Resubmission of InfoVis 2019 submission "Extending Spreadsheets to Support Seamless Navigation at Scale" to TVCG

A previous version of this paper was submitted to InfoVis 2019. This version was rejected as the paper co-chairs deemed that revising the paper would require more time than allocated to meet reviewer expectations. The paper co-chairs mentioned the following: "*We particularly encourage such revisions where submissions were positively received by reviewers, but the revisions required were deemed to be beyond the scope of the conference review cycle. If you address the issues raised and subsequently submit to TVCG, please make reference to this InfoVis submission and include a description of how you addressed the InfoVis reviewers' comments.*" The entire set of reviews are attached. In particular, the reviewers asked for the following changes for a revision:

C1: More clearly articulate the merits over alternative tools, such as Keshif.
C2: Convincingly explain the design choice around the binning mechanism or revise that approach.
C3: Justify the choice of comparing with Excel and note the limitations of that approach.
C4: Better justify the choice of tasks in the study.
C5: Explain the choice of the four research questions.
C6: Provide more details about the study, including reporting intra-participant differences.

We thank the reviewers and meta-reviewer for their detailed, constructive feedback. Taking this feedback into account, we have spent the last four months preparing a revised version of this paper. We believe the paper is substantially stronger as a result. Every section in the paper has been revised; the changes are highlighted in blue. We briefly describe the changes to each section before returning to the changes required in the revision (**C1**–**C6**).

Changes Organized by Section

- In Section 1, we now cite work that highlights the widespread adoption of spreadsheets even among users of advanced enterprise solutions, further justifying why we focused on addressing navigation challenges within spreadsheets. In fact, recent debate within the visualization community following VIS 2019 also echoes this view[1]—instead of designing a new sophisticated tool from scratch that may cater to a small population, we aim to enhance the user experience for existing spreadsheet tools with nearly a billion users. We further augment the definitions of spreadsheet interactions like scrolling and steering in Section 1, and now clearly articulate the challenges of designing a general-purpose plug-in for spreadsheets.
- In Section 2, our usage scenario has been substantially revised using Brehmer and Munzner's typology [1] to clarify the scope of tasks supported by spreadsheets, as well as those that are enhanced when using a spreadsheet with a NOAH plug-in; our new Table 1 provides use-cases supported within NOAH for each category.
- In Section 3, we now clearly articulate the differences in goals of spreadsheets and tabular data analysis tools (TDA), further justifying our choice of Excel, the most widely used spreadsheet tool, as our baseline in the evaluation study.
- In Section 4, we now explain the third design consideration for NOAH more clearly (**DC3**)—focusing on motivating the binning mechanism.
- In Section 5, we now justify why we are developing a multi-granularity overview, and explain why we opted for histograms as an overview representation, by contextualizing our approach using prior work on multi-scale aggregation [2].
- In Section 6, we further clarified the goals of our study and reframed our research questions to better reflect those goals. Our research questions haven't been substantially altered; rather, they have been grouped together and made more precise. In this version, we now explain our choice for the quiz tasks and discuss three limitations of our study in detail.
- Section 7 has been completely revamped to better reflect our research questions, while adding more qualitative observations regarding the user experience with NOAH. We now include additional analysis results on intra-participant differences.
- In Section 8, we have added a summary of our takeaways from the evaluation study, while discussing additional limitations of NOAH and future enhancement opportunities.

Changes Organized by Reviewer Concerns (C1–C6)

1) *Merits over other tools* (**C1**): Throughout the paper (and especially in Sections 1, 3, and 6), we have emphasized that the main contribution of NOAH is its *design as a general-purpose navigation plug-in* to any existing spreadsheet system. As spreadsheets have a massive user-base, enhancing exploration and formula computation on large datasets while maintaining the spreadsheet look-and-feel as much as possible, has the potential to impact hundreds of millions of spreadsheet users. Most of these users employ spreadsheets as their primary data management and analysis tool while shunning enterprise solutions with more advanced features. Therefore, our goal is to develop a solution to improve navigation within spreadsheets. Indeed, we could have tried to enhance navigation in other tools, such as Keshif, or Tableau, like the reviewers suggest—but we would be forcing spreadsheet users to adopt an entirely new tool, something that they are clearly loath to do.
2) *Explanation of the binning mechanism* (**C2**): We now explain our choice of the binning mechanism in the context of the third design consideration (*DC3*), as detailed in Section 4. We further expand on this in the blurb titled "Why a Multi-granularity Binned Overview" in Section 5. In brief, we opted for binning to provide a clear and concise representation of the overall data distribution while minimizing user's back and forth movement across multiple screens during navigation. At some level, we do need to limit the number of values displayed so that the overview fits on the screen: binning is a natural solution for that issue. We cite related work on multi-scale aggregation [2] to motivate

1. https://twitter.com/FILWD/status/1187411664611749888

the choice of a multi-granularity overview and show that binned aggregation via histograms [3] is a suitable representation of such an overview. However, we acknowledge limitations for this choice for binning categorical data and discuss possible solutions in Section 8 in the context of our user study findings.

3) *Justification of Excel as a Baseline* (**C3**): Since our goal is to improve spreadsheet user experience—and there are very good reasons for doing so, as outlined in item 1) above—the natural comparison point is Excel. That said, beyond popularity, we have provided a thorough justification for what spreadsheet systems like Excel offer relative to tabular data analysis tools (TDA), making it a more appropriate point of comparison. Our justification for using Excel over tabular data analysis (TDA) tools involves a) highlighting the appeal of spreadsheets among users who shun more advanced enterprise tools (Section 1), b) identifying the scope of tasks supported by spreadsheets and TDA tools (Section 2 and 3), c) explaining that the goal of spreadsheets is to present the raw data as is, amenable to editing, formula computation, and comparison of derived data, all done in-situ, unlike TDA tools that hide the data while providing summarized statistics (Section 3), and d) highlighting the importance of addressing the shortcomings of a tool, the user base of which far exceeds that of TDA tools (Section 6).
4) *Enhancing the user study design* (**C4, C5**): We have added additional explanations regarding the goals of the study while better articulating the research questions, provided a better justification for the choice of the tasks for the study (see Section 6.1) and added a detailed discussion on the limitations of the study. Specifically, we merged our previous *RQ1* (evaluating the quiz task performance) with *RQ4* (summarizing user feedback on performance via a survey) into a unified research question (new *RQ1*) on the participants' navigation performance. Moreover, we merged previous *RQ2* and *RQ3* that explored how specific features of NOAH affected participants' navigation experience into one broad question (new *RQ2*) that explores their overall experience with NOAH and its components. Moreover, we have justified the choice of the quiz tasks by a) relating their goals with the scope of tasks supported by spreadsheets (see Section 2), and b) explaining how these tasks can reveal how people may use NOAH for spreadsheet navigation. Finally, we have identified three limitations of the study, resulting from the sample size and the study design, which we justify based on the scope of the study.
5) *Reporting Intra-participant Differences* (**C6**): We performed additional analysis of the study data to report intra-participant submission time differences for the quiz phase tasks. While we summarize the results of the analysis in Section 7.1.1, the detailed result can be found in AppendixAGP: pls add; insights on trends sounds vague let's be more clear. We find that, despite a few exceptions, across all tasks, participants' task submission times were faster using NOAH—all but one participants completed at least four tasks in less time using NOAH compared to Excel. However, we already obtained similar observations from Figure 5 in Section 7.1. Therefore, we have included the results of the intra-participant analysis in Appendix.

Extending Spreadsheets to Support Seamless Navigation at Scale

Sajjadur Rahman, Mangesh Bendre, Yuyang Liu, Shichu Zhu, Karrie Karahalios, Aditya Parameswaran

Abstract—Spreadsheets are one of the most popular tools for ad-hoc exploration and analysis of data. Despite that, exploring and analyzing spreadsheet datasets that span more than a few screens via operations such as scrolling or issuing formulae, is often overwhelming for end-users. Users easily lose context as they explore the data via scrolling and suffer from cognitive and mechanical burdens while issuing formulae on data spanning multiple screens. We propose integrating a navigation plug-in with spreadsheets to support the seamless exploration of large datasets that are increasingly the norm. Our interface, NOAH, developed using lessons from classical overview+detail interfaces, embeds a multi-granularity zoomable overview alongside the spreadsheet. Users can employ the overview to explore the data at various granularities. Furthermore, they can issue formulae over subsets of data without performing cumbersome scrolling or range selection operations, enabling users to gain a high or low-level perspective of the spreadsheet data. NOAH preserves spreadsheet semantics and look and feel, while introducing such enhancements. Our user study demonstrates that NOAH makes it more intuitive, easier, and faster to navigate spreadsheet data compared to traditional spreadsheets like Microsoft Excel, for a variety of navigational tasks; participants made $2.5\times$ fewer mistakes in NOAH than in Excel while being twice as fast in completing the tasks.

Index Terms—Spreadsheet navigation, data exploration, overview+detail, zooming.

1 Introduction

With a user base of more than one-tenth of the world's population, ***spreadsheets are by far the most popular medium for ad-hoc exploration and analysis of data*** [4]. Studies show that information workers prefer to operate on their data within spreadsheets while shunning enterprise solutions with more advanced analytical features [5], [6]. One popular joke among those developing business intelligence applications is that the "export to excel" button is the third-most commonly used button from the menu bar, after OK and Cancel [7]. Spreadsheets enable users to view, structure, and present data in an intuitive tabular layout, wherein users can map their data and tasks; this tabular layout is essential to the popularity of spreadsheets [8].

Using this tabular layout effectively involves navigation, *i.e.*, *"the process of viewing and manipulating the computer display to show another portion of the information space"* [9]. Navigation is supported via two unit operations, scrolling and steering. Scrolling is the action of moving displayed text or graphics up, down, or across a computer screen, in order to view different parts of the spreadsheet. For example, when analyzing data, users may scroll to compare data across different screens, or to get a high-level view of the overall spreadsheet. Steering, on the other hand, involves clicking the left mouse button and then dragging the mouse pointer through the spreadsheet to select a specific region. For example, to issue a formula, users may steer to select the subset of the data to be operated on as an argument within the formula. Most frequently used spreadsheet formulae require users to perform steering actions [10], [11]. Overall, both scrolling and steering are crucial as users navigate spreadsheets to identify, compare, and summarize data.

However, navigating spreadsheets using scrolling or steering is challenging, since spreadsheet data span multiple screens, making it hard to synthesize, analyze, makes sense of, or operate on it [8], [12]. With the ease of data generation, and with spreadsheets now supporting increasingly larger datasets, *e.g.*, Google Sheets now supports five million cells [13], a $12.5\times$ increase from the previous limit of 400K cells, *navigating data within spreadsheets is only becoming even harder*, thanks to multiple inter-related reasons:

- *Loss of overview and context.* When navigating spreadsheets, users can easily lose the context of where they are and where they should go next [12]. The only navigational context provided by spreadsheets is the built-in scrollbar that acts as a one-dimensional overview and indicates the user's current location on the sheet. However, since this overview does not capture the layout and structure of the data, users are forced to mentally assimilate the layout and recall it on-demand, as they navigate a spreadsheet.
- *Cognitive and mechanical burdens.* The lack of contextual cues leads to severe cognitive and mechanical burdens for users [14]. Users often end up taking their own drastic measures to avoid getting lost; for example, some users create personalized overviews extrinsic to the spreadsheet, by sketching maps of spreadsheets on paper [12]. Other users add their own landmarks such as headers or colored cells, as a visual affordance

- *Sajjadur Rahman, Yuyang Liu, Shichu Zhu and Karrie Karahalios are with the Department of Computer Science at University of Illinois, Urbana-Champaign. Email: {srahman7, yuyangl2, szhu28, kkarahal}@illinois.edu.*
- *Mangesh Bendre is with VISA Research. Email: mbendre@visa.com.*
- *Aditya Parameswaran is with the I School and EECS at the University of California, Berkeley. Email: adityagp@berkeley.edu.*

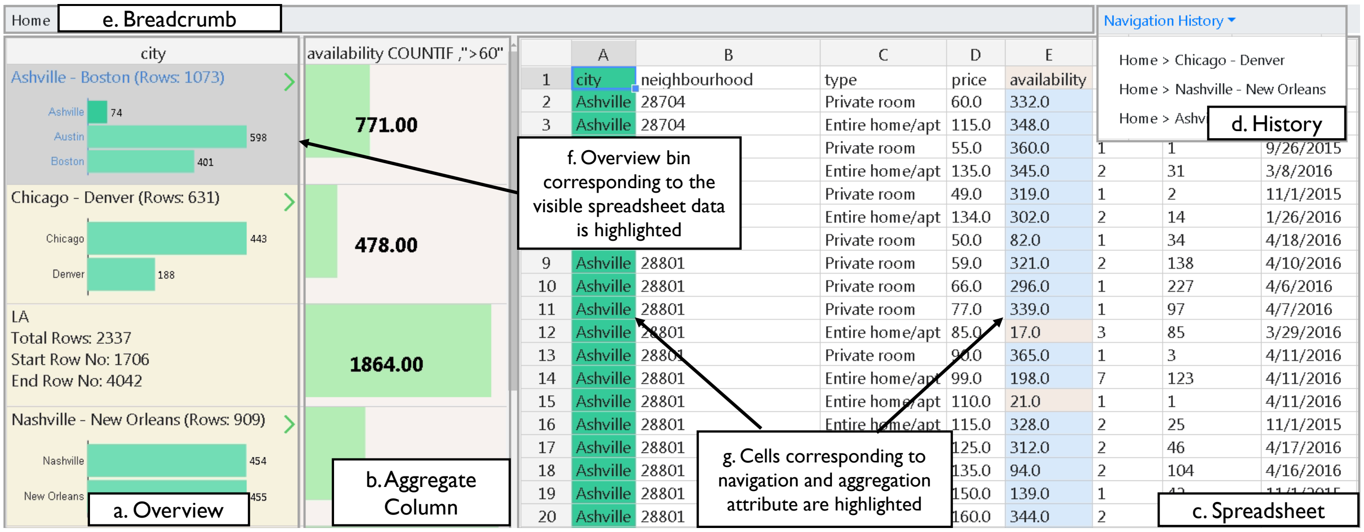


Fig. 1. NOAH: navigation interface consisting of (a) a zoomable overview and (b) an aggregate column integrated with (c) a spreadsheet. A context bar consisting of (d) a navigation history displaying locations visited so far using the overview, and (e) a breadcrumb showing the current navigation path (*e.g.*, Home). (f) The users current focus in the spreadsheet is highlighted on the overview. (g) Columns corresponding to the navigation attribute (city) and aggregate column (availability) are highlighted on the spreadsheet.

to assist in navigation [12]. Steering via dragging the mouse pointer across multiple screens to select a subset of data as input to a formula can often be challenging as well: the only remedy is for users to abandon steering entirely and instead remember the range of the subset of data of interest, and then correctly enter this range as the argument to the formula, often giving rise to errors that are increasingly prevalent in spreadsheets [15].

- *Visual discontinuities.* The limited viewport afforded to the user introduces a visual discontinuity between the information being displayed. For example, comparing spatially separated subsets of data within the spreadsheet requires moving back and forth between multiple viewports, which can be overwhelming [8], [12]. As an alternative, users tend to copy subsets of data side by side to reduce the visual discontinuity [8], [12], which is cumbersome.

Overall, while navigating present-day spreadsheets, ***users often lose context, get overwhelmed, and experience visual discontinuities.*** Addressing these challenges requires considerable manual effort. As we will argue in Section 3, existing spreadsheet features such as pivot tables, named ranges, and subtotals, partially alleviate some of the aforementioned challenges but do not eliminate them entirely. For example, pivot tables generate a summary while losing the correspondence between the raw data and the summary, while named ranges require users to manually associate names with ranges of data.

So, how do we support more effective navigation of data within spreadsheets? One approach would be to try to integrate an overview of the overall structure of the data along with the spreadsheet [16] resulting in a classical overview+detail interface where the spreadsheet is the detailed view. Overview+detail interfaces are used to facilitate navigation in various domains such as text editors and maps [14]. Users can manipulate the overview or detailed view, to perform high-level or low-level operations, respectively. Overview+detail interfaces have been shown to be effective in these domains, reducing cognitive load for users by providing them the big picture first, helping them quickly assimilate the information space [14]. Our goal is to *integrate an overview plug-in with spreadsheets* that captures the overall structure of the data, while supporting interactions that address the difficulties in typical navigational operations like scrolling and steering. ***It is essential that our interface is a plug-in that enhances the capabilities of spreadsheets that so many users are used to and reliant on, as opposed to a potentially jarring or confusing replacement for spreadsheets.***

However, while an overview plug-in for spreadsheets does seem appealing and natural, developing it leads to several challenges.

- *Overview modality.* One could simply add a zoomed out version of the entire sheet as a pane on the side to create a spreadsheet overview, as in popular presentation software like Microsoft PowerPoint, or text editors like Sublime Text. The zoomed out overview would display the data at a lower magnification. Unfortunately, this approach would not suffice for a spreadsheet. An overview should provide a comprehensible big picture view; for a spreadsheet of numbers, text, or formulae, when zooming out beyond a point, an overview displayed at such low magnification would be unreadable. Another approach, adopted by map tools like the early versions of Google maps, is to use the overview to provide a global context of the user's current location currently displayed in the zoomed-in detailed view [14]. While the overview remains static, users can perform semantic zooming operations [17] on the detailed view which allows objects to be represented differently at different scales. Since spreadsheets already display the raw data, zooming into and out of a detailed view consisting of this raw data is not meaningful. How do we design an overview to dynamically change as users seek a more fine-grained or coarse-grained view of the overall structure of the data?

- *Construction of the overview.* Given a spreadsheet with many rows, one approach to constructing a dynamic overview is mapping rows of data to high level groups, similar to online maps. In online maps, cities are grouped into states and states are grouped into countries, forming a multi-granularity hierarchy. How do we automatically group spreadsheet rows together in a similar "meaningful" way such that this grouping applies to all data types, including strings and numbers? If the automatically generated grouping is not semantically meaningful, how do we allow the users to customize the grouping modality? How do we facilitate interactions that enable users to view the overview at multiple granularities?
- *Operations on the overview.* Following the construction of a dynamic overview, the next challenge is to design simple interactions that achieve similar outcomes as scrolling and steering. For example, an alternative to scrolling can be to leverage the groups of the overview to access the rows mapped to that group. As the granularity of the dynamic overview changes, how do we efficiently update the mapping from spreadsheet rows to the finer or coarser groups so that scrolling remains seamless? Similarly, how do we leverage the overview to steer spreadsheet data (i.e., select a range of data) for formula computation? How do we present the results of the formula within the overview? One approach can be to adopt the pivot table-like presentation of results. Within their summary view, pivot tables display aggregate formula results (*e.g.*, SUM, COUNT) alongside each group. However, unlike pivot tables, users can explore the overview at multiple granularities. As the granularity changes, the grouping of rows also changes; making the previous formula results inconsistent with the new groups. How do we recompute the results of a formula in a convenient manner as the granularity changes without requiring the users to reissue the formula from scratch?
- *Seamless integration as a plug-in.* Finally, how can we design an overview with a generic set of features, that can be integrated with any existing spreadsheet tool, operating on any dataset, without impacting existing functionalities or look-and-feel? How do we ensure that the interactions supported by the overview are consistent with traditional spreadsheet semantics, and complement existing spreadsheet interactions? How do we enable coordinated interactions across both views, *i.e.*, the overview and the raw spreadsheet, such that they remain consistent at any given time?

NOAH: a navigation plug-in for spreadsheets. We address the aforementioned challenges in NOAH, an in-situ **n**avigation interface for **o**verviewing and **a**nalyzing spreadsheet data **h**olistically. NOAH is constructed as a plugin to an existing spreadsheet tool, DATASPREAD [18], an open-source scalable web-based spreadsheet. While NOAH's design is not tied to DATASPREAD, we opted not to use other popular spreadsheet tools like Google Sheets and Microsoft Excel because they are closed source. Figure 1 shows a snapshot of NOAH. When the user chooses to explore the data by a specific attribute, a multi-granularity overview is constructed and displayed within NOAH, next to the raw spreadsheet data (Figure 1a). Users can zoom into or out of the overview to obtain a fine or coarse-grained perspective of the data distribution. The distribution at each granularity is captured by a histogram, enabling users to assimilate the data via summary statistics. Each bin (group) of the histogram is mapped to a collection of rows in the spreadsheet. Cumbersome scrolling operations are eliminated in favor of a few clicks on the overview interface. Instead of steering to analyze the data, users can issue formulae on the overview with interactions similar to pivot table construction, and view results on a separate *aggregate column*, alongside the overview (Figure 1b). In this manner, users can issue formulae on different subsets of the data while remaining on the same screen, reducing visual discontinuity. NOAH ensures that there is coordination between the overview and the spreadsheet: for example, panning and zooming on the overview are reflected on the spreadsheet by displaying the spreadsheet data corresponding to the bin currently in focus in the overview. Finally, NOAH automatically creates contextual and historical information (Figure 1d and 1e) while displaying visual cues (Figure 1f and 1g) so that users don't lose context during navigation.

The primary contribution of our work is twofold:

- We formalize the design of a general navigation (overview+detail) interface for exploration and analysis of large spreadsheets. We realize this design in the form of NOAH, a plugin to a spreadsheet tool, ensuring that interactions supported by NOAH complement existing spreadsheet operations.
- We conduct a user study to evaluate the benefits and limitations of this plugin. The study required users to perform tasks that were representative of popular spreadsheet operations, using both Microsoft Excel and NOAH. The study shows that compared to Microsoft Excel, participants were able to complete spreadsheet navigation tasks correctly and quickly in NOAH. Participants made $2.5\times$ **fewer** mistakes while being $2\times$ **faster** with a NOAH-integrated spreadsheet than with Excel.

2 NOAH USE CASES

Users prefer spreadsheets over enterprise solutions to view, explore, and analyze data [5], [6]. To understand the scope of typical user tasks on spreadsheets, we make use of the typology of abstract data exploration tasks [1]—see Table 1. This typology characterizes the range of domain-independent tasks performed on visual representations of data, developed after analyzing task classification systems in over two dozen papers, and has been applied to a variety of scenarios, including developing models for visualization system design [19], designing task taxonomies for cartograms [20], and defining the scope of tasks in a number of domains, *e.g.*, interactive task authoring [21], document mining [22], multivariate network analysis [23], mass cytometry [24], among others. While all the tasks in Table 1 can be performed using spreadsheets, NOAH enhances the experience for many of these tasks, indicated by a checkmark (✓). We describe these tasks in the context of a real usage scenario for NOAH below. Additional details regarding how NOAH complements spreadsheets in accomplishing these tasks can be found in the Appendix.

TABLE 1
Example use cases where NOAH provides benefits beyond spreadsheets (labeled by ✓ if improved; × if it remains the same), employing Brehmer and Munzner's typology [1].

Purpose	Use Cases
Consume	`discover` (✓: *generation of hypotheses*, e.g., Rebecca finds a trend in larger cities and wants to check if it is present in smaller cities), `present` (✓: *communication of information*, e.g., Rebecca sees the overall availability trends in the context of raw listings, and can present this view to her co-workers), `enjoy` (✓: *casual encounters with visualization*, e.g., Rebecca uses the overview "at a glance" to understand which cities are present in the dataset, and how many listings are present per city)
Search	`explore/browse` (✓: *searching based on characteristics where location is unknown/known*, e.g., Rebecca tries to find Chicago listings with availability greater than 60 days), `locate/lookup` (✓: *searching based on entities where location is unknown/known*, e.g., Rebecca wants to find all entries corresponding to a given city like Chicago)
Query	`identify` (✓: *returning the characteristics of entity found during search*, e.g., Rebecca wants to examine Chicago listings to assess typical availabilities of listings in Chicago), `compare` (✓: *returning characteristics of multiple entities*, e.g., Rebecca wants to compare listing patterns in Boston to that of Chicago), `summarize` (✓: *returning characteristics of several entities*, e.g., Rebecca wants to gain an understanding of overall rental patterns across cities)
Produce	`export/save` (×), `generate/record` (✓: *generation or recording of new information*, e.g., Rebecca issues an aggregate formula to generate summary availability statistics across cities)

We now describe a usage scenario that illustrates the benefits of integrating NOAH into typical spreadsheets. Let's assume that Rebecca, a journalist, is exploring the *Inside Airbnb* dataset [25], a dataset of all the Airbnb listings across different US cities. This dataset was created to investigate the long-standing accusation that many listings in Airbnb are illegally run as hotel businesses, while avoiding taxes; any listing available for rent for more than 60 days a year is considered to be operated as a hotel [26].

Given that this is the first time she's examining this dataset, Rebecca wants to first gain a bird's eye view of the data. Without NOAH, Rebecca would have had to use a pivot table (discussed in Section 3) to construct a summary—however, since this summary is disconnected from the underlying data, it is hard for Rebecca to map the summary statistics to the raw data to obtain further details about listings from any given city. If she wanted to examine listings from a specific city, Rebecca would have to either use search capabilities or perform an explicit filter for this information, and would have to switch back and forth between the pivot table results and the raw listings, present at disparate locations on the spreadsheet. Even at the first step of exploration, Rebecca would experience *substantial cognitive burdens, loss of context, and visual discontinuities*, with subsequent steps becoming progressively more challenging.

Using NOAH, she organizes the overview by city and starts casually exploring the dataset, understanding which cities are present, and roughly how many listings does each city have—with NOAH providing a high-level overview of cities (Figure 1a) (`enjoy`). The overview consists of sorted non-overlapping bins containing one or more cities. She can click on any bin and the corresponding data will be displayed at the top of her screen. For example, clicking on the *Ashville-Boston* bin displays the Ashville listings (Figure 1c); she can similarly find and examine properties of the Chicago listings by clicking on the *Chicago-Denver* bin (`locate` followed by `identify`). She can also zoom into bins using the "⟩" arrows, zoom out of bins using the "⟨" arrows, and pan by clicking on various bins at the same level. We discuss the construction of the overview and associated interactions in Section 5.

Next, say Rebecca wants to analyze one of the larger cities to understand the overall renting pattern (`summarize`). She studies a few cities at a time, examining and comparing the number of listings for each city, as displayed on the overview (`compare`). She decides to focus on Boston, her hometown, and wants to find out how many listings in Boston violate the "rent availability > 60 days" condition (`identify`). In a typical spreadsheet, Rebecca needs to manually steer and then select the Boston listings as input to a `COUNTIF` formula that counts the number of rows that satisfy the above mentioned condition. Using NOAH, she can zoom into the *Ashville-Boston* bin (Figure 2a and 2b) and then issues a `COUNTIF` operation on the overview (`generate`). The result is displayed as an *aggregate column* alongside the overview (Figure 1b). Rebecca learns that more than half of the listings in Boston are effectively operating as hotels (`discover`)—a large number!

Based on this insight, Rebecca then wants to understand availability statistics for an even larger city, Chicago (`compare`). As she uses the overview to navigate to Chicago, NOAH automatically updates the aggregate column to the `COUNTIF` formula results for Chicago (`identify`), without Rebecca needing to reissue it by performing a cumbersome steering operation as in traditional spreadsheets. Rebecca learns that Chicago exhibits a similar renting pattern as Boston, with more than half the listings operating as hotels. She can then hypothesizes that this trend may hold for all large cities, and can check whether the smaller cities have a different pattern (`discover`). Note that, the rows that satisfy the "rent availability > 60 days" condition, are listed in the spreadsheet adjacent to the overview in sky blue (Figure 1g) (`explore`). With the raw data presented side-by-side, she can also dive into other attributes of the listings operating as hotels to see if there are any other identifying characteristics, *e.g.*, if they are all managed by a small number of agencies acting as individual renters (`identify`).

Finally, as Rebecca navigates the data, her navigation history (Figure 1d), *i.e.*, recently visited cities, and current navigation path (Figure 1e) are kept up-to-date, allowing her to maintain context during navigation (`record`). She can revisit any previously visited cities (`lookup`) by simply clicking on the relevant path in the navigation history.

Overall, with NOAH, users can quickly comprehend the data via the overview, access any region within the data without having to scroll endlessly, and request additional details on demand without having to steer across multiple screens. As users navigate and analyze the data, they can revisit previously accessed data via the navigation history, not losing context of what they have explored.

3 Related Work

We now discuss tools and techniques that partially address the limitations of navigating data, both inside and outside spreadsheets.

3.1 Spreadsheet Tools and Prototypes

Both commercial spreadsheet tools as well as academic prototypes provide partial solutions to navigational challenges.

Microsoft Excel. Excel enables users to manually create references to a spreadsheet region using the named ranges [27] feature, accessible from the menu bar. Users can click on a named range to navigate to the referred region. However, the onus is on the user to create named ranges for each region of interest. The pivot table [28] feature allows users to create a summary view to compare subsets of data without having to provides a summary view, enabling users to compare subsets of data without having to navigate to various locations within the sheet. This summary is placed in a separate region of the spreadsheet, preventing users from accessing the data underlying the summary, impeding navigation. A similar overview feature, `SUBTOTAL` [29], adds a new row at the end of each distinct subset of data with summary information. Users can expand the summary to view the actual spreadsheet data. However, for datasets with many subsets (*e.g.*, for numeric data), the number of new lines inserted (i.e., the summary) can itself become very large, spanning multiple screens, and can cause increased visual discontinuity during navigation. Finally, NodeXL [30] is a plug-in that provides a spreadsheet network overview and supports navigational operations, *e.g.*, zooming in/out, dynamic filtering, on the overview; this plug-in only supports network datasets, such as biological or social networks.

Google Sheets Explore. Google Sheets Explore [31] provides an overview of the data by auto-generating charts of data statistics. Users can specify queries to the system (similar to a web search) asking for different summary statistics. While Explore is a convenient means to understand high-level data characteristics, it doesn't address the navigational challenges related to scrolling and steering.

Scalable Spreadsheet Summarization and Exploration. Smart-drill-down [32] generates an interactive summary of a large spreadsheet table as a collection of rules; users can drill-down to a specific rule to view more fine-grained rules. Hillview [33] displays the approximate results of group-by queries on large spreadsheet tables. While these tools support summarization at scale, providing an overview of the spreadsheet, they don't preserve spreadsheet semantics, nor do they make it easy to scroll or steer through large spreadsheets. ABC [34] and DATASPREAD [35] support interactive exploration of very large spreadsheet datasets, beyond main-memory limits, maintaining spreadsheet look-and-feel, but do not provide any new spreadsheet capabilities to assist with navigation. We build NOAH as a plugin to DATASPREAD, since it is open-source.

Interactive Tables. TableLens [36] is a focus+context view for browsing numerical information in tables, looking much like a spreadsheet with embedded bar charts. Cells out of focus display graphical bars proportional in length to the underlying values, providing a visual overview of the data, while cells within the user's current focus are magnified and display the graphical bars and the raw data. Ideas similar to TableLens have been adopted by DataLens [37] for visualizing digital calendars, and by FOCUS [38] and InfoZoom [39] for exploring database query results. Like TableLens, NOAH embeds graphical bars, but within the overview to depict the underlying data distribution. NOAH captures the user's current focus by highlighting the corresponding bin in the overview. While TableLens provides an easy mechanism to get a high-level view of the data and spot outliers, it suffers from the same disadvantages that focus+context views have relative to overview+detail ones. Unlike NOAH, which supports multiple granularities via binning, TableLens only supports one granularity (zoomed in or zoomed out): beyond a certain size, navigating (scrolling or steering) the zoomed out data is still cumbersome for users. Moreover, TableLens does not maintain the spreadsheet look-and-feel or capabilities.

Visual Interactive Spreadsheets. VisSh [40], SI [41], SSR [42], ASP [43], and PhotoSpread [44] extend the input/output capabilities of cells within spreadsheets, to display charts, animation, photos, or geometric objects, or accept input via direct manipulation dialogs, among others. While these tools allow users to represent and manipulate data in a more flexible manner, which in turn could help users getting a high-level sense of the data, they do not necessarily help users navigate data more effectively.

3.2 Spreadsheet Alternatives

We draw from work on navigation interfaces in non-spreadsheet interfaces as well.

Overview+Detail Interfaces. Cockburn et al. [14] provides a detailed survey of overview+detail and zooming interfaces. To improve navigation within large documents, overview+detail interfaces [45], [46] allow users to interact with an overview as they explore the document. Zooming interfaces [47], [48] provide a multi-granularity overview of the data and support interactions like zoom in/out to navigate across various granularities. We follow the same analogy of providing an overview of the spreadsheet first, allowing users to drill-down further.

Multiple Coordinated Views. Multiple coordinated views [49], *e.g.*, *Snap* [50], *Elastic Documents* [51] connect multiple views, for example, an overview and a detailed data view while enabling coordination between these views through brushing and linking. Similarly, NOAH connects spreadsheets with an overview and updates the spreadsheet as users interact with the overview and vice-versa.

Tabular Data Analysis (TDA). Visualization tools such as Tableau [52], Power BI [53], Keshif [54], Voyager [55] and analytical tools such as SPSS [56], SAS [57], can all provide summaries of tabular data in various forms (visualizations, aggregate statistics). These summaries are static overviews of the data—much like pivot tables, these summaries are not dynamically linked to nor are co-located with the underlying raw data. For example, Keshif [54] can display all the unique values corresponding to an attribute of interest, *e.g.*, cities of the Airbnb data [25]. However, users cannot view or inspect the raw data corresponding to each city in a spreadsheet-like tabular setting, while being able to edit this raw data at will. With TDA tools, the spreadsheet look-and-feel is lost, and as a result, users lose the ability to directly manipulate raw data, derive new data, and issue formulae for free-form analysis. Therefore, the goals of spreadsheets differ from TDA tools in two ways: a) facilitating direct manipulation of raw data in-situ and b) enabling arbitrary derivation of new data and summaries using various operations involving navigation, *e.g.*, issuing formulae. NOAH being a plug-in to spreadsheets, provides a unified interface that upholds both these goals while enhancing navigational capabilities for spreadsheet users.

4 NOAH: Design Considerations

In this section, we outline our design considerations for a spreadsheet navigation interface. Our design considerations were informed by prior work on information visualization [1], [58], overview+detail interfaces [14], multiple-coordinated views [59], and refined through our experiences across multiple design iterations.

DC1. Construct the overview in-situ. An overview helps users get a high-level picture of the data. However, maintaining the overview in a separate location from the data can lead to loss of context; instead, having it co-located with the data can help users make rapid glances to explore information between a bird's-eye view and a close-up detail [16].

DC2. Ensure reduced visual discontinuity while providing details on demand. Users often need to access subsets of data, and study their properties in detail, *e.g.*, via steering. Navigating back and forth between different subsets of data can lead to increased visual discontinuity. The interface should allow users to compute such details for various data subsets on demand [58]. The interface should maintain visual continuity as users navigate to a different subset, recomputing the details for the new subset.

DC3. Balance the screen space afforded to the overview. As the overview has limited screen-space available, we need to consider the trade-off between visual discontinuity (*DC2*) and clarity. Displaying a fine-grained overview improves visual clarity while increasing visual discontinuity—users need to scroll through the overview to access distant subsets of data. Displaying a coarse-grained overview decreases visual discontinuity at cost of reduced visual clarity—the overview may span too many data subsets and appear visually cluttered. The interface should further allow users to control the screen-space allocated to the overview.

DC4. Enable coordination between the spreadsheet and overview. Since users can view the overview and the spreadsheet simultaneously, interactions on both need to be linked [49], *i.e.*, an interaction on one should be reflected on the other [59]. For example, as a user scrolls through the spreadsheet, the user's current focus should be highlighted on the overview. However, not all interactions need to be interlinked, *e.g.*, changing the font size of a spreadsheet cell need not lead to a change in the overview.

DC5. Facilitate customization of the overview. As the overview is automatically generated, it may not reflect domain-specific context known only to the user [34]. For example, an overview constructed on a grading spreadsheet by binning nearby scores may not match the letter grade ranges that the instructors have in mind. Allowing users to customize the overview is therefore essential.

DC6. Display contextual and historical navigation information. The interface should record navigation history, allowing users to revisit previously visited locations [58], while also displaying their current navigation path for context.

5 User Interface

We now explain the design of NOAH's components and implementation details.

5.1 In-situ Overview

NOAH constructs the overview in-situ (**DC1**) next to the spreadsheet on an attribute of the spreadsheet dataset called the *navigation attribute*, selected by the user. Any attribute type that can be ordered can be a navigation attribute, *e.g.*, text, numbers. The overview is constructed at multiple granularities. Each granularity is divided into non-overlapping groups of data called *bins*. As shown in Figure 2d, an overview of the Airbnb data on the navigation attribute "city" has granularity levels. The highest (coarsest) granularity level consists of four bins. Figure 2a depicts the first four bins, the first of which is *Ashville-Boston*. Each bin contains summary information regarding the data subset/region it spans, *e.g.*, starting row and ending row number, and the total number of rows the region spans. Each bin displays an overview of the next (finer) granularity (if any) with embedded bar charts. For example, in Figure 2d, the topmost bin (*Ash-Bos*) spans three cities (*Ashville*, *Austin*, *Boston*), each of which is a bin in the next (finer) granularity. Correspondingly, Figure 2(a) shows three horizontal bar charts for the first *Ash-Bos* bin, one for each bin in the next granularity. Since the third bin from the top (*LA*) spans only one city, no bar chart is embedded. Users can perform different operations on the bins, *e.g.*, clicking to pan and semantic zooming in/out [17]. NOAH supports other interactions atop this multi-granularity overview, *e.g.*, customization and aggregation. We discuss these interactions in the context of our design considerations in Section 5.1.1.

Why a Multi-granularity Binned Overview? A conventional design for overviews within popular interfaces is as a spatially partitioned collection of thumbnails on the left of the standard detailed view, similar to Microsoft Power Point or Adobe Reader. However, displaying too many thumbnails results in increased scrolling to access distant thumbnails, increasing visual discontinuity. On the other hand, displaying too few thumbnails reduces visual discontinuity, but at the cost of visual clarity—the thumbnails appear cluttered and fail to represent the underlying data clearly [14]. To strike a balance between these two objectives (**DC3**) we designed a multi-granularity overview that abstracts the data at varying levels of detail. Multi-granularity representations have been shown to scale better to larger datasets—-presenting information at multiple granularities makes visual representations more perceptually scalable and less cluttered [2]. Thus, the multi-granularity overview of NOAH provides an alternative to the aforementioned conventional spatially partitioned single-granularity representation of the data space, *e.g.*, in Power Point, by allowing users to control the scale at which the overview should be displayed [14]. Users can resize the overview to control the amount of spreadsheet data that remains visible. Users can also hide the overview if required.

The data structure underlying the overview is an equi-depth histogram constructed on the values in the navigation attribute column.AGP: Briefly describe what equi-depth histograms are; or don't mention it here, just say histogram

– we can defer details later Equi-depth histograms are commonly used for summarizing the statistical properties of data, with applications such as database query optimizers [60]. AGP: only one application? The binned representation of the overview using equi-depth histograms aggregates data and visualizes density by counting the number of data points falling within each bin. AGP: This previous sentence feels repetitive. Binned aggregation highlights both high level (*e.g.*, densities) and low level (e.g., outliers) details of large datasets, while enabling multi-granularity representation of data via the choice of bin size [3]. AGP: Clarify what is known from prior work and what you are proposing. Here, it is intermixed. Moreover, binning reduces visual discontinuity during navigation as users are able to view an overview that fits in the computer screen. As a result, users can quickly navigate the data—the bins act as landmarks in the overview, enabling users to skip irrelevant bins and quickly navigate to the desired subset of data. We now discuss how the overview is constructed.

Overview Construction. The equi-depth histogram can be constructed on any data types that can be ordered, *e.g.*, text, numbers, dates. For example, in the usage scenario explained in Section 2, the journalist grouped the data into cities for ease of navigation when exploring the larger cities in the Airbnb dataset. Each bin in the equi-depth histogram contains the same number of items, where each item is a value. For example, when constructing the overview on city, each value in the city column is assigned to a bin. The bins are constructed top-down (see Figure 2d). NOAH divides each of the bins at level k into new bins to construct the next lower level $k+1$, again, by applying the same concept of equi-depth histograms. If each value of the navigation attribute column was unique, *e.g.*, if it was a numerical ID, then construction of the histogram would be easy: each bin of the equi-depth histogram would contain almost the same number of items, where each item corresponds to one unique value of the attribute. Unfortunately, in practice, for many attributes, the same value is often repeated. For example, there are multiple listings per city. Therefore, an equi-depth histogram on the attribute city will result in consecutive bins sharing items of the same unique city value, resulting in undesirable overlap. Instead, we construct a best effort equi-depth histogram that is as close to an equi-depth histogram as possible, while ensuring that the ranges represented by each bin have no overlap. AGP: Aditya: would be valuable to describe the algorithm to increase technical depth. Saj: can we add a pseudocode in appendix and refer to that? AGP: Yes please.

5.1.1 Operations and Interactions

We now discuss the operations and interactions that can be performed on the overview.

Navigational Operation: Clicking. When a user clicks on a specific bin, NOAH displays the corresponding spreadsheet data; users can use this to jump to a specific spreadsheet location without having to scroll endlessly. For example, in Figure 2b, as the user clicks on the *Boston* bin, the data corresponding to Boston is displayed (Figure 2c). Note that the click operation is different from the traditional spreadsheet `Filter` operation. `Filter` hides spreadsheet data that do not satisfy the filtering condition while clicking brings the desired subset of data in view without hiding the rest. Users are free to navigate to other portions through scrolling even after clicking a bin, unlike filtering, where users need to issue another `Filter` to bring other data into view.

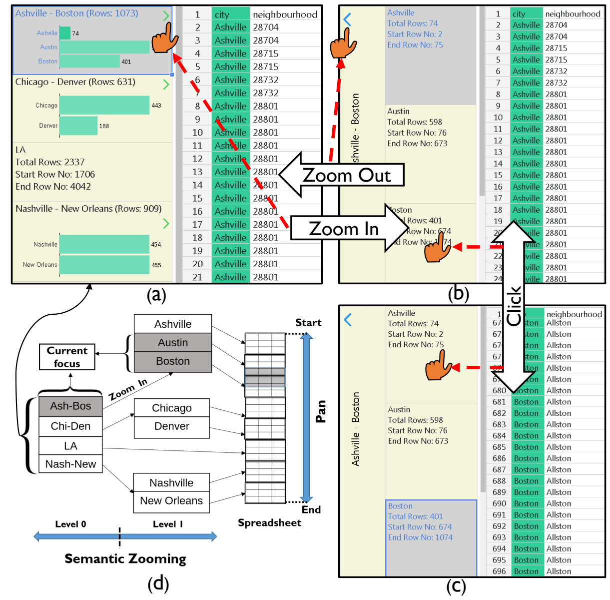


Fig. 2. Navigational operations. (a) The overview at the highest level of granularity. (b) A zoomed in view of the *Ashville-Boston* bin. (c) As the user clicks on the *Boston* bin, the Boston listings are displayed on the sheet. The *Boston* bin is highlighted in gray to indicate users current focus. (d) Conceptualizing the multi-granularity overview.

Navigational Operation: Semantic Zooming. Users can zoom into a specific bin to view more fine-grained information or zoom out to view more coarse-grained information, via semantic zooming [17]. For example, in Figure 2a, from the bin *Ashville-Boston* when the user zooms in to the next level, NOAH displays the bins *Ashville*, *Austin*, and *Boston* (Figure 2b). If the user zooms out of the current granularity, again NOAH displays the bins *Ashville-Boston*, *Chicago-Denver*, and others. Users can only zoom into any bin that contains multiple unique values. For example, in Figure 2d, at level 2, each bin corresponds to one city. Therefore, users can only click on those bins to bring that data into view, and cannot zoom in further. One issue with zooming interactions is discoverability of the zoom operation [14]. We circumvent this (see Figure 2c) by providing the root of the bin under selection for zoom out, and arrows for clicking to zoom in ("〉") and out ("〈").

Customizing the Overview. As NOAH constructs the overview automatically, the overview binning or organization may not capture domain-specific context or user needs. NOAH enables users to customize this organization (**DC5**). At any granularity, users can merge multiple consecutive bins into a single bin, or split a bin into multiple bins. Say the user wants to compare summary statistics of Boston and Chicago. In the current organization these two cities are in two different bins (see Figure 3a). Using the bin customization feature, the user can merge the two bins *Ashville-Boston* and *Chicago-Denver* to create a new bin *Ashville-Denver*. Users can now zoom into this bin and compare summary statistics of the cities in the same view. The interactions for splitting a bin depend on the data type. If the navigation attribute is textual, any bin can be split into as many bins as the number of unique values that bin contains. If the navigation attribute is numeric, users can split the bin into any arbitrary number of bins. Note that NOAH does not allow users to rearrange the order of the bins. Since the overview represents a histogram, the bins are ordered—reshuffling the bins violates that order.

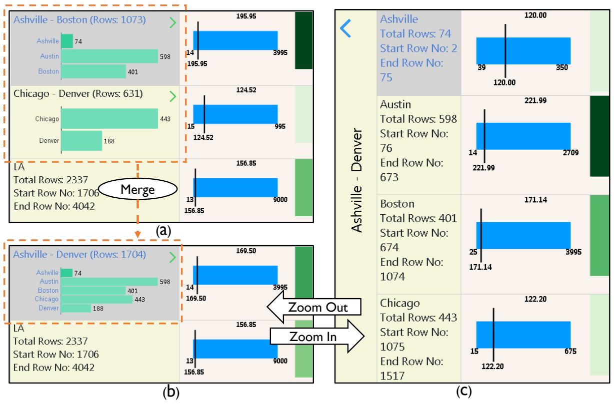


Fig. 3. (a) Chart view of the aggregate column. (b) A new bin is created by merging the top two bins. (c) Zooming into the newly created bin.

5.1.2 Coordination Between Overview and Spreadsheet

NOAH supports coordination between the overview and the corresponding spreadsheet data (**DC4**), *i.e.*, interactions on the overview may be reflected on the spreadsheet and vice-versa. One example of this coordinations is indicating the navigation attribute on the spreadsheet using color (see the lime green column in Figure 1c) as user constructs the overview. However, not all overview interactions are coupled with the spreadsheet and vice versa. The coupling depends on the user's current focus—*to ensure consistency between the overview and the spreadsheet, any interaction on either interface that changes the current focus must be reflected on the other interface*. We now provide examples of both coupled and decoupled interactions.

Coupled interactions. Clicking a bin is an example of a coupled interaction as the user actively changes the focus to another bin on the overview. To reflect the change, NOAH populates the corresponding spreadsheet data on the screen. As the user scrolls on the spreadsheet, again the current focus changes and the corresponding bin on the overview is highlighted. For example, in Figure 2c, as the user clicks on the Boston bin, the spreadsheet displays the Boston listings. Conversely, as the user scrolls up, both Austin and Boston listings appear in the current window of the spreadsheet. Therefore, both the Austin and Boston overview bins are highlighted (see Figure 2d).

Decoupled interactions. When a user zooms into a bin that is already in the user's current focus, the spreadsheet view does not change. For example, in Figure 2a, the user zooms into the *Ashville-Boston* bin; here, the spreadsheet view stays the same (see Figure 2b). Similarly, the zoom out operation is decoupled. When a user zooms out, the overview displays a coarser granularity view of the user's current focus. Since the focus stays the same, there's no need to update the spreadsheet view. Similarly, operations like panning on the overview without clicking, and customizing the overview do not change user's current focus and are therefore decoupled. Online maps also adopt similar decoupling of the overview and detail [14]. However, their goal is to reduce network and computational overload, whereas in our case, the decoupling is based on the user's current focus.

5.2 Aggregate Columns

Users can issue spreadsheet formulae on the overview to compute aggregates for the data in each bin. The results are displayed as an *aggregate column* (see Figure 1b). Each entry in the aggregate column corresponds to the adjacent bin in the current granularity of the overview. For example, in Figure 3c, the aggregate column displays four aggregate statistics, one per bin. Users can issue several formulae simultaneously, each giving rise to a new aggregate column. However, adding an aggregate column takes up screen space, shrinking the spreadsheet view. As a workaround, users can resize or remove aggregate columns if required (**DC3**). When the user issues a formula on the overview, the spreadsheet column corresponding to the aggregate column is highlighted in grayish orange (see Figure 1c)—another example of coupled interaction. For conditional formulae like `COUNTIF`, cells that satisfy the condition are highlighted, *e.g.*, in Figure 1c, the cells with availability ≥ 60 are colored in sky blue. In this manner, users can quickly determine which cells are relevant to the aggregation operation.

Creating an aggregate column on the overview mimics how users create pivot tables. Users are not required to explicitly type formulae; rather they simply select the formula from a drop-down menu, and provide the necessary formula parameters to a form. The aggregate column can employ any statistical or mathematical formulae that operate over a range of data. Therefore, creating an aggregate column is equivalent to selecting subsets of data on the sheet, *i.e.*, steering, and then executing a formula on this subset, helping users avoid cumbersome steering operations. We have classified the formulae supported into five categories: a) summary (*e.g.*, min, max, average), b) frequency (*e.g.*, mode, large, small), c) conditional (*e.g.*, countif, sumif), d) spread (*e.g.*, var,stdev), and e) others (*e.g.*, sum, count).

Users can view the results either as raw values or as charts, and can toggle between the two. Raw values are displayed along with a colored bar, the *value bar*, whose length is proportional to the corresponding aggregate (see Figure 1b). Users can use the lengths to visually compare across bins. The chart representation varies depending on the formula type. All other categories except for the *others* category can be represented by charts. Figure 4 shows the chart representation for these categories along with different visual cues that highlight formula results as well as other information. We discuss these representations in detail in the Appendix.

Finally, we note that the aggregate column is kept in sync with the bins as users zoom in and out, eliminating repeated steering operations. NOAH does not maintain any additional data structure for the aggregate column. The histogram underlying the overview records the result of the aggregate column entries corresponding to the bins. Next, we discuss how NOAH maintains user's navigational context.

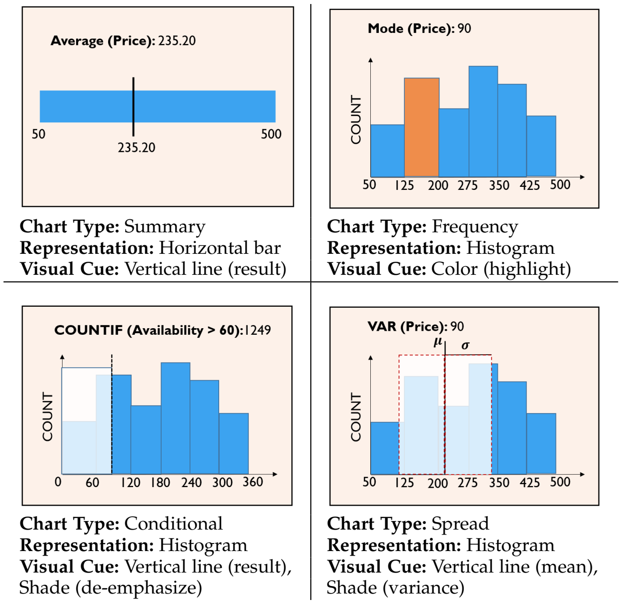


Fig. 4. Formula types and their chart representations.

5.3 Context Bar

The context bar consists of two components: a) a breadcrumb, and b) a navigation history. The breadcrumb [61] displays the current navigation path (see Figure 1e), thus maintaining the users' navigation context (**DC6**). Each component of the breadcrumb corresponds to a bin in the user's current navigation path. Therefore, users can visit any bins within the current navigation path by clicking on an appropriate component of the breadcrumb, without having to zoom in or zoom out. NOAH also maintains a list of recently visited bins (**DC6**) (see Figure 1d).

5.4 Implementation

We have integrated NOAH with DataSpread [35], a web-based spreadsheet. The DataSpread back-end maintains the histogram data structure and supports the aggregate column computation via its built-in formula engine. The NOAH front-end is built with HTML/CSS/JS technologies along with the D3 framework [62] for generating charts.

6 Evaluation Study Design

In this section, we present the design of a user study to evaluate whether NOAH helps address spreadsheet navigational challenges.

6.1 Study Design and Participants

The goal of our study is to evaluate whether NOAH addresses the navigational challenges in traditional spreadsheets. Furthermore, we wanted understand how the presence of the features introduced by a plugin like NOAH impacts a spreadsheet user's navigation experience as they explore and make sense of data. Therefore, we decided to compare a NOAH-integrated spreadsheet with a typical, popular one, Excel, for various navigational tasks in the spreadsheet domain. Similar domain specific-evaluations have been performed for measuring the performance of various overview+detail interfaces, *e.g.*, database browsing [50] or tree navigation [63]. As explained in Section 3, the goals and user populations of spreadsheets and TDA tools are quite different. Therefore, we did not consider TDA tools for the comparative study. Our study was designed to answer the following questions: AGP: The RQs feel repetitive. What did Karrie think. She was fine with it. I think there are subtle differences. the RQ1 tries corresponds to quantifiable performance metric like accuracy, latency. RQ2 is qualitative feedback of user experience.

- **RQ1.** How did the addition of an overview like NOAH impact a user's spreadsheet navigation performance measured in terms of navigation task submission speed and accuracy and quantitative survey metrics such as speed of use?
- **RQ2.** How did NOAH and it's components affect spreadsheet users' navigational experiences?

Study Design. We conducted a 2×2 (2 datasets, 2 tools) mixed design within-subject study. The two tools used in the study were: Microsoft Excel, and NOAH integrated within DataSpread [18]. We chose Excel for our comparative study because it is the most popular spreadsheet in use today. The study consisted of three phases: (a) an introductory phase explaining the essential features of NOAH via a video tutorial, followed by a warm-up session where participants explored a flight dataset [64] in NOAH to familiarize themselves with its features, (b) a quiz phase where the participants first used both the tools to perform targeted tasks on two different datasets (described later) followed by a survey to provide feedback on their impressions about both Excel and NOAH, and (c) a semi-structured interview to collect qualitative data regarding the quiz phase.

Datasets. We used two datasets—the birdstrikes (used in Keshif [54] and Voyager [55]), and the Airbnb [25] datasets. These datasets were chosen for their understandability to a general audience. The birdstrikes dataset records instances of birds hitting aeroplanes in different US states. The dataset has 10,868 records and 14 attributes (eight categorical, one spatial region, one temporal, four numeric). The Airbnb dataset was larger than the birdstrikes dataset. To ensure a fair comparison across tools, we created a sampled version of the original Airbnb dataset with 10,925 records, by uniformly sampling 10% of the records from each US city. This dataset contained 15 attributes (six categorical, two spatial region, one temporal, six numeric).

Participants. We recruited 20 participants (11 female, 9 male) via flyers across the university and via a university email newsletter. The average age of the participants was 31.06 years ($\sigma = 12.44$). The participants came from different backgrounds, *e.g.*, engineering (seven), business (five), administration (five), and natural science (three). During recruitment, prospective participants filled out an interest form where they answered questions about their spreadsheet expertise, types of spreadsheet tasks performed, and usage of spreadsheet operations. Participants were asked to rate their expertise with different spreadsheet software, *e.g.*, Excel and Google Sheets, and their frequency of using various spreadsheet tasks *e.g.*, data management, data analysis, statistical modeling, and what-if analysis. We also asked participants about their familiarity with basic mathematical and statistical spreadsheet functions as well as advanced operations, *e.g.*, pivot table, `SUBTOTAL`, and conditional formatting. To ensure that participants experience with spreadsheets didnt affect their performance during the quiz phase, we only recruited participants who rated their experience with Excel to be greater than four on a scale of one (no expertise at all) to five (very experienced). The selected participants were familiar with performing various tasks on spreadsheets, *e.g.*, maintaining, tracking, and analyzing, making predictions, and performing comparisons. All of the participants were familiar with the basic mathematical and statistical functions supported by Excel. Each participant received $10 per hour at the end of their session.

6.2 Study Procedure

We now explain each of the phases of our study in more detail.

Phase 1: Introduction to NOAH. We began the study by showing a six-minute video tutorial explaining the features of NOAH on a dataset of all the flights across the US for January 2018 [64]. The participants then explored the same dataset using NOAH to familiarize themselves with the tool for about 10 minutes. The quiz phase began as soon as the participants finished their exploration.

TABLE 2
Quiz tasks for the birdstrikes dataset. The use cases correspond to the task typology discussed in Section 2

Category	Question	Use case
steer	Organize the data by State. How many flights that had damages (damage = 1) originated from Florida?	`summarize`
identify	How many flights in the currently visible spreadsheet window have damages?	`identify`
steer	How many flights that had damages originated from California?	`summarize`
compare (2)	Which state between Florida and California has a higher number of flights with damages?	`compare`
compare (N)	Find the state with the most birdstrike occurrences.	`compare`
customize	Organize the data by *altitude*. What is the average cost of damages for altitude bin 0-450?	`generate`

Phase 2: The Quiz Phase. The purpose of the quiz phase was to evaluate the effectiveness of the features supported by NOAH in addressing spreadsheet limitations. During the quiz phase, each participant performed specific tasks on the two datasets in two sessions, using Excel for one and NOAH for the other. Each session was followed by a survey, described later. We alternated the order of the datasets between consecutive participants. The order of the tools was alternated between every two participants. We developed an online JavaScript-based quiz system that recorded user responses and submission times. We also recorded the participants interactions with both tools using screen capture software. Participants were informed that they can refer to the Internet for help as many times as they wanted. However, due to their familiarity with Excel, none of the participants required external help. For reference, we also provided a printed handout to the participants that contained screenshots with the features of NOAH.

Quiz Tasks. We designed six tasks grouped into four categories: steer (two tasks), compare (two tasks), identify (one task), and customize (one task), representing the Table 1 uses cases `summarize`, `compare`, `identify`, and `generate`, respectively. All of the tasks except the customize task, are representative of the navigation interactions required for the most frequently issued spreadsheet formulae [10], [11]. The customize tasks was designed to mimic the scenario where a user would group data subsets of interest and required participants to utilize the bin customization feature. The tasks were presented in the same order as shown in Table 2 for the birdstrikes dataset. The questions for the Airbnb dataset were similar. The order of the questions mimics the spreadsheet navigation scenario presented in Section 2.

The steer tasks required participants to use the aggregate column feature in NOAH as opposed to steering in Excel. The identify task required participants to interact with the detailed view and relate the aggregate result of the first steer task with the raw spreadsheet data. The compare (2) task asked the participants to compare the results of the first steer task (*e.g.*, Florida) with that of the second steer task (*e.g.*, California) which would require them to use the context bar to revisit a previously visited bin. The compare (N) task involved comparing statistics of all the data subsets. The purpose of the compare (N) task was to see how increasing the number of subsets would affect navigation. As explained earlier, the customize task was designed to evaluate the utility of bin customization feature.

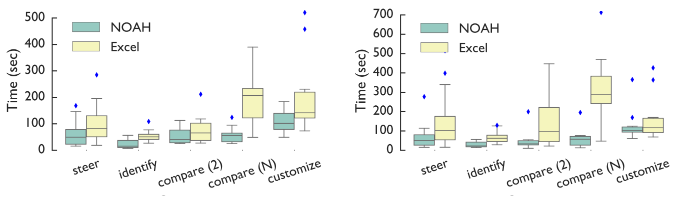
Survey. After each session, participants rated the corresponding tool used on six metrics: confidence, comprehensibility, level of satisfaction, ease and speed of use, and ease of learning for spreadsheet navigation, on a Likert scale from one (*e.g.*, strongly disagree) to seven (*e.g.*, strongly agree). The survey asked multiple questions related to these metrics, 15 in total, to ensure reliability. Participants were also asked to mention the positive and negative aspects of both tools. The survey was designed to evaluate *RQ4*.

Evaluation. We evaluated the accuracy (either 0 or 1) and completion time for each of the six tasks. We combined this analysis with qualitative survey, interview, and screen/audio recording data to provide insights that can be explained with multiple information sources. For example, we analyzed the video recordings of participants' interaction with the tools during the quiz phase. We further analyzed the survey responses to quantify the usability of both the tools.

Phase 3: Interview Phase. Following the survey, we conducted a semi-structured interview to identify participants preferred tools for different tasks and to understand the reasoning behind their choices. We also asked participants to comment on the usefulness of different features provided by NOAH and Excel.

6.3 Study Limitations.

Our study has several limitations that can be strengthened by future larger-scale and more fine-grained studies. One limitation arises from our participant demographics. We conducted the study with 20 participants from four different backgrounds. Our participant pool demographics partially represent the demographics of the general audience intended for NOAH. A larger sample with more participants that better represented the spreadsheet user population would have provided more ecological validity to generalize our findings.



(a) birdstrikes **(b) airbnb**

Fig. 5. Submission times per category for each dataset. Median submission times are much smaller for NOAH compared to Excel.

Another limitation of the study is that we only compared the performance of a NOAH-integrated spreadsheet with a traditional spreadsheet. We did not evaluate specific spreadsheet features like pivot table and `SUBTOTAL` due to their limitations in the types of interactions supported (see Section 3). Moreover, instead of forcing the participants use unfamiliar spreadsheet operations, we wanted them to utilize the typical spreadsheet operations that they are comfortable with, to complete the tasks and see how introduction of NOAH affected their navigation experience.

Finally, we did not isolate the effects of the individual features of NOAH to better understand the implications of those. For example, the effect of binned overview (visual clarity versus visual continuity), display layout (screen space trade-off), and contextual presentation of data (raw text versus chart representation of aggregate columns). However, the goal of the study was to understand participants' navigation experience in the presence and absence of NOAH. A more fine-grained study can be conducted in the future, to inspect the contribution of individual components of NOAH in further detail.

7 Results

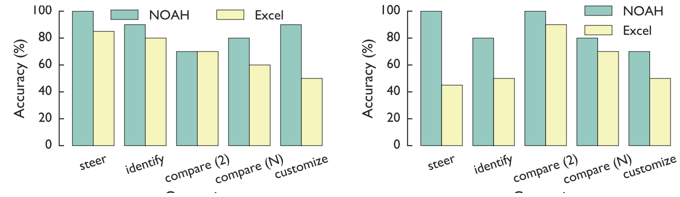
In this section, we analyze the quantitative and qualitative data collected during the quiz and interview phases to address our research questions.

7.1 RQ1. Navigation performance of participants

To answer RQ1, we first compare our participants task completion times and accuracies in NOAH and Excel while analyzing the survey response.

7.1.1 *Faster navigation without sacrificing accuracy*

In Figure 5a and 5b, we show the distribution of submission times of participants for the four task categories, for birdstrikes and Airbnb respectively. For most categories, *participants' median submission times using* NOAH *were less than the fastest submission times using Excel*. This suggests that the new capabilities offered by NOAH made spreadsheet navigation faster for these tasks. We also analyzed the intra-participant submission time difference, and that supported the aforementioned observations. Overall, majority of the submission times using NOAH were faster compared to Excel—19 out of the 20 participants completed at least four tasks in less time using NOAH compared to Excel. The submission time differences were more prominent for the steer, identify, and compare (N) tasks. Moreover, for all tasks except customize, the difference in submission times was statistically significant. Both the intra-participant difference and the statistical significance test results are discussed in detail in the Appendix.



(a) birdstrikes **(b) airbnb**

Fig. 6. Per category accuracy for each dataset. Participants attained higher accuracy while completing tasks in NOAH compared to Excel.

In Figure 6a and 6b, we show the percentage of correct submissions for the four quiz task categories, for the birdstrikes and Airbnb datasets, respectively. For all the tasks except for the fourth task, compare (2), for which the accuracy was the same for both tools, participants attained slightly higher accuracy with NOAH compared to Excel. However, the difference in accuracies was statistically significant for the steer tasks only (see Appendix).

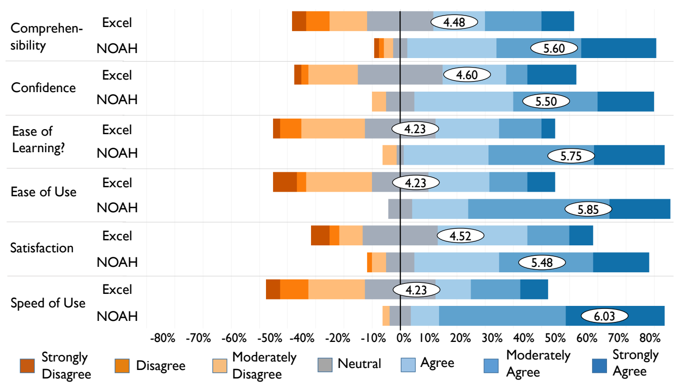


Fig. 7. Participants found NOAH to be easier to use compared to Excel while being faster in completing tasks involving navigation.

7.1.2 *Participants' preferences:* NOAH *VS Excel*

Figure 7 shows a diverging stacked bar chart representation of the survey results in which participants rated their experience with Excel and NOAH. For each metric mentioned in Section 6, there are two stacked bar charts, one for Excel and one for NOAH. Each component within a stacked bar represents the percentage of responses for the corresponding rating, where the ratings are on a scale of one one (strong disagreement) to seven (strong agreement). The average rating for each metric is represented with a white ellipse. Notably, NOAH had a higher average rating than Excel for all the metrics. The aforementioned observation was further validated by a statistical significance test—the *Wilcoxon Signed-rank Test* (see Appendix). In particular, participants felt that using NOAH was faster and easier compared to Excel.

7.2 RQ2. Impact of NOAH and its components on spreadsheet navigation

To answer RQ2, we assessed how NOAH components impacted navigation in our study.

7.2.1 Binned Overview

Data navigation at scale. Without an overview, participants found it difficult to perform various tasks in Excel. One participant ($P11$) commented—"*Excel can get overwhelming if you have a lot of data in it and sometimes with that data finding things can be difficult*". Participants ($N = 6$) mentioned that they would prefer NOAH over Excel when the dataset is large: "*If I just had a large amount of data then I would prefer to use* NOAH *because then you would be able to see all of it (bins) at once*" ($P2$). NOAH's binned overview helped participants comprehend the overall structure of the data better and prioritize the bin they want to visit next, enabling faster navigation. One participant ($P5$) commented: "*I think it was just a little bit easier to navigate and find where things were because you could already see what bins had what.*" Another participant ($P1$) said: "*I like* NOAH *a lot better. It was a lot easier to look up different data and it was a lot quicker too*".

Enhanced navigation with guided bin customization. The bin customization feature enabled users to personalize the overview based on their specific needs. One participant ($P16$) commented: "*I did like the fact that it lets you take a data sheet and, in some way, containerize the stuff you care and the stuff you don't care about.*" 14 out of 20 participants found the bin customization feature to be useful. Participants preferred the feature to Excel's filtering feature when working with numeric data—"*That was so much easier in* NOAH *than it was in Excel to be able to specify the range that you wanted it to go in*" ($P17$). Our analysis of the video recordings revealed that, for the birdstrikes dataset in Excel, the customize task involved filtering out the desired values from a total of 451 unique values. As a result, participants had to manually filter a large number of values and took more time to submit their responses compared to NOAH when they were able to use the bin customization feature. However, the time taken for this task was higher than other tasks in NOAH, as it required participants to restructure the overview before any calculation could be performed. Unfamiliarity with customization operations also contributed to higher task completion times.

Unfamiliar interactions. Participants were unfamiliar with the bin customization feature. This unfamiliarity led to some participants ($N = 5$ out of 20) preferring Excel over NOAH. One participant ($P11$) commented: *Since I'm not used to spreadsheet data being presented that way, it took a little bit of getting used to.* Participants found some of the terminology used in the interface—*e.g.*, explore, bin—quite unfamiliar ($N = 14$). Moreover, two participants didnt understand how the bins were constructed.

Overview presentation across data types. Although participants appreciated the binned representation of the overview for numeric data, a number of participants ($N = 6$) stated that they would've preferred a pivot table-like single level overview for categorical data where each bin corresponds to one item. One participant ($P13$) commented: "*I would prefer it start with all the bins split, and then I can merge them as I want.*" Another participant ($P4$) said—"*When I started, it* (NOAH*) had already grouped them, I think, alphabetically. So, that creates an extra step in that I then have to go split them and then re-merge them.*"

7.2.2 Aggregate Column

Ease of issuing formulae. The steer tasks required participants to issue a `COUNTIF` formula on a data subset. Participants found scrolling and steering in Excel to be cumbersome while issuing formulae—"*The one thing with Excel is I always try to go to the bottom of the data and type in the formula, and with something really long like this, the scrolling is a little bit cumbersome*" ($P4$). With NOAH, participants avoided (a) scrolling by using clicking or zooming operations, and (b) steering by performing aggregate operations on the overview. Participants ($N = 13$) found it easier to issue formulae using the aggregate column feature. One participant ($P3$) commented: "*And that creates convenience sort of because then you don't have to memorize anything and using the system becomes easier.*" Another participant ($P13$) commented: "*There were some formulas to calculate, that were definitely easier in* NOAH *because the aggregate column did all the work and showed me the results.*" However, two participants found the aggregation operations applied on the bins to be opaque compared to Excel where a user can directly manipulate the formula.

Higher efficiency in issuing formulae. While the accuracies and submission times for the steer tasks in Excel varied significantly across datasets, using NOAH, participants exhibited higher accuracies and faster submission times irrespective of the dataset (see Figure 5 and 6). The automated and steering-free aggregate column feature of NOAH contributed to high accuracies (100%) for the steer tasks. One participant ($P12$) commented: "*With* NOAH, *you don't have to highlight every number versus Excel where you actually have to select everything.*" All of the 14 inaccurate submissions with Excel involved steering incorrect spreadsheets regions. 11 of the inaccurate submissions were with the Airbnb dataset. However, in NOAH, participants were able to avoid steering by utilizing the aggregate column feature. Analysis of screen recordings of Excel usage revealed that for birdstrikes dataset, several participants used the *autosum* feature to quickly count the number of 1's in a binary-valued column that was involved in the steering task. Summing up binary values is equal to the number of 1's in the collection. Other participants used the status bar at the bottom of the spreadsheet that displayed the sum of the cells in the selected column. In both cases, participants avoided steering the data. On the other hand, for Airbnb dataset, participants could not use these shortcuts as the column that was involved in the steering task was non-binary (it had 365 different values). Participants, at times, steered incorrect regions, resulting in inaccurate responses for the task. Therefore, the participants ability to avoid steering depended on the data type. Failure to avoid steering often led participants to selecting an incorrect range of data ($N = 14$), resulting in incorrect responses.

7.2.3 Detailed View

Accelerated data inspection. For the *identify* task, participants had to skim through all the cells in the current window in Excel, resulting in higher completion times. Even though Excel provides a conditional formatting feature, that adds one additional step when performing the *idenify* task. In NOAH, participants benefited from having visual cues in the form of colored cells, helping them relate the aggregate column with the raw data—*You didn't have to do any additional steps and it was a visual cue right there, made it very quick to count it up* ($P17$). Another participant ($P9$) commented—"*In Excel, you would have to do your own condition of formatting. But you have to build that every time you need to ask a question. This one* (NOAH) *at least something is pre-built in, and you can easily count.*" However, one participant ($P3$) pointed out the fact that, when the data corresponding to the bin does not fit in the screen, they had to scroll through the data to identify relevant information.

7.2.4 Context Bar

Utilizing history to avoid repeated execution. For the compare (2) task, participants utilized the context bar to navigate to the previously visited bin for the first steer task. As NOAH automatically materialized the aggregate summaries of a previously visited bin, participants were able to view the aggregate column values instantly without having to repeat the same command. However, as Excel did not preserve any navigation history, participants had to re-execute the first steering operation. As a result, the submission times for compare (2) tasks were lower in NOAH compared to Excel (see Figure 5). One participant ($P16$) commented—"*Once I got familiar with the interface, it was easy to just say, I want to see this state, and I like that fact that like automatically it goes into the bins on NOAH, gave me summary information.*" Another participant ($P9$) said—"*Noah was easy to find and compare and toggle in between.*"

Visual discontinuity during navigation. For compare (N) tasks, participants had to perform N comparisons in NOAH while issuing the aggregate column operation once. However, comparison among N bins resulted in increased visual discontinuity. This lead to some ($N = 4$ out of 20) incorrect submissions. In excel, the experience was worse, as the participants had to perform N steering tasks. As a result, in Excel, compare (N) task submission times were very high compared to compare (2) tasks (see Figure 5). In addition, the accuracies of the compare (N) task in Excel were extremely low.

8 Discussion and Conclusion

In this paper, we have introduced NOAH, an in-situ navigation interface, designed as a spreadsheet plugin. Using NOAH, users can get a birds-eye view of the data, with the ability to scroll or seek additional details on demand, using a multi-granularity overview, as well as aggregate columns that eliminate cumbersome steering operations. Quantitatively, we find NOAH to speed up navigation without compromising the accuracy of the tasks. Qualitatively, study participants identify it as positively impacting their experience while overviewing and navigating large datasets, and issuing formulae. However, participants identified unfamiliarity and lack of transparency of operations as some of the limitations of NOAH. In this section, we discuss how to address these limitations and highlight other future enhancement opportunities for NOAH.

Transparency and documentation. Several operations in NOAH are quite different from typical spreadsheet interactions, *e.g.*, zooming, bin customization. Moreover, participants found some of the terminologies quite different from typical spreadsheet terminologies (see Section 7.2). In addition, some participants complained about the lack of explanation surrounding the overview construction and aggregate column computation. In the future, these issues can be addressed by using more relatable terminologies and improved documentation.

Maintaining spreadsheet look and feel. In subsequent versions of NOAH, we can further display the appropriate formula for each bin as users hover over the corresponding cell on the aggregate column. Moreover, participants ($N = 5$) noted the fact that NOAH currently does not support user defined formulae, another possible future enhancement. Bin customization in NOAH is performed using a menu bar which adds an additional step. In Excel, the cell splitting and merging operations are direct and only requires a single click. Similar direct adjustment of data grouping strategies have been explored for visualization tools [65] and can be introduced in a future version of NOAH.

Binning for different data types. The experience surrounding the construction of the overview can be further improved, specially for categorical data. Currently, the bins of the overview can be customized only after the overview is constructed. Providing the users the capability to select the representation (similar to bin customization) of the overview could have addressed this issue. Understanding the impact of these representation choices for the overview can be an interesting future research.

Scope of overview-spreadsheet coordination. Spreadsheet users may perform various edit operations, *e.g.*, updating values, adding/deleting rows/columns. However, NOAH currently assumes the data to be read-only. In our next version, we can add support for propagating the spreadsheet updates to the overview. Moreover, the charts displayed in an *aggregate column* are non-interactive, *i.e.*, users cannot interact with the charts to visually look up relevant or interesting data points within the spreadsheet. In the future, we plan to extend NOAH to support visual querying through the charts in an aggregate column, similar to multi-modal linked selections in Keshif [54].

Enhancing the navigation experience. NOAH currently constructs the overview on a single attribute. We can add support for multi-attribute navigation (*e.g.*, explore the Airbnb data by city and neighborhood), and multi-level navigation (*e.g.*, explore the neighborhoods after zooming into a specific city in the Airbnb data). Furthermore, bin customization currently supports changing the bin boundaries only while maintaining the current order. Supporting user defined ordering to allow arbitrary reshuffling of the bins can be another enhancement.

Expanding the scope of supported operations. Spreadsheet operations that involve working with subsets of spreadsheet data, *e.g.*, sorting, filtering, copy-pasting, can also be supported by NOAH. Therefore, it is necessary to completely characterize the scope of spreadsheet operations that can be supported by NOAH. Other enhancement includes adding annotations, *e.g.*, visual cues, texts, to the overview and then exporting the customized overview—a required feature for information seeking tools [58].

Beyond Tabular Data. NOAH operates only on tabular data. However, spreadsheets can be semi-structured—formulae and text can be interspersed with tabular data. We can extend NOAH to act as a map highlighting heterogeneous regions on such complex spreadsheets which users can utilize to navigate the spreadsheet. We can leverage existing work on spreadsheet table detection [66], and property identification [67] to construct the map.

From Perceptual to Interactive Scalability. The current version of NOAH addresses the perceptual scalability challenges while navigating Excel-scale (one million rows) data. As modern spreadsheets continue to support increasingly larger datasets—DataSpread [35] supports one billion rows—the interactions proposed in this paper may violate the interactive response time bound of 500 ms [68]. This opens the door to a new set of research challenges that may range from approximate query processing to progressive data analytics.

NOAH represents our first step towards a general purpose spreadsheet navigation plug-in that can make spreadsheets more effective on large datasets that are increasingly the norm.

References

[1] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2376–2385, 2013.

[2] N. Elmqvist and J.-D. Fekete, "Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 439–454, 2009.

[3] Z. Liu, B. Jiang, and J. Heer, "immens: Real-time visual querying of big data," in *Computer Graphics Forum*, vol. 32, no. 3pt4. Wiley Online Library, 2013, pp. 421–430.

[4] "How finance leaders can drive performance," https://enterprise.microsoft.com/en-gb/articles/roles/finance-leader/how-finance-leaders-can-drive-performance/.

[5] Y. E. Chan and V. C. Storey, "The use of spreadsheets in organizations: Determinants and consequences," *Information & Management*, vol. 31, no. 3, pp. 119–134, 1996.

[6] N. Raden, "Shedding light on shadow it: Is excel running your business," *DSSResources.com*, vol. 26, 2005.

[7] E. Bakke and D. R. Karger, "Expressive query construction through direct manipulation of nested relational results," in *Proceedings of the 2016 International Conference on Management of Data*. ACM, 2016, pp. 1377–1392.

[8] B. A. Nardi and J. R. Miller, *The spreadsheet interface: A basis for end user programming*. Hewlett-Packard Laboratories, 1990.

[9] D. V. Beard and J. Q. W. II, "Navigational techniques to improve the display of large two-dimensional spaces," *Behaviour & Information Technology*, vol. 9, no. 6, pp. 451–466, 1990.

[10] D. A. Bradbard, C. Alvis, and R. Morris, "Spreadsheet usage by management accountants: An exploratory study," *Journal of Accounting Education*, vol. 32, no. 4, pp. 24–30, 2014.

[11] B. R. Lawson, K. R. Baker, S. G. Powell, and L. Foster-Johnson, "A comparison of spreadsheet users with different levels of experience," *Omega*, vol. 37, no. 3, pp. 579–590, 2009.

[12] J. Watts-Perotti and D. D. Woods, "How experienced users avoid getting lost in large display networks," *International Journal of Human-Computer Interaction*, vol. 11, no. 4, pp. 269–299, 1999.

[13] "Google sheets limit," https://support.google.com/drive/answer/37603.

[14] A. Cockburn, A. Karlson, and B. B. Bederson, "A review of overview+ detail, zooming, and focus+ context interfaces," *ACM Computing Surveys (CSUR)*, vol. 41, no. 1, p. 2, 2009.

[15] R. R. Panko, "What we know about spreadsheet errors," *Journal of Organizational and End User Computing (JOEUC)*, vol. 10, no. 2, pp. 15–21, 1998.

[16] J. Grudin, "Partitioning digital worlds: focal and peripheral awareness in multiple monitor use," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2001, pp. 458–465.

[17] K. Perlin and D. Fox, "Pad: an alternative approach to the computer interface," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, 1993, pp. 57–64.

[18] "DataSpread," https://dataspread.github.io/.

[19] M. Meyer, M. Sedlmair, P. S. Quinan, and T. Munzner, "The nested blocks and guidelines model," *Information Visualization*, vol. 14, no. 3, pp. 234–249, 2015.

[20] L. McNabb and R. S. Laramee, "Survey of surveys (sos)-mapping the landscape of survey papers in information visualization," in *Computer Graphics Forum*, vol. 36, no. 3. Wiley Online Library, 2017, pp. 589–617.

[21] J. Fulda, M. Brehmel, and T. Munzner, "Timelinecurator: Interactive authoring of visual timelines from unstructured text," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 300–309, 2015.

[22] M. Brehmer, S. Ingram, J. Stray, and T. Munzner, "Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2271–2280, 2014.

[23] J. Pretorius, H. C. Purchase, and J. T. Stasko, "Tasks for multivariate network analysis," in *Multivariate Network Visualization*. Springer, 2014, pp. 77–95.

[24] T. Höllt, N. Pezzotti, V. van Unen, F. Koning, E. Eisemann, B. Lelieveldt, and A. Vilanova, "Cytosplore: Interactive immune cell phenotyping for large single-cell datasets," in *Computer Graphics Forum*, vol. 35, no. 3. Wiley Online Library, 2016, pp. 171–180.

[25] "The inside airbnb dataset." http://insideairbnb.com/get-the-data.html.

[26] "A Lone Data Whiz Is Fighting Airbnb—and Winning," https://www.wired.com/2017/02/a-lone-data-whiz-is-fighting-airbnb-and-winning/.

[27] "Define and use names in formulas," https://www.excel-easy.com/examples/names-in-formulas.html.

[28] "Pivot Table," https://en.wikipedia.org/wiki/Pivot_table.

[29] "SUBTOTAL," https://exceljet.net/excel-functions/excel-subtotal-function.

[30] D. L. Hansen, B. Shneiderman, and M. A. Smith, *Analyzing social media networks with NodeXL: Insights from a connected world*. Morgan Kaufmann, 2010.

[31] "Google explore." https://www.techrepublic.com/article/how-to-use-the-new-google-explore-feature/.

[32] M. Joglekar, H. Garcia-Molina, and A. Parameswaran, "Smart drill-down: A new data exploration operator," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1928–1931, 2015.

[33] "Hillview." https://github.com/vmware/hillview.

[34] V. Raman *et al.*, "Scalable spreadsheets for interactive data analysis," in *ACM SIGMOD Workshop on DMKD*, 1999.

[35] M. Bendre *et al.*, "Towards a holistic integration of spreadsheets with databases: A scalable storage engine for presentational data management," in *ICDE*, April 2018.

[36] R. Rao and S. K. Card, "The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1994, pp. 318–322.

[37] B. B. Bederson, A. Clamage, M. P. Czerwinski, and G. G. Robertson, "Datelens: A fisheye calendar interface for pdas," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 11, no. 1, pp. 90–119, 2004.

[38] M. Spenke, C. Beilken, and T. Berlage, "Focus: the interactive table for product comparison and selection," in *Proceedings of the 9th annual ACM symposium on User interface software and technology*. ACM, 1996, pp. 41–50.

[39] M. Spenke and C. Beilken, "Visualization and analysis of formula one racing results with infozoom-the demo." in *INTERACT*, 2003.

[40] F. Nunez and E. Blake, "Vissh: A data visualisation spreadsheet," in *Data Visualization 2000*. Springer, 2000, pp. 209–218.

[41] M. Levoy, "Spreadsheets for images," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 1994, pp. 139–146.

[42] E.-H. Chi, P. Barry, J. Riedl, and J. Konstan, "A spreadsheet approach to information visualization," in *Information Visualization, 1997. Proceedings., IEEE Symposium on*. IEEE, 1997, pp. 17–24.

[43] K. W. Piersol, "Object-oriented spreadsheets: the analytic spreadsheet package," in *ACM Sigplan Notices*, vol. 21, no. 11. ACM, 1986, pp. 385–390.
[44] S. Kandel, A. Paepcke, M. Theobald, H. Garcia-Molina, and E. Abelson, "Photospread: a spreadsheet for managing photos," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2008, pp. 1749–1758.
[45] A. Cockburn, C. Gutwin, and J. Alexander, "Faster document navigation with space-filling thumbnails," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 1–10.
[46] S. Kratz, I. Brodien, and M. Rohs, "Semi-automatic zooming for mobile map navigation," in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*. ACM, 2010, pp. 63–72.
[47] D. Nekrasovski, A. Bodnar, J. McGrenere, F. Guimbretière, and T. Munzner, "An evaluation of pan & zoom and rubber sheet navigation with and without an overview," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 11–20.
[48] C. Plaisant, J. Grosjean, and B. B. Bederson, "Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation," in *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.* IEEE, 2002, pp. 57–64.
[49] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on*. IEEE, 2007, pp. 61–71.
[50] C. North and B. Shneiderman, "Snap-together visualization: can users construct and operate coordinated visualizations?" *International Journal of Human-Computer Studies*, vol. 53, no. 5, pp. 715–739, 2000.
[51] S. K. Badam, Z. Liu, and N. Elmqvist, "Elastic documents: Coupling text and tables through contextual visualizations for enhanced document reading," *IEEE transactions on visualization and computer graphics*, 2018.
[52] "Tableau software." https://www.tableau.com/.
[53] "Microsoft power bi." https://powerbi.microsoft.com/en-us/.
[54] 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*, vol. 24, no. 8, pp. 2339–2352, 2018.
[55] 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 & Computer Graphics*, no. 1, pp. 1–1.
[56] M. J. Norušis, *The SPSS guide to data analysis*. Spss, 1986.
[57] I. SAS, "Sas user's guide: statistics," *SAS Institute, Cary, NC*, 1985.
[58] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *The Craft of Information Visualization*. Elsevier, 2003, pp. 364–371.
[59] 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*. ACM, 2000, pp. 110–119.
[60] S. Chaudhuri, R. Motwani, and V. Narasayya, "Random sampling for histogram construction: How much is enough?" *ACM SIGMOD Record*, vol. 27, no. 2, pp. 436–447, 1998.
[61] "A Lone Data Whiz Is Fighting Airbnb—and Winning," https://en.wikipedia.org/wiki/Breadcrumb_(navigation).
[62] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.
[63] D. V. Beard and J. Q. W. II, "Navigational techniques to improve the display of large two-dimensional spaces," *Behaviour & Information Technology*, vol. 9, no. 6, pp. 451–466, 1990.
[64] "The us flight dataset." https://www.transtats.bts.gov/.
[65] A. Sarvghad, B. Saket, A. Endert, and N. Weibel, "Embedded merge & split: Visual adjustment of data grouping," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 800–809, 2018.
[66] H. Dong, S. Liu, S. Han, Z. Fu, and D. Zhang, "Tablesense: Spreadsheet table detection with convolutional neural networks," in *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.
[67] Z. Chen, S. Dadiomov, R. Wesley, G. Xiao, D. Cory, M. Cafarella, and J. Mackinlay, "Spreadsheet property detection with rule-assisted active learning," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 999–1008.
[68] Z. Liu *et al.*, "The effects of interactive latency on exploratory visual analysis," *IEEE TVCG*, vol. 20, no. 12, pp. 2122–2131, 2014.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.

APPENDIX

In this supplementary material we highlight the use cases that NOAH is suited for, the in-situ chart representations within the overview, NOAH s underlying system architecture, and statistical significance results from our user study.

NOAH Use Cases

As mentioned in Section 4, the goal of designing an overview interface is to not only enable seamless navigation but also to facilitate functionalities to help users visually seek information about the data. Therefore, we draw parallels to the typology of abstract visualization tasks [1] to identify the use cases that NOAH is best suited for. We are specifically interested in exploring the different task purposes mentioned in the typology. The task purposes can be specified at three different levels: consume and produce (high-level), search (mid-level), and query (low-level). Table 1 captures all three levels and highlights where NOAH can be used.

At a very high level, users may utilize NOAH to verify some phenomenon in the data (`discover`) or consume the data for casual interest (`enjoy`). Example of `discover` includes a journalist trying to verify whether listings in Airbnb are run as hotels (see Section 2). Moreover, users may utilize the overview to communicate details about the data (`present`). However, visualization tools are more suitable for such a purpose. Finally, users can customize (`create`) the overview to fit their exploration purpose. Here, the high-level goal is to *Produce* a new representation. However, the present version of NOAH does not allow users to add annotations, *e.g.*, visual cues, texts, to the overview (`annotate`).

Regardless of the high-level purposes, users may want to find elements of interest in the overview (*search*) and seek additional details about those elements (*query*). In Section 5, we discussed the features and interactions supported by NOAH and showed how these interactions enable users to *search* and *query* the data. Users can `browse` and `explore` the data using navigational operations, *e.g.*, clicking and zooming, to find elements of interest. Note that if users know either the identity or location of the information they are searching (*e.g.*, the name of the city in the Airbnb data), they can simply use traditional spreadsheet operations like *VLOOKUP* (`lookup`) or *CTRL+F* (`locate`) (see Table 1). On the other hand, clicking or zooming is suitable when the identity or location of the information is unknown. Therefore, NOAH complements spreadsheets in accomplishing users *search* goal via navigational operations. Once the user finds the elements of interest, they can further `summarize` the elements via aggregate columns, `compare` those elements, and `identify` the raw data corresponding to the summaries in the spreadsheet.

Explaining the Chart Representations

In this section, we explain the chart representation of all the formula categories discussed in Section 5, also displayed in Figure 4.

Summary. The result of a *summary* formula, *e.g.*, `AVERAGE`, is depicted using a horizontal bar. The bar represents the range of the data subset the bin spans with the minimum and maximum values annotated within the chart. A vertical line is used to highlight where the result lies within the range.

Conditional. The result of a *conditional* formulae, *e.g.*, `COUNTIF`, is depicted using a histogram. The histogram captures the distribution of the attribute, on which the formula has been applied, *e.g.*, the *availability* attribute discussed in Section 2. Shading is used to de-emphasize data ranges that do not satisfy the condition. A vertical line is used to highlight where the result lies within the distribution.

Frequency. The result of a *frequency* formula, *e.g.*, `mode`, is also depicted using a histogram. The bin in the histogram that contains the result is rendered with "orange" color. A vertical line is used to highlight where the result lies within the distribution.

Spread. Finally, the result of a *spread* formula, *e.g.*, `mode`, is depicted using a histogram. A similar shading technique as the *conditional* formula is used to highlight the standard deviation. The mean of the is highlighted using a vertical line.

In the chart representation mode, each entry of the aggregate column contains one additional visual cue—a color bar with shades of green on the right of the chart (see Figure 3). The darker the color, the higher the value corresponding to that entry. Users can utilize the color intensity to compare the results among different aggregate column entries.

Implementation and Architecture

In this section, we provide an overview of the infrastructure of NOAH. We integrate NOAH as a data exploration plugin withinDataSpread [18], an open-source scalable web-based spreadsheet.

.1 Underlying Data Structures

As explained in Section 5, the underlying data structure representing the overview is an in-memory equi-depth histogram. NOAH constructs the histogram on demand based on the navigation attribute. In the beginning, only the highest granularity bins are constructed. As users perform ad-hoc interactions on the data, the interface is updated on the fly. For example, when a user zooms into a specific bin, NOAH again constructs an equi-depth histogram on the data corresponding to that bin on demand. To enable seamless integration of the overview with the spreadsheet data, we leverage the hierarchical positional indexes used by DataSpread [35] to access the spreadsheet data. The index is essentially an order statistics tree [35] built on the position (*e.g.*, row number) of the spreadsheet data. For any given navigation attribute, a new positional index is constructed first. NOAH then leverages the positional mapping to access the underlying data corresponding to the navigation attribute and constructs the histogram depicting the overview. Each bin in the histogram maintains positional information regarding its elements, *i.e.*, starting and ending index of each unique element in the bin (*e.g.*, cities in the Airbnb data). Therefore, NOAH can be integrated into any spreadsheet and requires only access to the positional mapping structure of that spreadsheet.

System Architecture

We now explain the system architecture of NOAH. The NOAH client is a web-based front-end that captures user input and renders both the navigation interface and the spreadsheet based on the results returned by the back-end. The front-end is responsible for capturing user input

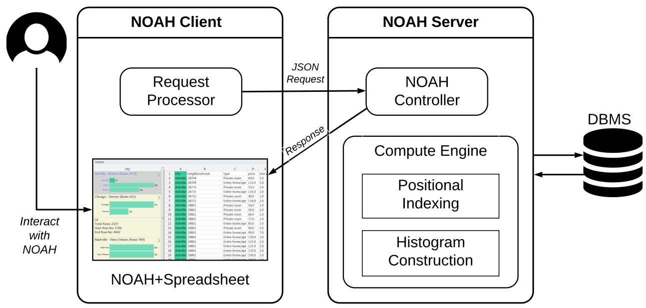


Fig. 8. System architecture.

and rendering components of the navigation interface, *i.e.*, overview, aggregate column, and context bar. Given any interaction by the user on the front end, the *request processor* issues a request to back end. The back end navigation controller receives the request from the front end. After processing the request that corresponds to some front-end user interaction, the *request processor* sends a response to the front-end encoded in *json*. For requests involving the spreadsheet, *e.g.*, scrolling, the *request processor* leverages the *positional index* to access the spreadsheet data. For requests involving the navigation interface, *e.g.*, zoom in/out, the *request processor* leverages the *compute engine* to manage the equi-depth histogram on demand. The compute engine is also responsible for processing analytical operations. We leverage DATASPREADs built-in formula engine to support the analytical operations.

RESULTS

In this section, we evaluate the statistical significance of the task performance results and survey responses presented in Section 7.

.2 Intra-participant differences.

Figure 9 depicts the intra-participant submission time differences between NOAH and Excel, across all the quiz tasks. For a given task and a participant, the corresponding circle denotes by what percentage (between 0.30% to 94.33%) a system is faster than the competing system. The color of the circle denotes which system was faster (green: NOAH faster, yellow: Excel faster). The larger the circle the faster the corresponding system is. For example, participant $P9$'s submission time was 91.78% faster with NOAH for the compare (N) task and 72.62% faster with Excel for the compare (2) task. Figure 9 confirms that majority of the submission times using NOAH were faster compared to Excel—nineteen out of the twenty participants completed at least four tasks in less time using NOAH compared to Excel. The submission time difference is even more apparent for the steer, identify, and compare (N) tasks.

For compare (2), customize, and the second steer task, less than a third of the participants' submission times were faster using Excel. For the second steer tasks, four out of the six participants that submitted answers quickly using Excel compared to NOAH, utilized the *autosum* shortcut

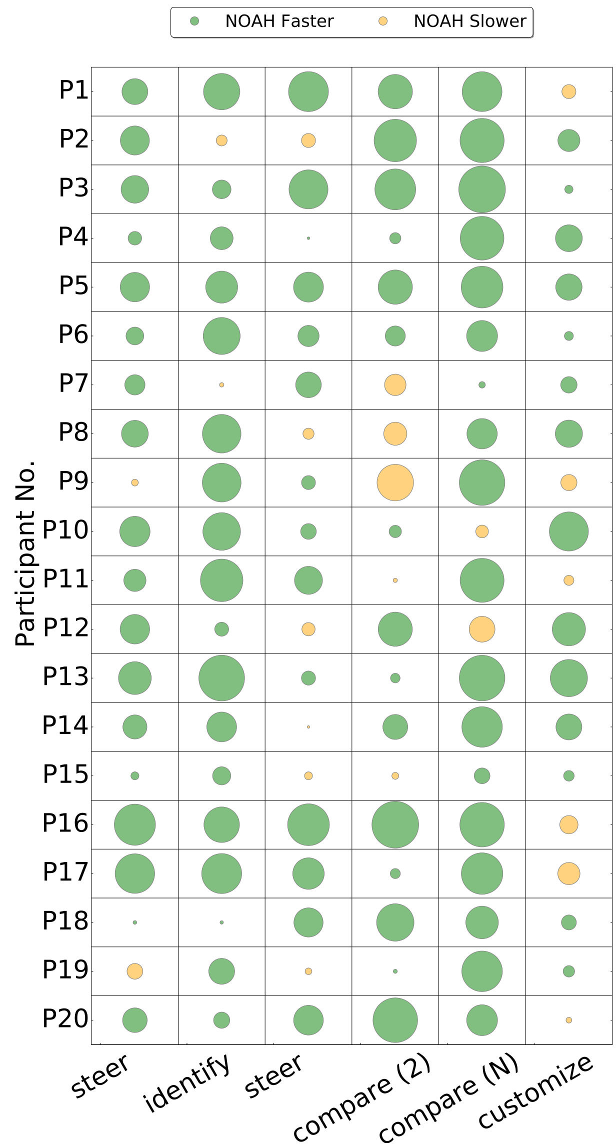


Fig. 9. Intra-participant submission time differences between NOAH and Excel across the quiz tasks.

for the birdstrikes dataset. For the compare (2) tasks, a number of participants ($N = 3$) that required more time to submit answer using NOAH, first used the bin customization feature to view all the unique bins which contributed to the higher submission time compared to Excel. On the other hand, as explained earlier, the unfamiliarity with the bin customization feature contributed to higher submission time using NOAH for six participants (yellow circles).

.3 Statistical Significance: Task Performance

Since we conducted our user study on a small population (20 participants), we further evaluated the statistical significance of the task performance results, *i.e.*, accuracy and completion time. To measure the significance of the task completion times, we ran *Mann-Whitney's U test* (as completion times did not follow a normal distribution). *For all of the tasks except the customize task, we found a significant effect of the tools, i.e., the response times for the tasks significantly differed by the choice of the tool* (see Table 3). On the other hand, we ran the *Fisher's exact test* that measures the statistical significance of categorical data (0/1 accuracy): *only for steer tasks, the percentage of the accuracy of submissions significantly differed by the choice of the tool*.

TABLE 3
Statistical significance of submission time and accuracy comparisons between NOAH and Excel. (*) indicates statistically significant.

Question	Category	Time (p value)	Accuracy (p value)
$Q1$	Steer	$\mathbf{0.0007}$ (*)	$\mathbf{0.0033}$ (*)
$Q2$	Identify	$\mathbf{2.49 \times 10^{-5}}$ (*)	0.7475
$Q3$	Steer	$\mathbf{0.0043}$ (*)	$\mathbf{0.0202}$ (*)
$Q4$	compare (2)	$\mathbf{0.0154}$ (*)	1
$Q5$	compare (N)	$\mathbf{5.83 \times 10^{-6}}$ (*)	0.48
$Q6$	Customize	0.1207	0.0959

.4 Statistical Significance: Subjective Ratings

We further conducted a statistical significance test—the *Wilcoxon Signed-rank* test—on the survey responses which showed that for all the metrics, the ratings significantly differed by the choice of the tool, *i.e.*, NOAH or Excel. The distribution of the ratings for none of the criteria followed a normal distribution.

TABLE 4
Survey results. (*) indicates statistical significance.

Metric	NOAH	Excel	p value
Ease of Learning	$\mu = 5.75$, $\sigma = 1.02$	$\mu = 4.22$, $\sigma = 1.41$	$\mathbf{1.49 \times 10^{-7}}$ (*)
Speed of Use	$\mu = 6.03$, $\sigma = 0.99$	$\mu = 4.22$, $\sigma = 1.65$	$\mathbf{1.68 \times 10^{-7}}$ (*)
Ease of Use	$\mu = 5.88$, $\sigma = 0.90$	$\mu = 4.33$, $\sigma = 1.71$	$\mathbf{7.85 \times 10^{-6}}$ (*)
Confidence	$\mu = 5.50$, $\sigma = 1.79$	$\mu = 4.60$, $\sigma = 1.50$	$\mathbf{0.0096}$ (*)
Comprehensibility	$\mu = 5.60$, $\sigma = 1.27$	$\mu = 4.48$, $\sigma = 1.65$	$\mathbf{0.0006}$ (*)
Satisfaction	$\mu = 5.48$, $\sigma = 1.16$	$\mu = 4.52$, $\sigma = 1.49$	$\mathbf{0.0018}$ (*)