Evaluating the Usability of Differential Privacy Tools with Data Practitioners

Ivoline C. Ngong, Brad Stenger, Joseph P. Near, and Yuanyuan Feng,
University of Vermont

https://www.usenix.org/conference/soups2024/presentation/ngong

This paper is included in the Proceedings of the
Twentieth Symposium on Usable Privacy and Security.

August 12–13, 2024 • Philadelphia, PA, USA

978-1-939133-42-7

Evaluating the Usability of Differential Privacy Tools with Data Practitioners

Ivoline C. Ngong
University of Vermont

Brad Stenger
University of Vermont

Joseph P. Near
University of Vermont

Yuanyuan Feng
University of Vermont

Abstract

Differential privacy (DP) has become the gold standard in privacy-preserving data analytics, but implementing it in real-world datasets and systems remains challenging. Recently developed DP tools aim to make DP implementation easier, but limited research has investigated these DP tools' usability. Through a usability study with 24 US data practitioners with varying prior DP knowledge, we evaluated the usability of four open-source Python-based DP tools: DiffPrivLib, Tumult Analytics, PipelineDP, and OpenDP. Our study results suggest that these DP tools moderately support data practitioners' DP understanding and implementation; that Application Programming Interface (API) design and documentation are vital for successful DP implementation and user satisfaction. We provide evidence-based recommendations to improve DP tools' usability to broaden DP adoption.

1 Introduction

Advances in big data analytics have propelled the collection and processing of massive amounts of data, including sensitive data such as medical records, financial information, and other personally identifiable information. The analysis of this sensitive data may result in the accidental leakage of individuals' data [40, 56], even when anonymization techniques are used [15, 16, 34, 60]. Differential privacy (DP) can mitigate these risks [24, 25] by guaranteeing the results of statistical analyses will not reveal too much personal information about any individual. By adding carefully calibrated noise to data, DP protects sensitive data while still revealing high-level statistical insights. Due to its tremendous potential to revolutionize privacy-preserving data analysis, DP attracts considerable research [25]. Leading government organizations and technology companies, including the U.S. Census Bureau [64], Google [29], Apple [6], and Microsoft [44] have also adopted DP to protect individuals' data privacy.

However, current DP adoption is limited outside of large organizations and companies [20], primarily because implementing DP from scratch is complex and error-prone [37]. DP implementations must carefully account for the privacy budget, generate appropriate random noise, and require systems to be safe against known side-channel vulnerabilities. Additionally, scaling these systems to real-world datasets often requires significant software engineering effort.

To address these challenges, various tools, frameworks, and libraries [21, 22, 27, 30, 36, 41, 52–54, 57, 58, 65, 68, 69] (collectively called "DP tools" hereafter) have been developed to make DP implementation accessible to **data practitioners** — defined in this paper as professionals who have data analysis and programming skills but may not be familiar with DP. These DP tools intend to help data practitioners implement DP solutions without privacy failures. Currently, no research has systematically evaluated the usability of these DP tools; therefore, it remains unclear if they truly enable data practitioners to effectively implement DP solutions. If not, usability may be the bottleneck for wider DP adoption.

In this study, we have assessed four open-source Python-based DP tools through a mixed-methods usability study with 24 US data practitioners, evaluating four widely-used usability criteria–learnability, efficiency, error prevention, and user satisfaction [51] – to investigate three research questions:

- How effectively can DP tools help data practitioners understand DP concepts? (RQ1: DP Understanding)
- How effectively can DP tools help data practitioners implement DP solutions? (RQ2: DP Implementation)
- How satisfied are data practitioners with DP tools for their DP implementation? (RQ3: User Satisfaction)

We conducted the first comprehensive cross-tool usability study of four Python-based DP tools with data practitioners. The focus on data practitioners—the potential adopters of DP—enriches the currently end user-centered DP user research. Our contribution lies in the identification of these DP tools' usability issues and in our recommendations to improve DP tools' usability to facilitate broader DP adoption.

2 Related Work

DP and Implementation Challenges. Differential privacy (DP) [24, 25] is a formal privacy definition designed to allow statistical analysis while protecting information about individuals. Differentially private analyses, often called *mechanisms*, typically add random noise to analysis results in order to achieve privacy. Formally, two datasets $D, D' \in \mathcal{D}$ are called *neighboring datasets* if they differ in one person's data, and a mechanism $\mathcal{M}$ satisfies (ε, δ)-DP if for all neighboring datasets D and D' and sets of outcomes S:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta$$

The ε parameter is the *privacy parameter* or *privacy budget*; a smaller ε results in stronger privacy, while a larger ε results in weaker privacy. Noise drawn from the Laplace or Gaussian distributions can be used to achieve differential privacy.

Existing DP Tools. Implementing DP mechanisms is challenging. Data practitioners must determine the amount of noise to add, limit the total privacy budget, and ensure the system is free of common DP bugs [17, 32, 35, 45]). Numerous tools and libraries have attempted to make implementing DP easier for data practitioners [21, 22, 27, 30, 36, 41, 52–54, 57, 58, 65, 68, 69], often by handling the tricky parts of DP automatically. For example, tools may calculate sensitivity automatically [41, 58, 69] and ensure the privacy budget is not violated [22, 27, 36, 41, 52–54, 65, 68]. They may provide vetted implementations of basic DP mechanisms like the Laplace mechanism [21, 30, 52, 53, 68], and some also support machine learning applications [21, 53, 68]. Notably, DPCreator [22] and Private data Sharing Interface (PSI) [27] provide graphical interfaces designed for non-experts; the other tools require data science knowledge but reduce the need for DP expertise.

User Research around DP Understanding. Existing user research around DP understanding mostly focuses on end users, whose data would be in a differentially private dataset. Bullek et al. [14] examined if animated spinners can effectively communicate DP privacy guarantees to end users. Their participants preferred spinners with higher privacy levels but did not fully trust the spinners. Cummings et al. [19] studied how various DP explanations impact end-user perceptions. They found DP explanations raised participants' expectations of privacy, but did not increase their willingness to share data. Other studies explored how to better communicate DP concepts to end users. Xiong et al. [66] assessed the use of scenarios to communicate the privacy guarantees of three different DP models with participants from the USA and India. Kühtreiber et al. [38] replicated this study with participants from Germany. Both studies indicated that end users lack understanding of DP and highlighted a need for more effective DP communication. German participants were more willing to share data compared to those in the USA and India. Ashena et al. [7] also found interactive visual tools helped communicate the trade-off between accuracy and privacy loss. These studies suggest that end users have difficulty understanding DP and reservations towards DP's privacy protection.

Currently, limited user research has examined the perspective of data practitioners, who particularly need adequate DP understanding to correctly implement it. One notable study by Nanayakkara et al. [49] tested an interactive interface called Visualizing Privacy (ViP) with data practitioners without DP background, and found visualizing relationships between ε, accuracy, and disclosure risk helped them judge DP noise. Our study extends this line of research to investigate if DP tools could assist data practitioners' DP understanding.

Usability around DP Implementation. Garrido et al. [28] interviewed 24 privacy practitioners and identified both organizational and technical challenges for DP implementation in the industry. Their findings suggested that API-based DP tools could streamline data access integration and DP implementation across the enterprise.

A few studies have evaluated the usability of specific DP tools. Murtagh et al. [46] studied the usability of the web-based DP tool Privacy-preserving Integration (PSI) tool. Study participants succeeded at assigned tasks, but also identified areas of confusion and error. Sarathy et al. [59] conducted a usability study with 19 non-expert participants using the DP Creator prototype to understand perceptions, challenges, and opportunities around DP analysis. Their findings highlight user challenges including users' poor understanding of decision implications, and difficulty accessing raw data and managing workflows. We expand prior research to evaluate the usability of multiple DP tools with data practitioners.

Recently, Govtech Singapore conducted a usability assessment of DP tools [61]. They compared the same four Python-based DP tools' capabilities in analysis, security, usability, and differential privacy, generating a usability benchmark for these tools. This was an expert heuristic review without user testing, which can be subjective and lacks depth compared to our usability study that involves data practitioners.

Usability of Non-DP Tools. While our study focuses on DP tools' usability, it is critical to draw implications from prior research on non-DP tools. There is usability research on non-DP data science tools that require programming skills. Akil et al. [4] compared the usability of three prominent distributed data processing platforms for cloud computing (MapReduce, Spark, and Flink). They found ease of use, learnability, language support, auto-configuration, and community support can make big data platforms more usable to data scientists. Mehta et al. [42] evaluated five large-scale image analysis systems (SciDB, Myria, Spark, Dask, and TensorFlow) and found various usability problems, including lack of support for user-provided Python code and manual tuning requirements for efficient execution. These studies show that data science tools often fail to support data practitioners in certain data processing and analysis tasks beyond programming. Since applying DP involves data science tasks, our study investigates if DP tools share similar usability issues as other data science tools.

Software engineering researchers have identified many usability issues in the technical documentation provided by developer- or programmer-facing software tools [2,3,43,63], such as inconsistent content quality (e.g., readability, completeness, up-to-dateness) and poor navigation within the documentation. Additionally, Becker et al.'s systematic review of text-based programming error message research revealed diagnostic messages generated by compilers are often unhelpful to programmers [10]. For example, compiler error messages written in natural language were as difficult to read as source code [8], and many error messages were poorly designed, particularly for novice programmers [55]. Recommendations to improve programming error messages include increasing error message readability, reducing users' cognitive load, providing context to localize the problem, and showing examples, solutions, or hints to programmers [8,9,33,62]. In our study, we also examine how DP tools could leverage existing usability best practices from these non-DP software tools.

3 Methods and Study Design

We chose usability testing methods [23,50] to observe data practitioners' efforts to understand and implement DP using DP tools. Usability testing can identify impediments to data practitioners' DP implementation. We used surveys, interviews, and think-aloud protocol [18] for the data to answer our research questions. We executed the usability test remotely to reach a wider pool of participants. Research has shown that remote synchronous usability tests align closely in efficacy with traditional lab-based tests [5].

3.1 Selection of Differential Privacy Tools

To select tools for our study, we first conducted a review of available DP tools and decided on inclusion criteria based on study goals and feasibility that would allow for direct comparisons between tools. We required that tools: (1) be open source, (2) support standard statistical queries (count, sum, average, etc.), (3) have comprehensive documentation, and (4) provide a Python API. Based on these criteria, we did not include graphical applications like DPCreator [22] or Private data Sharing Interface (PSI) [27], or machine learning tools [53,68]. We eliminated Chorus [36,58], GoogleDP [30], Privacy on Beam [57], PINQ [41], and ZetaSQL [65, 69] due to lack of Python support. We included the remaining four tools in our study: OpenDP [52], PipelineDP [54], DiffPrivLib [21], and Tumult Analytics [11].

3.2 Study Procedures

3.2.1 Recruitment & Screening

This study received approval from our university's Institutional Review Board (IRB). We conducted a pilot study with four graduate students (one per tool) from our university and compensated them 25 US dollars each. Outcomes included adjusting study time allocation, increasing participant compensation, and clarifying survey and interview questions.

ID	Tool	DP Expertise	DP Answers Correct	Black or Hispanic	Non-Male
E005	DiffPrivLib	Expert	4/4		
E008	DiffPrivLib	Expert	3/4	x	x
E011	DiffPrivLib	Expert	4/4		x
N002	DiffPrivLib	Novice	0/4		x
N004	DiffPrivLib	Novice†	3/4		x
N011	DiffPrivLib	Novice	1/4		x
E002	OpenDP	Expert	3/4		
E007	OpenDP	Expert	4/4		
E012	OpenDP	Expert	4/4		
N003	OpenDP	Novice	2/4	x	x
N008	OpenDP	Novice	2/4		x
N012	OpenDP	Novice†	3/4	x	x
E001	PipelineDP	Expert	4/4		x
E004	PipelineDP	Expert	4/4		
E009	PipelineDP	Expert	3/4		
N005	PipelineDP	Novice	1/4		x
N009	PipelineDP	Novice	1/4		x
N013	PipelineDP	Novice	1/4		
E003	Tumult	Expert	3/4		
E006	Tumult	Expert	4/4	x	
E010	Tumult	Expert	4/4		
N006	Tumult	Novice	1/4		x
N007	Tumult	Novice†	3/4		
N010	Tumult	Novice	0/4		x

Table 1: Summary of 24 study participants. The † symbol denotes participants who were initially categorized as DP experts by the eligibility survey but then re-categorized based on incorrect answers to DP questions in the post-task interview.

We aimed to recruit at least 24 *data practitioners*, with a balanced ratio between *DP novices* and *DP experts*, according to best practices for usability testing with developers in the privacy and security field [1]. We posted the study recruitment advertisement with a link to our eligibility survey on the Women in Machine Learning and OpenDP mailing lists, on Reddit in data science-related subreddits, and on LinkedIn.

The eligibility survey (Appendix A) determined participants' eligibility, obtained potential participants' informed consent, and gathered information about their data science and DP expertise. We deemed respondents eligible if they self-reported adequate data science experience (questions 1-3) and correctly answered at least one Python question (questions 4-5). We initially categorized respondents to be *DP experts* if they correctly answered 3 out of 4 DP knowledge questions (questions 8-11), and *DP novices* otherwise. We finalized the DP expert/novice categorization after each session by assessing participants' answers to DP questions in the post-task interview (Appendix D) since multiple-choice questions in the eligibility survey were subject to guessing. This led to the re-categorization of 3 participants as DP novices (see Table 1).

Of the 109 respondents who started our eligibility survey, 83 completed it and 47 were eligible. We invited all 47 eligible respondents to the study, prioritizing underrepresented females due to diversity goals and timeline constraints.

We chronologically assigned confirmed participants to tools equally using the initial DP expert/novice categorization. After the initial tool assignment, we confirmed with each participant that they had not used the assigned DP tool before. We continued recruitment after adjusting 3 participants' expert/novice categorization until we reached our recruitment target with a balanced expert/novice ratio.

26 confirmed participants completed their study sessions but we excluded two from data analysis (N001, E012): One due to the participant's inadequate Python skills, and the other due to an unavoidable session disruption that shortened task completion time. A summary of the 24 study participants, their tool assignments, and their responses to the eligibility survey appear in Table 1. Participants' ages spanned from 18 to 40, but most (14) fell between 25-34 years. Our sample consisted of 54% females, 38% males, 4% non-binary individuals, and 4% who chose not to specify their gender. We conducted all usability test sessions on Microsoft Teams, following specific guidelines to maintain consistency. After the study session, each participant was compensated with a gift card of 40 US dollars for up to 1.5 hours of study time.

3.2.2 Pre-task Procedures

Before commencing usability study tasks, we made sure participants shared their screens and understood the think-aloud protocol. Participants also reviewed a handout that covered the fundamentals of DP and a tutorial of their assigned DP tool with executable sample DP tasks in Jupyter Notebook (see Appendix B). The handout and the tutorial provided participants necessary background for the study tasks but may introduce confounding factors to study results (detailed in Section 5.1).

We informed participants that they could refer back to the handout and the tutorial, consult the tool's official documentation, and use Google search during the study. We asked them not to use how-to resources, like StackOverflow. This ensured that participants had access to essential general resources (e.g., Python libraries) to complete the study tasks, but not to existing solutions to prevent cheating.

3.2.3 Usability Testing Tasks

We designed three usability testing tasks on differentially private data analysis, shown in Table 2. We modeled the tasks on a demo in Pipeline DP's documentation [54], changing it to a new, synthetic dataset that counted restaurant visits across a week (see Appendix E). Our tasks involved common data analysis operations supported by all four DP tools (i.e., count, sum, mean). Participants had one hour to complete the tasks by writing Python code in a shared Jupyter notebook.

The three tasks were the same across DP tools. The assigned total privacy budget was $\varepsilon = 1.2$ for all the tasks. Participants could set all other parameters themselves (including the per-task privacy budget). We encouraged participants to articulate their thought process using the think-aloud method, and we recorded both their spoken insights and on-screen actions during the study.

Task	Description
Task 1	How crowded is the restaurant on weekdays? (total number of visits for each weekday)
Task 2	Total amount of time spent by visitors on each weekday (exclude weekends).
Task 3	Average amount of time spent by visitors on each weekday (exclude weekends)

Table 2: Usability testing tasks. See Appendix E for details of the dataset used and Appendix F for solutions.

3.2.4 Post-task Procedures

Participants completed a post-task survey and a post-task interview (Appendices C and D). The survey repeated DP questions from the eligibility survey to assess participants' DP understanding after the study. It also gathered data on participants' study experiences. The post-task interview gathered qualitative data for deep insights into participants' challenges during the study and their preferences for DP tools.

3.3 Usability Measurements and Data Analysis

3.3.1 RQ1: DP Understanding

Even experienced data scientists sometimes fail to grasp the intricacies of DP [19,67]. The DP tools in our study all aim to make DP more understandable to data practitioners. To assess these tools' effectiveness in supporting DP understanding, we used the same four DP knowledge questions in the eligibility survey and the post-task survey to compare participants' pre-task and post-task DP knowledge differences. To mitigate confounding factors introduced by pre-task procedures, we also analyzed participants' explanations of key DP concepts in our post-task interview and their reported useful sources for DP understanding from post-task survey and interview.

3.3.2 RQ2: DP Implementation

We used three widely-used usability criteria — learnability, efficiency, and error prevention — to assess how effective the tools support DP implementation [51]. **Learnability** measures if new users can successfully use a specific tool or interface. We use task success and failure rates [12] to measure DP tools' general learnability. Specifically, we evaluated whether users succeeded or failed to complete tasks and assessed the correctness of their completed tasks against our reference solutions. **Efficiency** measures how fast users can accomplish tasks with a specific tool or interface. We recorded the time taken to complete each task to measure DP tools' efficiency. **Error prevention** is about how well a tool prevents user errors and, in the cases of error, how well a tool facilitates error identification and recovery. We define errors during DP implementation as interruptions of progress toward task completion and qualitatively analyzed these interruptions from the screen recordings, think-aloud, and post-task interviews. Additionally, we analyzed participants' post-task survey responses to identify the factors that impacted DP implementation.

3.3.3 RQ3: User Satisfaction

We first quantitatively evaluate user satisfaction using the two standardized measurements: the System Usability Scale (SUS) [13] and the Net Promoter Score (NPS) [31]. Since DP tools are specialized data science tools, we slightly customized the wording of SUS and NPS. We also analyze the qualitative data from post-task interviews, including their overall user experiences and areas of improvement, to yield insights into user satisfaction with these DP tools.

3.3.4 Mixed-Methods Data Analysis

We report the descriptive statistics by tool to allow usability comparison across the four DP tools examined. We also report key statistics by participants' prior DP expertise level, either expert or novice, so that usability is recognized relative to participants' prior DP knowledge. Due to the small sample size, we refrained from performing statistical tests to avoid over-generalization (details in Section 5.1).

The first and the second authors also rigorously analyzed the qualitative data collected from this study, including transcripts of audio recordings, video recordings of participants' screens, and Jupyter notebooks from all sessions. The two authors used a hybrid thematic analysis approach combining inductive and deductive coding [26] to annotate the data. They created the initial codebook from the pilot sessions and continuously refined it through research team discussions during the full study data analysis. The finalized codebook included both qualitative codes (e.g., type of challenges during implementation, misunderstandings of DP concepts) and quantitative counts derived from qualitative assessment (e.g., number of correctly completed tasks, time taken for each task). Then the first and the second authors independently coded all qualitative data using the codebook. They resolved all coding conflicts either through their own discussion or through seeking consensus from the whole research team.

4 Results

4.1 RQ1: DP Understanding

4.1.1 Pre- and Post-Task Response to DP Questions

Figure 1 reports the number of correct answers to the DP knowledge questions before and after study tasks, organized by participants' DP expertise level and by tool. Specifically, Figure 1a shows that experts provided similarly high-level of correct answers pre- and post- tasks, possibly due to their familiarity with DP concepts. However, novices showed a boost in their DP understanding as shown by the rise in correct answers from pre-task to post-task. This result indicates that our study procedures, including the DP implementation tasks, particularly helped novices understand DP concepts. Figure 1b shows the pre- and post-task DP knowledge difference across tools. All of the tools except OpenDP boosted participants' DP understanding, where DiffPrivLib saw the greatest jump from 15 to 20 correct answers. Note that the study-provided handout and tutorials also impact participants' post-task DP understanding (see Section 4.1.3).

4.1.2 Participants' Explanation of DP Concepts

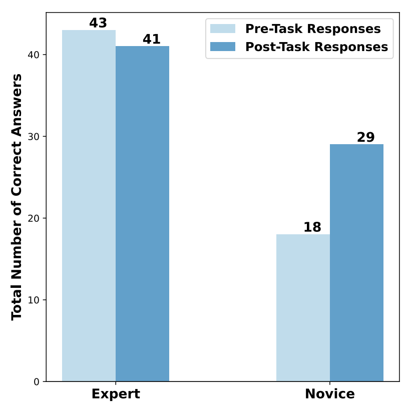
To further investigate participants' understanding of DP, we looked at how they described key DP concepts in their own words during the post-task interview (Appendix D questions 2-3), focusing on DP, ε, privacy budget for each task, and total privacy budget for all tasks. We aimed to see if participants understood that the privacy budget and ε essentially refer to the same concept and that the total privacy budget across multiple analyses accumulates the ε values (i.e., sequential composition). We considered participant responses to be correct if they were factual and includes details similar to our sample correct answers in Appendix D.

Table 3 details the number of participants who could accurately explain DP concepts, divided both by their level of expertise and by the assigned tool. All 12 expert participants accurately explained the concept of DP. For example, one expert provided a robust definition, stating, *"Differential privacy is a mathematical definition for privacy that basically says that if we compute an analysis with a particular individual's data or without it, we should get similar outputs. Whether or not somebody participates in the data set, the outcome should be pretty much the same"*(E011). This explanation is correct because it clearly describes how randomness is used in analyses to ensure results are consistent, whether an individual's data is included or not, and emphasizes the importance of the privacy parameter. When discussing the privacy budget, 11 out of 12 experts explained how the budget was allocated in individual tasks, and 10 out of 12 were able to describe how these budgets add up to form the total privacy budget.

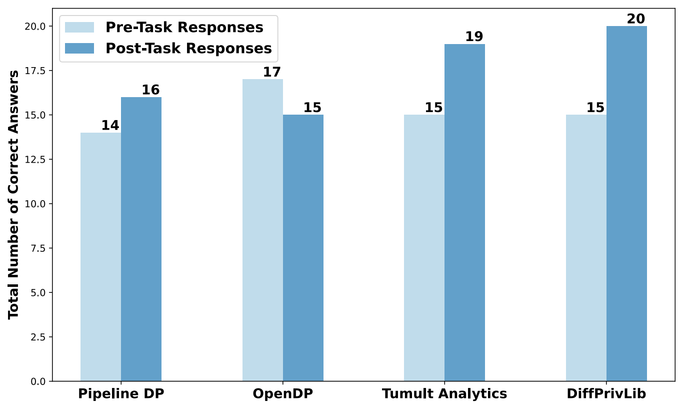
Only 9 out of 12 novices could adequately explain DP in their own words, often missing critical details. A typical novice explanation was less precise, *"From what I remember, it's like some sort of tool-based guarantee for privacy over millions of users..."* (N006), which lacks specificity and critical details about the mechanism of DP, like the privacy parameter. In their understanding of the privacy budget for each task, 8 out of 12 novices had a basic grasp. 7 out of 12 demonstrated an understanding of privacy budget accumulation, suggesting areas of confusion among novices.

The above difference between experts and novices shows the importance of participants' prior DP knowledge on their understanding. Additionally, each assigned tool had no clear impact on participants' understanding of DP, as reflected in Table 3.

Notably, participants gave incorrect answers for the question "what was the total privacy budget for the whole notebook?" more often than for the other questions (Table 3). This result was consistent across experts and novices, and across tools, suggesting that composition is a difficult concept and should be made clear by DP tools. For example, one novice



(a) By expertise level



(b) By tool

Figure 1: Total number of correct answers to DP knowledge questions before and after study tasks.

participant answered: *"I got confused between these, total budget and the amount of epsilon for each of the individual tasks. That part, I didn't get it<"* (N010)

4.1.3 Useful Sources for DP Understanding

Participants selected all sources that helped them understand DP concepts during the study in the post-task survey, as shown in Figure 2. The figure displays the average rankings of resources, where resources are ranked based on participants' preferences from 1 (most helpful) to 4 (least helpful). Figure 2a indicates that the handout and the tutorials supported their DP understanding more than DP tools' official documentation across all tools, while participants' prior DP knowledge played a key role. Figure 2b shows that experts relied heavily on their prior DP knowledge, while novices used the handout and the tutorials to understand DP concepts. This result suggests educational sources like the handout and the tutorials provided in this study benefit data practitioners' DP understanding, while DP tools' documentation lacks such support.

Post-task interview data suggests that concrete examples (like the ones in our tutorials) and short explainers (like the handout) helped participants understand important DP concepts, as E001 commented: *"It also helped to have the tutorial. ... if you had only given me the documentation... it would have taken me much longer to put it together."* Note that the handout and the tutorial were part of the study instrument, so we cannot fully attribute novices' increased DP understanding in Figure 2a to the DP tools themselves.

Moreover, our qualitative data indicates that participants could use more help with DP's actual privacy protection. In the case of ε-values and privacy budgets, we asked participants how strong they thought the privacy protection was for their just-completed task. One DP expert (E006) confidently said:*"That's the hard question to answer. The total privacy budget for all of the tasks was 1.2, a value that is in line with recommended guidelines. [ε] is around 1.0. So, maybe that's somewhat strong."*. Other responses lacked consistency and confidence: *"I think [ε] should be much lower...probably around .5 or probably even lower..."*(E003) and *"Pretty strong...very strong, actually"*(N007).

4.2 RQ2: DP Implementation

4.2.1 Learnability

Task completion and correct rates. To measure learnability, we evaluated the completeness and correctness of participants' solutions. We considered tasks **complete** when code executed without error and produced correctly formatted responses, and **correct** when they satisfied DP and had comparable utility to our reference solutions.

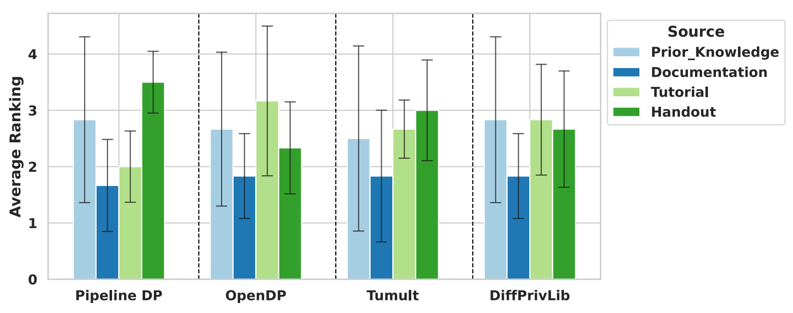
Figure 3a shows the completion rates for three usability testing tasks across four tools: all DiffPrivLib participants completed all three tasks, while none of the OpenDP participants completed tasks #2 or #3. Tumult Analytics and PipelineDP results fall between these two extremes, with all participants completing at least task #1.

The varying completion rates may derive from the different API designs of the tools. DiffPrivLib provides a minimal API and encourages users to use it in combination with Python data analytics libraries like Pandas. Similarly, Tumult Analytics mimics an existing data analytics API called Spark. OpenDP, in contrast, does not leverage mainstream Python libraries for a learning scaffold. Participant comments on API design from post-task interviews lend support to this finding. For example, one expert (E006) liked the similarity of the Tumult Analytics to Spark: *"I think the fact that it was very similar to Spark was really helpful...I have a decent amount of experience with Spark and Pandas, so that was very intuitive."*

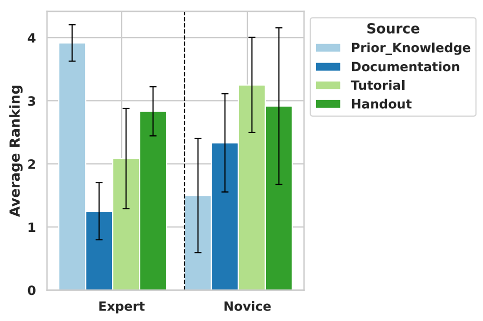
Figure 3b shows the task correctness rates. Some participants completed tasks but incorrectly, so the correctness rates are no larger than the corresponding completion rates. Combined, the completion and correctness rates show that: (1) complete Tumult Analytics and OpenDP solutions were all correct; (2) complete PipelineDP solutions were mostly—but

Question	Experts n = 12	Novices n = 12	DiffPrivLib n = 6	OpenDP n = 6	PipelineDP n = 6	Tumult n = 6
After completing the tasks, can you explain differential privacy to me in your own words?	12	9	5	6	5	5
What was the privacy budget for each task?	11	8	4	5	5	5
What was Epsilon?	12	9	5	6	5	5
What was the total privacy budget for the whole notebook?	10	7	4	4	4	5

Table 3: Number of correct answers to post-task survey questions measuring the understanding of DP concepts, disaggregated by level of expertise and by tool. Experts answered more of these questions correctly than novices, but the assigned tool had no clear impact on the number of correct answers. See Appendix C for sample correct answers.

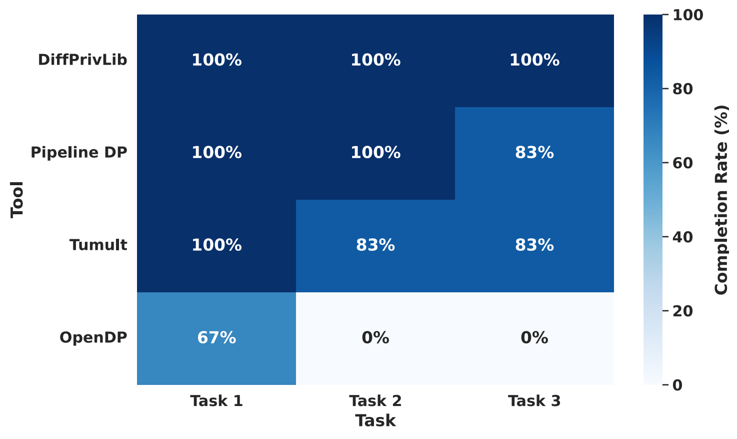


(a) Useful sources by tool

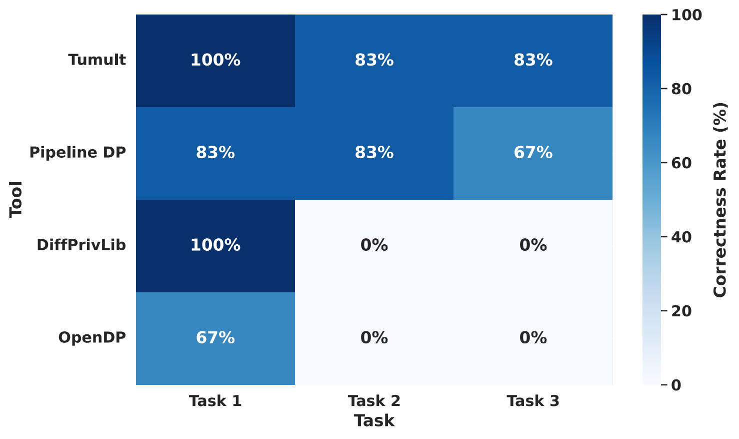


(b) Useful sources by expertise level

Figure 2: Useful sources that support participants' DP understanding, by tool (a) and by expertise level (b).

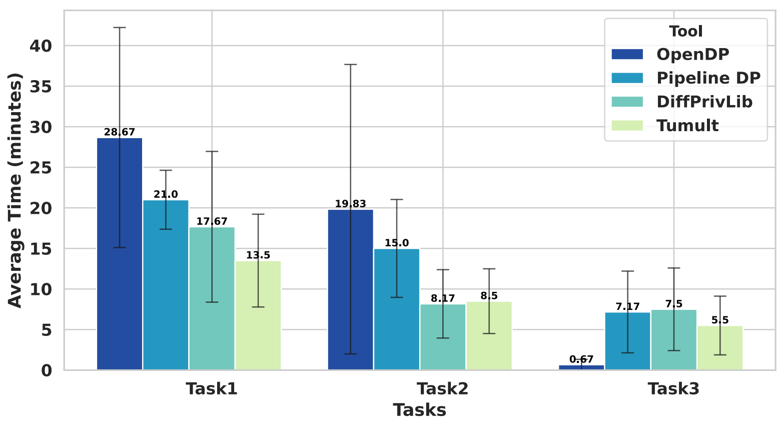


(a) Task completion rates by task and tool

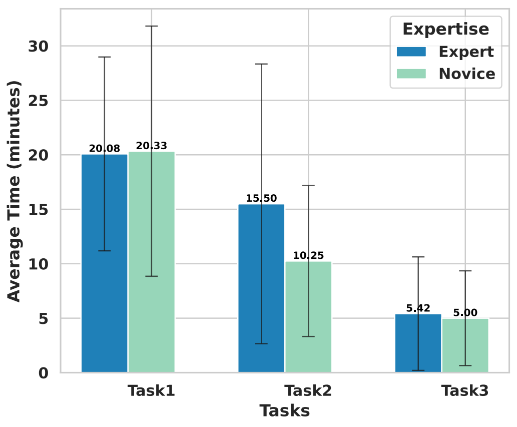


(b) Task correctness rates by task and tool

Figure 3: Learnability of DP tools measured by (a) task completion rates and (b) task correctness rates. Each cell represents the percentage of participants who completed or correctly completed the task using the tool.



(a) Average time taken by tool



(b) Average time taken by expertise level

Figure 4: Average task completion time: (a) by tool (b) by expertise level.

not all—correct; (3) Complete DiffPrivLib solutions were all *incorrect* for tasks #2 and #3.

Causes for incorrect implementation. Qualitative analysis of the screen recordings revealed the causes of some incorrectly completed tasks First, all six DiffPrivLib participants failed to apply the correct **sensitivity**, which refers to the upper and lower bounds that provide the extent of valid DP, in tasks #2 and #3. (Task #1, a counting query, has a sensitivity of one, a value that is intuitively correct.) DiffPrivLib does not signal any error related to sensitivity bounds, even though this mistake violates DP. Some expert participants were uneasy about their approach for setting sensitivity, but even these participants were not able to produce correct solutions.

Second, some tools lack **feedback** about query results' correctness. For example, one PipelineDP participant (E004) used strings (rather than integers) as grouping keys, resulting in histograms containing only 0s, and the participant did not notice the mistake. The participant later discussed this in the post-task interview, *"It's the right number of attributes and it's the right metric...the result is very noisy,"* but he added, *"I don't know if there's a way to check the final [privacy] budget."* In this case, Pipeline DP's lack of feedback affected the solution's correctness but did not violate DP.

Finally, confusion about **whether and how the tools handle the privacy budget** led to incorrectness, particularly for Pipeline DP and DiffPrivLib. E009 commented on PipelineDP: *"I would expect maybe that [a] budget accountant object could tell me my budget so far. [I'm] looking for a way to figure out how much I spent so far."* And N011 on DiffPrivLib: *"[I'm] confused about how the privacy budget would be handled at the object level. When creating the mechanism objects, should I use the same object for every analysis...and the ε will add up to the right number...can you compose all of those together? That wasn't totally clear to me."*

4.2.2 Efficiency

To measure efficiency, we calculated the time taken to complete each task by reviewing the screen recordings.

Figure 4a shows the time taken on each task by tools. OpenDP participants spent the most time on task #1 (nearly 30 minutes on average), while Tumult Analytics participants spent the least (fewer than 15 minutes on average), with DiffPrivLib and PipelineDP falling in between. The time taken for Task #2 shares a similar trend while all participants spent less time on task #2 than task #1. However, time taken for task #3 varied. OpenDP participants spent almost no time on task #3, while participants using the other three tools spent similar amounts of time on task #3, but less than that of tasks #1 and #2. The total time limit (1 hour) imposed on all tasks may affect the time spent on task #3. OpenDP participants spent nearly all of the allotted time on tasks #1 and #2, leaving little time for task #3. Participants using the other tools either finished task #3 quickly or ran out of time. *"I think I wasted a lot of time trying to find what I don't know,"* said E013.

Figure 4b shows the time spent on each task, by participants' expertise level. For tasks #1 and #3, novices and experts took roughly the same amount of time; for task #2, however, experts took *longer* than novices. Qualitative analysis from participants' think-aloud showed that experts' confidence, curiosity, and skills prompted them to explore task solutions. E005 spent time *"investigating the number of visitors that show up multiple times per day"* only to find the occurrences are rare in the data, concluding that *"we can just set the sensitivity to one."* Other experts also spent time examining API functions, honing DP parameters, checking results, and exploring alternative approaches. Novice users, in contrast, typically accepted the tool's default settings and did not spend time considering these issues. *"I'm not familiar with all the different functions,"* N011 told us in think-aloud while looking at different options for DP noise, and added post-task, *"The land of functions are [sic] totally wild to me."*

Tool	Stucks	Unstucks	Unstuck %
DiffPrivLib	31	27	87.1%
OpenDP	79	22	27.8%
Pipeline DP	54	38	70.4%
Tumult	43	38	88.4%
DP Expertise	**Stucks**	**Unstucks**	**Unstuck %**
Experts	105	64	63.8%
Novices	43	16	56.9%
Stuck Type	**Stucks**	**Unstucks**	**Unstuck %**
DP	4	3	75.0%
Documentation	65	31	47.7%
Python	28	27	96.4%
Results	18	4	22.2%
Task	34	31	91.2%
Tool	58	29	50.0%

Table 4: Stuck counts, unstuck counts, and unstuck percentages by DP tool, participants' DP expertise, and stuck type

4.2.3 Error prevention

We consider interruptions of progress toward task completion as errors during DP implementation and call them "stucks". We also examine error recovery when participants resolve these interruptions and call them "unstucks."

Stuck and unstuck statistics. We report the counts for stuck and unstuck, as well as the unstuck percentages in Table 4, organized by DP tool, by participants' DP expertise, and by stuck type (defined in Table 5). Tool-wise, participants assigned to Tumult Analytics (38/43, 88%), DiffPrivLib (27/31, 87%), and PipelineDP (38/54, 70%) often managed to get unstuck, but those assigned to OpenDP rarely got unstuck (22/79, 28%). Expertise-wise, DP experts (67/105, 64%) and novices (58/102, 57%) had similar unstuck percentages.

Stuck types. We identified six types of stuck in Table 5 and contextualized them with qualitative data. The most frequent stuck type was **documentation stucks** (65 counts, 48% unstuck percentage), where participants had difficulty finding answers in tools' documentation. *"I can imagine how to do this without this library,"* said E011 (DiffPrivLib), *"I'm trying to see...how to translate that into the library."* Second was **tool stucks** (58 counts, 50% unstuck percentage), where participants struggled to execute tools' function calls or to interpret tools' error messages. Participants would either fail to grasp tool basics: *"I don't get the terminology or the syntax,"* said E003 (Tumult Analytics); or, the issue with the tool was a specific aspect of DP: *"I'm trying to figure out how I actually tell the session what the sensitivity of the query is,"* said E006 (Tumult Analytics). **Task stucks** were common (34 counts, 91% unstuck percentage) but most participants got unstuck by asking researchers for task clarification. For example, days in our dataset are integers, 1-7. E005 (DiffPrivLib) asked, *"Can I ask is day one equal to Monday and Day 7 equal to Sunday?"* Participants also experienced **Python stucks** (28 counts, 96% unstuck percentage) but almost always got unstuck by consulting Python sources.

Usability issues. Our qualitative analysis articulated how DP tools' usability issues with their documentation and APIs caused errors and hindered error recovery. **DP tools' documentation** presented many usability problems. E001 found the upper bound for data values in PipelineDP unclear: *"I'm not super sure about this maximum value because I'm not sure if I interpret it correctly [in] the documentation."* Other participants hoped the documentation would provide more details about different API functions, such as the best DP mechanism for a data analysis task. E005 commented on DiffPrivLib's documentation, *"I do think that sometimes when you present people with a suite of 16 options, it's important to detail what the differences are and when one option might be more effective than another."* The format of the documentation was also challenging. Participants struggled due to the lack of organization of OpenDP's documentation. E013, for one, *"got lost in it."* **DP tools' APIs** also caused DP-specific errors. Participants struggled with API instructions to set parameters for DP mechanisms. And if the tools' parameters were idiosyncratic, the user interaction was less intuitive. *"I was not confident because I didn't know what the library was doing"* and *"I wasn't sure what the argument [meant],"* E004 (PipelineDP) said, *"I don't really know in the end if I computed what I was really expecting to compute."*

Error recovery was challenging. N012 was frustrated by the lack of examples *"...I couldn't get examples of people running into the same problem."* Error messages sometimes were unhelpful. For example, OpenDP's API returned error messages in Rust, and not translated to the API's Python wrapper. E002 said: *"I don't really know any Rust. Coming from a Python experience, [it] might be better to have error messages in Python that indicate the error in the line of Python."*

4.2.4 Factors impacting DP Implementation

Participants' post-task survey responses revealed the factors that helped or hindered their DP implementation in the study. Full results appear in Figures 5 and 6 in Appendix G.

9 out of 12 novices and all 12 experts reported the tutorial helped their DP implementation, with tool documentation (5 novices and 8 experts) and their data science skills (7 novices and 8 experts) close behind. Notably, none of the participants assigned to OpenDP reported that their data science skills or the tool's documentation were helpful, possibly due to how OpenDP's API differs from mainstream Python libraries. E002, someone familiar with data frames and method chaining in other Python libraries, failed to understand basic OpenDP syntax, *"I don't know what you call that little stream operator thingy."* E012 said that *"it's written in a very C-heavy style as opposed to a Python-style that most people are used to."*

8 out of 12 novices were hindered by lack of prior DP knowledge, while 5 out of 12 experts reported being hindered by DP tools' documentation in completing the tasks. Novices

Stuck Type (Abbreviation)	Definition
DP misunderstanding (DP)	Incorrectly interpreting or applying DP.
Documentation stuck (Documentation)	Struggle to interpret documentation descriptions.
Expected result stuck (Result)	Answer from a DP tool query that is not in line with expected DP values.
Python stuck (Python)	Don't know the correct Python or Pandas function to use.
Question stuck (Task)	Misinterpretation of a Task assignment, or need to clarify a Task detail.
Tool stuck (Tool)	Don't know the correct DP tool function to use. Failing to interpret error codes.

Table 5: Definitions of six types of stuck from our qualitative analysis

like N009 (Pipeline DP) *"took a lot of time understanding what the metrics are and what each parameter is"* and N010 (Tumult Analytics) *"got confused between the total [privacy] budget and the [epsilon] for each of the individual tasks."* One expert, E003, asked for *"a step-by-step guide on how you can how you can use Tumult Analytics for your particular use case."* These results suggest that DP tools should help enhance novices' DP knowledge and improve their documentation to support experts' DP implementation.

4.3 RQ3: User Satisfaction

4.3.1 Quantitative Ratings

The Net Promoter Score (NPS) and System Usability Scale (SUS) metrics from the post-task survey showed that participants were most satisfied with DiffPrivLib and least satisfied with OpenDP. DiffPrivLib had the highest NPS (33.33), followed by Tumult Analytics (-16.67), PipelineDP (-33.33), and OpenDP(-66.67). Similarly, DiffPrivLib had the highest SUS score (63.89), followed by Tumult Analytics (57.64), PipelineDP (54.51), and OpenDP (38.19). Full statistics appear in Figure 7 in Appendix G. These ratings align with the task completion rates associated with each tool (Figure 3a)—DiffPrivLib had the highest user satisfaction ratings and the highest completion rate, followed by Tumult Analytics, PipelineDP, and OpenDP. This suggests that participants were most satisfied with tools that made it easy for them to complete the study tasks.

4.3.2 Qualitative Results by Tool

Our qualitative results from the post-task interviews triangulated the above quantitative ratings and articulated participants' user experience with each tool, as described below.

DiffPrivLib received positive comments about its API and documentation: *"I liked the API of the tool. I thought the documentation was pretty clear" (E005)*. Participants also liked its compatibility with familiar libraries: *"I really liked that it integrated nicely into a library that I already have worked with, Pandas..." (E011)* and felt comfortable with the tool by the end of the session: *"Now...I'm on task three, I feel like I have a hang of the pattern...this isn't adding that much more time to my typical process" (E011)*.

Tumult Analytics was acclaimed for its intuitive API, as E010 said:*"Similarity with Pandas was definitely a plus. That's probably the best thing they've done there."* However, E003 expressed frustration with its documentation: *"It was just a single-page documentation and I had to like scroll all the way down to find the exact syntax."*. The feedback addressed the user need for improved documentation navigation.

PipelineDP exhibited documentation challenges: *"The documentation was quite incomplete...sometimes it just had one sentence about terms like 'Max contribution' or 'Max value' and it wasn't really clear to me what that meant" (E004)*. Participants also wanted the ability to search: *"What [does] the documentation say about the budget? I don't have a way to search this page" (E001)* and found error messages confusing: *"I think the error message wasn't super clear and it would be tough to debug" (E004)*. Several participants wished for examples in the documentation: *"Functions should contain some examples...[like] what each parameter is..." (N009)*.

OpenDP had usability issues with its error messages and documentation: *"The error messages I'm getting here come from Rust and I don't know what it means" (E007)*, and *"The documentation wasn't useful...[I] felt like it was a little confusing...." (N008)*. OpenDP participants also wished for examples:*"It would have been a lot more helpful if there were examples" (N012)*.

Overall, these findings on user satisfaction echo prior results — DP tools' API design and documentation quality are paramount to data practitioners' user experience.

5 Discussion and Recommendations

5.1 Limitations

We acknowledge several limitations of this study. First, we only evaluated four open-source Python-based DP tools for fair comparison across tools so that our results cannot represent all DP tools. Similarly, the findings may not generalize to all data practitioners due to our small US sample. However, our sample is similar to prior usability studies evaluating security/privacy tools with developers [39,48] to generate valid insights. Therefore, we refrained from performing statistical tests to avoid over-generalization of the statistical results from this study to all DP tools or data practitioners. Instead, we emphasized key descriptive statistics and qualitative results.

Second, our study instrument introduced confounding factors because the handout and tutorials (Section 3.2) helped participants understand DP and complete study tasks. However, we had to prioritize study feasibility to ensure participants with varying prior knowledge had the necessary information to get started. To minimize this bias, we ensured our handout and tutorials did not reveal answers to study questions or tasks, and we remained cognizant when analyzing and reporting study results.(see Sections 4.1.3 and 4.2.4).

Moreover, we only evaluated the usability of three first-step DP data analysis tasks. The results may not reflect the usability of the full capability of the examined DP tools. However, usability issues surfaced in these first-step tasks hinder developers' adoption of software tools or APIs [1,47], our recommendations for usability improvements still benefit other DP tools and encourage overall DP adoption.

5.2 Provide Usable Documentation

Our results highlighted usability issues with DP tools' official documentation, leading to the following recommendations.

Improve documentation navigation. Participants generally experienced difficulty navigating DP tools' official documentation, including technical documentation on APIs. Firstly, despite the fact that all four tools provide how-to guides with code examples in their documentation, participants struggled to find specific guides that matched their data analysis tasks at hand. For example, the descriptions of these guides are often generic and contain DP terminology (e.g., "how to perform counting queries with the Laplace mechanism"), which is unfriendly to DP novices. The mismatch between documentation contents and the practical development tasks often caused poor documentation findability [3], which can be mitigated by providing more accurate and readable task descriptions that align with users' goals [63]. Secondly, our participants disliked DP tools' single-page formatting (see Section 4.3.2). This formatting uses a single web page to organize documentation for every API function within a module, which can be lengthy, worsening the findability problem. In contrast, mainstream Python libraries (e.g., NumPy, Pandas) use one page per documented function and are easier to navigate. Additionally, some participants also hoped to be able to search within DP tools' documentation, which also resonates the proposed techniques to improve the usefulness of software documentation [2]. Our findings echo prior software engineering research on usable documentation [2, 3]. We believe DP tools can leverage existing best practices for good software documentation, such as providing intuitive task descriptions, improving information organization, and adding a search or recommender tool to improve documentation navigation.

Include DP-specific examples and advice. Many participants found the documentation for the API function they wanted to use but had trouble understanding the descriptions of DP-specific parameters and were not able to find examples that made use of the documented function. Some requested more use cases and code examples within DP tools' documentation (see Section 4.2.3). These results are consistent with prior research on developers' need for documentation [2, 43]. Moreover, some DP novice participants had trouble deciding which DP mechanism to use—for example, when given a choice between the Laplace, Gaussian, or Geometric mechanisms. Existing tool documentation fails to address these questions since it predominantly emphasizes **how** to use a specific mechanism rather than **which** mechanism to choose. To make DP tools' documentation truly usable for data practitioners, we recommend that DP tools go beyond generic best practices for usable documentation and include DP-specific advice that would particularly benefit DP novices.

5.3 Improve Error Prevention & Recovery

The study findings yielded rich insights into how DP tools can prevent errors and help users recover from errors.

Warn users about severe DP violations. PipelineDP, Tumult Analytics, and OpenDP actively prevent DP violations—they require users to wrap sensitive data using special objects. They provide error messages when users attempt to perform actions that would violate DP. DiffPrivLib, on the other hand, relies on the user to avoid DP violations; for example, DiffPrivLib asks users to set the sensitivity for every mechanism and does not check that the specified sensitivity has been correctly enforced for the input data. This explains that all of the participants assigned to DiffPrivLib completed all three tasks, but *every single participant* violated DP in their solutions for tasks #2 and #3 and failed to correctly complete them (see Figure 3 in Section 4.2.1). Thus, we recommend that DP tools proactively warn users when DP might be violated.[1]

Improve error messages. When errors occurred, many participants had difficulty diagnosing and recovering due to poorly designed error messages (Sections 4.2.3 and 4.3.2). In particular, participants assigned to PipelineDP and OpenDP described confusion over the meaning of error messages, and trouble finding documentation to understand and fix the problem. This resonates with prior research on unhelpful compiler error messages of non-DP tools [8, 9, 55].Additionally, OpenDP further confused users who primarily have a Python background with error messages generated in the programming language Rust. We first recommend DP tools learning from general best practices to improve error message readability and provide examples, solutions, and hints [10]. Additionally, DP tools should consider the average DP knowledge of their intended users and offer support when the error is DP-related (e.g., pointers to resources on DP violations).

Ensure clarity in privacy budget setting and tracking. Some participants failed to explain the total budget (Section 4.1.2) and many were concerned with setting or tracking the privacy budget with different DP tools. Tumult Analytics

[1]DiffPrivLib raises a "privacy leakage warning" in some situations that may violate DP (e.g., when setting parameters based on the data), but not in all such cases. In particular, when the programmer uses an external library like Pandas to produce an aggregate result—as all of the participants in our study did—DiffPrivLib cannot enforce sensitivity bounds on the query and does not raise a warning.

asks users to set the total and per-query budget with required API calls. This process was not as clear in other DP tools: Some participants assigned to PipelineDP and DiffPrivLib were not sure whether the library keeps track of the privacy budget at all. This confusion did not necessarily result in failure to complete the study tasks, but it would result in unintended DP violations in real-world implementations. We recommend that DP tools clearly convey how to set the privacy budget and how the tool accounts for the total budget.

Balance DP violation prevention and general usability. We also observe the tension between preventing DP violation errors and maintaining the tool's usability (Sections 4.2.1 and 4.3.1). OpenDP's strict API was effective at preventing DP violations, but OpenDP had lower completion rates and satisfaction ratings. DiffPrivLib's flexible API resulted in many DP violations but received high completion rates and satisfaction scores. Tumult Analytics seems to strike the best balance. Its API was effective at preventing DP violations where users had high completion rates and satisfaction ratings. This indicates that DP tools may need to balance between their goal to prevent DP violation errors and the tool's usability.

5.4 Make API Design Intuitive

Our findings reveal participants' unique experiences with the APIs of DP tools, leading to the following recommendations.

Leverage users' familiarity with mainstream APIs. Results in Sections 4.2.1 and 4.2.2 suggest that participants implemented DP more successfully with DP tools that incorporate mainstream APIs that they are familiar with. Specifically, the intersection of DiffPrivLib with ubiquitous libraries like Pandas, garnered commendation. This cohesive integration provided a scaffold for new learning and obviated the need for relearning. Tumult Analytics was also appreciated for the way its API mimicked that of Spark. In contrast, PipelineDP provides an API centered on performing multiple aggregations at once, and OpenDP provides an API that focuses on transformations and composition. Neither is similar to mainstream data science APIs, which impeded participants' DP implementation in the study. To make APIs more usable, we recommend DP tools prioritize API designs that allow data practitioners to transpose their extant data science knowledge to the DP context, augmenting overall satisfaction.

Assist users in setting DP-related metadata via APIs. Setting DP-related metadata (e.g. total privacy budget, ε per query, upper bound on data values) is key to DP implementation. DiffPrivLib includes **default values** for metadata. The choice to use default values simplifies the API, but may result in users accidentally accepting inappropriate default values. DiffPrivLib provides warnings when default values could result in DP violations. This helped participants to complete the tasks correctly and suggests that default values can be effective if appropriately selected and implemented. The other three tools require **users to specify DP-related metadata** via APIs. Experts appreciated that Tumult Analytics explicitly asks users to set per-query and total privacy budgets. However, DP novices may struggle with manually setting metadata, like with other non-DP tools [42]. Participants found PipelineDP's API for setting metadata confusing and struggled with settings like `max_value`, `partition_extractor`, and `privacy_id_extractor`. For OpenDP, our participants found its API, including the metadata portion, difficult to use.

We recommend that DP tools should**carefully design APIs to obtain this metadata**, as well as assist users in configuring key DP-related metadata, including exposing metadata settings, providing documentation for each metadata setting, and auto-filling appropriate default values.

5.5 Help Users with DP Foundations

Our study surfaced a general need for additional resources to help data practitioners better understand DP concepts. We found that many novices had difficulty understanding and describing the privacy budget (Section 4.1.2), and that both novices and experts sometimes had trouble describing the strength of the privacy guarantee (Section 4.3.2). These results reinforced previous findings that DP concepts are complex and difficult to communicate [14, 19, 38, 66], which inspire the following recommendations to address the challenge.

Provide general educational materials. Section 4.1.3 suggests that our study instrument boosted participants' DP understanding. DP tools may be able to replicate this effect by providing or directing users to general DP educational materials, similar to the handout and tutorials in this study.

Support privacy guarantee communication. Our participants had difficulty explaining the strength of the privacy guarantees, and several participants were unsure if their DP outputs would be private enough to be shared or published. We encourage DP tools to provide users additional community resources [4] on privacy guarantee (e.g., how to communicate the guarantee when disseminating DP analyses.)

6 Conclusion

We presented the first comprehensive usability study that evaluates four open-source Python-based DP tools with data practitioners. Our findings suggest that DP tools should provide easy-to-navigate, DP-specific documentation, enhance error prevention and recovery capabilities, improve API designs to ease users' learning curves, and offer resources to strengthen users' DP foundations. We aim for our findings and recommendations to facilitate broader DP adoption.

Acknowledgments

The authors thank the SOUPS reviewers and shepherd for their hepful suggestions that resulted in significant improvements to the paper. This work was supported in part by an Amazon Research Award.

References

[1] Yasemin Acar, Sascha Fahl, and Michelle L Mazurek. You are not your developer, either: A research agenda for usable security and privacy research beyond end users. *2016 IEEE Cybersecurity Development (SecDev)*, pages 3–8, 2016.

[2] Emad Aghajani, Csaba Nagy, Mario Linares-Vásquez, Laura Moreno, Gabriele Bavota, Michele Lanza, and David C Shepherd. Software documentation: the practitioners' perspective. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 590–601, 2020.

[3] Emad Aghajani, Csaba Nagy, Olga Lucero Vega-Márquez, Mario Linares-Vásquez, Laura Moreno, Gabriele Bavota, and Michele Lanza. Software documentation issues unveiled. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1199–1210. IEEE, 2019.

[4] Bilal Akil, Ying Zhou, and Uwe Röhm. On the usability of hadoop mapreduce, apache spark & apache flink for data science. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 303–310. IEEE, 2017.

[5] Morten Sieker Andreasen, Henrik Villemann Nielsen, Simon Ormholt Schrøder, and Jan Stage. What happened to remote usability testing? an empirical study of three methods. CHI '07, page 1405–1414, New York, NY, USA, 2007. Association for Computing Machinery.

[6] Apple. Apple: Differential Privacy Overview, 2023. https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf.

[7] Narges Ashena, Oana Inel, Badrie L. Persaud, and Abraham Bernstein. Casual users and rational choices within differential privacy. In *Proceedings of the 2024 IEEE Symposium on Security and Privacy*, pages 88–88, 2024.

[8] Titus Barik, Justin Smith, Kevin Lubick, Elisabeth Holmes, Jing Feng, Emerson Murphy-Hill, and Chris Parnin. Do developers read compiler error messages? In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 575–585, 2017.

[9] Brett A. Becker. An effective approach to enhancing compiler error messages. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE '16, page 126–131, New York, NY, USA, 2016. Association for Computing Machinery.

[10] Brett A Becker, Paul Denny, Raymond Pettit, Durell Bouchard, Dennis J Bouvier, Brian Harrington, Amir Kamil, Amey Karkare, Chris McDonald, Peter-Michael Osera, et al. Compiler error messages considered unhelpful: The landscape of text-based programming error message research. *Proceedings of the working group reports on innovation and technology in computer science education*, pages 177–210, 2019.

[11] Skye Berghel, Philip Bohannon, Damien Desfontaines, Charles Estes, Sam Haney, Luke Hartman, Michael Hay, Ashwin Machanavajjhala, Tom Magerlein, Gerome Miklau, et al. Tumult analytics: a robust, easy-to-use, scalable, and expressive framework for differential privacy. *arXiv preprint arXiv:2212.04133*, 2022.

[12] Nigel Bevan. Practical issues in usability measurement. *Interactions*, 13(6):42–43, 2006.

[13] John Brooke. Sus: a "quick and dirty'usability. *Usability evaluation in industry*, 189(3), 1996.

[14] Brooke Bullek, Stephanie Garboski, Darakhshan J Mir, and Evan M Peck. Towards understanding differential privacy: When do people trust randomized response technique? In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3833–3837, 2017.

[15] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019.

[16] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[17] Sílvia Casacuberta, Michael Shoemate, Salil Vadhan, and Connor Wagaman. Widespread underestimation of sensitivity in differentially private libraries and how to fix it. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 471–484, 2022.

[18] Lynne Cooke. Assessing concurrent think-aloud protocol as a usability test method: A technical communication approach. *IEEE Transactions on Professional Communication*, 53(3):202–215, 2010.

[19] Rachel Cummings, Gabriel Kaptchuk, and Elissa M Redmiles. " i need a better description": An investigation into user expectations for differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3037–3052, 2021.

[20] Damien Desfontaines. *Lowering the cost of anonymization*. PhD thesis, ETH Zurich, 2020.

[21] DiffPrivLib, 2023. https://github.com/IBM/differential-privacy-library.

[22] DP Creator, 2023. https://github.com/opendp/dpcreator.

[23] Joseph S Dumas and Janice Redish. A practical guide to usability testing, 1999.

[24] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[25] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[26] Jennifer Fereday and Eimear Muir-Cochrane. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods*, 5(1):80–92, 2006.

[27] Marco Gaboardi, James Honaker, Gary King, Jack Murtagh, Kobbi Nissim, Jonathan Ullman, and Salil Vadhan. Psi ({\Psi}): a private data sharing interface. *arXiv preprint arXiv:1609.04340*, 2016.

[28] Gonzalo M Garrido, Xiaoyuan Liu, Florian Matthes, and Dawn Song. Lessons learned: Surveying the practicality of differential privacy in the industry. *Proceedings on Privacy Enhancing Technologies*, 2:151–170, 2023.

[29] Google. Google: Differentially private heatmaps, 2023. https://blog.research.google/2023/04/differentially-private-heatmaps.html.

[30] Google's differential privacy libraries, 2023. https://github.com/google/differential-privacy.

[31] Douglas B Grisaffe. Questions about the ultimate question: conceptual considerations in evaluating reichheld's net promoter score (nps). *Journal of Consumer Satisfaction, Dissatisfaction and Complaining Behavior*, 20:36, 2007.

[32] Samuel Haney, Damien Desfontaines, Luke Hartman, Ruchit Shrestha, and Michael Hay. Precision-based attacks and interval refining: how to break, then fix, differential privacy on finite computers. *arXiv preprint arXiv:2207.13793*, 2022.

[33] Björn Hartmann, Daniel MacDougall, Joel Brandt, and Scott R. Klemmer. What would other programmers do: suggesting solutions to error messages. CHI '10, page 1019–1028, New York, NY, USA, 2010. Association for Computing Machinery.

[34] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

[35] Jiankai Jin, Eleanor McMurtry, Benjamin IP Rubinstein, and Olga Ohrimenko. Are we there yet? timing and floating-point attacks on differential privacy systems. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 473–488. IEEE, 2022.

[36] Noah Johnson, Joseph P Near, Joseph M Hellerstein, and Dawn Song. Chorus: a programming framework for building scalable differential privacy mechanisms. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 535–551. IEEE, 2020.

[37] Daniel Kifer, Solomon Messing, Aaron Roth, Abhradeep Thakurta, and Danfeng Zhang. Guidelines for implementing and auditing differentially private systems. *arXiv preprint arXiv:2002.04049*, 2020.

[38] Patrick Kühtreiber, Viktoriya Pak, and Delphine Reinhardt. Replication: The effect of differential privacy communication on german users' comprehension and data sharing attitudes. In *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, pages 117–134, 2022.

[39] Tianshi Li, Yuvraj Agarwal, and Jason I Hong. Coconut: An ide plugin for developing privacy-friendly apps. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4):1–35, 2018.

[40] Wired Magazine. T-Mobile's $150 Million Security Plan Isn't Cutting It, 2023. https://www.wired.com/story/tmobile-data-breach-again/.

[41] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30, 2009.

[42] Parmita Mehta, Sven Dorkenwald, Dongfang Zhao, Tomer Kaftan, Alvin Cheung, Magdalena Balazinska, Ariel Rokem, Andrew Connolly, Jacob Vanderplas, and Yusra AlSayyad. Comparative evaluation of big-data systems on scientific image analytics workloads. *arXiv preprint arXiv:1612.02485*, 2016.

[43] Michael Meng, Stephanie Steinhardt, and Andreas Schubert. Application programming interface documentation: What do software developers want? *Journal of Technical Writing and Communication*, 48(3):295–330, 2018.

[44] Microsoft. Microsoft AI: Differential Privacy, 2023. https://www.microsoft.com/en-us/ai/ai-lab-differential-privacy.

[45] Ilya Mironov. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 650–661, 2012.

[46] Jack Murtagh, Kathryn Taylor, George Kellaris, and Salil Vadhan. Usable differential privacy: A case study with psi. *arXiv preprint arXiv:1809.04103*, 2018.

[47] Varvana Myllärniemi, Sari Kujala, Mikko Raatikainen, and Piia Sevońn. Development as a journey: factors supporting the adoption and use of software frameworks. *Journal of software engineering research and development*, 6:1–22, 2018.

[48] Alena Naiakshina, Anastasia Danilova, Christian Tiefenau, Marco Herzog, Sergej Dechand, and Matthew Smith. Why do developers get password storage wrong? a qualitative usability study. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 311–328, 2017.

[49] Priyanka Nanayakkara, Johes Bater, Xi He, Jessica Hullman, and Jennie Rogers. Visualizing privacy-utility trade-offs in differentially private data releases. *Proceedings on Privacy Enhancing Technologies*, 2022(2):601–618.

[50] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann, 1994.

[51] Jakob Nielsen. Usability metrics: Tracking interface improvements. *IEEE software*, 13(6):1–2, 1996.

[52] OpenDP, 2023. https://github.com/opendp/opendp.

[53] Nicolas Papernot. Machine learning at scale with differential privacy in TensorFlow. In *2019 USENIX Conference on Privacy Engineering Practice and Respect (PEPR 19)*, 2019.

[54] PipelineDP, 2023. https://pipelinedp.io/.

[55] James Prather, Raymond Pettit, Kayla Holcomb McMurry, Alani Peters, John Homer, Nevan Simone, and Maxine Cohen. On novices' interaction with compiler error messages: A human factors approach. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, ICER '17, page 74–82, New York, NY, USA, 2017. Association for Computing Machinery.

[56] Associated Press. Wawa agrees to payment, security changes for '19 data breach, 2022. https://apnews.com/article/technology-pennsylvania-malware-attorney-generals-office-0ebedd8dce8bf0e21833f52944a48b56.

[57] Privacy on Beam, 2023. https://github.com/google/differential-privacy/tree/main/privacy-on-beam.

[58] Chorus Repository, 2023. https://github.com/uvm-plaid/chorus.

[59] Jayshree Sarathy, Sophia Song, Audrey Haque, Tania Schlatter, and Salil Vadhan. Don't look at the data! how differential privacy reconfigures the practices of data science. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2023.

[60] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[61] Anshu Singh and Syahri Ikram. Benchmarking differential privacy python tools. https://github.com/dsaidgovsg/benchmarking-differential-privacy-tools, 2023.

[62] V Javier Traver. On compiler error messages: what they say and what they mean. *Advances in Human-Computer Interaction*, 2010:1–26, 2010.

[63] Christoph Treude, Martin P Robillard, and Barthélémy Dagenais. Extracting development tasks to navigate software documentation. *IEEE Transactions on Software Engineering*, 41(6):565–581, 2014.

[64] U.S. Census Bureau. Why the Census Bureau Chose Differential Privacy, 2023. https://www.census.gov/library/publications/2023/decennial/c2020br-03.html.

[65] Royce J Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. Differentially private sql with bounded user contribution. *Proceedings on Privacy Enhancing Technologies*, 2:230–250, 2020.

[66] Aiping Xiong, Tianhao Wang, Ninghui Li, and Somesh Jha. Towards effective differential privacy communication for users' data sharing decision and comprehension. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 392–410. IEEE, 2020.

[67] Aiping Xiong, Chuhao Wu, Tianhao Wang, Robert W Proctor, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Using illustrations to communicate differential privacy trust models: An investigation of users' comprehension, perception, and data sharing decision. *arXiv preprint arXiv:2202.10014*, 2022.

[68] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.

[69] ZetaSQL differential privacy extension, 2023. https://github.com/google/differential-privacy/tree/main/examples/zetasql.

A Eligibility Survey

For questions that test participants' understanding, we highlight the correct answer in bold.

Eligibility Questions after displaying IRB-approved consent form

- I have read and understood the information above. No/Yes
- I want to proceed to complete the eligibility survey for this research study. No/Yes
- Are you at least 18 years old? No/Yes
- Do you reside in the United States? No/Yes
- Have you performed statistical data analysis in Python? No/Yes
- Have you used the Jupyter Notebook before? No/Yes
- Are you willing to participate in a study to evaluate a data science tool that will require you to code in Python in a Jupyter Notebook? No/Yes
- Are you willing to participate in a 1.5-hour usability study remotely via Microsoft Teams? No/Yes

Questions on Python, DP, and basic demographics

1. How many years have you been coding in Python?
   (a) 0-1
   (b) 2-3
   (c) More than 3

2. How many years have you been using the Jupyter Notebook?
   (a) 0-1
   (b) 2-3
   (c) More than 3

3. Which of the following best describes how you use Python and the Jupyter notebook for statistical analysis?
   (a) They are my preferred language/tool
   (b) I am comfortable using them but I prefer other languages/tools (e.g., R)
   (c) I can work with them but often need to resort to documentation
   (d) I rarely use them and need additional time to get familiar with them.

4. Use "set" instead of "list" as a Python data structure for a sequence of elements when:
   (a) elements will be appended to increase the size of the sequence
   (b) the order of items is important
   (c) **it is important to know if the sequence contains a specific item**
   (d) it is important to know the item with maximum value in the sequence
   (e) I don't know

5. What is the output of the following code?

```
str1 = "DataScience is fun!"
print(str1[4:12])
```

   (a) **Science**
   (b) Data Sci
   (c) aScience
   (d) Error
   (e) I don't know

6. Have you heard of the term differential privacy (DP) before?
   (a) No
   (b) Yes

7. Have you ever written code to implement differential privacy (DP) in any capacity?
   (a) No
   (b) Yes

8. In differential privacy, which value of the privacy parameter ε provides stronger privacy?
   (a) **ε = 0.1**

(b) $\varepsilon = 1.0$

(c) I don't know

9. Releasing two differentially private statistics, one with $\varepsilon_1 = 0.1$ and the other with $\varepsilon_2 = 0.5$, results in a total privacy loss of:

    (a) $\varepsilon = 0.1$

    (b) $\varepsilon = 0.5$

    (c) **$\varepsilon = 0.6$**

    (d) $\varepsilon = 0.05$

    (e) I don't know

10. If the mechanism M returns a number and satisfies differential privacy with $\varepsilon = 0.1$, does abs(M(x)) satisfy differential privacy, where abs is the absolute value function?

    (a) No, not necessarily

    (b) **Yes, for $\varepsilon = 0.1$**

    (c) Yes, for some $\varepsilon > 0.1$

    (d) I don't know

11. Which of the following is an advantage of using Differential Privacy?

    (a) It guarantees complete anonymity of the data subjects

    (b) It ensures that the data is completely accurate

    (c) **It provides a tradeoff between privacy and utility of the data**

    (d) It is a computationally simple method for preserving privacy in large datasets

    (e) I don't know

12. What is your age?

13. What is your gender?

14. Are you an undergraduate or a graduate student?

B The Handout and the Tutorials

Available at Open Science Framework (OSF): https://osf.io/ag2fj/?view_only=29a9bc2a30574befa9f3d0643951b9c6

C Post-Task Survey

For questions that test participants' understanding, we highlight the correct answer in bold.

1. Please enter your participant ID

2. Please rate the following statements using the [Likert] scale indicated below.

    (a) I think that I would like to use [DP tool] frequently.

    (b) I found [DP tool] unnecessarily complex.

    (c) I thought [DP tool] was easy to use.

    (d) I think that I would need the support of a technical person to be able to use [DP tool].

    (e) I found the various functions in [DP tool] were well integrated.

    (f) I thought there was too much inconsistency in [DP tool].

    (g) I would imagine that most people would learn to use [DP tool] very quickly.

    (h) I found [DP tool] very cumbersome to use.

    (i) I felt very confident using [DP tool].

    (j) I needed to learn a lot of things before I could get going with [DP tool].

    (k) I found [DP tool] introduced DP concepts appropriately for me to perform the tasks.

    (l) I feel I have to learn DP concepts more systematically to solve the tasks.

3. If another data scientist that you know needs to use differential privacy in their data analysis, how likely is it that you would recommend Tumult Analytics to them?

    - 10-point Likert scale, "Not at all likely" to "Extremely likely"

4. If you completed at least one task in the study, what helped you successfully complete the task(s)? Choose all that apply.

    (a) The Differential Privacy handout (including the video)

    (b) The [DP tool] tutorial (including its examples)

    (c) The official [DP tool] documentation

    (d) My prior data science skills (like Python, Pandas, statistics, etc.)

    (e) My prior knowledge of Differential Privacy

    (f) Other (please specify)

    (g) N/A (I didn't complete any tasks)

5. What hindered your completion of the tasks? Choose all that apply.

    (a) The Differential Privacy handout (including the video)

    (b) The [DP tool] tutorial (including its examples)

    (c) The official [DP tool] documentation

    (d) My prior data science skills (like Python, Pandas, statistics, etc.)

(e) My prior knowledge of Differential Privacy

(f) Other (please specify)

6. If the mechanism M returns a number and satisfies differential privacy with $\varepsilon = 0.1$, does abs(M(x)) satisfy differential privacy, where abs is the absolute value function?

   (a) No, not necessarily

   (b) **Yes, for $\varepsilon = 0.1$**

   (c) Yes, for some $\varepsilon > 0.1$

   (d) I don't know

7. In differential privacy, which value of the privacy parameter ε provides stronger privacy?

   (a) **$\varepsilon = 0.1$**

   (b) $\varepsilon = 1.0$

   (c) I don't know

8. Releasing two differentially private statistics, one with $\varepsilon_1 = 0.1$ and the other with $\varepsilon_2 = 0.5$, results in a total privacy loss of:

   (a) $\varepsilon = 0.1$

   (b) $\varepsilon = 0.5$

   (c) **$\varepsilon = 0.6$**

   (d) $\varepsilon = 0.05$

   (e) I don't know

9. Which of the following is an advantage of using Differential Privacy?

   (a) It guarantees complete anonymity of the data subjects

   (b) It ensures that the data is completely accurate

   (c) **It provides a tradeoff between privacy and utility of the data**

   (d) It is a computationally simple method for preserving privacy in large datasets

   (e) I don't know

D Post-Task Interview

For questions that test participants' understanding, we give sample correct answers after each question.

Thank you for completing/making an effort to complete the tasks with [tool] and the post-task survey. Now we have a few questions for you to reflect on your experience with the study.

1. After completing the tasks, can you explain differential privacy to me in your own words?
   Correct answer: Differential privacy is a formal property that limits the distributional difference between a statistic computed on one dataset and the same statistic computed on a neighboring dataset.

2. Consider the tasks you worked on just now, can you explain:

   (a) What was the privacy budget for each task?
   Correct answer: Depends on the parameters used by the participant—it should be equal to the value of ε used in each task's solution.

   (b) What was Epsilon?
   Correct answer: Same as (a).

   (c) What was the total privacy budget for the whole notebook?
   Correct answer: $3 \times$ (answer from (a)), by sequential composition.

   (d) If the results you computed were released to the public, how strong would you expect the privacy protection for individuals in the original data to be?

3. During this study, what helped you most in understanding the concepts (e.g., privacy budget, Epsilon) that we discussed just now? Please rank the following options from "most useful" to "least useful".

   (a) The Differential Privacy handout (including the video)

   (b) The [DP tool] tutorial (including its examples)

   (c) The official [DP tool] documentation

   (d) My prior knowledge of Differential Privacy

   (e) Other (please specify)

4. When using [tool] in the study, what aspects/components of [tool] do you think are helpful for you to complete the tasks?

5. When using [tool] in the study, what aspects/components of [tool] do you think are frustrating for you to complete the tasks?

6. After this study, what recommendation(s) do you have to improve the usability of [tool]?

7. Can you tell us what helped you successfully complete the task(s)?

   (a) The Differential Privacy handout (including the video)

   (b) The [DP tool] tutorial (including its examples)

   (c) The official [DP tool] documentation

(d) My prior data science skills (like Python, Pandas, statistics, etc.)

(e) My prior knowledge of Differential Privacy

(f) Other (please specify)

8. Can you tell us what hindered your completion of the (task)?

    (a) The Differential Privacy handout (including the video)

    (b) The [DP tool] tutorial (including its examples)

    (c) The official [DP tool] documentation

    (d) My prior data science skills (like Python, Pandas, statistics, etc.)

    (e) My prior knowledge of Differential Privacy

    (f) Other (please specify)

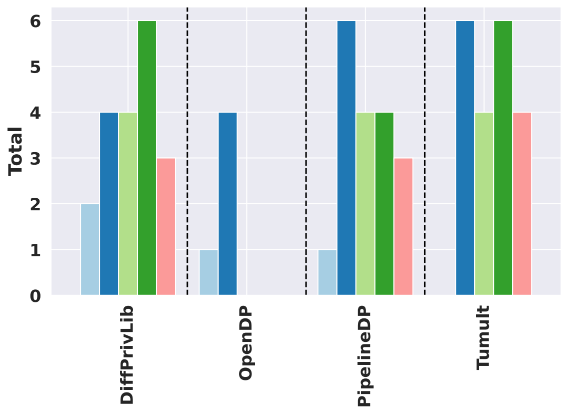
E Task Dataset

The dataset used for the tasks was provided to participants in a CSV file. This is a synthetic dataset that counted restaurant visits across a week, where each record represented a distinct visit with a visitor ID. The full dataset is available at Open Science Framework (OSF): https://osf.io/ag2fj/?view_only=29a9bc2a30574befa9f3d0643951b9c6

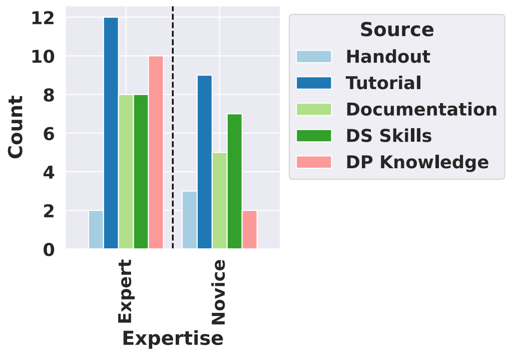
F Task Solutions

We wrote sample solutions for the three tasks from Table 2 for each tool we studied. Participant solutions were usually similar, but not necessarily identical. We will make these sample solutions available publicly via Open Science Framework on publication of the paper.

G Additional Figures

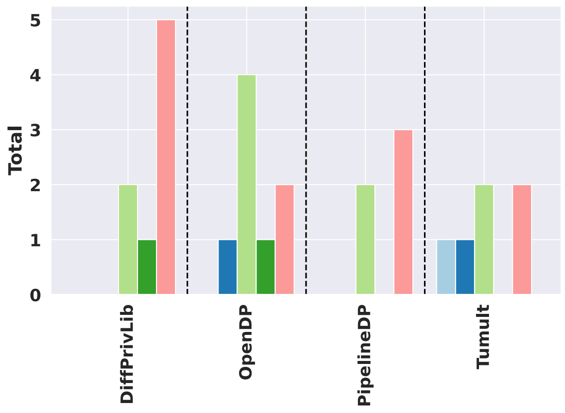


(a) By Tool

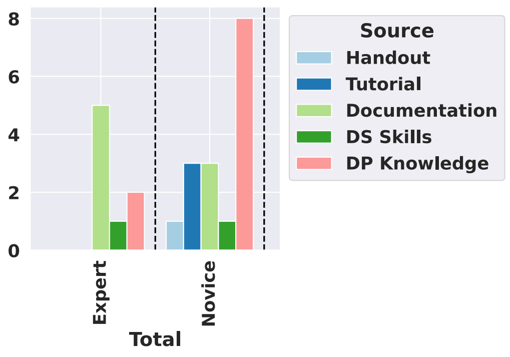


(b) By Expertise

Figure 5: Factors helping task completion by tool and expertise.

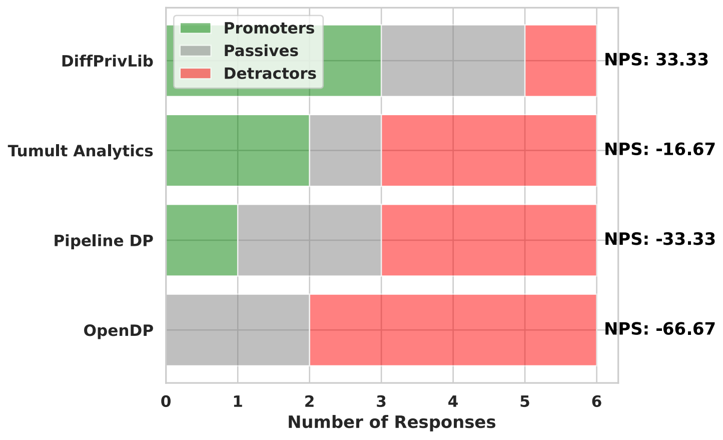


(a) By Tool

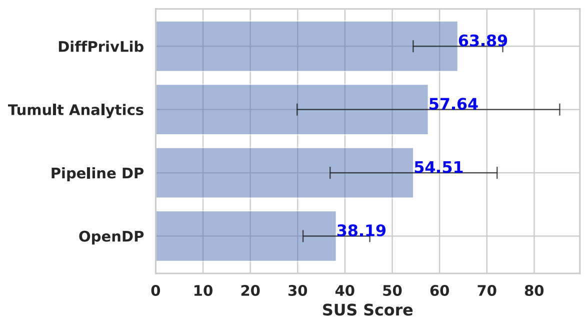


(b) By Expertise

Figure 6: Factors hindering task completion by tool and expertise.



(a) Net Promoter Score (NPS) results



(b) System Usability Scale (SUS) results

Figure 7: User satisfaction scores: (a) Net Promoter Score (NPS), and (b) System Usability Score (SUS).