Seeing it all at once: Maintaining Context for Seamless Navigation and Computation on Spreadsheets

ABSTRACT

After nearly five decades since they were first introduced, spreadsheets are still incredibly popular for ad-hoc exploration and analysis of data. Despite that, manipulating spreadsheet datasets that span more than a few screens via operations such as scrolling or issuing formulae is often overwhelming and error-prone. Users easily lose context as they scroll through the data and suffer from cognitive and mechanical burdens while issuing formulae on data spanning multiple screens. We propose new navigation workflows via a plug-in to support the seamless exploration of large spreadsheet 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. They can issue formulae over subsets of data without performing cumbersome scrolling or range selection, 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 demonstrated that NOAH made 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× fewer mistakes in NOAH while being twice as fast in completing the tasks.

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 [30]. Studies show that information workers prefer to operate on data within spreadsheets while shunning enterprise solutions with more advanced analytical features [10, 43]. A 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 [1]. 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 their popularity [34].

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"* [2]. 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 screens. 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 [7, 24]. Overall, both scrolling and steering are crucial for directly manipulating data as users perform navigation and computation on spreadsheets.

However, when spreadsheet datasets span multiple screens, the direct manipulation paradigm [49] that works so well on data that fits in a single screen starts to display substantial limitations. While navigating large datasets using scrolling or steering, spreadsheet users find it difficult to analyze, make sense of, or operate on such data [34, 56], due to multiple inter-related reasons:

- *Loss of overview and context.* When navigating spreadsheets, users can easily lose context of where they are and where they should go next [56]. 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.
- *Visual discontinuities.* The limited viewport afforded to the user introduces a visual discontinuity between 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 [34, 56]. As an alternative, users tend to copy subsets of data side by side to reduce this visual discontinuity [34, 56], which is cumbersome and error-prone.
- *Cognitive and mechanical burdens.* The lack of contextual cues leads to severe cognitive and mechanical burdens [12]. Users often end up taking drastic measures to avoid getting lost; for example, some users create personalized overviews extrinsic to the spreadsheet, by sketching maps of spreadsheets on paper [56]. Moreover, issuing a simple spreadsheet formula on data spanning multiple screens may require significant manual effort. Users may either select the data via steering, *i.e.*, dragging the mouse pointer, or enter the data range as the argument to the formula after having memorized the range. To make matters worse, both these approaches often lead to errors due to incorrect range selection [39].

Overall, while navigating and performing computation on spreadsheet datasets that span multiple screens, ***users often lose context, experience visual discontinuities, get overwhelmed, and introduce errors***. With the ease of data generation, and with spreadsheets now supporting increasingly larger datasets, *e.g.*, Google Sheets now supports five million cells [18], a 12.5× increase from the previous limit of 400K cells, *navigating data within spreadsheets is only becoming*

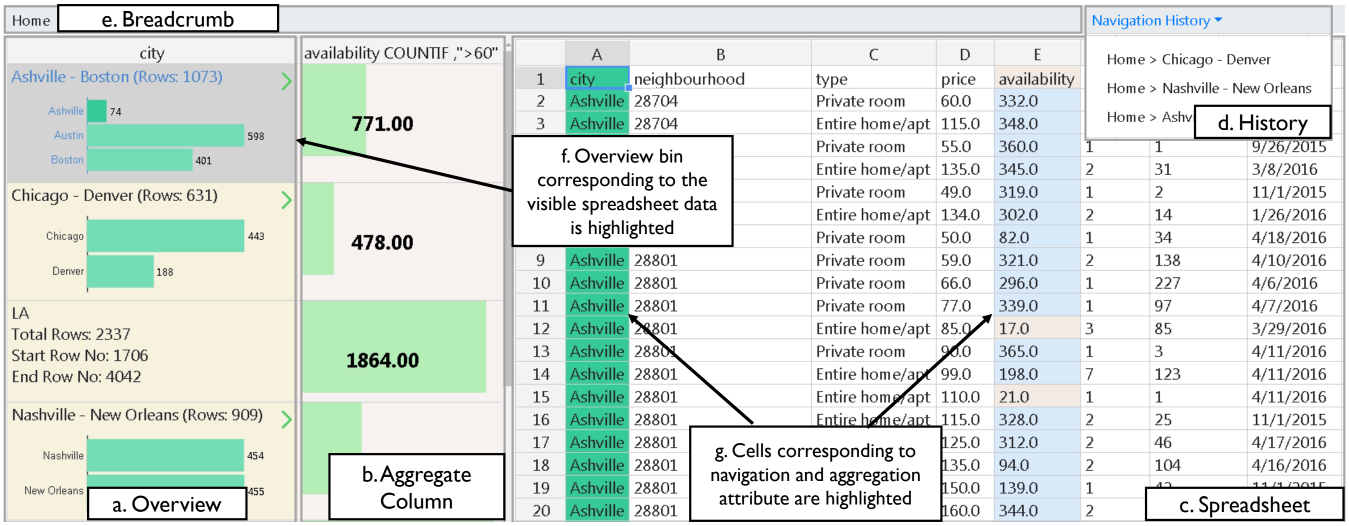


Figure 1. The NOAH interface consisting of (a) a zoomable overview and (b) an aggregate column integrated with (c) a spreadsheet. A context bar consisting of (d) a navigation history displaying locations visited so far using the overview, and (e) a breadcrumb showing the current navigation path (*e.g.*, Home). (f) The user's 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. A video demo of NOAH can be found at https://www.youtube.com/watch?v=iZsboe3x680.

harder. These datasets wouldn't necessarily be regarded as "big data" but are rather large from a perceptual standpoint: imagine scrolling through a million rows, a hundred rows at a time—this would take ten-thousand individual scrolling interactions. Despite spreadsheets being around for nearly five decades, little work has addressed the aforementioned challenges that stem from navigating datasets spanning multiple screens. Existing spreadsheet features such as pivot tables, named ranges, and subtotals, partially alleviate some of these challenges but do not eliminate them entirely. For example, pivot tables generate a static summary in a separate area of the spreadsheet while losing the correspondence between the raw data and the summary, while named ranges require users to manually associate names with ranges of data, a replacement for sketches made on paper [56].

So, how do we support more effective navigation of spreadsheet data? One approach would be to integrate an overview of the overall structure of the data along with the spreadsheet [19] resulting in a classical overview+detail interface where the spreadsheet is the detailed view. Overview+detail interfaces have been shown to be effective in various domains such as text editors and maps [12], reducing cognitive load for users during navigation by providing them with the big picture first. Such interfaces combine fine-grained information into coarse-grained high level groups, *e.g.*, combining states to form countries in Google maps, helping users quickly assimilate the information space [12].

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

- *Overview design and construction.* Given a large spreadsheet, how do we *design an overview* that captures the overall structure of the data while providing context and reducing visual discontinuities as users navigate the spreadsheet? How do we ensure that this overview *facilitates the search* for individual rows or groups of rows of interest to the user? How do we *construct a coarse-grained representation* of the overview by grouping spreadsheet rows together such that this grouping applies to all data types? How do we *dynamically adapt* the grouping modality so that the overview remains interpretable irrespective of the scale of the data?
- *Overview capabilities and integration.* It is essential that the overview we construct 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. How can we *seamlessly integrate* our overview with any current spreadsheet tools without impacting existing functionalities or look-and-feel? How do we *facilitate simple interactions* on the overview that help users avoid endless scrolling during spreadsheet exploration and error-prone steering while issuing formulae across multiple screenfuls of data? As user perform various interactions, how do we *ensure consistency* across both views, *i.e.*, the overview and the raw spreadsheet?

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 [14], 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. NOAH enables new navigation workflows for performing spreadsheet tasks involving scrolling and steering in a rapid and error-free fashion.

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 cumbersome and error-prone steering to analyze the data, users can issue formulae directly on the overview with interactions similar to pivot table construction, and view results on a separate *aggregate column*, alongside the overview (Figure 1b). Users can issue formulae on different subsets of the data while remaining on the same screen, reducing visual discontinuity. NOAH ensures coordination between the overview and the spreadsheet: for example, zooming on the overview is 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 an overview interface that enables new workflows—alternatives to cumbersome scrolling and steering—when performing navigation and computation on large spreadsheets. We realize this design in the form of NOAH, a plugin to a spreadsheet tool, ensuring that interactions supported by NOAH complement and extend 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 Excel, participants were able to complete spreadsheet navigation tasks correctly (2.5× **fewer mistakes**) and quickly (2× **faster completion time**) using NOAH.

NOAH USE CASES

NOAH introduces new exploration workflows supported by interactions on a multi-granularity overview; this multi-granularity overview in turn abstracts the raw spreadsheet data using high-level visual summaries. Therefore, to understand the scope of typical user tasks on spreadsheets supported by NOAH, we make use of the typology of abstract data exploration tasks [9]—see Table 1. This typology characterizes the range of domain-independent tasks performed on visual representations of data and has been applied to a variety of scenarios, including developing models for visualization system design [29], designing task taxonomies for cartograms [28], and defining the scope of tasks in a number of domains, *e.g.*, interactive task authoring [16], document mining [8], multivariate network analysis [42], 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 ✓). We describe these tasks in the context of a real usage scenario below.

Rebecca, a journalist, is exploring the *Inside Airbnb* dataset [13], 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 [23]. 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 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—she would have to switch back and forth between the pivot table results and the raw listings, present at disparate locations on the spreadsheet. Using a workflow possible through 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, as shown in Figure 1c, clicking on the *Ashville-Boston* bin displays the Ashville listings (`locate`). We discuss the construction of the overview and associated interactions in the following sections.

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)

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 .

Next, say Rebecca wants to analyze one of the larger cities to understand the overall renting pattern (`summarize`). 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 issue a `COUNTIF` operation on the overview (`generate`). The result is displayed as an *aggregate column* alongside the overview (Figure 1b). 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 (`identify`). As she uses the overview to navigate to other cities, *e.g.*, Chicago, and compare the rent availability (`compare`), NOAH automatically updates the aggregate column results corresponding to that city (Chicago). In traditional spreadsheets, she would have to reissue the steering operation for each city being compared from scratch, which is cumbersome.

Finally, as Rebecca navigates the data, her navigation history (Figure 1d), *i.e.*, recently visited cities, and current navigation path (Figure 1e), is 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. Maintaining the navigation history in traditional spreadsheets can be tedious as she has to manually create named-ranges. We explain this feature later.

RELATED WORK

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

Commercial Tools. Microsoft Excel enables users to manually create references to a spreadsheet region using the named ranges [45] 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 [58] and SUBTOTAL [32] features allow users to create a summary view to compare subsets of data without having to navigate to various locations within the sheet. Pivot table is placed in a separate disconnected region of the spreadsheet, preventing users from simultaneously accessing the data underlying the summary. SUBTOTAL [32], on the other hand, embeds the summary with the raw data. However, for datasets with many subsets (*e.g.*, for numeric data), both the summary views can become very large, spanning multiple screens, and cause increased visual discontinuity during navigation. Google Sheets Explore [17] provides an overview of the data by auto-generating charts of data 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 Prototypes. Smart-drill-down [21] 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 [54] displays the approximate results of group-by queries on large spreadsheet tables. ABC [44] and DATASPREAD [5] enables users to interact with very large spreadsheet datasets, beyond main-memory limits. However, none of these systems provide any new spreadsheet capabilities to assist with navigation. We build NOAH as a plugin to DATASPREAD.

Visual Interactive Spreadsheets. VisSh [37], SI [25], SSR [11], ASP [41], and PhotoSpread [22] 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, they do not necessarily help users navigate data more effectively.

Interactive Tables. TableLens [46] enables users to 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. Similar ideas have been adopted by DataLens [4] for visualizing digital calendars, and by FOCUS [52] and InfoZoom [51] for exploring database query results. However, navigating (scrolling or steering) data that spans multiple screens is still cumbersome in TableLens.

Tabular Data Analysis (TDA) Tools. Visualization tools such as Tableau [53], Power BI [31], Keshif [60] and analytical tools such as SPSS [36], SAS [48], 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 [60] can display all the unique values corresponding to an attribute of interest, *e.g.*, cities of the Airbnb data [13]. 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.

Overall, none of the existing systems provide a unified interface that enhances navigational capabilities of users while simultaneously upholding the following goals of spreadsheet tools: a) facilitating the direct manipulation of data in-situ and b) enabling arbitrary derivation of new data and summaries using various navigational operations, *e.g.*, issuing formulae. We discuss how NOAH achieves these goals next.

NOAH: DESIGN CONSIDERATIONS

In this section, we outline our design considerations for creating new spreadsheet navigation workflows aimed at eliminating cumbersome scrolling and steering. We then operationalize these workflows in NOAH. Our design considerations were informed by prior work on information visualization [50, 9], overview+detail interfaces [12], multiple-coordinated views [55], 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 [19].

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 [50]. 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 [47], *i.e.*, an interaction on one should be reflected on the other [55]. 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 [44]. 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 [50], while also displaying their current navigation path for context.

USER INTERFACE

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

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 [40]. 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 later.

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 [12]. 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 [15]. Thus, the multi-granularity overview of NOAH provides an alternative to the 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 [12]. Users can resize the overview to control the amount of spreadsheet data that remains visible.

The data structure underlying the overview is a histogram constructed on the values in the navigation attribute column. Histograms result from *binned aggregation*—consecutive data points are grouped into bins (or groups), where each bin represents a collection (group) of data points and is associated with a count aggregate, capturing the number of data points that fall in that group. In addition to providing high level (*e.g.*, densities) and low level (e.g., outliers) details, binned aggregation techniques enable a multi-granularity visual representation of data by varying the bin size and have therefore been used in interactive visualization of large scale datasets, *e.g.*, in *imMens* [27]. An additional benefit of a binned overview for spreadsheets is a decrease in visual discontinuity during navigation. As users are able to view an overview that fits in the computer screen, they 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. To generate the overview, we construct an equi-depth histogram. Equi-depth histograms are commonly used for summarizing statistical properties of data, with applications in database systems for query optimization and approximate query processing [20] and in data mining applications for distribution fitting in data streams [33], among others. 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 earlier, the jour-

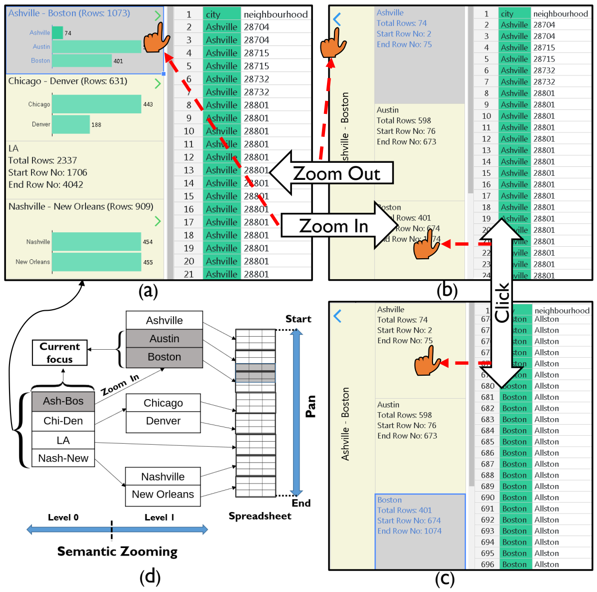


Figure 2. Navigational operations. (a) The overview at the highest level of granularity. (b) A zoomed in view of the *Ashville-Boston* bin. (c) As the user clicks on the *Boston* bin, the Boston listings are displayed on the sheet. The *Boston* bin is highlighted in gray to indicate user's current focus.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

nalist grouped the data into cities 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 initially computes the highest granularity of the overview. As users perform ad-hoc interactions on the overview, the interface is updated on the fly. For example, when a user zooms into a specific bin, NOAH constructs an equi-depth histogram on the data corresponding to that bin on demand. NOAH divides each bin 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.

Operations and Interactions

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

Navigation via Clicking. Clicking a bin is an example of a coupled interaction as the user actively changes the focus to another bin on the overview. 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). Conversely, as the user scrolls up on the spreadsheet, both Austin and Boston listings appear in the screen. As the current focus changes, both the Austin and Boston overview bins are highlighted (see Figure 2d). 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.

Navigation via Semantic Zooming. Users can zoom into a bin to view more fine-grained information or zoom out to view more coarse-grained information, via semantic zooming [40]. The zooming operation is decoupled—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, from the bin *Ashville-Boston* when the user zooms in to the next level, NOAH displays the bins *Ashville*, *Austin*, and *Boston* (Figure 2b); here, the spreadsheet view stays the same. If the user zooms out of the current granularity, again NOAH displays the bins *Ashville-Boston*, *Chicago-Denver*, and others. The zoom out operation is also decoupled—when a user zooms out, the overview displays a coarser granularity view of the user's current focus.

Customizing the Overview. As NOAH constructs the overview automatically, the overview binning 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 then 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.

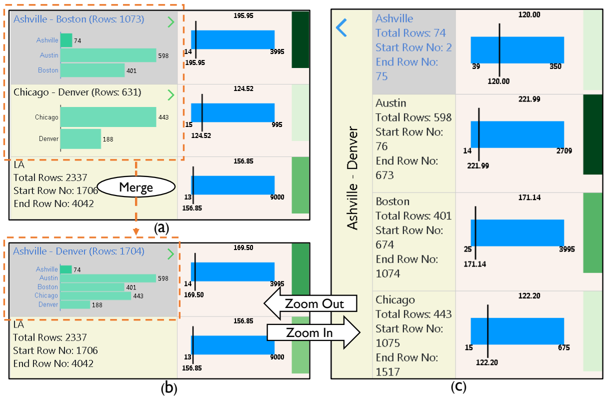


Figure 3. (a) Chart view of the aggregate column. (b) A new bin is created by merging the top two bins. (c) Zooming into the newly created bin.

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 view the results either as raw values (see Figure 1b) or as charts (see Figure 3), and can toggle between the two. Users can issue several formulae simultaneously, each creating 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. 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.

Context Bar

The context bar consists of two components: a) a breadcrumb, and b) a navigation history. The breadcrumb [57] 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).

Implementation

We have integrated NOAH with DATASPREAD [5], 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 [6] for generating charts.

To enable seamless integration of the overview with the spreadsheet data, we leverage a positional index used by DATASPREAD [5] that allows data to be referenced by position. The index is an order statistics tree [5] on the position, *i.e.*, *row id*. For any given navigation attribute, for example "city", the data is first sorted by the attribute and then a new positional index is constructed. NOAH leverages this index to retrieve the entire collection of (*row id, attribute value*) pairs and then constructs a best effort equi-depth histogram (discussed earlier) on the attribute values, *e.g.*, city names, to depict the overview. As the data is sorted by the attribute value and each value is associated with a *row id*, each bin in the overview can be mapped to a sequence of *row id*s. When users click on a bin, NOAH uses this mapping to display relevant spreadsheet rows. Similar ideas can be adopted to integrate NOAH into any existing spreadsheet tools by providing access to the underlying positional mapping structure.

EVALUATION STUDY DESIGN

In this section, we present the design of a user study aimed at understanding the impact of our in-situ plugin while performing navigation and computation on spreadsheet data. Existing spreadsheet tools do not employ specific plug-ins for navigation. Since there is no equivalent tool for comparison, we studied how participants' spreadsheet navigation experience vary with the presence or absence of such a plug-in. That is, we compared a NOAH-integrated spreadsheet tool with a popular spreadsheet tool, Microsoft Excel, across various tasks. (Participants were free to use any Excel capabilities, including subtotals, pivot tables, and named ranges that partially address some of the navigational concerns in conventional spreadsheets.) These tasks were representative of the use cases presented in Table 1. Similar domain specific-evaluations have been performed for evaluating various overview+detail interfaces, *e.g.*, database browsing [35] or tree navigation [3]. Our study was designed to answer the following questions:

- **RQ1.** How does the integration of an overview plugin like NOAH impact the usability and efficiency of navigation workflows in spreadsheet tools?

- **RQ2.** How do the various components of NOAH impact users' navigational experiences?

Study Design and Participants. The two tools used in the study were: the Microsoft Excel spreadsheet tool, and NOAH integrated as a plug-in within the DATASPREAD [14] spreadsheet tool (henceforth, referred to as NOAH). We chose Excel for our comparative study because it is the most popular spreadsheet in use today; we chose DATASPREAD because the source code was open sourced allowing an easy integration. We recruited 20 participants (11 female, 9 male) via a university email newsletter. To ensure that prior experience with spreadsheets didn't affect the performance of participants 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) while signing up for the study. All of the participants were familiar with the basic mathematical and statistical functions supported by Excel. The study consisted of three phases: (a) an introductory phase to help participants familiarize themselves with NOAH, (b) a quiz phase where the participants used both the tools to perform targeted tasks on two different datasets (described later), and (c) a semi-structured interview to collect qualitative feedback regarding the quiz phase. The study protocol along with the list of tasks, surveys, and interview questionnaires are included in the supplementary material.

Datasets. We used two datasets—the birdstrikes [60, 59], and the Airbnb [13] 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 enforce parity among datasets, 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 has 15 attributes (six categorical, two spatial region, one temporal, six numeric).

Study Procedure

We now explain each of the phases of our study.

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 [38]. The participants then explored the same dataset using NOAH to familiarize themselves with the tool for about 10 minutes.

Phase 2: The Quiz Phase. The purpose of the quiz phase was to evaluate the effectiveness of NOAH in addressing spreadsheet navigation limitations. Each participant performed specific tasks on the two datasets in two sessions, using Excel for one and NOAH for the other. We alternated the order of the datasets between consecutive participants. The order of the tools was alternated between every two participants. Each session was followed by a survey, described later. 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.

Quiz Tasks. For each dataset, we designed six tasks across five categories: steer (two tasks), find (one task), Compare (2) (one task), Compare (*N*) (one task), and customize (one task), encompassing six of the seven task typology use cases underlying the *Search*, *Query*, and *Produce* purposes: `lookup`/`locate`, `identify`, `browse`, `compare`, `summarize`, and `generate` (see Table 1). These tasks were selected to mimic a typical spreadsheet analysis workflow and are representative of navigation interactions required for the most frequently issued spreadsheet operations [7, 24]. The tasks were presented in the same order as shown in Table 2 for the birdstrikes dataset. The tasks for the Airbnb dataset mimic a scenario similar to the example in the usage scenario section.

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

Table 2. Quiz tasks for the birdstrikes dataset. The task purposes and use cases correspond to the task typology discussed in the usage scenario.

Survey. After each session, participants rated the corresponding tool 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).

Evaluation. We evaluated the accuracy and completion time for all of the tasks. We combined this analysis with the findings from a qualitative survey, interview, and screen/audio recording data to provide insights that can be corroborated across multiple sources. Moreover, we analyzed the survey responses to quantify the usability of both the tools. We then measured the statistical significance of the comparisons between the two systems.

Phase 3: Interview Phase. 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 features present NOAH and Excel.

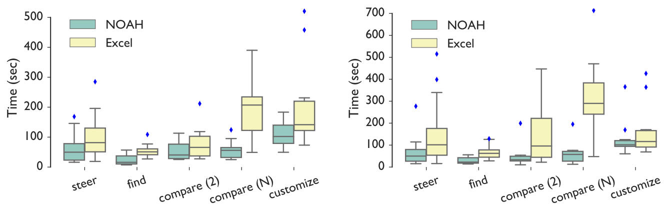
RESULTS

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

RQ1. Impact of overview-spreadsheet integration on navigation performance and spreadsheet usability

To answer RQ1, we first compare task completion times and accuracies in NOAH and Excel and then analyze the survey responses that evaluate the usability of the tools.

Faster navigation without sacrificing accuracy

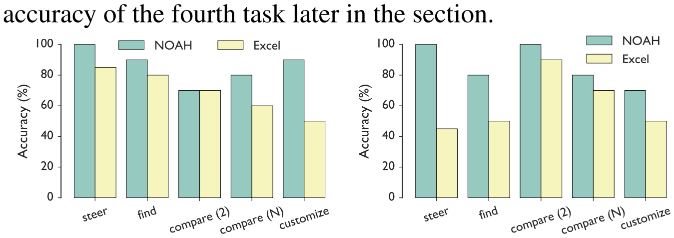


(a) birdstrikes **(b) airbnb**

Figure 4. Submission times per category for each dataset. Median submission times are much smaller for NOAH compared to Excel.

In Figure 4a and 4b, we show the distribution of submission times of participants for the five 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 observation suggests that the capabilities offered by NOAH made spreadsheet navigation faster for these tasks. The majority of submission times using NOAH were faster than Excel—19 out of the 20 participants completed at least four tasks in less time using NOAH.

In Figure 5a and 5b, 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. We discuss the accuracy of the fourth task later in the section.



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

Figure 5. Per category accuracy for each dataset. Participants attained higher accuracy while completing tasks in NOAH compared to Excel.

Statistical Significance. We further evaluated the statistical significance of the task performance results, *i.e.*, completion time and accuracy. 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). We ran the *Fisher's exact test* that measures the statistical significance of categorical data (0/1 accuracy): *the percentage of the accuracy of submissions significantly differed by the choice of the tool for the steer tasks only*.

Question	Category	Time ($p < 0.05$)	Accuracy ($p < 0.05$)
$Q1$	Steer	**0.0007** (*)	**0.0033** (*)
$Q2$	Identify	$\mathbf{2.49 \times 10^{-5}}$ (*)	0.7475
$Q3$	Steer	**0.0043** (*)	**0.0202** (*)
$Q4$	Compare (2)	**0.0154** (*)	1
$Q5$	Compare (N)	$\mathbf{5.83 \times 10^{-6}}$ (*)	0.48
$Q6$	Customize	0.1207	0.0959

Table 3. Statistical significance of submission time and accuracy between NOAH and Excel. (*) indicates statistical significance.

Participants preferred NOAH *to Excel*

Table 4 shows the results of the survey in which participants rated their experience with NOAH and Excel. Notably, NOAH had a higher average rating than Excel for all the metrics. 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.

Metric	NOAH	Excel	$p < 0.05$
Ease of Learning	$\mu = \mathbf{5.75}$, $\sigma = 1.02$	$\mu = 4.22$, $\sigma = 1.41$	$\mathbf{1.49 \times 10^{-7}}$ (*)
Speed of Use	$\mu = \mathbf{6.03}$, $\sigma = 0.99$	$\mu = 4.22$, $\sigma = 1.65$	$\mathbf{1.68 \times 10^{-7}}$ (*)
Ease of Use	$\mu = \mathbf{5.88}$, $\sigma = 0.90$	$\mu = 4.33$, $\sigma = 1.71$	$\mathbf{7.85 \times 10^{-6}}$ (*)
Confidence	$\mu = \mathbf{5.50}$, $\sigma = 1.79$	$\mu = 4.60$, $\sigma = 1.50$	**0.0096** (*)
Comprehensibility	$\mu = \mathbf{5.60}$, $\sigma = 1.27$	$\mu = 4.48$, $\sigma = 1.65$	**0.0006** (*)
Satisfaction	$\mu = \mathbf{5.48}$, $\sigma = 1.16$	$\mu = 4.52$, $\sigma = 1.49$	**0.0018** (*)

Table 4. Participants found NOAH to be easier to use compared to Excel while being faster in completing tasks involving navigation. (*) indicates that the difference in rating is statistical significance.

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

To answer RQ2, we assess how NOAH's components, *i.e.*, the binned overview, aggregate column, and context bar, impacted participants' navigation. For each observation, we present participant feedback from the interview phase.

Binned Overview: Customizable Hierarchical Organization

Overall, the binned overview prevented participants from being overwhelmed during navigation, especially at scale. Personalizing the overview enabled participants to define their own grouping of the data, resulting in a more meaningful overview presentation. However, the newer interactions at times deviated from spreadsheet semantics, contributing to a steeper learning curve.

Overviews aid navigation at scale. Participants found it difficult to perform various navigation tasks in Excel, especially at scale. NOAH, on the other hand, helped participants avoid endless scrolling via clicking and semantically zooming on the overview, and provided cues for what to explore next via the bins of the overview. 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 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. 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.*"

Overview customization enables related data to be analyzed together in task-specific ways. Bin customization enabled participants 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.*" Participants (14 out of 20) 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*" (*P*17). Our analysis of the video recordings revealed that for the birdstrikes dataset in Excel, the customize task involved filtering out certain values from a total of 451 unique values. This manual filtering led to a significant delays in task completion, compared to the bin customization feature in NOAH. However, the time taken for this task was higher than other tasks in NOAH, as it required participants to restructure the overview before performing any calculation.

Overview customization interactions have a steeper learning curve. Unfamiliarity with the customization interactions in NOAH contributed to a higher completion time for the customize task compared to other tasks. The unfamiliarity led to some participants ($N = 5$) preferring Excel over NOAH for this task—"*Since I'm not used to spreadsheet data being presented that way, it took a little bit of getting used to*" (*P*11). Moreover, two participants requested further clarification on how the bins were constructed during the interview.

Tradeoffs between hierarchical and flat overviews. While participants generally appreciated the binned representation of the overview for numeric data, six participants stated that they would have preferred a non-hierarchical overview for categorical data, where each bin corresponds to one item. One participant (*P*13) commented: "*I would prefer it start with all the bins split, and then I can merge them as I want.*"

Aggregate Column: In-situ Steering-free Computation

The aggregate column feature enabled participants to avoid cumbersome steering interactions, resulting in faster and more accurate analysis compared to Excel. However, comparing the analysis results of more than two data subsets resulted in increased visual discontinuity.

Cumbersome steering replaced by a few button clicks with aggregate columns. 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*" (*P*4). With NOAH, participants avoided steering by using aggregate column feature on the menu-bar and selecting the appropriate formula. Multiple participants ($N = 13$) found it easier to issue formulae using this feature. One participant (*P*13) 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.

Issuing formulae is faster and more accurate with aggregate columns. 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 4 and 5). The automated and steering-free aggregate column feature of NOAH contributed to high accuracies (100%) for the steer tasks. One participant (*P*12) 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 an incorrect spreadsheet region; 11 of the inaccurate submissions were with the Airbnb dataset. Analysis of screen recordings revealed that, for Excel, with the birdstrikes dataset, several participants used the *autosum* feature to quickly count the number of 1's in the binary-valued column 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 resulting in fewer errors. For the Airbnb dataset, participants had to manually steer the data as they could not use these shortcuts—the column involved in the steering task was non-binary (it had 365 different values).

Visual discontinuity during comparison while reduced, was not completely eliminated. For Compare (N) tasks, participants had to perform N comparisons in NOAH while issuing the aggregate column operation once. However, the 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 separate steering tasks. As a result, in Excel, the Compare (N) task submission times were very high compared to Compare (2) tasks (see Figure 4). In addition, the accuracies of the Compare (N) task in Excel were lower ($N = 7$ out of 20 submissions were inaccurate).

History, Context, and Coordination

The context bar enabled participants to revisit aggregation results of previously explored bins without having to reissue the aggregate column operation. The coordination between the overview and the raw spreadsheet data further helped participants relate the aggregate column results with the raw data.

History helps avoid repeated interactions. For the Compare (2) task in NOAH, all of the participants used the context bar to navigate to a bin previously visited for the first steer task. As the bin currently being displayed was changed, the aggregate column was automatically updated to display values corresponding to that bin, enabling participants to view the aggregate column values instantly without having to reissue the operation. On the other hand, 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 faster in NOAH compared to Excel (see Figure 4). One participant (*P*9) said—"*Noah was easy to find and compare and toggle in between (bins).*"

Overview-spreadsheet coordination helps relate interactions on the overview with the raw data. The coordination between the overview and spreadsheet in NOAH enabled users to quickly relate their interactions on the overview with the raw spreadsheet data. For example, for the find task, participants had to find all the cells within the spreadsheet that satisfied a condition corresponding to the preceding steer task. To do so, they had to either perform the conditional formatting operation to highlight relevant cells or skim through all the cells in the current window in Excel, resulting in higher completion times. In NOAH, participants benefited from having visual cues in the form of automatically colored cells, helping them relate the aggregate column with the raw data—"*In Excel, you have to add your own condition for formatting. But you have to build that (conditional formatting) every time you need to ask a question.* NOAH *at least (has) something pre-built in, and you can easily count*" (*P*5).

DISCUSSION AND FUTURE WORK

In this section, we highlight the limitations of our user study, discuss the implication of the study takeaways for existing spreadsheet tools, and propose extensions to NOAH.

User Study Limitations

Our study has a few limitations that can be strengthened by future larger-scale and finer-grained studies. Firstly, our participant pool demographics only partially represented the demographics of the general audience intended for NOAH. A larger sample with more participants with a range of skillsets and backgrounds that better represents the spreadsheet user population would have provided more ecological validity to generalize our findings. Secondly, 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. They violated many of the proposed design considerations; we discussed their limitations in the related work section. However, a future study targeted at evaluating the pros and cons of these features for spreadsheet navigation would be valuable. Finally, while we did present the impact of various components of NOAH, we did not isolate the effects of individual features during our study. For example, we did not study the effects of the binned overview (visual clarity versus visual continuity) and display layout (screen space trade-off) in isolation. Therefore, a more fine-grained study that teases apart the contribution of individual components of NOAH is warranted.

Implication for Existing Spreadsheet Tools

Existing spreadsheet tools like Excel and Google Sheets can benefit from adopting the novel navigation workflows proposed in this paper by integrating NOAH. From a data model standpoint, the multi-granularity overview of NOAH can be constructed by leveraging the built-in positional mapping structure already used by the spreadsheet tools for recording and maintaining position. Since NOAH is a plug-in to spreadsheets as opposed to operating entirely within the sheets, NOAH doesn't require substantial changes to the core spreadsheet internals, and can be engineered to act externally to the sheet, while still reading spreadsheet data and triggering spreadsheet computation. Instead, what would be necessary would be external coordination modules that synchronize operations on NOAH with the sheet; for example, a click on the overview should trigger a scroll in the sheet. Scrolling on the spreadsheet should not only display new data but also update the bin highlighted in the overview.

Limitations of NOAH and Possible Enhancements

We now discuss the limitations of NOAH and how we can possibly overcome them.

Enabling more flexible overview binning. The experience surrounding the construction of the overview can be further improved, especially for categorical data. Currently, the bins of the overview can be customized only after the overview is constructed. Providing the users the ability to select the representation (similar to bin customization) of the overview at the outset could have possibly addressed this issue. Understanding the impact of these representation choices for the overview is an interesting open question. Moreover, 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).

Broadening the scope of overview-spreadsheet coordination. Spreadsheet users often perform various edit operations, *e.g.*, updating values, adding/deleting rows/columns. However, our current NOAH implementation assumes the data to be read-only. In our next version, we can add support for propagating 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 [60].

Achieving scalability beyond traditional spreadsheet limits. The current version of NOAH addresses the navigation challenges for Excel-scale (one million rows) data. As modern spreadsheets continue to support increasingly larger datasets—DATASPREAD [5] supports one billion rows—the interactions proposed in this paper may violate the interactive response time bound of 500 ms [26]. This opens the door to a new set of research challenges that may range from approximate query processing to progressive data analytics.

CONCLUSION

In this paper, we proposed new navigation workflows to make spreadsheets more effective at supporting the exploration of large datasets that are increasingly the norm. We operationalized these workflows in NOAH, a general purpose spreadsheet navigation plug-in. Using NOAH, users were able to see a bird's eye view of the data, with the ability to scroll or seek additional details on demand via a multi-granularity overview, as well as employ aggregation in-situ, eliminating cumbersome steering operations. Quantitatively, we found that NOAH sped up navigation without compromising accuracy. Qualitatively, study participants identified NOAH as positively impacting their experience while overviewing, navigating, and performing computation on large datasets. Finally, we identified several enhancements to NOAH while discussing how the proposed navigation workflows can be integrated into existing spreadsheet tools.

REFERENCES

[1] Eirik Bakke and David R Karger. 2016. Expressive query construction through direct manipulation of nested relational results. In *Proceedings of the 2016 International Conference on Management of Data*. ACM, 1377–1392.

[2] David V Beard and JOHN Q WALKER II. 1990a. Navigational techniques to improve the display of large two-dimensional spaces. *Behaviour & Information Technology* 9, 6 (1990), 451–466.

[3] David V Beard and JOHN Q WALKER II. 1990b. Navigational techniques to improve the display of large two-dimensional spaces. *Behaviour & Information Technology* 9, 6 (1990), 451–466.

[4] Benjamin B Bederson, Aaron Clamage, Mary P Czerwinski, and George G Robertson. 2004. DateLens: A fisheye calendar interface for PDAs. *ACM Transactions on Computer-Human Interaction (TOCHI)* 11, 1 (2004), 90–119.

[5] Mangesh Bendre and others. 2018. Towards a Holistic Integration of Spreadsheets with Databases: A Scalable Storage Engine for Presentational Data Management. In *ICDE*.

[6] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics* 17, 12 (2011), 2301–2309.

[7] David A Bradbard, Charles Alvis, and Richard Morris. 2014. Spreadsheet usage by management accountants: An exploratory study. *Journal of Accounting Education* 32, 4 (2014), 24–30.

[8] Matthew Brehmer, Stephen Ingram, Jonathan Stray, and Tamara Munzner. 2014. Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2271–2280.

[9] Matthew Brehmer and Tamara Munzner. 2013. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2376–2385.

[10] Yolande E Chan and Veda C Storey. 1996. The use of spreadsheets in organizations: Determinants and consequences. *Information & Management* 31, 3 (1996), 119–134.

[11] EH-H Chi, Phillip Barry, John Riedl, and Joseph Konstan. 1997. A spreadsheet approach to information visualization. In *Information Visualization, 1997. Proceedings., IEEE Symposium on*. IEEE, 17–24.

[12] Andy Cockburn, Amy Karlson, and Benjamin B Bederson. 2009. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)* 41, 1 (2009), 2.

[13] Murray Cox. 2020. The inside airbnb dataset. (2020). Retrieved May 5, 2020 from `http://insideairbnb.com/get-the-data.html`

[14] DataSpread. 2020. Unifying spreadsheets and databases. (2020). Retrieved May 5, 2020 from `https://dataspread.github.io/`

[15] Niklas Elmqvist and Jean-Daniel Fekete. 2009. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics* 16, 3 (2009), 439–454.

[16] Johanna Fulda, Matthew Brehmel, and Tamara Munzner. 2015. TimeLineCurator: Interactive authoring of visual timelines from unstructured text. *IEEE transactions on visualization and computer graphics* 22, 1 (2015), 300–309.

[17] Google. 2020a. Google Explore. (2020). Retrieved May 5, 2020 from `https://support.google.com/docs/answer/6280499`

[18] Google. 2020b. Google Sheets scale. (2020). Retrieved May 5, 2020 from `https://developers.google.com/drive/answer/37603`

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

[20] Yannis Ioannidis. 2003. The history of histograms (abridged). In *Proceedings 2003 VLDB Conference*. Elsevier, 19–30.

[21] Manas Joglekar, Hector Garcia-Molina, and Aditya Parameswaran. 2015. Smart drill-down: A new data exploration operator. *Proceedings of the VLDB Endowment* 8, 12 (2015), 1928–1931.

[22] Sean Kandel, Andreas Paepcke, Martin Theobald, Hector Garcia-Molina, and Eric Abelson. 2008. Photospread: a spreadsheet for managing photos. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1749–1758.

[23] Miranda Katz. 2017. A Lone Data Whiz Is Fighting Airbnb—and Winning. (2017). Retrieved May 5, 2020 from `https://www.wired.com/2017/02/a-lone-data-whiz-is-fighting-airbnb-and-winning/`

[24] Barry R Lawson, Kenneth R Baker, Stephen G Powell, and Lynn Foster-Johnson. 2009. A comparison of spreadsheet users with different levels of experience. *Omega* 37, 3 (2009), 579–590.

[25] Marc Levoy. 1994. Spreadsheets for images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 139–146.

[26] Zhicheng Liu and others. 2014. The effects of interactive latency on exploratory visual analysis. *IEEE TVCG* 20, 12 (2014), 2122–2131.

[27] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. 2013. imMens: Real-time Visual Querying of Big Data. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 421–430.

[28] Liam McNabb and Robert S Laramee. 2017. Survey of Surveys (SoS)-Mapping The Landscape of Survey Papers in Information Visualization. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 589–617.

[29] Miriah Meyer, Michael Sedlmair, P Samuel Quinan, and Tamara Munzner. 2015. The nested blocks and guidelines model. *Information Visualization* 14, 3 (2015), 234–249.

[30] Microsoft. 2020a. How finance leaders can drive performance. (2020). Retrieved May 5, 2020 from `https://enterprise.microsoft.com/en-gb/articles/roles/finance-leader/how-finance-leaders-can-drive-performance/`

[31] Microsoft. 2020b. Power BI. (2020). Retrieved May 5, 2020 from `https://powerbi.microsoft.com/en-us/`

[32] Microsoft. 2020c. SUBTOTAL. (2020). Retrieved May 5, 2020 from `https://support.office.com/en-us/article/subtotal-function-7b027003-f060-4ade-9040-e478765b9939`

[33] Hamid Mousavi and Carlo Zaniolo. 2011. Fast and accurate computation of equi-depth histograms over data streams. In *Proceedings of the 14th International Conference on Extending Database Technology*. ACM, 69–80.

[34] Bonnie A Nardi and James R Miller. 1990. *The spreadsheet interface: A basis for end user programming*. Hewlett-Packard Laboratories.

[35] Chris North and Ben Shneiderman. 2000. Snap-together visualization: can users construct and operate coordinated visualizations? *International Journal of Human-Computer Studies* 53, 5 (2000), 715–739.

[36] Marija J Norušis. 1986. *The SPSS guide to data analysis*. Spss.

[37] Fabian Nunez and Edwin Blake. 2000. Vissh: A data visualisation spreadsheet. In *Data Visualization 2000*. Springer, 209–218.

[38] Bureau of Transportation Statistics. 2020. The US flight dataset. (2020). Retrieved May 5, 2020 from `https://www.transtats.bts.gov/`

[39] Raymond R Panko. 1998. What we know about spreadsheet errors. *Journal of Organizational and End User Computing (JOEUC)* 10, 2 (1998), 15–21.

[40] Ken Perlin and David Fox. 1993. Pad: an alternative approach to the computer interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, 57–64.

[41] Kurt W Piersol. 1986. Object-oriented spreadsheets: the analytic spreadsheet package. In *ACM Sigplan Notices*, Vol. 21. ACM, 385–390.

[42] Johannes Pretorius, Helen C Purchase, and John T Stasko. 2014. Tasks for multivariate network analysis. In *Multivariate Network Visualization*. Springer, 77–95.

[43] Neil Raden. 2005. Shedding light on shadow IT: Is Excel running your business. *DSSResources.com* 26 (2005).

[44] Vijayshankar Raman and others. 1999. Scalable Spreadsheets for Interactive Data Analysis. In *ACM SIGMOD Workshop on DMKD*.

[45] Named ranges. 2020. Define and use names in formulas. (2020). Retrieved May 5, 2020 from `https://www.excel-easy.com/examples/names-in-formulas.html`

[46] Ramana Rao and Stuart K Card. 1994. 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, 318–322.

[47] Jonathan C Roberts. 2007. 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, 61–71.

[48] Institute SAS. 1985. SAS user's guide: statistics. *SAS Institute, Cary, NC* (1985).

[49] Ben Shneiderman. 1993. 1.1 direct manipulation: a step beyond programming languages. *Sparks of innovation in human-computer interaction* 17 (1993), 1993.

[50] Ben Shneiderman. 2003. The eyes have it: A task by data type taxonomy for information visualizations. In *The Craft of Information Visualization*. Elsevier, 364–371.

[51] Michael Spenke and Christian Beilken. 2003. Visualization and Analysis of Formula One Racing Results with InfoZoom-the Demo.. In *INTERACT*.

[52] Michael Spenke, Christian Beilken, and Thomas Berlage. 1996. FOCUS: the interactive table for product comparison and selection. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*. ACM, 41–50.

[53] Tableau. 2020. Tableau Software. (2020). Retrieved May 5, 2020 from `https://www.tableau.com/`

[54] VMware. 2020. Hillview. (2020). Retrieved May 5, 2020 from `https://github.com/vmware/hillview`

[55] Michelle Q Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. 2000. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*. ACM, 110–119.

[56] Jennifer Watts-Perotti and David D Woods. 1999. How experienced users avoid getting lost in large display networks. *International Journal of Human-Computer Interaction* 11, 4 (1999), 269–299.

[57] Wikipedia. 2020a. Breadcrumb. (2020). Retrieved May 5, 2020 from https://en.wikipedia.org/wiki/Breadcrumb_(navigation)

[58] Wikipedia. 2020b. Pivot Table. (2020). Retrieved May 5, 2020 from https://en.wikipedia.org/wiki/Pivot_table

[59] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization & Computer Graphics* 1 (2016), 1–1.

[60] Mehmet Adil Yalçın, Niklas Elmqvist, and Benjamin B Bederson. 2018. Keshif: Rapid and expressive tabular data exploration for novices. *IEEE transactions on visualization and computer graphics* 24, 8 (2018), 2339–2352.