiple hypothesis testing (in brackets) [27, 6]. This correction has a higher ability to correctly reject

[2] Stata is a software package for statistical analysis, see `https://www.stata.com`.

false null hypotheses than previous techniques such as Bonferroni and Holm [17]. Following common standards in fields with a long tradition of statistical analysis such as economics, and considering that, by controlling the family-wise error rate, the Romano-Wolf correction is more conservative than alternative approaches to multiple hypothesis testing (giving us robust levels of significance), we interpret estimates as significant as soon as the p-values are below the 0.1 threshold (instead of the 0.05 threshold which would be more common in empirical software engineering).

2.3.1 Defining Categories of Interest

We studied four user characteristics that we considered could be relevant in explaining differences in how respondents use Coq: their experience with Coq, their learning status, their application-domain for Coq, and their location. More specifically:

Experience level We define two binary variables that illustrate various stages of experience with Coq. The first variable represents whether the user has more than 2 years of experience, and the second variable whether the user has more than 5 years of experience. About 64% of our sample has more than 2 years of experience and about 40% more than 5. The omitted category represents users with less than 2 years of experience. We use these variables (corresponding to arbitrary thresholds) instead of a numeric variable representing the number of years of experience because experience does not always have a linear association with outcomes, such as the likelihood of using a particular feature. In particular, we can expect that, beyond some level of experience, additional years of experience are very unlikely to be associated with any additional changes in the outcome.

Learners We define a binary variable corresponding to learners. To define learner, we use the answer to the question "For what purpose have you used Coq?". A learner is a respondent who answered any of "Learning Coq" or "Learning something other than Coq" to this (multiple-choice) question, without also responding any of "Teaching Coq", "Teaching something other than Coq", "Academic research", or "Industrial research / application". About 17% of our sample are defined as learners. The way that learners use Coq may be related to their learning context, and thus significantly differ from other Coq users. Understanding these differences could inform researchers and engineers trying to make proof assistants easier to learn.

Application domains for using Coq Coq can be used for various applications: verifying software, hardware, or theoretical systems, formalizing existing or new mathematical results, etc. The way of using Coq may differ depending on the application domain (extraction comes to mind as being specifically relevant for software verification), but also on the users' background. This is why we decided to test whether software verification specifically (the most common objective for Coq users) was related to differing practices. About 67% of our sample use Coq for software verification.

Location Various world regions have different research traditions (such as putting more or less focus on theory) and may also have different technical culture. This is why it seemed important to evaluate whether this relates to different ways of using Coq. Our baseline is Europe, where about half of our respondents live. We define two binary variables to represent the respondents' location: one for North America (which represents about 25% of our respondents' location) and one for the rest of the world (excluding both North America and Europe). This is why Europe does not appear explicitly in our tables.

2.3.2 Controls

In addition to the variables presented above, our regressions include additional variables as controls. In particular, we include: the age of respondents, their gender (more precisely, a binary variable for whether the respondent declared being a woman), their operating system (two binary variables for Windows and macOS), their research area (a binary variable for whether the respondent does research in math or logic), their employment (a binary variable for whether the respondent is employed by a private organization), and whether they hold a PhD. We do not analyze the results of the regressions for these variables; they are only there to avoid capturing something that could be correlated both to the independent variables and to the outcome. For example, if we include years of experience without including age, we might be capturing the effect of being of a certain generation on using a specific feature that was particularly popular during a specific period. Because years of experience is correlated with age, part of the association between age and the outcome is captured by the coefficient of years of experience. Yet, in this example, the experience would not be relevant.

2.3.3 Selecting Population to Analyze

Our survey received 466 submitted responses. However, for the statistical analyses, we restrict ourselves to the 390 respondents who say that they have used Coq in the last year (as what we want to assess is current Coq practices). Furthermore, we can only perform regressions on data where our (dependent and independent) variables, including our controls, are defined. We remove 18 respondents who did not provide their age. This results in looking at a subsample of 372 responses. On one specific outcome (IDE satisfaction), we look at an even smaller subsample of 367 responses, because respondents need to have answered at least one IDE satisfaction question for this outcome to be defined.

3 Descriptive Observations and Comparison to Other Communities

3.1 Descriptive Analysis and Observations

After the survey closed, the WG performed a descriptive analysis of the data. The results were continuously shared with the community through a series of articles posted to the Coq Discourse forum. The topics were as follows: *Who is using Coq and in what context?* [9]; *How are people using Coq?* [10]; *How is Coq used? (features, tools, libraries)* [11]. They include many plots deployed from a GitLab repository to a perennial URL,[3] in SVG and PNG format. Results from the articles and additional results on renaming Coq were also presented at the Coq Workshop 2022 [1, 2]. Specific results were shared as repository issues:

Proof General `https://github.com/ProofGeneral/PG/issues/671`
Company-Coq `https://github.com/cpitclaudel/company-coq/issues/258`
CoqIDE `https://github.com/coq/coq/issues/16580`
VsCoq `https://github.com/coq-community/vscoq/issues/308`
Coqtail `https://github.com/whonore/Coqtail/issues/277`
jsCoq `https://github.com/jscoq/jscoq/issues/261`
coq_jupyter `https://github.com/EugeneLoy/coq_jupyter/issues/46`
Continuous integration `https://github.com/coq-community/manifesto/issues/141`

[3] `https://thzimmer.gitlabpages.inria.fr/coq-survey-2022-assets/asset-listing.html`

In addition, the WG prepared detailed internal reports and presentations to the Coq core team on the results to the renaming questions to help the team make a decision regarding a possible renaming of Coq. At the time of writing, no decision has been made official yet.

Here, we present only a small fraction of these descriptive results: a selection of demographic markers in Figure 1 and IDE and operating system (OS) use in Figure 2. We comment on these while comparing the Coq community to two other relevant communities, and refer the reader to the Discourse forum articles for a more detailed presentation [9, 10, 11].

3.2 Comparison of the Coq Community to Other Communities

In this section, we compare the Coq community (CC) to two distinct baseline populations. We use the 2022 Stack Overflow survey results [28] as a proxy for the "general" programming community, since Stack Overflow is a popular website used by millions of software developers. We will refer to this population as SO. We use the 2022 Haskell survey results [15] as a proxy for the functional programming (FP) community, as Haskell is a mature but not obsolete functional programming language. We believe this population is interesting due to the proximity between the ITP and FP communities. We will refer to this population as HC.

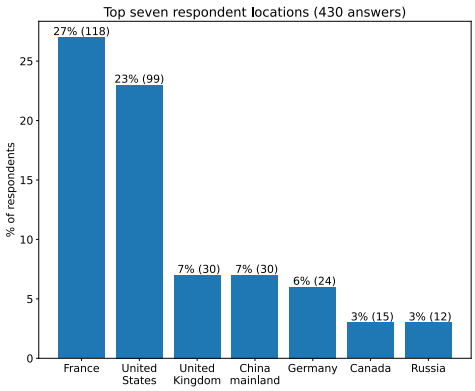
Location. The United States is consistently one of the top-5 locations, as are Germany and the United Kingdom, for all three populations. As can be seen in Figure 1a, CC has a large ratio of French respondents, even outnumbering US respondents, likely for historical reasons (Coq was originally developed in France). A notable difference is that Indian respondents, numerous in the SO population, are rare in the FP and CC populations, indicating a missed opportunity for FP and ITP.

Gender. There is no data for HC, but a comparison can be done between CC and SO. It shows the same trend of an overwhelming ratio of respondents identifying as men (about 90%), while only about 6% identify as women, and the remaining 4% as non-binary or other.

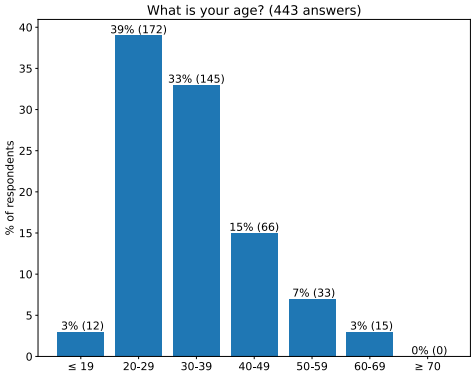
Age. There is no data for the HC population, but SO and CC (Figure 1b) can be compared. Although the bins are slightly different between SO and CC (shifted by five years), the overall trend seems similar between the two populations. Very young programmers (less than 20 years old) are rare (about 5%), with the bulk of programmers being approximately between 20 and 40 years old. Older programmers (more than 50 years old) represent about 10% of both populations.

Education. CC has fewer bins for education than either HC or SO (Figure 1c). We enable direct comparisons between surveys by merging bins where possible and dropping bins and corresponding answers otherwise, since bins are mutually exclusive. This analysis yields responses for no diploma for SO 3%, HC 2%, CC 1%; high school diploma for SO 26%, HC 17%, CC 5%; bachelor's degree for SO 45%, HC 37%, CC 18%; master's degree for SO 23%, HC 30%, CC 31%; PhD degree for SO 3%, HC 14%, CC 46%. These results place HC between SO and CC in terms of educational achievement, but closer to SO given that CC has such a high skew towards doctoral degrees.

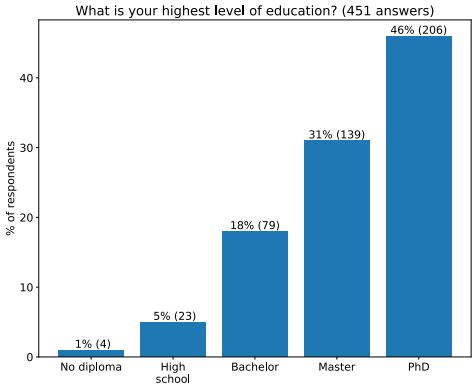
Academic use. We now compare the "academic" bins of the three surveys, even if their definitions are not exactly the same. In both the CC survey and the HC survey, respondents are asked if they operate in an academic context and if they are students or not. In the



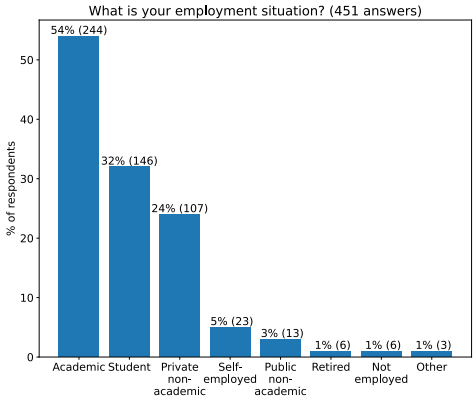
(a) Top locations of residence of the respondents.



(b) Age of the respondents.



(c) Highest education level completed by respondents.

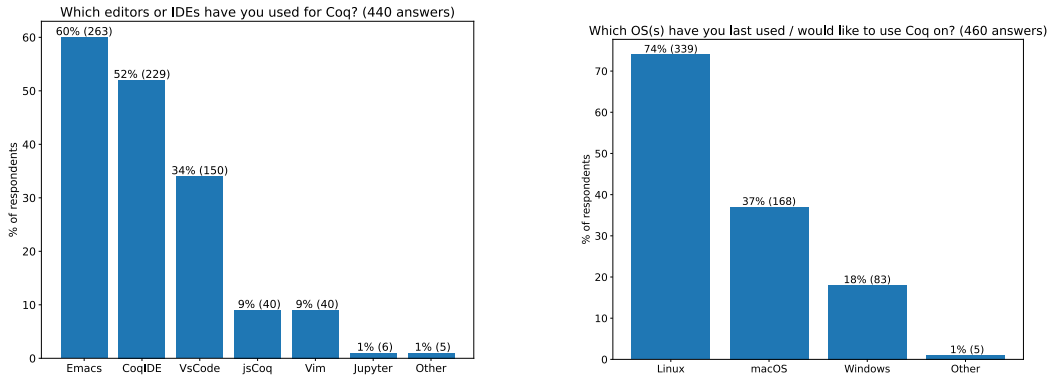


(d) Employment situation of the respondents.

Figure 1 Plots depicting the location, age, education level and employment situation of the respondents to the 2022 Coq Community Survey.

SO survey, academic researcher and student are exclusive categories. In CC, as can be seen in Figure 1d, 54% of respondents are academically employed, and 32% are students. In HC, 18% of the respondents use Haskell in an academic context and 15% are students. In comparison, only 4.42% of Stack Overflow survey respondents are academic researchers and 9.13% are students. In conclusion, CC and HC respondents operate in an academic context much more frequently than SO respondents. Even in comparison with the HC respondents, CC is even more tied to academia, where it represents about half of the respondents.

IDEs. The bins are not the same across the surveys, but we can compare the following three main IDEs: Vim, Emacs, and VS Code. See Figure 2a for IDE use in the CC population. VS Code is used by about 70% of SO, 43% of HC and 34% of CC. Emacs is used by about 5% of SO, 30% of HC and 60% of CC. Vim is used by about 23% of SO, 40% of HC and 9% of CC. When considering only Emacs, Vi(m), and VS Code, the Coq survey answers suggest that Coq users are significantly more likely to use Emacs than either Haskell users or developers on Stack Overflow. This is partly driven by long-time Coq users preferring Emacs (and its ProofGeneral package) and VS Code support for Coq being more recent.



(a) IDEs used by the respondents.

(b) Operating systems used by the respondents.

Figure 2 Plots on IDE and OS use of the survey respondents.

OSes. The bins are not the same across the surveys, but we can compare the following three main OSes: Linux, Windows and macOS. For the SO survey, the best data is professional usage as it reflects the coding environment. See the data for CC in Figure 2b. Windows is used by about 49% of SO, 15% of HC and 18% of CC. macOS is used by about 31% of SO, 33% of HC and 37% of CC. Linux is used by about 40% users of SO, 86% of HC and 74% of CC.

These percentages suggest that Linux-based operating systems are dominant among both Haskell and Coq users, with only a small minority of Windows users and a similarly-sized minority using macOS in both communities. In contrast, among Stack Overflow users, a near-majority use Windows. Windows use is slightly higher among Coq survey respondents than Haskell respondents, which is partly driven by new Coq users.

4 Analysis of Coq Use for Different Population Groups

In this section, we analyze how different population groups we identified in the survey use Coq differently, by looking at various outcomes.

4.1 Installation of Coq, Usage of Packages and Features, and CI

In this section, we analyze the results of Table 1, which contains the following outcomes:

opam A binary variable indicating whether respondents installed Coq with opam (a software package manager for OCaml [23] and for Coq as well [29]), either directly, or relying on wrapper scripts provided by the Coq Platform [24]. Because Coq has been around for so long, there are many possible ways to install it. It is packaged in many Linux distributions and generic package managers, and there are binary installers for the main operating systems. The most flexible installation method is by using opam, because it gives access to the whole ecosystem of Coq packages. This is the most common installation method (used by 76% of respondents), but it is also more complex, so the expectation is that users start relying on it only when they encounter the need.

External general libraries A binary variable for whether respondents are "casual" or "advanced" users of any of the external general libraries Coq-Extlib, Coq-std++, Math Classes, Mathematical Components, or TLC. Coq is distributed with a standard library and some embedded tools, but it also comes with a rich package ecosystem. Power users

Table 1 Usage of opam, ecosystem libraries, tools, automation, SSReflect, Continuous Integration (CI), and extraction.

VARIABLES	(1) opam	(2) External general libraries	(3) Number of external tools	(4) Automation	(5) SSReflect	(6) CI	(7) Extraction
> 2 years of experience	**0.183*** **	0.107	-0.118	0.110	-0.0196	0.0512	**0.231*** **
	(0.0576)	(0.0663)	(0.233)	(0.0620)	(0.0644)	(0.0600)	(0.0653)
	[0.008]	[0.253]	[0.818]	[0.253]	[0.818]	[0.720]	[0.005]
> 5 years of experience	0.00177	**0.169** **	**0.945*** **	**0.186** **	**0.138* **	**0.320*** **	**0.169** **
	(0.0594)	(0.0684)	(0.241)	(0.0639)	(0.0664)	(0.0619)	(0.0673)
	[0.967]	[0.045]	[0.001]	[0.016]	[0.059]	[0.001]	[0.045]
Learner	**-0.282*** **	-0.0882	0.193	-0.175	-0.0363	**-0.197** **	-0.0513
	(0.0619)	(0.0712)	(0.250)	(0.0665)	(0.0691)	(0.0645)	(0.0701)
	[0.001]	[0.685]	[0.872]	[0.116]	[0.872]	[0.042]	[0.872]
Software verification	0.0470	**0.165** **	**0.742*** **	**0.192*** **	**0.126* **	0.0371	**0.290*** **
	(0.0473)	(0.0544)	(0.191)	(0.0509)	(0.0528)	(0.0493)	(0.0536)
	[0.533]	[0.013]	[0.002]	[0.002]	[0.051]	[0.533]	[0.001]
North America	0.00277	0.00986	0.0795	-0.125	-0.0637	0.00597	0.00654
	(0.0519)	(0.0597)	(0.210)	(0.0558)	(0.0580)	(0.0541)	(0.0588)
	[1.000]	[1.000]	[0.999]	[0.170]	[0.860]	[1.000]	[1.000]
Other locations	-0.00968	-0.124	-0.0873	-0.0269	-0.0673	-0.122	**0.172* **
	(0.0617)	(0.0710)	(0.250)	(0.0663)	(0.0689)	(0.0643)	(0.0699)
	[0.960]	[0.358]	[0.956]	[0.956]	[0.802]	[0.290]	[0.089]
Additional controls	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mean of dependent variable	0.758	0.366	0.995	0.702	0.288	0.315	0.446
Observations	372	372	372	372	372	372	372

Note: Each column corresponds to a regression. Standard errors are in parentheses, Romano-Wolf adjusted p-values in brackets. Stars represent different levels of significance (*** p<0.01, ** p<0.05, * p<0.1) according to Romano-Wolf adjusted p-values. Independent variables are described in §2.3.1 and additional controls in §2.3.2. Mean of dependent variable corresponds to the mean of the outcome of each column.

rely on this package ecosystem to be more productive. With this outcome and the next one, we evaluate how different types of users take advantage (or not) of the package ecosystem. Only 37% of respondents are users of a general external library.

Number of external tools A natural number variable counting how many external tools respondents said they were using ("Casual user" or "Advanced user") among a long list: Mtac2, MetaCoq, and Coq-Elpi (from the "Tactic languages" question), as well as all 16 tools from the "Ecosystem plugins and tools" question. Respondents to the survey only use 0.995 of the 19 listed external tools on average.

Automation A binary variable for whether respondents are "casual" or "advanced" users of any of several core automation tools (Micromega, Nsatz, Ring, and the solvers for logic and equality from the standard library, such as firstorder) or external automation tools (AAC Tactics, CoqHammer, and SMTCoq). 70% of respondents are using such a tool.

SSReflect A binary variable for whether respondents are "casual" or "advanced" users of the SSReflect proof language. SSReflect is a consistent set of tactics providing an alternative to the standard tactics. It has been distributed as part of the main Coq package for more than 5 years. 29% of our respondents use SSReflect.

CI A binary variable for whether respondents use Continuous Integration (CI) on any of their projects. CI is the practice of automatically running tests at every change when developing software. This is a very common practice today, which helps to produce better quality software [13]. CI can also be used on mechanized proof projects, usually not to run tests, but simply to verify that the proofs are still accepted. This is particularly useful when developing Coq libraries, because CI can help ensure that a library remains compatible with several consecutive versions of Coq, even though its maintainers only

test one version on their machines. Because of this main application, it is expected that respondents who do not yet have big enough projects will not use CI. 31% of our respondents declared they do.

Extraction A binary variable for whether respondents are "casual" or "advanced" users of any of the officially supported extraction targets. Extraction is a Coq feature allowing the automatic generation of executable code in OCaml, Haskell or Scheme from a program written in Gallina, the functional programming language embedded inside Coq. Since Coq can be used to prove a Gallina program correct with respect to its specification, extraction is a useful mechanism to produce executable code for this program, and it is expected to be strongly associated with using Coq for verifying software. 45% of our respondents said they use extraction.

Experience level

We observe statistically significant results relating experience to each of our outcomes. For most of our outcomes (external general libraries, number of external tools, automation, SSReflect and CI), the threshold to see a statistically significant difference is to reach more than 5 years of experience. The most important jump being in the use of CI, which is of the same magnitude as the mean of the variable over our population.

Sometimes, 2 years of experience are enough to see significant differences: the use of opam increases (18 percentage points (pp) more respondents using opam) at the 2 years of experience threshold, and the use of extraction increases twice, at each of the two experience thresholds (23 pp more respondents using extraction after 2 years, and 17 pp more respondents using extraction after 5 years).

Learners

Learners use opam significantly less often (28 pp fewer respondents use opam when considering learners compared to non-learners with every other independent variable fixed, including experience). This may be explained by learners having a stronger need for an easy installation method and a lesser need for the access to packages that installing with opam provides. The only other statistically significant difference robust to multiple hypothesis testing is in the use of CI. 20 pp fewer learners use CI for their projects compared to non-learners, which can be explained by learners not having any project that really requires the use of CI. Other coefficients are estimated to be negative as well, but are not statistically significant or not robust to multiple hypothesis testing corrections.

Software verification

Besides experience level, the other category of users associated with many statistically significant differences in their use of Coq are respondents applying Coq to do software verification. These users are more likely to use external general libraries (16 additional pp), they use more external tools (an average of 0.74 additional tools, which is close in magnitude to the average number of external tools used by our respondents), they are more likely to use automation, more likely to use SSReflect, and more likely to use extraction. In short, we can say that these respondents (who represent about 67% of our respondents) are the ones who make the most use of the Coq package ecosystem and features. This may not come as a surprise for, e.g., the use of extraction, which is a feature particularly suited to software verification, but may be more surprising when talking about the use of external general libraries or SSReflect, given that MathComp (one of these general libraries) and SSReflect were not created to verify software, but to formalize pure mathematics.

Beyond the previously listed significant results, we can also note a significant, but difficult to interpret difference on the use of extraction by respondents from other locations (excluding Europe and North America).

4.2 User Interfaces

Table 2 Usage of Integrated Development Environment (IDE).

VARIABLES	(1) Emacs	(2) CoqIDE	(3) VS Code	(4) jsCoq
> 2 years of experience	0.0164	**0.220*** **	0.00848	0.0135
	(0.0651)	(0.0679)	(0.0679)	(0.0418)
	[0.991]	[0.009]	[0.991]	[0.991]
> 5 years of experience	0.0778	-0.0318	-0.0134	-0.0146
	(0.0672)	(0.0700)	(0.0700)	(0.0431)
	[0.688]	[0.963]	[0.963]	[0.963]
Learner	-0.109	0.124	-0.0897	0.0963
	(0.0699)	(0.0729)	(0.0729)	(0.0448)
	[0.265]	[0.265]	[0.265]	[0.147]
Software verification	0.120	0.0376	-0.110	0.0431
	(0.0535)	(0.0557)	(0.0557)	(0.0343)
	[0.118]	[0.515]	[0.170]	[0.398]
North America	**0.132***	-0.117	-0.00667	-0.0416
	(0.0587)	(0.0612)	(0.0612)	(0.0376)
	[0.077]	[0.145]	[0.899]	[0.458]
Other location	-0.0743	-0.0147	-0.0140	-0.0759
	(0.0697)	(0.0727)	(0.0726)	(0.0447)
	[0.629]	[0.976]	[0.976]	[0.250]
Additional controls	Yes	Yes	Yes	Yes
Mean of dependent variable	0.608	0.516	0.360	0.091
Observations	372	372	372	372

Note: Each column corresponds to a regression. Standard errors are in parentheses, Romano-Wolf adjusted p-values in brackets. Stars represent different levels of significance (*** p<0.01, ** p<0.05, * p<0.1) according to Romano-Wolf adjusted p-values. Independent variables are described in §2.3.1 and additional controls in §2.3.2. Mean of dependent variable corresponds to the mean of the outcome of each column.

We relate the probability to have used one of four IDEs for Coq (Emacs, CoqIDE, VS Code and jsCoq) to our previously defined categories. The user interface, a.k.a. IDE (Integrated Development Environment), is a fundamental part of proof assistants such as Coq, as it is the interaction medium between users and the system.

Our survey included many IDE questions, meant to get a better idea of their user base, their limitations, and to be able to inform stakeholders. As mentioned in Section 3.1, we already shared these specific results with stakeholders by opening issues in relevant projects.

In Table 2, we look at the responses to the question "Which editors or IDEs have you used for Coq?" and relate the results for four of the most used IDEs to our previously defined categories of respondents. Among the four selected IDEs, the two with the most users according to the survey results are Emacs and CoqIDE (used respectively by 61% and 52% of respondents). This is explained by observing that they were the only two available

options for many years. Recently, more focus has been put on developing more modern user interfaces, e.g., based on the very popular Visual Studio Code (VS Code, used by 36% of respondents), or directly in the browser, with jsCoq (used by 9% of respondents).

Unfortunately, most of the significant results obtained in this table were not robust to Romano-Wolf corrections for multiple hypothesis testing. The only two robust results are:

1. Users with more than 2 years of experience are more likely to have used CoqIDE. This can be interpreted by CoqIDE being the most standard user interface for Coq, since it is distributed along the system, and the question being formulated as "Which editors or IDEs have you used for Coq?" and not "Which editors or IDEs are you *currently using* for Coq?". Thus, users are very likely to have used CoqIDE at some point when they are experienced enough.
2. Users in North America are 13 pp more likely to have used Emacs, compared to the Europe baseline, which may be explained by cultural differences or IDE penetration.

We do not get any other statistically significant results, although some are not very far from being significant. In particular, the learner category looks like it could be more frequently using jsCoq, but it is very difficult to draw reliable conclusions given the small number of jsCoq users among our respondents (40).

5 Satisfaction and Needs of Different Population Groups

In this section, we look at how our different population groups perceive Coq and what needs they express. This should inform researchers willing to improve proof assistants for specific use cases or target audiences.

We base our analysis on the responses to the satisfaction questions for the six IDEs that we included additional questions about, as well as one question on the importance of various improvements to be possibly made to Coq, one question asking what additional extraction targets respondents would like to have, and one question asking what additional languages to support in the documentation. Figure 3 presents the descriptive results to the question about improvements to be possibly made to Coq.

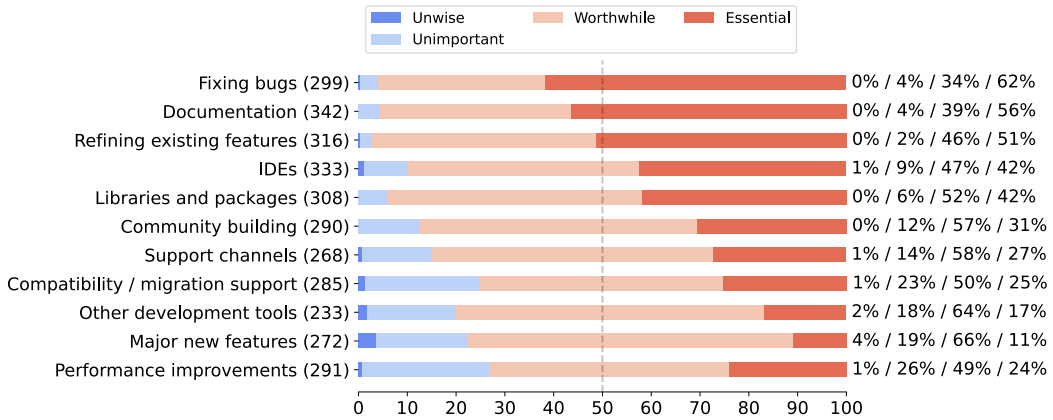


Figure 3 Results to the question "In order to make you more productive in Coq and to encourage others to learn and use Coq, how important are improvements in the following areas (relative to their current state)?". On the right hand-side, the percentages correspond, in order, to the proportion of "Unwise", "Unimportant", "Worthwhile", and "Essential" answers. Improvement categories are ordered according to the majority judgment methodology [14].

Table 3 Expressed satisfaction and needs.

VARIABLES	(1) IDE satisfaction	(2) Essential technical vs community improvements	(3) New extraction targets	(4) Other languages
> 2 years of experience	0.116	0.105	-0.0154	-0.0483
	(0.105)	(0.185)	(0.0510)	(0.0406)
	[0.632]	[0.826]	[0.826]	[0.632]
> 5 years of experience	**-0.236***	**0.484****	-0.00963	-0.00944
	(0.108)	(0.191)	(0.0526)	(0.0419)
	[0.085]	[0.047]	[0.976]	[0.976]
Learner	-0.224	-0.0625	0.0519	-0.00920
	(0.112)	(0.199)	(0.0548)	(0.0436)
	[0.159]	[0.936]	[0.704]	[0.936]
Software verification	0.0429	0.187	**0.163*****	0.0340
	(0.0856)	(0.152)	(0.0419)	(0.0333)
	[0.624]	[0.545]	[0.002]	[0.559]
North America	0.0721	-0.286	-0.00620	0.000884
	(0.0942)	(0.167)	(0.0460)	(0.0366)
	[0.779]	[0.265]	[0.982]	[0.982]
Other locations	0.00952	-0.219	0.0872	**0.122****
	(0.113)	(0.198)	(0.0546)	(0.0434)
	[0.923]	[0.415]	[0.304]	[0.038]
Additional controls	Yes	Yes	Yes	Yes
Mean of dependent variable	3.049	0.325	0.153	0.089
Observations	367	372	372	372

Note: Each column corresponds to a regression. Standard errors are in parentheses, Romano-Wolf adjusted p-values in brackets. Stars represent different levels of significance (*** p<0.01, ** p<0.05, * p<0.1) according to Romano-Wolf adjusted p-values. Independent variables are described in §2.3.1 and additional controls in §2.3.2. Mean of dependent variable corresponds to the mean of the outcome of each column.

Table 3 contains the following outcomes:

IDE satisfaction A natural number variable defined as the maximum IDE satisfaction level (from 0 to 4) for each respondent. We say that the IDEs each respondent assigned a maximum satisfaction level to are their favorites. To be defined, this outcome needs respondents to have answered at least one IDE satisfaction question, which is why we have 5 fewer observations for this outcome. The average satisfaction level given to the favorite IDE is 3, which corresponds to "Satisfied".

Essential technical vs community improvements An integer number variable defined as the difference between the number of essential technical improvements and the number of essential community improvements according to each respondent. We separate Coq improvements (from the question of Figure 3) into two categories: technical improvements (fixing bugs, refining features, IDEs, development tools, new features, performance) and community improvements (documentation, libraries, community building, support channels). We exclude the "Compatibility / migration support" improvement because it could be perceived as belonging to any of the two categories depending on how "support" is interpreted. We count for each of these categories the number of essential improvements, and we take the difference. Thus, a respondent with a positive value for this variable is listing more technical improvements as essential, and a respondent with a negative value for this variable is listing more community improvements as essential. On average,

respondents listed 0.3 additional essential technical improvements compared to community improvements, but note that the list of technical improvements contained two more items than the list of community improvements.

New extraction targets A binary variable for whether respondents provided a non-empty answer to the question on new extraction targets. 15% of the respondents asked for new extraction targets using this open text question.

Other languages A binary variable for whether respondents provided a non-empty answer to the question on languages to support in the documentation. 9% of our respondents asked for documentation in languages other than English using this open text question.

Experience level

Respondents with more than 5 years of experience are significantly more likely to assign a lower satisfaction score (0.24 lower on average) to their favorite IDE. They also list more essential technical improvements than community improvements, compared to other respondents (a 0.48 average difference).

We can explain experienced respondents being less satisfied with their IDE and more adamant about technical improvements by them having learned limitations of their IDEs or other technical limitations during their years of experience.

Other results

The other two statistically significant differences (robust to multiple hypothesis testing) are:

1. Respondents using Coq for software verification are more likely to request new extraction targets (an additional 0.16 pp, which is of the same magnitude as the mean of the variable over our population). Once again, this is not surprising as extraction is particularly suited to software verification, but it is limited by the restricted number of programming languages being currently officially supported as extraction targets.
2. Respondents in other locations (outside Europe and North America) are more likely to request support for new languages in the documentation (an additional 0.12 pp, which is largely superior to the mean of the variable over our population).

6 Threats to Validity

Several characteristics of our analysis represent a possible threat to validity. In terms of external validity (i.e., the validity of applying the conclusions of our study outside the specific context), there are two major concerns. First, it is possible that the Coq community behaves differently from other ITP communities. This can be due both to technical reasons (e.g., different proof assistants may be best suited to different use cases) and to sociological reasons ("birds of a feather flock together"), as could be observed, e.g., with the success of Lean among mathematicians. Second, even within the Coq community, the people who responded to the survey did not form a random sample of the community. It is likely that survey respondents had characteristics different from the average Coq user. In particular, and given the length of the survey, only people with enough time and motivation completed the survey in full. The results then concern this subsample of users and may not be extrapolated to the whole community. We took measures to avoid selection bias due to the use of the English language, but by only making the survey available in a second language (Chinese), and not many more, we are likely to have missed potential respondents in other linguistic communities. We tried to reach Coq users broadly by advertising the survey on many forums and mailing

lists, including language-specific ones, on social networks, and on the Coq website, but each choice of platform necessarily introduced bias in the sample. There is no way to completely avoid that beyond trying to reach even more broadly. For instance, we tried to reach learners as well by asking teachers to advertise the survey to their students, but we do not know how many did so. Our analyses of how different categories of respondents use Coq differently or express different needs reduce the issues of having a biased sample by including many controls, but we very likely missed some important ones.

Beyond external validity, there are other factors that can lead to misleading results. For example, the relatively low number of respondents may reduce the precision of the estimates, increase confidence intervals and reduce the power of the analysis (i.e., the ability to find a statistically significant effect when it exists). The order in which questions were asked may have affected the answers, which is a well-known phenomenon [18]. It was to mitigate this kind of issue that we decided, for example, to place demographics questions last. Respondents may have misunderstood some questions or questions options. We took many measures to make questions as clear as possible (using simple English, collaboratively writing and proofreading questions, using beta testers), but, despite that, we became aware after launching the survey that a few questions were still ambiguous, or did not have the originally intended meaning.

7 Discussion and Conclusions

We presented the design, deployment, and summaries and analyses of the responses to the Coq Community Survey 2022. Besides providing up-to-date descriptive data on the Coq community and decision support for the Coq Team, we hope our work can be useful for constructing future surveys targeted at ITP communities. During analysis of tentative future surveys, changes could be tracked for recurring questions and results of our regression analyses, e.g., on users performing software verification, could be replicated.

As part of the supplementary material to our paper [12], we provide 1) the survey definition in LimeSurvey and HTML format, 2) the Jupyter Notebook and Stata code we used to analyze the response data, 3) plots of answers to all closed questions and plots of some interactions between questions, 4) manual analysis of some open-text questions, and 5) answers to open-text questions. In particular, the code in our supplementary material shows how we generated the plots in this paper. However, for GDPR conformance, we do not include the raw response data that could fully replicate plots and other analyses.

Planning, running, and analyzing the survey was work intensive. The WG decided to announce a retention period of one year for the raw survey response data up front. However, this turned out to be a limiting decision in terms of the WG's time and resources. Keeping raw data is a delicate matter, since respondents and their answers may sometimes be identified by correlating survey response data with data from other sources. Nevertheless, we believe asking respondents for permission to store their data for more than one year is warranted, due to the time required for rigorous response analysis.

An important problem with recurring surveys is responder retention, as highlighted by the significant drop in the number of responses to the OCaml community survey between 2020 and 2022 [21]. Three years or more may be needed between surveys in an ITP community to ensure there will be enough responses to repeat previous analyses in a reliable way. In a small community, if filling in a survey becomes perceived as time-consuming routine work of little value, many members may not volunteer for the task. Making most or all questions optional does not address this issue.

The survey included some open-ended questions, whose responses we do not discuss in this paper. Future surveys could code the answers to these questions and include new multiple-choice questions. Since our analyses indicate that experience has a large impact on Coq user behavior, future research could investigate in more detail how users gain experience, and the effectiveness of their methods to improve. Another avenue is to look in detail at methodological problems that users face when applying Coq in their domain, and how users develop workarounds and solutions to these problems.

References

1. Ana de Almeida Borges, Jean-Rémy Falleri, Jim Fehrle, Emilio Jesús Gallego Arias, Érik Martin-Dorel, Karl Palmskog, Alexander Serebrenik, and Théo Zimmermann. Coq Community Survey 2022: Summary of Results, abstract. The Coq Workshop 2022, August 2022. URL: https://coq-workshop.gitlab.io/2022/abstracts/Coq2022-04-01-community-survey.pdf.
2. Ana de Almeida Borges, Jean-Rémy Falleri, Jim Fehrle, Emilio Jesús Gallego Arias, Érik Martin-Dorel, Karl Palmskog, Alexander Serebrenik, and Théo Zimmermann. Coq Community Survey 2022: Summary of Results, slides. The Coq Workshop 2022, August 2022. URL: https://coq-workshop.gitlab.io/2022/slides/Coq2022-04-01-community-survey.pdf.
3. Andrew W. Appel. Coq's vibrant ecosystem for verification engineering (invited talk). In *Proceedings of the 11th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2022, pages 2–11, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3497775.3503951.
4. Jasmin Christian Blanchette. Formalizing the metatheory of logical calculi and automatic provers in Isabelle/HOL (invited talk). In *International Conference on Certified Programs and Proofs*, CPP 2019, pages 1–13, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3293880.3294087.
5. Thomas Braibant. Coq survey, 2014. URL: https://github.com/braibant/coq-survey/blob/master/popl-coq.pdf.
6. Damian Clarke, Joseph P. Romano, and Michael Wolf. The Romano–Wolf multiple-hypothesis correction in Stata. *The Stata Journal*, 20(4):812–843, 2020. doi:10.1177/1536867X20976314.
7. The Coq Survey Working Group. Coq Community Survey 2022 in Chinese, 2022. URL: https://thzimmer.gitlabpages.inria.fr/coq-survey-2022-assets/LimeSurvey/questionnaire_chinese.html.
8. The Coq Survey Working Group. Coq Community Survey 2022 in English, 2022. URL: https://thzimmer.gitlabpages.inria.fr/coq-survey-2022-assets/LimeSurvey/questionnaire_356388_en.html.
9. The Coq Survey Working Group. Coq Community Survey 2022 Results: Part I (blog post), 2022. (Who is using Coq and in what context?). URL: https://coq.discourse.group/t/coq-community-survey-2022-results-part-i/1730.
10. The Coq Survey Working Group. Coq Community Survey 2022 Results: Part II (blog post), 2022. (How people are using Coq? — OS, IDEs, CI/CD). URL: https://coq.discourse.group/t/coq-community-survey-2022-results-part-ii/1746.
11. The Coq Survey Working Group. Coq Community Survey 2022 Results: Part III (blog post), 2022. (How is Coq used? — features, tools, libraries). URL: https://coq.discourse.group/t/coq-community-survey-2022-results-part-iii/1777.
12. Ana de Almeida Borges, Annalí Casanueva Artís, Jean-Rémy Falleri, Emilio Jesús Gallego Arias, Érik Martin-Dorel, Karl Palmskog, Alexander Serebrenik, and Théo Zimmermann. Supplementary material for the article "Lessons for Interactive Theorem Proving Researchers from a Survey of Coq Users", May 2023. doi:10.5281/zenodo.7930567.
13. Paul Duvall, Steve Matyas, and Andrew Glover. *Continuous Integration: Improving Software Quality and Reducing Risk*. Addison-Wesley Professional, first edition, 2007.

14 Adrien Fabre. Tie-breaking the highest median: alternatives to the majority judgment. *Social Choice and Welfare*, 56(1):101–124, 2021.

15 Taylor Fausak. State of Haskell survey, 2022. URL: https://taylor.fausak.me/2022/11/18/haskell-survey-results/.

16 Sacha Greif and Eric Burel. State of JS, 2022. URL: https://2022.stateofjs.com/en-US/.

17 Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL: http://www.jstor.org/stable/4615733.

18 Glenn D Israel and CL Taylor. Can response order bias evaluations? *Evaluation and Program Planning*, 13(4):365–371, 1990.

19 Mark L. Mitchell and Janina M. Jolley. *Resarch Design Explained*. Wadsworth, Cengage Learning, 7th edition, 2010.

20 OCaml Software Foundation. OCaml user survey, 2020. URL: https://discuss.ocaml.org/t/suggestions-from-the-ocaml-survey-result/6791.

21 OCaml Software Foundation. OCaml users survey, 2022. URL: https://ocaml-sf.org/docs/2022/ocaml-user-survey-2022.pdf.

22 Stephen O'Grady. The RedMonk programming language rankings: June 2022, October 2022. URL: https://redmonk.com/sogrady/2022/10/20/language-rankings-6-22/.

23 opam development team. OCaml package manager, 2023. URL: https://opam.ocaml.org.

24 Karl Palmskog, Enrico Tassi, and Théo Zimmermann. Reliably reproducing machine-checked proofs with the Coq Platform. In *Workshop on Reproducibility and Replication of Research Results*, 2022. URL: https://arxiv.org/abs/2203.09835.

25 Benjamin C. Pierce, Chris Casinghino, Michael Greenberg, Vilhelm Sjöberg, and Brent Yorgey. *Software Foundations*. Electronic textbook, 2011. Japanese translation. URL: http://proofcafe.org/sf/.

26 Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hriţcu, Vilhelm Sjöberg, and Brent Yorgey. *Logical Foundations*, volume 1 of *Software Foundations*. Electronic textbook, 2022. Chinese translation, version 5.7. URL: https://coq-zh.github.io/SF-zh/lf-current/index.html.

27 Joseph P Romano and Michael Wolf. Stepwise multiple testing as formalized data snooping. *Econometrica*, 73(4):1237–1282, 2005.

28 Stack Overflow. Stack Overflow developer survey, 2022. URL: https://survey.stackoverflow.co/2022/.

29 The Coq Development Team. opam archive for Coq, 2023. URL: https://github.com/coq/opam-coq-archive.

30 Mairieli Wessel, Alexander Serebrenik, Igor Wiese, Igor Steinmacher, and Marco A. Gerosa. What to expect from code review bots on GitHub? a survey with OSS maintainers. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*, SBES '20, pages 457–462, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3422392.3422459.

31 James R. Wilcox. Why is the Coq logo made to look like a penis?, April 2021. URL: https://sympa.inria.fr/sympa/arc/coq-club/2021-04/msg00006.html.

32 Jeffrey M Wooldridge. *Introductory econometrics: A modern approach*. Cengage learning, 6th edition, 2015.