How R Developers explain their Package Choice: A Survey

Addi Malviya-Thakur*[||][‡‡], Audris Mockus[†][||], Russell Zaretzki[‡][||] Bogdan Bichescu[§] [||], Randy Bradley[¶][||]
*amalviya@vols.utk.edu, [†]audris@utk.edu, [‡]rzaretzk@utk.edu,[§]bbichescu@utk.edu,[¶]rbradley@utk.edu
[||]*University of Tennessee* , Knoxville, TN, USA
[‡‡]*Oak Ridge National Laboratory, Oak Ridge, TN, USA.*

Abstract—**Background: Contemporary software development relies heavily on reusing already implemented functionality, usually in the form of packages.**

Aims: We aim to shed light on developers' preferences when selecting packages in R language.

Method: To do that, we create and administer a survey to over 1000 developers who have added one of two common dataframe enhancement libraries in R to their projects: *data.table* or *tidyr*. We design a questionnaire using the Social Contagion Theory (SCT) following prior work on technology adoption and ensure that key dimensions affecting developer choice are considered.

Results: Of the 1085 developers we contacted, 803 completed the survey asking them to prioritize various factors known to affect developer perceptions of package quality and to provide their background. Most developers self-identified as data scientists with two to five years of work experience. We found significant differences between the preferences of developers who chose *data.table* and *tidyr*. Surprisingly, package reputation based on easy-to-see measures, such as the number of stars on GitHub, was not an important factor for either group.

Conclusions: Our findings demonstrate the inherently social nature of package adoption. They can help design future studies on how different populations of developers make decisions on which software packages to use in their projects. Finally, package developers and maintainers can benefit by better understanding the prime concerns of the users of their packages.

Index Terms—**Empirical Software engineering, Software engineering research, Software Supply chains, Software measurement, Code reuse, User behavior, Social Contagion Theory, Social aspects, R System**

I. Introduction

Open-source software has produced an immense number of software products conveniently provided in the form of packages that the end users could easily reuse. Package managers provide a centralized resource for such reuse, but it is generally unclear according to which criteria packages are selected, especially when multiple similar options are available. Analytic Hierarchy Process (AHP) has been used to compare software applications based on multiple criteria, including functionality, user-friendliness, dependability, and cost-effectiveness [16], [50], [52]. However, AHP is a relatively complex methodology that requires a high level of expertise and training, requires a significant amount of data, is time-consuming, and sometimes an opaque approach. In addition, there has not been much work done to develop a general approach that can be used to choose any software package, even if the methodologies for software selection offered in various research mainly follow the same procedure [22], [35]. Selecting a software package depends on factors like popularity, documentation, support, maintainability, license, and performance. Understanding these factors helps developers create high-quality packages, leading to wider community acceptance, increased reusability, and faster innovation. Motivated by these studies, our study aims to better understand the rationale and process for selecting packages by other developers. Although our study derives the understanding from two widely used R-language packages, we believe our exhaustive study can be generalized and extended to guide future research on selecting the most appropriate packages in other languages and package creators to create more competitive offerings.

We create and administer a survey focused on collecting developer priorities concerning their choice of a package. Since package selection is a form of technology adoption [2], [25], we apply Social Contagion Theory (SCT) [4] to consider factors that could influence the final selection. SCT stipulates that the ultimate selection is based on exposure (the need to be aware of the technology), infectiousness (the technology must provide some tangible benefits), and susceptibility (the adopter has to have a need for the particular technology), which we operationalize via measures of the environment, the package, and the developer. From the methodological perspective, our survey addresses the elements of multifaceted explorations of complex issues related to human aspects in software engineering development [43]. We achieve this by looking at two distinct real-life contexts of developers and introducing them to their projects.

To select the sample of survey respondents, we employ a mixed-method approach [6], where we analyze large volumes of data to select candidates for a survey and carefully select two candidate packages that provide similar functionality. Our target population is individuals who a) are using the R language, b) have created a public git repository, and c) were the first person to add a dependency on one of two commonly used data frame packages (*data.table [13]* or *tidyr [55]*) to their repository. The R programming language requires data reading and handling capabilities for data science and other fields. These two packages are well-established and heavily used packages for working with data in R. The number of final respondents used in the examination of the survey is 752

978-1-6654-5223-6/23/$31.00 ©2023 IEEE

of over a thousand people surveyed.

Our work aims to understand the factors that drive developers to select specific packages, as there is a limited understanding of users and contributors of public and open-source packages. Our survey captures the demographics and preferences of the developers to gain insight into this area.

The survey is assembled to discover developers' choice of packages. So, to get a clear response, we focused on a specific context where respondents have used these packages rather than randomly assembling the responders' population for the survey. So, it could be construed that our generalizability is limited, but our approach receives more in-depth and detailed feedback from the respondents. The survey respondents' population is chosen according to two criteria: a) to reduce the variability of the responses, we focus on a single programming language (R) and a pair of packages (*data.table* and *tidyr*); b) to ensure that all respondents are not just speculating about the topic, but had faced the problem and were authorized to make a decision.

Specifically, our survey examines *who are the individuals who create such projects and what their self-declared background and experience are*. It is reasonable to assume that background and experience affect how individuals prioritize package choice. The question of who are the individuals behind the massive growth of open-source projects is only partially answered[1] [2]. We, for example, do not know the proportion of data scientists who contribute to public repositories containing the R language. Also, since public repositories may be created for a number of reasons and package preferences may be partly dictated by these reasons (for example, in a class project teaching features of a specific package), we would like to know *the specific reasons why the repository was created.* Additionally, we probe this work's ultimate objective, which is divided into two parts. *First, we asked survey participants why they made the specific package selection they did.* Asking specific questions in particular contexts produces more accurate and reliable responses [26]. *Finally, we explore the criteria the respondents claim to use to prioritize package selection.*

Major Contributions:

- We report the distribution of different types of participants in R language-related public source code repositories. In particular, most participants self-declared not as developers but as data scientists, with only a small fraction being software developers or students. Most of the respondents had between two and five years of work experience.
- We applied the Social Contagion theory in the context of developer priorities used in package adoption.
- We report the distribution of R-language-related public repositories according to their purpose. The results indicate that personal research projects predominate with a smaller fraction of projects involving software development and an even smaller set focused on training.
- We obtain insight into why two commonly used R-language packages were selected by the individuals who chose to use them. While specific to these two packages, such detailed insights provide an empirical basis for further investigations into reasons for code reuse.
- Our study found that easily observable measures of package quality, such as stars or forks, were not considered important by survey respondents when prioritizing packages. Performance and compatibility with other software used in a project were considered more important. These findings can inform the development of package quality measures that align with users' criteria for selecting software, making the reuse process easier and more successful;
- We investigated if the package selection preference varies among the sub-populations of users. Specifically, we find that the preferences expressed by users of *data.table* differ from those of *tidyr* users. This suggests not only that different users may need different types of support, but, even more importantly, hints that there may be a need for multiple libraries with similar functionality but distinct non-functional characteristics.

The rest of the paper starts with a review of related work in Section II, a description of the research methods used to obtain the developer sample, survey instrument, and methodology used to analyze the survey results is illustrated in Section III. The results are presented in Section IV followed by analysis in Section V, discussion in Section VI, limitations in Section VII, and in Section VIII, we conclude our paper.

All data and materials used in this study are publicly available in the replication package: https://anonymous.4open.science/r/ReplicationPackageRSurveyPaper-0898/. The replication package includes survey questions, data, and code.
Background and

II. Background and Related Work

Here we review prior work on package selection, criteria, and relevant factors that contribute to package usage.

A. Studies about R Package Selection

R is becoming a more popular software environment among scientists and practitioners for data analytics and statistical computation Muenchen et. al. [56], Wendt et. al [36], and Wickham et. al [54]. Since there are thousands of packages, it is a nontrivial exercise to select which package works for a given situation and how to assess its viability. Individuals base their selection of packages on a host of criteria that ensure usefulness, dependency, and reliability [11], among others. In [10], [38], [53], [57], authors proposed various guidelines for locating relevant packages and choosing which package is suitable for a given application. In [12], [27], [41], authors suggested several critical quality criteria, including people prior, forced competence, and indirect data towards establishing trust in the R packages. Previous work, such as [23], focused on comparing the features provided by the packages and not the selection criteria made by the end users.

[1] https://opensourcesurvey.org/2017/

[2] https://octoverse.github.com/

Our work primarily focuses on tapping developers' requirements, given multiple packages with nearly identical features available. Along these lines, another work by Hesselbarth et. al [21] focused on the functionality available within R and suggested making it more feature rich. However, this work also fails to address the underlying motivation of the developer community in selecting specific packages. Several studies examined a systematic approach to selecting R packages based on the task. They provided a list of recommended packages for common tasks, including studies that evaluate the performance of popular R packages for data science tasks, including data manipulation, visualization, and machine learning. They provided a comprehensive benchmark for package selection. These studies show the importance of carefully choosing the right R packages for specific data analysis jobs.

B. Social Contagion

Social contagion theory is a sociological concept that proposes people's behaviors and dispositions can be influenced by those of their social network [5]. Although there may not be any study on how to use social contagion theory to choose software packages, although when evaluating software options, decision makers may be influenced by the perspectives and experiences of others within their organization or industry. Decision makers may be more likely to choose a particular software package based on social influence if it is widely adopted and highly recommended by colleagues or industry professionals [29]. The social contagion theory can also be applied to adopting novel software tools or features. For example, suppose a new feature or tool is heavily promoted and adopted by early adopters in an organization or industry. In that case, it can be disseminated to other members of the network through word-of-mouth recommendations or social proof. Social contagion theory can influence decision makers' perceptions of software packages and features based on the experiences and opinions of others in their network [46]. There is a growing amount of study on the use of social networks in software development, although there may not be specific studies or research articles that combine social contagion theory and software package selection. To this end, our work is novel in using social contagion as a way to examine package selection.

III. Research Methodology

This section illustrates the survey methodology, design, timeline, and approaches to pursuing it through a planned execution. Subsequently, we elaborate on the methodology used to analyze the survey results and delineate the criteria for choosing and endorsing a specific package. We focus on developers who contribute to public repositories that contain R language source code and could be reused by other developers. Thus, project repositories that are not public are excluded from this population.

A. Survey Motivation

Developers could benefit from knowing which factors contribute to the long-term and high-quality support and enhancements of their software packages. It is not clear what criteria developers use when selecting packages, such as recency, availability, advertising, marketing, or word of mouth. Answering these questions could provide deeper insights into package adoption and help direct efforts for wider community acceptance and integration with third-party software. Higher quality packages with lower risk of non-adoption could lead to greater reuse and rapid innovation.

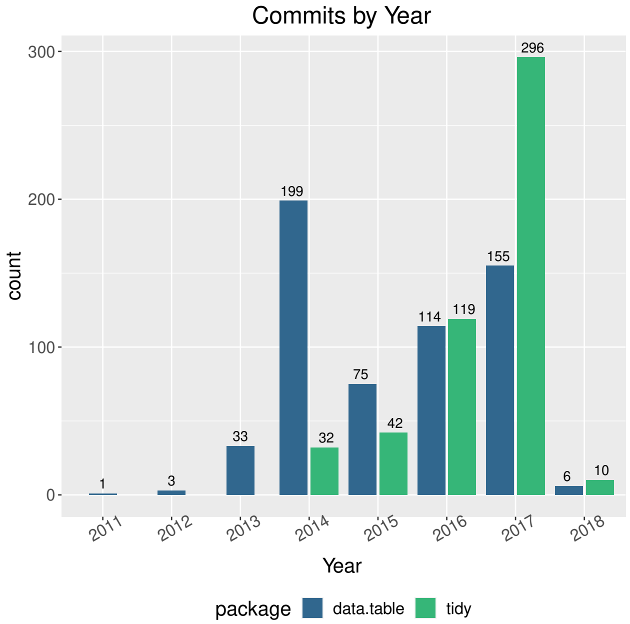


Fig. 1: Data mined from WoC

B. Selection of Survey Participants

This sub-section focuses on survey respondents' selection and summarizes their criteria. Random sampling and causal effects have long been associated with adequate participant selection [32], [45]. In [28], [34], [44], [45], the authors thoroughly examined the challenges surrounding participant selection, logistics, and their implications for many risks to the validity of the experiment, including the current state of participation selection procedures in software engineering experiments. One way to select participants is to ensure hands-on experience working on the scope of the survey [14], [15]. This provides some baseline level on the validity and authenticity of the responses received from the participants and contributes to strength representativeness [1], [8], [9]. Some have suggested the use of crowd-sourcing. However, its use is questionable since it may not yield an appropriate number of participants eligible for the survey [49]. In our work, we study demographic characteristics (age, gender, etc.) to leverage the understanding of their decisions in selecting the R packages. Studies such as [58] and [19] have done extensive work on demographics and statistics of data scientists, including gender, race, wage gaps, age, education levels, industry and experience, among others from which we have benefited in our work. In the proposed work, we perform a more targeted study of the demographics of data scientists to correlate with the selection, influence, and usage of the R package in software development processes. The limitations are:

- Respondents failed to submit the full survey.
- Respondents intentionally made incorrect input by selecting the first available option in all questions.
- sampling error, coverage error, and non-response error.
- Difficult to gauge respondents misunderstand what is being asked or otherwise provide information.

In order to overcome these challenges, we consider only the individuals who used the R language, contributed to open source, and introduced a new package into their repository. We expect that these criteria exclude irrelevant respondents/opinions, and also allow us to tease out potentially subtle differences between the sub-population of developers who ended up choosing different packages. To construct our survey population, we mined World of Code (WoC) [31] and selected over 1 billion commits on GitHub that had added *data.table* or *tidy* packages to project dependencies. Thus, based on commits from WoC, we ended up with 1085 survey candidates. Figure-1 shows the distribution of the survey candidates by the first time they introduced the corresponding package into their repository. The chart shows that *data.table* became available earlier and that *tidy* became extremely popular toward the end of the sampling period.

C. Description of Survey

The survey is designed based on Social Contagion Theory, to understand the decision-making process of a developer selecting a package compared to alternatives and what indicators play a critical role in the selection of packages. The survey has the following major components that attempt to capture user response - personal experience as a software developer, baseline of the studies, and introduce characteristics that create a diversion. The survey did ask about job experience, projects, and their status in the meantime. Afterward, the survey enquires why a specific software is selected over others.

D. Survey Instrument Development

The web-based survey instrument was organized into four sections: (a) Learn about the purpose of the project; (b) Reasons for choosing a particular package; (c) Factors that influenced the choices; and (d) Background about the participants. The survey form had questions with multiple or single checkboxes, drag-and-drop option menus, and short-answer input areas. The questions were nominal (ranging from 0 to 10, from not considered to very important), and others were subjective, requiring free-flow input of the text. The survey was voluntary, so participation in the research study should not take more than five minutes. The results of this study are to be used for scholarly purposes only, and the aggregate results are to be published with open access while doing our best to keep respondents' information confidential. Beforehand, a consent form was requested as part of the submission. A completed questionnaire constitutes consent to participate in the study for disclaimer purposes. The Institutional Review Board also approved the survey to ensure that privacy, including any information shared by the participants, is handled based on the guidelines received[3]. In addition, ethical discussions were held with respondents during the initial phases of the survey.

1) Purpose: The survey asks about the purpose of the project where the R packages were used to form a baseline understanding and motivation of the usage. This includes whether the project was completed for a class or training that the participant took at some time, whether it was personal research, or whether the project was intended for use by a wider audience, such as developers of other packages. An ”other” option attempts to capture a vast array of reasons that are otherwise impossible in a limited set of options.

2) Reasons: The next section of the survey determines the reasons for the selection of the *data.table* or *tidy* package [51]. It begins with the confirmation that either of the packages was used in its software development lifecycle. The next question asks which of the following more closely reflects why they chose to use a specific package. The possible options include the core 'data.frame' object lacking needed functionality, compatibility with other packages in the project, being recommended by others, or the package being included unintentionally. The survey asked descriptive responses (based on relevancy and to incorporate a free form of sharing the expression) about the types of criteria that are typically used to make a decision when choosing a package to use.

3) Influencing Factors: This part of the survey consists of questions based on the understanding of the researchers, the existing literature, and the SCT (Table -I). Based on our experience and relevancy to software development activity, we have identified 13 key factors (mapped to SCT) that influence developers' decisions when choosing to use a package in their project (as shown in Figure-2). The section asks the participating developers to rank each factor according to how important it was in making a helpful discussion on StackExchange about *data.table* or *tidy* their choice.

4) Participant Background: The last part of the survey asks participants about their background and their personal software development experience at the time of commits made to repositories containing either *data.table* or *tidy* R package, such as level of software development experience, gender, data scientist/engineer, among others.

The popularity of the R software environment for data analytics and statistical computation has resulted in the availability of hundreds of packages, making it difficult to choose the right package for a given application. Various guidelines and criteria for the selection of reliable and relevant packages have been proposed in previous studies [10], [11], [35], [54]. However, the majority of these studies focused on comparing packages based on functionality, ignoring the motivation of the developer in package selection. Our work seeks to fill this need by focusing on the needs of developers when selecting packages with similar capabilities. This manuscript underlines the necessity of carefully selecting the appropriate package for certain data analysis projects and offers practical guidance on

[3] https://shorturl.at/foHN3

how to do so. Next, we discuss the method section on survey design and development.

The remainder of the section provides detail of the research approach employed to study the survey responses and identify the factors that led to selecting and recommending a particular package. In addition to exploratory analysis, we use social contagion, net promoter score, correlation, and regression analysis to discover and ensure the validity and reliability of the results obtained.

E. Social Contagion Theory

Today, software development has become a social phenomenon, with several teams working together and solving challenging problems through interactions, message boards, external guidance, and communication [33], [47]. Several studies have been conducted based on social contagion to explain the interaction among individuals and their choices through each other's influence [17], [30], [42]. Social Contagion Theory (SCT) is a psychological and sociological concept that proposes behaviors, emotions, ideas, and attitudes that can spread through social networks, similar to how a contagious disease spreads throughout a population. In package selection, we use SCT to infer the choices and the reason behind them.

In our work, we examine what SCT factors drive the respondents' selection of the two packages. Social contagion theory concludes that emotions, actions, and ideas can spread across networks, similar to how illnesses spread through populations. Many elements, including social relationships, frequency of interaction, and influenceability, encourage this phenomenon. Social contagion can profoundly affect the understanding of human behavior, decision-making, and social dynamics.

F. Net Promoter Score (NPS)

We assess the likelihood of one developer recommending one of the two studied R packages to other developers. NPS is a loyalty metric that assesses the likelihood that an individual would refer something tangible to a friend or colleague. Since its introduction, it has been a popular metric for measuring loyalty and satisfaction.

On a scale from 0 to 10, how likely are survey respondents to suggest one of two R packages to others? We examine the respondents on a scale based on their responses. They are:

- Survey respondents who are exceptionally satisfied with the specific package are inclined to suggest it to others.
- Survey respondents who are satisfied with the specific package but are unlikely to suggest it to others.
- Unsatisfied survey respondents and where the odds are they might share bad opinions about the R packages.

Subtracting the percentage of those not recommended from the percentage of recommended yields the Net Promoter Score. The score ranges from 0 (if all respondents share a bad opinion) to 10 (if all respondents are exceptionally positive). We have used the Net Promoter Score [40] ranking of the objective question (0 = least, 10 = most likely recommended) as shown in Figure3. NPS is a potent indicator since it is simple to comprehend and can be used to evaluate respondents' strong preferences toward a package. In addition, it helps identify strengths and weaknesses by revealing what respondents like and dislike about the R packages. Net Promoter Score is a useful tool to measure the loyalty and advocacy of respondents and to drive the adoption of their preferred package(s).

G. Correlation Analysis

Correlation analysis aims to identify and quantify the relationship among the package selection variables (e.g., Performance, Growth, etc.). The primary objective is to determine whether there is a relationship between a pair of these selection variables and the strength of that relationship.

$$r = \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{(X-\bar{X})^2}\sqrt{(Y-\bar{Y})}^2} \qquad (1)$$

The Pearson correlation coefficient expresses the magnitude and direction of a linear relationship between two continuous variables. The Pearson correlation coefficient, commonly known as Pearson's r, has a value between -1 and +1, with -1 representing a perfect negative correlation, 0 indicating no correlation, and +1 indicating a perfect positive correlation.

H. Regression Analysis

Logistic regression is used to differentiate between these two populations of users. Regression analysis is a way to use statistics to look at how one or more independent factors relate to a dependent variable.

$$Y_i = f(X_i, \beta) + \epsilon_i \qquad (2)$$

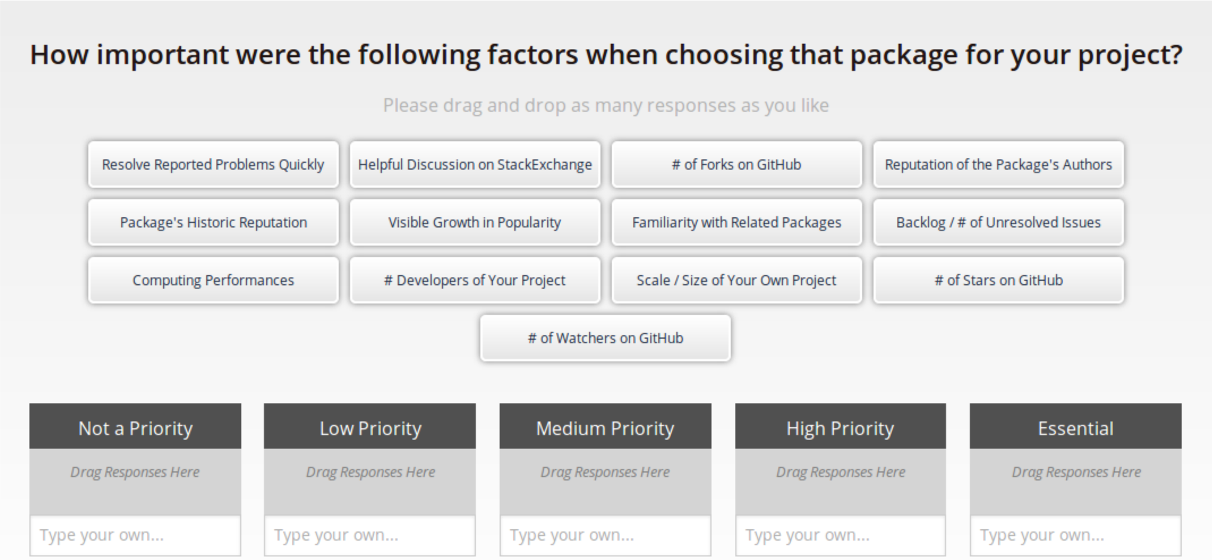
We want to study whether there are statistically significant differences between the priorities of the developers choosing *tidy* and *data.table*. The resulting coefficients show to what extent various aspects may have influenced the respondents' choices. We estimate a logistic regression model using the glm (generalized linear model) function in R.

I. Analysis of Subjective Responses

The survey responses are collected across several categories, and sometimes they are submitted as subjective responses in the form of free text by the respondents. The authors used the card sorting technique, a method for organizing and gathering information from the survey results [48]. Using this method, we categorized and sorted replies based on how respondents perceived or preferred the survey questions. The collected and sorted data helps to understand how respondents group and organize their package selection responses.

IV. Results

The survey was emailed to 1085 individuals; out of that, 803 sent their responses (74%). We removed observations that did not meet the purpose of this survey. They include i) Respondent indicated use for training only; ii) Respondent indicated that the library was included unintentionally; and iii) Commitments that were made before 2015 (that is, when we collected data and the subsequent year when we


How important were the following factors when choosing that package for your project?

Please drag and drop as many responses as you like

Fig. 2: Factors when choosing the package

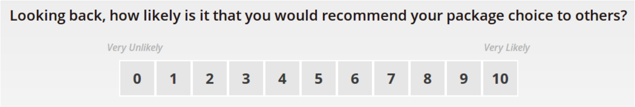

Looking back, how likely is it that you would recommend your package choice to others?

Fig. 3: Net Promoter Score for R package recommendation

administer the survey). After calculating the removal from the original number of responses received, we ended up with 752 responses.The final data set combines data for the *data.table* with 318 respondents and *tidy* with 434 responders. There are several different background characteristics of the respondents that we have captured in the survey. It includes average experience, primary programming language, type of job, sex, and age, among other demographic characteristics.

The analysis of the responses to the survey indicates that most of our respondents have between two and five years of experience and use primarily R as their programming language. In addition, most of them are data scientists working on at least 2-3 projects at any given time. Furthermore, our survey found that most of them are English speakers and are male between the ages of 25 and 34. In general, our survey respondents belong to a vibrant community of data scientists and software developers. Also, the recorded feedbacks were reliable and accurate.

Both packages have existed for quite some time and their evolution increased their *exposure* to the developer community. *data.table* is several years older and, presumably because of it, is more widely deployed. The indicators on the number of packages previously adopted them and the developer community sending such indicators would further amplify exposure, thereby raising the possibility of their increased adoption.

Figure-4, shows the purpose of the project for which each package was selected. Personal research dominates followed by software development work and training. The fractions are similar for both projects with `data.table` having a slightly higher fraction devoted to personal research and `tidy` for development work, though the differences are not significant.

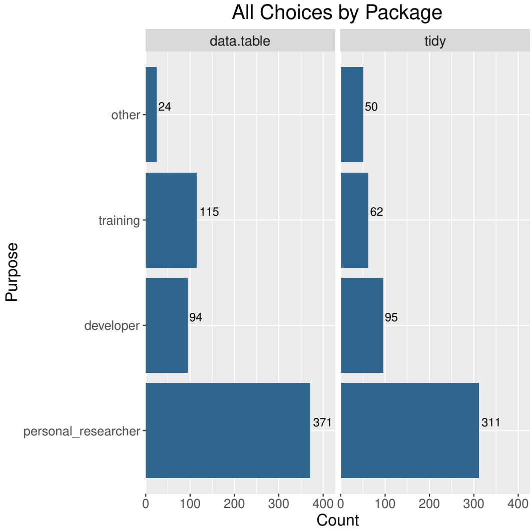


Fig. 4: Purpose of the project

A. Net Promoter Score

The infectiousness of a package can be partly gauged by the enthusiasm of the users. In the business world, one of the basic measures of consumer enthusiasm is the net promoter score (NPS). One simple question is the basis for the NPS - *How likely are you to recommend the brand to your friends or colleagues, using a scale from 0 to 10?* Respondents who give a score of 0-6 belong to the Detractors group. These unhappy customers will tend to voice their displeasure and can damage a brand. The passive group gives a score of 7-8 and is generally satisfied but is also open to changing brands. Promoters are loyal enthusiasts who give a score of 9-10, and these coveted cheerleaders will fuel the growth of a brand [39].

The survey results indicate that *data.table* has an industry average NPS of 28. 6%, while *Tidyr* has an average score of 59.4%. In the subsequent text, the answers to the remaining

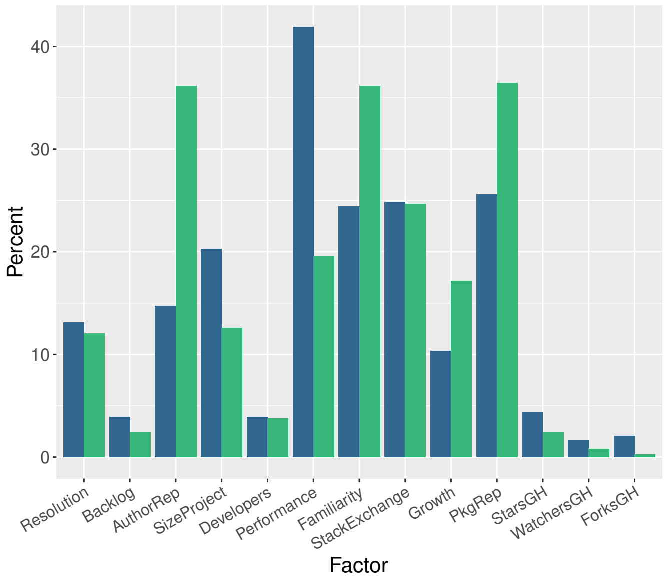


Fig. 5: Distribution of results across the community

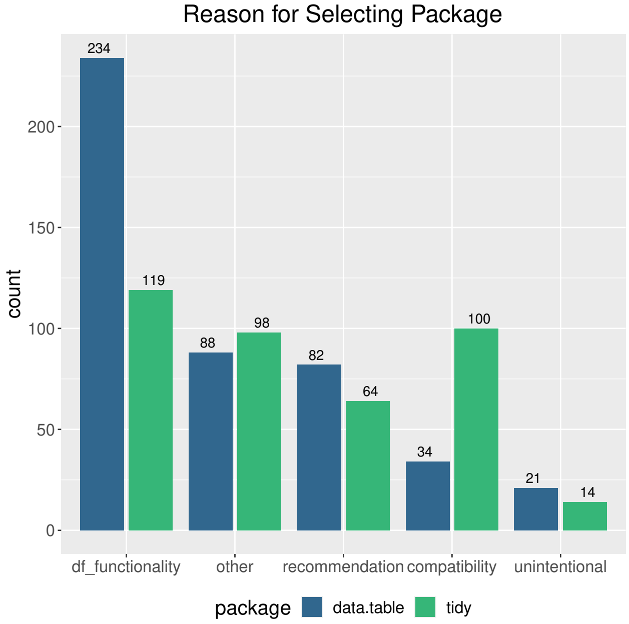


Fig. 6: Selection of packages

questions address what makes certain users so much more enthusiastic about their choice of data structure library. This was an essential lesson in the survey to understand and measure the participants' reactions to recommending the packages.

Figure-6 shows the distribution of stated reasons for choosing a package. It appears that the main reason why respondents use *tidyr* over *data.table* is that the latter object lacks compatibility with other packages that developers are presumably already using, such as packages from *Tidyverse*. The main reason *data.table* is preferred is its functionality (presumably capable of handling larger datasets).

TABLE I: Package Selection based on Social Contagion

Selection	**Package**	**Social Contagion Category**
Resolution	tidyr	infectiousness
Items in Backlog	tidyr	
Author Reputation	tidyr	
Developer Project Size	data.table	susceptibility
Number of Developers	tidyr	
Performance Needs	data.table	
Familiarity	tidyr	
StackExchange Discussion	data.table	exposure
Popularity Growth	tidyr	
Historical Reputation	tidyr	
Number of Stars	data.table	
Number of Watchers	data.table	
Number of Forks	data.table	

B. Factors Influencing Package Selection

Figure-5 shows the distribution of respondents' chosen priorities. The respondents designated these as essential or high priority reasons that influenced them to select either *data.table* or *tidyr* packages. Tidyr users value the author's and the package reputation. Also, familiarity(e.g. abundant online resources) is important to them when using *tidyr*. On the other hand, the *data.table* users' top preference is the computing performance, memory efficiency, concise syntax, and overall responsiveness in the result generation capability. Furthermore, in the case of *tidyr*, respondents voted high for familiarity, while in the case of *data.table* familiarity was not the main selection criteria. The *tidy* emphasizes compatibility and flexibility, which helps the respondents when they need to develop scalable code. In comparison, *data.table* is faster and has a small footprint, allowing for faster development and execution for small and large datasets. *tidy* users emphasize effective mitigation of package issues, future growth prospects and new features, familiarity, and coherent dependency on their development environment. The information provided by those who participated in the survey, users of *data.table* and *tidy* have stated that they regard StackExchange as a source of technical information. Finally, the *data.table* users are more concerned with the ranking of the package, performance needs, and how their large projects could accommodate the package.

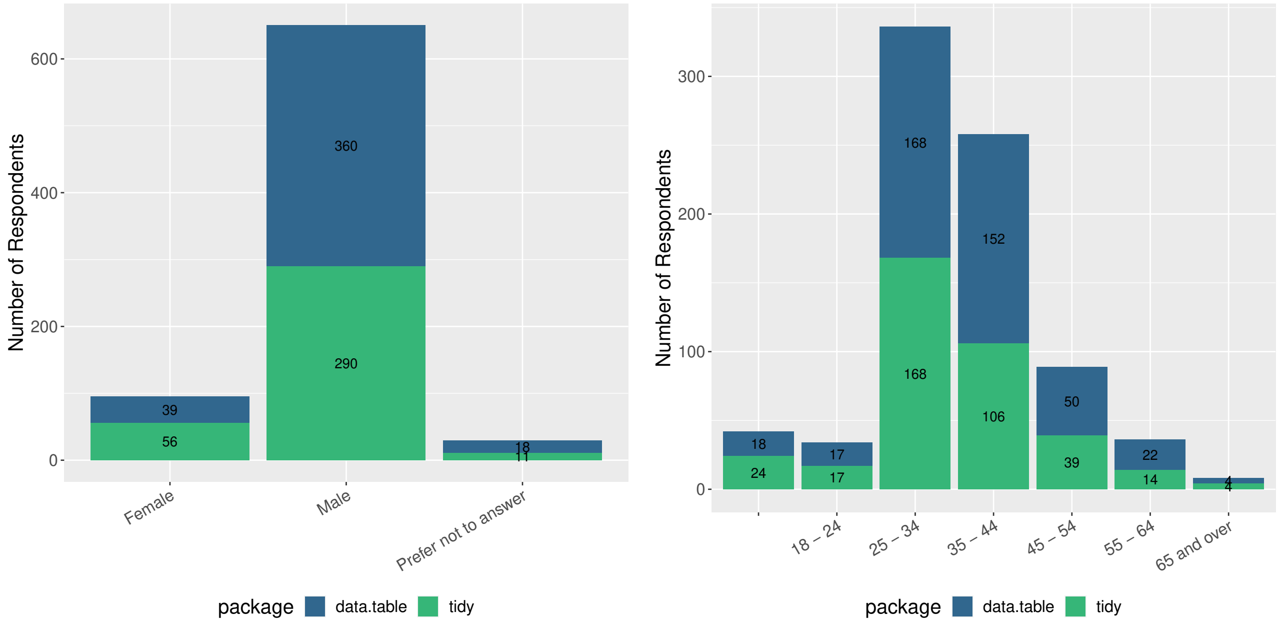
Finally, the secondary set of criteria that differed in popularity between the two packages were: *tidy* users value visible growth in user base (exposure) more than *data.table* users while *data.table* users tend to consider the scale of their own project (contagion) when prioritizing package selection.

1) Analysis of Respondents' Background: In Figure7, we have shown the background characteristics of the survey respondents. As noted above, most are 24- to 34-year-old male data scientists programming in R with two to five years of experience. Males and respondents 35 years and older appear to prefer *data.table* over *tidy*, while females and the younger group have chosen *tidy* more frequently.

V. Analysis

A. Social Contagion

We have mapped the SCT choices to the survey respondents' selection of either *data.table* or *tidyr* package in Table-I. We



(a) Gender

(b) Age

Fig. 7: Respondents' demographic background

found that *tidyr* is more infectious according to all three measures (which is consistent with the high NPS score). On the other hand, *data.table* appears to have more dominant exposure metrics consistent with its longer and wider deployment. A package's performance tops the individual selection criteria, followed by its reputation and the author's reputation. Thus, all three social contagion criteria are critical for a high package adoption. Also, packages' star ratings, forks, and watchers do not necessarily translate to a proportional adoption rate. Next, we perform correlation analysis on survey selection criteria to infer meaningful relationships among them.

B. Correlation Analysis

The correlations among the survey selection criteria variables that respondents identify as driving the package adoption are seen in Figure-8. Using the Pearson correlation coefficient, we examine the strength and direction of the linear relationship between these variables [37]. Our first discovery is a strong positive relationship between the package's functionality and performance. Thus, it may be the case that the respondents' understanding of survey questions related to functionality includes nonfunctional requirements such as performance. Also, the correlation between the reputation of the Author and that of the package is high, and the correlation between familiarity with the package and the Author's reputation is also high. This appears to reflect the dual reasons to select *tidyr*. We also find that the preferences for the watcher, fork, and star ratings of the project are correlated, presumably reflecting the respondents' perceptions that they all measure the same dimension. Furthermore, familiarity with the package correlates with package compatibility. Surprisingly, package recommendation negatively correlates with the Author's reputation and compatibility. This may reflect that we are observing two populations of users: some focus on package recommendations, while others focus more on the Author's reputation and package compatibility. Other significant correlations among the variables are either mild or appear to indicate no relationship. In the next section, we perform a logistic regression analysis to model the chances that a respondent would select *tidyr* (vs *data.table*).

TABLE II: Logistic Regression Model

SCT Category	**Factors**	**Estimate**	**Std. Error**	**z value**	**Pr(> $\|z\|$)**
	(Intercept)	0.27	0.13	2.07	0.04
Infectiousness	Resolution	0.09	0.08	1.14	0.25
	Backlog	0.01	0.14	0.10	0.92
	AuthorRep	0.35	0.08	4.63	0.00
Susceptibility	SizeProject	-0.12	0.07	-1.59	0.11
	Developers	-0.03	0.11	-0.26	0.80
	Performance	-0.53	0.06	-8.56	0.00
	Familiarity	0.36	0.07	5.37	0.00
Exposure	StackExchange	-0.20	0.06	-3.10	0.00
	Growth	0.12	0.08	1.48	0.14
	PkgRep	0.15	0.07	2.03	0.04
	StarsGH	-0.05	0.13	-0.40	0.69
	WatchersGH	0.13	0.20	0.65	0.52
	ForksGH	-0.27	0.17	-1.56	0.12

C. Regression Analysis

We use logistic regression to differentiate between these populations of respondents who use *tidyr* and *data.table*. Table-II presents the results of the analysis. Each coefficient shows to what extent that predictor had influenced the respondents' choice of *tidyr*. We use the glm (generalized linear model) function in R to perform the calculations. Each one-unit change in *PkgRpt* (Package Reputation) will increase the log odds of *tidyr* getting selected by 0.14936, and its p-value indicates that it is borderline significant $p - value = 0.04$. Also, each unit increase in AuthRep increases the log odds of getting a *tidyr* selection by 0.35, and the p-value indicates that it is statistically significant. McFaddens pseudo-R^2 is 0.2, which is considered an "excellent" fit by [20], even though pseudo-R^2 has a number of limitations [3].

AuthorRep, Performance, Familiarity, and StackExchange show statistical significance (p-values are below 0.01), and PkgRep has p-values below 0.05. Regression analysis quantifies the extent of statistically significant differences between the priorities of the developers choosing *tidyr* and *data.table*. Specifically, along the infectiousness dimension, respondents who chose *tidyr* have statistically significant stronger preference for AuthorRep than respondents choosing *data.table*. Along the susceptibility dimension, *tidyr* users preferred familiarity, while *data.table* preferred performance. Along the exposure dimension, *data.table* users preferred StackExchange questions, while package reputation was more important for *tidyr* users. None of the other predictors shows statistically significant differences between these two groups of users.

Next, we discuss the findings of our work and illustrate its impact on packages' adoption.

VI. Discussion

Our results are a step towards getting a clearer picture of the criteria used to select packages. We find some criteria are not easily visible to developers. Creating tools that make such criteria easier to gauge could benefit the community and reduce the hurdle to assess the suitability of packages.

A. Reason for the Package Selection

We examined the reasons behind the selection of one of the two packages by the respondents in their software development or data analysis work when "others" is selected. The analysis of the responses indicates that the respondents' primary concern is the limitation imposed by *data.frame*, which is the default option for creating tabular data: the core concept used in most of the R statistical modeling tools. When selecting packages, the survey respondents sought performance, reputation, compatibility, and reusability with other packages in their software stack. Traditional measures such as ranking or number of stars were not deemed essential criteria.

B. Criteria for Prioritizing the Packages

Our study examined criteria for prioritizing packages based on their observable attributes. The results showed that computer performance, helpful discussions on StackExchange forums, and familiarity with other packages were among the top criteria. However, survey respondents deemed criteria such as the number of stars, forks, and watchers irrelevant. For *data.table*, performance is the most important, followed by its reputation on StackExchange or similar forums. For the *tidy* package, the highest priority is familiarity, followed by the author's reputation, since this package is useful when developers need to develop scalable code that others can readily understand. Users may require different levels of support, and there may be a need for multiple libraries with comparable functionality but varying non-functional qualities.

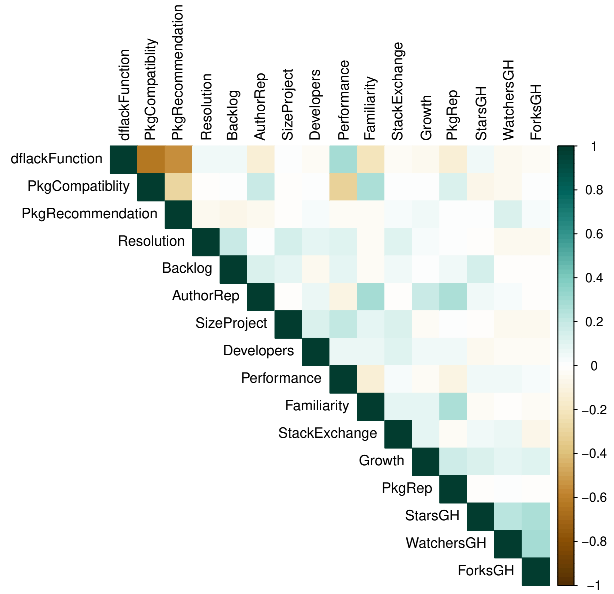


Fig. 8: Correlation matrix between influencing factors

C. How do users recommend packages?

We found the adopted respondents are likely to promote them by a large margin, especially for tidy. However, several respondents neither promote nor detract others from adopting. We found only a tiny cohort of survey respondents likely to detract others from adopting their chosen package, possibly because our survey respondents actually adopted the package in at least one of their projects.

D. Implications

Our studies provide meaningful implications for package developers, users, and the community to understand the critical parameters and priorities driving package selection. The analysis of survey responses reveals significant implications.

- Developers must focus on producing packages with improved performance, compatibility, and reusability across the software stack. This is especially crucial given the limitations of R's default data structure.
- Developers should not rely solely on traditional metrics such as stars, forks, or watchers when recommending packages. Functional, user-friendly, and well-documented products should be prioritized.
- Developers should prioritize the design of their packages to be compatible with popular packages to accommodate users' desire for familiarity.
- Users place a premium on support and community engagement when evaluating packages. So, developers should prioritize creating packages with an active user community that can provide guidance and assistance.

- When designing their packages to satisfy the diverse needs of users, developers must prioritize functionality, performance, compatibility, and community involvement.

To this end, we believe that our key findings provide a deeper implication for everyone, including developers, users, and the community, eventually improving the status quo of the open-source software ecosystem.

VII. Limitations and Threats to Validity

The survey study can be expanded to cover more packages and programming languages beyond the two repositories and one language currently considered. Hence, further work is needed to generalize the claims for other packages and programming languages. Additionally, it is possible that sampling bias may lead to consistent overestimation or underestimation of relevant parameters in the study, while a sample size of two increases the likelihood of chance observations. To minimize coding bias, neutral and objective language was used in the survey questions, and multiple response options were provided. However, it is possible that subjective judgment in descriptive responses may still occur due to a small number of coders who may have their own biases.

We selected the entire population of contributors to OSS to help with the generalizability of the results. Still, not everyone has responded to the survey, and the OSS contributors may differ from other developers. The assumptions were checked for regression and other statistical analysis, but some effects may have been unmeasured. The loss of respondents resulted in possible bias. Only individuals with public repositories were surveyed, so the generalizability to other contexts is unclear.

As it represents responses to a single survey item, the validity and reliability of any survey's NPS score ultimately rely on many responses from individual human users. Researchers have questioned whether NPS is a reliable predictor of package growth [18]. Also, studies have pointed out that there is no empirical evidence for the claim that the "likelihood to recommend" question predicts package adoption better than others (e.g, as overall satisfaction, etc.) and that it measures no different factors from other common obeying questions [24].

We highlight two types of validity threats: internal and external [7]. From the internal validity perspective, the respondents may have misinterpreted or did not fully understand some questions (e.g., what constitutes software functionality). Our survey was designed to be succinct; however, it is possible that the respondents derived multiple inferences. We expect that using the theoretical framework of SCT reduces such a possibility. From an external validity perspective, our study yielded insightful data on how best to instruct package adoption and broader implications for their design and philosophy (e.g., package performance metric, open issues, and compatibility). Finally, while the survey reached hundreds of respondents, inviting more respondents would further bolster our findings.

VIII. Conclusion

Our work contributes to understanding what drives the adoption of OSS based on an extensive analysis of the two most widely used R packages. We evaluated the developer behavior in package selection and found that easily visible characteristics, such as popularity, were not key factors.

For the Developer Community: This study lays the foundation for research and education of the developer community about a more efficient selection of packages, libraries, or other reusable components for their development environment.

For Technology Integration Professionals: Selecting the appropriate package is a monumental challenge when developing large-scale deployment and development applications. This work provides key information on the gaps, stability, scaling, and usage of packages based on requirements and scenarios that would be critical to mission-critical applications.

For Open-source Foundation Community: The findings may lead to a rethinking of the metrics of ranking open source software and would help to develop a better recommendation system for the community. Additionally, open source foundations could provide coordinated efforts toward growing a stable and collaborative community for the future development needs of open source packages and allow their effective usage.

Reliability and Validity: To ensure the reliability and validity of our survey, we considered participants' perspectives. Although the survey was conducted a few years ago, we recognized that some comments related to the questions were already seven years old at the time. To facilitate a comprehensive understanding, we included contextual information such as comments and metadata. We also included links to the actual commits to which we referred, allowing participants to revisit and investigate further. This approach allowed participants refresh their memory of the subject matter.

Our future research will prioritize evaluating the validity of claims across a wider range of packages and programming languages. Additionally, we aim to develop a comprehensive toolchain and recommendation engine to aid in the selection of packages. We will delve into the approaches employed by developers from diverse backgrounds when making package choices and analyze their impact on software adoption. We will explore how package selection criteria influence software quality metrics and shape developers' preferences.

IX. Acknowledgement

This work was supported by NSF awards 1633437, 1901102, and 2120429.

This manuscript has been authored by UT-Battelle, LLC, USA under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid up, irrevocable, worldwide license to publish or reproduce the published form of the manuscript, or allow others to do so, for U.S. Government purposes. The DOE will provide public access to these results in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

REFERENCES

[1] Bilal Amir and Paul Ralph. Poster: There is no random sampling in software engineering research. In 2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion), pages 344–345, 2018.

[2] Martin Peclat Andrea Baranzini, Stefano Carattini. What drives social contagion in the adoption of solar photovoltaic technology. GRI Working Papers 270, Grantham Research Institute on Climate Change and the Environment, July 2017.

[3] Thomas Simon Baguley. Serious stats: A guide to advanced statistics for the behavioral sciences. (Pseudo-R2 and related measures. Online Supplement). Palgrave Macmillan, 2012.

[4] Nicholas A. Christakis and James H. Fowler. Social contagion theory: examining dynamic social networks and human behavior. Statistics in Medicine, 32(4):556–577, 2013.

[5] Nicholas A Christakis and James H Fowler. Social contagion theory: examining dynamic social networks and human behavior. Statistics in medicine, 32(4):556–577, 2013.

[6] John W Creswell. Mixed-method research: Introduction and application. In Handbook of educational policy, pages 455–472. Elsevier, 1999.

[7] Campbell D and Stanley JC. Experimental and quasi-experimental designs for research. Rand McNally, 1963. (Accessed on 09/03/2022).

[8] Rafael Maiani de Mello, Pedro Correa da Silva, Per Runeson, and Guilherme Horta Travassos. Towards a framework to support large scale sampling in software engineering surveys. In Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14, New York, NY, USA, 2014. Association for Computing Machinery.

[9] Rafael Maiani de Mello and Guilherme Horta Travassos. Surveys in software engineering: Identifying representative samples. In Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '16, New York, NY, USA, 2016. Association for Computing Machinery.

[10] Alexandre Decan, Tom Mens, Maelick Claes, and Philippe Grosjean. On the development and distribution of r packages: An empirical analysis of the r ecosystem. In Proceedings of the 2015 European conference on software architecture workshops, pages 1–6, 2015.

[11] Alexandre Decan, Tom Mens, Maëlick Claes, and Philippe Grosjean. When github meets cran: An analysis of inter-repository package dependency problems. In 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), volume 1, pages 493–504. IEEE, 2016.

[12] Alexandre Decan, Tom Mens, Maëlick Claes, and Philippe Grosjean. When github meets cran: An analysis of inter-repository package dependency problems. In 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), volume 1, pages 493–504, 2016.

[13] Matt Dowle and Arun Srinivasan. data.table: Extension of 'data.frame', 2021. R package version 1.14.2.

[14] Davide Falessi, Natalia Juristo, Claes Wohlin, Burak Turhan, Jürgen Münch, Andreas Jedlitschka, and Markku Oivo. Empirical software engineering experts on the use of students and professionals in experiments. Empirical Softw. Engg., 23(1):452–489, feb 2018.

[15] Robert Feldt, Thomas Zimmermann, Gunnar R. Bergersen, Davide Falessi, Andreas Jedlitschka, Natalia Juristo, Jürgen Münch, Markku Oivo, Per Runeson, Martin Shepperd, Dag I. SjØberg, and Burak Turhan. Four commentaries on the use of students and professionals in empirical software engineering experiments. Empirical Softw. Engg., 23(6):3801–3820, dec 2018.

[16] X. Franch and J.P. Carvallo. Using quality models in software package selection. IEEE Software, 20(1):34–41, 2003.

[17] Rajdeep Grewal, Gary L. Lilien, and Girish Mallapragada. Location, location, location: How network embeddedness affects project success in open source systems. Management Science, 52(7):1043–1056, 2006.

[18] Bob E. Hayes. The true test of loyalty. Quality Engineering, 54:53–54, 2009.

[19] Genevieve Hayes. What does a data scientist really look like? - kdnuggets. https://www.kdnuggets.com/2018/11/data-scientist-look-like.html, 06 2018. (Accessed on 05/03/2022).

[20] David A. Hensher and Peter R. Stopher. Behavioural travel modelling. Behavioural Travel Modelling, pages 1–871, 5 2021.

[21] Maximillian H K Hesselbarth, Jakub Nowosad, Johannes Signer, and Laura J Graham. Open-source tools in r for landscape ecology. Current Landscape Ecology Reports, 6:97–111, 2021.

[22] Anil S Jadhav and Rajendra M Sonar. Evaluating and selecting software packages: A review. Information and Software Technology, 51:555–563, 2009.

[23] Rocío Joo, Matthew E Boone, Thomas A Clay, Samantha C Patrick, Susana Clusella-Trullas, and Mathieu Basille. Navigating through the r packages for movement. Journal of Animal Ecology, 89:248–267, 2020.

[24] Timothy L. Keiningham, Bruce Cooil, Tor Wallin Andreassen, and Lerzan Aksoy. A longitudinal examination of net promoter and firm revenue growth. Journal of Marketing, 71(3):39–51, 2007.

[25] David J Langley, Tammo HA Bijmolt, J Roland Ortt, and Nico Pals. Determinants of social contagion during new product adoption. Journal of Product Innovation Management, 29(4):623–638, 2012.

[26] Enrique Larios Vargas, Maurício Aniche, Christoph Treude, Magiel Bruntink, and Georgios Gousios. Selecting Third-Party Libraries: The Practitioners' Perspective, page 245–256. Association for Computing Machinery, New York, NY, USA, 2020.

[27] Jeff Leek. Simply statistics: How i decide when to trust an r package. https://simplystatistics.org/posts/2015-11-06-how-i-decide-when-to-trust-an-r-package/, 12 2015. (Accessed on 04/18/2022).

[28] Valentina Lenarduzzi, Oscar Dieste, Davide Fucci, and Sira Vegas. Towards a methodology for participant selection in software engineering experiments. In Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). ACM, oct 2021.

[29] Jwen Fai Low, Tennom Yathog, and Davor Svetinovic. Software analytics study of open-source system survivability through social contagion. In 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pages 1213–1217, 2015.

[30] Jwen Fai Low, Tennom Yathog, and Davor Svetinovic. Software analytics study of open-source system survivability through social contagion. In 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pages 1213–1217. IEEE, 2015.

[31] Yuxing Ma, Chris Bogart, Sadika Amreen, Russell Zaretzki, and Audris Mockus. World of code: An infrastructure for mining the universe of open source vcs data. In Proceedings of the 16th International Conference on Mining Software Repositories, MSR '19, page 143–154. IEEE Press, 2019.

[32] Yuxing Ma, Audris Mockus, Russel Zaretzki, Randy Bradley, and Bogdan Bichescu. A methodology for analyzing uptake of software technologies among developers. IEEE Transactions on Software Engineering, 48(2):485–501, 2022.

[33] Yuxing Ma, Audris Mockus, Russell Zaretzki, Bogdan Bichescu, and Randy Bradley. A methodology for analyzing uptake of softwaretechnologies among developers. IEEE Transactions on Software Engineering, 2020.

[34] Jefferson Seide Molléri, Kai Petersen, and Emilia Mendes. Survey guidelines in software engineering: An annotated review. In Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '16, New York, NY, USA, 2016. Association for Computing Machinery.

[35] Ali R Montazemi, David A Cameron, and Kalyan Moy Gupta. An empirical study of factors affecting software package selection. Journal of Management Information Systems, 13:89–105, 6 1996. doi: 10.1080/07421222.1996.11518113.

[36] Robert A Muenchen. The popularity of data analysis software. URL http://r4stats. com/popularity, 2012.

[37] Karl Pearson. Note on Regression and Inheritance in the Case of Two Parents. Proceedings of the Royal Society of London Series I, 58:240–242, January 1895.

[38] Peter Li. packageRank: Computation and Visualization of Package Download Counts and Percentiles, 2020. R package version 0.3.5.

[39] Qualtrics. How to calculate net promoter score (nps) in 2023. https://www.qualtrics.com/experience-management/customer/measure-nps/, 5 2018. (Accessed on 07/07/2023).

[40] Frederick Reichheld. The one number you need to grow. Harvard business review, 81:46–54, 124, 1 2004.

[41] Joseph Rickert. What makes a great r package? - rstudio. https://www.rstudio.com/resources/rstudioconf-2018/what-makes-a-great-r-package-joseph-rickert/, 2 2018. (Accessed on 04/18/2022).

[42] Gianluca Roveda. Mining git based software repositories. 2018.

[43] Crowe S, Cresswell K, Robertson A, Huby G, Avery A, and Sheikh A. The case study approach. BMC Medical Research Methodology, 2011.

[44] Iflaah Salman, Ayse Tosun Misirli, and Natalia Juristo. Are students representatives of professionals in software engineering experiments? In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, volume 1, pages 666–676, 2015.

[45] William R Shadish. Experimental and quasi-experimental designs for generalized causal inference. Wadsworth Cengage Learning, Belmont, CA, 2002.

[46] Zakariyah Shoroye, Waheeb Yaqub, Azhar Ahmed Mohammed, Zeyar Aung, and Davor Svetinovic. Exploring social contagion in open-source communities by mining software repositories. pages 120–127. Springer International Publishing, 2015.

[47] Zakariyah Shoroye, Waheeb Yaqub, Azhar Ahmed Mohammed, Zeyar Aung, and Davor Svetinovic. Exploring social contagion in open-source communities by mining software repositories. In International Conference on Neural Information Processing, pages 120–127. Springer, 2015.

[48] D. Spencer and J.J. Garrett. Card Sorting: Designing Usable Categories. Rosenfeld Media, 2009.

[49] Kathryn T. Stolee and Sebastian 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, New York, NY, USA, 2010. Association for Computing Machinery.

[50] Omkarprasad S. Vaidya and Sushil Kumar. Analytic hierarchy process: An overview of applications. European Journal of Operational Research, 169:1–29, 2 2006.

[51] David Waldron. R packages: dplyr vs data.table - david waldron. https://www.waldrn.com/dplyr-vs-data-table/, 11 2018. (Accessed on 07/07/2023).

[52] Dieter Welzel and Hans-Ludwig Hausen. A five step method for metric-based software evaluation — effective software metrication with respect to quality standards. Microprocessing and Microprogramming, 39:273–276, 1993.

[53] Caroline J. Wendt and G. Brooke Anderson. Ten simple rules for finding and selecting R packages. PLOS Computational Biology, 18(3):e1009884, 2022.

[54] Hadley Wickham. R packages: organize, test, document, and share your code. " O'Reilly Media, Inc.", 2015.

[55] Hadley Wickham. tidyr: Tidy Messy Data, 2021. R package version 1.1.4.

[56] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, et al. Welcome to the tidyverse. Journal of open source software, 4(43):1686, 2019.

[57] Guangchuang Yu. dlstats: Download Stats of R Packages, 2019. R package version 0.1.3.

[58] zippia. Data scientist demographics and statistics [2022]: Number of data scientists in the us. https://www.zippia.com/data-scientist-jobs/demographics/, 01 2021. (Accessed on 05/03/2022).