Swipytics: Leveraging the Short-form Video Stream Metaphor in Exploratory Data Analysis

Jiwon Choi*
Match Group AI

Sehi L'Yi†
Hong Kong University of Science and Technology

Seojin Kim‡
North Carolina State University

Jaemin Jo§
Sungkyunkwan University

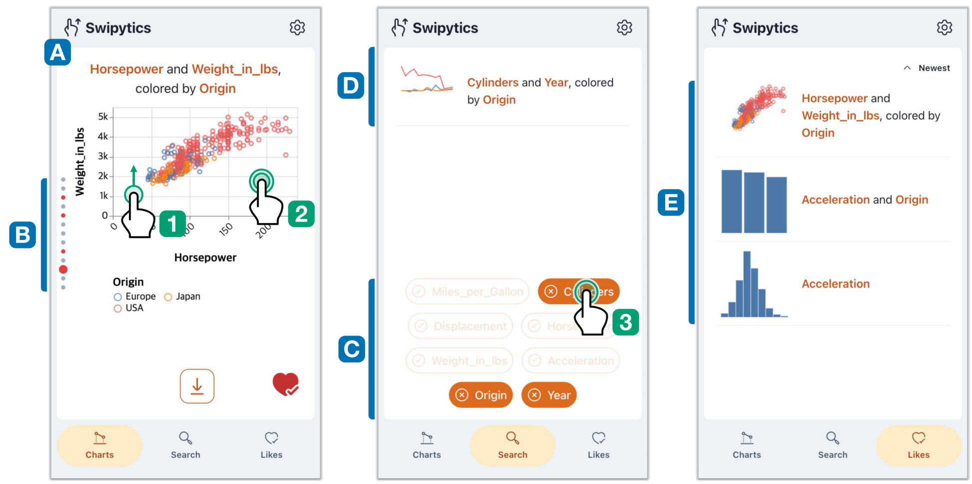

Figure 1: The three tabs and three exploration interactions in Swipytics. *Left – Charts Tab*: a short-form-video-inspired carousel where swiping up ((1)) moves from the current chart (A) to a recommendation based on history and preferences; liking by double-tap or heart button ((2),) steers related recommendations; the History Indicator (B) shows prior states ( ) and liked charts ( ). *Middle – Search Tab*: users query charts by tapping attribute tags ((3), C), view matching charts (D), and add them to the Charts Tab. *Right – Likes Tab*: users revisit liked charts (E).

Abstract

We present Swipytics, a mobile interface designed to lower the onboarding barrier to exploratory data analysis (EDA) by leveraging the metaphor of short-form video streams. While EDA tools are increasingly accessible to a broad and diverse audience, they often require users to first learn the interface through verbal instructions or lengthy video tutorials, leaving an onboarding barrier. To lower this barrier, we leverage the metaphor of short-form video streams, offering minimal and familiar user interactions for EDA, such as swipe gestures for exploring new visualizations, with visualizations recommended one at a time in a linear, user-controlled sequence. Our user study demonstrates that participants quickly achieved high levels of proficiency and confidence in EDA after a short self-learning period (approximately 100 seconds on average).

Index Terms: Visualization, Exploratory Data Analysis

1 Introduction

Exploratory data analysis (EDA) is a fundamental step in data science workflows, where users iteratively manipulate and visualize data to uncover insights [16]. This iterative process requires the user to construct and interpret a sequence of meaningful visualizations, which can be complex and time-consuming, especially for novices [5] and non-programmers [13]. The visualization community has explored remedies to lower this cost, such as providing more familiar interactions like post-WIMP interactions [9, 11], or alleviating manual configuration through visualization recommendations [8,27,28,32]. While these techniques have proven effective, they still demand that users first learn to operate the interface, typically through guided tutorials, interactive tours, or verbal instructions lasting tens of minutes in practice [4,9,27,28,32]. We refer to this initial training requirement as the *onboarding barrier*.

Aiming to lower the onboarding barrier, we present Swipytics, a mobile interface that leverages the metaphor of short-form video platforms (e.g., TikTok, YouTube Shorts, and Instagram Reels)—among today's most prevalent media consumption platforms. Our goal is not to maximize expressive power, but rather to ensure that novices can begin exploring data immediately, shifting their cognitive effort from *learning* the interface to *interpreting* visualizations. Swipytics employs a carousel of visualizations, analogous to the carousel of short-form videos, that users scroll through with simple swiping, with new visualizations recommended based on exploration history. In designing Swipytics, we formulated three design rationales grounded in prior research on novices' barriers, resulting in an interface that offers minimal, familiar interactions with visualizations delivered at a controlled and linear pace. A user study with 14 participants showed that participants acquired high proficiency and confidence after a self-learning period of under two minutes on average, without any tutorial videos or verbal instruction.

2 Related Work

Prior research has identified barriers to the initial use of visualization systems and proposed remedies, including post-WIMP interactions and recommendations. Building on these efforts, our work

*e-mail: jiwon.choi@match.com
†e-mail: sehi@ust.hk
‡e-mail: seojin.kim.ko@gmail.com
§Corresponding author. e-mail: jmjo@skku.edu

combines both in a familiar metaphor-based interface to mitigate the onboarding barrier [22].

Barriers in Visualization Systems. Despite the potential of visualization, steep learning curves and interaction barriers persist for novices [5, 12–14, 29]. Lam [13] identified seven interaction costs classified into three critical "*gulfs*": the *gulf of execution* (difficulty translating intentions into actions), the *gulf of evaluation* (difficulty assessing system state), and the *gulf of goal formation* (difficulty deciding on analysis goals). Subsequent research [5] elaborated on specific hurdles like data selection and visual mapping. To mitigate these, researchers explored literacy assessment [18], post-WIMP interactions [9, 11], and visualization recommendation [8, 27]. Recent work highlights that overcoming these hurdles requires effective visualization onboarding to fill users' knowledge gaps [22]. These studies informed our design (Sec. 3.1), aiming to alleviate the *three gulfs* [13] through gesture-based interactions and recommendations. In the context of Lam's framework [13], the onboarding barrier we aim to address encompasses all three gulfs. Novices not only struggle to translate their analytical intentions into system operations (execution), but also face difficulties in formulating meaningful questions (goal formation) and interpreting unfamiliar visual outputs (evaluation) when using a new system without effective onboarding mechanisms.

Post-WIMP Interactions for Visualization. Visualization systems increasingly employ post-WIMP interactions like pen, touch [9], gestures, and speech [11] to alleviate the gulf of execution [13]. Mobile visualization further motivates such designs: small screens and touch input constrain view coordination, control density, and precise manipulation, making desktop EDA interfaces hard to transfer to mobile [19, 21]. However, these advanced interactions often require a learning curve. For example, users can be confused by numerous touch gestures or forget specific speech commands, reverting to explicit modalities like touch [11]. These difficulties stem from the low discoverability of implicit post-WIMP interactions [9]. We address this discoverability issue while preserving effectiveness by exploiting the familiar metaphor of short-form video streams, leveraging users' existing mental models.

Visualization Recommendation. Visualization recommendation alleviates the gulf of execution by suggesting visual encodings or meaningful data subsets. Several systems automate visual encoding selection [8, 28]. For instance, Voyager [28] uses heuristics, Draco [31] formalizes design knowledge as constraints, and VizML [8] leverages deep learning. Other systems focus on identifying noteworthy insights, using deviation-based metrics (SeeDB [23]). In EDA, systems recommend subsequent visualizations based on the current view (e.g., Voyager [27, 28], ChartSeer [32]). Prior EDA research shows that revisiting analysis history helps refine exploration goals and analytic trajectories [1,6,26]. However, reviewing multiple recommendations simultaneously can increase cognitive load, and these systems often demand lengthy video tutorials or verbal guidance, hindering onboarding. We use a familiar metaphor to lower early exploration cost and support goal formation through history-aware recommendations.

3 The Swipytics Interface

3.1 Design Rationale

We clarify how Lam's three *gulfs* [13] inspired our design rationales for lowering early interaction costs. We describe how each rationale addresses the training requirement novices face when learning to use a visualization system, i.e., what we collectively refer to as the onboarding barrier, drawing on barriers identified in prior work [5, 7, 12, 29].

DR1. Use minimal and familiar interactions. Swipytics targets users with minimal prior knowledge of visualization design (e.g., marks, channels, and mappings) [7] and EDA. Providing numerous interaction options to users often makes them feel lost or overwhelmed among the many available choices. To address this, we employ a small set of interactions inspired by familiar short-form video interfaces, such as swipe gestures, to minimize the *gulf of execution* [13, 17] for the user engaging in EDA. These minimal interactions enable the user to focus on EDA rather than struggling with the construction of visualizations, a major barrier identified in previous studies [5, 7, 12, 29].

DR2. Show simple visualizations at a controllable pace. Existing EDA systems typically present multiple coordinated views either as lists or dashboards [28, 32]. However, this can increase cognitive load and potentially overwhelm the user [12, 25]. To address this, rather than presenting complex multivariate charts or multiple views at once, Swipytics surfaces one simple visualization at a time, i.e., univariate, bivariate, or at most trivariate, each time the user swipes. This linear, slideshow-like pacing [20] lets users inspect charts sequentially with reduced visual overload, while exploration history contextualizes the current state [13, 17].

DR3. Make the exploration sequence linear. While expert analysts typically navigate through complex exploration graphs or trees [27, 28], novices often struggle to design and determine their exploration paths independently [13, 29]. To alleviate this, we choose to keep the exploration path linear, i.e., the user navigates through visualizations in a sequence. Inspired by how experts construct and explore analysis graphs [1, 26], we make Swipytics automatically compute relationships with prior analysis history to recommend a linear exploration path. Motivated by the gulf of goal formation [13], this design reduces the burden of choosing an initial path, while history-aware recommendations cue emerging goals during exploration.

3.2 Views and Interactions

To reduce the initial learning friction, we leverage widely used interactions and interfaces from short-form video platforms (**DR1**), such as swipe gestures (vertical navigation), double-tap (like), hashtags, and content recommendation (Fig. 1). We treat each visualization as analogous to a short-form video, and the data attributes of a visualization as hashtags; built around this metaphor, the Swipytics user interface consists of three tabs (Fig. 1), Charts Tab, Search Tab, and Likes Tab, which can be navigated using the tab bar at the bottom of the interface.

Charts Tab. The Charts Tab shows one recommended chart at a time (Fig. 1A), similar to short-form video interfaces and narrative slideshows [20]. The user can swipe up the chart to see the next chart recommended by the system or swipe down to see the previous chart (Fig. 1(1), **DR1**). To help the user navigate the carousel of charts, Swipytics provides a compact indicator on the left edge of the screen, the History Indicator (Fig. 1B). Each chart is represented with a circle, where the enlarged circle indicates the currently displayed chart (**DR3**).

Consistent with most short-form video interfaces, the user can express their preference for a certain chart (i.e., the "Like" interaction) by either double-tapping a chart (Fig. 1(2)) or tapping the heart button (Fig. 1). The red circles in the History Indicator correspond to the liked charts. This interaction steers the recommendation engine to recommend charts related to the liked charts. Note that Swipytics does not precompute the carousel of charts; once the user arrives at the end of the carousel, only the next chart is computed on the fly. The collection of liked charts can be quickly browsed in the Likes Tab (Fig. 1, Right).

Search Tab. The Charts Tab carousel may not suffice for targeted EDA tasks, such as inspecting a chart with attributes of interest [1]. To address this, we design the **Search Tab** where the user can choose data attributes of interest and see the charts that include the attributes. Previous studies employed shelf-configuration interfaces [27,28] for the targeted tasks. However, using such interfaces can be challenging for the user who lacks a prior understanding of visualization encoding (i.e., visual mapping barrier [5]). Moreover, shelf-configuration interfaces use drag-and-drop interactions, which are difficult to perform on touch devices [9].

Instead, we draw inspiration from hashtag-based search interaction in short-form video platforms, treating data attributes associated with a chart as the chart's tags (Fig. 1C). Instead of manually specifying visual mappings between data attributes and visual channels, the user can tap the data attributes (i.e., tags) they want to visualize in the Search Tab (Fig. 1(3)). The matched charts are shown in the list (Fig. 1D), and the user can tap a chart of interest to add it to the end of the carousel in the Charts Tab.

Likes Tab. The Likes Tab enables the user to see their liked charts. Charts are displayed with simplified thumbnails (Fig. 1E) that omit several visual elements (e.g., grid, axis, title, and legend) to enhance recall of previously liked charts [6]. The user can quickly revisit past charts in the Charts Tab by tapping a thumbnail (**DR3**).

3.3 Implementation Details

In this section, we describe the implementation details of Swipytics's heuristic-based recommendation algorithm, which emphasizes simplicity. Our algorithm operates in two steps: 1) Attribute Selection, which scores possible analysis states (i.e., sets of attributes to be displayed) [1,27,28], and 2) Visual Mapping, which selects a high-scoring state and maps it to an appropriate visualization. Further details on each step can be found in Appendix A and B. A live demo and the open-source implementation are available at `https://github.com/jiwnchoi/Swipytics`.

Attribute Selection. This step evaluates analysis states using a scoring function based on exploration history. Each state is a set of a maximum of three attributes (**DR2**) with three types: *numeric*, *categorical*, or *temporal*. We exclude categorical attributes with cardinality above ten to prevent visual clutter on small screens. The scoring function combines four weighted components. 1) *Relevance* maintains exploration continuity, scoring higher for states sharing attributes with recent charts. 2) *Preference* aligns with user interests, assigning higher scores to states containing attributes from visualizations marked "Like" (Fig. 1♥). 3) *Simplicity* prioritizes simple analysis states (i.e., univariate), mitigating complexity during initial exploration and aligning with natural exploration patterns [1]. 4) *Freshness* balances novelty and revisitation, assigning the highest scores to unencountered states while gradually increasing scores for shown states.

Visualization Mapping. After selecting the highest-scoring state, the algorithm maps it to a suitable visualization. This mapping uses chart templates indexed by the types of selected attributes. For example, a combination of one temporal and one numeric attribute yields a line chart, while two numeric attributes make a scatterplot. Inspired by previous studies [8, 27, 31], we constructed these templates with Draco 2 [31] using example datasets [24] and stored them in a hashmap. Using templates instead of invoking the recommendation engine at interaction time reduced latency and removed noticeable delays.

4 User Study

We conducted an observational user study evaluating the effectiveness of our system in lowering the onboarding barrier, focusing on self-learning experiences and exploration patterns. We chose a qualitative approach over quantitative baseline comparisons because comparable mobile-first EDA systems are lacking and existing desktop tools assume prior visualization knowledge. This let us investigate how users adopt Swipytics's metaphor-based design.

4.1 Participants and Setup

We recruited 14 university students (eight females, six males; ages 19–30) from diverse backgrounds. Following prior work [30], we selected participants without formal training or experience in visual analysis tools (e.g., Tableau). Participants used their own smartphones (11 unique iOS/Android devices) in the 45-minute on-site study and received ∼\$21 USD. Sungkyunkwan University IRB approved the study (No. 2024-11-025).

4.2 Study Protocol

Introduction (10 min). Participants completed a demographic questionnaire, consented, and took the Mini-VLAT test [18] to ensure minimal visualization literacy. All scored above our 2.9 screening threshold ($\mu = 8.71$, $\sigma = 2.11$, Min $= 5.00$). We then installed Swipytics on their smartphones.

Self-learning Phase (up to 5 min). Participants freely explored the Cars dataset [24] in the system for up to five minutes. We intentionally used a lightweight tutorial sheet (Appendix C), similar to common first-use mobile overlays [10], rather than verbal instruction or lengthy training [9,27,28,30], to study whether the interface could be appropriated under minimal and unsupported onboarding conditions. They could stop early when ready, then rated their perceived proficiency and confidence on a 5-point Likert scale.

Exploration Phase (15 min). Following Voyager's design [28], participants were asked to "comprehensively explore the data" without specific tasks to avoid bias. We used the Movies dataset [24] (8 quantitative, 2 categorical, 1 temporal attribute). Dataset questions were allowed, but questions about using Swipytics were restricted to observe natural usage. Participants could take notes and finish early if satisfied; we logged all interactions.

Debriefing (10 min). Participants verbally articulated findings using saved charts and notes, then completed the System Usability Scale (SUS) [2] and a custom questionnaire. To verify whether the brief self-learning was sufficient, we asked the two questions on proficiency and confidence again. Three additional questions assessed recommendation helpfulness, chart interpretability, and smartphone usability (Appendix D). We concluded with semi-structured interviews on exploration patterns and responses.

4.3 Results & Analysis

Participants generated 573 charts ($\mu = 40.93$, $\sigma = 16.05$): 175 univariate ($\mu = 12.50$, $\sigma = 3.41$), 356 bivariate ($\mu = 25.43$, $\sigma = 13.15$), and 42 trivariate ($\mu = 3.00$, $\sigma = 4.22$), *liking* 72 charts (♡ → ♥, $\mu = 5.14$, $\sigma = 4.93$). More charts were added via Charts Tab swiping ($\mu = 28.21$, $\sigma = 15.57$) than the Search Tab ($\mu = 12.71$, $\sigma = 14.48$).

Figure 2 shows diverse exploration patterns. Participants appropriated Swipyticsin varied ways rather than following a single scripted mode. Many began with lightweight discovery via repetitive swiping, used the Search Tab for targeted follow-up, and returned to the Charts Tab for insights (P1, P5, P8, P12). This suggests complementary rather than redundant roles. They also progressed from simpler to more complex (e.g., trivariate) charts, suggesting paced exploration rather than immediate overload, aligning with Battle and Heer's findings on open-ended exploration [1]. While most followed this swipe-then-search pattern, exceptions existed: P4, P7, and P14 used only the Charts Tab, whereas P9 and P10 preferred or relied on the Search Tab.

Self-learning took under two minutes ($\mu = 101.07$s, $\sigma = 18.18$s). Immediately afterward, participants rated proficiency ($\mu = 4.29$, $\sigma = 0.83$) and confidence ($\mu = 4.71$, $\sigma = 0.47$) highly. After exploration, proficiency slightly increased ($\mu = 4.57$, $\sigma = 0.51$), while confidence stayed consistent ($\mu = 4.71$, $\sigma = 0.47$), suggesting that the initial brief onboarding was sufficient for comfort, with subsequent use mainly reinforcing it. Swipytics received favorable

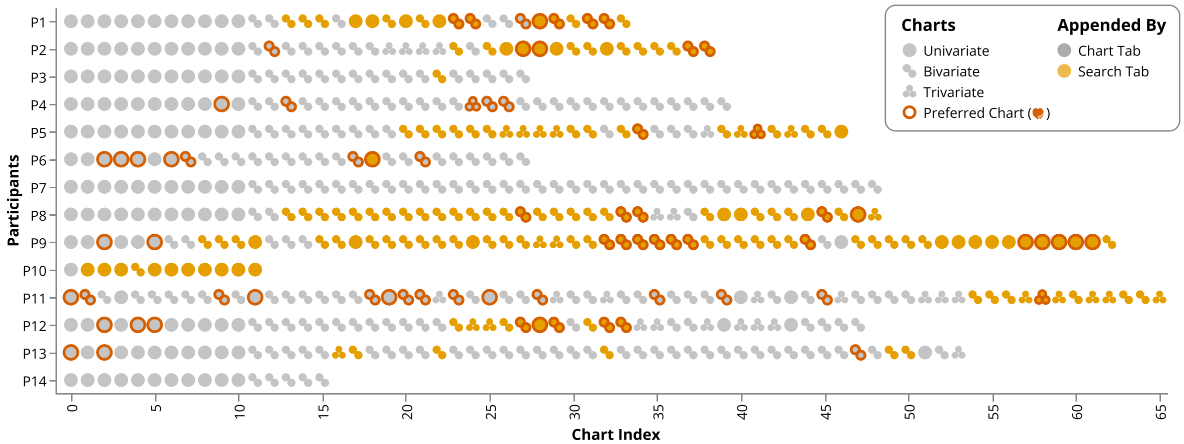

Figure 2: User interaction and exploration patterns during the exploration phase. Participants appropriated Swipytics in heterogeneous ways rather than following a single scripted mode. For many, swiping supported lightweight discovery while the Search Tab enabled targeted follow-up, suggesting complementary rather than redundant roles. The gradual shift from univariate to more complex charts suggests paced exploration rather than immediate overload.

helpfulness ($\mu = 4.50,\ \sigma = 0.65$), interpretability ($\mu = 4.21,\ \sigma = 0.89$), and smartphone usability ($\mu = 4.93,\ \sigma = 0.27$) ratings. Finally, Swipytics achieved a "good" SUS score of 83.75 (> 68) [3].

5 Discussion

5.1 Metaphor-Based Design for EDA

We observed that the familiar metaphor effectively lowered the onboarding barrier. Participants reported feeling accustomed to the system in less than two minutes on average through the short self-learning period. Feedback from participants such as P13 confirmed this: *"I could work independently... without needing someone to explain things to me."* Users not only quickly learned basic interactions but also transferred their existing mental models from short-form video interfaces to EDA. P11 noted, *"I repeatedly swiped the carousel and specified Likes... similar to what I do in Instagram Reels."* This suggests that familiar metaphors can effectively transfer interaction habits.

Furthermore, users could understand the recommendation algorithm's behavior through the interface. P9 observed, *"Looking at the History Indicators, I could see that the system recommends charts related to the ones I had seen before."* This interpretability led to diverse strategic approaches. For instance, P4 and P5 recognized the prioritization of univariate charts early on and adapted their behavior accordingly. These findings highlight the importance of sensible algorithm design and complementary visual cues in reducing the gulf of evaluation [17].

5.2 Flexibility and Adaptation

Participants demonstrated diverse exploration patterns and interaction preferences. After targeted searches, P12 used swiping for discovery, while P10 relied solely on the Search Tab. Similarly, the Like interaction (Fig. 1) was used for both bookmarking and recommendation steering. P9 used Likes to focus on specific attributes early on, while P6 used them as rough anchors in the analysis history. These variations demonstrate Swipytics's flexibility in accommodating different exploration strategies according to users' needs and understanding.

5.3 Limitations of Metaphor-Based Design

One limitation is the emphasis on sequential viewing over fine-grained configuration. P5 expressed a desire for manual axis configuration to better examine correlations. Additionally, Swipytics's linear exploration poses challenges for non-linear comparison tasks. While the History Indicator (Fig. 1B) helped, sequential navigation remained inefficient for comparing non-adjacent items, leading users to develop strategies like repetitive swiping or Likes Tab thumbnails. Accordingly, the short-form video metaphor may be better suited to low-friction entry into EDA than to later-stage analysis that requires branching, revisiting, and comparison [1, 6, 26]. Finally, our relatively young university-student participants may have been more familiar with short-form video interactions; users unfamiliar with such streams may find the interaction logic less intuitive, limiting broader onboarding and generalizability.

6 Limitations and Future Work

One limitation is the lack of quantitative comparison between Swipytics and existing systems. Establishing a suitable baseline was challenging because few EDA interfaces are designed for small-screen mobile devices while mitigating the onboarding barrier as effectively as ours [19]. Thus, it was difficult to fairly compare exploration effectiveness, such as using insight-based metrics [27, 28], with a similar training period. In future work, we plan to develop metaphor-free alternatives, such as shelf-based configurations, to enable more direct comparative evaluations.

Furthermore, our findings may be constrained by the dataset used. We used the Movies dataset [24] for its familiarity. Using a familiar dataset may have helped participants formulate questions, letting us focus on the effectiveness of the metaphor while minimizing dataset complexity. However, this raises a practical question: How effectively can participants explore datasets without domain expertise? How users formulate meaningful questions with unfamiliar datasets requires further study. Moreover, because participants were non-experts, their questionnaire responses should be interpreted as perceived usefulness rather than objective evidence of analytical quality.

7 Conclusion

In this paper, we presented Swipytics, a novel mobile interface that mitigates the onboarding barrier to EDA by leveraging the familiar metaphor of short-form video streams. By adopting this metaphor, Swipytics provides a simplified, controllable, and linear exploration experience, offering visualizations one at a time through minimal and familiar interactions. Our user study demonstrated that participants, even without prior experience with EDA or visualization tools, achieved high proficiency with Swipytics after an average self-learning period of under two minutes, using only a lightweight tutorial sheet, without verbal instruction or video tutorials.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190421, Artificial Intelligence Graduate School Program (Sungkyunkwan University)) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00221186).

References

[1] L. Battle and J. Heer. Characterizing exploratory visual analysis: A literature review and evaluation of analytic provenance in tableau. *Computer Graphics Forum*, 38(3):145–159, 2019. doi: 10.1111/cgf.13678 2, 3, 4, 6

[2] J. Brooke. SUS: A "quick and dirty'' usability scale. *Usability evaluation in industry*, 189(3):189–194, 1996. 3, 7

[3] J. Brooke. SUS: A retrospective. *Journal of usability studies*, 8(2), 2013. 4

[4] V. Dhanoa, A. Hinterreiter, V. Fediuk, N. Elmqvist, E. Gröller, and M. Streit. D-tour: Semi-automatic generation of interactive guided tours for visualization dashboard onboarding. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):721–731, 2025. doi: 10.1109/TVCG.2024.3456347 1

[5] L. Grammel, M. Tory, and M.-A. Storey. How information visualization novices construct visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):943–952, 2010. doi: 10.1109/TVCG.2010.164 1, 2, 3

[6] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, 2008. doi: 10.1109/TVCG.2008.137 2, 3, 4

[7] J. Heer, F. van Ham, S. Carpendale, C. Weaver, and P. Isenberg. Creation and collaboration: Engaging new audiences for information visualization. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, eds., *Information Visualization: Human-Centered Issues and Perspectives*, pp. 92–133. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-70956-5_5 2

[8] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. VizML: A machine learning approach to visualization recommendation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300358 1, 2, 3

[9] J. Jo, S. L'Yi, B. Lee, and J. Seo. Touchpivot: Blending wimp & post-wimp interfaces for data exploration on tablet devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 2660–2671. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3025453.3025752 1, 2, 3

[10] C. Kelleher and R. Pausch. Stencils-based tutorials: design and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, p. 541–550. Association for Computing Machinery, New York, NY, USA, 2005. doi: 10.1145/1054972.1055047 3

[11] Y.-H. Kim, B. Lee, A. Srinivasan, and E. K. Choe. Data@hand: Fostering visual exploration of personal data on smartphones leveraging speech and touch interaction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445421 1, 2

[12] B. C. Kwon, B. Fisher, and J. S. Yi. Visual analytic roadblocks for novice investigators. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 3–11, 2011. doi: 10.1109/VAST.2011.6102435 2

[13] H. Lam. A framework of interaction costs in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1149–1156, 2008. doi: 10.1109/TVCG.2008.109 1, 2

[14] S. Lee, S.-H. Kim, Y.-H. Hung, H. Lam, Y.-A. Kang, and J. S. Yi. How do people make sense of unfamiliar visualizations?: A grounded model of novice's information visualization sensemaking. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):499–508, 2016. doi: 10.1109/TVCG.2015.2467195 2

[15] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956. 6

[16] T. Milo and A. Somech. Automating exploratory data analysis via machine learning: An overview. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, p. 2617–2622. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3318464.3383126 1

[17] D. A. Norman. *The design of everyday things*. MIT Press, 2013. 2, 4

[18] S. Pandey and A. Ottley. Mini-vlat: A short and effective measure of visualization literacy. *Computer Graphics Forum*, 42(3):1–11, 2023. doi: 10.1111/cgf.14809 2, 3

[19] R. Sadana and J. Stasko. Designing multiple coordinated visualizations for tablets. *Computer Graphics Forum*, 35(3):261–270, 2016. doi: 10.1111/cgf.12902 2, 4

[20] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, 2010. doi: 10.1109/TVCG.2010.179 2

[21] L. S. Snyder, R. A. Rossi, E. Koh, J. Heer, and J. Hoffswell. Interaction Techniques for Exploratory Data Visualization on Mobile Devices. In C. Tominski, M. Waldner, and B. Wang, eds., *EuroVis 2024 - Short Papers*. The Eurographics Association, 2024. doi: 10.2312/evs.20241057 2

[22] C. Stoiber, M. Wagner, F. Grassinger, M. Pohl, H. Stitz, M. Streit, B. Potzmann, and W. Aigner. Visualization onboarding grounded in educational theories. In D. Albers Szafir, R. Borgo, M. Chen, D. J. Edwards, B. Fisher, and L. Padilla, eds., *Visualization Psychology*, pp. 139–164. Springer International Publishing, Cham, 2023. doi: 10.1007/978-3-031-34738-2_6 2

[23] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: Efficient data-driven visualization recommendations to support visual analytics. *Proc. VLDB Endow.*, 8(13):2182–2193, sep 2015. doi: 10.14778/2831360.2831371 2

[24] Vega datasets. `https://vega.github.io/vega-datasets`, 2024. Accessed: 2024-11-18. 3, 4

[25] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, p. 110–119. Association for Computing Machinery, New York, NY, USA, 2000. doi: 10.1145/345513.345271 2

[26] K. Wongsuphasawat, Y. Liu, and J. Heer. Goals, process, and challenges of exploratory data analysis: An interview study, 2019. 2, 4

[27] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016. doi: 10.1109/TVCG.2015.2467191 1, 2, 3, 4

[28] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 2648–2659. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3025453.3025768 1, 2, 3, 4

[29] M. A. Yalçin, N. Elmqvist, and B. B. Bederson. Cognitive stages in visual data exploration. In *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization*, BELIV '16, p. 86–95. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2993901.2993902 2

[30] M. A. Yalçın, N. Elmqvist, and B. B. Bederson. Keshif: Rapid and expressive tabular data exploration for novices. *IEEE Transactions on Visualization and Computer Graphics*, 24(8):2339–2352, 2018. doi: 10.1109/TVCG.2017.2723393 3

[31] J. Yang, P. F. Gyarmati, Z. Zeng, and D. Moritz. Draco 2: An extensible platform to model visualization design. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pp. 166–170, 2023. doi: 10.1109/VIS54172.2023.00042 2, 3

[32] J. Zhao, M. Fan, and M. Feng. Chartseer: Interactive steering exploratory visual analysis with machine intelligence. *IEEE Transactions on Visualization and Computer Graphics*, 28(3):1500–1513, 2022. doi: 10.1109/TVCG.2020.3018724 1, 2