


How the R community creates and curates knowledge: an extended study of stack overflow and mailing lists

Alexey Zagalsky¹ · Daniel M. German¹  ·
Margaret-Anne Storey¹ · Carlos Gómez Teshima¹ ·
Germán Poo-Caamaño¹

© Springer Science+Business Media, LLC 2017

Abstract One of the effects of social media's prevalence in software development is the many flourishing communities of practice where users share a common interest. These large communities use many different communication channels, but little is known about how they create, share, and curate knowledge using such channels. In this paper, we report a mixed methods study of how one community of practice, the R software development community, creates and curates knowledge associated with questions and answers (Q&A) in two of its main communication channels: the R tag in Stack Overflow and the R-Help mailing list. The results reveal that knowledge is created and curated in two main forms: participatory, where multiple users explicitly collaborate to build knowledge, and crowd-sourced, where individuals primarily work independently of each other. Moreover, we take a unique approach at slicing the data based on question score and participation activities over time. Our study reveals participation patterns, showing the existence of prolific contributors: users who are active across both channels and are responsible for a large proportion of the answers, serving as a bridge of knowledge. The key contributions of this paper are: a characterization of knowledge artifacts that are exchanged by this community of practice; the

Communicated by: Romain Robbes, Christian Bird, and Emily Hill

✉ Alexey Zagalsky
alexeyza@uvic.ca

Daniel M. German
dmg@uvic.ca

Margaret-Anne Storey
mstorey@uvic.ca

Carlos Gómez Teshima
cagomez@uvic.ca

Germán Poo-Caamaño
gpoo@uvic.ca

¹ University of Victoria, Victoria, BC, Canada

reasons why users choose one channel over the other; and insights on the community participation patterns, which indicate an evolution of the community and a shift from knowledge creation to knowledge curation.

Keywords Mining software repositories · Empirical study · Qualitative study · Survey · Stack overflow · R · Mailing list

1 Introduction

The emergence and adoption of socially enabled tools and channels (e.g., GitHub, Stack Overflow, mailing lists) has fostered the formation of large *communities of practice* where users share a common interest, such as programming languages, frameworks, and tools (Storey et al. 2014). These communities rely on many different communication channels, but little is known about how they create, share, and curate knowledge using such channels.

One prominent community of practice is the group that has formed in support of the R programming language, an open source project without commercial backing that relies heavily on its rapidly growing and highly heterogeneous software development community. The R community plays an important role in diffusing the R language: users have access to numerous resources for learning the language and receiving help, such as mailing lists, blogs, books, online and offline courses, and question & answer sites (e.g., Stack Overflow). While the R community benefits from this vast and rich corpus of knowledge, it also drives the creation and curation of the information.

Without a single entity directing and controlling it, the R language has grown organically from its community. Similar to other communities of practice, knowledge is exchanged and curated in many communication channels, and two particular communication channels are at the center of this process: the *R-help mailing list* and *Stack Overflow*. The R-help mailing list was created to assist those using the language, and while Stack Overflow is not specifically oriented towards R, its section dedicated to R (the R tag) has grown rapidly.¹

Stack Overflow has revolutionized the way programmers seek knowledge (Li et al. 2013; Vasilescu et al. 2014), assuming the role of a capable “expert on call” that is able—and willing—to answer questions of any level of difficulty about any programming technology (R included). Stack Overflow’s gamification features normally guarantee that enthusiastic experts will answer questions, often within minutes of being posted (Mamykina et al. 2011). Equally important is the ability of Stack Overflow’s users to curate the knowledge being created, making sure that the best answers surface to the top and become a valuable asset to those seeking an answer now or in the future. Stack Overflow has become a popular and effective tool for creating, curating, and exchanging knowledge, including knowledge about the R language.

One would expect that the traffic on the R-help mailing list would begin to fizzle as Stack Overflow popularity increased. If Stack Overflow is so effective at matching those who seek knowledge with those that have it, doesn’t that obviate most of the need for the R-help mailing list? Yet that does not appear to be the case as the R-help mailing list has maintained a steady level of activity, implying that it is still an important resource for the R community. In fact, it appears as if the mailing list and Stack Overflow complement each other.

¹<http://www.r-bloggers.com/r-is-the-fastest-growing-language-on-stackoverflow/>

There are obvious inherent differences between both communication channels. On the one hand, mailing lists unite users by subscription, creating a tight community, but their content lacks organization, except for the natural structure provided by email metadata (e.g., subjects, threading, authors, dates), and they are not optimized for long-term storage and retrieval. On the other hand, Stack Overflow's community is not as tight as the R-help mailing list community, but the channel is optimized for the curation and long-term storage of knowledge. However, little is known about the differences in how people use both communication channels, such as how the types of questions and answers sought in one channel compare to the other, why users choose one channel over the other, why some users participate in both channels, and how participants perceive each communication channels.

In this extended study, we first focus on characterizing knowledge artifacts (we presented the initial research phases in our earlier paper, Zagalsky et al. 2016). We empirically compare how knowledge, specifically knowledge manifested as questions and answers, is sought, shared, and curated on both the R-help mailing list and Stack Overflow. We then build on these findings and focus on the knowledge curation process in the R community. We examine the participation patterns and behavior of users in both sub-communities, and seek to learn more about community's prolific members and knowledge curators. Our research employed a mixed methods *exploratory case study* methodology to answer the following research questions:

- RQ1** What types of knowledge artifacts are shared on Stack Overflow and the R-help mailing list within the R community?
- RQ2** How is the knowledge constructed on Stack Overflow and the R-help mailing list?
- RQ3** Why do users post to a particular channel and why do some post to both channels?
- RQ4** How do users participate on both channels over time?
- RQ5** Are there significant differences in participation activity between community users?

By mining archival data, we identified and categorized the main types of knowledge artifacts found on the R-help mailing list and in Stack Overflow (RQ1). The emerging categories form a *typology* (see Table 2) that allows researchers to study and characterize Q&A knowledge dissemination within a community of practice. We used this typology to study how knowledge is constructed and shared on Stack Overflow and the R-help mailing list. We found that these channels support two distinct approaches for constructing knowledge—*participatory knowledge construction* and *crowd knowledge construction*—however, each channel supports them differently (RQ2). Our findings indicate that participatory knowledge construction is more prevalent on the R-help mailing list, while crowd knowledge construction is more prevalent on Stack Overflow.

We found that some contributors are active on both channels. As a result, we conducted a survey to investigate the benefits they gain by doing so (RQ3). But beyond that, we wanted to examine how participation differs between Stack Overflow and the R-help mailing list over time and how long users participate on the two channels (RQ4). Additionally, we wanted to understand the behavior and *participation patterns* of the contributing users (RQ5). We focused on several sets of contributors: those who rarely contribute, the top contributors, and those who contribute to both channels. Our results show that a great majority of participants are fleeting and a small number of individuals are responsible for most answers. Furthermore, our findings indicate that both channels are reaching maturity: for the R-help mailing list, this means a steady flow of questions; for Stack Overflow, there is a continuous decrease of questions with a positive score (number of positive votes minus negative votes), hinting to the fact that, as time progresses, the most sought after questions have already been asked.

The findings we report and discuss in this paper show how channel affordances and community rules (e.g., topic restriction and gamification) influence knowledge construction and curation. This has implications on other open source projects or companies and should be considered by those that are thinking of using Stack Overflow instead of or in addition to email for knowledge sharing. This information can also help guide which behaviour patterns project or community leaders should monitor over time.

2 Background

The R project² was born in 1993 as a free and open source programming language and software environment for statistical computing, bioinformatics, and graphics (Ihaka and Gentleman 1996). R's popularity has continuously increased over the years: in 2016, IEEE Spectrum ranked it as the 6th most popular language.³

The R community is composed of: (1) *R-core*, a team of 20 software developers that maintain and evolve the R language; and (2) *Periphery*, which includes everyone else (language users and package developers).

The R community is an eclectic open source community that goes beyond software development and includes biologists and statisticians with no or limited programming experience. Its entire history of mailing list communication is archived and publicly available. The R community has also been the subject of extensive research in community evolution (German et al. 2013; Vasilescu 2014) and the interplay between channels (Vasilescu et al. 2014).

Our study focused on the analysis of Stack Overflow and the R-help mailing list, two channels in the R community. We chose them because they are the main channels that provide Q&A support to the community.

2.1 The R-help Mailing List

There are several mailing lists to help R community users solve programming problems with the R language: *R-help*, *R-package-devel*, *R-devel*, *R-packages*, *R-announce*, and *Bioconductor*. However, R-help is the main mailing list for discussing problems and solutions using R. Other messages are also encouraged, such as documentation, benchmarks, examples, and announcements.

The R-help mailing list used to be the main communication channel for asking and answering questions within the R community, but a significant number of users migrated to Stack Overflow (Vasilescu et al. 2014). Despite the reduced number of users, the R-help mailing list is still very active—on average, a subscriber may receive approximately 25 emails a day (as of October 2016).

2.2 Stack Overflow

In contrast to the R-help mailing list, Stack Overflow incorporates a rich visual and user-friendly interface with social media and gamification features. The social aspect of the

²<https://www.r-project.org/>

³<http://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>

website improves participation and provides strong support for creating and sharing knowledge as well as encouraging informal mentorship (Jenkins 2009; Storey et al. 2014). Meanwhile, Stack Overflow's gamification features provide reputation points and badges to reward user participation and earn them points that enable functionality inside the site. It has been reported that Stack Overflow's gamification mechanisms boost participation (Vasilescu 2014) and enable mutual assessment (Singer et al. 2013).

2.3 Stack Overflow vs. Mailing Lists

Software development is a knowledge-building process (Naur 1985). Due to the emergence of socially enabled tools and channels and the formation of communities of practice (Storey et al. 2014), it is important to understand how knowledge is created and shared within these communities. In this study, we focus on knowledge in the form of questions and answers within the R community.

Other researchers have also examined communities using Stack Overflow and R-help. As part of a study on the transition to gamified environments, Vasilescu (2014) examined the popularity of Stack Exchange (including the Stack Overflow R tag) and mailing lists within the R community. He found that the number of message threads on the R-help mailing list had decreased since 2010, while the number of R-related questions asked on the Stack Exchange network had increased. Vasilescu also examined the *difference in activity* between contributions made by users active on both channels and users focused on a single medium. Similar to Vasilescu, Squire (2015) studied a project's *transition to the Stack Overflow gamified channel*. She focused on examining whether four software projects that moved from mailing lists to Stack Overflow showed improvements in terms of developer participation and response time. She found that all four projects showed improvements on Stack Overflow compared to mailing lists. However, she also found that several projects moved back to using mailing lists despite achieving these improvements. The reasons for moving back included poor support for discussion in Stack Overflow and also closing of questions that were thought to be relevant but were not suited to Stack Overflow.

In our study, we examined Stack Overflow's R tag and the R-help mailing list to better understand the *knowledge types* used. This allowed us to characterize the different approaches for seeking and sharing knowledge on each channel. We found that both channels have knowledge support for question and answers, however, there are important differences between the two channels. For example, Stack Overflow's competitive environment gives more reputation points⁴—a sought after reward—to those who have an answer accepted than to those participating in other activities (e.g., editing). Additionally, upvoted questions and answers can receive many points over time if the question is frequently voted up. Other forms of participation (e.g., commenting) receive badges,⁵ which are finite and have less visibility than the number of reputation points achieved. For these reasons, we believe that participants have an incentive to be the first to provide the correct answer rather than improve other answers and participate in discussions. Moreover, we also found users of the R community that were active on both channels. As a result, we sought to understand *why users post to a particular channel*, and then understand *user participation activity*

⁴<https://stackoverflow.com/help/whats-reputation>

⁵<https://stackoverflow.com/help/badges>

over time. This raises important questions about the role of newcomers, prolific members, contributors, and curators in communities of practice.

2.4 Community Participation

A community of practice is the embodiment of its members, their shared knowledge, relations and social interactions, and the activities that foster learning and participation. Wenger et al. (2009) wrote about communities of practice from the perspective of learning, focusing on the role of technology in the formation of such ‘digital habitats’: “While there is no question that digital habitats can give rise to new communities—by connecting people across time and space, by creating new spaces for engagement, by revealing affinities for shared domains, and by providing information about people—we need to make a clear distinction between the technology and the social conditions and processes that bring a community together. Just because the technological container remains, it does not mean the community is still functioning and alive.” In this study, we examine not only the channels and knowledge artifacts, but also the community participants and their activities.

We explore participation patterns in the R community and focus on two specific types of participants: *newcomers* and *prolific members*. Both groups of participants are important for a community’s growth and well-being. Through a process of formal or informal mentorship, newcomers begin with simple peripheral tasks and participate in limited activities (physically or socially), gradually doing more until they become experienced members. “The social structure of this practice, its power relations and its conditions for legitimacy define possibilities for learning” (Lave and Wenger 2002). Through these peripheral activities, novices gain ‘legitimacy’ and become acquainted with the community’s rules, norms, principles, and vocabulary.

Experienced and prolific members tend to be very active within the community; they are often experts who want to keep the system clean with valuable content. Many of these members often serve as moderators or *knowledge curators* (referred to as “caretakers” by Srba and Bielikova 2016), bridging between different groups of people and nurturing the community. In the R community, we found that a small group of prolific members are responsible for a large proportion of answers.

3 Methodology

The main goal of this work was to empirically compare how knowledge, specifically knowledge in question-and-answer (Q&A) form, is sought, shared, and curated on both the R-help mailing list and the R tag on Stack Overflow. We also aimed to gain insights on the *knowledge curation process* used within the R community (e.g., participation patterns and behaviors). We used a mixed methods *exploratory case study* methodology (Creswell 2009; Runeson et al. 2012) to answer our research questions (presented in Section 1) and we describe our research process timeline in Fig. 1.

This study employed two research methods over three phases: we *mined archival data* and conducted a *qualitative survey*. In the first phase, we sampled and qualitatively analyzed the mined data to characterize the types of discussions that occur. In the second phase, we surveyed members of the R community to validate our interpretation of the results from the previous phase. And in the third phase, we conducted a quantitative analysis of the archival data and focused on investigating participation patterns on both channels.

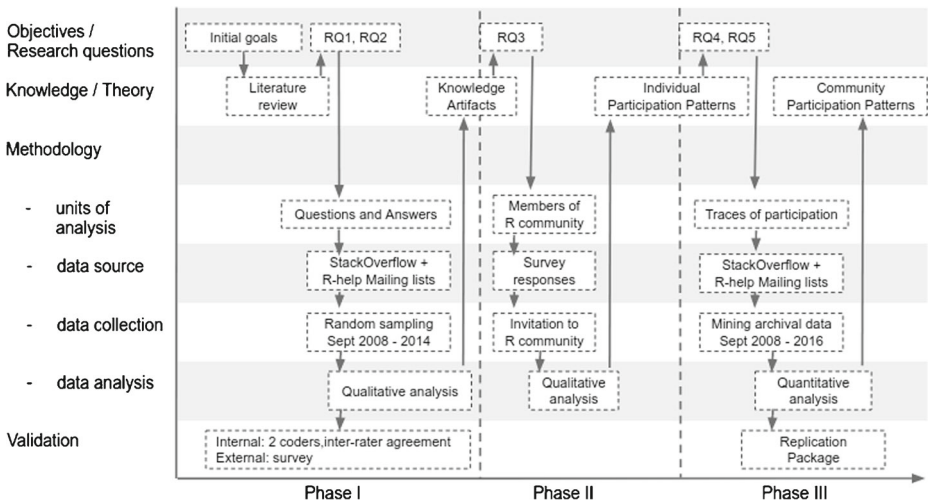


Fig. 1 A timeline of our research process

3.1 Phase I: Characterizing Types of Knowledge Artifacts

We mined data from the public archives of both the R-help mailing list and Stack Overflow. The R-help mailing list archive started in 1997, while the archives for Stack Overflow started in 2008 (when it was created). To make the datasets comparable, we analyzed both datasets from September 2008 until September 2014, a period of time that both channels were available.⁶

3.1.1 Data Collection Process

For the Stack Overflow data, Stack Exchange releases a new data dump of all their Websites every three months.⁷ We used only the questions, comments, and answers that included the tag `r` or its two synonyms⁸ (`rstats` and `r-language`), and information about the users who contributed to them.

For the R-help mailing list data, we retrieved MBOX files of the mailing list archives from the R-help website. We downloaded all the threads and extracted the separate emails and corresponding metadata from each thread. As opposed to Stack Overflow, the R-help mailing list data doesn't include metadata to indicate whether a message is a question or an answer. Thus we used the following heuristic to classify the R-help messages: if a message contained an `In-Reply-To` or a `References` header, it was classified as an answer (225,254 responses); otherwise, a message was classified as a question (119,145 questions).

To study participation and determine which users were active in both channels, we assigned them a unique person identifier and performed a unification of identities. We extracted email addresses from the R-help mailing list data using the `From` field, and using a

⁶In the third phase of our study, we extended the mined datasets up to September 2016

⁷<http://stackexchange.com/sites>

⁸<http://stackoverflow.com/tags/r/synonyms>

Message	Channel	Question	Answer	Update	Comment	Flag	Resource	Knowledge construction	Memos
MessageId: 1716012 Subject: Stopwatch function in R Date: 2009-11-11	SO	Environment	-	Expansion; Non-labelled	-	-	Official documentation: Expand	Participatory	Software development questions Solving an error Should be flagged as Debuggin Many answer
MessageId: 1340054801327-4633754.post Subject: [R] (1-1e-100)=-1 true?	RH	Discrepancy	-	-	-	-	-	Crowd	Well explained question

Fig. 2 Example of data coding. Each row is a threaded message. Questions, comments, and answers are identified with the number in the first column. Columns in yellow (columns 4-10) contain the code for each message type. The last two columns contain the memos and URLs

conservative approach, identified messages sent by the same person but from different email addresses. As a result, the number of unique individuals identified on R-help was reduced from 36,600 to 31,729 (a reduction of 15%)—a similar number was reported by Vasilescu et al. (2014). Next, by using MD5 hashes of the email addresses extracted from the R-help mailing list and Stack Overflow, we identified 1,449 persons who used both media channels.

To prepare and process the data, we wrote our own scripts (these are included in the replication package⁹). To ensure accurate results when processing the R-help mailing list, we followed a series of recommendations proposed by Bettenburg et al. (2009): extracted messages, removed duplicates, removed signatures, and reconstructed discussion threads. We unified identities of contributors in R-help using our own email unification tool.¹⁰ To identify people common to both channels, we compared the hash of the email in the Stack Overflow data against the hash of every email used by a person. If there was a match, we considered the person in R-help to be the same as the person in Stack Overflow.

3.1.2 Data Analysis Process

We followed an inductive approach (Runeson et al. 2012) to analyze the data from Stack Overflow and the R-help mailing list. To reduce the risk of bias (Runeson et al. 2012), the analysis was conducted by two computer scientists with a background in qualitative data analysis. To answer RQ1 and RQ2, we randomly sampled and iteratively coded questions in both channels to characterize the types of discussions that occur. This process was continued until we reached thematic saturation (Bowen 2008), i.e., no new themes were identified and no issues were raised regarding existing categories, which amounted to 400 threads in each channel. To answer RQ3, we focused on questions with identical subjects that were posted to both channels by the same author: we found and analyzed 79 such threads.

We used memoing, affinity diagrams, and a code book to support the data analysis process. We wrote reflective memos in a spreadsheet next to the applicable codes (see example in Fig. 2). These memos were used to create the codes and hypotheses about the relationships between concepts. We coded in multiple sessions, which allowed us to iteratively refine the definitions in the code book. Each entry is associated with a title, a formal definition, an example, and notes from the researchers. For inter-rater reliability, we used the Cohen Kappa inter-rater agreement coefficient (Stemler 2004). Although it is suggested that one should aim for coefficient values above 0.6 to obtain substantial results (Landis and Koch 1977), we aimed for 0.8 or above based on our previous experience with this method

⁹Our scripts, sample data, and coded data are openly available at <https://zenodo.org/record/831805>

¹⁰<https://github.com/dmgerman/unify-perl>

(Gomez et al. 2013). We used this coefficient after each coding session as a way to trigger discussion and to further refine the codes if necessary.

The analysis process required an *understanding of the context* surrounding each message. The process consisted of: (1) gathering the required information from each channel (i.e., the message analyzed, the relevant thread), and (2) mapping the messages from each channel to a specific knowledge type (see Section 4.1). The mapping was necessary as each channel contained a different data structure. We defined the following mappings between messages in both channels:

- Question:** The message is the first in the thread and contains the main question.
- Answer:** The message provides a solution to the main question in the thread.
- Update:** The message provides a modification to a question or an answer made by the author of said question or answer.
- Comment:** The message offers clarification to a specific part of the question or answer.
- Flag:** The message requests attention from the moderator or other community members (e.g., repeated questions, spam, or rude behavior).

As opposed to Stack Overflow, where the metadata indicates the corresponding mapping, for R-help we needed to manually examine the context of each message and classify them based on the knowledge artifact definitions. We elaborate on these definitions in Section 4.1.

3.2 Phase II: Exploring Why Users Post to a Particular Channel

The analysis from Phase I revealed that some developers are active on both channels, and in some cases, even post the same questions. To further understand this phenomena and explore the perceived benefits of using one channel over the other, we conducted a survey with users of the R community.¹¹ To test and refine the questions, format, and tone, we piloted the survey twice. We promoted our survey on Twitter, Reddit, the R-help mailing list, and Meta Stack Exchange to reach users of both channels and minimize selection bias. However, our survey invitation on Stack Exchange was deemed off topic and deleted a few minutes later. In total, we received 37 responses, 26 of which were valid (invalid responses occurred if the session ended or the participant did not complete the survey).

3.3 Phase III: an Extended Investigation of Participation Patterns

In this phase, we focused our analysis on the behavior and participation patterns of community users on the R-help mailing list and Stack Overflow. Since this phase was conducted two years after our initial phase, we extended our analysis to include archival data from the beginning of each channel up to September 2016 (for both channels). Table 1 depicts a summary of the data used for this phase of the study.

As before, we needed to compare identities between the two communication channels. However, due to privacy concerns, Stack Overflow has now removed all email information from their current dumps (the field is present but empty). Previously, Stack Overflow included the hash of email addresses. For this reason, we used two different data dumps of Stack Overflow: the first one was dated September 2014 (which contains the SHA of the email addresses); the second was dated in March 2017 (which does not contain any participant email information). We used the first dataset only when comparing identities between

¹¹ A copy of the survey is available at <http://cagomez.com/lime/index.php/857211?lang=en>

Table 1 Raw data collected for each channel, up to September 2016

Type	R-help	Stack overflow
Questions	124,791	150,707
Answers	150,919	204,468
Comments	88,685	617,460
Different individuals	31,699	63,372

the two communication channels, while for the rest of the analysis, we used data from both channels (up to September 2016).

4 Findings

To understand how knowledge in the form of questions and answers is created, shared, and curated, we first identified and categorized the main types of knowledge artifacts contained within messages on the R-help mailing list and the Stack Overflow ‘R tag’ (RQ1). The emerging categories formed a typology and allowed us to identify and describe two approaches for constructing the knowledge supported by these channels (RQ2). Interestingly, we found that some developers are active on both channels, and in some cases, even post the same questions. As a result, we investigated the benefits they gain by doing so (RQ3). We also present our findings about participation on the two channels over time (RQ4) and look closely at the different levels of participation activity between community users (RQ5).

4.1 What Types of Knowledge Artifacts are Shared on Stack Overflow and the R-help Mailing List

To answer RQ1, we randomly sampled message threads from both Stack Overflow and the R-help mailing list, where each thread included a question and the associated responses. We identified five types of artifacts that capture knowledge: (1) Questions; (2) Answers; (3) Updates; (4) Flags; and (5) Comments. Through our analysis, we further divided these types into subtypes.

Table 2 presents our typology of knowledge artifacts, their descriptions, and their frequency in the data sample. Even though we did not aim for a statistically significant sample size, the size of this sample (400 threads in each channel) guarantees a confidence level of approximately $95\% \pm 5\%$ for both channels. Using the Chi-square test of independence, we tested whether the frequency distribution of the types and subtypes of questions was different between the two channels. Specifically, we tested if the distribution frequency of each subtype (as shown in Table 2) was statistically different between R-help and Stack Overflow. We also compared the overall frequency for each artifact subtype. In all cases, the distributions were found to be statistically different (with $\rho \ll 0.001$ in all cases).

Questions and Answers: Questions express one or more problems or concerns faced by a user on the R-help mailing list or on Stack Overflow, whereas answers represent solutions to questions. We observed that the types of questions on Stack Overflow are more specific than those on the R-help mailing list and are more likely to consist of tutorials. Stack Overflow also contains more answers per question: 2 per question compared to 1.4 for R-help (see

Table 2 Typology of knowledge artifacts found on both Stack Overflow (SO) and the R-help (RH) mailing list, their frequency, and relative proportion in the analyzed sample

		SO	RH	Prop SO	Prop RH
Questions					
<i>How-to</i>	Asks how to do something specific.	166	103	41.50%	25.75%
<i>Discrepancy</i>	Asks about an unexpected result of a specific function, process, or package.	53	88	13.25%	22.00%
<i>Conceptual/Guidance</i>	Asks for conceptual clarification or guidance on topics related to R or statistics.	48	49	12.00%	12.25%
<i>Bug/Error/Exception</i>	Asks for a solution to or reasons for an error message.	27	48	6.75%	12.00%
<i>Decision help</i>	Asks for advice in making a decision.	36	35	9.00%	8.75%
<i>Code reviewing</i>	Asks for a code review, explicitly or implicitly.	34	21	8.50%	5.25%
<i>Set-up</i>	Asks for possible ways to set up the R environment before or after deployment.	15	31	3.75%	7.75%
<i>Non-functional</i>	Asks for help (or suggestions) with a non-functional requirement such as performance or memory usage.	14	11	3.50%	2.75%
<i>Future reference</i>	Asks a question (often self-answering it) that might not exist on the channel, but that is interesting enough to warrant a thread for future reference.	5	4	1.25%	1.00%
<i>Other</i>	Asks for assistance unrelated to the channel, or the message contains unrelated information (e.g., announcements, ideas for improvement).	2	10	0.50%	2.50%
	Total	400	400	100%	100%
Answers					
<i>Explanation</i>	Provides an explanation of an approach that answers the question and lists steps on how to implement it.	203	101	25.15%	17.44%
<i>Source code</i>	Provides a source code snippet as a solution without an extensive explanation about the answer.	198	102	24.54%	17.62%
<i>Redirecting</i>	Provides a link to an existing solution that is not in the thread (e.g., external application, tutorial, project).	163	87	20.20%	15.03%
<i>Clue/Hint/Suggestion</i>	Provides a possible way to fix the issue without actually solving it.	43	105	5.33%	18.13%
<i>Alternative</i>	Provides a different approach to a solution that is related to but not exactly what is being asked (e.g., mathematical approach, data structure modification).	33	98	4.09%	16.93%
<i>Tutorial</i>	Provides a set of steps to teach people how to solve the issue.	105	15	13.01%	2.59%
<i>Announcement</i>	Provides a notification about some artifact (e.g., packages, libraries).	8	33	0.99%	5.70%

Table 2 (continued)

		SO	RH	Prop SO	Prop RH
<i>Opinion</i>	Provides an opinion or an expansion of another answer by including scenarios and examples.	49	35	6.07%	6.04%
<i>Benchmark</i>	Provides a benchmark of multiple solutions posted by others or compares different answers.	5	3	0.62%	0.52%
	Total	807	579	100%	100%
Updates					
<i>Expansion</i>	Expands the question or answer by providing scenarios or examples.	116	83	18.92%	33.60%
<i>Correction</i>	Corrects format, grammar, spelling, and semantic mistakes.	301	2	49.10%	0.81%
<i>Explanation</i>	Explains or clarifies a specific point in the question or answer, such as why the user chose a specific data structure, or the meaning of a variable.	83	95	13.54%	38.46%
<i>Announcement</i>	Announces specific events (e.g., bounties, future updates).	27	3	4.40%	1.21%
<i>Background</i>	Adds additional context to the question or answer.	74	57	12.07%	23.08%
<i>Solution</i>	The user answers their own question.	12	7	1.96%	2.83%
	Total	613	247	100%	100%
Flags					
<i>Repeated question</i>	Notifies a user that the question has been answered previously.	48	8	59.26%	14.81%
<i>Off-topic/ Opinion</i>	Identifies questions that are unrelated to the channel's interests, or requests answers based on opinion.	22	19	27.16%	35.19%
<i>Not an answer</i>	Indicates answers that are out of scope of the question or that do not answer the question.	0	27	0.00%	50.00%
<i>Too localized</i>	Indicates questions that are too specific and might not help future readers.	6	0	7.41%	0.00%
<i>Unclear</i>	Indicates questions that are difficult to understand.	5	0	6.17%	0.00%
	Total	81	54	100%	100%
Comments					
<i>Correction/ Alternative</i>	Suggests a change to a question or answer, offers an alternative solution or a correction.	102	89	18.15%	33.33%
<i>Expansion</i>	Provides additional information.	127	65	22.60%	24.34%
<i>Compliment/ Critic</i>	Posts something good, offers thanks, or provides an opinion or criticism.	157	52	27.94%	19.48%
<i>Clarification</i>	Provides (or requests) additional information about a question or answer.	98	28	17.44%	10.49%
<i>External reference</i>	References an external resource.	78	33	13.88%	12.36%
	Total	562	267	100%	100%

Table 2). However, R-help answers tend to offer more suggestions or alternatives than Stack Overflow answers.

Updates: An update is a modification of a question or an answer. In Stack Overflow, updates are presented in one of two ways:

Labeled updates are explicitly shown in the body of questions and answers next to a label that identifies the update (e.g., edit, update, and p.s.). When multiple update labels appear in a message, each label is accompanied by a number (e.g., “[Edit 1:]”), a date (e.g., “Edit/Update (April 2011):”), or a bulleted list (e.g., “EDIT: - anova... -drop1...”).

Non-labeled updates are only visually recognizable through the message history system. The only indication of a change is a box at the end of a message that identifies the user who performed the change and the date when it occurred.

We found that non-labeled updates are often used to correct formatting, grammar, semantic mistakes, and spelling, or to incorporate explanations, examples, and suggestions without changing the meaning of the question or answer. Labeled updates are for everything else.

On the R-help mailing list, all communication occurs through email, and authors do not explicitly tag messages as updates. For this reason, we define an update on R-help as *a message sent to a thread where the author has already participated once*.

Regarding update frequency in our sample, the Stack Overflow R tag contained 2.5 times more updates than the R-help mailing list. Corrections are more common on Stack Overflow (almost 50%), while R-help updates are often related to the adding of information to a thread (providing background, expansion, and explanation).

Flags: Flags are used to alert users that a question or an answer does not match community expectations. Stack Overflow contains a flagging mechanism that’s often used to get a moderator’s attention. These flags can accomplish various objectives: mark a message as containing spam or rude/abusive behavior, or identify duplicate questions, off-topic messages, unclear questions, opinion-based questions, and low-quality answers. Depending on the type of flag, this can lead to a thread being closed or the loss of user reputation points.

The R-help mailing list doesn’t have a built-in flagging mechanism. However, R-help users utilize the concept of flags, which we define as *messages used to call the attention of other community users*, similar to the way flags are used in Stack Overflow. For example, a community member indicated that a question is off topic by responding that “*the main questions here are not R-related, but statistical modeling questions, and much too broad for the R list*”.

In terms of their frequency, R tag posts on Stack Overflow contained 1.5 times more flags than posts on the R-help mailing list. Stack Overflow flags are primarily used to mark repeated questions. In contrast, flags on R-help are often used to indicate that a previous answer is incorrect.

Comments: In Stack Overflow, comments are considered “temporary ‘Post-It’ notes left on a question or an answer”.¹² Comments are located below each question or answer and can be used to clarify information or follow up with further details. On the R-help mailing list, we define comments as messages written to *improve an answer or as a follow-up to a discussion*. It should be noted that for an email to qualify as a comment, it must not

¹²<http://stackoverflow.com/help/privileges/comment>

be written by the person who asked or answered the original question. Otherwise, the message would be considered an update. Because both Stack Overflow and the R-help mailing list permit participants to ask multiple questions in the same thread, the subcategories of comments are not mutually exclusive.

Regarding the frequency of comments, the main difference between the two channels is that Stack Overflow comments are less likely to be considered corrections or alternatives (Correction/Alternative subcategory) than on the R-help mailing list. The Stack Overflow R tag sample also contained 2.1 times more comments than the R-help sample (see Table 2).

4.2 How Knowledge is Constructed on Stack Overflow and the R-help Mailing List

Our analysis helped us identify two different approaches for constructing knowledge (RQ2) on Stack Overflow and the R-help mailing list: participatory knowledge construction and crowd knowledge construction.

Participatory knowledge construction is an approach where answers are created through the cooperation of multiple users in the same thread. Participants complement each other's solutions by discussing the pros and cons of each answer and by adding different viewpoints, additional information, and examples. This process is similar to a team working together towards a common objective.

Crowd knowledge construction leverages the experiences of many users who work in a relatively independent manner. Each user contributes to the thread, adding variety to the pool of solutions. However, the user's priority is to provide a correct answer and not to discuss other solutions. This is comparable with the concept of a group in which people work towards the same objective but not necessarily together (e.g., Amazon's Mechanical Turk). Participants can vote on others' ideas, but the main idea is not constructed through discussion.

On the R-help mailing list, *participatory knowledge* construction occurs when: (1) previous answers are included in the current answer, with clear links between them; or (2) a reply contains a direct reference to other answers or authors. Figure 3 depicts two examples of the way participatory knowledge occurs on the R-help mailing list: direct citation of the author of a previous answer, and inferable links between answers.

On Stack Overflow, *participatory knowledge* construction takes place when: (1) one can infer a link between answers, through either a direct or indirect reference; or (2) comments complement the answer or directly cite another author. Participatory knowledge construction also occurs in different places on Stack Overflow, perhaps as a consequence of its rich interface. We observe this type of knowledge construction when a user answers a question and directly cites or links to someone else's answer in the thread, or when a user cites someone else's question or answer in a comment (a typical case is linking to a previously asked question). Figure 4 depicts an example of participatory knowledge construction on Stack Overflow: when an answer was deemed insufficient, a user helped out by adding a comment and referencing another author's answer.

On the R-help mailing list, *crowd knowledge* construction takes place when different messages respond directly to the original question, rather than to another response.

On Stack Overflow, *crowd knowledge* construction occurs when: (1) there is no direct or inferable reference between answers; or (2) an answer is a variation of one of the other

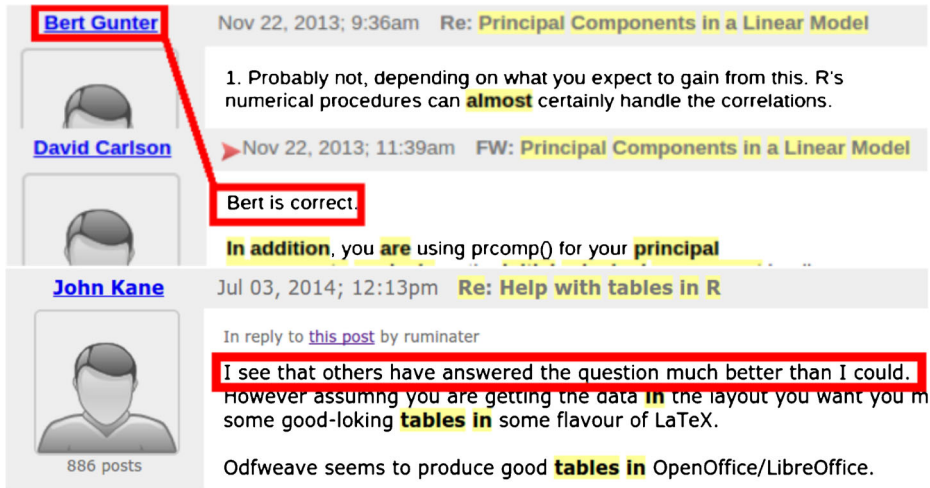


Fig. 3 Participatory knowledge construction on the R-help mailing list

answers in the thread. Figure 5 depicts an example of crowd knowledge construction on Stack Overflow. As can be seen from the figure, two of the three answers provided the same solution.

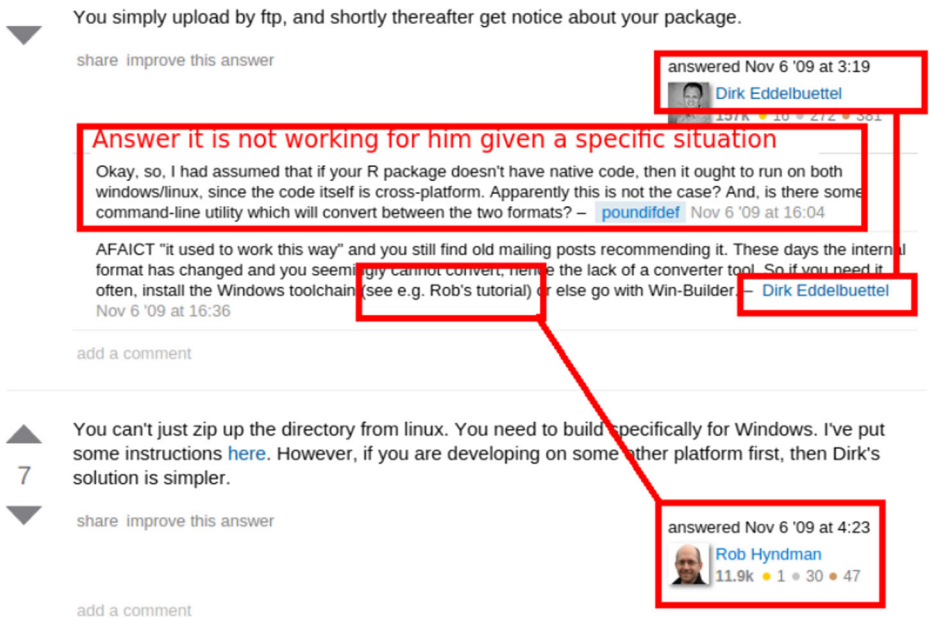


Fig. 4 Example of participatory knowledge on Stack Overflow: users built on the comments and answers of other users



Fig. 5 Example of how crowd knowledge construction occurs on Stack Overflow. The three authors provided similar answers, but did it independently of each other

4.3 Why Users Post to a Particular Channel or to Both Channels

From our survey, we were able to learn why some R community members preferred one channel over the other—we summarize their responses below. Using results from the analysis of the archived data, we discuss why some members post some questions to both channels.

4.3.1 Why Participants Post on Stack Overflow

Survey participants preferred using Stack Overflow for several reasons: (a) the ability to gain peer recognition (the advantage of gaining points—and visibility—is a major draw of Stack Overflow); (b) its rich and user-friendly interface; (c) answers are straight to the point; (d) questions are usually answered faster on Stack Overflow than on the R-help mailing list; and (e) it is easy to search for previous questions and answers.

However, the respondents reported a few main drawbacks of using Stack Overflow: (a) there is an overabundance of related questions; (b) one requires a certain level of experience to understand some of the answers; and (c) Stack Overflow's strict rules only allow questions and their answers, they do not allow discussions nor questions about opinions.

4.3.2 Why Participants Post on the R-help Mailing List

Survey participants reported a few benefits of using the R-help mailing list: (a) the email format is convenient; (b) following the mailing list provides awareness and increases learning in new topics; (c) there is more flexibility regarding the topics that one can discuss; and (d) there is a high level of participation from experienced users. The respondents did note a couple of disadvantages of R-help: (a) some discussions lead to aggressive behavior; and (b) searching the archives is not easy.

4.3.3 Why Participants Post to Both Channels

Our analysis of the archived data revealed that some users (79 in our sample) posted the same question on both channels. Based on the responses from the survey, we identified that being active on both channels brings benefits to those asking and answering questions (RQ3):

Find a better answer: One channel might result in a better answer than the other.

Support follow-up questions: We found that the R-help mailing list is often used to conduct follow-up discussions on specific answers provided to Stack Overflow questions. Stack Overflow's focus is on finding an answer to a question and provide a rudimentary method to discuss the specifics of an answer, either by adding comments to the answer—which cannot be threaded—or by asking another question (related to the answer). In contrast, a discussion on R-help can continue long after an answer has been found through follow-up questions involving a variety of people, not just the person who asked the original question.

Speed up answers: Users ask the same question on both channels in order to get an answer faster. However, in many cases this behavior is not encouraged by the community as it is deemed impolite.¹³ This seems to be a matter of opinion, as other community members argue in favor of cross-posting questions.¹⁴

4.4 How Participation Differs Between the Two Channels Over Time

Vasilescu et al. (2014) studied participation in the R-help and Stack Overflow communities over time. Their research showed strong evidence that knowledge seeking activities are moving from R-help to Stack Overflow (as of the end of 2013). They also noticed a trend indicating that Stack Overflow continues to grow and R-help continues to decrease. These findings prompted us to also consider how participation differs between the two channels over time (RQ4). However, we take a different approach by slicing the data based on question score and participation activities over time. In the following, all findings come from our extended dataset that covers September 2008 to September 2016 for both channels (unless specified otherwise).

We first explore the evolution of the number of questions in both channels as the main proxy of activity (the results are shown in Fig. 6). One aspect that is changing in Stack Overflow is the growth of questions that receive an overall negative score. A score in Stack Overflow is the sum of positive votes minus negative votes, which can be considered an indication of the question's quality. As can be seen in Fig. 6, the number of questions with an overall positive score has flattened and is starting to decrease. However, if we inspect the trends over the last eight years, this might be misleading. By looking at the most recent period (Jan. 2015 - Sept. 2016), Fig. 7 shows that both channels are relatively flat in terms of the overall number of questions, but the number of questions in Stack Overflow is between 10 and 20 times the number of questions found in R-help. Additionally, the proportion of questions with a positive score in Stack Overflow has been decreasing steadily (shown in Fig. 8).

¹³https://stackoverflow.com/questions/3892033/r-why-this-doesnt-work-matrix-rounding-error#comment4151921_3892033

¹⁴<http://meta.stackoverflow.com/questions/266053/is-it-ok-to-cross-post-a-question-between-non-stack-exchange-and-stack-exchange>

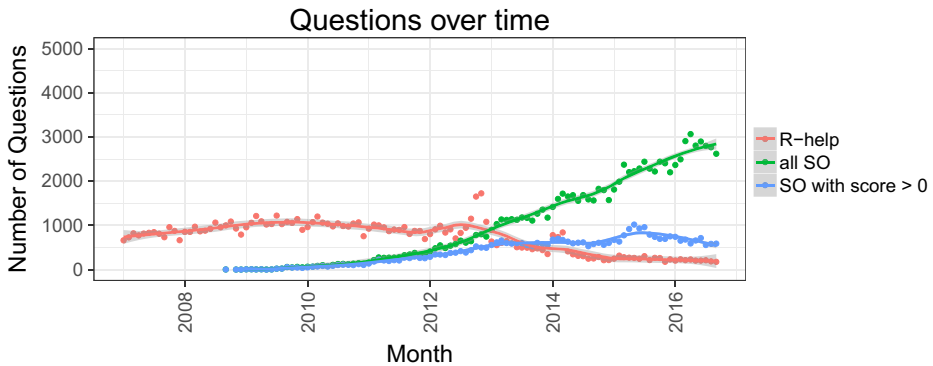


Fig. 6 The number of questions asked over time: Stack Overflow activity has been much greater than R-help activity, however, the number of questions with a positive score has flattened

As the Stack Overflow community grows and the number of questions increases, the community faces several challenges that may explain the decrease in questions with positive score. One challenge is handling duplicate questions¹⁵ (cut-and-paste duplicates, accidental duplicates, and borderline duplicates). Another challenge is dealing with low quality questions: these questions are often poorly described, not directly related to R, or posted by users who didn't put much effort into searching for the answer themselves (referred to as "help vampires" by Srba and Bielikova 2016). At the same time, many questions are left unanswered and with no indication from the community (i.e., zero score). Srba and Bielikova (2016) found that there was a constant increase of poor quality questions, unanswered questions, and deleted questions on Stack Overflow.

We explored three potential reasons for the decrease in questions with a positive score:

1. We looked at the number of questions marked as duplicates. We found this proportion is increasing, but the overall number of questions marked as duplicates is only 3% of all questions.
2. Then we counted the number of questions with a negative score, but this only accounts for 2.9% of all questions.
3. Finally, we measured the number of posts with a zero score. We found that 29.2% of all posts have a score equal to zero. A small proportion of these questions (3%) had a zero score after being voted up and down.

We posit that the increase in the number of questions with zero score (and the corresponding decrease in questions with a positive score) is because newer questions tend to be too focused or obscure to be of interest to others, but further research is required to verify this assertion.

4.4.1 How Long Users Participate in the Channels

An important measure of a community and its ability to curate and maintain knowledge is whether their users continuously participate over time, even if their participation is small. To help us measure the continuity of participation, we divided the history of both channels

¹⁵<https://meta.stackoverflow.com/questions/315293/answering-borderline-duplicate-questions>

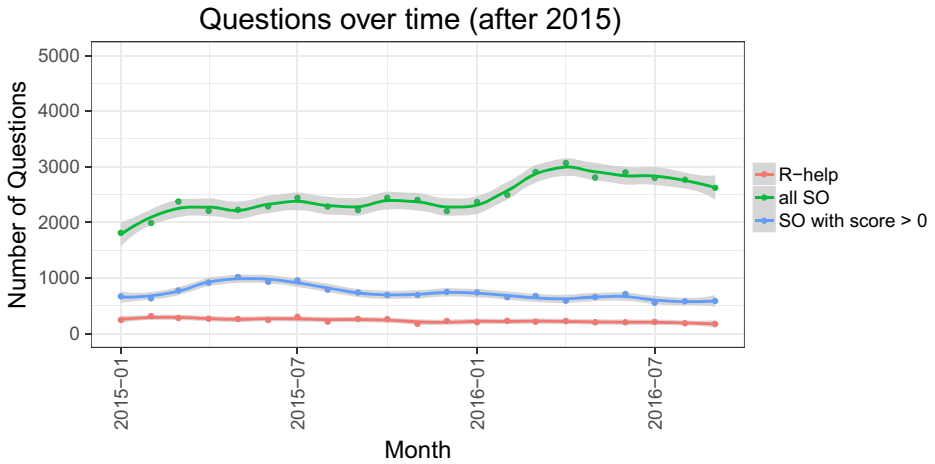


Fig. 7 The number of questions asked after January 2015: both channels have flattened, but the number of Stack Overflow questions with a positive score continues to decrease

into month-long segments. In R-help, we considered that a user participated in a given month if they posted at least one email to the list during that period. For Stack Overflow, we considered that a user participated in a given month if they posted, responded to, or commented on at least one question.

The results show that users do not participate in either channel for a long period of time—their participation tends to be brief. Figure 9 shows the accumulated proportion of users that participated during a given number of months (not necessarily consecutively). The curves are very similar and skewed: 62% of R-help users and 65% of Stack Overflow users are active for a single month only; 90% of R-help users are active for 5 months or less; 90% of Stack Overflow users participate 4 months or less.

We noticed that a large proportion of users ask questions but never contribute answers. The proportion of those who never post an answer in R-help is 57.3%, while it is 63.1% in

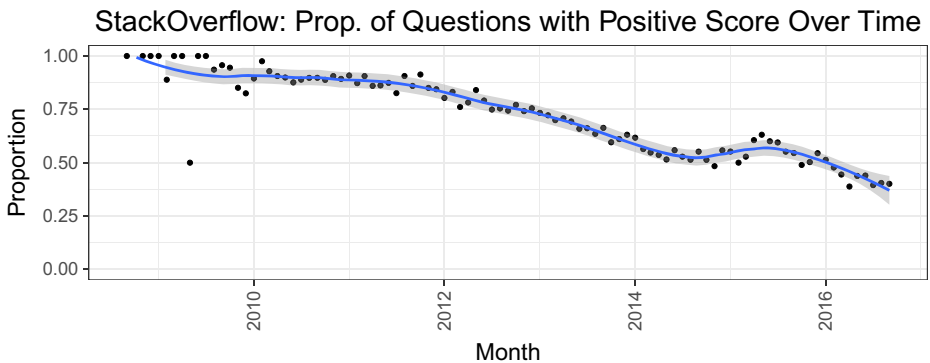


Fig. 8 The proportion of Stack Overflow questions with a positive score has been decreasing steadily

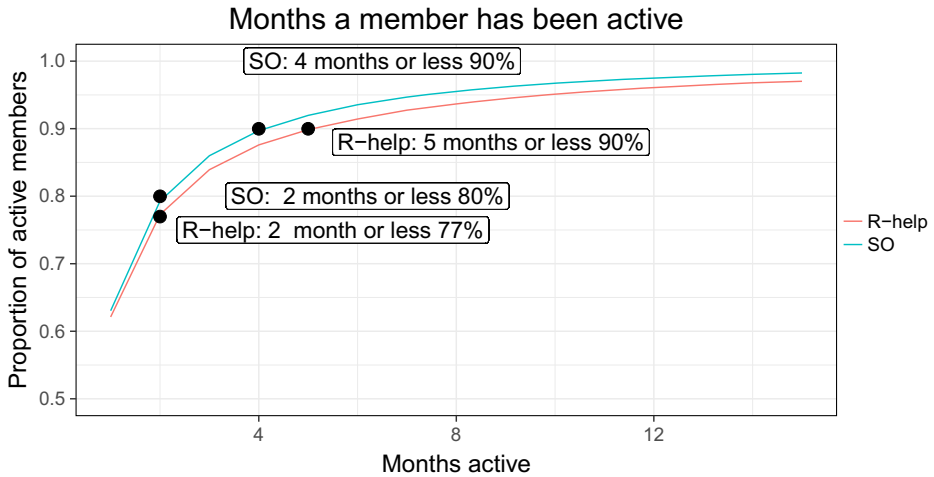


Fig. 9 The number of months a user has been active. This plot shows the accumulated proportion of users who have been active for a given number of months: 62% of R-help users and 65% of Stack Overflow users are active for 1 month only; 90% of R-help users are active 5 or months or less; 90% of Stack Overflow users participate 4 or months or less. Months do not have to be consecutive

Stack Overflow.¹⁶ Figures 10 and 11 show the cumulative proportion of users that participate in either channel for a given number of months. As can be seen, people who answer questions tend to stay around longer in R-help than those contributing questions only. However, for Stack Overflow, the difference is very small: both people posting questions and those that answer them do not stay around very long.

4.5 Are There Significant Differences in Participation Activity Between Community Members?

To consider participation activity within the two channels, we first consider new users by counting how many newcomers there are to both channels and checking if they are likely to post again. Secondly, we look at contributors that are responsible for the vast majority of posts on both channels.

4.5.1 Activity of New Users

New users are important for the survival and continuity of a community. Unfortunately, since we can only track users when they actively participate in a channel, we do not know who is a passive participant. Hence, we use the date of a first contribution as the proxy of when a user joins the community. Figure 12 shows the proportion of users who have participated for the first time in a particular month. Both sub-communities show a similar pattern: in R-help, between 25% and 35% of all users are first-time participants; in Stack Overflow, between 40% and 50% are first-time participants. In recent years, both channels have seen a slight decline and many of these first-time participants only contribute once

¹⁶There is a threat to validity for this result in the R-help data: Stack Overflow separates responses into comments and answers, however, R-help does not have this distinction. For R-help, we consider that any direct reply to an email is an answer; and we consider a reply to an answer to be a comment.

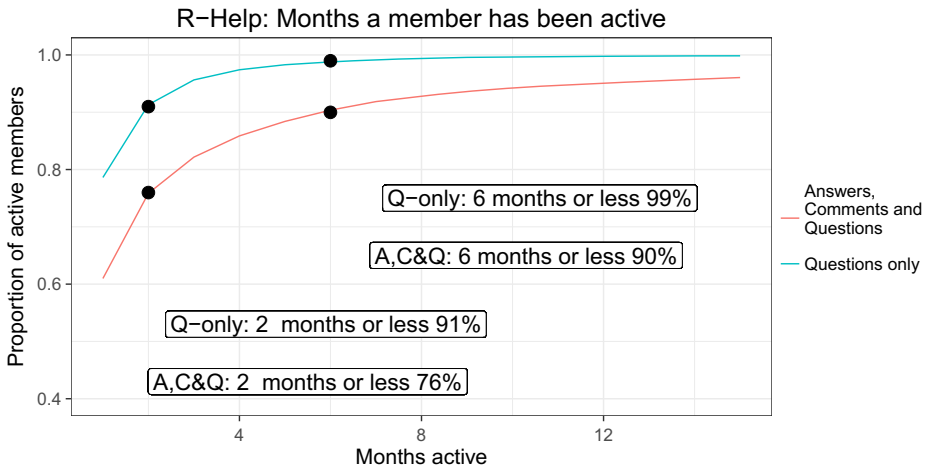


Fig. 10 The months an R-help user has been active according to whether they only ask questions or only answer questions (and potentially ask questions, too). People who answer questions tend to stay around much longer than those who only ask questions

(see Fig. 13): except for the first 1.5 years of Stack Overflow, both channels see that the proportion of one-time participants is between 10% and 20%. In fact, the number of users who participate only once (at any time) is relatively large over the lifetime of the channels: 43% in R-help and 30% in Stack Overflow.

4.5.2 Common Users of Both Channels

We found 1,449 unique users that participated in both channels. For this, we compared and matched the identities of contributors between both channels between September 2008

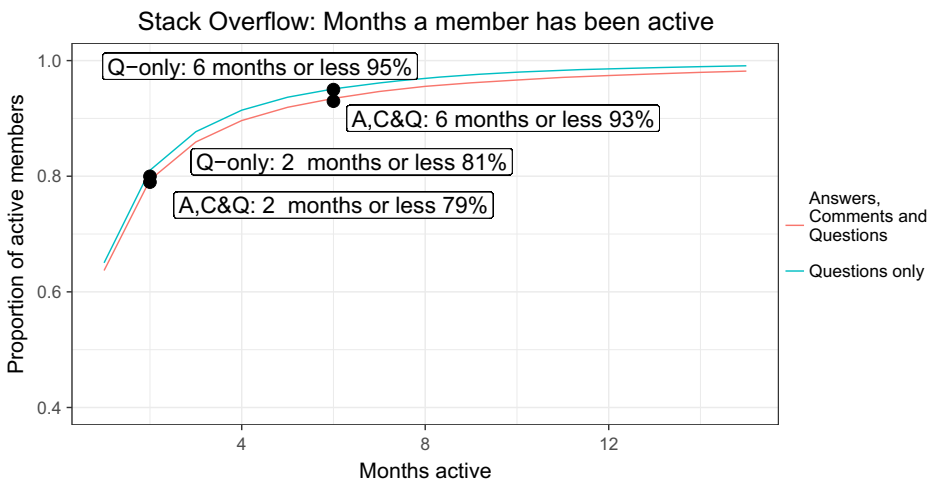


Fig. 11 The months a Stack Overflow user has been active according to whether they only ask questions or only answer questions (and potentially ask questions, too). Both types of users do not stay around very long

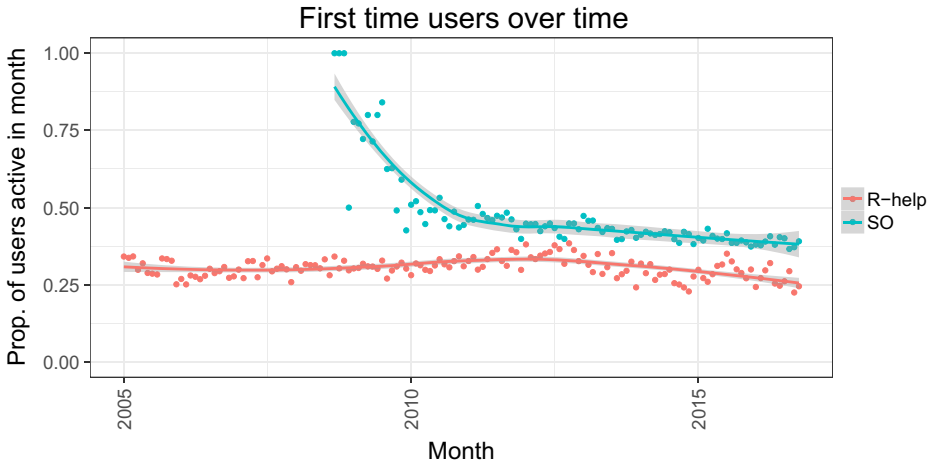


Fig. 12 Proportion of new users over time. The top lines (*thicker*) correspond to the number of users who post their first question in that month. The bottom (*thinner*) lines correspond to the subset of new users who only ask one question and then never participate again

and September 2014 (September 2014 is the last Stack Overflow data dump with the email hashes). The number of members who participated in both sub-communities is relatively small (2.5% of all users), yet a handful of these contributors (also referred to as “caretakers” by Srba and Bielikova 2016) are responsible for a very large proportion of the answers in both channels. Figure 14 shows the accumulated proportion of answers that have been contributed by these authors. The top contributor has contributed 3.9 and 3.7% of answers in R-help and Stack Overflow, respectively. In fact, the top 6 contributors to both channels are responsible for 10% of the answers (and only one of those belongs to the *R-core* group). Here answers on Stack Overflow include both posts and comments to questions.

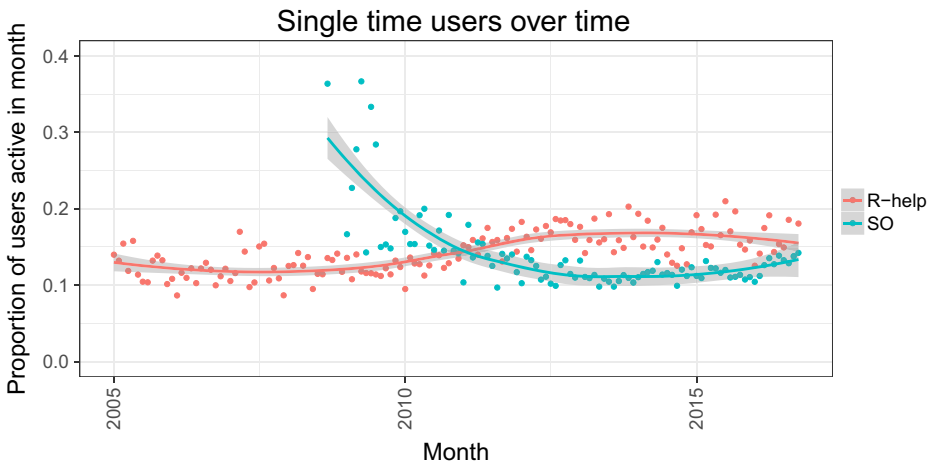


Fig. 13 Proportion of one-time users in any given month

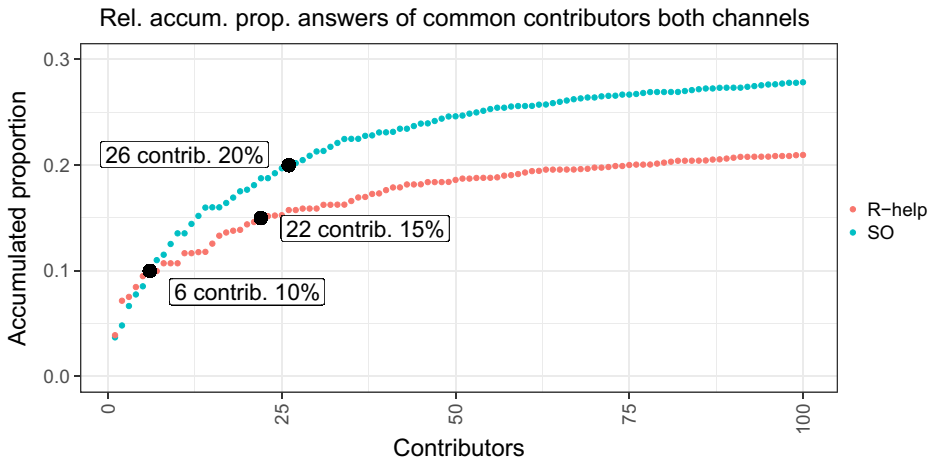


Fig. 14 Accumulative proportion of answers by the most prolific users who post to both channels. As we can see, the top 6 most prolific users contribute approximately 10% of the answers. This plot only reflects posts between Sept. 2008 and Sept. 2014, the period when we could compare and match identities between both channels

4.5.3 Prolific Contributors

Both channels benefit from a handful of prolific members who answer most questions. Thus we refer to these users as *prolific contributors*. Figure 15 shows the most prolific contributors in each channel, ordered by the number of answers they have contributed. The vertical axis shows the accumulated proportion of contributions to all answers in each channel. In R-help, 8 users have contributed 25% of all answers and 40 have contributed 50% of all answers. In Stack Overflow, the distribution is not as steep, yet 27 users have contributed 25% of all answers and 146 have contributed 50% of all answers (including comments).

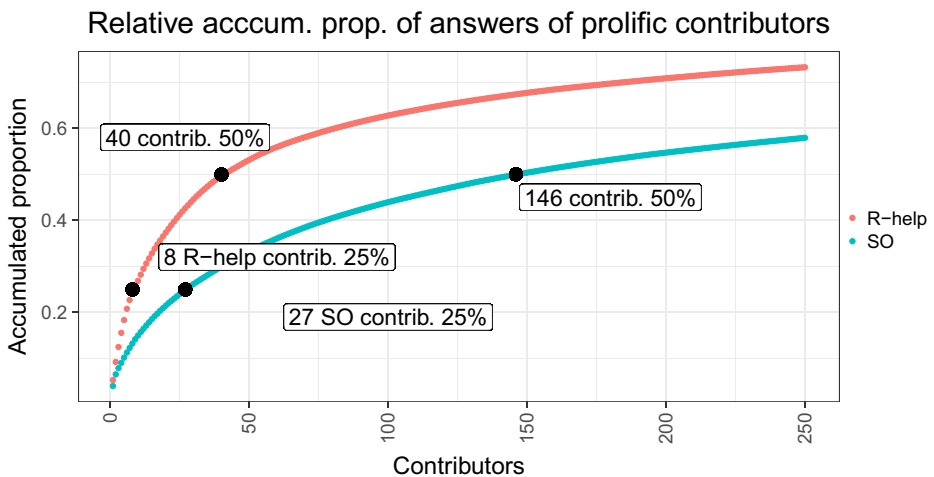


Fig. 15 Accumulative proportion of answers by the most prolific users of each channel. The top 8 and top 27 contributors to R-help and Stack Overflow, respectively, are responsible for 25% of all the answers

This means that 0.13% of the R-help mailing list users contribute 50% of all the answers, while 0.23% of the Stack Overflow users contribute 50% of all the answers. We identified the professions of the top 10 contributors to both channels: five professors, three package maintainers, a core developer, and one retired person. For contributors who participated in only one of the channels:

- We identified the professions of 9 of the top 10 contributors in Stack Overflow: one professor, two package maintainers, three data scientists, one consultant, one retired person, and one anonymous contributor (there is no personal information in their profile).
- In the case of R-help, there are four package maintainers, one book author/professor, two professors, one core developer/professor, one consultant, and one retired person.

5 Discussion

In this section, we reflect on the results presented in the previous section, distill some themes from this work and relate them to relevant related research, and point out possibilities for future work. As we present the themes, we provide representative quotes extracted from the survey we conducted in the second phase of our study, using P# to indicate the participant ID.

5.1 Knowledge Creation and Curation

Our results show that both channels we studied provide similar knowledge support for asking questions and providing answers. However, there are some important differences between the channels, which we discuss in detail below and summarize in Table 3.

5.1.1 Knowledge Construction

Stack Overflow’s gamification mechanism encourages users to be the first to answer questions (Singer et al. 2013). In contrast, the R-help mailing list is a less competitive environment where users tend to build on other responses: users work as a team rather than acting as individuals searching for points. As a result, knowledge on Stack Overflow is built in a more crowdsourced manner, while knowledge on the R-help mailing list is usually built in a participatory manner.

Since the competitive Stack Overflow environment creates an incentive to be the first to answer rather than improve and build on other answers, it is common to find questions with several answers that provide the same information. For example, three of the six answers in the Stack Overflow question titled “*Resources for learning SAS if you are already familiar*

Table 3 Comparison of the ways knowledge is shared on Stack Overflow and the R-help mailing list

	Stack overflow	R-help
Knowledge construction	Mainly crowd	Mainly participatory
Topic restriction	Yes	No
Emphasis	Curating knowledge	Developing knowledge

with R ¹⁷ reference the same books. And while Stack Overflow provides a powerful curation mechanism to ensure the best answers make it to the top, it does not explain why an answer is better than another.

In contrast, the R-help mailing list fosters a participatory environment where users discuss proposed answers—users tend to provide more background to answers and explain the rationale behind them. For example, the question “*Arrange elements on a matrix according to rowSums + short ‘apply’ Q*” that was posted to both Stack Overflow¹⁸ and R-help¹⁹ and illustrates how the two communities build knowledge. On Stack Overflow, each participant contributed a solution without any evidence of having collaborated with others. In fact, the person who asked the question actually stated that both answers worked. However, they selected one as the chosen answer and commented on the other with “*many thanks for your help, all! these did the trick*”, and there was no reflection on why they did so. On the R-help mailing list, users complemented each other’s answers by providing further information and insights to existing answers.

Vasilescu et al. (2014) showed that users who are active on both channels tend to provide answers faster on Stack Overflow than on R-help, suggesting that they are motivated by the gamification aspects of Stack Overflow, and thus, tend to gravitate towards crowd knowledge construction. While the crowd-based approach is prevalent, the construction of knowledge on Stack Overflow is not limited to the crowd-based approach. Participatory knowledge construction can also be observed, such as the up/down voting of questions and through the provision of comments. However, in most cases, participatory knowledge construction on Stack Overflow is used for editing answers (e.g., correcting grammar) or linking to previously asked questions. Similarly, some knowledge on the R-help mailing list is constructed in a crowd-based manner, but this is much less prevalent than participatory construction.

Tausczik et al. (2014) examined how users of Math Overflow, a Q&A platform for mathematicians, collaborate and construct knowledge. They found that collaboration was diverse and fell on the spectrum between *independent* (crowd-based) and *interdependent* (participatory). Similar to our findings with Stack Overflow, the most common collaborative act was of an independent nature (i.e., providing information), while other contributions that built on existing work were less common (e.g., clarifying questions, critiquing answers, revising answers, and extending answers).

Our results seem to imply that Stack Overflow’s gamification features, which have been effective at creating a large corpus of questions and answers, have the side effect of reduced collaborative knowledge creation between users. In their study on building Stack Overflow reputation, Bosu et al. (2013) proposed six strategies for increasing reputation score, including *be the first to answer* and *answer at off-peak hours*, which indicates crowd knowledge creation. Furthermore, while Stack Overflow gives people the ability to vote on comments, it does not reward points to users that post comments. For example, some users search Stack Overflow for answers within comments and convert them to proper answers to gain reputation points.²⁰

¹⁷<http://stackoverflow.com/questions/501917/resources-for-learning-sas-if-you-already-familiar-with-r>

¹⁸<http://stackoverflow.com/questions/4333171/arrange-elements-on-a-matrix-according-to-rowsums-short-apply-q>

¹⁹<http://r.789695.n4.nabble.com/Arrange-elements-on-a-matrix-according-to-rowSums-short-apply-Q-td3068744.html>

²⁰<http://duncanlock.net/blog/2013/06/14/the-smart-guide-to-stack-overflow-zero-to-hero>

An important research question that arises from these findings is whether Stack Overflow's model can be improved to provide better participatory knowledge construction support without hindering its ability to curate information for future use.

5.1.2 Topic Restriction

Stack Overflow's participation rules only permit questions related to programming that have a clear answer, restricting the topics that can be discussed. In contrast, the R-help mailing list is suitable for discussing any topic related to the R language. For example, questions related to R but not focused on software development are not rejected by the R-help mailing list community—topics that trigger discussion are welcomed.

Stack Overflow questions that trigger a discussion are flagged as opinion-based or off-topic and typically closed. Correa and Sureka (2014) found that 18% of deleted questions on Stack Overflow are subjective (i.e., ask for opinion). For example, a question titled “*What's a good example of really clean and clear [R] code, for pedagogical purposes?*”²¹ was flagged as *off-topic* because the question was not related to software development. An R-help user wrote a fine explanation of the purpose of each channel in a message on the mailing list:²²

“Got an R programming question that you think has a definite answer? Post to [Stack Overflow]. Want to ask something for discussion, like what options there are for doing XYZ in R, or why lm() is faster than glm(), or why are these two numbers not equal? Post to R-help. Questions like that do get posted to [Stack Overflow], but we [vote] them down for being off topic and they disappear pretty quickly”.

Squire reported that, despite the gains in participation and the response time provided by Stack Overflow, many development communities keep using mailing lists, either as a primary communication channel or as part of a hybrid solution where multiple channels are used, thus allowing for non-restrictive topics and fostering discussion (Squire 2015). Mailing lists are also favored for their simplicity and guaranteed delivery (i.e., knowing who will receive the email) (Zhang et al. 2015).

5.1.3 Curated Knowledge and Knowledge Development

One of the main benefits of Stack Overflow's crowd-based knowledge construction is the creation and curation of a pool of questions and answers. In contrast, R-help provides an environment in which users develop knowledge through participation, but this knowledge is not curated for future use. This makes it difficult for those that didn't participate in the creation of the knowledge (either actively or passively) to reuse the information.

While Stack Overflow has been successful, some users feel that by not fostering discussion, it restricts thinking that might lead to better answers, as P26 explained:

“Many developers share my view that [Stack Overflow] is a very bad model, ... [it] removes the value added by reading list traffic that doesn't seem directly relevant to

²¹<http://stackoverflow.com/questions/1739442/whats-a-good-example-of-really-clean-and-clear-r-code-for-pedagogical-purpos>

²²<http://r.789695.n4.nabble.com/creating-an-equivalent-of-r-help-on-r-stackexchange-com-was-Re-Should-there-be-an-R-beginners-list-td4684587.html#a4684954>

a currently conceptualized question, but which may lead to a new conceptualization (out-of-the-frame thinking). [Stack Overflow] cannot do that”.

Similarly, P35 stated that they use the R-help mailing list if the questions are not 100% “*help-me-to-code-this*”.

However, Stack Overflow excels at creating and organizing a corpus of questions and answers. Its curation mechanisms provide tools for keeping the channel clean of what seems to be unnecessary information (e.g., flagging questions, deleting comments, editing messages, and demoting irrelevant answers), as P14 explained:

“[Stack Overflow] is an excellent model for providing a rich resource for users of R, which the R-help mailing list was not. Ability to include light markup, render code blocks nicely, no nested email threads all helps the experience of searching for and finding the help that a user needs, and I want to contribute to that”.

Another interesting aspect emerging from our findings is that the activity on the R-help mailing list is only marginally smaller than on Stack Overflow (the proportion of responses in each category fluctuated between 1.4 and 2 times). Further research is required to assess and verify the quality and effectiveness of answers.

5.1.4 Maturing as a Community: from Knowledge Creation to Knowledge Curation

One of the major discoveries from Phase III of our study is the reduction in growth of active participation on Stack Overflow (asking or answering questions). In particular, the number of new questions with an overall positive score has started to decrease over time, but this is to be expected: it’s likely that most sought after questions have already been asked. Given the large number of questions asked through Stack Overflow, many questions today are expected to be either duplicates of previous questions, be of low intrinsic value (too specific to be useful to others), or will end up flagged as not being a question. This decline of Stack Overflow participation has been discussed by community members²³ and the Stack Overflow development team,²⁴ and both agree that Stack Overflow “*is not declining, it is serving its purpose quite well*”.

When we consider that the popularity of R continues to increase, this implies that when new members to the R community seek answers, they might not need to post questions—a testament to the value of the knowledge currently gathered by Stack Overflow. This also seems to imply that the activities of Stack Overflow contributors are shifting to tagging, flagging (e.g., as duplicates, unclear, or off-topic), or editing existing questions as opposed to answering new questions.

The frequency of postings on the R-help mailing list has now stabilized at approximately 200 questions per month. It is possible that by curating answers to the most frequently asked questions, Stack Overflow has contributed to a reduction of questions posted on the R-help mailing list, and the R-help community may now be able to concentrate on higher level questions than those asked in the past. It is also possible that R-help is now discussing questions that would be flagged as invalid on Stack Overflow. Further research should study the types of discussions that are now occurring on R-help and compare them to those from before the rise of Stack Overflow.

²³<https://www.javacodegeeks.com/2016/09/stopped-contributing-stackoverflow-not-declining.html>

²⁴<https://stackoverflow.blog/2016/10/podcast-89-the-decline-of-stack-overflow-has-been-greatly-exaggerated/>

5.1.5 Active vs. Passive Participation

We found that more than 30% of users in both channels contributed only once, usually to ask a question. It is likely that these people continue to passively participate (e.g., consuming content), however, it is not clear why this is the case. It is possible that participants do not feel welcome, or that they feel that actively participating in each channel is not worth their time and effort. Another potential explanation is that these users contribute only when they have a very hard question for which they cannot find an answer anywhere else. Future work is needed in this area.

Similarly, approximately two thirds of users in both channels participate for a single month and then never actively participate again. This set of users includes the previous set (those that contribute only once) and might have related reasons for the halt in participation. It is also possible that these users are evaluating the channel, and after a short period of time, decide that they do not want to contribute further.

Nonetheless, new users continue to join both channels. In recent months, between 25% and 35% of users in any given month are new. This continuous growth seems to guarantee a steady flow of questions and answers in both channels, however, the experience and knowledge of the departed contributors is no longer present. It is possible that they become passive participants who are ready to continue contributing in the future. More research is needed to fully understand these behaviours.

5.1.6 Prolific Contributors

There is a handful of users that are responsible for a large proportion of answers (less than 0.25% of users in both channels contribute 50% of the answers). It is not clear if the success of both channels can be attributed to them or not: e.g., it is possible that without them, other users might have answered those questions. Further research should look into the motivations of these individuals, consider the time they commit to help others, and also compare these patterns of activity with similar channels to see if this is a common phenomenon.

Several of these prolific contributors actively participate in both channels. The top contributor is responsible for almost 4% of the answers in either channel, and combined, the top 6 have authored approximately 10% of the answers. These prolific contributors are likely serving as a bridge of knowledge, moving information from one channel to the other.

When we looked at the professions of these contributors, we found that many of them are professors, book authors, or both. This implies that their goals include to learn and impart knowledge; for these reasons they probably see their participation as a major source of knowledge and an opportunity to diffuse it. Another group of prolific contributors are package maintainers who are likely experts in R and their corresponding packages, and perhaps see these channels as mechanisms to provide support to their users. These assertions should be confirmed in future work.

5.2 Threats to Validity

Here we examine and discuss threats to the validity of our approach (Runeson et al. 2012).

Construct validity: To reach the emerging themes, we relied on subjective human judgment during the data coding phase. Researchers had to decide if a message fell within a specific coding category. To alleviate this issue, two researchers coded the qualitative

data as part of the analysis process. We applied the Cohen Kappa coefficient on categories that were not mutually exclusive, but whose purpose was to trigger discussion between coders. We set a threshold of 0.8 as the minimum to obtain agreeable results, which is higher than the 0.6 suggested in the literature (Landis and Koch 1977). Regarding the quantitative analysis, there are several threats to validity associated with the mining of the R-help mailing list. First, our method to identify unique individuals might be inaccurate, however, we achieve results similar to those in Vasilescu et al. (2014). Second, in our statistical analysis we considered the first email to the list as a question. And direct reply is considered an answer and further replies are considered comments; to do this we relied on the presence of *In-Reply-To* and *References*, which not all messages might include. On the other hand, Stack Overflow clearly identifies individuals and classifies posts into questions, answers and comments. It is also likely that a small proportion of emails are announcements or non-question emails. Regarding the mapping of individuals between the two datasets, as Stack Overflow stopped publishing the md5 of users, it is likely that the number of common contributors between the two channels is underestimated.

Internal validity: Stack Overflow's data is structured while the R-help mailing list consists of unstructured data. As a result, some of the mapping between the two channels was straightforward (e.g., a follow-up to a reply is a comment to that question), while in other cases it wasn't as obvious (e.g., identifying some emails as questions). To reduce the risk of bias when mapping the messages between channels (in the qualitative analysis), two researchers performed the mapping.

External validity: Our case study was exploratory in nature and we purposefully aimed to study the R community. Many R users are likely to be *casual developers* with limited or non-existent programming experience, with backgrounds that vary from biology to statistics, and thus, our findings may not apply to other developer communities. However, since Stack Overflow and mailing lists are widely used by other communities, we believe that our findings may be extended to these groups as well (Squire 2015). We do not claim the generalizability of our findings to other communication channels (e.g., Slack, GitHub), and further research is required to examine how knowledge is shared on other channels used by developers.

6 Conclusions

The purpose of this study was to understand how the R community collaborates when using different communication channels in the creation and curation of knowledge. In particular, we concentrated on studying how this community has used Stack Overflow (using the R tag) and the R-help mailing list to both ask and answer questions. Our analysis of a random sample of 400 threads from each channel shows that both channels are active communication channels where participants are willing to help others.

We found that knowledge contributed in response to a question can be classified into four main categories: answers, updates, flags, and comments. The number of responses in each of these categories was between 1.4 and 2.5 times greater on Stack Overflow than on the R-help mailing list. While all four types of contributions exist in both channels, they exhibit differences. For example, on Stack Overflow, answers are more focused towards step-by-step tutorials, while R-help answers are more likely to be suggestions or alternatives. Similarly, on Stack Overflow, updates are focused on language (grammar and spelling), while on R-help, updates are expansions of previous responses.

The analysis of these questions and answers shows that knowledge is constructed in each channel in a different manner. On Stack Overflow, there is a tendency to use a crowd approach: participants contribute knowledge independently of each other rather than improve other answers. This is likely a result of the gamification used on Stack Overflow where the person who provides the best answer is the one that gains the most points. In contrast, the R-help mailing list uses a participatory approach where participants are more likely to build on other answers, collaborating to find the best solution.

Another important difference between both channels is that Stack Overflow focuses on making knowledge available for future retrieval; knowledge on the R-help mailing list focuses on the discussion of knowledge, but not in its long-term storage or retrieval. Respondents to our survey also commented that while it is easy to find answers on Stack Overflow and make sense of them, on R-help it is not only hard to find the relevant answers to a question, but it is also hard to see how the many responses to a question relate to each other, and ultimately, what the best answer may be.

Another result of our research is that participants appear to prefer Stack Overflow for asking questions that are expected to have a direct answer. They prefer to use the R-help mailing list when the question requests opinions (Stack Overflow forbids them) or when they want to reach core developers of the R project. Some participants ask the same question in both channels in the hopes of gaining the advantages both channels offer. Additionally, R-help has the ability to complement Stack Overflow by providing a medium where the rationale of answers can be discussed.

A major result of our quantitative study is the impact of a small fraction of the members of both channels who answer a large proportion of the questions in both channels. There is also a dedicated core of members who contribute to both channels, connecting the two communities. At the same time, a large proportion of members participate very few times over a short period of time.

It was also interesting to observe that the frequency of new questions in Stack Overflow is starting to slow down, especially those with a positive score. This effect is probably due to the fact that the most frequently asked questions in R have already been asked and answered; the new questions are either inherently more difficult or more specialized and therefore have a smaller pool of members who would benefit from them. However, research is needed to understand this effect.

Overall, this research shows that the R community is committed to using both channels to help others. Each channel has advantages and disadvantages, and the community appears to be using both effectively to create and curate knowledge regarding the R language. Furthermore, our typology of knowledge artifacts summarized in Table 2 can be used by other researchers that wish to study and understand how knowledge is constructed and curated in other channels or across other communities. As new channels (such as Slack) become more widely adopted, studying these newer channels and comparing them to existing channels is an imperative aspect of understanding knowledge formation in software development.

Acknowledgments The authors would like to thank Cassandra Petrachenko for the editing support and valuable comments that contributed to this work. We also thank Lorena Castañeda for her assistance with the data collection and analysis processes. Finally, we thank the R community users that responded to our survey. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

- Bettenburg N, Shihab E, Hassan A (2009) An empirical study on the risks of using off-the-shelf techniques for processing mailing list data. In: ICSM'09 Proceedings of the 25th International Conference on Software Maintenance, pp 539–542
- Bosu A, Corley CS, Heaton D, Chatterji D, Carver JC, Kraft NA (2013) Building reputation in stackoverflow: An empirical investigation. In: Proceedings of the 10th International Conference on Mining Software Repositories, MSR '13, pp 89–92
- Bowen GA (2008) Naturalistic inquiry and the saturation concept: a research note. *Qual Res* 8(1):137–152
- Correa D, Sureka A (2014) Chaff from the wheat: Characterization and modeling of deleted questions on stack overflow. In: Proceedings of the 23rd International Conference on World Wide Web, WWW '14, pp 631–642
- Creswell J (2009) *Research design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications
- German D, Adams B, Hassan A (2013) The evolution of the r software ecosystem. In: 2013 17th European Conference on Software Maintenance and Reengineering (CSMR), pp 243–252
- Gomez C, Cleary B, Singer L (2013) A study of innovation diffusion through link sharing on stack overflow. In: Proceedings of the 10th International Conference on Mining Software Repositories, pp 81–84
- Ihaka R, Gentleman R (1996) A language for data analysis and graphics. *J Comput Graph Stat* 5(3):299–314
- Jenkins H (2009) *Confronting the Challenges of Participatory Culture: Media Education for the 21st Century*. The John D. and Catherine T. MacArthur Foundation Reports on Digital Media and Learning MIT Press
- Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. *Biometrics* 33(1):159–174
- Lave J, Wenger E (2002) Legitimate peripheral participation in communities of practice. *Supporting Lifelong Learn* 1:111–126
- Li H, Xing Z, Peng X, Zhao W (2013) What help do developers seek, when and how? In: 2013 20th Working Conference on Reverse Engineering Reverse Engineering (WCRE). IEEE, pp 142–151
- Mamykina L, Manoim B, Mittal M, Hripcsak G, Hartmann B (2011) Design lessons from the fastest Q&A site in the west. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, pp 2857–2866
- Naur P (1985) Programming as theory building. *Microprocessing Microprogramming* 15(5):253–261
- Runeson P, Host M, Rainer A, Regnell B (2012) *Case Study Research in Software Engineering: Guidelines and Examples*. Wiley
- Singer L, Figueira Filho F, Cleary B, Treude C, Storey M-A, Schneider K (2013) Mutual assessment in the social programmer ecosystem: an empirical investigation of developer profile aggregators. In: Proceedings of the 2013 Conference on Computer Supported Cooperative Work, CSCW '13, pp 103–116
- Squire M (2015) Should we move to Stack Overflow?: measuring the utility of social media for developer support. In: 37th International Conference on Software Engineering, pp 219–228
- Srba I, Bielikova M (2016) Why is stack overflow failing? preserving sustainability in community question answering. *IEEE Softw* 33(4):80–89
- Stemler SE (2004) A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. *Pract Assess Res Eval* 9:4
- Storey M-A, Singer L, Cleary B, Figueira Filho F, Zagalsky A (2014) The (r) evolution of social media in software engineering. In: Proceedings of the on Future of Software Engineering, FOSE 2014, pp 100–116
- Tausczik YR, Kittur A, Kraut RE (2014) Collaborative problem solving: A study of mathoverflow. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing, CSCW'14, pp 355–367
- Vasilescu B (2014) *Social aspects of collaboration in online software communities*. PhD thesis, Eindhoven University of Technology
- Vasilescu B, Serebrenik A, Devanbu PT, Filkov V (2014) How social Q&A sites are changing knowledge sharing in open source software communities. In: Proceedings of the 17th ACM Conf. on Computer Supported Cooperative Work and Social Computing, pp 342–354

- Wenger E, White N, Smith JD (2009) Digital habitats: Stewarding technology for communities. CPsquare
- Zagalsky A, Teshima CG, German DM, Storey M-A, Poo-Caamaño G (2016) How the r community creates and curates knowledge: a comparative study of stack overflow and mailing lists. In: Proceedings of the 13th International Conference on Mining Software Repositories. ACM, pp 441–451
- Zhang AX, Ackerman MS, Karger DR (2015) Mailing lists: Why are they still here, what is wrong with them, and how can we fix them? In: Proceedings of the 33rd SIGCHI Conference on Human Factors in Computing Systems



Alexey Zagalsky is a PhD candidate under the guidance of Margaret-Anne Storey at the University of Victoria. He received his Bachelor's and Master's degrees in Computer Science from Tel Aviv University, Israel. He focuses on software engineering, studying the interplay between developers, tools, their activities, and how all of it affects collaboration and communication. He is also interested in human-computer interaction, human aspects in software engineering, and computer supported collaborative learning. He has published papers in a variety of conferences, including ICSE, CSCW, and MSR, and his current research aims to form a theory of knowledge in software engineering.



Daniel M. German is Professor in the Department of Computer Science at the University of Victoria, where he does research in the areas of mining software repositories, open source software engineering, and intellectual property.



Margaret-Anne Storey is a Professor of Computer Science and the Director of the Software Engineering program at the University of Victoria. She holds a Canada Research Chair in Human and Social Aspects of Software Engineering. Her research goal is to understand how technology can help people explore, understand, and share complex information and knowledge. She evaluates and applies techniques from knowledge engineering, social software, and visual interface design to applications such as collaborative software development, program comprehension, biomedical ontology development, and learning in Web-based environments.



Carlos Gómez Teshima received his bachelor's degree in Business Administration and Computer Systems Engineering from Icesi University, Colombia. Carlos holds a master's degree in Computer Systems Engineering from Universidad del Valle, Colombia, and a second master's degree in Computer Science from the University of Victoria, Canada. Carlos is ITIL certified with more than seven years of software development experience. His research interests focus on understanding the interplay between developers, and how they affect and are affected by the tools they use.



Germán Poo-Caamaño received his PhD in Computer Science from the University of Victoria, Canada. His research interests include release engineering, software licensing, software ecosystems, free and open source software, and mining software repositories.