Knowledge Curation in a Developer Community:
A Study of Stack Overflow and Mailing Lists

by

Carlos Arturo Gómez Teshima
B.Sc., Universidad Icesi, Colombia, 2005
M.Sc., Universidad del Valle, Colombia, 2013

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Knowledge Curation in a Developer Community:
A Study of Stack Overflow and Mailing Lists

by

Carlos Arturo Gómez Teshima
B.Sc., Universidad Icesi, Colombia, 2005
M.Sc., Universidad del Valle, Colombia, 2013

Supervisory Committee

Dr. Margaret-Anne Storey, Supervisor
(Department of Computer Science)

Dr. Daniel M. German, Departmental Member
(Department of Computer Science)

Supervisory Committee

Dr. Margaret-Anne Storey, Supervisor
(Department of Computer Science)

Dr. Daniel M. German, Departmental Member
(Department of Computer Science)

ABSTRACT

Media channels play an important role in the flow, construction, and curation of knowledge in software development. Understanding how developers use media channels is key to improving developer practices and supporting channel evolution. In this thesis, I investigate the way developers use media channels to curate knowledge within the R software development community. By applying a case study methodology consisting of mining archival data and survey methods, I investigate the R community on Stack Overflow and the R-help mailing list, using a qualitative approach. The findings reveal that Stack Overflow and mailing lists foster knowledge co-construction differently—crowd-sourced and participatory respectively. Furthermore, developers use actively both channels to optimize knowledge exchange and curation.

My thesis contributes to the understanding of knowledge curation by developer communities, and describes a model for a systematic comparison of two or more media channels, within a community of practice. This model allows knowledge categorization and can be used in future studies to explore knowledge flow within multiple media channels. Moreover, based on my observations in conjunction with the survey data analysis, I extracted a set of recommendations to assist practitioners in the use of multiple Question and Answer (Q&A) channels.

Contents

List of Tables

List of Figures

Acknowledgements

This thesis would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

- Prof. Margaret-Anne Storey for her guidance, encouragement, and for giving me the opportunity to pursue something I never imagined I would be doing.
- Lorena Castañeda, my wife, without whom this effort would have been worth nothing. Your love, support and constant patience have taught me so much about sacrifice, discipline and compromise — even if there were times when you said "I told you so".
- Jose Manuel, my son, for taking care of himself when I was trying to focus on this thesis, and helping to take care of his brother.
- David Alejandro, my son, who was born before this thesis was completed and who spent many days with my wife to allow me to focus. I am deeply sorry for the time we spent apart.
- Cassandra Petrachenko, Germán Poo-Caamaño, Alexey Zagalsky, and Maryi Arciniegas for the talks, the editing of the this thesis, and recommendations through these years.
- Chisel Lab and everybody in it during my time there, for being a great and supportive environment, making it easy to work and to laugh.

Dedication

This thesis is dedicated to Jose Manuel and David Alejandro.

Chapter 1
Introduction

Today's software development is more about just-in-time learning than reading manuals [16]. With instantaneous access to information on the Web, programmers do not have to be experts in a particular technology to build an application. Programmers can consult a variety of online resources (e.g., Stack Overflow[1], and Nabble[2]) by entering combinations of keywords in a search engine (e.g., Google[3]). If they cannot find the correct answer there are many easy-to-use *media channels* for assistance. For example, after posting a question to a specific media channel, programmers can access high-quality answers from a global user base that can easily address most programming problems within minutes or seconds [28].

Media channels play an important role in today's knowledge economy, as well as the collaboration, coordination, and communication activities that occur between programmers. However, selecting the most appropriate media channel to transmit an idea can be challenging, given the variety of equally suitable tools and sites. To decide, a programmer considers which of the many channel characteristics will benefit most. Consideration might be some of the following: experts on the channel, flexibility on topics allowed, or even if the channel is asynchronous, socially enabled, or has gamification elements [53].

Media channels are more than just delivery systems—they connect users with a *community of practice* or groups of people with a common interest. Each community has its own implicit or explicit norms (e.g., see the R mailing list posting guide[4]). Any violation of the community norms or channel rules may result in unfriendly responses from the community, or being flagged with a bad reputation (e.g., Stack Overflow flags[5]). Depending on

[1] http://stackoverflow.com/
[2] http://www.nabble.com/
[3] http://www.google.com/
[4] https://www.r-project.org/posting-guide.html
[5] http://stackoverflow.com/help/privileges/flag-posts

the media channel, a bad online reputation can affect real life events. For instance, Singer *et al*. found that reputation on Stack Overflow is used by recruiters to assess programmer performance [37].

There are multiple studies that investigate software development media channels, which provide insights on the way channels are used [15, 45, 38], topic trends [5, 22, 55], best practices [3, 48, 1], and *social programmer* behaviours [25]. Understanding channels is key to improve the developer practices, communication, coordination, and knowledge sharing. However, a review of the literature substantiates that only a few studies investigate the interplay between channels. There are studies that provide valuable insights on channel migration processes [53], synergy between channels [52, 7, 22], and channel usage [44, 45], and yet, there is a need to analyse and compare media channels and the way programmers use them [50]. Questions raised that still beg unanswered are: Is one communication channel replacing the other, or are they cooperating?, Why communities have more than one channel to solve the same problem? In which circumstances one communication channel should be used over another?.

This thesis investigated the way *knowledge* (or user generated content) is curated within a particular software development community. For this study I chose the R community, since it provided broader relevance outside the software development community by including users with no or limited programming experience (e.g., biologist or statisticians). My overarching goal was to provide tools for further studies that analyse and compare the knowledge flowing through media channels. Thus, the R developer community has the potential for a broader selection of users' backgrounds and more diverse knowledge types.

By applying a qualitative *exploratory case study* methodology, as proposed by Runeson *et al*. [34], I analysed the R community on Stack Overflow[6] and the R-help mailing list[7], that is, the main programming related Q&A channels that the R community contains. Additionally, I conducted a survey to bring further insights on the findings. With this findings, I constructed a series of categories that supports knowledge classification and knowledge comparison of the main type of messages (i.e., questions, answers, updates, flags and comments), which these two channels provided. Based on the knowledge categories analysis, I compared the way knowledge was shared on Stack Overflow and the R-help mailing list. Finally, I provided a set of recommendations to assist in the usage of multiple Q&A channels, and when linking resources that are external to both channels.

[6] From now on, every time I refer to Stack Overflow, I are referring to the R community on Stack Overflow.
[7] `https://stat.ethz.ch/mailman/listinfo/r-help`

1.1 Research Questions

This thesis is based on an open challenge from Vasilescu's dissertation *Social Aspects of Collaboration in Online Software Communities* [50], that states *"...to better understand the effects associated with a transition from mailing lists to social Q&A and, e.g., whether mailing lists will eventually die off, future research could also consider analysing the content of the discussions from the two venues..."*, thus my desire is to understand the knowledge flow through channels that serve the same purpose.

My overarching goal was to investigate the way programmers share knowledge on Q&A channels, the interplay of Q&A media channels within a community, the construction of knowledge on media channels, a set of recommendation to use multiple media channels, and the way programmers use external resources in their messages. Therefore, the research presented in this thesis is motivated by the following research questions:

RQ-1. What types of knowledge are shared on Stack Overflow and the R-help mailing list within the R community?

RQ-2. How is the knowledge constructed on Stack Overflow and the R-help mailing list?

RQ-3. How does the sharing of links on Stack Overflow and the R-help mailing list support knowledge construction?

RQ-4. Why do certain users post to both Stack Overflow and the R-help mailing list?

1.2 Contributions

The contributions of this thesis are summarized as follows:

Comparison of how knowledge is shared on the two channels. I compared the way knowledge is shared on both channels based on the findings of this thesis and the survey data. My objective was to identify the differences of how knowledge is shared on Stack Overflow and the R-help mailing list.

Categorization of messages on Q&A media channels. I built a categorization of knowledge based on the analysis of data that flows through media channels. My objective was that categories should support further studies when comparing media channels based on the knowledge flowing through them. With these categorizations I gained

insights about knowledge that flows through the channels, and the differences between them.

A set of recommendations for using multiple Q&A media channels. I created and provided a set of recommendations for using multiple media channels based on the observation and analysis of the data. It is meant to improve multiple media channel usage by providing a best practices reference.

A tool to extract information from mailing lists. Mailing list repositories such as R-help contain valuable information about user behaviours, best practices, topics, problems, and discussions. However, such information exists as unstructured data that needs special processing before it can be studied. To that end, I developed GTMail[8], a software tool capable of dealing with multiple standardization issues when presented on mailing list data (e.g., duplicates, text formatting), extracting URLs embedded in email bodies, eliminating unnecessary information, and uploading the data to a database for further analysis. My tool is compatible with MBOX mailing list formats, and therefore, can be used in any other research that involves mailing list repositories.

Figure 1.1 depicts the mapping between my research questions and my contributions. In the figure, GTMail tool is mapped with all research questions because the tool processes the data used in this thesis.

1.3 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 presents the context and background of this work, including media channels and communities of practice. It also introduces the R community along with the two media channels investigated as part of this thesis: the R-Help mailing list and the Stack Overflow R Tag.

- Chapter 3 describes the elements of my methodology (i.e., research questions, case study method, study design, and content analysis definition), as well as the constructivist position taken in this work, and the study design (i.e., case study and survey).

- Chapter 4 presents the results of data analysis aligned to the research questions. It contains a classification of knowledge type, an explanation of how the knowledge is

[8]This tool is available online `https://github.com/cagomezt/GTMail`.

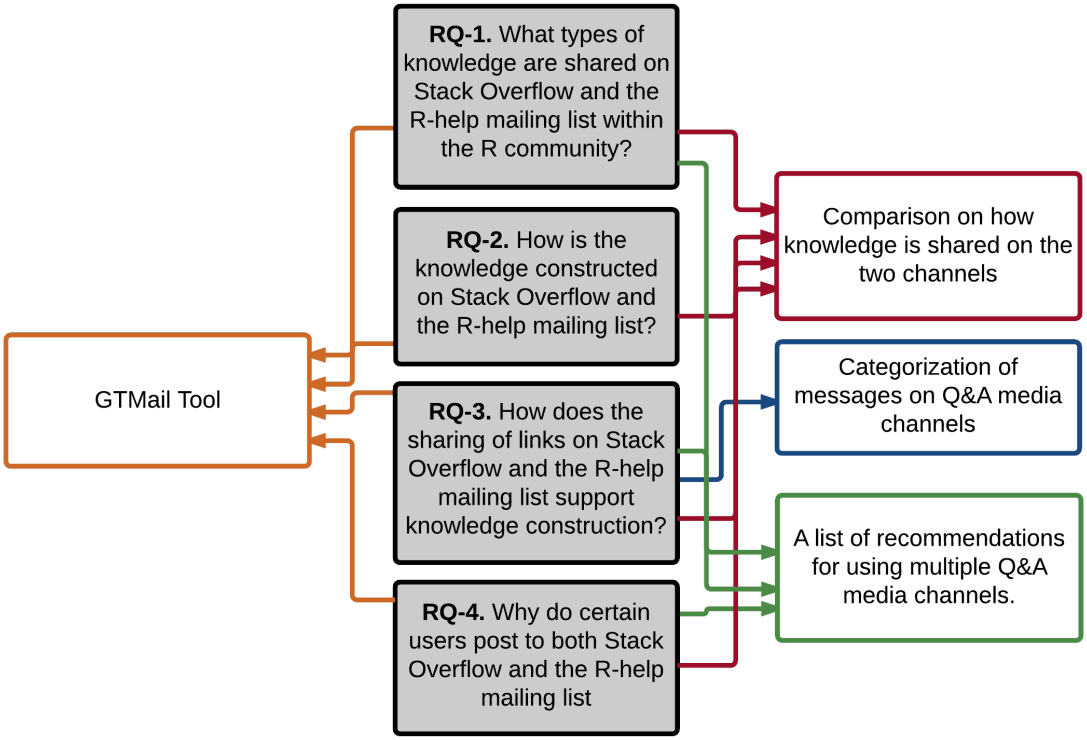


Figure 1.1: Mapping between the research questions in this thesis and the contributions.

constructed, a description of the roles of links on the construction of knowledge, and an analysis of why users post in both media channels. Further, this chapter presents the survey results with pros and cons of both media channels.

- Chapter 5 presents a theory that integrates the findings with the survey results to better understand the interplay of media channels.
- Chapter 6 presents a discussion of my contributory results, consisting of a comparison between Stack Overflow and the R-help mailing list, a Q&A knowledge categorization, a set of recommendations when using multiple channels, and a description GTMail tool that makes it different to other approaches.
- Chapter 7 describes the threats to validity found in this work, including internal and external validities, and standards of rigour.
- Chapter 8 proposes future work based on my findings.
- Chapter 9 presents the conclusions of this thesis.

Chapter 2
Background

Prior to the 21st century, books and classrooms were the main way to learn new programming languages and to answer questions. Software development was an activity performed by small geographically co-located groups using email and phone calls as the main way to coordinate activities, ask questions, collaborate with others, and share knowledge [45].

The emergence of new *media channels* (e.g., wikis, forums, and Q&A Websites) and *communities of practice* caused a stir in the industry. Project-related activities are now scattered among many channels (e.g., bug trackers, source code repositories, and project management tools) [15], and learning new programming languages has become a just-in-time activity performed with the help of online resources (e.g., Stack Overflow) [36, 46, 16]. Many projects are now global and open to the public through online repositories (e.g., GitHub[1] and Bitbucket[2]), collaboration is not limited by geographical barriers, and a new type of programmer has emerged: *the social programmer*.

In contrast to traditional programmers, multiple sources of information make awareness one of the main issues that social programmers have to overcome on a daily basis. According to Storey *et al*. [46, 45], the variety of channels available and personal preferences or company standards, imposes the social programmer to use multiple channels in unison. Regardless of how social programmers select their preferred channels, they have to invest time in learning the way each channel works. Also, channels are becoming increasingly complex with more options for communicating, making media literacy a complex issue.

These changes have attracted much of attention in the academy, and researchers have identified various aspects of media channels within communities of practice. For instance, we have algorithms to detect experts on social channels [30, 31], models that explain the propagation of information through channels [21, 20], an understanding of the relationships

[1] `https://github.com/`
[2] `https://bitbucket.org/`

between the evolution of the community and its products [11], and discovered ways that social programmers are using media channels [40, 39, 32]. However, there are still many issues that current programmers need to understand, including the synergy between media channels and the way media channels are affecting communities of practices. Based in my review, just a few researchers have investigated these topics. Bird *et al*. [7] correlated the activity in mailing lists with the activity in source code; Storey *et al*. [45, 46] identified the role of social media in software engineering; Kavaler *et al*. [22] identified a complementary perspective on using APIs and the questions asked on Stack Overflow; and, Vasilescu *et al*. [52] investigated the interplay between Stack Overflow and the software development process, which were reflected on changes committed in a source code management system (i.e., GitHub).

The remainder of this chapter describes the background elements of this study, including the definition of media channels and communities of practice. It also introduces the R community along with the two main Q&A media channels selected for this inquiry: the R-help mailing list and the Stack Overflow.

2.1 Media Channels

According to the Oxford dictionary, a medium[3] is *"a means by which something is communicated or expressed"*. Furthermore, a channel is *"a method or system for communication or distribution"*. Taken together, a media channel[4] *"is a method or system by which information is communicated or distributed to others using different means"*.

From the aforementioned definition, we know that a media channel is composed of users, messages, and a channel. *Users* are the active part of the media channel and are also responsible for the creation of messages. *Messages* contain the knowledge that is to be transmitted to the receiver and can take different forms depending on the channel's characteristics (e.g., text, graphics, video, sound, or a combination of characteristics). The *channel* provides a method or system to coordinate, communicate, collaborate and share knowledge with other users [45].

According to Storey *et al.* [45], channel affordances are affected by their characteristics. Therefore, depending on the channel, some tasks are easier to accomplish than others. For instance, Stack Overflow is changing the way in which developers collaborate, share knowledge, learn, and communicate among themselves [45], and may even replace the

[3] http://www.oxforddictionaries.com/definition/english/medium
[4] http://www.oxforddictionaries.com/definition/english/channel

mailing list usage [50]. This is a consequence, according to Vasilescu *et al*., of Stack Overflow's gamification system, rich interface, and social media features.

Other authors have focused their efforts on different components and aspects of channels. Treude *et al*. categorized questions according to their topic [48], Asaduzzaman *et al*. [3] investigated the characteristics of unanswered questions, and Jiang *et al*. studied the way messages are disseminated on social coding sites. Lastly, Vasilescu *et al*. proposed a method to quantify the risk of not having maintainers for code implemented in a certain programming language [54].

2.2 Community of Practice

According to Wenger [58], a community of practice is defined as *"groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly"*. In contrast with formal work groups and project teams, community members are part of the community by their own will [58]. Members work towards a common objective, learning from each other, and helping each other in the process.

The core components of a community of practice are the domain, the practice, and the community [56]. The *domain*, or shared interest, defines the identity of the community. The *practice* identifies members of a community as *practitioners* that are constantly developing and sharing a set of resources (e.g., tools, documentation, histories, or experiences) to address recurring problems. While the *community*, comprises the activities in which members engage in discussions to help each other share information, enabling them to learn from the community.

A community of practice is more than the sum of its parts. It helps members solve problems quickly, transfer best practices, develop professional skills, identify experts, form social bounds between members, and drive strategies [56, 45]. Communities of practice also accumulate and update knowledge through practitioners [57], enabling them to take a collective responsibility for managing the knowledge according to their needs [56]. Wenger [56] proposes that given the proper structure, practitioners can be the best option to manage the construction of knowledge (e.g., Stack Overflow).

Communities of practice are like living organisms, evolving and adapting according to their context, producing new tools for the community, and external sites. Communities change their practices and structure regularly while adapting dynamically to new situations. For example, Mozilla adopted the Mercurial tool [33] and changed their version release strategy [23] as a way to keep up with a fast changing business environment.

2.3 The R Community

The R project[5] was born in 1993, as a free and open source programming language and software environment for statistical computing, bioinformatics, and graphics [17]. R is an implementation of the S programming language combined with lexical scoping inspired by Scheme. It was created by Ross Ihaka and Robert Gentleman and is now developed by the R Development Core Team. Today, the R community contains more than 2 million users, classified into two groups: 1.*R-core* (with 20 users) consists of the software development team that maintains and evolves the R language, and 2.*Periphery* includes everybody else (i.e., language users and package developers).

I chose to study the R community because it exemplifies a typical open source community, and has been evolving for almost 20 years. It provides broader relevance outside the software development community, since it includes users with no or limited programming experience (e.g., biologists or statisticians). Its entire history of mailing list communication is archived and publicly available. Recently, the R community was also the subject of extensive research in community evolution [11] and the interplay between channels [53].

In this thesis, I wanted to identify the interplay between media channels that serve the same purpose within a community. Thus, I have focused my efforts on analysing the R-help mailing list and Stack Overflow. As media channels, the R-help mailing list and Stack Overflow provide similar benefits to the R community (i.e., Stack Overflow[6] and R-help[7]). The R-help mailing list and Stack Overflow are one of the many channels available within the R community. However, I chose them because they are the channels which description are more similar in terms of the community support.

R-help Mailing List

Among the communication channels that the R community uses (e.g., SVN, bug tracker system, and R Journal), there is a group of mailing lists devoted to helping community members answer questions and solve problems related to programming and the R language: the R-help, R-package-devel, R-devel, R-packages, and Bioconductor mailing lists. Through email, R users can send their questions to different mailing lists depending on the topic. Members subscribed to the R mailing lists can contribute by answering the user

[5] https://www.r-project.org/
[6] http://stackoverflow.com/tour
[7] https://www.r-project.org/mail.html

directly or posting to the list. In the last case, the email subject is kept as an identifier for the reader.

The main objective of the R-help mailing list is to discuss problems and solutions using R. However, other messages are encouraged, such as announcements (not covered by 'R-announce' or 'R-packages'), documentation of R, benchmarks, and examples using R. It is worth noting that the R-help mailing lists are used by people who want to use R but are not necessarily knowledgeable about (or interested in) programming. As a mailing list, R-help does not provide a user interface to manage the email threads.

The R-help mailing list used to be the main media channel for asking and answering question within the R community. According to Vasilescu *et al*. [53], a significant number of users migrated from the R-help mailing list to Stack Overflow. Despite the reduced number of users, the R-help mailing list is still a very active list; on average, a subscriber can receive 55 emails a day.

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 web-site improves participation and provides strong support for creating and sharing knowledge as well as encouraging informal mentorship [19, 45]. Meanwhile, gamification provides a system based on reputation points and badges to reward users' participation[8], thus earning points that enable functionality inside the site. For example, 20 points allow users to participate in the site's chat rooms, 100 points allow users to edit wiki posts, 2000 points allow users to edit questions and answers, and with 25000 points, users can access site analytics. Stack Overflow also provides trophies for display in users' profiles[9], and a bounty reputation system to attract the interest of unanswered questions. According to various studies, gamification mechanisms boost participation [49] and enable mutual assessment [37].

Stack Overflow's interface is rich with information. Figures 2.1 and 2.2 depict the interface separated into two sections. Figure 2.1 describes the post in relation to the *question*. The elements are numbered from 1 to 8, and are described as follows: (1) the title of the question; (2) the number of positive votes for the question, as well as two buttons (up and down arrows) to allow users to vote up (positive) or down (negative); (3) a star button to mark the question as a favourite and the number of users that have marked the question as

[8] `http://stackoverflow.com/help/privileges`
[9] `http://stackoverflow.com/help/badges`

such; (4) tags applied to the question; (5) a button to add a short, text-based comment to the question (posted below the button); (6) the body of the question which might contain, along with the description, other aids such as images, source code, examples, and links; (7) the last user that edited the question along with their reputation points;. and (8) information about the user who posted the question, including their alias, silver and copper badges, and the date of the posted question.

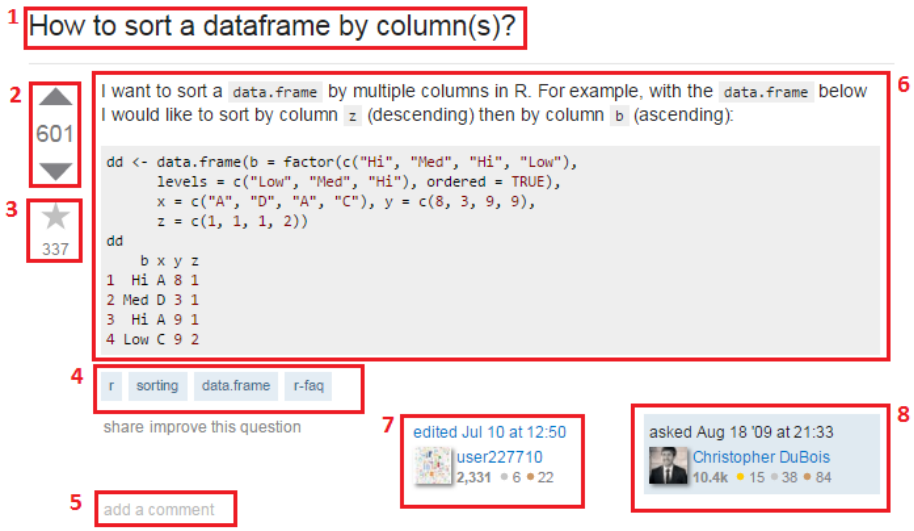


Figure 2.1: Stack Overflow interface [Question section]

Figure 2.2 shows the post in relation to the *answer* located below the question in the interface. The elements are numbered 1 to 8, and are described as follows: (1) the number of answers provided to the question; (2) sorting buttons that allow users to display the answers by latest activity, oldest first, or most recent first; (3) the number of positive votes for the answer, as well as two buttons (up and down arrows) to allow users to vote up (positive) or down (negative); (4) a check mark to indicate that the owner of the question marked the answer as the solution to the question; (5) the body of the answer which might contain, along with the proposed solution, other aids such as images, source code, examples, and links; (6) the last user that edited the question along with their reputation points; (7) information about the user who posted the question, including their alias, silver and copper badges, and the date of the posted question; and (8) the comments to the answer, which are fairly short and limited to include only text.

The adoption of social media has occurred at a much faster rate than any previous communication technology [29]. In the last decade, Stack Overflow has become the most popular media channel for answering software development related questions, nearly replacing

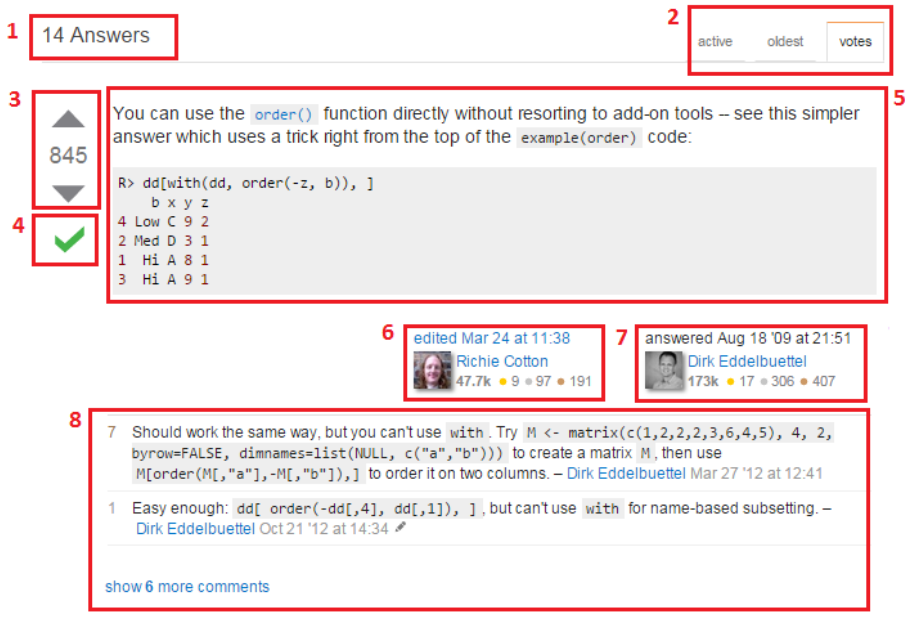


Figure 2.2: Stack Overflow interface [Answer Section]

previous methods of communication that accomplished the same objective (e.g., mailing list) [53]. Figure 2.3 shows the number of questions asked each month on Stack Overflow, Cross Validate and the R-help mailing list (TOP), and the number of questions answered on the R-help mailing list (after September 2008) and Stack Exchange each month (BOTTOM). Despite Stack Overflow's advantages over Q&A mailing lists such as the R-help (i.e., social network features, gamification environment and rich visual user interface), there are still many users who prefer the latter. Later in this thesis, we learn about the way programmers use Stack Overflow and the R-help mailing list to gain and share knowledge.

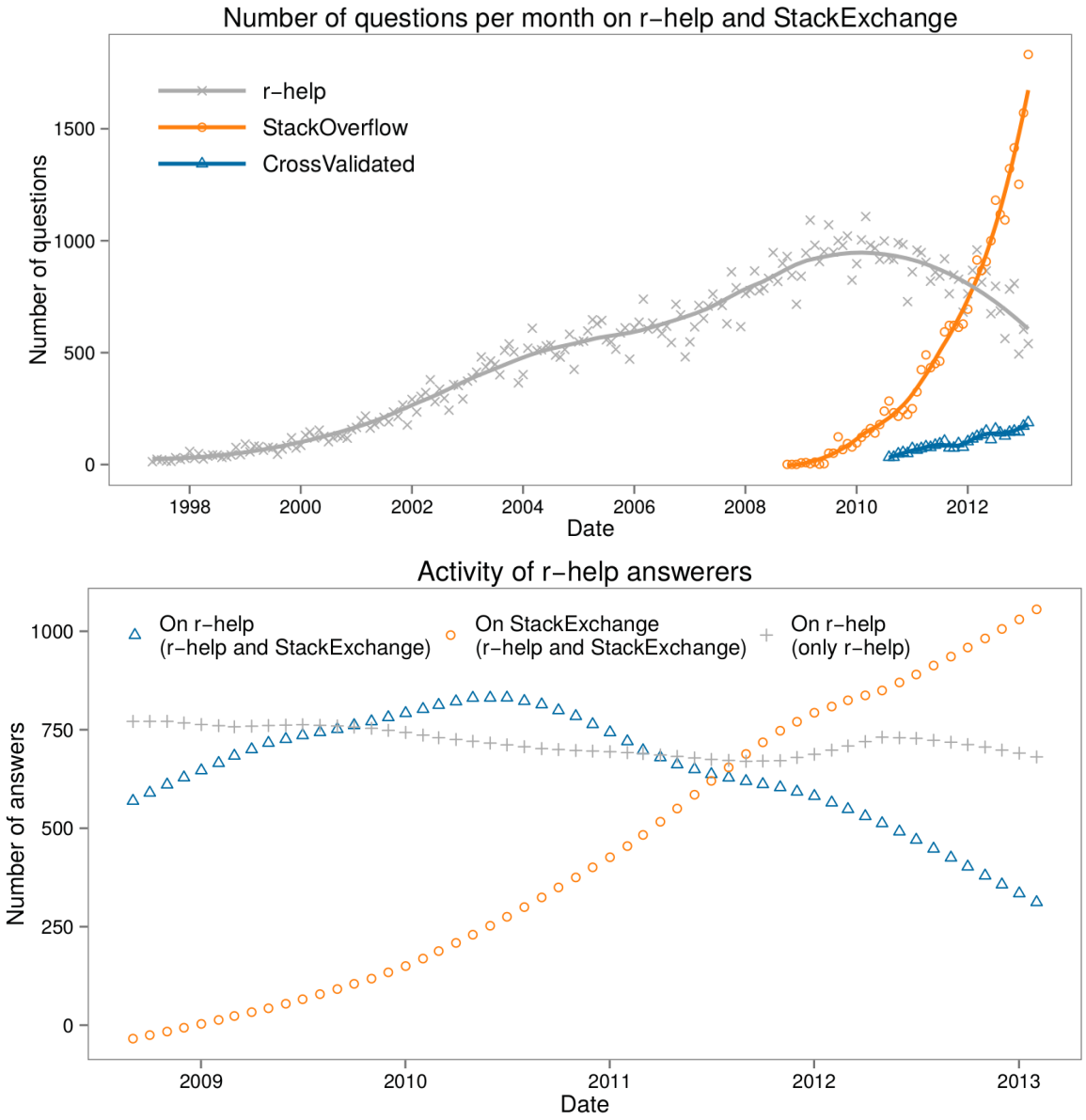


Figure 2.3: (TOP) Number of questions asked (threads started) each month on R-help and Stack Exchange (Stack Overflow and Cross Validated) [53]. (BOTTOM) The number of questions answered on the R-help mailing list (after September 2008) and Stack Exchange each month: participants exclusive to the mailing list versus those also active on Stack Exchange [53].

Chapter 3
Methodology

This chapter describes the elements of the methodology, including the research questions, the adopted case study methodology, and the phases of the study. This chapter also outlines the procedure used to collect and analyse the data in this study.

3.1 Research Questions

The four research questions that guided this thesis are:

RQ-1. What types of knowledge are shared on Stack Overflow and the R-help mailing list within the R community? In the R community, the R-help mailing list serves the same purpose as Stack Overflow. This led to the question of *what types of knowledge are shared on Stack Overflow and the R-help mailing list?* To answer this question I proceeded to analyse and categorize the knowledge in questions, answers, updates, comments and flags on Stack Overflow and the R-help mailing list. Based on the analysis I was able to contrast the way knowledge flows through Stack Overflow and the R-help mailing list.

RQ-2. How is the knowledge constructed on Stack Overflow and the R-help mailing list? As discussed before, Stack Overflow and the R-help mailing list support the R community. Such a statement implies that the interactions hosted by these two media channels are of a collaborative nature. I wondered if the same applies to the creation and sharing of knowledge in these two channels. My goal was to identify the mechanisms and strategies on Stack Overflow and the R-help mailing list used to construct knowledge collaboratively and individually (if any).

RQ-3. How does the sharing of links on Stack Overflow and the R-help mailing list support knowledge construction? On the Internet, links support the reuse and referencing of data from other resources. Links contain information that is valuable for messages, and depending on how they are used, links can support knowledge sharing practices in different ways. For instance, a link can expand what is known about a topic by referencing more complete sources of information, or provide data to reproduce certain behaviours on source code examples. Previously, I have identified the types of links on Stack Overflow and how they support diffusion of knowledge [13]. For this study, I pursued the identification of how links contribute to the construction of knowledge. Thus, I categorized links posted in the body of messages on Stack Overflow and the R-help mailing list based on their type (e.g., Q&A Website, and Forums), and how each type of link supported the knowledge construction.

RQ-4. Why do certain users post to both Stack Overflow and the R-help mailing list? As mentioned by Vasilescu [50], there is a group of users that are active on Stack Overflow and the R-help mailing list. I wondered if there were any advantages or disadvantages on using both channels. With that in mind, I identified a list of active users in both channels and used open coding methods to analyse their posts.

3.2 Case Study Methodology

As claimed by Yin [59, 60], a case study facilitates the exploration of a phenomenon within its context using a variety of data sources. In software engineering, a *case study* is defined as *"an empirical enquiry that draws on multiple sources of evidence to investigate one instance (or a small number of instances) of a contemporary software engineering phenomenon within its real-life context, especially when the boundary between phenomenon and context cannot be clearly specified"* [34].

According to Yin [60], a case study should be used when: (1) "How" or "why" questions are trying to be answered; (2) the researcher cannot manipulate the behaviours of those involved in the study; (3) the context is an important part of the study; (4) there are no clear differences in what is happening between the phenomenon and the context; and (5) when multiple sources of evidence have to be covered. Because these conditions apply to the nature of this study and its research questions, I selected the case study methodology for this work. Specifically, this thesis is an exploratory case study to explain the interplay of multiple media channels within a community in terms of the knowledge created and

shared.

The study is divided in two phases that were performed in parallel: mining of data archives, and the survey. Figure 3.1 depicts the general organization of the study design. In the next sections of this chapter, each phase is explained in detail.

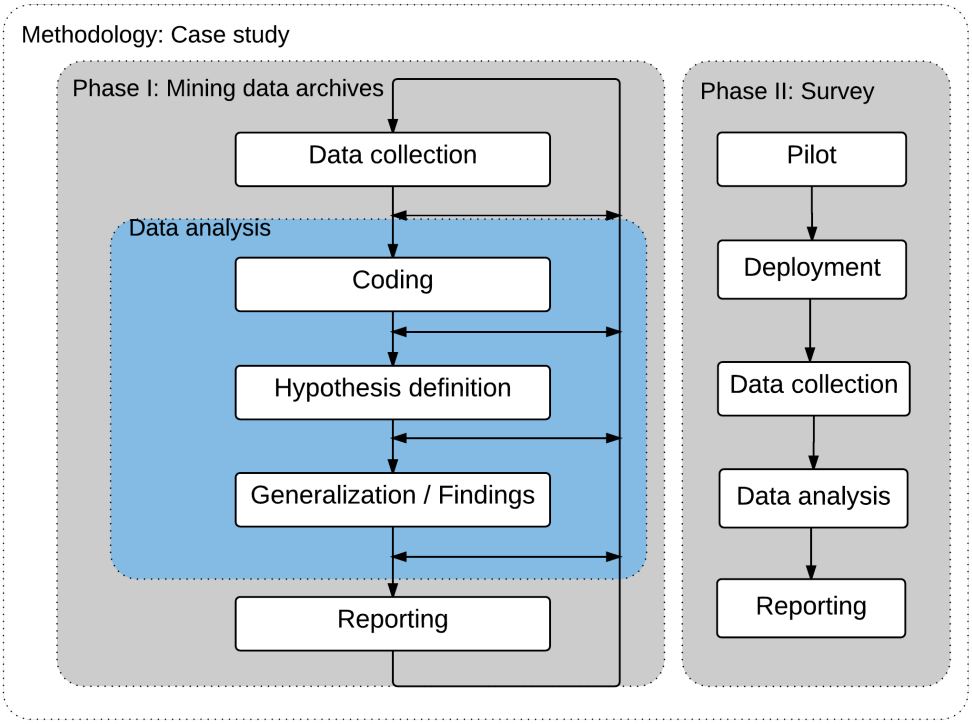


Figure 3.1: General overview of the study design

3.3 Phase 1: Mining Data Archives

The mining of the data archives method involved a three step process: data collection, data analysis, and reporting [34]. The **data collection** step involved in gathering the body of data required for analysis. This data was a selected set of R-related posts from Stack Overflow and the R-help mailing list. In the **data analysis** step, I analysed the data looking for answers for the research questions. Finally, the **report** step, consolidated the results, which are presented in Chapter 4.

3.3.1 Data Collection and Preparation

Stack Overflow and the R-help mailing list store their messages in publicly available archives. The records available for Stack Overflow start in 2008 (the birth of Stack Overflow), while the R-help archives go back to 1997. To make both data sets comparable, I analysed the data from 2008 until 2013, a period of time that both channels were available simoultaneously.

Users can obtain Stack Overflow's archived message data using a variety of different mechanisms: (1) directly, through the Stack Overflow Website, (2) using Stack Exchange online query services, or (3) through a dump file[1], containing data from all the Stack Exchange Websites in XML[2] format (a new version is released every three months). For this thesis, I used the data provided by the dump file. The R-help mailing list data (i.e., emails sent to the list) is available through the R community Website as MBOX[3] files organized monthly from April 1997 until January 2015.

To prepare the data, I used two software tools depending on the data set. (1) to process the Stack Overflow data, I used a modified version of Sam Saffron's application, So-Slow[4]; and, (2) to process the R-help mailing list archives, I wrote a software application, based on the Bettenburg *et al* [6] recommendations of how to process mailing list data. I followed the process depicted in Figure 3.2. First, I extracted the archived data. Then, I used the aforementioned tools to pre-process the archive files, and then load the data into a database. Next, I executed custom queries to obtain random samples data to analyse.

Table 3.1 depicts a summary of the data uploaded into the database. The R-help has more questions, answers, and users than Stack Overflow, due the fact that there is approximately ten years of additional data. Only Stack Overflow's data contains "comments" information, so this field is empty for the R-help mailing list column.

The following subsections detail the analysis process for each media channel.

The R-help Mailing List

As stated earlier, the R-help archives are in the MBOX format. However, the information inside of the email is still unstructured data. The MBOX format separates the metadata (header) from the content (body), but there is not a clear division between what are source code examples, the sender's message and signature, and other semantic elements that might

[1] https://archive.org/details/stackexchange
[2] https://en.wikipedia.org/wiki/XML
[3] https://en.wikipedia.org/wiki/Mbox
[4] https://github.com/SamSaffron/So-Slow

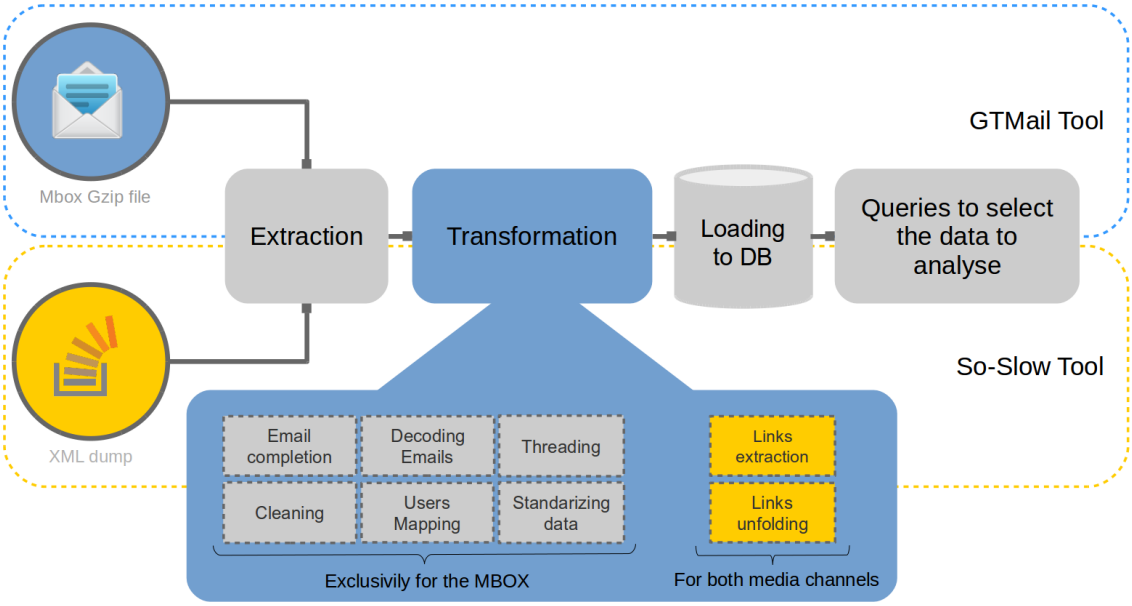


Figure 3.2: Process applied to the data of both media channels.

Table 3.1: Raw data collected for each channel.

Type	R-help	Stack Overflow (r—tag)
Questions	101,931	67,393
Answers	213,366	99,620
Comments	-	286,124
Users	39,150	26,324

exist on emails. The MBOX format only provides certain information about the email, such as the sender and receiver's email addresses, the subject of the email, and the emails on the thread. Figure 3.3 depicts an example of a well-formed MBOX file. In the top, a clear defined information that can be extracted from the email (header), but the body is unstructured. To analyze the body, it has to be cleaned from noisy data such as signatures and quoting text.

Bettenburg *et al.* [6] proposed a series of recommendations for proper processing of mailing list data, to ensure accurate research results. In my search of existing tools, I found a couple used for research that handles MBOX data, such as Herraiz *et al.* tool, MailingList-

```
From alias@domain.xxx DAY MON   dd 24H:MM:SS YYYY
From: alias@domain.xxx (<author name>)
Date: <Date>
Subject: <Subject>
In-Replay-To: <replay hash id>
References: <list of email hashes on the thread>
Message-ID: <email hash id>

<body>

On <date>, <replay author name> wrote:
> <reply body>
```

Figure 3.3: Example of a well-formed email in the MBOX format.

Stats[5], and REmail[6] tool. However, I could not find evidence of how MailingListStats was constructed, or if it is resilient to MBOX format inconsistencies. While the REmail tool was meant for a totally different purpose— to match source code with emails from projects' archives.

To pre-process the R-help archives, I created a Java application based on Bettenburg's recommendations that: (1) standardizes the MBOX format considering spacing and email address formatting, (2) extracts information from the MBOX files like sender's date, subject and message, (3) groups e-mails into threads using Jamie Zawinski's algorithm[7] which provides support for sub-threading (threads that might exist at the inside of a main thread), (4) removes duplicated emails, (5) removes URLs in footnotes and signatures, (6) reconstructs threads when neither *Reference* nor *Reply* appear in th header, but the body of the message shows text from previous emails (for this purpose, I matched e-mails by subject and organized them by arrival time),(7) extracts URLs and unfolds shortened URLs, (8) downloads emails with coding problems from the URL left by the mailing list server after scrubbing the body, (9) solves text encoding issues (i.e., text that it is not in UTF-8 format), (10) transforms the email addresses to MD5 hashes, (11) changes the creation date (the R-help mailing list time zone is UTC+2) to UTC (Stack Overflow's server time), and (12) uploads the data to our database. Figure 3.4 depicts the entity–relational model used to store the data from the mailing list. Examples of how the tool stores data, and threading are presented in Appendix D.

[5] https://github.com/MetricsGrimoire/MailingListStats
[6] https://code.google.com/p/r-email/
[7] https://www.jwz.org/doc/threading.html

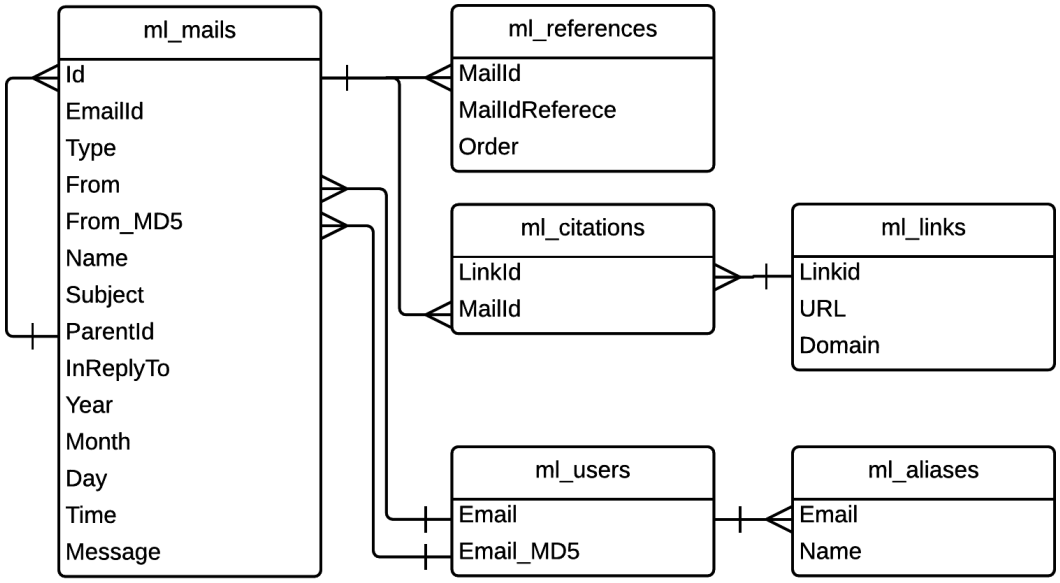


Figure 3.4: Entity Relation Diagram of the R-help mailing list data.

Stack Overflow

Every three months, Stack Exchange releases a new data dump file in XML format that contains data from all their Websites[8]. However, the last dump file containing email addresses as MD5 hashes was released in September 2013— the Stack Overflow dump files produced after September 2013 do not provide users' email addresses. The email hashes were used to match users from Stack Overflow with users in the R-help mailing list. This technique was used to answer RQ4 and is explained later in this chapter. Because of this, I used the data dump file from September 2014, but updated the table `users` with the hashes in the dump file from September 2013, for whose `IDs` were identical in both data sets. In case a user from the 2013 data file did not exist in the 2014 data (e.g., as consequence of the *right to be forgotten*[9]), I ignored the user.

As stated previously, I used a modified version of Sam Saffron's application, So-Slow. The purpose of this is to extract the information in the file using XML tags (e.g., post, user, and comment), and load it in a database. I filtered all R-related data by selecting only messages with the R tag (i.e., `r`) and its synonyms[10] (i.e., `rstats` and `r-language`). Figure

[8] `http://stackexchange.com/sites`
[9] `https://en.wikipedia.org/wiki/Right_to_be_forgotten`
[10] `http://stackoverflow.com/tags/r/synonyms`

3.5 depicts the entity–relational model used to store the Stack Overflow data.

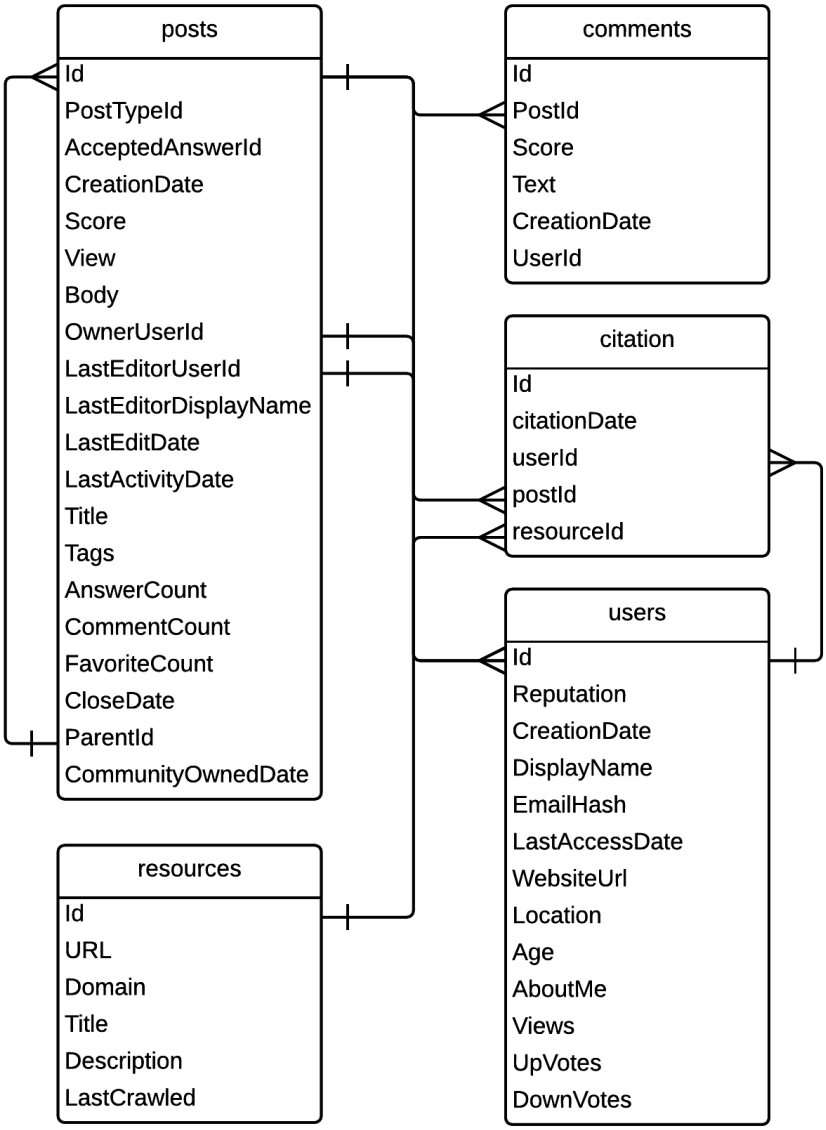


Figure 3.5: Entity–Relation Diagram of the Stack Overflow data.

Data Merging

There are some studies that propose different techniques for merging users' identities by analysing the data from multiple repositories (e.g., mailing lists, bug tracking information, and source code management tools) [7, 24, 53]. Bird *et al*. [7] proposed a heuristic to

match users' identities across multiple mailing list archives by combining parts of user names and email addresses. For example, the *cagomezt* prefix is likely to belong to *Carlos Arturo Gomez Teshima*. Furthermore, Kouter *et al*. [24] used a natural language processing technique called Latent Semantic Analysis to merge identities on very noisy data. However, it has been demonstrated that all existing approaches produce false positives and false negatives [12].

For this work, I used the approach proposed by Vasilescu *et al*. [53], which is the most conservative technique considering the available data [50]—it does not use any method to infer email addresses based on user name. Vasilescu's technique consists of matching Stack Overflow's email MD5 hashes with the MD5 hash version of email addresses from the R-help mailing list data. With this technique, the resulting set included 1,421 different users with the same email address on both media channels.

Because Stack Overflow only provides the email addresses as MD5 hashes, and to make both data sets comparable, the mailing list emails were converted to their corresponding MD5 hashes.

It is important to note that MD5 hashes are not *collision resistant*[11] and therefore, this could possibly lead to false positive resistant outcomes. However, it is unlikely for two different email addresses to share a MD5 hash. According to the Request for Comment (RFC) 1312[12] from the Internet Engineering Task Force (IETF), the probability to find a MD5 collision is less than $1/2^{64}$.

Data Analysis Process

I used a qualitative data analysis approach to study the data that flows through Stack Overflow and the R-help mailing list. A qualitative and exploratory approach best suits research when a concept or phenomenon requires more understanding, since there is little pre-existing research [10].

In particular, I used the inductive approach of Runeson *et al*. [34] to analyse the data from Stack Overflow and the R-help mailing list. This approach is iterative, across the study it is necessary to switch between data selection and data analysis, or between data reporting and data collection. To reduce bias, it is advised the involvement of multiple researchers [34]. As a consequence, this study was conducted by two researchers, both computer scientists with a background in qualitative data analysis.

[11] `https://en.wikipedia.org/wiki/Collision_resistance`
[12] `https://www.ietf.org/rfc/rfc1321.txt`

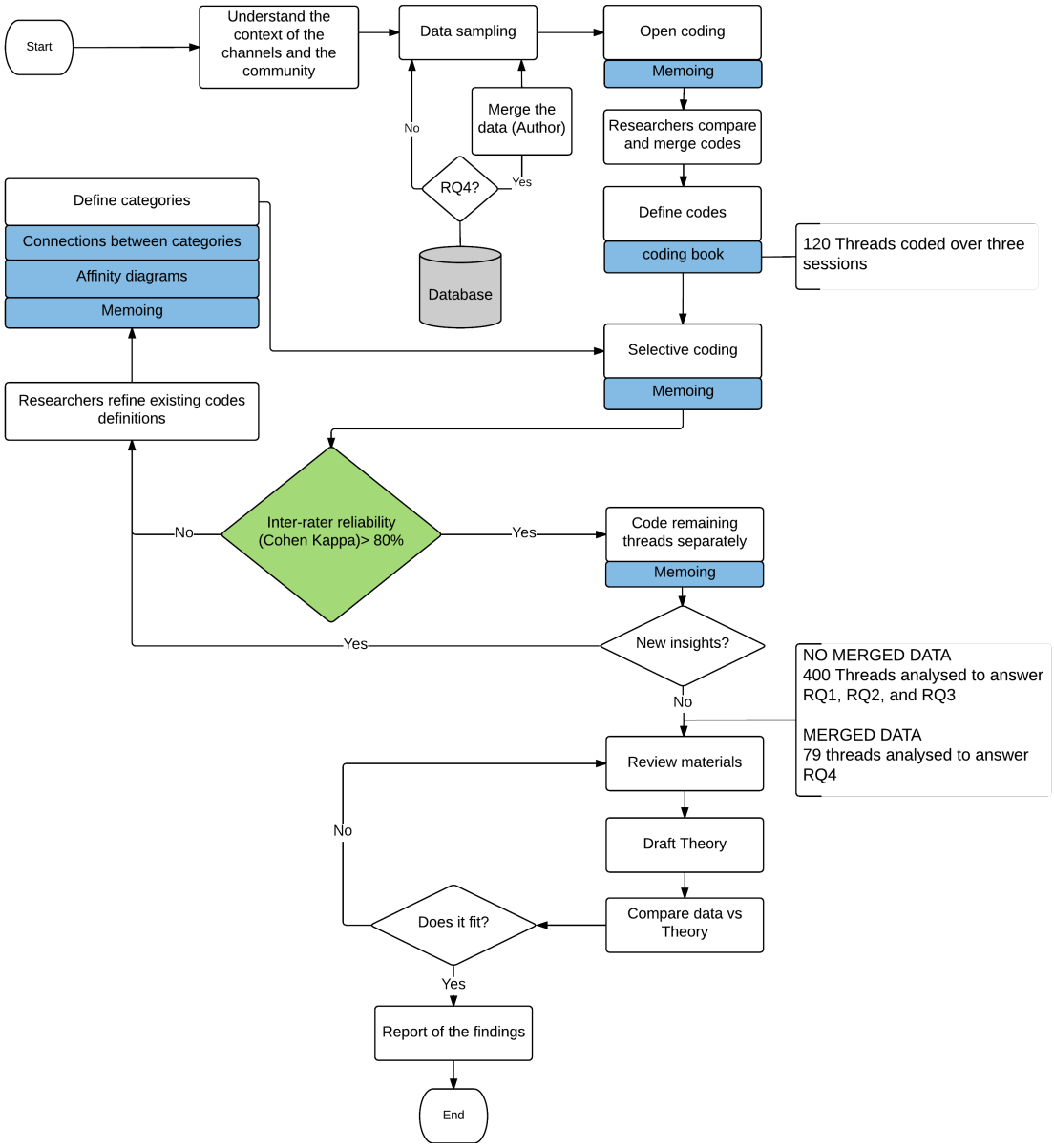


Figure 3.6: Qualitative approach used to analyse Stack Overflow and the R-help mailing list data. This chart shows the process and techniques (coloured figures) used to analyse and develop the findings of this work.

Figure 3.6 depicts a visual explanation of the data analysis process for this study. My colleague and I refined our codes and categories by repeating numerous times the process of collecting and analysing the data.

The next sections, for the sake of clarity for the reader, have been presented in a linear fashion. However, the process as depicted in Figure 3.6 is not linear.

Techniques Used to Support the Analysis

Figure 3.6 contains some coloured shapes that depict the techniques used to support the data analysis, which are explained as follow:

- **Memoing** refers to the act of taking notes (coding) about what the researcher is learning from the data during the analysis [14], for example, the hypotheses regarding a code, and relationships between concepts. During this stage, reflective memos were written in a spreadsheet next to the applicable codes as the researcher coded (see figure 3.7). These memos were used to create the codes, and hypotheses about the relationships between concepts.

- **Affinity diagrams** is a technique that allows one to organize ideas and data into groups and to find the relationships between concepts [35]. During the study I used affinity diagrams when discussing new insights with my colleague, and while defining categories and relationships between them.

- **Inter-rater agreement *Cohen Kappa*** is a coefficient used to measure the agreement between two coders who classify items into mutually exclusive categories [43]. Ladis and Koch suggest that values above 0.60 or 60% to obtain substantial results [18]. In a previous study [13], we used the same coefficient to measure agreement between coders. Based on this experience, I set a value above 0.80 or 80% as the minimum to obtain substantial results. In this study, I used the Cohen Kappa coefficient after each coding session as a way to trigger discussion.

- **Code book** is the book that contains the definitions of the codes that the researchers look during the data analysis [27]. Codes are the building blocks for theory and foundations on which the researcher's argument rest. We coded an initial set of 120 threads over three sessions. In each session, my colleague and I separately coded 40 threads. The multiple sessions allowed us to refine definitions in the codebook. Each entry in the code book is associated with a title, a formal definition, an example, and space for notes from the researcher. The final version of the codebook with the corresponding categorizations are detailed in Chapter 4.

The Analysis Process

The focus of the analysis is to *understand the context of the media channels and the community*. The process consisted of: First, a recollection of the official information for both

channels and the community to build a background of the community of practice and the channels studied. From the channels, I collected posting guides, rules, channel objectives, and competitors, whereas from the community I collected the number of members, how it works, and the media channels that the community uses.

Second, a mapping between messages from Stack Overflow (i.e., question, answer, update, comment, and flag) with messages on the R-help mailing list. This is to overcome how the data is structured in both channels. Stack Overflow has a clear delimitation of what is a question, an answer, a comment, a flag and an update, while the R-help mailing list is just plain text. The mapping of messaged between both channels was as follows:

- **Question:** the message is the first on the thread, and it contains the main question.
- **Answer:** the message provides a solution to the main question of the thread.
- **Update:** the message claims for a modification to a question (or answer) made by the author of such a question (or answer).
- **Comment:** the message offers a clarification to a specific part of the question or answer.
- **Flag:** the message requests attention from the moderator (e.g., repeated questions, spam, or rude behaviour).

Next, for the *data sampling* step, to answer RQ1, RQ2 and RQ3, I used a simple database query that selected a time frame and randomly returned threads from each channel. The data set was capped at 400 threads for each channel (0.4% and 0.6% of the data available at the time of writing this thesis for the R-help mailing list and Stack Overflow respectively), when my collaborator and I deemed our observations as being saturated. To answer RQ4, I used the same query as before, but I added a condition that matched, on both channels, messages with the same subject written by the same author (we merged the data). Given that only 79 threads were returned from this query, my colleague and I analysed the entire population available. The executed queries are presented in Appendix B.

To code the data, we used an *open coding* technique that involves reporting *what the researcher saw* during each coding session. The researcher has to keep in mind, all the time, the objective of the study and perform the coding based on it. Each researcher coded the data on a separated way. During the coding session, we wrote memos as needed, and marked repetitive patterns. Later, we met to compare and discuss findings, and begin developing codes.

From our initial codes, we began the process of creating a *coding book* to outline definitions. This set of codes were used later during the *selective coding* step. At this point, the researcher stops coding every occurrence, and begins seeing larger trends and connections within the data and codes. It is possible that during the *selective coding* step some codes have to be reformulated, or maybe split into more codes. Also, it is possible to formulate completely new codes as needed. Whenever there is a new code or any is changed, it is necessary to go back and recode the material.

As a coding tool, my colleague and I used a spreadsheet in which each row represents a message of the thread. If for any reason a message appeared to fit in more than one category, each researcher selected, at their own discretion, a primary category to represent the message. Figure 3.7 depicts an example of the coding spreadsheet that we used. The number in the first column identifies if the message is a question, an answer or a comment. For instance, if the number assigned to the question was "1", then the answers were enumerated with consecutive numbers separated by a point (e.g., 1.1); and the comments were enumerated in a similar way to enumerated answers, but using three numbers: the first number represents the question, the second represents the answer, and the third represents the comment consecutive (e.g., 1.0.1). The second column contains the message, the third column the channel, the fourth column the question categorization, and so on. The last column contains the URL to the thread on the channel. Inside each cell, a semicolon (;) represents a sub-category, and the double pipe (||) divides two different ideas (e.g., in the *MEMOS* column), or indicates that a message was re-classified after an update (e.g, *ANSWER* column).

At the beginning of the coding, before we created the code book, the spreadsheet had only the *ID*, the *MESSAGE*, the *MEMOS*, and the *URL* columns. During each iteration, the spreadsheet was updated with the classification and type of messages that my colleague and I were defining.

Originally, both researchers read the threads directly from the spreadsheet. However, this method of reading turned out uncomfortable, and we fell back to read the threads directly from each channel rather than the spreadsheet.

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID	Message	Channel	Question	Answer	Update	Comment	Flag	Resource	Knowledge construction	Memos	URL
2	1	MessageId: 1716012 Subject: Stopwatch function in R Date: 2009-11-11 15:39:27	SO	Environment	'-	Expansion; Non-laballed	'-	'-	Official documentation;Expand	Participatory	Software development questions || Solving an error || Should be flagged as Debuggin || Many answer written in 2015 ||	http://stackoverflow.com/questions/1716012/s
3	1.0.1	for what it's worth, googling [r] tic toc now (Feb 2013) gives lots of answers; don't know	SO	'-			Solution/Alternative		'-	'-	Comment written 2 years later	
4	1.1	"There are plenty of profiling tools in R, as Dirk mentioned. If you want the simplicity of tic/toc, then you can do	SO	'-	Clue || Source code	Expansion; Labelled	'-	'-	Official documentation;Expand	Participatory 1.3; Confirmation	After the update the user provide tailored code to solve the question of the user. || cannibalised MATLAB package to write	
5	1.2	There is a MATLAB emulation package matlab on CRAN. It has implementations of tic and toc (but they	SO	'-	Redirecting	'-	'-	'-	'-	Participatory 1.2; Alternative	Answer with package || Credited answer from author of 1.3 || embedded code from that show the function on the package	
6	1.3	"Direct equivalents of tic and toc do not exist. Please see help(system.time) as	SO	'-	Clue	'-	'-	'-	Official documentation;Expand	'-	First to answer (8 minutes after the question)	
7	2	MessageId: 1340054801327-4633754.post Subject: [R] (1-1e-100)==1 true?	RH	Discrepancy	'-	'-	'-	'-	'-	Crowd	Well explained question	http://r.789695.n4.nabble.com/1-1e-100-
8	2.1	This is standard behaviour for a floating-point computational system like R. You might like to	RH	'-	Redirecting	'-	'-	'-	Official documentation;Expand	Crowd		
9	2.2	You are working close to, and also beyond, the boundary of what R can internally	RH	'-	Explanation;Statistical	'-	'-	'-	'-	Crowd	Teaching || Statistical explanation	

Figure 3.7: Example of the way we coded the data. Each row of the spread sheet represents a message of the thread. Questions, comments, and answers can be identified with the number on the first column. The columns in yellow contain the codification for each message type. The last two columns contain the memos and the URL of the thread.

3.4 Phase 2: The Survey

In *phase 2*, I conducted a survey[13] with members of the R community with the purpose of obtaining additional insights on the findings. First, I created a draft of the survey and did two pilots: (1) with colleagues in our research group, and (2) with R users at the University of Victoria. The objective was to test and refine the questions, tone, rankings, and the format of the survey. The survey questions were structured into five sections: (1) the user, (2) Stack Overflow use, (3) the R-help mailing list use, (4) Stack Overflow and the R-help mailing list if both used, and (5) resources linked to used. Survey's sections 2, 3 and 4 would only become active if the participant was a user of the channel.

I announced our survey on Twitter, Reddit, the R-help mailing list, and Meta Stack Exchange to reach users of both channels, and minimize the selection bias. However, on Stack Exchange the announcement was not well received and therefore was deleted a few minutes later after posting it. I received 26 valid responses out of 32 from the R community members. The survey did not collect any personal information. The questions posed in this survey are listed in Appendix C.

[13] A copy of the survey is published in `http://goo.gl/mxmH5J`

Chapter 4
Findings

This chapter presents: (1) a characterization of the non-mutually exclusive categories and properties of Stack Overflow and R-help mailing list according types of knowledge the channels contains; (2) remarks about the ways knowledge is constructed on these two media channels; (3) an explanation of how links support the construction of knowledge; (4) a characterization of the knowledge based on the analysis of active users using both channels; and (5) interesting remarks about users regarding their behaviour on these two media channels.

4.1 RQ-1. What types of knowledge are shared on Stack Overflow and the R-help mailing list within the R community?

As mentioned above, the R-help is not a specialized mailing list, therefore we were motivated to investigate whether Stack Overflow shares the same types of knowledge as R-help. As a result, we were able to identify that the messages from the Stack Overflow R tag and R-help mailing list contain five types of knowledge: (1) Questions; (2) Answers; (3) Updates; (4) Flags; (5) Comments.

Questions

Questions express one or more problems or doubts that a Stack Overflow or R-help user is experiencing. Through our coding my colleague and I identified 10 type of categories related to questions. This is explained as a result of the analysis of the R-help mailing list that by definition is more flexible on the topics allowed on the channel (e.g., announcements)

Our 10 categories related to *questions* are:

(1) How-to: Questions that ask how to do something specific.

(2) Bug/Error/Exception: Questions that ask for a solution or reasons for a error message.

(3) Discrepancy: Questions that ask about an unexpected result of a specific function, process, or package.

(4) Set-up: Questions that ask for possible ways to set up the R environment before or after deployment.

(5) Decision help: Questions that ask needs help making a decision.

(6) Conceptual/Guidance: The user requests a conceptual clarification or guidance on topics related with R or statistics.

(7) Code reviewing: Questions that ask for a code review explicitly or implicitly.

(8) Non-functional: Questions that ask for help (or suggestions) with a non-functional requirement such as performance, and memory usage.

(9) Future reference: Questions that users ask—and normally answered themselves—that might not exist on the channel, but that are interesting enough to create a thread as a future reference.

(10) Other: Questions that ask for other assistance (i.e., questions not related to the channel) or the message contains unrelated information (e.g., announcements, ideas for improvement).

Table 4.1 shows examples from each channel for every type of knowledge shared through questions.

Table 4.1: Examples of questions from both channels by type of knowledge.

Type of Knowledge	Stack Overflow	R-help

How-to	*"...Does anyone know a way of subsetting those 3 months of the time series?..."* URL: Q6356829[1]	*"...but I can't figure out how to resize each panel along the y axis and show only categories that have corresponding x values in each panel..."* URL: QQat1yH[2]
Bug / Error / Exception	*"I'm getting a weird error when training a glmnet regression..."* URL: Q1712316	*"Gives the error messages: Error in coxme.varcheck..."* URL: QKkYBe6
Discrepancy	*"...But for some reason, a lot of lines get merged – e.g., in row 500 of my data frame, I'll see something like..."* URL: Q1407647	*"When I use wilcox.test, I get vastly different p-values than the problems from Statistics textbooks"* URL: QnXVLyD
Set-up	*"When running Sweave from emacs-ess, errors are provided with a code chunk number. Is there an easy way to navigate among the code chunks by number?..."* URL: Q4501404	*"Hi, exist any way to create a windows installable package from a Linux R installation"* URL: QcOHVdp
Decision help	*"I have been asked to change a software that currently exports .Rdata files so that it exports in a 'platform independent binary format' such as HDF5 or netCDF..."* URL: Q7838027	*"I have two time series. Both measure the same thing and I would like to determine which one is noisier..."* URL: QytDnBU
Conceptual / Guidance	*"What are some practices I can adopt so that my code will always be a pleasure to work with? I'm thinking about things like"* URL: Q1266279	*"I would like to understand what are the units defined on the y-axis when you plot the one-dimensional predictions (histograms) from lda() (MASS) discriminant function objects?..."* URL: QkaP4Up
Review	*"I'm trying to get all five outputs from the 5 data frames within the list x at the same time, but I am stuck here..."* URL: Q17998174	*"...ghyp package and simulated series of t-distributed variables when suddenly i was not able to reproduce the log likelihood values reported by the package..."* URL: QH8GFiU

[1]URL transformation for StackOverflow: QXXXXXXXX where XXXXXXXX is the id of the question, such that http://stackoverflow.com/questions/XXXXXXXX/

[2]URL transformation for R-help mailing list: QYYYYYY where YYYYYY is the URL shortened id, such that http://goo.gl/YYYYYY

Non-functional	*"The best implementation I could come up with uses a nested-loop, which I realize is probably the least efficient,...There must be a more efficient way of doing this?"* URL: Q1510039	*"...is there a better or more efficient way of doing this, maybe with apply or something..."* URL: QdjtmdC
Future reference	*"...I know the answer and I already tend to avoid sapply. I just wish there was a nice answer here on SO so I can point my coworkers to it. Please, no "read the manual" answer..."* URL: Q12339650	*"I've just posted a demo made with the rgl package to Youtube, visible here: [link] For future reference, here are the steps I used: 1. Design a shape to be displayed, "* URL: QHnUvMB
Other	*"I would like to learn some SAS because I am interested in a few industries that tend to use it exclusively."* URL: Q501917	*"...SolutionMetrics is presenting R and S+ courses in Brisbane, Melbourne & Sydney - August & September, 2013 ..."* URL: QZ5PVl2

Answers

Answers represent solutions to questions. my colleague and I found nine types of knowledge related to *answers*:

(1) Redirecting: The user provides a link to an existing solution that is not in the thread (e.g., external application, tutorial, or project).

(2) Tutorial: The user answers the question with a set of steps in order to teach people how to solve the issue.

(3) Source code: The user provides a chunk of source code as the solution without an extensive explanation about the answer.

(4) Clue/Suggestion/Hint: The user provides possible ways to solve the issue without solving it.

(5) Alternative: The user provides a different approach to a solution that is related to but not exactly what the user is asking for (e.g., mathematical approaches, data structure modification).

(6) Explanation: The user provides a solution that explains the approach to answer the question and lists steps on how to do it.

(7) Announcement: The user provides a notification about something (e.g., packages, libraries).

(8) Benchmark: The user provides a benchmark of multiple solutions posted by authors of the answer or compares answers on the thread.

(9) Opinion: The user provides their own opinion or expands other answers by adding scenarios and examples. On Stack Overflow there is a check mark element that represent the acceptance of the answer (see figure 2.2).

Table 4.2 shows examples from each channel for every type of knowledge shared through answers.

Table 4.2: Examples of answers from both channels by type of knowledge.

Type of Knowledge	**Stack Overflow**	**R-help**
Redirecting	*"What about [Rattle]?"* URL: Q1386767	*"There's also the work of a former PhD student in our Dept: [here]"* URL: Qbv8QCL
Tutorial	*"The difference between the two calls is small, but it can have important consequences. Especially if you write production code and/or are concerned with correctness in your research, it's best to avoid unnecessary repetition of variable names"* URL: Q1296646	*"The quick answer is that in the ANOVA situation where you are interpreting individual level parameters, you are testing for the difference of a particular group from a shared mean (the intercept) across all three groups, whereas with the t-test you are only considering two groups at a time... "* URL: QN2pXTv
Source code	*"How about: [source code] which yields: [output]"* URL: Q2391364	*"...I think this comes close to what you want (escaping manual work). [source code]"* URL: QUl9rOJ
Clue / Suggestion / Hint	*"Without knowing the particulars of this packages, John Chambers "Software for Data Analsys" (2008, Springer) has a good discussion on debugging, for example via..."* URL: Q1712316	*"GWAF uses the kinship package. The documentation is pretty good for it, and I've used it successfully. It may be helpful to get that working before trying automate some tasks using GWAF."* URL: QIm8MRf

Alternative	*"And, in case you were dealing with an estimated quantity, plotmathgrDevices also offers the possibility of adding a hat to your greek letter..."* URL: Q6044800	*"I have coded up the algorithm from the Cameron and Turner paper... It is not designed to work with actual "streaming" data — I don't know how to do that"* URL: QKlLXrY
Explanation	*"Trying to shoehorn the data into a data frame seems hackish to me. Far better to consider each row as an individual object, then think of the dataset as an array of these objects..."* URL: Q2321786	*"First define a function from those points:... and now you can apply integrate() or trapz()... trapz()..."* URL: QwFq3RY
Announcement	*"...I recently added sort.data.frame to a CRAN package... If you are one of the original authors of this function, please contact me..."* URL: Q1296646	*"...SolutionMetrics is presenting R and S+ courses in Brisbane, Melbourne & Sydney - August & September, 2013..."* URL: Qrj6jdL
Benchmark	*"...Benchmarks: Note that I loaded each package in a new R session since... dd[with(dd, order(-z, b)),] 778..."* URL: Q1296646	*"...the test of system.time is : [answer 1] [time] [answer 2] [time]..."* URL: QYeOygd
Opinion	*"Agreed that Sweave is the way to go, with xtable for generating LaTeX tables..."* URL: Q1429907	*"I don't think we (the R foundation) will ever change away from "R"..."* URL: Q0TIukq

Updates

An update is a modification to a question or answer. On the R-help mailing list, updates are not easily identifiable as a consequence that all communications are presented as plain text emails. Therefore, I defined updates on the R-help mailing list as *emails submitted by the author of the question or answer*.

In contrast, on Stack Overflow updates are presented in multiple ways:

- **Labelled updates** are explicitly shown in the body of questions or answers next to a label that identifies the update (e.g., edit, update, and p.s.). In the case where multiple update labels appear in a message, each label is accompanied by a number (e.g., *"[Edit 1:]"* URL: Q1452235), by a date (e.g., *"Edit/Update (April 2011):"* URL: Q1452235), or by a bulleted list (e.g., "EDIT: - anova... -drop1..." URL: Q7273695)

- **Non-labelled updates** are only visually recognizable through the message history

system. There is no clear indication of the change except for a box at the end of the message that contains the user who performed the change and the date when they changed it.

Depending of the type of update (i.e., labelled or non-labelled), the usage is different. Non-labelled updates are mainly used to correct format, grammar, spelling, and semantic mistakes, or to incorporate explanations, examples, and suggestions without changing the meaning of the question or answer. Labelled updates are for everything else. Our analysis showed that there are six types of knowledge related to *updates*:

(1) Announcement: Announces specific events (e.g., bounties, future updates).

(2) Background: Adds additional context to the question or answer (e.g., what the user did previously or what the user already knows).

(3) Correction: Corrects format, grammar, spelling, and semantic mistakes.

(4) Expansion: Expands the question or answer by providing scenarios or examples.

(5) Explanation/Clarification: 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.

(6) Solution: The user answers their own question.

Table 4.3 shows examples from each channel for every type of knowledge shared through updates.

Table 4.3: Examples of updates from both channels by type of knowledge.

Type of Knowledge	Stack Overflow	R-help
Announcement	*"Update: Added bounty. Interested to know differences ..."* URL: Q1395102	*"That works as well. I'll collate your response and a couple of others, and post tomorrow."* URL: QV92PUr
Background	*"[Edit 1:]: To be clear, I know that there are C and C++ (and Java, Python, etc.) interfaces to R (rJava, rcpp, Rpy, etc.)..."* URL: Q1452235	*"...about 4 years ago, I asked for speedier alternatives to lm (and you helped me on this one, too), and then checked into the speed/accuracy..."* URL: QHlYXB9
Correction	*"...www.ptechnologies.org..."* → *"...[www.ptechnologies.org][1]..."* URL: Q5706756	*"correcting a typo (400 MB, not GB)..."* URL: Qc9XR11

Expansion	*"edit: combined with plyr, this becomes not bad at all..."* URL: Q7290947	*"Clarified summary of problem: I have an excel spreadsheet has a row for every student registered at...."* URL: QgYc0oz
Explanation / Clarification	*"EDIT: the original file doesn't contain progressive numbers, so this is not a solution..."* URL: Q1874443	*"Changing from ts.union() to ts.intersect() did not make a difference. So, I went back to bare metal by using awk to generate two-column text files..."* URL: Q57hxb8
Solution	*"PROBLEM SOLVED: The problem was that "trialid," a factor variable, had levels that did not include "1"... "* URL: Q15563589	*"Two solutions proposed – not entirely orthogonal, but both do the trick. Instead of nesting cbin in a loop (as I did originally – OP, below)..."* URL: Q7jnLEh

Flags

On Stack Overflow, a flag is a mechanism to get a moderator's attention. Flags are short announcements shown below a question, containing information about the alias of the user who flagged the question and the reason why it was flagged—the reason is shown on the table 4.4 as a subtype. Flags can accomplish a variety of objectives: they mark spam messages, rude or abusive behaviour; they identify messages that do not answer the question, duplicate questions, off-topic messages, unclear questions, questions that are too broad, primarily opinion-based questions, and low-quality answers. Depending of the type of the flag threads can be closed (e.g., duplicate question, and off-topic), or users can lose reputation point (i.e., rude or abusive behaviour). Figure 4.1 depicts an example on a flagged message on Stack Overflow.

On the R-help mailing list, the flag concept does not exist. However, based on the definition of flags on Stack Overflow, I defined flags as messages used to call the attention of other community members. Based on the previous definition, flags on the R-help mailing list are used to keep a healthy community, promote discussion, and call the attention of community members to certain issues. In contrast with Stack Overflow, flags might be used by the person who asked or answered a question. Due to the plain text format of messages on the R-help mailing list, flags can be mixed among the text of answers, comments, questions and updates. Flags in the R-help mailing list do not constrain users to answer questions, or to clarify what it is already asked.

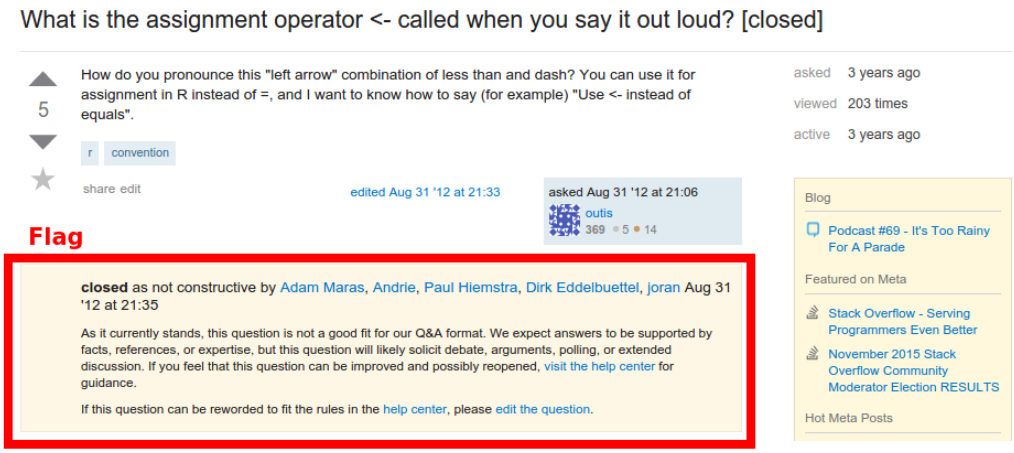


Figure 4.1: Flagged post on Stack Overflow

Our analysis showed that there are six types of knowledge related to *flags*:

(1) **"Off-topic"**, **"Not constructive"** or **"opinion-based"**: They are used to identify questions that are not related to the list's interests or which answers are based on the opinion of channel participants (not actual facts). These flags can be assigned for different reasons or they can be used on different ways.

- *Typographical error*: The problem was caused by typographical error.
- *Debugging help*: User is asking help for debugging.
- *Book, tool, software library*: User is asking for a book, tool, software library, tutorial or other off-site resource.
- *Minimal understanding of the problem*: User did not demonstrate a minimal understanding of the problem being solved.
- *Insufficient information*: the question lacks sufficient information to diagnose the problem
- *Extra information*: User provide extra information that is not related with the questions but that it might be interesting for the user who asked the question.
- *Homework*: When users ask something that looks like an assignment.

(2) **"Not an answer"**: It is used to emphasize *alternative answers* that are outside of the scope of the question, or to identify that a specific solution that does not answer the question.

(3) The **"Repeated question"** flag is used to acknowledge that the user is asking a repeated question. For example, because they could not find a proper answer that fit their needs, or because people did not search the archives to see if the question had already been asked.

(4) The **"Too localized"** flag is used for questions that are too specific and do not help any future reader.

(5) **"Unclear"** and **"Not a real question"** flags are for questions that are difficult to understand.

Table 4.4 shows examples from each channel for every type of knowledge shared through flags. Some flags and sub-categories are specifically of one of the channel, to the best of our knowledge, there is no example available. It is worth mentioning that there are *temporary flags*, and therefore, flags in one channel might not longer exist in our data. For instance, in Stack Overflow an *offensive* flag expires after 48 hours, which explains why there are no examples under the type *rude or abusive behaviour* in Table 4.4.

Table 4.4: Examples of flags from both channels by type of knowledge.

Type	**Stack Overflow**
Too broad	*"What tricks or functions do you use?"* URL: Q32888757
Duplicate	*"Possible duplicate of Reading multiple files into R - Best practice"* URL: Q32900021
Unclear	**Not a real question**: *"Is there any R package (or C++) that has sieve bootstrap? (The bootstrap is a method for estimating the distribution of an estimator or test statistic by resampling one's data or a model estimated from the data..."* URL: Q12207195 **Unclear**: *"I have a text file in the following format and I wish to extract certain lines using R. read.table("") also does not seem to work [code]"* URL: Q12070554

Off-topic	**Typographical error**: *"Missing closing quotes after 9 in states"* URL: Q32918405 **Debugging help**: *"But this code is not working. Any hints would be appreciated."* URL: Q32903807 **Book, tool, software library**: *"I want to view the R source code with any comments included, to see how the author of the package is running his code..."* URL: Q18005488 **Not constructive**: *"How do you pronounce this "left arrow" combination of less than and dash?..."* URL: Q12222481 **Minimal understanding of the problem**: *"It may sound trivial in this example, but it's not in my sample. for the computation of the means, sds ect..."* URL: Q18028285 **Insufficient information**: *"I'm a beginner in r programming and I'm trying to sample 25 cells all of them separated by a minimum distance..."* URL: Q18067248
Repeated question	*"This question has been asked before and already has an answer. If those answers do not fully address your question..."* URL: Q15301476
Too localized	*"I have a folder of 100 CSV files. Each CSV file has same variables over same time period. My instructor has asked me to write a R script that will allow me to read each file to a separate Dataframe, and then call each Dataframe in R console, to get the summary statistics of the data..."* URL: Q14253317

Type	**R-help**
Rude or abusive behaviour	*"...next time you decide to answer my question with "RTFM", please also include the number of the page that answers my specific question."* URL: Q1MupSk
Not an answer	**Alternative solution**: *"This is not an answer to your question, but I have used SparseM package to represent large travel time matrices efficiently... if the traveltime matrix is symmetric."* URL: QnnUxbM; **Not an answer**: *"Thank you very much for pointing me to meta-analysis. Although it may not solve my problem with the normalization"* URL: QxfTDgx
Too broad	*"Your request seems too broad to allow a more focused response. Perhaps we could be more helpful if you told us what you are trying to accomplish."* URL: QkDYuwL
SPAM	*"SPAM?"* URL: QUHjz4f

Off-topic	**Off-topic** *"The main questions here are not R-related, but statistical modeling questions, and much too broad for the R list."* URL: Q3AGaEi **Extra information**: *"N.B. Off topic. This is an incredibly nice feature of R. SAS overparameterizes the design matrix and employs the sweep algorithm to zero out redundant parameters."* URL: QIpGVMq **Homework**: *"This is not a homework help-line..."* URL: QKMYykN
Repeated question	*"I'm very sorry for my repeated question, which i asked 2 weeks ago, namely:"* URL: QGbek3R

Comments

Stack Overflow defines comments[3] as *"...temporary 'Post-It' notes left on a question or answer.."*. Comments can be visualized in a specific section below each question or answer. On Stack Overflow, comments can be used as a follow-up to questions, to answer a question, or to clarify a question.

On the R-help mailing list, I defined comments as messages written to *improve an answer in response to an incomplete question, or as follow-up on a discussion*. Most importantly, messages should be written by a *different person than the author of the question or answer* to which they are responding. The difference between an update and a comment on the R-help mailing list is the motivation of the user who wrote the message and the author of the message. For example, if user A posts a question, B asks A to clarify something about the question, and A answers back to B, then the last message is an update and the message sent by B to A is a comment (see Figure 4.2 top). Similarly, if user A posts a question, B answers A's question, A asks B to clarify something about the answer, and B answers back to A, then the second message is an answer, the third message is a comment, and the last message is an update (see Figure 4.2 bottom).

Our analysis showed that there are five types of knowledge related to *comments*:

(1) Clarification/Related question: Provides (or requests) additional information about a question or answer.

(2) Expansion: Provides additional information.

(3) Correction/Solution/alternative: Suggests a change to a question or an answer, offers an alternative solution or a correction.

[3] `http://stackoverflow.com/help/privileges/comment`

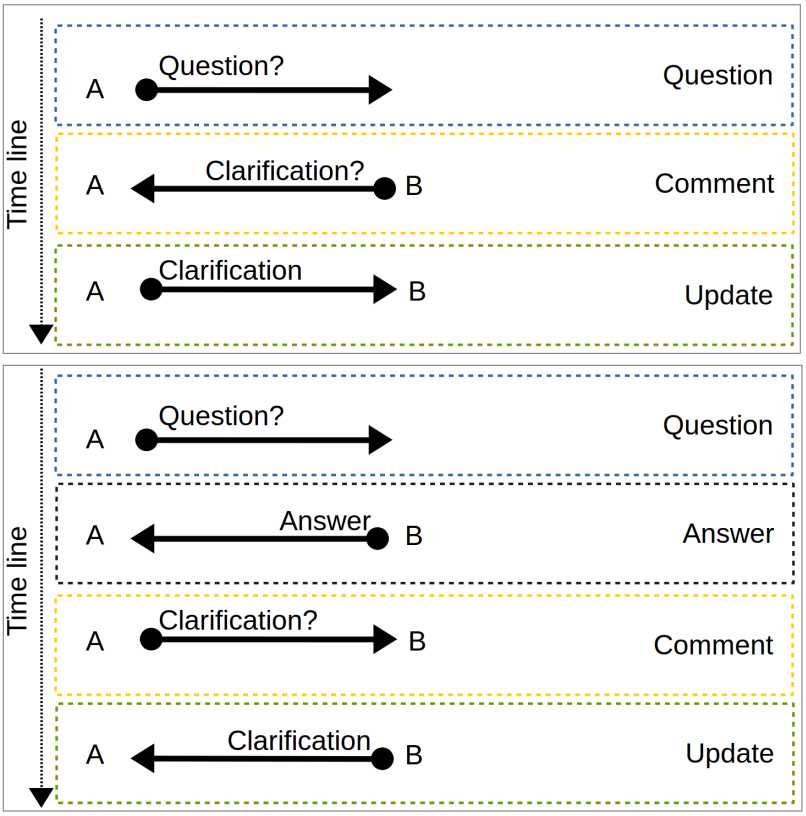


Figure 4.2: The arrows represent a message sent to the mailing list, and the labels specify the motivation behind the message. *Example 1 (Top)*: a user A posts a question; later, B asks to A to clarify something about the question; and A answers back to B. *Example 2 (bottom)*: a user A posts a questions; later, B answers A's question; A asks to B to clarify something about the answer; and B answers back to A.

(4) Compliment/confirmation/Criticism: Posts something good, offers thanks, provides an opinion or criticise someone.

(5) External reference: References an external resource.

Table 4.5 shows examples of comments posted on each channel.

Table 4.5: Examples of comments from both channels by type.

Type	Stack Overflow	R-help
Clarification / Related question	*"I'm not sure what that means... > _ >"* URL: Q1497539 or *"... are you aware of any coding style guidelines similar to the Python PEP?..."* URL: Q1266279	*"What "initial result"?"* URL: QyFlTUV or *"...I will actually be installing on a VM on top of an Intel box. Does that change things?..."* URL: Q5ksQUp
Expansion	*"Character vectors are supported in HDF5"* URL: Q7838027	*"Just to follow up on Ted's comments.... for other points in polygon algorithms. Eric Haines has made some performance evaluation of the methods... "* URL: QkHlQa6
Correction / Alternative / Solution	*"...2 simplifications: since you already are using within, there's no need to use theTable$Position, and you could just do sort(-table(...)) for decreasing order."* URL: Q5208679	*"> You seem confused. Not particularly, but he needs to be aware of _which_ shell R is executing in system() calls. These things work for me:"* URL: Qbdbs2x
Compliment / confirmation / Criticism	*"Agreed, I'm making substantial use of 'sqldf'..."* URL: Q11784115 or *"+1 because those packages are great for analyzing dataframes..."* URL: Q3375808	*"Thanks ray, I really appreciate your concern..."* URL: QkHlQa6
External reference	*"See also [here] and [here]. The R Inferno is also another great read."* URL: Q9508518	*"I think this paper elucidates the problem Bert mentioned... paper..."* URL: QkHlQa6

The format of the message in Stack Overflow and the R-help mailing list, permits participants to ask multiple questions in the same thread. Therefore, the categorization of knowledge presented in this chapter is non-mutually exclusive.

4.2 RQ-2. How is knowledge constructed on Stack Overflow and the R-help mailing list?

Through our analysis of Stack Overflow and the R-help mailing list threads, I identified two different approaches to constructing knowledge:

- **Participatory knowledge construction** refers to answers that are created based on the cooperation of multiple users in the same thread. Participants complement each other's questions by providing pros and cons about the answer, different viewpoints, or additional context and examples. Participatory knowledge is comparable with the team concept in which people work together in a cooperative way for the same objective.

- **Crowd knowledge construction** leverages the experiences of many users; it allows them to contribute their individual explanations and practices adding variety to the pool of solutions. Crowd knowledge is comparable with the group concept in which people work towards the same objective but not necessarily together. Participant can vote over others ideas, but the idea is not constructed through a discussion process.

On the R-help mailing list, participatory knowledge takes place when: (1) previous answers are included in the actual answer and it is possible to infer a link between them; or (2) when a direct reference to other answers or authors is expressed in the message. Figure 4.3 depicts two examples of the way participatory knowledge occurs on the R-help mailing list: direct citation of the author of a previous answer (top), and inferable links between answers (right).

On Stack Overflow, participatory knowledge takes place when: (1) it is possible to infer a link between answers (i.e., direct or indirect reference); or (2) comments complement the answer, or directly cite another author.

On Stack Overflow, participatory knowledge happens in different places, perhaps as a consequence of its rich interface. We see this, when a user answers a question and directly cites or links to the author of another answer in the thread, or when a user cites the author of a question or answer in a comment made on that question or answer, by including more information or by suggesting that the answer provided is an additional solution. Figure 4.4 depicts four examples where participatory knowledge occurs in Stack Overflow: one answer references the author of another answer (top left); the comments expand the answer

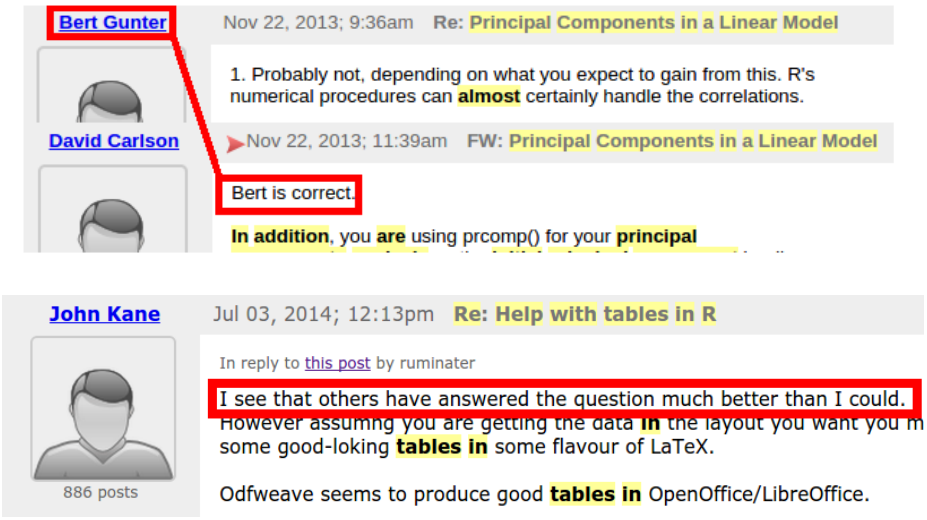


Figure 4.3: Participatory knowledge on the R-help mailing list.

(top right); the comments reference another answer for a specific scenario (bottom left); and an inferable link between the actual answer and a previous one (bottom right).

Crowd knowledge construction is observable when: 1. There is not a direct or inferable reference between answers, 2. Answers are a variation of one of the answers on the thread. Figure 4.5 depict an example of how crowd knowledge construction is visible on Stack Overflow. As can be seen from the figure, there are two of the three answers that provided a repeated solution.

4.3 RQ-3. How does the sharing of links on Stack Overflow and the R-help mailing list support knowledge construction?

External resources contain information that is valuable for questions and answers. More importantly, they contribute to the construction of knowledge, often by simplifying answers and other contributions. However, depending on the type of resource (e.g., forum, Q&A Website, or Wiki), the support might be different. We found that each type of resource is shared and supports knowledge on Stack Overflow and the R-help mailing list in the same way.

Our analysis showed that there are 11 resource types:

(1) Q&A channels: A online media channels for asking questions and answers (e.g., Stack

Figure 4.4: Participatory knowledge on Stack Overflow.

Overflow and the R-help mailing list).

(2) Source code management systems, project hosting or issue trackers: Online services that provided support for management of changes to documents (e.g., GitHub),

Figure 4.5: Examples of how crowd knowledge construction occurs.

hosting of projects (e.g., SourceForge or Google Code), and management of the software issues (e.g., Bugzilla).

(3) File hosting: Online services that provide support for management of files (e.g., MediaFire and Dropbox).

(4) Digital Libraries, papers, books and journals: Online services that provide access to papers, books, journals (e.g., R Journal arXiv).

(5) Forums: Online discussion website where people can hold conversations in the form of posted messages (e.g., Google Groups).

(6) Wiki: Online website that allows collaborative modification of its content through the web browser (e.g., Wikipedia).

(7) Blog: An informal website to published discussion, ideas, and tutorial (e.g., Wordpress).

(8) Official documentation: Documentation published by the original source of the technology (e.g., CRAN)

(9) Libraries, packages and applications: collection of non-volatile source code or resources used by programs (e.g., GGPLOT) or entire software solutions (e.g., RSTudio)

(10) Online environments: website that provide online environments such as IDE or notepads with source code syntax highlighting (e.g., GitHub Gits, or Pastebin)

(11) Other: Websites from companies, non- official documentation sites, or any other online services do not include on the previous resources list.

Additionally, we identified 10 types of resources that are linked to support knowledge construction:

(1) Answers: Links to an answer on a Q&A media channel.

(2) Tutorial/Guide: Links to tutorial or guides.

(3) Source code/Examples: Links to source code example, implementation, or final products.

(4) Channel: Link to a an specific media channel.

(5) Expand/Background: Links to a source of information that should be read.

(6) Books and papers: Links to books, papers, and journals

(7) Hacks: Links to workarounds.

(8) Images: Links to images.

(9) Data: Links to data.

(10) Announcement: Links to company or personal webpage or project.

Table 4.6 summarizes the types of external resources my colleague and I found in our study and the ways these resources support the construction of knowledge.

Table 4.6: External resources and the construction of knowledge.

Resource Type	Ways to Support Knowledge
Q&A channels	**Answer**: *"...Here's the salient Python code you can retrofit into your calls in rJython, borrowing directly from the aforementioned blog post: [code]..."* URL: Q10027560 **Tutorial / Guide**: *"As a end note, I read How to make a great R reproducible example? and tried my best"* URL: Q5963269 **Channel**: *"...If you have questions about xcode, ..., then a better place might be the r-sig-mac mailing list: [here]..."* URL: QYBySST9 **Source code/Examples**: *"The following function (found here) works well for messages containing ASCII characters..."* URL: Q10027560

Source code management systems, project hosting or issue trackers	**Answer**: *"Look into Sweave or knitr to combine R and LaTeX for producing reports. hth, Ingmar [Sweave] [knitr]"* URL: QvJEJlU **Expand / Background**: *"All I can suggest in this case is to update R and run update.packages() as indicated by FAQ1: [here]"* URL: Qd2rylY **Tutorial/Guide**: *"...This way is taken from this rCharts 'official' tutorial (find m1 plot code)..."* URL: Q19339766 **Books and papers**: *"If you are still unable to find it, please go to: [LaTeX document on GitHub] It is a compiled version of the manual. "* URL: Qq6b1Rr **Hacks**: *"...As per mentioned by @GeorgeDontas in the comments, there is a little hack [Issue on GitHub] that could allow to change the labels of the x axis to dates instead of 'w.01, w.02'..."* URL: QLJzWS6
File hosting	**Images**: *"I'm trying to plot some data over a background image. The problem is that both layers end up using the same scale. This is unfortunately problematic. An example. I want to plot some data over this image ... Plotting the sample data, I get this..."* URL: Q16409935 **Source code / Examples**: *"...I am doing sentiment analysis using score.sentiment... from the below link you can see how this work... [r-file on dropbox]..."* URL: Q9g8UPi **Data**: *"I have weekly sales observations for several products drawn via ODBC. Source data is available at [here] "* URL: QtMpK31
Digital Libraries, papers, books and journals	**Tutorial / Guide**: *"You can reduce the text size using the "grid.gedit" approach described at the end of the ggplot book, available on Hadley Wickham's website: [ggplot book]"* URL: Qxho3GJ **Source code / Examples**: *"...Meanwhile, I would like to point out that any of these linear restrictions can be reformulated as an exclusion restriction, see for example the remark on pages 5-6 of [Gutenbrunner et al. paper] or the references cited there. And using this sort of parametrization you can use anova.rq()..."* URL: QZ7OYEm **Expand/Background**: *"...I started by reading through [Ricci distribution manual]... "* URL: QM21jAN or **Expand**: *"I suspect you are trying to find your way into Circle 6 of 'The R Inferno' but haven't yet got in. [R Inferno]"* URL: QxF1j3R

Forums	**Expand / Background**: *"...to include the raw HTML in my page but I receive a 404 error with these as well as when I attempted to include it as an iframe within R-Shiny (following this [Discussion Google Groups])..."* URL: Q23618813 **Tutorial / Guide**: *"Success. For myself and FWIW to other useR's here's how i spent the sunny half of my sunday to achieve it :/... ## Per-[ubuntuforums]"* URL: Q7cpqVb **Source code / Examples**: *"...this might be what you need (the following example is taken from [lispforum])..."* URL: Q3300900 **Channel**: *"There is a separate ggplot2 mailing list at [ggplot forum], please post future ggplot2 questions there."* URL: Q0c7ypZ **Answer**: *"...If you're looking to improve the approach itself, [see here]..."* URL: Q3137168
Wiki	**Expand / Background**: *"...how to properly [send Unicode email], which you're using Python to manage the underling SMTP connection for..."* URL: Q10027560
Blog	**Expand / Background**: *"...Slightly on topic, just yesterday, the CFPB announced they'll be using R in their work: [Instrumental Variables]..."* URL: QkvyDvZ **Tutorial / Guide**: *"Saving full environments is possible, but it is very easy to start loosing track on where each variable came from. You might want to use this process: [here]"* URL: QNWMm2w **Answer**: *"you can see the whole algorithm of sentiment analysis from the below link [here]"* URL: Q9g8UPi
Official documentation	**Expand / Background**: *"...You should be able to open a text connection using ?file with the open argument set to write..."* URL: Q19601034 **Answer**: *"However, you may be experiencing a problem in your PDF viewer (Preview?) due to anti-aliasing, which is noted here: [link]"* URL: QbESGwE **Tutorial/Guide**: *"...I am trying to grok the information [here], where it says..."* URL: Ql2EnfL
Libraries, packages and applications	**Answer**: *"RStudio IDE (Server) may be the answer to your question. Have a look at [RStudio]"* URL: Q13705519
Online environments	**Source code / Examples**: *"Here is my version of the function... It is available here as a gist..."* URL: Q10454973
Other	**Announcement**: *"...It looks like it is what powers the awesome [rdocumentation.org] site..."* URL: Q13705519 **Source code / Examples**: *"National Weather Service to generate graphics representing real-time hydrologic ensemble (probabilistic) forecasts. Go to: [here] to see. "* URL: QkvyDvZ

4.4 RQ-4. Why do Certain Users Post to Both Stack Overflow and the R-help Mailing List

For a community such as R, the variety of channels makes it possible to reach different users of the same community. Therefore, it is possible to find that a question is posted in both channels by the same user. I was motivated to understand the reasons, as well as the benefits or disadvantages of posting the same questions in both channels at the same time.

I identified that being active on both channels at the same time brings some benefits:

- **Improve the existing answer:** If a question hasn't been answered in one channel, an answer may exist in another channel.
- **Support follow-up questions:** If a question is already closed and further questions arise, users can use other channels to reach out for help.
- **Speeds up answers:** Users can ask the same question on both channels to speed up the process, and at the same time, get more points of view. However, this behaviour is not encouraged by the community as it is deemed impolite.

Table 4.7 provide examples of the benefits of using both channels.

Table 4.7: Examples of the benefits of using both channels.

Type	Description
Improve the existing answer	**Cross-posting warning**: A user asks a question on Stack Overflow (URL: Q17133550). A few days later, the user asks the same question on the R-help mailing list (URL: QhPZmXo) warning readers that she posted the same question on Stack Overflow without any successful answer. Years later, the question is answered on Stack Overflow.
Support follow-up questions	A user posts a question on the R-help mailing list URL: QQaUoJ3 and gets some answers. However, the follow-up question is not answered on the R-help list, so the user asks the extra questions on Stack Overflow. URL: Q12156939

Speed up answer	**Cross-posting without warning (no one noticed the cross-posting)**: A user sent an email to the R-help mailing list (URL: QZCCLj7), and a few minutes later, asked the same question on Stack Overflow (URL: Q12156939). On both channels, the community answered the question. Next, the user asked a follow-up question on just the R-help mailing list, which was answered satisfactorily. **Cross-posting without warning (someone noticed the cross-posting)**: A user asks a question on Stack Overflow, Cross Validate and the R-help mailing list. Someone notices it and posts back to the R-help list saying *"Crossposting to CrossValidated and StackOverflow and to Rhelp is deprecated. You should offer code and data and explain why the answers you have already been given are not adequate."* URL: QxW5gdw

4.5 User Behaviours

While analysing questions and answers, my colleague and I identified user behaviours that are not reflected in the categories we developed, but that I believe are worth mentioning. These behaviours provide evidence of their altruistic way of thinking and the strong commitment that users have within the community.

- **I answered my own question**: Some questions are answered by the same user who asked the question. They posted back to the channel to document their solution. (e.g., *"I've discovered the answer to my own question."* URL: Q18450396 or *"Just for the records (and if anyone ever wants to find the "solution"), I solved my own problem."* URL: Qr3z0DX).

- **I did it for you**: There are occasions when answering authors provides an extensive amount of source code to help others. For instance, *"I have coded up the algorithm from the Cameron and Turner paper. Dunno if it gives exactly the same results as my (Splus?) code from lo these many years ago..."* URL: QGXWGG3.

- **Answered, updated or continued years later**: Some answers are provided months or years after the question was asked. For instance, a user on Stack Overflow modified an answer to provide a more updated version of the source code (i.e., URL: Q1724024); and a question asked on the R-help mailing list in 2012 was continued two years later (i.e., URL: QkgSHZv).

- **Ideas for improvement or creation of the channel**: This behaviour is specific for the R-help mailing list. Sometimes users suggest modifications or new features to improve the channel. For instance a user proposes to create a package repository that can be accessible through a public wiki, or version control interface (i.e., URL: Qp0IunD).

I also identified two behaviours that might result in a bad response from the community:

- **Cross-posting:** The user posts the same question in both channels at the same time. For instance: *"-1 for cross posting to r-help – [user name]"* URL: Q5436630
- **Posting guidelines violation:** The user behaves in such a way that it becomes apparent that they did not read the posting guidelines. For instance, a user asked a question that seems to be the opposite of what the posting guide recommend, and someone answered: *"...If you read the Posting Guide I think you will find precisely the opposite expectation explicitly presented. Using my "cheeky code" would only be part of the recommended actions to take before posting if you follow the recommendations of the "Do your homework before posting:"..."* URL: QFUm1HC

4.6 Survey Results

As previously mentioned, the objective of the survey was to bring further insights on the study. In the survey, the main findings aligned with the thesis results come from the open questions that gather information about the experiences of the participants on Stack Overflow and the R-help mailing list—the complete version of the results is in Appendix A. Given the nature of my study, I was interested in the opinions regarding the preferences of one channel over the other, as well as any challenges that the participants might have experienced. Tables 4.8 and 4.9 summarize the results of the following open questions (scattered among sections 2 to 4 of the survey) in terms of the pros and cons of both media channels.

- *Have you experienced any challenges using Stack Overflow? Please elaborate.*
- *What motivates you to answer questions or add comments on Stack Overflow? Please elaborate.*
- *Have you experienced any challenges using the R-Help Mailing List? Please elaborate.*

- *What motivates you to answer questions on the R-Help Mailing List? Please elaborate.*
- *Why do you think the R-Help Mailing List has been replaced by Stack Overflow? Please elaborate.*
- *In what situations would you choose Stack Overflow over the R-Help Mailing List? Please elaborate.*
- *In what situations would you choose the R-Help Mailing List over Stack Overflow? Please elaborate.*

These questions were intended to provide insights to understand why these channels are used, as well as some opportunities for improvement. Table 4.8 refers to the pros and cons of Stack Overflow. The reported benefits of using Stack Overflow were as follows: peer recognition, a friendly and rich interface, answers are straight to the point, it is easy to search for information, and questions are answered faster. However, the drawbacks of Stack Overflow include: strict rules become an obstacle sometimes and add complexity, the abundance of related questions, certain level of experience is required to understand the answers, and limitations on the scope of topics to discuss.

Similarly, Table 4.9 refers to the pros and cons of the R-help mailing list. The benefits of using the R-help list include: the convenience of just handling email, the information can be used to learn, the answers are more focused, the participation of highly experienced users (i.e., rock stars), and its flexibility. The disadvantages that were reported include: sometimes there can be aggressive behaviour, performing search is not easy, email is not a desirable interface, and the lack of categorization given the massive volume of information.

Table 4.8: Summary of pros and cons for Stack Overflow. The numbers between square brackets correspond to how many users support the same topic (*) *UX* is the participants ID for the survey where X the participant ID

Pros
Peer recognition: [U3, U31] *"Stack is a nice place to get peer recognition. Plus it's very nice to be able to give back to the R community."* U3 **(*)**
Interface: [U14] *"SO 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, not have 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."* U14
To the point: [U3] *"...If I actually want an answer and not a lecture."* U3
Searchability: [U6,U16] *"SO shines when searching because of tags and ratings."* U6

Response time and clarity: [U35] *"I tend to prefer Stack Overflow for posting questions. The response time is often quite fast. The question and answers are preserved cleanly for the future."* U35 **Learning**: [U7,U23] *"If it is an area, I want to increase my proficiency in I may use it as a study device..."* U7

Cons

Strict: [U4] *"I find the rules about asking confusing and very strictly monitored. I often find questions that I would love to be answered get closed for being against the rules."* U4

Complexity: [U14] *"Only learning the Stack Exchange Ethos of what good questions & answers constitute and the way the system has evolved in terms of community/tag curation regarding questions (e.g. changes to close vote reasons)"* U14

Duplicity: [U2, U13] *"Too many related questions, needs i) de-duplication ii) more maintenance to avoid duplication"* U2

Friendliness: [U3,U35] *"If I'm feeling just a little too good about myself and I want someone on the internet to tell me what a n00b I am."* U3

Participants experience: [U25, U26] *"The widely varying expertise of the contributors necessitates careful consideration of some of the answers"* U25

Ownership of the information: [U27] *"...I don't know how SO is funded, but it looks for-profit, and many R users won't support it for that reason. There's also a concern that it may shut itself down at some point, and all the postings on it will be lost. There are multiple archives of the R-help list, so that's unlikely. There can't be multiple archives of SO, because they claim copyright on the compilation."* U27

Topic limitation: [U35, U31] *"...StackOverflow has more limited range of help topics (help for code only), whereas R-help is broader (philosophy, posting announcements, etc.)."* U35 **Answers context or background**: [U7,U26] *"Some answers are so terse as to be unhelpful to a beginner. While this does not prevent my use of the service, it does make it a bit less useful to inexperienced users. R-help tends to have much more context to the answers."* U7

Table 4.9: Summary of pros and cons for the R-help mailing list. The numbers between square brackets correspond to how many users support the same topic.

Pros

Convenience: [U6] *"it's right there in my email. I read it every day. if I see something interesting sometimes I answer."* U6

Learning: [U6, U22] *"If I want to take a break and learn some R, I read R-help for pleasure."* U6

Better answers, and more focused: [U7] *"Many of the core developers and primary educators provide great answers, that are well documented on R-help. It is certainly more focused."* U7

Rock stars: [U23, U5, U29] *"Older users are probably more familiar with mailing lists and these are the same people who are most knowledgeable."* U5 and *"If I really want an answer from someone in R-core or closely related people, I would definitely choose the mailing list."* U23

Practising: [U22] *"allows me to keep up with a random flow of how other people do things."* U22

Confrontation: [U22] *"It is less confronting to help out on the mailing list."* U22

Flexibility: [U31,U35] *"StackOverflow will close threads that are only "discussions", so these must happen on R-help."* U31 or *"If my question was not 100% help-me-code-this."* U35

Cons

Friendliness: [U23, U7, U31] *"Some of the most experienced developers can be a bit harsh when a question is badly posed, or if they don't see the meaning of the question. Usually not a big problem though. I think it is the worst for those who address r-help as a company customer support."* U23
Searchability: [U2] *"Search is bad, threading as well"* U2
Interface: [U14,U37,U4] *"...email is such a poor way relative to web 2.0 to present computer/software related problems"* U14
Volume of information: [U22] *"The volume of questions and responses and lack of categorization"* U22

In the following chapter I provide selected quotes from the survey, where each participant is identified by an anonymized identifier (U#).

Chapter 5
Theory

This chapter presents a theory that encapsulates the observations and insights found during the analysis of the data, including, the comparison of the way knowledge is shared on both channels, and the recommendations of the use of Q&A media channels

5.1 Comparison of How Knowledge is Shared on Both Channels

Based on the categorization performed, both channels provide roughly the same knowledge support for questions and answers. However, there are some differences between both channels which are summarized in Table 5.1. These observations are tendencies, and they are not behaviours unique of each channel.

Table 5.1: Comparison of the way knowledge is shared on Stack Overflow and the R-help mailing list.

	Stack Overflow	**R-help**
Knowledge construction	Crowd	Participatory
Topic restriction	Topic restriction	No topic restriction
Knowledge	Curated knowledge	Knowledge development

5.1.1 Knowledge construction

Stack Overflow's gamification system encourages participation by giving points to those who participate [37]. Even when the diversity of answers provided in Stack Overflow is high, users tend to not contribute (edit or comment) in such answers; instead, some users

provide their own answers. For instance, in the Stack Overflow thread *"Resources for learning SAS if you already familiar with R"*[1], three of the six answerers referenced the same books. The gamification mechanism gives reputation to those who answer the questions, even when each extra answer might not add any new insight about how to solve a specific problem. Stack Overflow curation mechanism (i.e., vote system) provides information about the popularity of answers, but not why, or how it is better than others answers.

In contrast, the R-help mailing list tends to be more collaborative on how users construct knowledge, and discuss proposed answers. Participants of the R-help mailing list tend to provide more background to the answers as well as to explain answers of other participants. For example, the question *"Arrange elements on a matrix according to rowSums + short 'apply' Q"*[2] was posted on both Stack Overflow and R-help mailing list. Both communities answered the question using a different knowledge construction approach. On Stack Overflow, each participant provided their own solution without any evidence of collaboration between them. Whereas users on the R-help mailing list complemented each other answers by providing extra information.

The Stack Overflow's knowledge construction is not limited to crowd knowledge construction, it also presents collaborative ways to construct knowledge. However, the crowd one seems more prevalent. On the R-help mailing list happens the same as on Stack Overflow, but the other way around.

5.1.2 Topic restriction

One of the best ways in which the R-help mailing list complements Stack Overflow is on the topics that can be posted on both channels, and the format of the questions that can be answered. On the R-help mailing list, questions related to R, but not focused on software development, are not rejected by the community (see section 4.1 Flags). Also, topics that trigger a discussion, even when they are not related to software development, are welcomed in to the R-help mailing list. For instance, when users discuss the creation or improvement of the R community channels (see section 4.5); or when a question about installing R on *Linux* is asked on the R-help mailing list (like *"R on X11 under Linux"*). In contrast, on Stack Overflow, questions that trigger discussion are flagged as opinion-based, or as off-topic, and they might be closed. For example, the questions *"What's a good example of really clean and clear [R] code, for pedagogical purposes?"*[3] was closed as off-topic

[1] `http://goo.gl/Mb4Pbk`

[2] R-help URL `http://goo.gl/PGflT5`, and Stack Overflow URL `http://goo.gl/a8AES8`

[3] `http://goo.gl/9JjZW1`

because the question was not related to software development.

As explained by, one of the participants discussing on *"creating an equivalent of r-help on r.stackexchange.com?"*[4] commented:

> *"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 [moderated] them down for being off-topic and they disappear pretty quickly."*

5.1.3 Curated knowledge and knowledge development

On the R-help mailing list questions tend to have more background than on Stack Overflow. The knowledge embedded in the R-help mailing list's answers can be used to learn new procedures, as well as identify the train of thought that guided participants when forming an answer. For instance, U26 explains:

> *"Because many developers share my view that [Stack Overflow] is a very bad model, and that the pulverisation of information into answers to apparent questions, removes the value added by reading list traffic that doesn't seem directly relevant to a currently conceptualised question, but which may lead to a new conceptualisation (out of the frame thinking). [Stack Overflow] cannot do that."*

Similarly, U35 explains that it uses the R-help mailing list if the questions are not 100% *"help-me-to-code-this"*.

In contrast, Stack Overflow shines when questions have to be kept for posterity. The curation mechanism provides tools to keep the channel clean of what seems to be unnecessary information (e.g., flagging questions, deleting comments, editing messages, and demoting irrelevant answers).

> *"[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, [and] not [having] 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."* [U14]

[4] `http://goo.gl/mTccwx`

5.2 Recommendations For Using Multiple Q&A Media Channels

One of the interests in this thesis was to derive a set of recommendations for using media channels, as a mechanism to improve the benefits of their usage. Based on the analysis of the *flags* (which are often used to point out users' behaviours); rules, manuals and FAQ resources from Stack Overflow and the R-help mailing list; threads that were posted in both channels (i.e., the same question by the same user in both channels); and the answers of the survey. From this data, I can provide a set of four recommendations. Table 5.2 presents a summary of the recommendations that I construct for using multiple channels.

Table 5.2: Recommendations for using multiple channels.

Recommendation
Choose the correct channel according to the topic, type of question, and audience
Read the user manuals, channel rules, and learn the basic concept of the technology used
Choose a channel according to the user experience
Provide a background to the question

5.2.1 Choose the correct channel

As described in Chapter 2, media channels provide a list of *topics* permitted, this are available either in the description of the channel or in their limitations. The control of topics is often regulated by the community or channel's moderators. U35 explains that *"...Stack-Overflow has more limited range of help topics (help for code only), whereas R-help is broader (philosophy, posting announcements, etc.)"*. Knowing what channel is *more suitable for a specific topic can improve the response time or quality of the answer by taking advantage of the community members' knowledge*.

Additionally, *choosing the proper channel keeps the knowledge where it is most useful, thus enhancing the quality of the content of the channel*. For example, in the R-help's thread *"Bumps chart in R"*[5], an user wrote: *"(cross posting to the ggplot2 group for posterity) Here's how I'd approach it..."*, that is, cross-posting the question—previously posted and answered on the R-help list—in order to keep a record of the knowledge where it reaches more users, and where it is more useful to the community.

[5] `http://goo.gl/EJHWrs`

In some cases there are questions that should be *answered by a specific group (e.g., r-core team) regardless of the topic*. U32 stated *"If I really want an answer from someone in R-core or closely related people, I would definitely choose the mailing list"*. For example, in the R-help's thread *"Cointegration and ECM in Package {urca}"*[6], a participant asked the R-core team directly how to solve a problem: *"Dear R Core Team, I am using package {urca} to do cointegration and estimate ECM model, but I have the following two problems..."*.

In this scenario, *websites of a specific package or library might be the best method to communicate directly with the creators of that technology*. In some cases, the description of the channel or package provides the necessary information, such as the maintainers or participants (e.g., R-help primary help webpage[7], *rcpp* package[8], or *r-tag* info page on Stack Overflow[9]). Figure 5.1 depicts an example of how developers of a package can be reached using Stack Overflow (on the left) or by email (on the right).

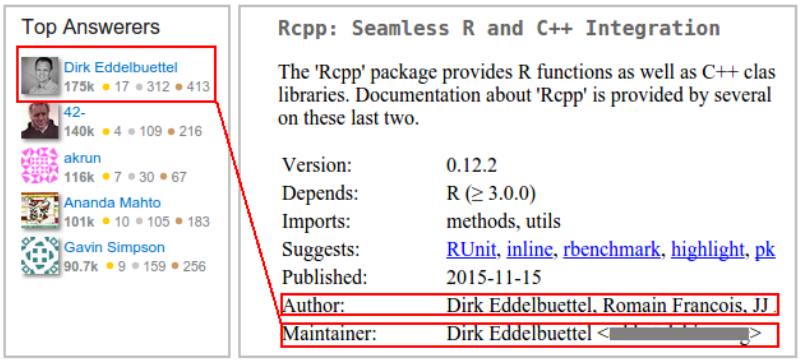


Figure 5.1: Example of how developers of the *rcpp* package can be reached. On the left, Stack Overflow, and on the right, the webiste of *rcpp*.

Finally, some channels are more suitable for certain *type or format of questions*. For example, R-help mailing list is a place for discussion, and Stack Overflow is a place for questions that have a clear answer.

[6] http://goo.gl/7olLv7
[7] https://stat.ethz.ch/mailman/listinfo/r-help
[8] https://cran.r-project.org/web/packages/Rcpp/index.html
[9] http://stackoverflow.com/tags/r/info

5.2.2 Read the user manuals, channel rules and learn the basic concept of the technology used

Through the study, I noticed that most of the harsh responses from the community were give to users who did not read the posting guide or learned the basic concept for each technology (e.g., *"An Introduction to R"*[10])—users should demonstrate a minimum understanding and use of the programming language. The community expects that if someone wants to use a channel, they should learn about it in advance, and learn the basics of the technology that they are using. For instance, in the R-help's thread *"Quantile"*[11] it is remarked the points of a guide that the user asking the question did not follow: *"...Please read the Posting Guide. It asks that you not crosspost. If you post a followup to rhelp, then the reading of the Posting guide will tell you that much more in the way of detail about your setup was requested..."*.

Depending on the channel, the amount of guide lines and posting guides available might differ. Stack Overflow provides user manuals for each of the main features of the channel such as badges, questions[12], answers[13], flags, comments, and reputation system[14]. In contrast, the R-help mailing list only has the general instructions[15] and the posting guide user manual[16], which make the R-help mailing list a more user friendly environment for new users in terms of what users have to read in advance.

Moreover, depending on the technology, there are some *community* user manuals that might be useful to read before participating in the channel. For instance, the post on Stack Overflow *"How to make a great R reproducible example?"*[17] provides tips and tricks for creating a reproducible example using the R language. Another example is the channel related user manual written by Hadley Wickham. It provides some tips for posting on the R-help mailing lists: *"...Before putting all of your code in an email, consider putting it on [GitHub Gist app]. It will give your code nice syntax highlighting, and you don't have to worry about anything getting mangled by the email system..."*

Finally, there are technology manuals like *"An Introduction to R"*), and the FAQ webpages that are available to the public—most of the time free of charge, from which any

[10] https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf
[11] http://goo.gl/Dc8gXw
[12] http://stackoverflow.com/help/how-to-ask
[13] http://stackoverflow.com/help/how-to-answer
[14] http://stackoverflow.com/help/whats-reputation
[15] https://www.r-project.org/mail.html#instructions
[16] https://www.r-project.org/posting-guide.html
[17] http://stackoverflow.com/questions/5963269/how-to-make-a-great-r-reproducible-example

user can learn the basic of each technology. For instance, the R community provides a compendium of PDF documents for new users on different languages[18]. While on Stack Overflow, supported technologies are provisioned with webpages and links to free and paid materials[19]. Members are able to reference these materials when needed, e.g., *"...You may want to acquaint yourself with the 'An Introduction to R' manual that came with your R installation to learn more about indexing."* Figure 5.2 depicts the webpage on Stack Overflow with references to free and paid materials that can be reached using the info tab (on the left), and the R project website with free user manuals for the R language (on the right).

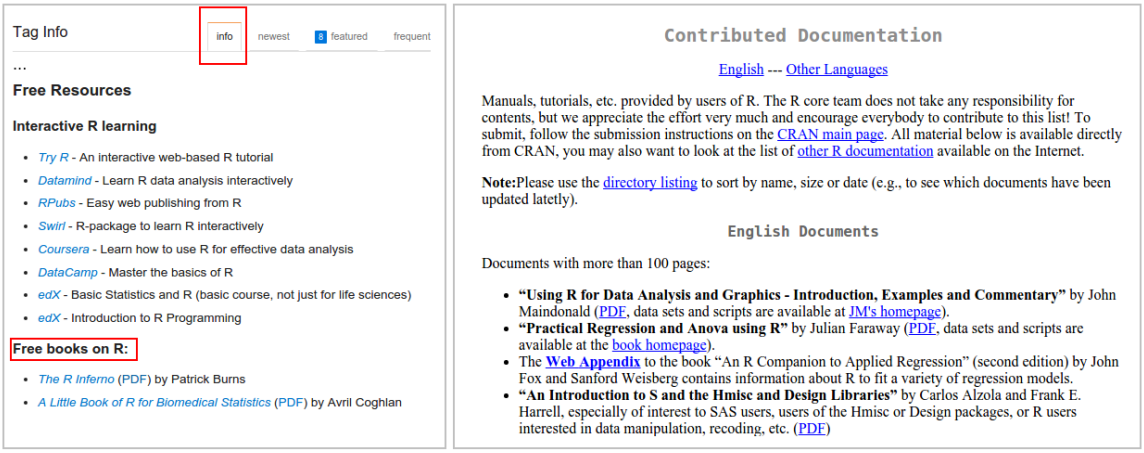


Figure 5.2: Examples of free and paid manuals available through Stack Overflow and the technology community websites

5.2.3 Choose a channel according to the user experience

Clearly, the variety of media channels can be overwhelming when having to choose one channel to post a question. Personal preferences (e.g., to use the most popular channel, or the one that answers question faster) are important elements when making a decision. It is always good to know the characteristics of the channels as a way to make a wise choice. For example, using Stack Overflow has many benefits such as low response time [28] and peer recognition [37], but there are many user manuals that should be read prior participation, and a bad reputation in the channel might affect users in real life [37]. U14 said that one of the biggest challenges of using Stack Overflow is learning the *ethos* of the channel (see table 4.8)

[18]The R Manuals are available at `https://cran.r-project.org/`
[19]Materials available at `http://stackoverflow.com/tags/r/info`

Sometimes communities have more than one channel that overlaps functionality with others. The R-help mailing list can be used for the same purpose as Stack Overflow, but it has a different audience as well as other benefits as demonstrated on this thesis. For instance, the R-help mailing list is less confronting (see table 4.9-*confrontation*); it can be used to learn rather than just get the answer (see table 4.9-*learning*, and 4.8-*To the point*); and it can be sometimes friendlier (see table 4.8-*Friendliness*).

5.2.4 Provide a Background to the Question

Sometimes, in spite of reading the documentation available, a user can fail to address the channel appropriately. The community may feel that the question asked, the information provided or something else is not in compliance with the expectations and rules of the channel (e.g., read the posting guide, demonstrate minimum knowledge about the technology used). In such cases, one should describe the documentation read, the attempts made to answer the question on their own, and what they are actually trying to achieve. This would avoid answers like *"read the manual"*, or *"read the posting guide"*, as well as helping the participants to help. As an example, in the thread *"lme4 GLMM"*[20], a user asked *"I'm very sorry for my repeated question, which I asked 2 weeks ago, namely: I'm interested in possibly simple random-part specification in the call..."*.

5.3 Recommendations For Using External Resources

When answering or asking questions, it is a common practice to provide links for sharing documentation, examples, source code, or other resources. As links point to online resources that might or not exist in the future, it is important to include the key points of the resource within the question or answer. For instance, when a question or answer contains information in an external file hosting service like Dropbox or Google Drive, the owner of the service account can break the link in any moment, leaving the message incomplete or impossible to reproduce, like the thread *"Is it possible to create a 3d contour plot without continuous data in R?"*[21]. U33 suggested: *"Questions should be self-contained as much as possible. Exceptions: recognizable links such as CRAN, R documentation, etc."*.

Based on our observations, I constructed a set of recommendations for links that are the exception of the rule:

[20] `https://goo.gl/Gbek3R`
[21] `http://goo.gl/5nanFU`

- *Well known websites*, that are expected to be maintained in the long term like Wikipedia, the official documentation in CRAN. For example, the Stack Overflow's thread *"calculating convolution of multinomial distribution"*[22] a user posted *'I'm doing a simulation where I need to calculate a [Wikipedia convolution] of [Wikipedia multinomial distributions]..."*.

- Resources that support or *expand the message*, but the important information is already explained in the message. For instance, the Stack Overflow's thread *"How do I save all the draws from a MCMC posterior distribution to a file in R"*[23] clarifies *"...You should be able to open a text connection using ?file [more information] with the open argument set to write..."*.

- The *material relevant to the message is too big*, such as papers, or demonstrations. For instance, the R-help's thread *"Using FUNCTION to create usable objects"*[24] a user stated *"I suspect you are trying to find your way into Circle 6 of 'The R Inferno' but haven't yet got in. [R Inferno]"*.

[22] http://goo.gl/OAXoCl
[23] http://goo.gl/6YfShw
[24] http://goo.gl/xF1j3R

Chapter 6
Discussion

In this chapter, I discuss the findings presented in this thesis related with the analysis of the knowledge flow through Stack Overflow and the R-help mailing list in relation with the literature. I also present the implication of the study and how it can be used for future research.

6.1 Comparison of the Way Knowledge is Shared on Both Channel

Based on my findings, Stack Overflow is more suitable for questions with a clear answer, and the R-help mailing list is more suitable to discuss topics that are in and out of the software development domain. The study by Squire [41] presents a similar difference between Stack Overflow and a mailing list of multiple projects through a different approach. By studying communities that migrated development support towards Stack Overflow, Squire found that the main reason for communities coming back to the mailing list are topic restriction and the question's format expected on Stack Overflow. Squire and I suggest that communities of practice should evaluate what are the real benefits of each channel before moving to newer technologies.

The comparison between channels presented in this thesis, as well as Vasilescu's work [53] about migration from Stack Overflow to the R-help mailing list, suggest that communities prefer to have access to a variety of answers (Stack Overflow), instead of single contextualized and discussed answers (the R-help mailing list). To support this inference I presented two ways the community of the channel constructs knowledge: crowd and participatory. The main benefit of using crowd knowledge construction is the existence of a pool of solutions, which combined with curation mechanisms, produces multiple ways to solve the

same problem (diversity of solutions) in a clear solution that can be reused.

Squire's study [41] as well as my findings suggest that Stack Overflow might not be enough to fully support software developers. As it is suggested by the amount of active users on the R-help mailing list, a community might require a place to discuss topics that are in and out of the software development domain. Additionally, the fragmentation of topics within the Stack Exchange's Q&A channels (each channel from Stack Exchange supports a small group of topics), the complex rules of their sites, and the gamification mechanism might be a difficult issue to handle for some users [51]. However, more studies are necessary in order to confirm my inferences.

Finally, this study shows that the difference between media channels might be more complex than just the differences based on channels' features. Every feature (e.g., gamification) might: affect the way in which community members work [2, 37], push away part of the community members [51], or change the way the knowledge is constructed.

6.2 Message Categorization

In this thesis my colleague and I categorised the messages and the resources that flow through Stack Overflow and the R-help mailing list. In a previous study, Treude *et al.* [48] categorized questions of different communities on Stack Overflow. The question categories **How-to**, **Bug/Error/Exception**, **Set-up**, **Decision help**, **Code reviewing**, and **Discrepancy** presented in this work, mirror the findings from Treude *et al*. The differences between my study and the study by Treude *et al.* lie on the type of messages (i.e., questions, answers, updates, flags and comments) and community that my colleague and I analysed. The question categorization presented in this work extends Treude *et al.* findings by identifying a similar set of question categories over a different community and media channel (the R-help mailing list).

I believe that my categorization can help future studies to compare the knowledge of multiple channels by providing a common ground for comparison. Furthermore, futures studies can extend our categories to include knowledge classification from other type of channels or domains. Understanding the knowledge that flows through the channels might help to improve the tools that support developers. Moreover, my knowledge categorization can be used to analyse in more detail the knowledge that flows through a channel. For instance, it can be used to analyse unanswered questions [3], successful answers [9], or identify topic trends for each categorization type.

6.3 Knowledge Construction

In this work, I present two ways to construct knowledge, the participatory and the crowd knowledge construction described in chapter 4. Tausczik *et al*. [47] found the same collaborative knowledge construction over another knowledge domain and channel, the mathematics domain on Math Overflow (a Q&A channel focused on solving mathematical problems). My findings extends what Tausczik *et al*. found to other media channels and domains.

The findings of this thesis can be used to identify how channels' features or community members might affect the construction of knowledge. For instance, I identified that gamification might affect collaboration between users. Users prefer to create their own answer instead of collaborating with others. Additionally, it might be possible for indirect collaboration, like the one happening on the comments on Stack Overflow to improve discussion and participatory knowledge construction if there was a mechanism to provide points for this type of participation. However, more studies are required to extend my observation to other domains, communities, and channels.

Moreover, through the survey I identified that there are certain benefits for keeping the history of the question available. As U26 said, there are some benefits to reading what a user thinks is not important for conceptualized questions, but which may lead to out of the frame thinking.

I believe that is important to understand how the knowledge is constructed on media channels, and how different mechanisms such as gamification or topic restriction can affect the knowledge construction [26]. Through this understanding, researchers can gain insights of how to support future media channels, and user diversity [50].

6.4 GTMail

As previously mentioned in section 3.3, I developed GTMail[1], a software tool used to preprocess the date from the R-help mailing list archives. A few months later after creating my tool, and while my colleague and I were doing the analysis of the data, I noticed that Bettenburg has a similar tool based on his research[2]. The main differences between my tool and Bettenburg's tool lays in the extra features added to deal with the particular issues presented in our data and the requirements of our study. Specifically, one of the requirements was downloading emails with coding issues from the URL left by the mailing list server

[1] Our open source JAVA tool is available online at `https://github.com/cagomezt/GTMail`

[2] `https://github.com/nicbet/MailboxMiner`

after scrubbing the body, as well as dealing with standardizations issues (e.g., email addresses), and extraction of URLs resources. All those changes can be added as extensions to Bettenburg's tool.

Moreover, the extraction of URLs resources can be useful to analyse technology diffusion over media channels. For instance, Squire *et al*. [42] analysed URLs on different mailing list archives to identify the diffusion of tools such Pastebin and GitHub Gist (tool that support snippet code sharing, and syntax highlighting).

One of the cleaning data processes that I never implemented in the GTMail tool was the elimination of source code from the body of the message. As a consequence, this thesis relies on manual procedures to code the data as it was unnecessary to create the cleaning data step. However, according to Bacchelli *et al*. [4], the process of differentiating source code from users' messages might be necessary when analysing unstructured data from programmers (e.g., emails) using natural language processing techniques (e.g., Latent Dirichlet allocation or latent semantic analysis). The approach of Bacchelli *et al*. [4] could be a valuable extension to Bettenburg's tool.

Chapter 7
Threats to Validity and Limitations

This chapter outlines the threats to the validity of this study. As we had multiple researchers coding the data, and found using a case study methodology to be a challenging task, multiple limitations surfaced.

Internal Validity

Internal validity refers to the amount of bias within a study[10]. In an exploratory case study, the researchers' actions and biases affect the finished work in each step, such as during the selection and analysis of data.

In this study, two researchers with a similar background—both computer scientists from the same country, with English as their second language— worked together to code the data. The open coding technique used in this work involved the analysis of large collections of messages based on the researchers' observations. Our background might have biased how we coded the data, as well as our understanding of the context of the study. Additionally, our results may have been influenced by the method we used to select data for analysis (as explained in section 3.3.1) or the pre-processing of the information that we did before loading the data to the database (as explained in section 3.3.1), which might be influence the reconstruction of threads for the R-help mailing list. Moreover, there may have been bias in the mapping of message types between Stack Overflow and the R-help mailing. As the R-help mailing list contains unstructured data, our understanding of that data and the observations we made played a big role in the mapping exercise.

In terms of the techniques applied during the study, I used Cohen Kappa coefficient on categories that were not mutually exclusive. The purpose of calculating the coefficient were solely to trigger discussion between coders. However, it might introduce bias to the study as a consequence of Cohen Kappa's limitations—it should be applied over mutually

exclusive categories.

As consequence of our exploratory case study methodology and the parallel execution of *phase 1* and *2*, some of the survey results were not aligned with our findings. Moreover, the survey recruitment methods listed in chapter 6 might have biased the population. For instance, when we announced the survey through Stack Overflow, moderators removed our message a few minutes after we posted; for this reason, it is possible that we did not reach enough of the Stack Overflow population. The number of responses to the survey and the high variability of the answers might also contribute to the analysis bias—given that we could not obtain enough participants, we had to support users' opinions with just a few responses. The survey questions were created solely by the author of this thesis, and while we tried to conduct pilot studies, no one else had a thorough enough understanding of the research. After the survey data was collected, the author was the only researcher that analysed the information. Finally, as a consequence of running the survey phase in parallel with the data analysis some of the survey responses are not aligned with the results of this thesis.

In terms of the tool presented in this thesis, our biggest limitation was identifying the existence of Bettenburg tool while we were already conducting the analysis. Indeed there are two different approaches that solve the same problem and that are based on the same process of data cleaning. However, some of the GTMail tool features can be applied to Bettenburg as an extension of his work.

Moreover, the GTMail tool was tested manually, and through out the study of the R-help mailing list data (see Appendix D). All tests that I performed were not exhaustive, or formal. In addition, the GTMail tool was not compared against other tools that accomplish the same objective (e.g., MailingListStats, and REmail), neither tested to be resilient to all the possible errors that might exist in the data. This issue might represent a threat to validity of the GTMail tool.

External Validity

External validity is concerned with the extent to which the findings from this work can be generalized [34, 8]. As a case study, it cannot be assumed that these cases can be generalizable until further evaluations have been conducted [59]. The best example of the previous statement is that this thesis studied developers from the R community, which are not typical programmers. I consider R language users as *casual developers* due to their limited (or non-existent) programming experience (e.g., biologists or statisticians). The

R programming language is used to solve statistical problems, but it does not create any software product or tool at the end of the programming process—the final product is a script that process a specific data. As a consequence, this study might not represent the knowledge that a typical software developer community shares.

According to Yin [59], *"a case study (as with experiments) rely on analytic generalization. In analytical generalization, the investigator is striving to generalize a particular set of results to some broader theory"*. For this reason, findings in this research should be tested in other communities and with other channels to see if our findings would apply to these other contexts. There are some of my findings that are prove to be true over other knowledge domains, communities, or media channels. For instance, there are other studies that support some of our findings in other communities, such as Tausczik *et al.* [47] (i.e., knowledge construction) and Treude *et al.* [48] (i.e., question categorization).

Standards of Rigour

This work incorporated a number of approaches to minimize the threats to validity mentioned above and to establish rigour [34, 59]. These approaches included triangulating multiple sources of data (i.e., survey, documentation, and messages from both media channels), randomly selecting data, and using multiple researchers to code the data. As a result, we were able to identify and report discrepancies and contradictions.

This study was conducted over a one-year period in which we were heavily immersed in the information within the channel, such as messages and official community documentation. This allowed us to gain a good understanding of the context and the community. When we began to code the data, I told my colleague to code by "what she observed", avoiding any previous knowledge she had related to the topic (if any). The description of what we had to code was intentionally vague in order to minimize researcher bias. As was expected, our categorization did not always align, however, we came to agree on our interpretation of the data before starting the next coding session. As mentioned before, we used an inter-rater agreement coefficient (i.e., Cohen Kappa) to measure our mutual understanding and to promote discussion.

In terms of the data selection process, we minimized the threats by randomly selecting the data to analyse (see chapter 3). This guaranteed that the information selected had the same probability of being chosen for the study.

In summary, even with all the countermeasures that we applied to minimize bias, our findings are limited by our experience in the qualitative research environment and our vi-

sion of the world.

Chapter 8
Future Work

In this thesis, I was motivated to investigate the way programmers use media channels to share *knowledge* within a particular software development community. I decided to study two channels within the R community: the R-help mailing list and Stack Overflow. During the development of this thesis, I identified several dimensions related to knowledge: a classification of types of knowledge, categories and properties; a set of recommendations for the usage of multiple media channels; and an understanding of knowledge construction. The following subsections outline areas for future work based on our findings.

Is the categorization of knowledge different in other software development communities?

I presented a model for analysing knowledge based on the comparison of two Q&A channels. While my study was limited to the R community and two of its channels (the R-help mailing list and Stack Overflow), other developer communities (e.g., Firefox, Linux, or Android) also use Q&A channels. I believe that the type of knowledge (i.e., categories and properties) might be very similar among these other communities, but found in different channels. To address this question, our model can be applied to other similar communities of practice.

Are the topics of discussion the same for the R-help mailing list and Stack Overflow?

I studied these two channels to understand the types of knowledge they shared, which I found to be very similar. However, I did not study the topics of discussion they contained. I am interested in identifying if both channels discussed the same topics with the purpose of assisting users when posting questions to the correct channel. To address this question, there are automatic techniques that can be applied. For instance, Latent Dirichlet Allocation (LDA)[1] is a natural language processing technique that is used to analyse large bodies of text.

What should be the next evolutionary step for the R-help mailing list?

My study identified that the R-help mailing list brings the discussion benefit to the R community. I feel that this shows its importance within the R community. Survey participants provided a list of things that can be implemented to improve the experience for R-help's users. I am interested in understanding the next logical step of evolution for the R-help mailing list to continue supporting its extensive user base. To address this question, functional and non-functional requirements need to be evaluated, along with the execution of user studies (e.g., surveys and interviews) to address specific usage concerns.

[1] https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

Chapter 9
Concluding Remarks

This thesis explores the knowledge that flows through Stack Overflow and the R-help mailing list within the R community. In my approach, we performed a systematic comparison of both media channels and performed open coding to understand and classify knowledge. In order to conduct a fair comparison, I collected raw data from both channels (i.e., XML and MBOX files), transformed this data to a common format (i.e., analysis of the messages to map questions, answers, and other elements), and uploaded the information into databases to facilitate queries and other study-related activities.

Two researchers performed iterative open coding and attended regular meetings to compare results. After the analysis of more than 500 threads, we identified five types of messages (i.e., question, answer, comment, update, and flag) and more than 35 categories (along with their properties), including how-to question, tutorial answer, announcement update, not-an-answer flag, and clarifications comment (for the complete list, refer to chapter 4.1).

Additionally, my colleague and I identified two mechanisms for the construction of knowledge: participatory, in which answers are created with the collaboration of various users; and crowd-based, which is non-collaborative and where solutions are posted without any acknowledgement to previous answers. We also analysed links and how these contribute to the construction of knowledge, such as by referencing source code, projects, and other posts related to a particular question.

Furthermore, my colleague and I identified user behaviours based on their comments and noted certain usage characteristics in both channels (i.e., analysing equivalent questions in the same subject, by the same user posted in both channels), including solving their own question and providing the answer, and cross-posting.

Finally, I conducted a survey that collected 26 answers from R users in order to bring further insights on the findings of this thesis. My survey provided us with a list (based on

the participants' opinions) of the pros and cons of using both channels that I incorporated into the set of recommendations for using multiple channels, and resources.

In this work, I pose that understanding the interplay between channels should be the next step to gain further insights into software development practices. To that end, I provided a contrast of the way knowledge is shared on Stack Overflow and the R-help mailing list, as well as an extensible categorization of Q&A channel messages that is meant to be used to compare and analyse knowledge in media channels. Additionally, I included a set of recommendations for using multiple media channels and external resources, a tool to pre-process MBOX format mailing list archives, and R community insights based on the analysis of the knowledge categorization and observations. I hope that my research will be useful to other researchers interested in how developers share knowledge.

Appendix A

This chapter present the results of the close question asked on the survey. Most of the closed questions were of multiple selection, which are presented on bar graphs, given that the total of the answers do not represent the number of participants who answered. The rest of answers are presented on pie charts as the number of the answers represent the 100% of the population.

A.1 Survey Results

The following section presents the results of our survey.

A.1.1 The Participants

As mentioned before, we distributed our survey through different media channels. We targeted users of R language regardless of their occupation and area of expertise. We used multiple-choice questions to get information that could help us profile the participants of our survey:

- *What is your area of expertise?*
- *What is your occupation?*
- *How would you describe your participation in the R community?*

Figure A.1 summarizes their responses. The majority of the participants selected areas of expertise different to computer science. It is worth mentioning, that Stack Overflow is a channel dedicated to software development topics. Therefore, it is possible that Stack Overflow does not fully support the communication needs of the R community. Among

the occupations reported included consultants, retailers, and government researchers. Surprisingly,many participants were package maintainers. We expected that this type of users would be more active in the channels of their corresponding packages rather than the ones we were studying.

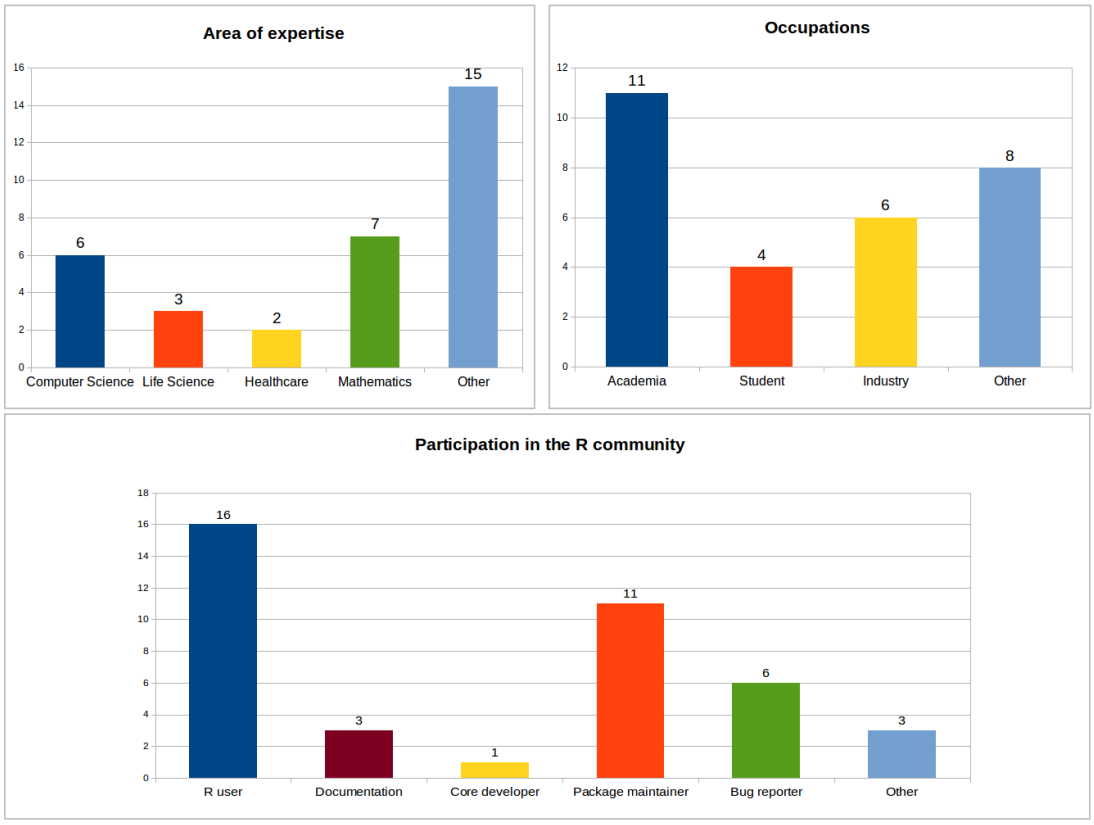


Figure A.1: Demographic profile of the participants.

Additionally, we asked participants if they had previous experience as programmers before learning R:

- *How would you rank yourself as a software developer?*
- *How would you rank yourself as an R user?*

Figure A.2 depicts the answers to these questions. Although, most users were non-computer-scientists, we were surprised to find that the majority have experience with programming.

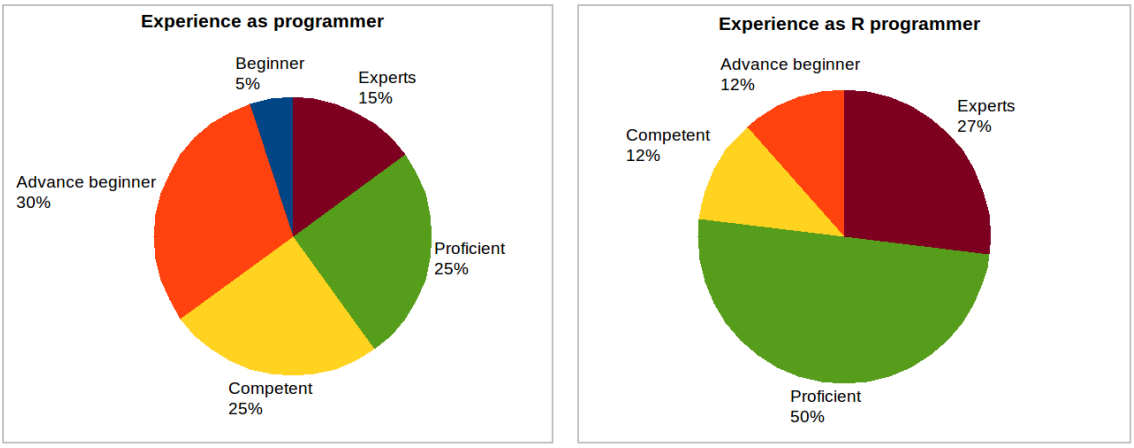


Figure A.2: (on the left) Programming experience as programmers; (on the right) Programming experience as R programmers

A.1.2 The Usage of the Media Channels

All of the 26 participants reported using of Stack Overflow. Given that most participants were experienced with programming, we could infer that would employ Stack Overflow. In contrast, only 20 participants used the R-help mailing list.

We were interested in the form of participation on each media channel. For this purpose we formulated closed questions in which users were presented multiple forms of participation (e.g., ask questions, write answers, edit answers or questions, add comments, and browse for information) and they had to provide a ranking value according a scale (never, rarely, sometimes, often, always, and no answer):

- *How do you participate on Stack Overflow/the R-help mailing list?*
- *When looking for an answer, what do you read?*
- *Before posting/writing your answer, what do you read?*

Figure A.3 depicts the participation on the channels. It was very interesting to see that the participants often consult the channels to browse for information and rarely to ask questions. By this, we can infer that most of the survey participant just consume channel knowledge.

Figure A.4 depicts what users read from the thread before posting an answer. As expected, most users of Stack Overflow look for the accepted answer, but still there is a very active participation on reading all answers and comments. In the case of the R-help mailing list, there is more variation in the responses. We think this is because of the email

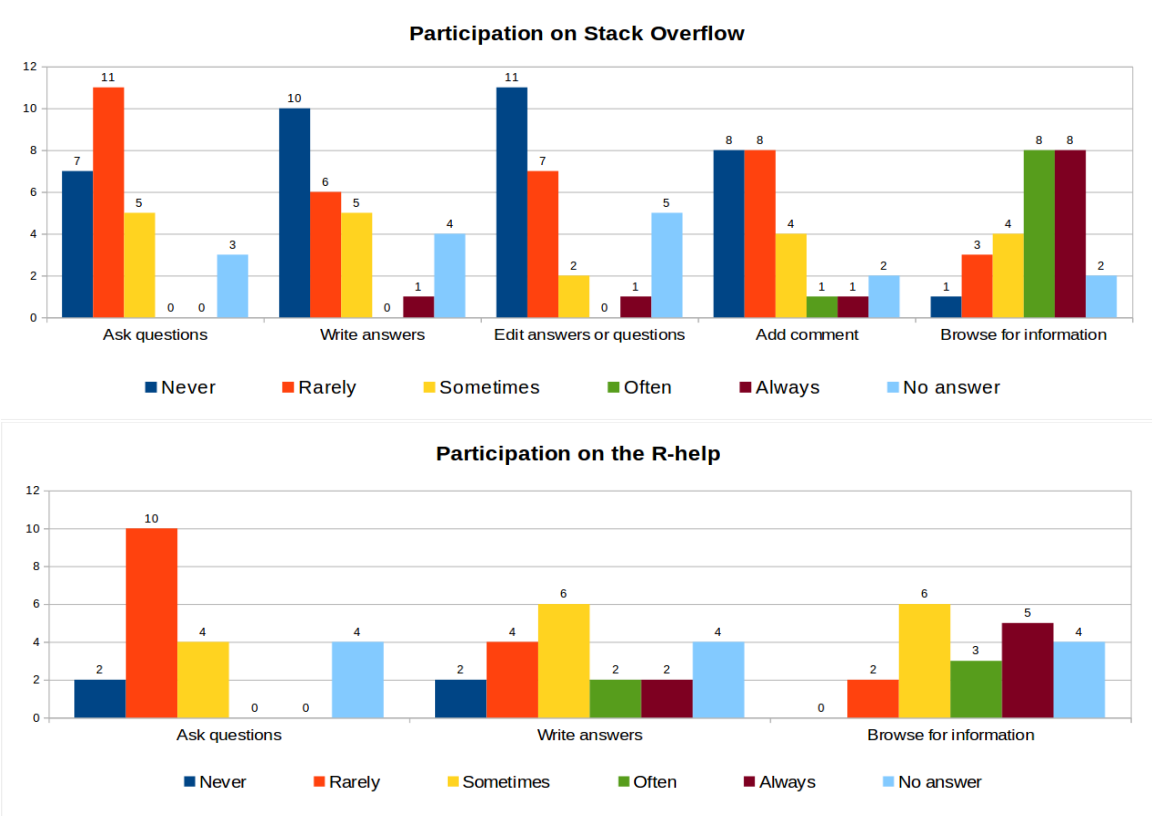


Figure A.3: Participation on Stack Overflow and R-help mailing list.

format requires a better examination of the thread before finding an answer or solution to a problem.

Figure A.5 depicts the behaviour of the participants before posting/writing an answer. As noted, many participants did not provide an answer to the question. This is not a surprise, since participants tend to look for answers rather than provide them. However, for those who did answers it is interesting that they try to avoid repeating answers by examining previous answers before posting. This means that the participants are interested to preserve the quality of the channel by not posting redundant information.

Finally, we asked about the usage of links on these channels. In this question, we provided multiple types of usage (i.e., input data, source code, documentation, external libraries, authors/users, publicity, and apps) and the participants were asked to provide a rank according to scale: Not important at all, Somewhat important, Important, Very Important, Not sure/Not Applicable, and No answer.

- *In your opinion, links are mechanisms to share.*

Figure A.6 depicts the opinions of the participants about resources usage. It is worth

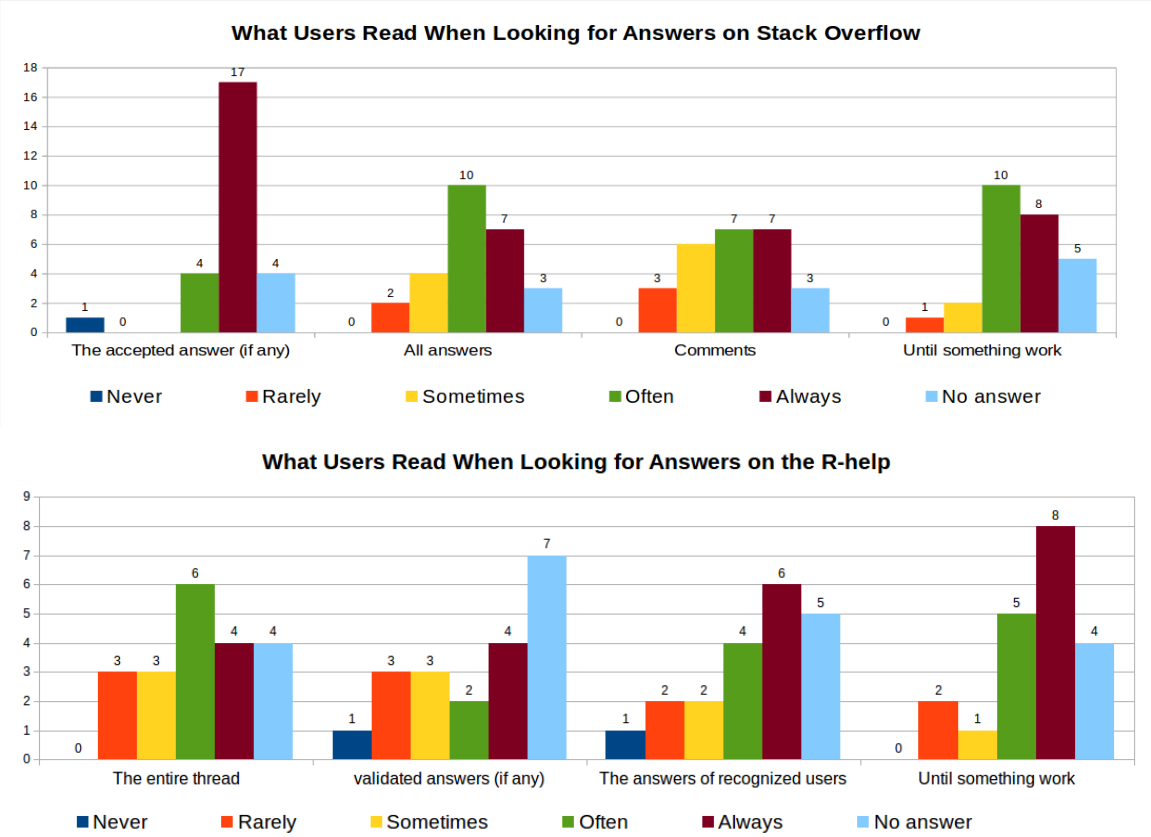
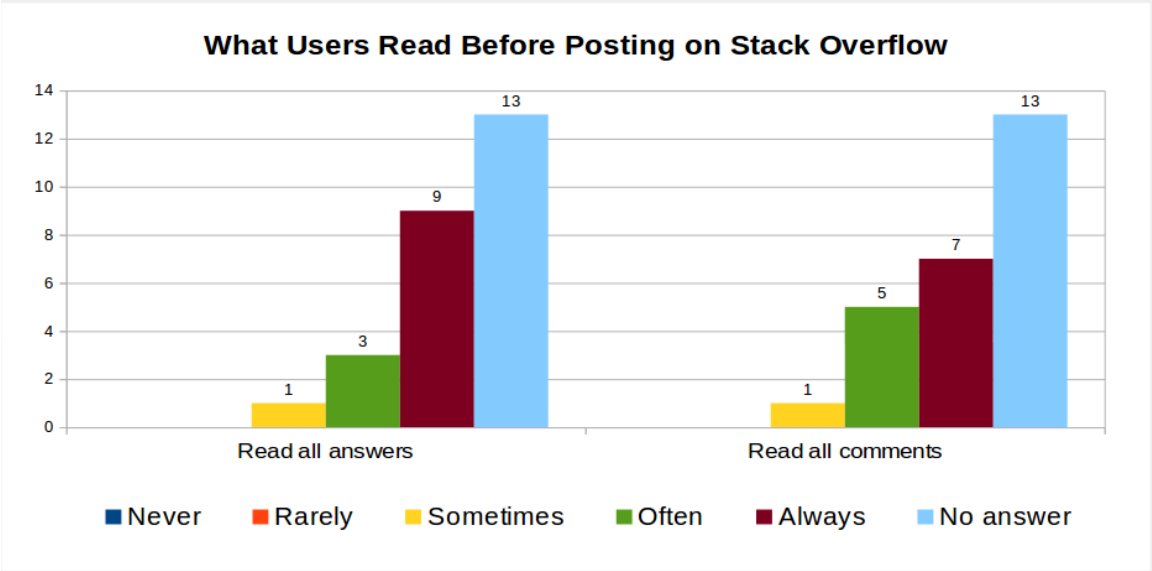


Figure A.4: Behaviour of the participants during enquiry process. Stack Overflow on top, R-help mailing list on the bottom.

mentioning that despite the variety of answers, we can highlight that documentation and source code are the most common reasons for providing links.



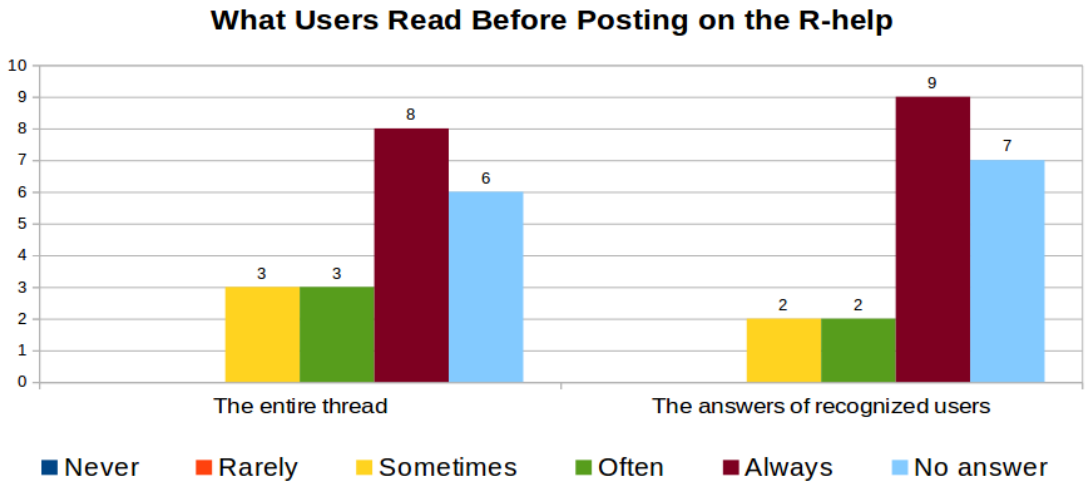


Figure A.5: Behaviour of the participants prior to a response.

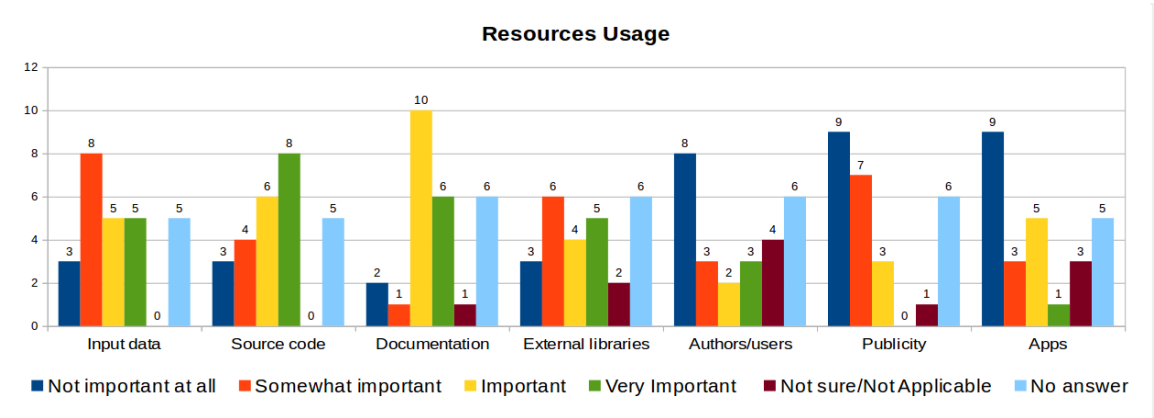


Figure A.6: How resources are used according to participants of the survey

Appendix B

This chapter presents the queries templates that I used to select the information for this study. Additionally, this chapter presents the queries used to create the table from which I extracted the information to code.

B.1 Database Queries

Random rows: The instruction to force PostgreSQL to return random results `ORDER BY RANDOM()`. Additionally, The instruction to force PostgreSQL to return a limited number of rows is `LIMIT <NUMBER>`. Code B.1 depicts the query template used to returns random.

```
SELECT *
  FROM <table>
 WHERE <date> BETWEEN <year1> and <year2>
 ORDER BY random() LIMIT <number>;
```

Code B.1: Query that returns a limited number of rows ordered randomly between two years.

Threads with the same title and authors: Using the MD5 hash I matched threads with the same author and the same title across the R-help mailing list and Stack Overflow. Code B.2 presents the query template used to match authors and titles on both channels.

```
SELECT *
  FROM <table> alias1
 WHERE EXISTS (SELECT 1
                 FROM <table> alias2
                WHERE alias2.title = alias1.subject
```

```
                AND alias2.md5 = alias1.md5
                AND alias2.posttypeid = 1)
  AND alias1.year between <year> and <year>;
```

Code B.2: Query that returns all the threads that share the same title and author.

Table creation: I created tables that contain all the possible information of Stack Overflow and the R-help mailing list on the spreadsheet format. Code B.2 shows the query used to create the tables for the threads

```
CREATE TABLE AvailableThreads AS
SELECT 'MessageId: ' || messageid || E'\n' ||
       'Subject: ' || subject || E'\n' ||
       'User: ' || "from" || E'\n' ||
       'Channel: ML' || E'\n' ||
       'Date: ' || datetime || E'\n' ||  E'\n' ||
       --trim(both ' ' from message)
       substring(message from 1 to 100) "message",
       'ML' "channel",
       (SELECT count(1)
          FROM ml_mail internal_ml
         WHERE ml_mail.messageid = internal_ml.messageid
           AND internal_ml.type = 2) num_answers,
       messageid
  FROM ml_mail
 WHERE messageid in ( SELECT messageid FROM ml_q_08_13)
  AND  ml_mail.type = 1
 UNION
SELECT 'MessageId: ' || CAST(id  as Text) || E'\n' ||
       'Subject: ' || title || E'\n' ||
       'User: ' || CAST(owneruserid as Text) || E'\n' ||
       'Channel: SOF' || E'\n' ||
       'Date: ' || creationdate || E'\n' || E'\n' ||
       substring(body from 1 to 100) "message",
       'SO' "channel",
       (SELECT count(1)
          FROM posts internal_post
         WHERE internal_post.parentid = posts.id
           AND internal_post.posttypeid = 2) num_answers,
       CAST(id  as Text) id
  FROM posts
   AND posts.posttypeid = 1;
```

Code B.3: Table template based on a query that returns all the possible threads that can be coded for this thesis.

B.2 Data

All the messages that I used for this study are available in PostgreSQL dump format at `https://github.com/cagomezt/RResearch`. The email address and the MD5 hash have been deleted from the data for privacy reasons.

Appendix C

This appendix lists the questions used in the survey highlighted in Chapter 3. The questions were open ended and close questions (i.e., multiple selection and unique selection). A copy of the survey is published in `http://goo.gl/mxmH5J`

C.1 Survey Questions

C.1.1 The User

- What is your area of expertise? (Computer Science, Mathematics, Life Science, Healthcare, and others).
- What is your occupation? (Academia, Student, Industry, and Other).
- Do or did you have experience as a software developer/programmer prior to learning or using R? (yes or not).
- How would you rank yourself as a software developer? (Beginner, Advance beginner, Competent, Proficient, and Expert).
- How would you rank yourself as an R user? (Beginner, Advance beginner, Competent, Proficient, and Expert).
- How would you describe your participation in the R community? (I'm just an R user, I contribute to the R documentation, I'm one of the R core developers, I write or maintain R packages, I submit R bugs, I am not Involved at all, and Other).
- Have you used Stack Overflow? (yes or no).

- How do you participate on Stack Overflow? (Never, Rarely, Sometimes, Often, Always).
  - Ask questions?
  - Write answers?
  - Edit answers or questions?
  - Add comment?
  - Browse for information?
- When looking for an answer, what do you read? (Never, Rarely, Sometimes, Often, Always).
  - The accepted answer (if any)?
  - All answers?
  - Comments?
  - Read the answers until find something that works for you?
- Before posting your answer, what do you read? (Never, Rarely, Sometimes, Often, Always).
  - Read all answers?
  - Read all comments?
- Have you experienced any challenges using Stack Overflow? Please elaborate.
- What motivates you to answer questions or add comments on Stack Overflow? Please elaborate.
- Have you used the R-Help Mailing List? (yes or no)
- How do you participate in the R-Help Mailing List? (Never, Rarely, Sometimes, Often, Always).
  - Ask questions?
  - Write answers?
  - Browse for information?

- When looking for an answer, what do you read? (Never, Rarely, Sometimes, Often, Always).
  - The entire thread?
  - Answers that were validated by the author of the question (if any)?
  - The answers of recognized users?
  - Read the answers until find something that works for you?
- Before writing your answer, what do you read? (Never, Rarely, Sometimes, Often, Always).
  - The entire thread?
  - The answers of recognized users?
- Have you experienced any challenges using the R-Help Mailing List? Please elaborate.
- What motivates you to answer questions on the R-Help Mailing List? Please elaborate.
- Why do you think the R-Help Mailing List has not been replaced by Stack Overflow? Please elaborate.
- In what situations would you choose Stack Overflow over the R-Help Mailing List? Please elaborate.
- In what situations would you choose R-Help Mailing List over Stack Overflow ? Please elaborate.
- When you see a link on a question, answer or comment. Do you click on it? Why? Please elaborate.
- In your opinion, links are mechanisms to share... (Not important at all, Somewhat important, Important, Very Important, Not sure/Not Applicable)
  - Input data?
  - Source code?
  - Documentation?

  - External libraries?
  - Authors/users?
  - Publicity?
  - Apps?

- Within the context of Stack Overflow and the R-Help Mailing List, can you think of any other benefits of using links? Please elaborate.

Appendix D

This appendix presents examples of the data that can be obtained from the GTMail tool, and how those examples are visualized in online R-help archives.

D.1 GTMail Tool

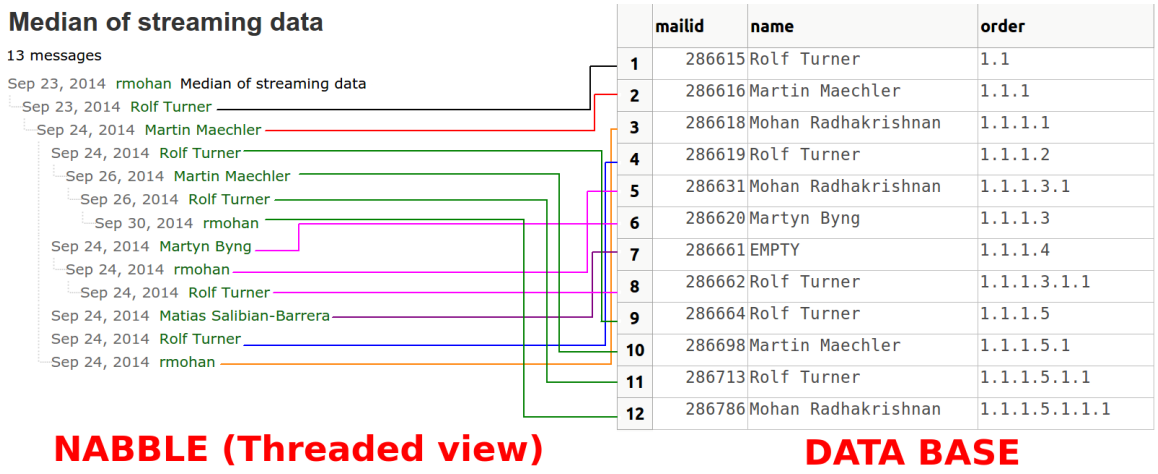
D.1.1 Threading

GTMail tool implements the Jamie Zawinski's algorithm[1] to do the threading of emails. Zawinski's algorithm is capable of dealing with threading and sub-threading. Figure D.1 shows an example of the thread *"Median of streaming data"* visualized on the Nabble website (an online R-help archive), and how the same data is stored in the database.

D.1.2 Messages

Figure D.2 depicts an example of how the messages are store in the database, as well as its visualization on the Nabble website.

[1] https://www.jwz.org/doc/threading.html



	mailid	name	order
1	286615	Rolf Turner	1.1
2	286616	Martin Maechler	1.1.1
3	286618	Mohan Radhakrishnan	1.1.1.1
4	286619	Rolf Turner	1.1.1.2
5	286631	Mohan Radhakrishnan	1.1.1.3.1
6	286620	Martyn Byng	1.1.1.3
7	286661	EMPTY	1.1.1.4
8	286662	Rolf Turner	1.1.1.3.1.1
9	286664	Rolf Turner	1.1.1.5
10	286698	Martin Maechler	1.1.1.5.1
11	286713	Rolf Turner	1.1.1.5.1.1
12	286786	Mohan Radhakrishnan	1.1.1.5.1.1.1

Figure D.1: Example of how we stored threads on the database after GTMail processed the data. (LEFT) An example of how the messages on a thread are visualized on Nabble. (RIGHT) How the information of threads is stored in the database

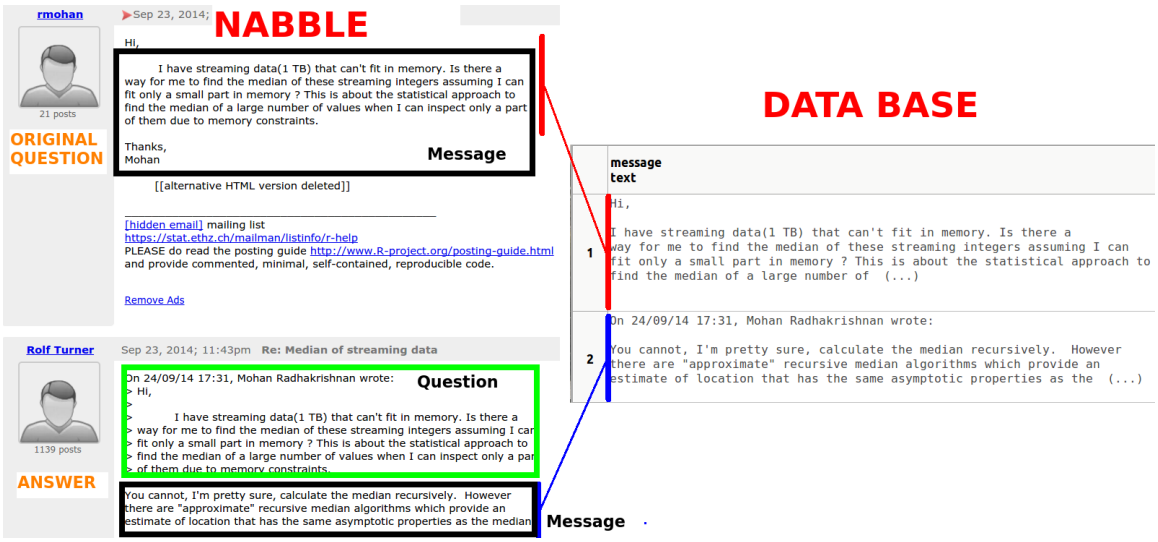


Figure D.2: Example of how messages are stored in the database, and how are they visualized in the Nabble website. (LEFT) The message visualized on Nabble. (RIGHT) The message as stored in the database

Bibliography

[1] M. Allamanis and C. Sutton. Why, when, and what: Analyzing stack overflow questions by topic, type, and code. In *Proceedings of the 10th International Working Conference on Mining Software Repositories*, pages 53–56. IEEE, 2013.

[2] J. Antin and E. F. Churchill. Badges in social media: A social psychological perspective. In *CHI 2011 Gamification Workshop Proceedings (Vancouver, BC, Canada, 2011)*, 2011.

[3] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider. Answering questions about unanswered questions of stack overflow. In *Proceedings of the 10th International Working Conference on Mining Software Repositories*, pages 97–100. IEEE, 2013.

[4] A. Bacchelli, T. Dal Sasso, M. D'Ambros, and M. Lanza. Content classification of development emails. In *Proceedings of the 34th International Conference on Software Engineering*, ICSE '12, pages 375–385, Piscataway, NJ, USA, 2012. IEEE Press.

[5] A. Barua, S. W. Thomas, and A. E. Hassan. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, page To appear, 2012.

[6] N. Bettenburg, E. Shihab, and A. E. Hassan. An empirical study on the risks of using off-the-shelf techniques for processing mailing list data. In *ICSM'09: Proceedings of the 25th IEEE International Conference on Software Maintenance*, pages 539–542. IEEE Computer Society, 2009.

[7] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan. Mining email social networks. In *Proceedings of the 2006 International Workshop on Mining Software Repositories*, MSR '06, pages 137–143, New York, NY, USA, 2006. ACM.

[8] A. M. T. Bobby J. Calder, Lynn W. Phillips. The concept of external validity. *Journal of Consumer Research*, 9(3):240–244, 1982.

[9] F. Calefato, F. Lanubile, M.-C. Marasciulo, and N. Novielli. Mining successful answers in Stack Overflow. In *MSR '15 Proceedings of the 12th Working Conference on Mining Software Repositories*, pages 430–433, 2015.

[10] J. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2009.

[11] D. German, B. Adams, and A. Hassan. The evolution of the r software ecosystem. In *Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on*, pages 243–252, March 2013.

[12] M. Goeminne and T. Mens. A comparison of identity merge algorithms for software repositories. *Science of Computer Programming*, 78(8):971 – 986, 2013. Special section on software evolution, adaptability, and maintenance & Special section on the Brazilian Symposium on Programming Languages.

[13] C. Gomez, B. Cleary, and L. Singer. A study of innovation diffusion through link sharing on stack overflow. In *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on*, pages 81–84, #may# 2013.

[14] T. Groenewald. Memos and memoing. In L. M. Given, editor, *In The Sage Encyclopedia of Qualitative Research Methods*, pages 506–507. SAGE Publications, Inc., 2008.

[15] A. Guzzi, A. Bacchelli, M. Lanza, M. Pinzger, and A. v. Deursen. Communication in open source software development mailing lists. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 277–286, Piscataway, NJ, USA, 2013. IEEE Press.

[16] B. Hartmann, S. Doorley, and S. R. Klemmer. Hacking, mashing, gluing: Understanding opportunistic design. *IEEE Pervasive Computing*, 7(3):46–54, #jul# 2008.

[17] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

[18] G. G. K. J. Richard Landis. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[19] H. Jenkins. *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, 2009.

[20] J. Jiang, L. Zhang, and L. Li. Understanding project dissemination on a social coding site. In *Reverse Engineering (WCRE), 2013 20th Working Conference on*, pages 132–141, #oct# 2013.

[21] F. Jin, E. Dougherty, P. Saraf, Y. Cao, and N. Ramakrishnan. Epidemiological modeling of news and rumors on twitter. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, SNAKDD '13, pages 8:1–8:9, New York, NY, USA, 2013. ACM.

[22] D. Kavaler, D. Posnett, C. Gibler, H. Chen, P. Devanbu, and V. Filkov. Using and asking: Apis used in the android market and asked about in stackoverflow. In *Social Informatics*, volume 8238 of *Lecture Notes in Computer Science*, pages 405–418. Springer International Publishing, 2013.

[23] F. Khomh, T. Dhaliwal, Y. Zou, and B. Adams. Do faster releases improve software quality? an empirical case study of mozilla firefox. In *Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on*, pages 179–188, #jun# 2012.

[24] E. Kouters, B. Vasilescu, A. Serebrenik, and M. van den Brand. Who's who in gnome: Using lsa to merge software repository identities. In *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*, pages 592–595, Sept 2012.

[25] J. Lang and S. Wu. Social network user lifetime. *Social Network Analysis and Mining*, 3(3):285–297, 2013.

[26] G. Li, H. Zhu, T. Lu, X. Ding, and N. Gu. Is it good to be like wikipedia?: Exploring the trade-offs of introducing collaborative editing model to q&a sites. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, pages 1080–1091, New York, NY, USA, 2015. ACM.

[27] K. M. MacQueen, E. McLellan, K. Kay, and B. Milstein. Codebook development for team-based qualitative analysis. *Field Methods*, 10(2):31–36, 1998.

[28] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann. 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, pages 2857–2866, New York, NY, USA, 2011. ACM.

[29] Michael Chui, James Manyika, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Hugo Sarrazin, Geoffrey Sands, and Magdalena Westergren. The social economy: Unlocking value and productivity through social technologies. Online, July 2012.

[30] A. Pal, R. Farzan, J. A. Konstan, and R. E. Kraut. Early detection of potential experts in question answering communities. In *UMAP*, pages 231–242, 2011.

[31] A. Pal, F. M. Harper, and J. A. Konstan. Exploring question selection bias to identify experts and potential experts in community question answering. *ACM Trans. Inf. Syst.*, 30(2):10, 2012.

[32] C. Parnin, C. Treude, and M.-A. Storey. Blogging developer knowledge: Motivations, challenges, and future directions. In *Program Comprehension (ICPC), 2013 IEEE 21st International Conference on*, pages 211–214, 2013.

[33] C. Rodríguez-Bustos and J. Aponte. How distributed version control systems impact open source software projects. In *Proceedings of the 9th IEEE Working Conference on Mining Software Repositories*, MSR '12, pages 36–39, Piscataway, NJ, USA, 2012. IEEE Press.

[34] P. Runeson, M. Host, A. Rainer, and B. Regnell. *Case Study Research in Software Engineering: Guidelines and Examples*. Wiley, 2012.

[35] R. Scupin. The kj method: A technique for analyzing data derived from japanese ethnology. *Human Organization*, 56(2):233–237, 1997.

[36] S. E. Sim and R. E. Gallardo-Valencia, editors. *Finding Source Code on the Web for Remix and Reuse*. Springer-Verlag, 2013.

[37] L. Singer, F. Figueira Filho, B. Cleary, C. Treude, M.-A. Storey, and K. Schneider. 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, pages 103–116, New York, NY, USA, 2013. ACM.

[38] L. Singer, F. Figueira Filho, and M.-A. Storey. Software engineering at the speed of light: How developers stay current using twitter. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, pages 211–221, New York, NY, USA, 2014. ACM.

[39] V. Singh, M. Twidale, and D. Nichols. Users of open source software - how do they get help? In *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, pages 1–10, Jan 2009.

[40] S. K. Sowe, I. Stamelos, and L. Angelis. Understanding knowledge sharing activities in free/open source software projects: An empirical study. *Journal of Systems and Software*, 81(3):431 – 446, 2008. Selected Papers from the 2006 Brazilian Symposia on Databases and on Software Engineering.

[41] M. Squire. Should we move to Stack Overflow?: measuring the utility of social media for developer support. In *37th International Conference on Software Engineering*, pages 219–228, 2015.

[42] M. Squire and A. Smith. The diffusion of pastebin tools to enhance communication in floss mailing lists. In *11th International Conference on Open Source Systems*, pages 45–57, 05/2015 2015.

[43] S. E. Stemler. A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. *Practical Assessment, Research & Evaluation, 9, 4*, 2004.

[44] K. T. Stolee and S. Elbaum. Exploring the use of crowdsourcing to support empirical studies in software engineering. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '10, pages 35:1–35:4, New York, NY, USA, 2010. ACM.

[45] M.-A. Storey, L. Singer, B. Cleary, F. Figueira Filho, and A. Zagalsky. The (r) evolution of social media in software engineering. In *Proceedings of the on Future of Software Engineering*, FOSE 2014, pages 100–116, New York, NY, USA, 2014. ACM.

[46] M.-A. Storey, C. Treude, A. van Deursen, and L.-T. Cheng. The impact of social media on software engineering practices and tools. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, FoSER '10, pages 359–364, New York, NY, USA, 2010. ACM.

[47] Y. R. Tausczik, A. Kittur, and R. E. Kraut. Collaborative problem solving: A study of mathoverflow. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '14, pages 355–367, New York, NY, USA, 2014. ACM.

[48] C. Treude, O. Barzilay, and M.-A. Storey. How do programmers ask and answer questions on the web? (nier track). In *Proceedings of the 33rd International Conference on Software Engineering*, ICSE '11, pages 804–807, New York, NY, USA, 2011. ACM.

[49] B. Vasilescu. Human aspects, gamification, and social media in collaborative software engineering. In *Companion Proceedings of the 36th International Conference on Software Engineering*, ICSE Companion 2014, pages 646–649, New York, NY, USA, 2014. ACM.

[50] B. Vasilescu. *Social Aspects of Collaboration in Online Software Communities*. PhD thesis, Eindhoven University of Technology, 11 2014.

[51] B. Vasilescu, A. Capiluppi, and A. Serebrenik. Gender, representation and online participation: A quantitative study. *Interacting with Computers*, pages 1–24, 2013.

[52] B. Vasilescu, V. Filkov, and A. Serebrenik. Stackoverflow and github: Associations between software development and crowdsourced knowledge. In *Proceedings of the 2013 ASE/IEEE International Conference on Social Computing*, pages 188–195. IEEE, 2013.

[53] B. Vasilescu, A. Serebrenik, P. T. Devanbu, and V. Filkov. How social q&a sites are changing knowledge sharing in open source software communities. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 342–354. ACM, 2014.

[54] B. Vasilescu, A. Serebrenik, and M. G. van den Brand. The babel of software development: Linguistic diversity in open source. In *Proceedings of the 5th International Conference on Social Informatics*, Lecture Notes in Computer Science, pages 391–404. Springer, 2013.

[55] S. Wang, D. Lo, and L. Jiang. An empirical study on developer interactions in stackoverflow. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 1019–1024. ACM, 2013.

[56] E. Wenger. Communities of practice: A brief introduction. Online, 2006. Visited on 2011.

[57] E. Wenger. Communities of practice and social learning systems: the career of a concept. In C. Blackmore, editor, *Social Learning Systems and Communities of Practice*, pages 179–198. Springer London, 2010.

[58] E. C. Wenger and W. M. Snyder. Communities of practice: The organizational frontier. *Harvard business review*, 78(1):139–146, 2000.

[59] R. K. Yin. *Case Study Research: Design and Methods, 4th Edition*. SAGE Publications, Inc, 4th edition, Jan. 2009.

[60] R. K. Yin. *A (very) brief refresher on the case study method*, chapter 1, pages 3–20. SAGE, 2012.