Noname manuscript No.
(will be inserted by the editor)

Selecting Third-party Libraries: The Data Scientist's Perspective

Sarah Nadi · Nourhan Sakr

Received: date / Accepted: date

Abstract With the increased reliance on data-driven decisions and software services, data scientists are becoming an integral part of many software teams and enterprise operations. To perform their tasks, data scientists rely on various third-party libraries (e.g., pandas in Python for data wrangling or ggplot in R for data visualization). Selecting the right library to use is often a difficult task, with many factors influencing this selection. While there has been a lot of research on the factors that software developers take into account when selecting a library, it is not clear if these factors influence data scientists' library selection in the same way, especially given several differences between both groups. To address this gap, we replicate a recent survey of library selection factors, but target data scientists instead of software developers. Our survey of 90 participants shows that data scientists consider several factors when selecting libraries to use, with technical factors such as the usability of the library, fit for purpose, and documentation being the three highest influencing factors. Additionally, we find that there are 11 factors that data scientists rate differently than software developers. For example, data scientists are influenced more by the collective experience of the community but less by the library's security or license. We also uncover new factors that influence data scientists' library selection, such as the statistical rigor of the library. We triangulate our survey results with feedback from five focus groups involving 18 additional data science experts with various roles, whose input allow us to further interpret our survey results. We discuss the implications of our findings for data science library maintainers as well as researchers who want to design recommender and/or comparison systems that help data scientists with library selection.

University of Alberta
Edmonton, AB, Canada
E-mail: nadi@ualberta.ca

The American University in Cairo
Cairo, Egypt
E-mail: n.sakr@columbia.edu

1 Introduction

Data is currently considered the world's most valuable resource [1]. As such, there has been a recent boom in the need for data scientists who are now an integral part of software teams [2] and general enterprise operations [3]. *Data scientists* are those who work with tremendous amounts of data in order to be able to work on the "real" problem [4], be it a core functionality of a product (e.g. Facebook's "People you may know") or a business decision (e.g. where to allocate inventory across Amazon's warehouses). Even in the world of software engineering, mining software repositories and software analytics have become essential to improving the quality and maintenance of software, for instance, as done by Microsoft via insights gathered from software history, telemetry, and other related artifacts [5].

With this growing demand for data science and analytics, various tools and technologies have been developed to support the tasks performed by data scientists, e.g. *data wrangling, data modeling,* and *data visualization.* In particular, *third-party libraries* (e.g. *pandas* [6], *tensorflow* [7] in Python or *ggplot* [8], *dplyr* [9] in R) provide ready-to-use functions and statistical tests common in data science work. For each functionality, a variety of software libraries may exist (e.g. in Python, *matplotlib or seaborn* may be used for data visualization), thereby making library selection for data scientists practically challenging in the same way it challenges software developers [10, 11]. Moreover, several recent articles and blogs [12–14] suggest that data scientists need support when selecting software libraries for their programming tasks and comparing the capabilities of several libraries that support the same task. However, sifting through all these articles to figure out which library to use is time consuming and requires manually comparing multiple opinions and information.

On the other hand, there is a lot of existing work on techniques and support tools that help software developers select third-party libraries [15–18] or that directly recommend a library to them [19]. Most of these support tools rely on comparing or recommending libraries based on selection factors (e.g., popularity, release frequency, or documentation) found in the software engineering literature [10, 20–22]. Analogically, data scientists would benefit from similar support systems that help them compare libraries or directly recommend the most-suited library to accomplish a task. However, the types of projects and processes data scientists follow are often different from those in traditional software development [23, 24]. For example, as opposed to software developers who often have requirements to follow in order to build a defined system, data scientists typically perform iterative exploration of the data [25]. Additionally, data scientists are usually trained on statistics and machine learning to enable them to practically train models, as opposed to software engineers whose education focuses on engineering trade offs and the construction and deployment of complete systems [26]. Therefore, we cannot assume that the previously proposed library selection factors for software developers influence data scientists' library selection in the same way. Thus, before designing library comparison, selection, or recommender systems for this population, we first need to assess which factors influence data scientists as they select their libraries.

To the best of our knowledge, there are no existing surveys on the factors that influence a data scientist's selection of a third-party library. As such, *our goal in this work is to understand which factors influence data scientists when selecting software libraries and determine if data scientists rate these selection factors differently from software developers.* Determining the influence of selection factors is a first step in providing support tools for library selection for data scientists.

To accomplish our goal, we replicate a recent FSE 2020 survey by Larios Vargas et al. [10] which asked software developers to rate the influence of 26 different library selection factors. These 26 library selection factors were collected from the literature as well as an initial interview study with 16 developers, and thus represent the most recent comprehensive list of library selection factors. To serve our goal, we shift the survey target population to data scientists. We use the same 26 technical, human, and economical factors presented in the original survey, but add additional background, experience and demographic questions to gain more context about our data scientist participants.

We survey 90 data scientists from across 21 countries. We then run five focus groups with 18 additional participants to triangulate our survey results and help us provide additional explanations from data science practitioners. We find that the usability of a library, its documentation, and its fit for the purpose of a task are the top three factors that influence data scientists when selecting libraries. Additionally, we discover three new factors specific to data scientists, including the statistical rigor of a library. Our results also reveal that there are several differences between data scientists and software developers when it comes to selecting libraries. Specifically, data scientists tend to value human factors more, where the activeness of the library's community and the experience of the community in using the library influence data scientists *more* than software developers. Our focus group participants explain that community support is essential since most data scientists at the beginning of their career heavily rely on word of mouth and peer advice, and even often fear playing around with a library or an API beyond existing code snippets they already see. On the other hand, some technical factors (e.g. whether the library is actively maintained, its security, or its maturity and stability) influence data scientists *less* than software developers when selecting their libraries. Our results have several implications for the maintainers of data science libraries as well as researchers who want to design decision support or recommender systems to help data scientists select third-party libraries, as we discuss in Section 6.

To summarize, this paper has the following contributions.

- Ratings from 90 data scientists from across 21 countries on how 26 previously established library selection factors influence their decisions to use a library.

- Statistical evidence that data scientists perceive the influence of 11 of these factors differently from software developers.
- New selection factors specific to data scientists, such as the statistical rigor of a library.
- List of the Python and R libraries that data scientists (commonly) use in their tasks, which can be used as a starting point for support tooling.
- Feedback and additional interpretation of our survey results from 18 additional data science practitioners through five focus groups.
- A discussion of the implications of our results for maintainers of data science libraries and for researchers designing library recommendation systems for data scientists.

All our data and analysis code is available on our online artifact page [27].

2 Related Work

To clarify our motivation and goals, we start by explaining the related literature. We divide related work into five parts: (1) work related to understanding data scientists and how they work, (3) work related to how software developers compare libraries, (4) work that explains differences between developers and data scientists, and (5) work that specifically focuses on library selection by data scientists.

Empirical Studies on Data Scientists Through studies on the background and work practices of data scientists [23,28,29], the literature mostly discusses the diverse backgrounds and career paths of data scientists, as well as the typical phases of a data science pipeline (e.g. data collection, data exploration, and visualization) [30].

There has also been emphasis on data scientists within the software engineering world. Most notably, Kim et al. [2] conduct an interview study with 16 participants from Microsoft to understand the emerging role of data scientists within software engineering teams. In this initial study, the authors focus on understanding why data scientists are needed in software engineering teams, their educational and training background, the type of work they do, and their working style. As follow up work, Kim et al. survey 793 professionals working at Microsoft to further investigate the role of data scientists on software teams [31]. One of the challenges they highlight is that data scientists are often frustrated by having too many (incompatible) tools to deal with. While the participants refer to tools from different languages and paradigms, such as Excel, R, and Python, the struggle extends to selecting a library or framework to use, as suggested by several online blogs [13,14,32].

In this work, we are motivated to understand the factors that influence a data scientist's decision to select a software library. Unlike Kim et al., our target population includes *all* professional data scientists, regardless of whether they are involved within a larger software engineering team or not. Today's data-driven society leverages data science to drive business decisions in a variety of fields, such as in banking or marketing [3]. Therefore, for our purposes of understanding the factors that influence library selection, we need to include all data scientists, regardless of the type of team they work in.

Library Comparison/Selection for Software Developers Extensive research studies the problem of library selection for software developers. This includes work on mining library migrations or replacements, where some researchers focus only on mining replacements from commit history [15, 33, 34], while others use such data to create decision support tools that recommend alternative libraries or versions [35, 36]. Recently, researchers have been studying how to identify and present developers with information about the factors they may care about when comparing libraries. For example, Uddin et al. [16] mine opinions about various library Application Programming Interfaces (APIs) from Stack Overflow [37] posts and categorize them based on various aspects such as documentation, usability, and security. Pano et al. [21] interviewed 18 decision makers regarding the JavaScript framework selection and derived a model of desirable Javascript framework adoption factors. Their factors are grouped into five categories: performance expectancy, effort expectancy, social influence, collegial advice, facilitating conditions, and price value.

Lopez de la Mora and Nadi [11, 18] propose a metric-based comparison of software libraries. They use the literature to identify which aspects developers may care about when using a software library and then define 10 metrics to measure these aspects. They mine library meta data from version control history, issue tracking systems, and Stack Overflow to calculate the metrics and then survey 61 developers to understand which of the proposed metrics influence the developer's selection of a library. Their study identifies popularity, security, and performance as the top influential metrics and additionally highlights that the influence of the proposed metrics vary by domain.

These results inspire the premise for our work: If the factors that influence library selection for the same target population vary by domain, they may also vary by target population, especially given the differences between data scientists and software developers (discussed next). Our motivation to eventually design decision support tools for professional data scientists drives the need for reconstructing similar studies/surveys in order to better understand what influences their library selection process. Therefore, we replicate the recent survey constructed by Larios Vargas et al. [10] and which identifies 26 technical, human and economical factors that may influence the developer's selection of a software library. In our work, we target professional data scientists instead of software developers. Given that our work uses Larios Vargas et al.'s survey questions, we separately elaborate on their study and outline the 26 factors at the end of this section.

Differences between Software Developers and Data Scientists While data scientists may have some commonality with software developers, there are several considerations that make these two populations different. From an education perspective, Kross and Guo [24] explain that those who pursue data science courses tend to be from varying ages and from a variety of academic backgrounds. For example, in a survey of 250 data scientists from across the globe, Harris et al. found that 40% of reported undergraduate degrees are in social or physical sciences, not related to computer science, math, statistics, or engineering [29]. Kross and Guo [24] also highlight how data scientists lie between two extremes: "*They share similarities with both software engineers (they aspire to write reusable analysis code to share with their colleagues) and end-user programmers (they view coding as a means to an end to gain insights from data).*" This unique placement motivates us to investigate whether the library selection problem for data scientists is different from that for software developers.

Zhang et al. [23] explore collaboration practices between data scientists and find that their communication focuses on data exploration and exchanging insights, whereas software developers work on a common code base and communicate mainly about issues related to that code. Moreover, data science projects involve more uncertainty and exploration than typical requirements driven software development, even if they will end up deployed as part of a larger software system. Studies of data science code and pipelines also show that data science code is mostly a linear orchestration of libraries for data manipulation [30,38] and that there is low modularization in terms of separation of concerns of the different stages in a typical pipeline (e.g., data preparation, data analysis etc.) [30]. In contrast, software engineers typically build large systems where they are encouraged to follow software engineering principles and often require complex intertwining of many libraries [39]. All the above differences, whether in terms of background, practices, or expectations, may affect the factors which data scientists typically look for when selecting libraries.

Library Selection for Data Scientists Ma et al. [40] propose a methodology for measuring the uptake of any software package/library by developers, based on supply chain networks and social contagion theory. As a case study, the authors apply their framework to two competing data frame implementations in R, `tidy` and `data.table`. While these are two packages that data scientists commonly use and were chosen due to the authors' familiarity with R, the paper does not specifically target data scientists overall and does not investigate *their* choices for selecting a package/library. Instead, our work focuses on directly getting data scientists' overall opinions and perceptions through a survey, allowing us to capture influencing factors that may not be directly measurable in the final uptake of a library. As part of their discussion, Ma et al. speculate that many of the influencing factors they find for these two packages seem to reflect social or cognitive biases, which further motivates us to directly survey data scientists beyond only these two R packages. We also note that all the metrics proposed by Ma et al. measure factors that are already included in the 26 factors in Larios Vargas et al.'s survey [10].

With the exception of research articles [12,41,42] or blog posts [13,14,32] that compare the performance and capabilities of machine learning or data science libraries for various tasks, we are not aware of any work that specifically looks into which factors influence data scientists' selection of a third-party library. These articles do however motivate the need for our work, since they highlight the abundance of available libraries data scientists need to compare and choose from.

The 26 Library Selection Factors Larios Vargas et al. [10] study which factors influence the software developer's decision when selecting third-party libraries. The authors follow a thorough and systematic process to reach a list of 26 factors. They first conduct an interview study with 16 practitioners (with *Software Developer* or *Software Engineer* in their job title/role) to identify 26 technical, human, and economical factors (18 of which were also supported by existing literature [11,18,20–22,34,43–45]), making their list the most comprehensive list of library selection factors to date and thus an appropriate starting point for our research. The authors then survey 116 practitioners to understand the influence of these factors on a larger scale. Their survey respondents are technical leads or software architects (36%), software developers (42%), researchers (11%), and project or product managers (5%), but do not include any data scientists.

Inspired by their work and the previously mentioned differences between software developers and data scientists, our paper surveys professional data scientists using the same 26 factors to understand which factors influence their selection of a third-party library, as well as whether there is a difference between data scientists and software developers in this regard.

The 26 factors are shown in Table 2, and we provide their brief descriptions below. In the table, we also use the same categorization Larios Vargas et al. [10] use to group these factors: **Technical Factors (T)** include factors related to functionality, quality, type of project, and release process. **Human Factors(H)** relate to stakeholders, organization, the individual, and the community. **Economic Factors (E)** relate to total cost of ownership and risk of a project. The table also shows the subcategories they use.

Technical Factors:

T1 **Brown- or green-field**: Whether the target project or software system is a new project (green field) or an existing project (brown field).

T2 **Size and complexity**: "The amount of code that the library has and whether the library offers way more functionalities than the ones needed."

T3 **Fit for purpose**: How much the purpose of the library matches with the needed requirements.

T4 **Alignment w/ architecture**: Whether these is an alignment between the library and the core technologies used in the target system or a good match with the overall software architecture.

T5 **Usability**: How easy is it to use the library.

T6 **Documentation**: Presence of (official) documentation and information to help use the library.

T7 **Security**: The overall security of the library.
T8 **Performance**: The overall performance of the library.
T9 **Well tested**: Whether the library has good testing measures to ensure its quality.
T10 **Active maintenance**: How often the library is maintained
T11 **Maturity and stability**: Whether the library is stable for usage.
T12 **Release cycle frequency**: The release cycle frequency of the library (e.g., weekly, monthly).

Human Factors:

H1 **Customers**: The final customers of the project.
H2 **Other teams**: Other teams with the selector's organization.
H3 **Project/product managers**: The selector's project and/or product managers.
H4 **Own team**: The selector's current team.
H5 **Type of industry:** The type of industry project belongs to.
H6 **Cultures and policies**: The culture and policies of the selector's organization/company.
H7 **Management and strategy**: Role of management in technology decisions.
H8 **Self-perception**: The selector's perception of the library (i.e., personal feelings)
H9 **Experience**: The collective experience of the community (library users) with the library.
H10 **Activeness**: How active the library's community is (i.e., size, responsive).
H11 **Popularity**: The popularity of the library.

Economic Factors:

E1 **Time and budget**: The budget and time available for developing your own library.
E2 **Licence**: The license (e.g., MIT, Apache) of the library.
E3 **Risk assessment**: The possible risks that the library might introduce to your project

3 Research Methods

The goal of this work is to *understand which factors influence data scientists when selecting software libraries and to compare if they rate selection factors differently from software developers.* To accomplish these goals, we use a questionnaire survey as our main research method to answer the following research questions:

RQ1 *Which factors influence data scientists when selecting software libraries?* In order to provide any support tools that help data scientists select a library to use, we first need to understand the factors they consider when selecting libraries.

RQ2 *Is there a difference between how data scientists and software developers rate library selection factors?* We want to understand if there is a difference between how software developers and data scientists choose software libraries, in terms of the factors that influence their decision. Any differences would affect the design of support tooling provided for these two groups.

As a secondary research method, we use follow up focus groups to triangulate our survey results with expert feedback and explanations. In this section, we describe our survey design, recruitment strategy, and the tools we use to analyze the collected survey data. We also describe the focus groups that we conduct after the survey.

3.1 Survey Design

Our survey mainly collects three categories of information[1] (1) Background and data science experience, (2) Demographic Information, and (3) Factor Ratings when selecting software libraries.

To create a profile for the participants' background and data science experience, we ask about their education background, highest degree, main resource for learning data science, and their self-declared confidence in their statistics and programming skills, as two core skills related to data science [29]. We also request their current position's title, location, business sector, and responsibilities.

To build more context around the participants' data science profile, we delve into the programming languages they use (if any) and ask them to list the libraries they use in each language as well as the corresponding data science tasks they typically use each library for. All previously mentioned questions are mandatory. Note that if a participant indicates that they do not code, this ends the survey since it does not make sense for us to ask them about library selection factors in that case.

Moreover, we complement the previous information with optional demographic questions related to current employment status, the size of the current organization, as well as years of professional experience, both in general and in data science specifically. Note that this optional demographic information is asked at the end of the survey, as to avoid fatigue and/or "lazy" answers for the mandatory questions.

The core of the survey is the factor ratings, where participants rate the level of how each factor (if at all) affects their selection of a software library for their data science work. To ensure that participants understand what to consider as

[1] Note that the layout of the survey sometimes combines questions of different categories to optimize the flow of the survey. For example, we ask participants about their current role at the beginning of the factor ratings to contextualize the information, while we keep all optional demographic questions at the end. The exact survey we use is available on our artifact page [27]

a software library, we provide them with the following information and instructions on the first page of the survey "*In this research, we want to understand how Data Scientists select the libraries they use when coding data science tasks. For the purposes of this survey, a data science library is broadly defined as an application (e.g., weka), package (e.g., dplyr in R), or third-party dependency (e.g., SciPy in Python).*" In the survey, we ask participants to rate the same 26 factors used by Larios Vargas et al. [10], which are explained in Section 2 and listed in Table 2, using the same rating scale. Similar to the original survey, we group the factors by categories into several rating grids. Participants rate each factor on a four-point Likert scale, with an additional null option, i.e. *"No influence" (N), "Low influence"(L), "Moderate influence"(M), "High influence" (H) and "I do not know/Not applicable"(N/A).*

Since data scientists may also consider additional factors not covered by the original 26 factors, we also ask participants an open-ended question about any additional factors they may take into account when selecting a library. This allows us to uncover any additional factors that may be specific to a data scientist's role. A similar open-ended question also appears in the original survey by Larios Vargas et al. [10].

All our survey data is collected anonymously. However, as a token of appreciation, participants can optionally provide their email to enter a raffle of one of four $50 Amazon gift cards. Our survey was also reviewed by the research ethics offices of our respective universities.

3.1.1 Pilot Survey

Before running the actual survey, we first conduct a pilot survey with three participants to make sure the survey instructions are understandable, the factors are clear, and that the Google form we use works as expected. Two out of three pilot participants are data scientists, while the third participant is a computer scientist who uses data science in their research. We ask the pilot participants to provide us any feedback they have about the survey and we also analyze their survey responses afterwards. For one of the data science pilot participants, we additionally ask them to explain how they understood each factor and verify that it aligns with the definitions used by Larios Vargas et al. [10]. We then fix the survey as needed accordingly to the pilot feedback we received.

We find that none of the pilot participants seemed to have difficulty with understanding the terminology used in the survey. However, one of the pilot participants complained about the large amount of demographic data and that they would have preferred to know about this upfront. This is why our final survey design has the essential required demographic data at the beginning of the survey (e.g. education, degree, learning resource) and all other demographic data (e.g., size of company, years of experience etc) as optional at the end of the survey, which allows a participant to completely skip this part of the survey if they do not wish to provide such information. There was also feedback on the length of the survey in terms of number of pages, which is why we streamlined the order of questions to reduce the number of pages. The survey structure, described in Section 3.1, reflects the final design of the survey after incorporating this feedback. Note that we do not include the data from these three pilot participants in our results.

3.2 Survey Participant Recruitment and Data Analysis

Sampling and Recruitment. Our target population is data science professionals who write some scripts/code to accomplish their tasks. We recruit participants through a variety of sampling strategies. We start with a mix of convenience and snowball sampling approach where we reach out to data scientists in our professional circles and ask them to fill out the survey and circulate it within their teams and professional networks.

Next, we search for additional participants outside of our network using LinkedIn search. We use the search keywords *data scientist, machine learning, data analyst* and more generally *data science*, and narrow the search by looking for participants in countries we do not already have responses from. We randomly choose candidates from the returned results and either directly message them using a purchased LinkedIn subscription that provides more InMail credits[2] or simply send connection requests with the survey details in the message. This recruitment method ensures that we do not rely only on convenience sampling from our own network. Additionally, it allows to recruit participants from diverse geographical regions. Overall, we contacted 158 participants on LinkedIn and 96 replied that they would fill the survey/share the link with their team (61% response rate). However, we do not track identifying information to verify that they actually filled it out (or if those who have not responded filled it out); we also do not know who they may have shared it with. Since we asked everyone we contacted (whether on LinkedIn or through convenience sampling) to circulate the survey link to other members in their team, which we cannot keep track of, we are unable to report an overall response rate [46].

Cleaning Responses. Overall, we received 432 responses. We exclude 7 of these responses, where the participants indicate that they do not code. We intentionally did not post on our social media accounts to avoid non-serious entries that are solely motivated by the raffle and to avoid getting responses from non-professional participants (e.g., students). This strategy worked well based on the quality of our first 22 responses, which we slowly collected over approximately 1 month. Nonetheless, one of our contacts with a wide data science network posted the survey on their social media account while specifically mentioning the raffle. This caused an extremely high number of entries to immediately trickle in. For example, we received 82 new responses within a span of 3 hours and 328 responses in 3 days. Such a high response within an extremely short period of time is atypical and suggests an issue with the quality

[2] `https://www.linkedin.com/help/linkedin/answer/1584/inmail-messages?lang=en`

of these responses. To overcome this, the two authors of this paper manually analyzed each entry to decide on whether to include it in the survey. We used two objective indicators to identify non-serious entries. The first is library names that are obviously fictional(e.g., “George Peobody” or “Intelligent library”). The second is obvious filler ratings, such as all “Low” or all “High” for all ratings. The two authors discuss any disagreements or borderline cases and resolve them. Our individual ratings have an almost perfect kappa score of 0.91, suggesting that it was easy to determine fake or filler responses from legitimate ones. We exclude 335 participants that match the two indicators of non-serious entries above. At the end, we include only 90 responses, and use these responses to answer our research questions.

Analysis Tools. To answer our research questions, we use Python to conduct our data analysis. We use SciPy [47] for our statistical tests, specifically Wilcoxon rank-sum test to compare distributions [48].To generate charts from our data, we use Python’s matplotlib library [49].

3.3 Focus Groups

Interpreting the findings from quantitative survey questions may require researcher speculation without access to follow up with participants (all responses are anonymous). Additionally, open-ended survey questions may suffer from ambiguity or incompleteness. For example, participants may add additional factors but not explain what they mean exactly or they may choose to skip a few optional questions. Thus, triangulating the results of our survey analysis with followup direct commentaries and feedback from data science experts and market practitioners can provide more insights and interpretations of the survey results. This provides a more comprehensive picture of the data scientist’s perspective on selecting libraries.

Therefore, we use the focus group method, which has been advocated as a quick and effective way to collect qualitative feedback [50], to gather additional insights to help interpret our survey results. We design the focus groups as semi-structured group conversations where we discuss our survey results and interpretations with data science experts. It is worth noting that we design the focus group questions and recruitment criteria based on the results of the survey since our goal is to use the focus groups to help us with interpretation and to avoid any researcher speculation. However, for better flow of the paper, we describe the focus groups as part of our research methods before presenting the survey results.

We select professionals from the top three business sectors in Figure 1b. We first choose well-known organizations in the respective sectors, diversifying between major corporations and smaller startups. In particular, we choose a well-known software engineering company to represent the *Software & Analytics* sector, a multinational telecommunication company and an EdTech

Table 1: Focus group participants

Group	Size	Participant Description
G1	3	Data science team at a multinational SE company
G2	4	Data science team from data science startup
G3	2	Machine learning team from same data science startup
G4	5	Data science team at an international telecommunication company (n=3) + AI team at an education technology startup (n=2)
G5	4	Data science team at an international bank

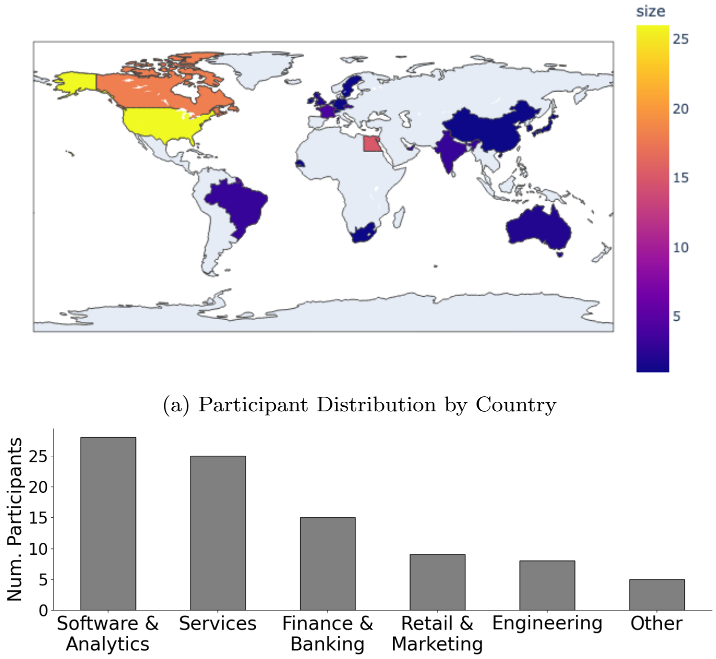
(education technology) startup to represent *Service* and, finally, an international bank to represent *Finance & Banking*. We also recruit a data science consultancy startup that provides consultation services (service branch) as well as two data science products (product branch). Adding this startup to our pool ensures exposure to professionals who perform data science tasks for diverse industries and use cases from both the service or analytics side, as well as the product engineering or operationalization side.

For each company, we ask the manager of the data science department (or equivalent) to refer us to three to five volunteers from their team with at least three years of experience and preferably diverse skills, backgrounds, and/or job titles. At the data science consultancy, we request participants from both the services branch and the product branch. Finally, we ask all nominees to confirm that they had not filled out our original survey to avoid double counting. Participating in these focus groups was voluntary with the consent that the data will remain confidential and will be reported only in aggregation. Overall, we conduct five focus groups with a total of 18 participants, shown in Table 1. We conduct each focus group as a semi-structured, unbiased conversation for 60-75 minutes addressing the following main points of discussion:

- Data Science roles and titles in the market.
- Verification of understanding of the 26 factors used in our survey.
- Library selection factors for data scientists.
- Differences between software developers and data scientists in terms of use cases and workflow.
- Perception of library selection support tools for data scientists (i.e., our motivation for this work).
- Commentary on survey results we need additional feedback or insights for.

4 Survey Results

Our survey results are based on the responses of 90 participants that match the filtering criteria discussed in Section 3.2. We first describe our participant profiles and then use the collected data to answer our research questions. We use our discussions through the focus groups to provide additional interpretation and context while presenting the survey results.



(a) Participant Distribution by Country

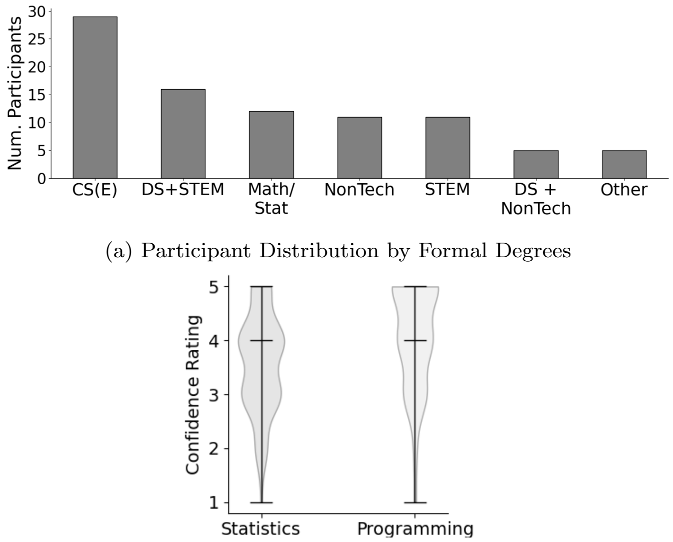
(b) Participant Distribution by Business Sector

Fig. 1: Participant Location and Industry Sector

4.1 Details of Our Participants

Countries and Sectors. Our participants work in 21 countries across 6 continents, shown in Figure 1a. To facilitate visualizing sectors in Figure 1b, we group them into five main categories and an *Other* group. We group sectors such as Education, Health Care, and Telecommunication under services, and sectors such as Natural Language and Tech under Software & Analytics. Our exact categorization can be found in the analysis scripts on our artifact page [27]. The figure shows that the top three sector groups our participants work in are Software & Analytics, Services, and Finance & Banking.

Professional Experience and Employment. The number of years of general professional experience of the 86 participants who reported it varies between 1 to 36 years, with a mean and median of 8 and 6 years, respectively. On the other hand, the number of years of professional experience in data science of the 86 participants who reported it ranges from 1 to 17 years, with a mean and median of 5 and 4 years respectively. This suggests that many of our participants worked in previous professional roles before switching to data science. Out of the 85 participants who reported employment status, the majority (80 %) are employed full time with the remainder being freelancers or part-time employ-



(a) Participant Distribution by Formal Degrees

(b) Participant Confidence in Programming and Statistics

Fig. 2: Participant Education and Skills

ees. Out of the 84 participants who answer the company size question, 6 % work for companies with 1-9 employees, 27 % for companies with 10 - 99 employees, 26 % for companies with 100 - 999 employees, 36 % for companies with more than 1000 employees, while the remaining 5 % report not applicable.

Data Science Education and Skills. Almost all our participants (97 %) have at least a bachelor degree, with 63 % having a graduate degree, as well. This is consistent with the results by Harris et al. [29] and Kim et al. [2, 31] who found that post-graduate degrees, especially PhDs, contribute to the working style of data scientists, as they need to identify important questions to ask as part of their work and iterate on them. Additionally, with the recent boom of the data science market, many data scientists switched careers by taking a Master's degree in data science when their undergraduate degree was different (or sometimes irrelevant).

We show the education specialization of our participants in Figure 2a. Note that the survey allowed participants to select multiple options in order to account for those with multiple formal degrees. Thus, we group participants into the following educational profiles: (1) *CS(E)* for participants whose degrees are only in computer science/engineering, (2) *math/stats* for participants with degrees only in mathematics or statistics, (3) *STEM* for participants whose degrees are only in other STEM fields, including other engineering disciplines, and (4) *NonTech* for participants with education fields such as business or social sciences. We then create additional categories for participants with multiple degrees: (5) *DS + STEM* for participants whose degrees combine Data science with any STEM field and (6) *DS + NonTech* for participants who have formal education in data science combined with a non-technical field. Overall, the educational background of our participants varies, with 33 % having formal education *only* in computer science or computer engineering, 13 % having formal education *only* in mathematics or statistics, 12 % with a formal education background *only* in a STEM field, other than computer science and mathematics, while 12 % have a formal education background *only* in a non technical field, such as business or various social sciences. For the remaining participants, 24 % combine a data science degree with their original technical or non technical degree, while 6 % have other combinations of degrees.

Overall, our participants learned data science through a variety of resources: 46 % of our participants learned data science formally, whether through a formal degree or non-major electives at school, 46 % used independent learning resources such as online tutorials, YouTube videos, etc., and 9 % took a form of non-degree online programs or certificates.

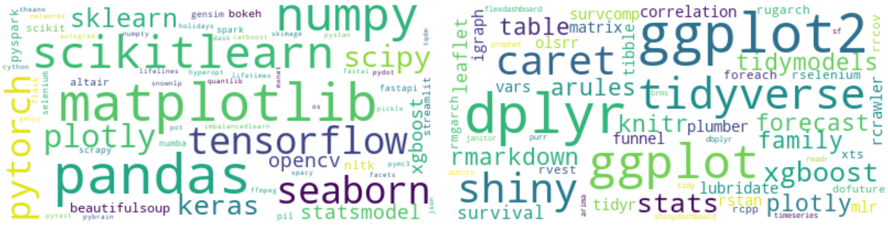
Figure 2b shows the participants' confidence in their programming and statistics skills on a five-point Likert scale (1 = Not confident at all, 5 = Very confident). As shown, the majority of our participants rated themselves highly in terms of programming skills, while the distribution of statistics skills is a bit more uniform. The median of both skills is, however, at 4 (= Confident).

The majority of our participants (83 %) use Python for their data science tasks. The other programming languages selected by our participants include R (44%), SQL (8%), and Matlab (5%). Thus, Python and R are the two most popular programming languages used by our participants, which aligns with existing findings of general data science demographic surveys [29, 51].

Data Science Role. Ignoring seniority in the job title, the majority of our participants (48 %) have explicit Data Scientist related titles (e.g., "Data Analyst" or "Data Scientist") or Machine Learning Developer/Engineer titles (16 %). While the remaining titles do not explicitly have a data science related term, the corresponding participants perform data science tasks, as indicated by the job responsibilities they describe. For example, three participants who work in HR or employment services roles use data science to analyze current or prospective employee information. Other participants had titles such as "Quantitative Researcher" or "Consultant" with concrete data analytics roles in their respective sectors. This matches the current state of the industry where data analysis and data science has become a pervasive tool used in various roles and sectors.

4.2 RQ1: Factors that influence data scientists

We now describe the findings of RQ1, specifically how the surveyed factors influence data scientists' selection of a software library. We also describe any additional factors mentioned by the participants.



(a) Python Libraries (n = 64 responses)

(b) R Libraries (n = 32 responses)

Fig. 3: Word Clouds of Commonly Used Libraries Among Participants

Library Context. To provide more context for the factor ratings, we first discuss the libraries that our participants use. We focus on Python and R for libraries since they are the two top programming languages used by our participants, as discussed in Section 4.1. Figure 3a illustrates a word cloud of all 60 unique Python libraries mentioned by 64 participants who provided an answer to this question, where bigger fonts correspond to higher frequencies. Figure 3b shows the same data for the 54 unique R libraries mentioned by 32 participants. The collected set of libraries gives us some context about the libraries that our participants had in mind as they were rating the factors.

Factor Ratings. We now discuss the rankings of the library selection factors for our data science participants. For each factor individually, we first exclude all responses that selected the *N/A / I do not know* option in order to normalize the computations to only participants who actually know and understand the factor (note that this still includes no influence ratings). Out of the remaining participants, we compute the percentage of those who rated the corresponding factor *highly*, i.e. rated the factor as being of *moderate* or *high* influence. We then rank all 26 factors based on descending order of these percentages. The rank of each factor is shown in Table 2. Finally, we consider a factor as being *important* if the percentage of respondents who rated the factor highly, as described above, is 75% or more. Based on this, we identify the following 7 important factors (also marked with an asterisk (*) beside their rankings in Table 2): (1) Usability (92 %), (2) Fit for purpose (86 %), (3) Documentation (87 %), (4) Activeness (84 %), (5) Maturity and stability (80 %), (6) Performance (80 %), and (7) Experience (76 %).

We find that the majority of these important factors are *technical factors*, mainly highlighting the high influence of the usability, functionality, and documentation of the library on the selection process. Additionally, the performance of the library is important for data scientists, which makes sense given the large amounts of data that data scientists usually deal with. Finally, data scientists value using mature and stable libraries, partially to save time on having to resolve deprecation issues or deal with poor integration issues (as supported by open-text answers from our respondents).

Two of the important factors are *human factors*. Specifically, we find that data scientists emphasize the role of the community through its experience and activeness, even more so than popularity (which is still highly valued but ranked less). Given that the majority of our respondents indicated having learned data science more or less independently, it makes sense for them to leverage the community more. Our focus groups also explain that many data scientists without any prior experience learn data science on their first job and utilize a known collection of community-based platforms for their learning. They also elaborate that Python and R have very active communities that share their experiences online, which they claim to be the main reason behind the popularity of these two programming languages among data scientists.

Finally, we find that *economical factors* are generally not among the top priorities in library selection for data scientists. This may be due to the fact that a lot of data science work is not involved in deployment or production, so licensing and budgeting is usually out of the scope of most data scientists' work and considerations.

It is also worth noting that we counted the number of *N/A's* for each factor and they were all under 11% (mostly 1 - 5%), except for one factor, namely Brown- or green-field, where around 29% of our participants did not deem it as applicable or did not even know what it was. We speculate that given less of a programming background among data scientists, it may have not been clear to our participants what Brown- or green-field even meant. We confirm this speculation by discussing it with our focus group participants. We note that some of them also did not understand this term. While other participants did understand it, they mention that this is unlikely a factor data scientists would understand and/or care about.

Additional Factors Mentioned by Participants. In our survey, we ask participants to optionally indicate any additional factors they take into account when selecting a library, which we may have not asked them about. There are 51 participants who answer this question, including 34 who indicate that there are no additional factors that they consider. We find that 10 of these responses reemphasize factors already among the 26 factors of the survey, thereby underscoring their importance. For instance, we are able to see that data scientists put much value on community support and are constantly looking for easy guidance that can reduce the time they need to learn a new library. On a technical level, they select libraries with better adaptability and integration. We now discuss the new factors mentioned by the remaining seven participants.

Table 2: Ratings for the 26 surveyed factors, comparing data scientist ratings from our survey with software developer ratings from Larios Vargas et al. [10]. Columns "H" (high influence), "L+M" (low or moderate influence), and "N" (no influence) indicate percentage of participants for each rating; we do not show the N/A percentage values, but they appear in the bar charts. Factors with statistically significant differences between data scientists and software developers are shown in blue, along with direction of difference (↑ is more for data scientists and ↓ is less). Rank column shows factor ranking according to the percentage of participants who rated the corresponding factor as moderate or high influence. Asterisk (*) marks factors where this percentage is >= 75%.

	Data scientists (N=90)					Software Developers (N=116) [10]				
	Distribution	H	L + M	N	Rank	Distribution	H	L+M	N	Rank
Technical Factors										
Software System										
Brown- or green-field		14 %	42 %	14 %	26		10%	55%	18%	19
Functionality										
Size and complexity		14 %	73 %	10 %	15		16%	71%	11%	18
Fit for purpose ↑		73 %	26 %	1 %	**3***		36%	52%	7%	11
Quality										
Alignment w/ architecture		36 %	51 %	8 %	10		21%	69%	5%	14
Usability		63 %	36 %	1 %	**1***		55%	42%	3%	**3***
Documentation		52 %	43 %	3 %	**2***		51%	47%	1%	**2***
Security ↓		21 %	62 %	16 %	18		39%	53%	4%	8
Performance		43 %	50 %	4 %	**6***		32%	63%	5%	**6***
Well tested		23 %	64 %	9 %	17		11%	68%	18%	17
Release										
Active maintenance ↓		30 %	60 %	7 %	13		44%	47%	7%	**5***
Maturity and stability ↓		42 %	49 %	8 %	**5***		62%	37%	0%	**1***
Release cycle frequency		8 %	66 %	23 %	23		3%	74%	22%	22
Human factors										
Stakeholders										
Customers ↑		12 %	48 %	30 %	25		7%	30%	54%	25
Other teams		8 %	63 %	26 %	24		4%	62%	23%	20
Project/product managers ↑		14 %	64 %	16 %	19		3%	48%	36%	26
Own team		30 %	59 %	10 %	8		30%	57%	7%	9
Organization										
Type of industry ↑		14 %	49 %	32 %	22		9%	46%	36%	23
Culture and policies ↑		14 %	53 %	26 %	20		8%	52%	28%	21
Management and strategy		13 %	53 %	23 %	21		8%	47%	35%	24
Individual										
Self-perception		29 %	61 %	10 %	12		16%	69%	12%	16
Community										
Experience ↑		43 %	50 %	6 %	**7***		23%	63%	9%	15
Activeness ↑		41 %	57 %	2 %	**4***		30%	62%	8%	13
Popularity		33 %	58 %	8 %	11		30%	59%	10%	7
Economical factors										
Total cost of ownership										
Time and budget		40 %	43 %	6 %	14		39%	48%	10%	12
License ↓		28 %	51 %	14 %	16		45%	42%	11%	10
Risk										
Risk assessment		38 %	54 %	7 %	9		38%	57%	5%	**4***

Five responses alert us to some additional considerations, which we highlight below by discussing quotes from our respondents (R). R19 mentions that it is important to consider the general completeness of the library's scientific scope. They give the example of survival libraries that must provide the p-values of a C-index, rather than only the C-index. Along the same lines, R54 lists *"trust in the statistical background of the library developer"* as an additional factor they consider. R46 says that they consider the *"consistency in [the] results of the algorithms in a[n] R library vs the corresponding Python library, especially when data science products have to be scaled up on Hadoop or Spark."* Also related to scalability, R32 lists the ability to handle large data sets. Finally, R71 mentions the extent to which a library enables the user to work on custom use cases.

There are two remaining ambiguous responses mentioning *Inertia* (R414) and *Speed to get insights* (R425) as additional factors. These respondents do not elaborate on what they exactly mean by those factors. Accordingly, we ask our focus group participants how they might interpret these comments. They suggest that inertia may refer to consistency in results produced by an algorithm if it is run multiple times, similar to the concern stated by R46 above. As for *Speed to get insights*, our focus group experts took this to either be related to how fast the library runs to produce results or how easy it is to learn and use a library to get results. They suggest that both of these explanations are already entailed within *Usability* and *Performance* from the original 26 factors and they would, therefore, not consider this as a new factor.

From the above, we conclude that data scientists draw our attention to the core role that statistics play in their work. As such, **statistical soundness** is a factor that may influence their library selection (R19 and R54). Moreover, the nature of the data scientist's work comes with some concerns about **consistency and scalablity** (R32, R46, R414) and **customization** (R71). All of these additional factors can be categorized as technical ones.

RQ1: We find 7 important factors for data scientists: 5 are technical (*usability, fit for purpose, documentation, maturity and stability and performance*), 2 are human (*activeness and experience*), while none are economical. Additionally, data scientists do consider three other factors when selecting libraries: *statistical soundness, consistency and scalablity, and customization.*

4.3 RQ2: Differences between data scientists and software engineers

In RQ2, we want to understand if there are any differences in terms of how data scientists and software developers rate the same library selection factors. Such differences may impact how support tools should be designed for these two populations.

We compare the distribution of ratings obtained by Larios Vargas et al. [10] to the distribution of ratings by our data scientist participants[3]. We use a two-sided Wilcoxon rank-sum test to compare the distributions for each factor rating, since we are interested in differences in either direction. We consider results with a `p-value` < 0.05 as statistically significant. To determine the direction of difference (i.e., whether this factor influences data scientists' selection *more* or *less* than developers), we compare the medians of the two rating distributions and break any ties using the means. Additionally, we calculate factor rankings based on the software developer ratings, following the same method we described for the ratings of data scientists in Section 4.2. We then compare the factor rankings and important factors we obtained for data scientists with those for software developers.

Table 2 shows the factor rankings for both data scientists and software developers, as well as an asterisk (*) beside factor rankings that were deemed as important. Upon a quick inspection, we can see that there are several factors that rank very differently between the two populations. For example, data scientists rank whether a project is brown or green field as the least factor influencing their decision to use a library (rank = 26), while software developers rank it higher at rank 19. On the other hand, fit for purpose is the third most influencing factor for data scientists, while it is ranked 11th for software developers.

When it comes to important factors, we find 6 important factors (i.e., those that >= 75% of participants rated as moderate or high influence) for software developers: (1) Maturity and stability (90%), (2) Documentation (89%), (3) Usability (86%), (4) Risk assessment (81%), (5) Active maintenance (80%), and (6) Performance (77%). Interestingly, all the important factors for software developers are technical or economical factors. In general, few human factors seem to make their way up the rank list for software developers. The only human factor for developers that comes in their top ten factors is own team, at rank 9. On the other hand, data scientists have two human factors (community experience and community activeness) among their 7 important factors, and three in the top ten factors. That said, both groups give fairly low ratings for the remaining stakeholders factors and all organization factors.

While these ranks provide some insights into differences, the differences we observe may not be statistically significant. Thus, we now turn to the results of the Wilcoxon rank-sum test to determine which factors have statistically significant differences between the two populations. We find that there are 11 factors where the difference between the distribution of ratings of data scientists and software developers is statistically significant. These 11 factors, highlighted in blue in Table 2 are: *fit for purpose, security, active maintenance, maturity and stability, customers, project/product managers, type of industry, community experience, community activeness, cultures and policies, and license.*

[3] Thanks to the authors for releasing their raw rating data [52], which allowed us to reproduce their results and enabled a direct distribution comparison

We now discuss the differences in directions for these 11 factors, while also including any difference in rankings. Our results show that whether a library matches the needed requirements influences data scientists' decision much *more* than software developers (rank 3 vs rank 11). It is interesting to see that for the remaining technical factors that we find statistical differences for, we always find that data scientists care *less* about these factors. Specifically, factors such as security (rank 18 vs. 8), active maintenance (rank 13 vs. 5), and maturity and stability (rank 5 vs. 1) influence data scientists' decisions *less* than they do for software developers, even though a factor such as maturity and stability is among the top most important factors for data scientists. Similarly, the specific license of a library is *less* important for data scientists when compared to software developers (rank 16 vs. 10). Our focus group discussions provide insights into interpreting these results. Specifically, our focus group participants mention that data scientists are not expected to be responsible for requirements such as *security* and *active maintenance.* They also rarely think about *licenses.* However, our experts in the focus groups indicate that a lot of client-facing data scientists do need to consider *maturity and stability* of their work. A lot of them value this factor in their own work, especially those who operate more on the production side of their respective companies and those who used to be software developers before switching to data science. Based on the survey results and focus group discussions, we conclude that even though *maturity and stability* is important for data scientists, it may still not be as important as for software developers. This makes sense since many data scientists are analysts that often do quick and "dirty" work for internal purposes, which would not require investing much thought into *maturity or stability.*

On the other hand, consistent with what we observe above, for all the human factors that we found statistically significant differences for, it is always the case that data scientists care *more* about these factors than software developers. Specifically, we find the biggest differences in community-related factors such as the community experience (rank 7 vs. 15) and community activeness (rank 4 vs. 13). Our focus groups agree that community is a major leverage for any data scientist. They elaborate that many data scientists learn data science "on the job" or through popular community platform, which matches what our survey participants indicated. Data scientists are typically familiar with a few popular active communities that they follow and trust. Note that around 70% of our focus group participants are data scientists without prior software development or programming experience. As part of our conversation, they reflected on their own experience by explaining that at the beginning of their career, they heavily relied on word of mouth and peer advice. They did not necessarily look at or understand a software library the same way a software developer may do. They even sometimes feared tinkering with the code and thus borrowed snippets as is. For them, a library is a set of APIs that they treat as black boxes. However, through community help and guidance, they were able to unravel the "mysteries" of a software library and even write ones from scratch. Nonetheless, even after becoming more experienced, they still considered the community as their number one reference whenever they look for answers. Such reflection underlines why a library's community support for is important to data scientists.

RQ2: We find the following statistically significant differences between the two populations. When compared to software developers, 7 factors influence data scientists' decision *more* (*how much a library fits the desired purpose, the final customers of the project, the project/product managers, type of industry, cultures and policies, the community experience, and how active the community is*) while 4 factors have *less* influence over data scientists' decision (*library security, its active maintenance, maturity and stability, and its license*).

5 Limitations and Threats to Validity

External Validity Our findings are based on the input of 90 participants. While this number of participants is close to that of Larios Vargas et al. [10], allowing us to compare findings, it may not be representative of the whole data science community. However, we find that the results of the latest 2020 Kaggle survey of data scientists [51] match some of the the distribution and background of our participants. For example, similar to our participants, 68% of their participants have a graduate degree. Additionally, the background, sectors, and years of experience of our participants varies considerably, suggesting that our data is not skewed towards certain sub-populations of data scientists. Nonetheless, we were limited by the number of participants in some of the sub-populations, which did not allow us to slice our data and analyze each sub-population separately.

Construct Validity Our survey measures how data scientists rate the influence of certain factors on their library selection process. It does not provide concrete metrics that measure these factors and accordingly, does not provide specific interpretations or definitions for each term. For example, community activeness could be measured/interpreted by the number of Stack Overflow questions or by how quickly issues are responded to [18, 40]. This could result in each participant rating the factor with a different measure/ interpretation in mind. However, given that we are replicating a previous survey to facilitate comparison, we did not want to alter any of the factors, including biasing the participants towards certain ways of measuring them. Additionally, our goal is not to determine the best way to measure a factor (which is an interesting future research direction none the less, with some recent efforts, including ours, in that direction [18, 40]), but rather to determine its overall influence on the selection process. The second author of this paper is also a data science researcher; we leveraged her expertise in the data science community to ensure that the terminology can still be interpreted by data scientists. Finally, to further mitigate the potential terminology misinterpretation threat, we also conducted a pilot survey to ensure that data scientists can understand the content of the survey. While the focus groups took place after the survey, we also leveraged them to double check that the factors are understandable by asking participants what they understood from each factor. All factors were clear to all participants, with the exception of brown/green field development.

We acknowledge that we use the term “data scientist” in this paper as an umbrella term for various roles related to data science, while these roles may have an impact on how a data scientist deals with libraries. For example, some online career articles differentiate between 10 different roles related to data science [53], such as data scientist, data engineer, data analysis, machine learning engineer etc. While our initial intention with having the job title and job description information in the survey was to differentiate between such roles, we found that the use of different titles by our participants is still not consistent. For example, some job titles did not even match these 10 roles while the participant provided a job description that involves data science tasks. We discuss this issue with our focus group participants and the majority believed that the market has a very fluid definition of data science roles. Since not all participants provide detailed job responsibilities that can allow us to accurately map them to the different expected job titles, we chose to avoid unsound statistical analysis based on small data slices or inaccurate qualitative analysis to differentiate or draw any conclusions related to the effect of the various roles. The same holds for differentiating data scientists by their background and experiences, and statistically determining if different backgrounds affect their factor ratings. We believe that future research can focus on understanding such differences within the data science population and their impact on library selection and usage.

Internal Validity Our choice to include a raffle with monetary compensation resulted in receiving many “nonsense” responses. We mitigated this issue through careful filtering of the responses to include only these responses that two of the authors independently deemed as legitimate. We used two objective criteria to determine valid responses. Given our almost perfect inter-rater agreement rate (kappa score of 0.91) and our decision to include only 90 high-quality responses instead of simply choosing to report a higher number of responses, we are confident in the quality of our data.

6 Significance and Implications

Our results have implications for two main stakeholders: *the developers or maintainers of data science libraries* and *researchers who plan on designing support tools for data scientists*. We discuss both below then propose potentially interesting future research directions related to data scientists’ library selection and support.

6.1 Implications for Data Science Library Maintainers

Our results show that the top three ranked factors for data scientists are usability, documentation, and fit for purpose. Thus, similar to what is known about software developers in the literature [10, 21, 54], data scientists will disregard libraries that are not well-documented or that are hard to use. Given that fit for purpose influences data scientists' selections even more so than software developers' selections, we recommend a library's documentation to clearly indicate the purpose of the library and types of data science tasks it supports. Our focus group findings also support this conclusion, and even indicate that knowing which tasks a library supports, rather than only which functionality, is important.

The new factors we find in RQ1 also indicate that statistical soundness is important, given the nature of the statistics-based work of a data scientist. Library maintainers may consider adding information about how certain calculations are performed and how the soundness of these calculations, including their consistency, is ensured. Such information may help data scientists to decide if they trust relying on a given library for their work. Further supporting our recommendation, our focus groups mention that data scientists with weak backgrounds in statistics find it hard to relate statistical concepts they read about to the libraries they are using. They find it hard to edit existing functions or libraries as they find little guidance on how to do so without violating any statistical rules. They also find it hard to try models that are less popular on the community blogs, as they have no way to validate the correctness of their work due to lacking statistical documentation.

6.2 Implications on Designing Support Tools for Data Scientists

Our findings directly impact how software engineering researchers and/or tool builders can design library selection support tools for data scientists. With the prevalence of data science in many businesses and applications, there has been recent effort in the software engineering community to design tools that support data scientists in their work. Examples include API migration support between data science libraries [55], code cleanup activities in Jupyter notebooks [56], or debugging and parameter tuning in data analysis scripts [57, 58]. Similar to such efforts, we believe that library selection decision support tools can help data scientists be more efficient in their work. Designing such support tools is one of the main motivations behind the research presented in this paper. Our discussions in the focus groups indicate that data scientists would find library selection support tools beneficial, which supports our motivation and highlights the importance of creating such tools. In particular, participants of four out of five focus groups independently indicated that one of the major struggles at their data science teams is that they do not find enough guidance on selecting the libraries that are best suited for the particular tasks or use cases they are faced with. Participants who were more involved with the engineering and production side, on the other hand, suggested that they would appreciate having a recommender system that compares and selects libraries based on factors they are interested in, similar to the ones that were investigated throughout our research. The results of our survey provide insights that can help with the design of such tools as follows.

Languages and Libraries Providing information to compare various libraries [18] or designing library recommender systems [36] often requires targeting a specific programming language. Our survey results confirm existing knowledge that while data scientists use several programming languages in their tasks, Python and R are the two top used languages. Thus, recommender systems designed for data scientists should target these two languages. Our analysis on commonly used Python and R libraries, shown in Figures 3a and 3b, provides these tool builders with empirical guidance on which libraries to address first in order to create useful tooling. Not only is this list useful for tooling related to library comparison, it is also useful for building any further support tools, such as library migration, code completion, or API recommendation.

Prioritizing Factors Displaying information about all 26 factors from the literature will likely overwhelm any end user. Similarly, a recommender considering all 26 factors may end up making useless recommendations, because it is unlikely that one library is "good" or "bad" on all factors. Thus, it is important to actually know the factors that influence the target audience's decision and either consider only these factors, or at the very least, weigh the factors differently. Our results provide rankings that can help with such a task. Specifically, tooling targeting data scientists should prioritize usability of the library, its documentation, its fit for their purpose, its community's activeness, its maturity and stability, its performance, and the collective experience of its community. Overall, data scientists seem to heavily rely on opinions and experiences of the community, as well as the overall human stakeholders in their team and organization. We find that organization factors do not highly influence library selection for data scientists, which suggests that they can be ignored or de-prioritized in any support tooling. Finally, we discover new factors that affect data scientists' library selection. Most importantly, as also confirmed from the focus group, statistical soundness and calculation consistency are important library selection factors. We recommend that these factors should be considered for any library selection support tools.

7 Conclusion

Data scientists use third-party libraries to accomplish many of their tasks. With the abundance of available libraries, it may not always be obvious which library to use. In this work, we surveyed 90 data scientists to understand which factors influence their selection of third-party libraries to use. Our results provide rankings of these factors and highlight multiple differences between software developers and data scientists when it comes to selecting libraries. Specifically, some technical factors such as security and active maintenance influence data scientists less than they influence software developers, while human factors such as the community's experience and its activeness influence data scientists more. We also triangulate our survey results with discussions with 18 data scientists through 5 separate focus groups. These discussions provide additional insights and interpretations to our survey results, and open up additional avenues for potential future work. Overall, our results and their implications help guide the design of future support tooling that can help data scientists select the most appropriate library for their needs. As future work, we plan on considering the implications in this paper to design library selection and usage support tooling for data scientists. We also recommend that future work focuses on investigating the impact of the differences among various roles and backgrounds of data scientists.

Acknowledgments

We would like to thank all our participants who filled out the survey. Thanks to Andrew Nady for help with early data analysis scripts. Sarah Nadi's research is funded by the Canada Research Chairs program.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. "The world's most valuable resource is no longer oil, but data," The Economist. https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data.
2. M. Kim, T. Zimmermann, R. DeLine, and A. Begel, "The emerging role of data scientists on software development teams," in *Proceedings of the 38th IEEE/ACM International Conference on Software Engineering (ICSE).* IEEE, 2016, pp. 96–107.
3. S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer, "Enterprise data analysis and visualization: An interview study," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2917–2926, 2012.
4. D. Patil, *Building data science teams.* O'Reilly Media, Inc., 2011.
5. J. Czerwonka, N. Nagappan, W. Schulte, and B. Murphy, "Codemine: Building a software development data analytics platform at microsoft," *IEEE software*, vol. 30, no. 4, pp. 64–71, 2013.
6. (2021) Pandas. [Online]. Available: https://pandas.pydata.org/
7. (2021) Tensorflow. [Online]. Available: https://www.tensorflow.org/
8. H. Wickham, W. Chang, T. L. P. Lionel Henry, K. Takahashi, C. Wilke, K. Woo, H. Yutani, and D. Dunnington. (2021) ggplot. [Online]. Available: https://ggplot2.tidyverse.org/
9. H. Wickham, R. François, L. Henry, and K. Müller. (2021) dplyr. [Online]. Available: https://dplyr.tidyverse.org/

10. E. Larios Vargas, M. Aniche, C. Treude, M. Bruntink, and G. Gousios, "Selecting third-party libraries: The practitioners' perspective," in *Proceedings of the 28th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering (ESEC/FSE)*. New York, NY, USA: Association for Computing Machinery, 2020, p. 245–256. [Online]. Available: https://doi.org/10.1145/3368089.3409711
11. F. L. De La Mora and S. Nadi, "Which library should i use?: A metric-based comparison of software libraries," in *Proceedings of the 40th IEEE/ACM International Conference on Software Engineering: New Ideas and Emerging Technologies Results (ICSE-NIER)*, 2018, pp. 37–40.
12. G. Nguyen, S. Dlugolinsky, M. Bobák, V. Tran, Á. L. García, I. Heredia, P. Malík, and L. Hluchỳ, "Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 77–124, 2019.
13. T. Sol. (2021) Choosing an open source machine learning library? here's the list! [Online]. Available: https://gbksoft.com/blog/choosing-an-open-source-machine-learning-library-heres-the-list/
14. S. A. Metwalli. (2020) Data visualization 101: How to choose a python plotting library. [Online]. Available: https://towardsdatascience.com/data-visualization-101-how-to-choose-a-python-plotting-library-853460a08a8a
15. C. Teyton, J.-R. Falleri, and X. Blanc, "Mining library migration graphs," in *Proceedings of the 19th Working Conference on Reverse Engineering (WCRE)*, 2012, pp. 289–298.
16. G. Uddin and F. Khomh, "Automatic summarization of API reviews," in *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '17, 2017.
17. R. El-Hajj and S. Nadi, "LibComp: An IntelliJ plugin for comparing Java libraries," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 1591–1595. [Online]. Available: https://doi.org/10.1145/3368089.3417922
18. F. L. de la Mora and S. Nadi, "An empirical study of metric-based comparisons of software libraries," in *Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering*, ser. PROMISE'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 22–31. [Online]. Available: https://doi.org/10.1145/3273934.3273937
19. F. Thung, D. Lo, and J. Lawall, "Automated library recommendation," in *Proceedings of the 20th Working Conference on Reverse Engineering (WCRE)*, Oct 2013, pp. 182–191.
20. R. Abdalkareem, O. Nourry, S. Wehaibi, S. Mujahid, and E. Shihab, "Why do developers use trivial packages? an empirical case study on npm," in *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 385–395. [Online]. Available: https://doi.org/10.1145/3106237.3106267
21. A. Pano, D. Graziotin, and P. Abrahamsson, "Factors and actors leading to the adoption of a javascript framework," *Empirical Software Engineering*, vol. 23, no. 6, pp. 3503–3534, 2018.
22. B. Xu, L. An, F. Thung, F. Khomh, and D. Lo, "Why reinventing the wheels? an empirical study on library reuse and re-implementation," *Empirical Software Engineering*, vol. 25, no. 1, pp. 755–789, 2020.
23. A. X. Zhang, M. Muller, and D. Wang, "How do data science workers collaborate? roles, workflows, and tools," *Proc. ACM Human-Computer Interaction*, vol. 4, no. CSCW1, May 2020. [Online]. Available: https://doi.org/10.1145/3392826
24. S. Kross and P. J. Guo, *Practitioners Teaching Data Science in Industry and Academia: Expectations, Workflows, and Challenges*. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–14. [Online]. Available: https://doi.org/10.1145/3290605.3300493
25. M. B. Kery, M. Radensky, M. Arya, B. E. John, and B. A. Myers, "The story in the notebook: Exploratory data science using a literate programming tool," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–11.

26. N. Nahar, S. Zhou, G. Lewis, and C. Kästner, "Collaboration challenges in building ml-enabled systems: Communication, documentation, engineering, and process," in *Proceedings of the 44th International Conference on Software Engineering (ICSE '22)*, 2022.
27. S. Nadi and N. Sakr, "Artifact for Selecting Third-party Libraries: The Data Scientist's Perspective," 9 2022. [Online]. Available: https://figshare.com/articles/dataset/Selecting_Third-party_Libraries_The_Data_Scientist_s_Perspective/16563885
28. M. Muller, I. Lange, D. Wang, D. Piorkowski, J. Tsay, Q. V. Liao, C. Dugan, and T. Erickson, "How data science workers work with data: Discovery, capture, curation, design, creation," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–15.
29. H. Harris, S. Murphy, and M. Vaisman, *Analyzing the analyzers: An introspective survey of data scientists and their work.* " O'Reilly Media, Inc.", 2013.
30. S. Biswas, M. Wardat, and H. Rajan, "The art and practice of data science pipelines: A comprehensive study of data science pipelines in theory, in-the-small, and in-the-large," *arXiv preprint arXiv:2112.01590*, 2021.
31. M. Kim, T. Zimmermann, R. DeLine, and A. Begel, "Data scientists in software teams: State of the art and challenges," *IEEE Transactions on Software Engineering*, vol. 44, no. 11, pp. 1024–1038, 2018.
32. S. Robinson. (2018) The best machine learning libraries in python. [Online]. Available: https://stackabuse.com/the-best-machine-learning-libraries-in-python/
33. C. Teyton, J.-R. Falleri, M. Palyart, and X. Blanc, "A study of library migrations in java," *J. Softw. Evol. Process*, vol. 26, no. 11, pp. 1030–1052, Nov. 2014.
34. A. Hora and M. T. Valente, "Apiwave: Keeping track of api popularity and migration," in *Proceedings of the 31st IEEE International Conference on Software Maintenance and Evolution*, ser. ICSME '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 321–323.
35. Y. M. Mileva, V. Dallmeier, M. Burger, and A. Zeller, "Mining trends of library usage," in *Proceedings of the Joint International and Annual ERCIM Workshops on Principles of Software Evolution (IWPSE) and Software Evolution (Evol) Workshops*, ser. IWPSE-Evol '09. New York, NY, USA: ACM, 2009, pp. 57–62.
36. F. Thung, D. Lo, and J. Lawall, "Automated library recommendation," in *20th Working Conference on Reverse Engineering (WCRE)*, Oct 2013, pp. 182–191.
37. (2021) Stack Overflow. [Online]. Available: https://stackoverflow.com/
38. F. Psallidas, Y. Zhu, B. Karlas, M. Interlandi, A. Floratou, K. Karanasos, W. Wu, C. Zhang, S. Krishnan, C. Curino *et al.*, "Data science through the looking glass and what we found there," *arXiv preprint arXiv:1912.09536*, 2019.
39. R. S. Pressman, *Software engineering: a practitioner's approach.* Palgrave macmillan, 2005.
40. Y. Ma, A. Mockus, R. Zaretzki, B. Bichescu, and R. Bradley, "A methodology for analyzing uptake of software technologies among developers," *IEEE Transactions on Software Engineering*, 2020.
41. I. Stančin and A. Jović, "An overview and comparison of free python libraries for data mining and big data analysis," in *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* IEEE, 2019, pp. 977–982.
42. J. Siebert, J. Groß, and C. Schroth, "A systematic review of packages for time series analysis," *Engineering Proceedings*, vol. 5, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2673-4591/5/1/22
43. A. Gizas, S. Christodoulou, and T. Papatheodorou, "Comparative evaluation of javascript frameworks," in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12 Companion. New York, NY, USA: Association for Computing Machinery, 2012, p. 513–514. [Online]. Available: https://doi.org/10.1145/2187980.2188103
44. B. A. Myers and J. Stylos, "Improving api usability," *Communications of the ACM*, vol. 59, no. 6, pp. 62–69, 2016.
45. M. Piccioni, C. A. Furia, and B. Meyer, "An empirical study of api usability," in *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2013, pp. 5–14.

46. P. Ralph, N. bin Ali, S. Baltes, D. Bianculli, J. Diaz, Y. Dittrich, N. Ernst, M. Felderer, R. Feldt, A. Filieri, B. B. N. de França, C. A. Furia, G. Gay, N. Gold, D. Graziotin, P. He, R. Hoda, N. Juristo, B. Kitchenham, V. Lenarduzzi, J. Martínez, J. Melegati, D. Mendez, T. Menzies, J. Molleri, D. Pfahl, R. Robbes, D. Russo, N. Saarimäki, F. Sarro, D. Taibi, J. Siegmund, D. Spinellis, M. Staron, K. Stol, M.-A. Storey, D. Taibi, D. Tamburri, M. Torchiano, C. Treude, B. Turhan, X. Wang, and S. Vegas, "Empirical standards for software engineering research," *arXiv preprint arXiv:2010.03525*, 2020.
47. T. S. community. (2021) SciPy library. [Online]. Available: https://www.scipy.org/
48. ——. (2021) Wilcoxon rank sum test. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ranksums.html
49. (2021) Matplotlib. [Online]. Available: https://matplotlib.org/
50. J. Kontio, L. Lehtola, and J. Bragge, "Using the focus group method in software engineering: obtaining practitioner and user experiences," in *Proceedings of the International Symposium on Empirical Software Engineering (ISESE'04)*. IEEE, 2004, pp. 271–280.
51. kaggle, "Kaggle's 2020 state of data science and machine learning survey," https://www.kaggle.com/kaggle-survey-2020, 2020.
52. E. Larios Vargas, M. Aniche, C. Treude, M. Bruntink, and G. Gousios, "Selecting third-party libraries: The practitioners' perspective," Aug. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3979446
53. (2020) What you should know about the different data science job titles. [Online]. Available: https://www.linkedin.com/pulse/what-you-should-know-different-data-science-job-big-data-scientist/
54. M. P. Robillard and R. DeLine, "A field study of API learning obstacles," *Empirical Software Engineering*, vol. 16, no. 6, pp. 703–732, 2011.
55. A. Ni, D. Ramos, A. Z. H. Yang, I. Lynce, V. Manquinho, R. Martins, and C. Le Goues, "Soar: A synthesis approach for data science api refactoring," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 2021, pp. 112–124.
56. H. Dong, S. Zhou, J. Guo, and C. Kästner, "Splitting, renaming, removing: A study of common cleaning activities in jupyter notebooks," in *Proceedings of the 9tn International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)*, 11 2021.
57. J. Hu, J. Joung, M. Jacobs, K. Z. Gajos, and M. I. Seltzer, "Improving data scientist efficiency with provenance," in *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, 2020, pp. 1086–1097.
58. C. Yang, S. Zhou, J. L. Guo, and C. Kästner, "Subtle bugs everywhere: Generating documentation for data wrangling code," in *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, vol. 11, 2021.